About This Document

Background

The DESpot help browser was created as a summer undergraduate research position in the Department of Computing and Software at McMaster University. Funding provided by the Department and by the National Science and Engineering Research Council of Canada (NSERC).

Using This Document

Traversing this document is made easy by providing the user an index on the left hand side which lists all the pages contained in this help file. Several hyperlinks are also distributed among most of the pages which link to different sections of the document. Back, Forward and Home buttons as well as a search bar are provided at the top to quickly find what you are looking for. A search is performed on all the pages of the help browser using exactly what you type in (case insensitive). You can search for multiple terms by separating each search term with a comma. For example, if you search for "DESpot help", the search results will only contain pages with these two words together in that order. If you search for "DESpot, help" then you will be returned pages that contain both these words somewhere in the document. Throughout this document you will find screenshots of the DESpot application. These can be clicked on to enlarge the image for easier viewing. You can also use the print button in the toolbar to print the current page or the entire document.

Printing This Document

To print this document click on the print button located in the toolbar of the Despot Help Browser. You have the option to print the current page you are viewing or the entire Despot Help document. You can also select whether you want images to be included in the print. If you are printing the entire document, you can chose **Compact Form**. By default, page breaks occur at the end of each topic when printing the entire document. If you select **Compact Form**, the page breaks will be removed and a new topic will occur directly after the previous topic has ended. Once you have selected the options you wish, click on the **Print** button and the Print Dialog will appear. This is where you can select your printer and printer settings.

At this point, you have the option to print the document to a PDF. To print to a PDF file, click on the … button located to the right of the **Output File** text box. This will open a file browser where you can choose the name and filepath for the PDF. When you click save, you will be redirected back to the Print Dialog and now the **Printer Name** will be **Print to File (PDF)**. Clicking **Print** will output the document to the selected printer.

Despot Help File

Primary Author: Adam Brousseau

Supervisor: Dr. Ryan Leduc

Please Note

Some content has been directly extracted from Magdin Stoica's original M.Eng project report:
Magdin Stoica, "DESpot – Unlock the DES potential," M.eng Project Report, Dept. of Computing and Software, McMaster University, 2007.

For more information about Discrete Event Systems see:
R.J. Leduc, "Hierarchical Interface-Based Supervisory Control with Data Events," International Journal of Control, vol. 82, no. 5, pp. 783 - 800, May, 2009.

Contents

Introduction

DESpot is a GUI modelling environment for Discrete Event Systems (DESs), in particular those used in hierarchical interface-based supervisory control (HISC). The environment allows the definition of DES as well as both flat and hierarchical systems of DES. Flat systems are modelled as a set of supervisor DES, a set of plant DES and a set of template DES. HISC systems are modelled as a set of subsystems and interfaces. Subsystems are modelled as a set of supervisor DES, a set of plant DES and a set of template DES. Interfaces are modelled as a set of supervisor DES and a set of interface template DES. DESpot provides an intuitive GUI for defining DES projects and provides a native implementation of HISC algorithms. DESpot supports HISC projects defined with low-data events. DESpot uses the synchronous product to combine the project DES to construct the system's closed-loop behavior.

DESpot also provides a graphical and tabular editor for defining the DES structure (states, events and transitions) and provides tools to check the consistency of the DES and its different properties (e.g. integrity, reachability, nonblocking). DES are saved by DESpot in an XML-based language into files with a *.DES extension.

DESpot provides a project editor for defining flat DES projects. The editor provides tools to verify nonblocking and controllability properties of flat projects.

DESpot provides a project editor for defining HISC projects. Using this editor the user can create high or low-level subsystems and interfaces. A subsystem is defined as a collection of plant DES that makes up the subsystem’s plant, a collection of supervisor DES that make up the subsystem’s supervisor and a collection of template DES that make up the subsystem's template. An interface is defined as a collection of interface DES and a collection of interface template DES. As part of the project editors, DESpot provides an intuitive and powerful UI for managing the project event set and the complex ownership and type rules that govern an HISC project. The project editor allows users to verify the consistency of the project as well its different properties (e.g. level-wise interface consistent, level-wise nonblocking, level-wise controllable).

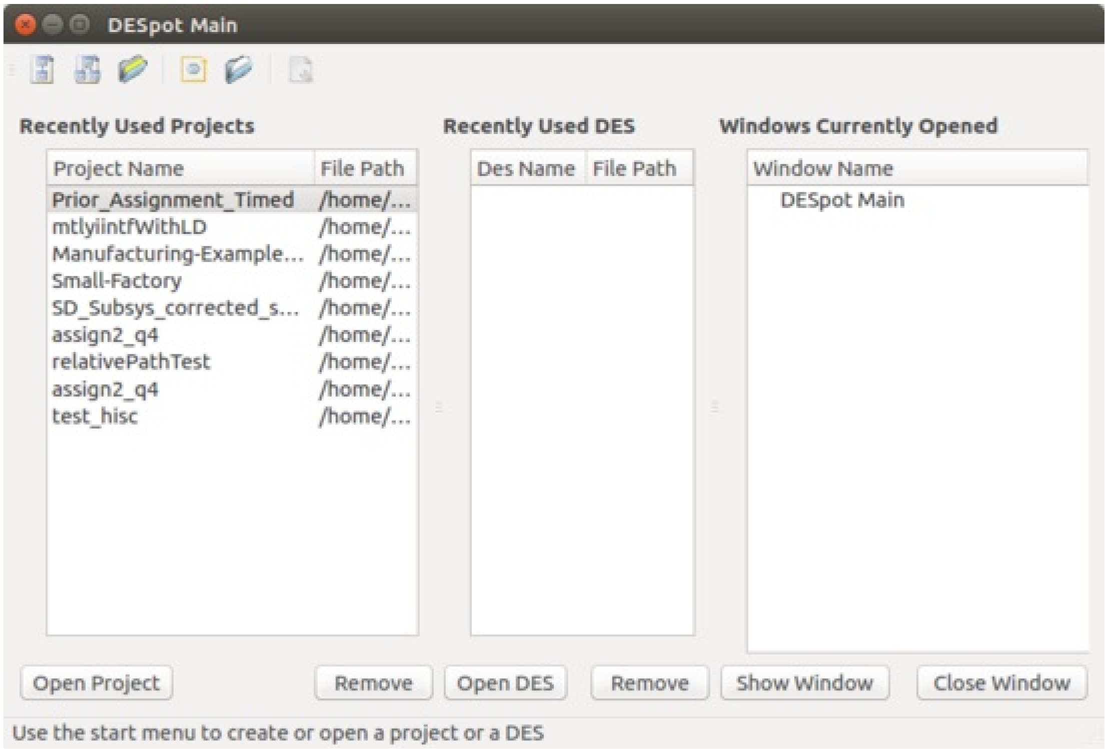
In addition, DESpot provides a DES Simulation Software (DSS) module for simulating either HISC or flat DES projects.

Projects are saved by DESpot in an XML-based language into files with a *.DESP extension. Within DESpot, all available characters for use in the naming of elements (subsystems, DES, events etc) are: **a**-**z**; **A**-**Z**; **0**-**9**; **.**_**-** . The character **%** is used for the naming of elements (states, DES name, events etc) in template DES and interface template DES. Note that "**.**" is currently reserved for future applications.

Please note that many items such as DES and subsystems, offer access to related functionality via context menus. Context menus can be accessed by right-clicking on the object.

The DESpot Main Window

When DESpot is started, the Main window is displayed.



Project Name	File Path
Prior_Assignment_Timed	/home/...
mtlyiintfWithLD	/home/...
Manufacturing-Example...	/home/...
Small-Factory	/home/...
SD_Subsys_corrected_s...	/home/...
assign2_q4	/home/...
relativePathTest	/home/...
assign2_q4	/home/...
test_hisc	/home/...

Des Name	File Path

Window Name
DESpot Main

Use the start menu to create or open a project or a DES

The DESpot main window contains three areas for browsing as well as a menu, a toolbar, and a button panel. The three browsing areas are for quickly finding recently used projects, recently used DES and for listing, in a hierarchical form, any windows that are currently open. A project or DES found in these lists can be opened for editing by double clicking on it. Items listed in the "Windows Currently Open" list can be double clicked on, which will bring that window to the top of all other windows.

The menu has four options which allow you to do a number of tasks. The **Start** menu has the option to Start a new Flat or HISC project, Open an existing project, Create a new Regression Test, or close the DESpot application. The **DES** menu has options to Create or Open a DES and a radio button which decides whether DES are opened in the Tabular or Graphical editor. The **Windows** menu gives the option to show or close the window which is highlighted in the "Windows Currently Open" area. The **Help** menu opens this help browser or an **About DESpot** dialog.

The toolbar has six buttons. From left to right, these buttons allow a user to Start a new flat or HISC project, Open a project, Create or Open a DES, and Create a new Regression Test. If one needs to work with a DES outside a project, they can do so from here.

The buttons along the bottom of the window do similar tasks as described above. The **Remove** buttons Remove remove the highlighted project or DES from the respective browsing area. This does not delete the files from disk, just from the **Recently Used** list.

Creating a Flat Project

From the DESpot Main Window menu bar or from the Project Editor menu bar, select

Start | New Project | New Flat Project or click on the **New Flat Project** button on the toolbar. A dialog box will popup asking for the *name* of the new project. Once a name has been entered, click **OK** and the Flat Project Editor window will open.

Creating an HISC Project

From the DESpot Main Window menu bar or from the Project Editor menu bar, select

Start | New Project | New HISC Project or click on the **New HISC Project** button on the toolbar. A dialog box will popup asking for the *name* of the new project. Once a name has been entered, click **OK** and the HISC Project Editor window will open.

Opening an Existing Project

From the DESpot Main Window menu bar or from the Project Editor menu bar, select

Start | Open Project or click on the **Open Project** button on the toolbar. This will open a file browser dialog to locate an existing project. If the project is in the "Recently Used Projects" list, click on the **Open Project** button Open Project to open the highlighted project or double-click on the project you wish to open from the list. This will open the selected project in the Project Editor window.

Creating a DES

From the DESpot Main Window menu bar select **DES | New DES** or click on the **New DES** button on the toolbar. This will open a dialog asking for the *name* and *type* (regular, subsystem, interface or interface template) of the new DES. The *type* property is only relevant for HISC systems but is used for all DES to allow DES to be used for multiple project types simultaneously. Essentially, des are type "regular" for a flat project, "subsystem" if the DES is to be part of an HISC subsystem, "interface" if the DES will be part of an HISC interface, and "interface template" if the DES will be template DES in an HISC interface. The type can be changed later using either the graphical or tabular des editors. Once the *name* and *type* have been entered, click **OK** and the graphical or tabular editor will open to allow DES creation.

Opening an Existing DES

From the DESpot Main Window menu bar select **DES | Open DES** or click on the **Open DES** button on the toolbar. This will open a file browser dialog to locate an existing DES. If the DES is in the "Recently Used DES" list, click on the **Open DES** button Open DES to open the highlighted DES or double click on the DES you wish to open from the list. This will open the selected DES in the graphical or tabular editor window.

Creating a Regression Test

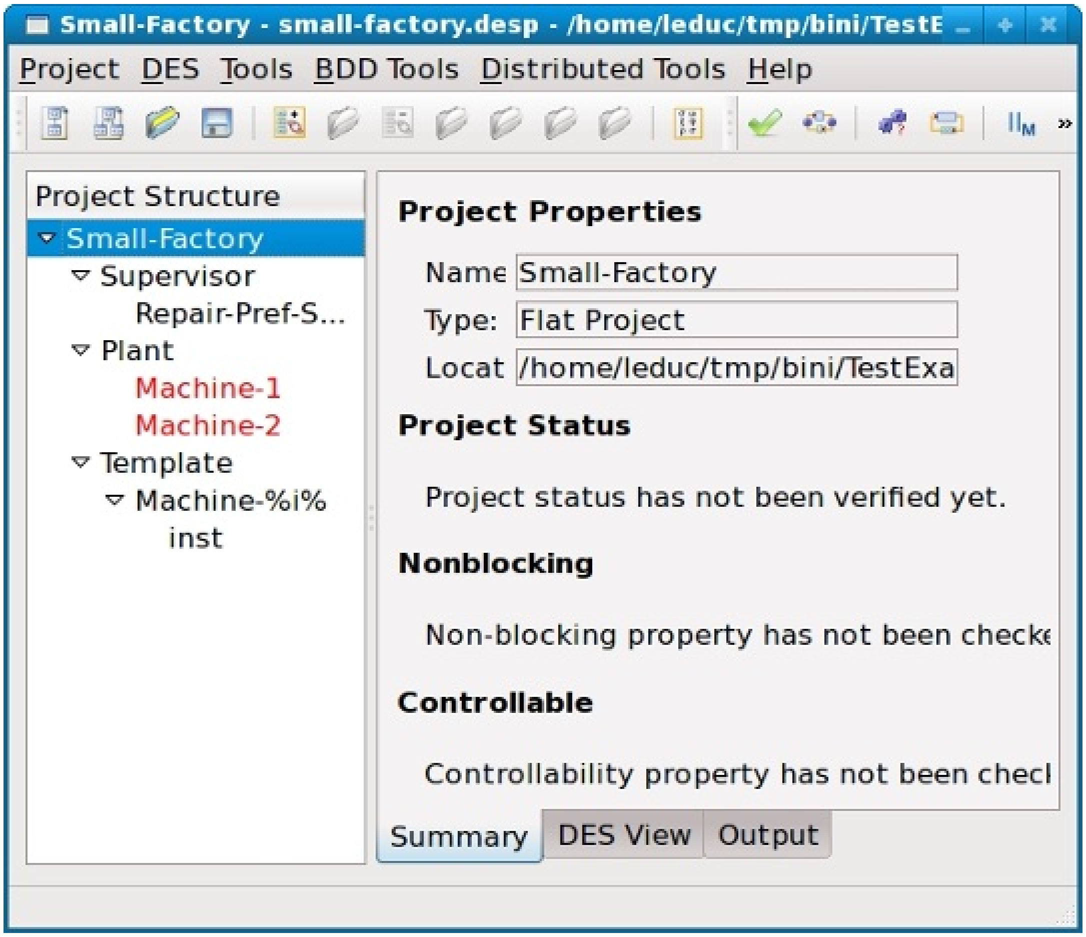
From the DESpot Main Window menu bar select **Start | New Regression Test** or click on the **New Regression Test** button on the toolbar. The Regression Test Editor window will open with a new Regression Test.

Project Editor

Flat Project Editor Window

The Flat Project Editor window (FPE) allows you to define, modify, or save flat projects. This is where you also check project properties such as the project’s validity, and whether it is nonblocking and controllable. The left pane of the project window shows the project structure: the DES that make up the supervisor, the DES that make up the plant and the DES that make up the template. The right pane shows the project details in the form of three tabs.

The Summary tab displays a summary of the project properties: the name of the project, its type and the location of the file where the project resides on disk. In addition to these properties the summary tab also shows several properties including the date and time they were checked. Note that if a property has not been checked, the pane that corresponds to it will read “'property name' has not been checked yet”.



Project Properties

Name Small-Factory

Type: Flat Project

Locat /home/leduc/tmp/bini/TestExa

Project Status

Project status has not been verified yet.

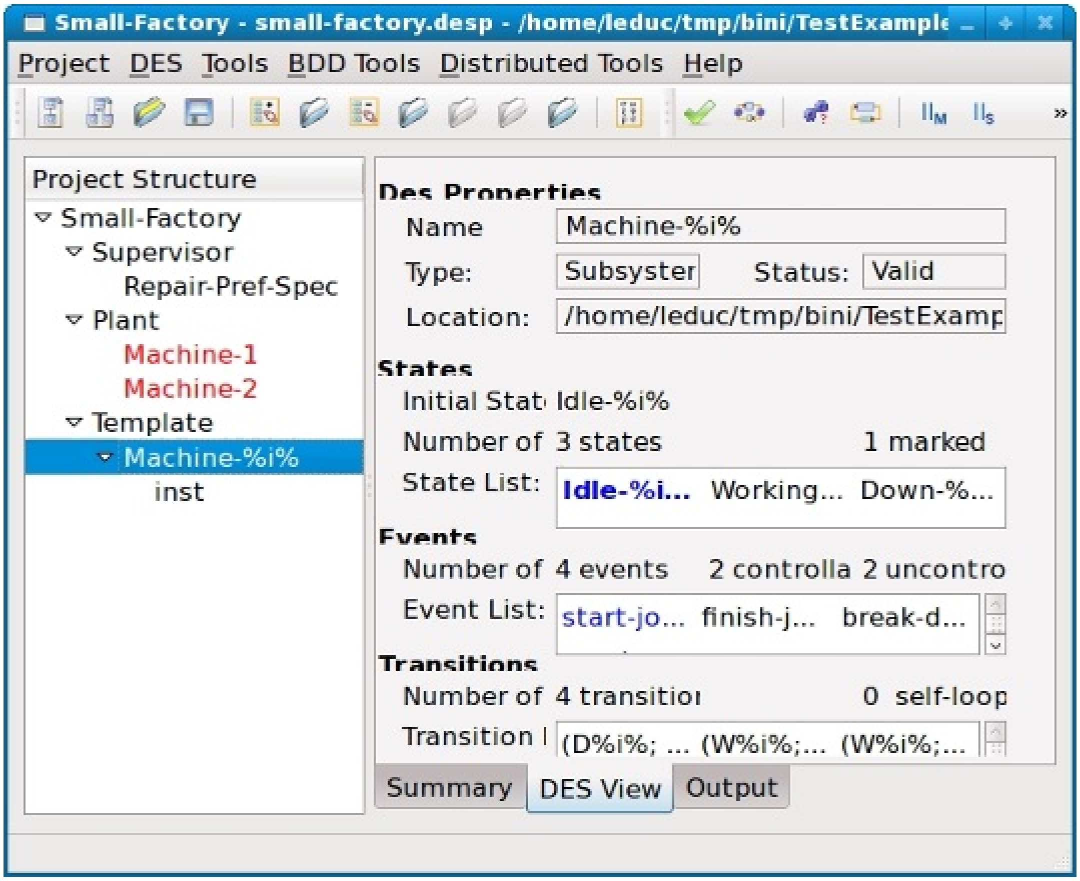
Nonblocking

Non-blocking property has not been checke

Controllable

Controllability property has not been checl

The **DES View** tab displays the contents of the currently selected DES in the project. The following figure shows the current DES is Machine-%i% and its contents are displayed in the **DES View** tab. Note that when hovering with the mouse over a state, event or transition, a tooltip will show all the necessary details. This view allows users to look at a DES without opening an editor.


Small-Factory - small-factory.desp - /home/leduc/tmp/bini/TestExample
Project DES Tools BDD Tools Distributed Tools Help
Summary
DES View
Output

Project Structure
- Small-Factory
  - Supervisor
    - Repair-Pref-Spec
  - Plant
    - Machine-1
    - Machine-2
  - Template
    - Machine-%i%
      - inst

Des Properties

Name: Machine-%i%
Type: Subsyster Status: Valid
Location: /home/leduc/tmp/bini/TestExamp

States

Initial Stat: Idle-%i%
Number of: 3 states 1 marked
State List: Idle-%i... Working... Down-%...

Events

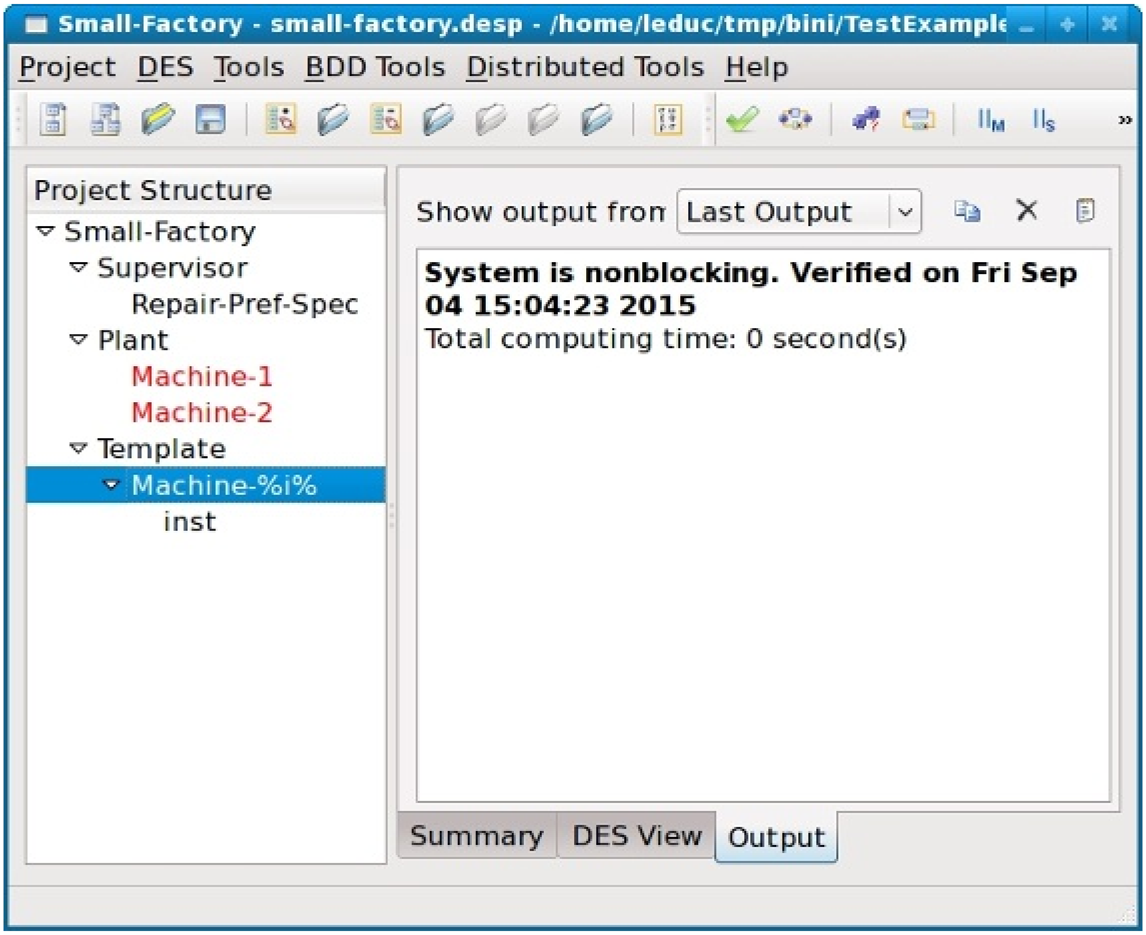
Number of: 4 events 2 controlla 2 uncontro
Event List: start-jo... finish-j... break-d...

Transitions

Number of: 4 transitio 0 self-loop
Transition l: (D%i%; ... (W%i%;... (W%i%;...

The **Output Tab** shown below displays the output of the different tools that are used in the editor. For example, it will show the output of the validity check, including any errors or warnings, the output of the controllability and nonblocking algorithms, as well as system measurements . The output from a specific tool can be selected by clicking on the Show Output From drop down box and selecting the desired output type.

Show output from: Last Output



```
System is nonblocking. Verified on Fri Sep
04 15:04:23 2015
Total computing time: 0 second(s)
```

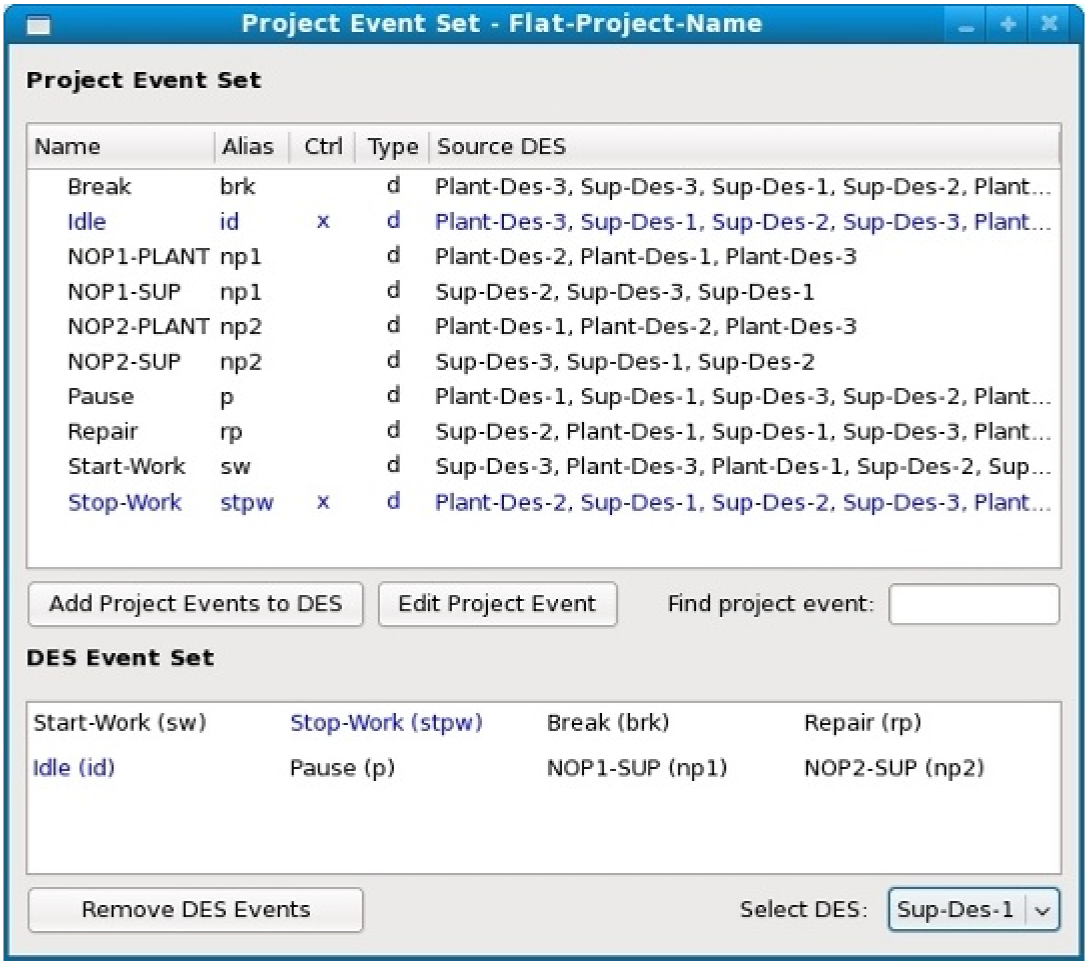
Flat Project Validity

A flat project is valid if it meets the following criteria:

1. The name of the project does not to contain invalid characters.
2. All DES must pass consistency check described in DES Tools.
3. If two different DES in the project contain an event with the same name, then the events must be identical in all their properties.
4. Every event in the event set of the project is used in at least one DES component belonging to the project.
5. The supervisor must contain at least one DES component.
6. The plant must contain at least one DES component.

FPE: Manage Events

From the **Project Menu**, select **Manage Events...** or from the toolbar, click on the Manage Events button σ ω ξ φ ρ σ. This will open the Project Event Set window.



Project Event Set

Name	Alias	Ctrl	Type	Source DES
Break	brk		d	Plant-Des-3, Sup-Des-3, Sup-Des-1, Sup-Des-2, Plant...
Idle	id	x	d	Plant-Des-3, Sup-Des-1, Sup-Des-2, Sup-Des-3, Plant...
NOP1-PLANT	np1		d	Plant-Des-2, Plant-Des-1, Plant-Des-3
NOP1-SUP	np1		d	Sup-Des-2, Sup-Des-3, Sup-Des-1
NOP2-PLANT	np2		d	Plant-Des-1, Plant-Des-2, Plant-Des-3
NOP2-SUP	np2		d	Sup-Des-3, Sup-Des-1, Sup-Des-2
Pause	p		d	Plant-Des-1, Sup-Des-1, Sup-Des-3, Sup-Des-2, Plant...
Repair	rp		d	Sup-Des-2, Plant-Des-1, Sup-Des-1, Sup-Des-3, Plant...
Start-Work	sw		d	Sup-Des-3, Plant-Des-3, Plant-Des-1, Sup-Des-2, Sup...
Stop-Work	stpw	x	d	Plant-Des-2, Sup-Des-1, Sup-Des-2, Sup-Des-3, Plant...

DES Event Set

Start-Work (sw) Stop-Work (stpw) Break (brk) Repair (rp)
Idle (id) Pause (p) NOP1-SUP (np1) NOP2-SUP (np2)

Select DES: Sup-Des-1

The event set of a project is made up of the union of the event sets of the plant DES and supervisor DES that are part of the project. Events from template DES are not considered part of the project and thus do not show up here.

Events are logically identified by their name hence two events with the same name have to be identical in all respects: alias, controllability and type. The project event manager is therefore useful for checking for any kind of inconsistencies and to be able to add an event from one DES to another in the same project. This ensures that two events that are supposed to be the same, are in fact identical. Events that have conflicts are shown in red. A tooltip helps users diagnose and resolve any conflicts.

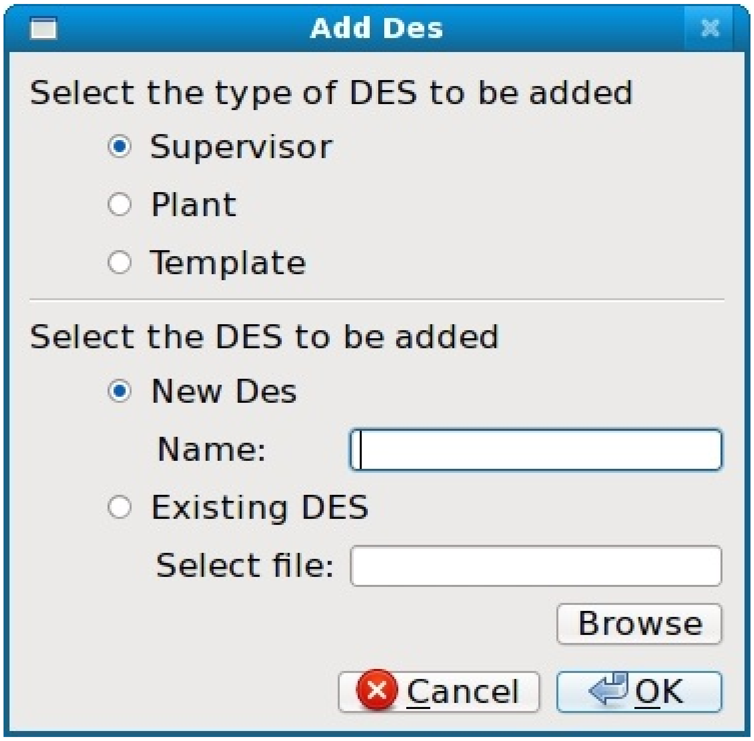
The **Add Project Events to DES** button Add Project Events to DES is used to move selected events to the DES currently displayed in the **DES Event Set** window. Hold **CTRL** when you select an event to select more than one at a time. The DES displayed in the **DES Event Set** window can be changed using the **Select DES:** drop-down box Select DES: Sup-Des-1. The **Remove DES Events** button

Remove DES Events will remove the selected events in the **DES Event Set** window from the DES currently being viewed.

The **Edit Project Event** button Edit Project Event will edit the event that is currently selected in the **Project Event Set** window. Changes made here will be reflected in every instance of the event in every DES in the project. Use this feature when you want to globally change an event. Use the **Edit Event** option in the Tabular or Graphical editor to edit an event in a specific DES only. By typing an event name in the **Find project event:** text box Find project event: □, you can quickly find the event you're looking for.

FPE: Add DES

From the FPE menu bar, select **DES | Add DES...** or from the toolbar, click on the **Add DES** button . This will open the dialog box shown below.



Use the first radio button to select whether the DES will be of type **Plant**, **Supervisor** or **Template**. In the text box, type the name of your new DES, or click Browse, to open an existing DES.

FPE Tools

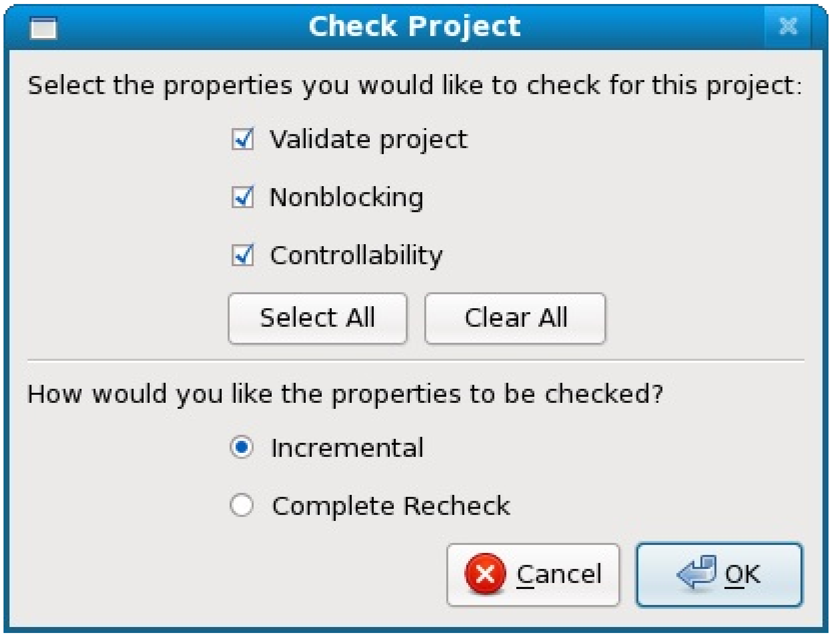
Verify Project Integrity

From the **Tools** menu, select **Verify Project Integrity** or from the toolbar, click on the Verify Project Integrity button ✓. This will run a series of checks to verify the project is Valid. If the project fails any of these checks, a message box will display the number of errors and warnings, and the Output Tab will provide a list of errors and warnings. Note that most algorithms (nonblocking, controllability, sync, etc) will refuse to run on an invalid project.

WARNING: if the project integrity check passes, but you get an integrity related error about DES when running an algorithm, run **Clean Project**, and then re-run the integrity test.

Check Project

From the **Tools** menu, select **Check Project** or from the toolbar, click on the Check Project button. This will open the following window.



Check Project

Select the properties you would like to check for this project:

☑ Validate project

☑ Nonblocking

☑ Controllability

How would you like the properties to be checked?

● Incremental

○ Complete Recheck

From here you can choose which properties to check. An **Incremental** check will check all the properties that you have selected and that have either not been checked, or have failed during the last check. A **Complete Recheck** will check all the properties you have selected regardless of current status.

Clean Project

From the **Tools** menu, select **Clean Project**. This will discard any data saved from a previous check and "clean" the project and DES of any precomputed information. This means that you will have to re-check the various properties of your project in order to have each property valid again. If a project was created by an earlier version of DESpot and is producing odd results, doing a "Clean Project" may solve the issue.

Nonblocking

From the **Tools** menu, select **Nonblocking** or from the toolbar, click on the Check Nonblocking button . This will run an automata based algorithm to check that the synchronous product of your project is nonblocking.

Controllabiltiy

From the **Tools** menu, select **Controllabiltiy** or from the toolbar, click on the Check Controllability button . This will run an automata-based algorithm to verify that the synchronous product of your project supervisor DES is controllable for the synchronous product of your plant DES. Note that the algorithm assumes that your closed-loop system is constructed using the synchronous product.

Project Meet

From the **Tools** menu, select **Project Meet** or from the toolbar, click on the Meet button ||M. **Please Note:** This will create a DES from the synchronous product of all project DES. In order to perform a meet operation, the event sets of all the DES in the project must be identical.

Project Sync

From the **Tools** menu, select **Project Sync** or from the toolbar, click on the Sync button ||S. This will create a DES from the synchronous product of all project DES.

Project Synthesis

From the **Tools** menu, select **Project Synthesis**. This will create a supremal, controllable, nonblocking supervisor for the project.
Please note: This feature is not yet implemented.

System Simulation

From the **Tools** menu, select **System Simulation** or from the toolbar, click on the System Simulation button . This will open the DES System Simulation window. Please note that the project must be Valid before the simulator can be used.

TDES Controllability

From the **Tools** menu, select **TDES Controllability**. This will run an automata-based algorithm to verify that the synchronous product of your project supervisor DES is TDES controllable for the synchronous product of your plant DES. Note that the algorithm assumes that your closed-loop system is constructed using the synchronous product. It is required that each DES in the project must contain a controllable event called tick, with alias t.

DESpot does not currently allow one to directly specify which events are forcible in a timed DES. Instead it uses the sampled-data supervisory control assumption that the set of forcible events is equal to the set of prohibitable (controllable non-tick) events.

Export to BDDsd

To export a project to BDDsd, open the **Project** menu and select **Export to BDDsd**. DESpot will ask you for the filename and filepath you wish to save your BDDsd project file. Click **Save** to finish the operation. Your project file and all required DES will now be converted into the BDDsd format. For a project to be eligible for export, every DES must contain a controllable event called **tick**, with alias **t**.

Fault-Tolerant Supervisory Control

DESpot includes tools to work with timed and untimed flat DES projects in order to verify properties for fault-tolerant supervisory control. In particular it include tools to verify fault-tolerant timed and untimed controllability and nonblocking properties, for several fault scenarios.

This help section explains how to specify the necessary parameters for these tools, as well as how to run these tools.

For more information on fault-tolerant supervisory control, please refer to:

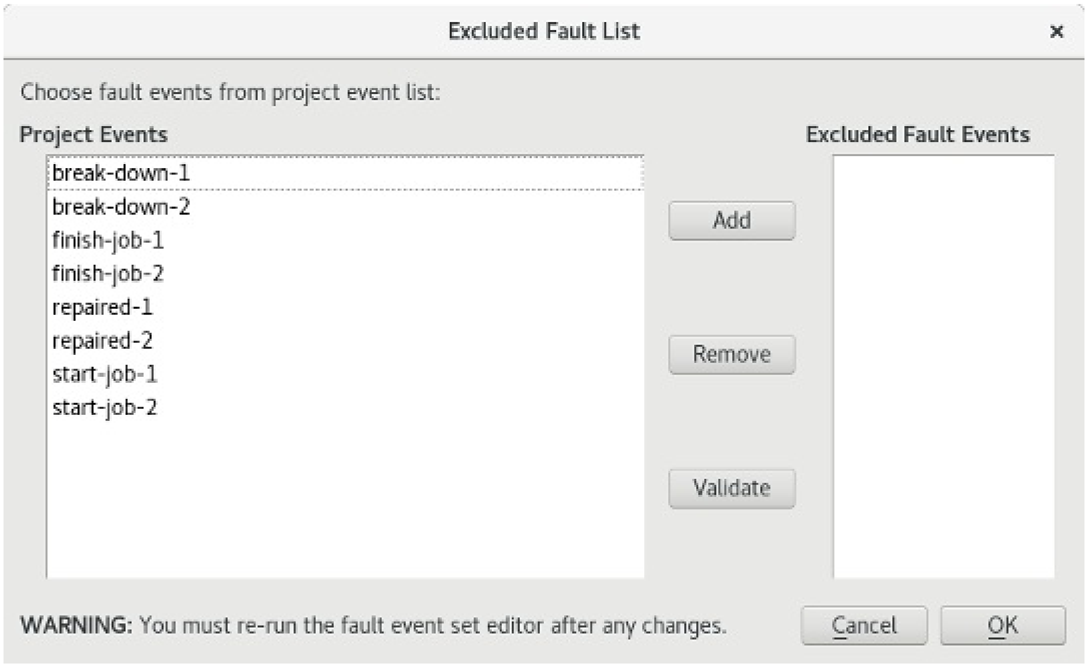
Aos Mulahuwaish, "Fault-Tolerant Supervisory Control," Phd Thesis, Dept. of Computing and Software, McMaster University, May. 2019.

Amal Alsuwaidan, "Timed Fault Tolerant Supervisory Control," M.A.Sc. Thesis, Dept. of Computing and Software, McMaster University, April 2016.

Specify Excluded Fault Set
Specify Standard Fault and Reset Sets
Running Fault-Tolerant Algorithms

Specifying Excluded Fault Events

In order to select the excluded fault events, $\Sigma_{\Delta F}$, press the $\Sigma_{\Delta F}$ button (it's labelled "Choose Excluded Fault Events), or click on **Select Excluded Faults** in the **Project** menu. This will bring up the following dialog:



Excluded Fault List

Choose fault events from project event list:

Project Events

break-down-1
break-down-2
finish-job-1
finish-job-2
repaired-1
repaired-2
start-job-1
start-job-2

Excluded Fault Events

WARNING: You must re-run the fault event set editor after any changes.

To add events to the excluded fault events set, select them in the left column, then click the **Add** button in the center of the dialog. To remove events from the excluded fault events set, select the events to remove from the right column, then click the **Remove** button. Once you are done selecting the excluded fault events, click **Ok**.

Clicking either **Validate** or **Ok** will run a check to make sure that the events selected for $\Sigma_{\Delta F}$ are valid according the (timed) FT concistent definition. For instance, any controllable events in the set will make this check fail. Fault events are also not allowed to belong to supervisors.

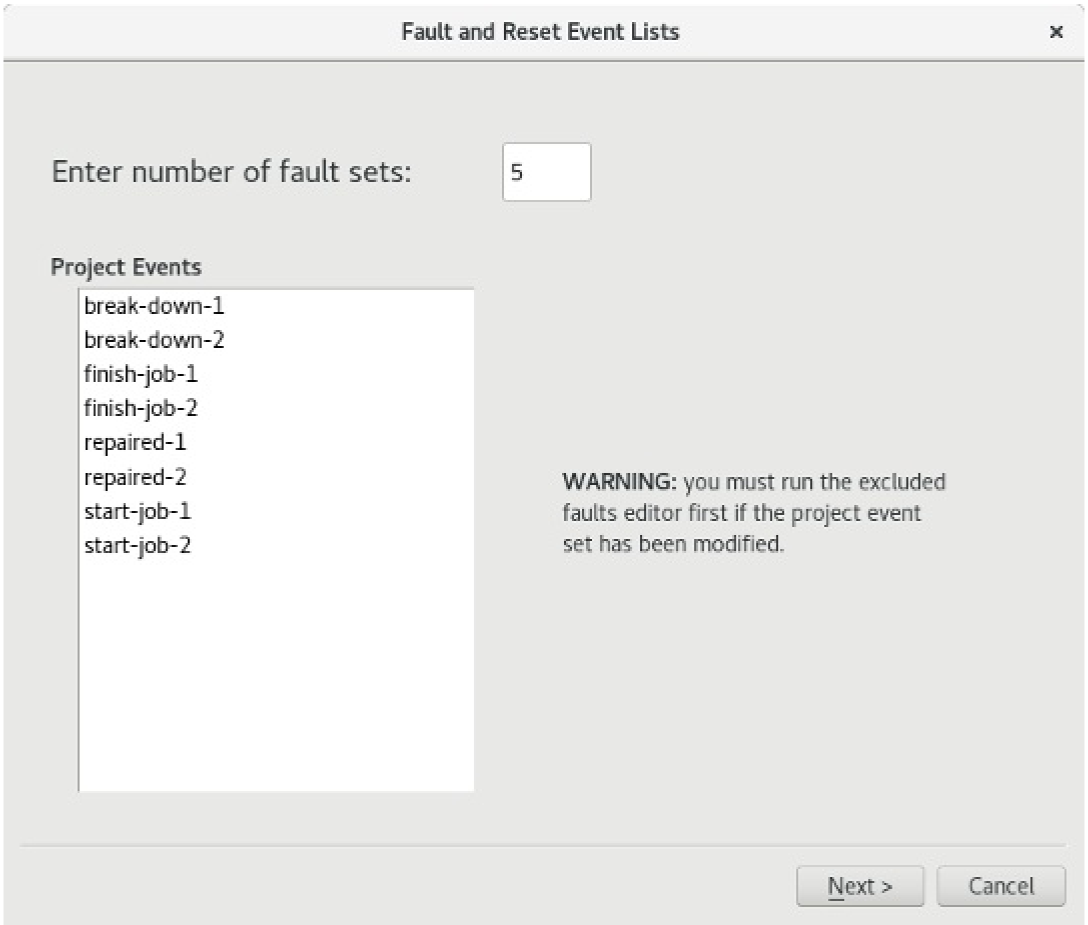
Warning: if you have made any changes to the project since this dialogue was run and the excluded fault set is non-empty, you must re-run this dialogue to ensure that the set still satisfies the (timed) FT concistent definition before running any fault-tolerant algorithms.

Warning: if the project contains excluded faults and standard fault and/or reset events, the excluded fault dialog should always be run before the standard fault and reset set dialog to ensure proper operation.

Specifying Standard Fault and Reset Events

Before specifying the *standard fault and reset sets,* make sure that you have specified the excluded fault events first. This dialog will give an error unless the excluded fault events dialog has already been run.

In order to select the *standard fault* (Σ_{Fi}) and *reset* (Σ_{Ti}) events, press the Σ_{F_i} button (labelled Choose fault event sets), or click on **Select Fault Sets** in the **Project** menu. This will bring up the wizard used to build the fault sets. The first screen will look like this:



Fault and Reset Event Lists

Enter number of fault sets: 5

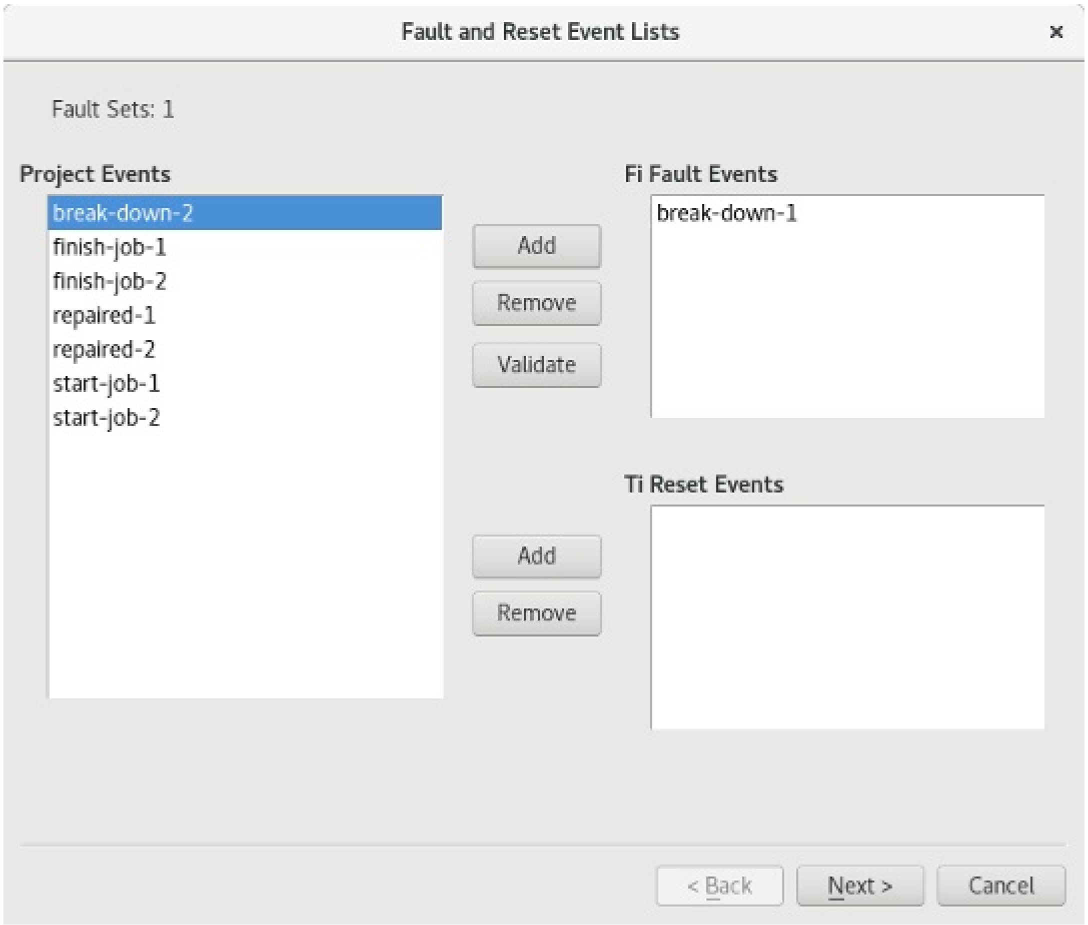
Project Events

- break-down-1
- break-down-2
- finish-job-1
- finish-job-2
- repaired-1
- repaired-2
- start-job-1
- start-job-2

WARNING: you must run the excluded faults editor first if the project event set has been modified.

Enter the number of standard fault sets to be entered, then press **Next**. If you set the number of fault sets to zero, all fault and reset sets will be removed from the project. If you decrease the number of fault sets from the current set amount (i.e. from 4 down to 2 sets), then the existing unused fault and reset sets will be removed (i.e. sets 3 and 4 will be removed) from the project.

You will now be presented with the following screen in order to build the first standard fault and reset set. If the project has exisiting fault/reset sets defined, they will populate the screen allowing for easy editing. Please note that as each fault and reset set must be pairwise-disjoint from each other, events already added to other sets (including the excluded fault set) will not show up in the dialog window.



Fault and Reset Event Lists

Fault Sets: 1

Project Events

break-down-2
finish-job-1
finish-job-2
repaired-1
repaired-2
start-job-1
start-job-2

Fi Fault Events

break-down-1

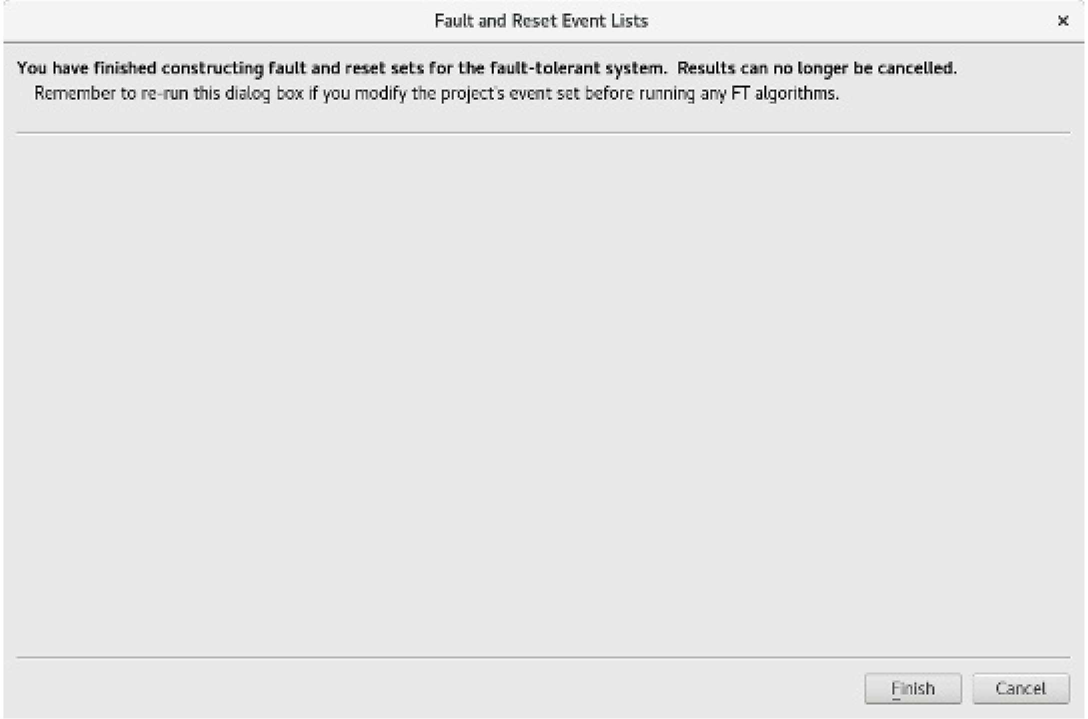
Ti Reset Events

Click on events in the left column, then click on the corresponding **Add** button to add them to either the *standard fault* event set or the *reset* event set. To remove events from either set, click on the event in the corresponding right column, and select remove.

When the **Next, Validate,** or **Commit** button is pressed, a check will be performed to make sure that the selection of fault events is valid according the (timed) FT concistent definition. For instance, any controllable events in the standard fault set will make this check fail. Fault events are also not allowed to belong to supervisors. Reset event sets may be left empty, but fault sets must contain at least one event.

If they are valid, then you will be brought to the next screen to build the next sets of standard fault and reset events.

Once all the sets of standard fault and reset event sets have been entered, the following window is displayed:



You have finished constructing fault and reset sets for the fault-tolerant system. Results can no longer be cancelled.
Remember to re-run this dialog box if you modify the project's event set before running any FT algorithms.

When **Finish** is pressed, the standard fault and reset event wizard will exit. The project is now set up to run fault-tolerant algorithms.

Warning: Once the final screen is reached, the project has been updated and it is too late to cancel the changes.

Warning: if you have made any changes to the project since this dialogue was run and the standard fault/reset sets are non-empty, you must re-run this dialogue to ensure that the sets still satisfy the (timed) FT concistent definition before running any fault-tolerant algorithms.

Running Fault-Tolerant Algorithms

Before running *fault-tolerant* algorithms, make sure that the excluded fault set and the standard fault and reset event sets have been set up. If at any point project events have been modified, or added or removed from the project, then go through these dialogs again to ensure that the sets still satisfy the (timed) FT concistent definition.

There are several different variations of the fault-tolerant controllability and nonblocking algorithms available. There are timed and untimed controllability algorithms as well as nonblocking algorithms that can be used on both timed and untimed systems.

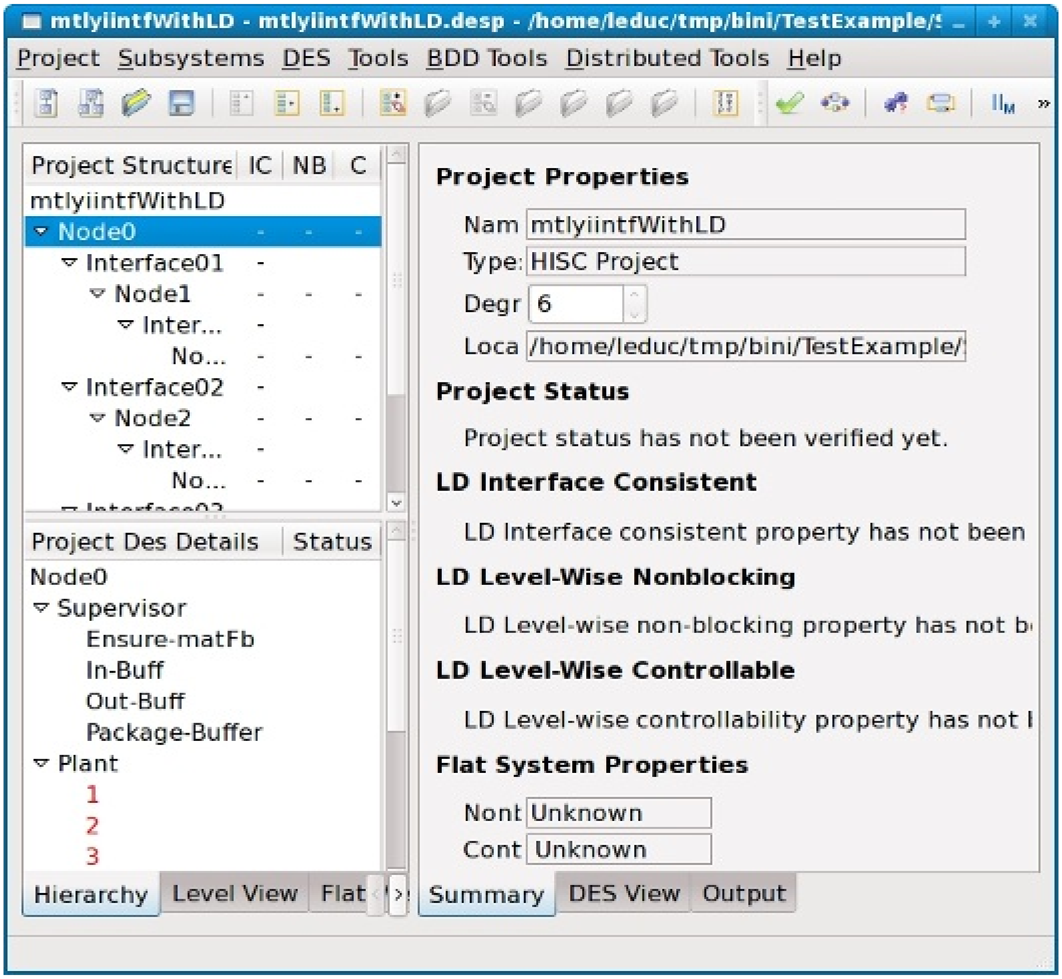
The standard algorithms can be accessed under the **Tools** menu. The BDD versions of the algorithms can be accessed from the **BDD Tools** menu.

Currently, the following fault-tolerant scenarios are supported: default, N-fault scenario, non-repeatable N-fault scenario, and the resettable fault scenario.

Warning: When the fault-tolerant algorithms execute, a temporary version of the project is constructed with added plant components required to evaluate the desired fault-tolerant scenario. Instead of checking if the original project is valid, instead the temporary project is checked.

HISC Project Editor Window

The HISC project editor is based on the flat one but it is more complex. A HISC project is organized into subsystems and interfaces. Each subsystem has a set of supervisor DES and a set of plant DES. Interfaces have a set of supervisor interface DES. Each low-level subsystem communicates with the high-level subsystem through a dedicated interface. HPE uses three alternative views to display this structure: Hierarchical View, Level View and Flat View.



Project Structure	IC	NB	C
mtlyiintfWithLD			
▿ Node0	-	-	-
▿ Interface01	-		
▿ Node1	-	-	-
▿ Inter...	-		
No...	-	-	-
▿ Interface02	-		
▿ Node2	-	-	-
▿ Inter...	-		
No...	-	-	-

Node0
▿ Supervisor
Ensure-matFb
In-Buff
Out-Buff
Package-Buffer
▿ Plant
1
2
3

Project Properties

Nam: mtlyiintfWithLD
Type: HISC Project
Degr: 6
Loca: /home/leduc/tmp/bini/TestExample/

Project Status

Project status has not been verified yet.

LD Interface Consistent

LD Interface consistent property has not been

LD Level-Wise Nonblocking

LD Level-wise non-blocking property has not b

LD Level-Wise Controllable

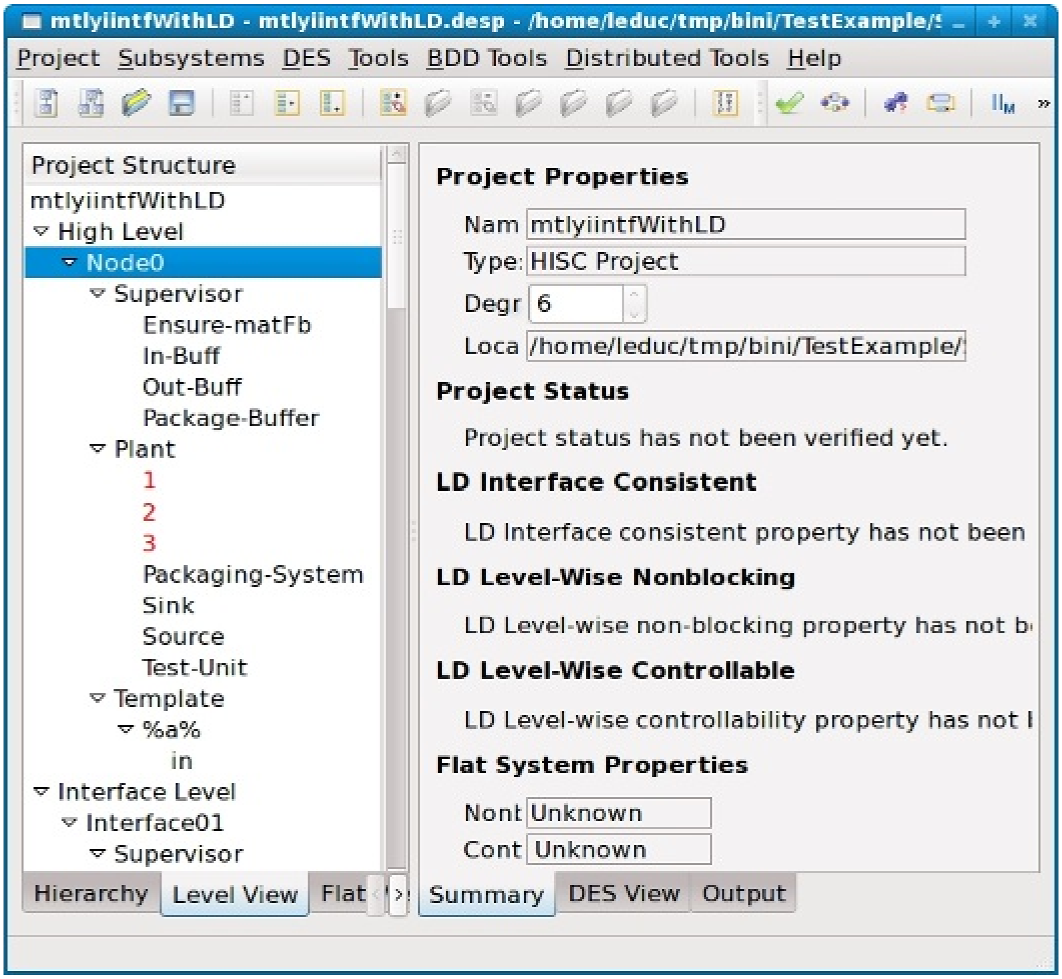
LD Level-wise controllability property has not

Flat System Properties

Nont: Unknown
Cont: Unknown

The **Hierarchical View** (shown above) is split into two panes: the top pane shows the project architecture – the subsystems and interface they use to communicate with each other; the bottom pane shows the details of the subsystem or interface currently selected in the top pane. The top pane shows the High-Level Subsystem as being selected hence the bottom pane shows the DES of the high-level subsystem split into two distinctive sets: the supervisor DES set and plant DES set. The **Summary Tab** is very similar to the one for flat projects but it shows properties specific to HISC projects. Please note that the **Flat System Properties** at the bottom of the summary tab, that is the nonblocking and controllable properties, are only set by the flat nonblocking check and the flat controllability check. The HISC property checks have no effect on them.

The **Level View** (shown below) displays the project structure by level. This view shows three levels: high-level, interface level and low-level. Under high-level, you can find the root node subsystem, under interface level, all the interfaces are displayed, and under low-level you can find all the subsystems that are below root node. Under each subsystem/interface you have access to the set of DES that is part of it. Note how the **DES View** Tab shown is similar to the one used in the Flat Project Editor. The **Output Tab** (not shown in figure) is also similar to the Output Tab used in the Flat Project Editor but it has additional options to display more data related to HISC projets.



Project Properties

Nam: mtlyiintfWithLD

Type: HISC Project

Degr: 6

Loca: /home/leduc/tmp/bini/TestExample/!

Project Status

Project status has not been verified yet.

LD Interface Consistent

LD Interface consistent property has not been

LD Level-Wise Nonblocking

LD Level-wise non-blocking property has not b

LD Level-Wise Controllable

LD Level-wise controllability property has not I

Flat System Properties

Nont: Unknown

Cont: Unknown

The **Flat View** displays the HISC project as if it was a flat project. This view is exactly the same view as the one used in the Flat Project Editor. All supervisor DES from both high-level and low-level subsystems together with all interface DES make up the supervisor, and thus they are located in the supervisor section.

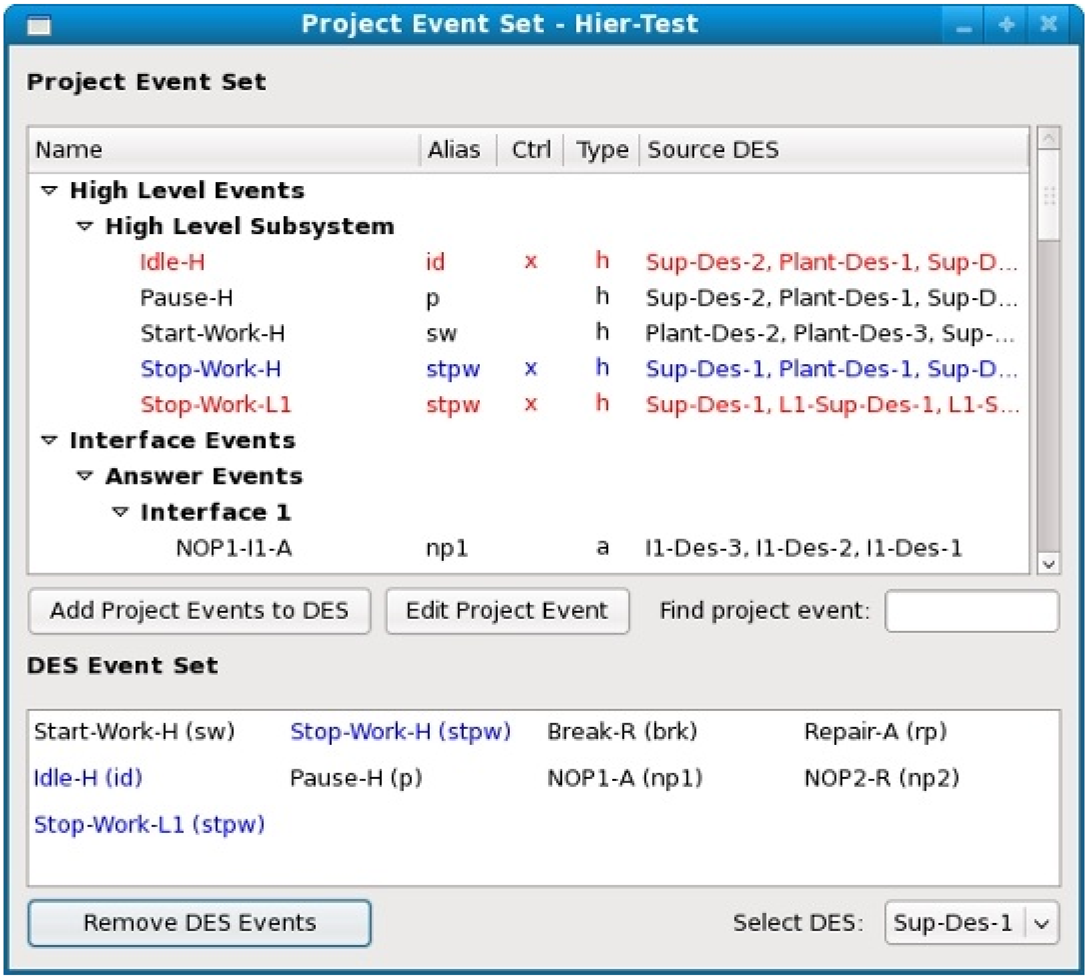
HISC Project Validity

A HISC project is valid if it meets the following criteria:

1. The name of the project does not contain invalid characters.
2. If two different DES in the project contain an event with the same name, then the events must be identical in all their properties (short name, controllability, type etc).
3. Every event in the event set of the project is used in at least one DES component belonging to the project.
4. The high-level subsystem must contain a valid name.
5. The high-level subsystem must contain at least one supervisor subsystem DES component and one plant subsystem DES component.
6. For every DES component (“the high-level component”) in the high-level subsystem, the following must be true:
   Every event in the event set of 'the high-level component” is of type default, answer, request or low data. All default events in the event set of the high-level subsystem are used exclusively in high-level subsystem components.
7. The project must contain at least one interface.
8. All interface names must be valid.
9. All interfaces must contain at least one interface DES each.
10. For every interface (“the interface”) and every component (“the DES component”) of “the interface” the following must be true:
    Every event in the event set of “the DES component” is typed as request, answer or low data and it is used exclusively by DES components of “the interface,” or in DES components of the subsystem above or DES components of the subsystem below “the interface”.
11. For each interface, there is a low-level subsystem.
12. All low-level subsystem names are valid.
13. All low-level subsystems contain at least one supervisor subsystem DES component and one plant subsystem DES component.
14. For every low-level subsystem, and for every component (“the DES component”) in the low-level subsystem the following must be true:
    Every event in the event set of “the DES component” is typed default, request, answer or low data and that all events of type default in the event set of “the DES component” are used exclusively by components in the low-level subsystem.
15. All Des that are part of the project must pass the consistency check described in DES Tools.

HPE: Manage Events

From the **Project Menu**, select **Manage Events...** or from the toolbar, click on the Manage Events button . This will open the Project Event Set window.

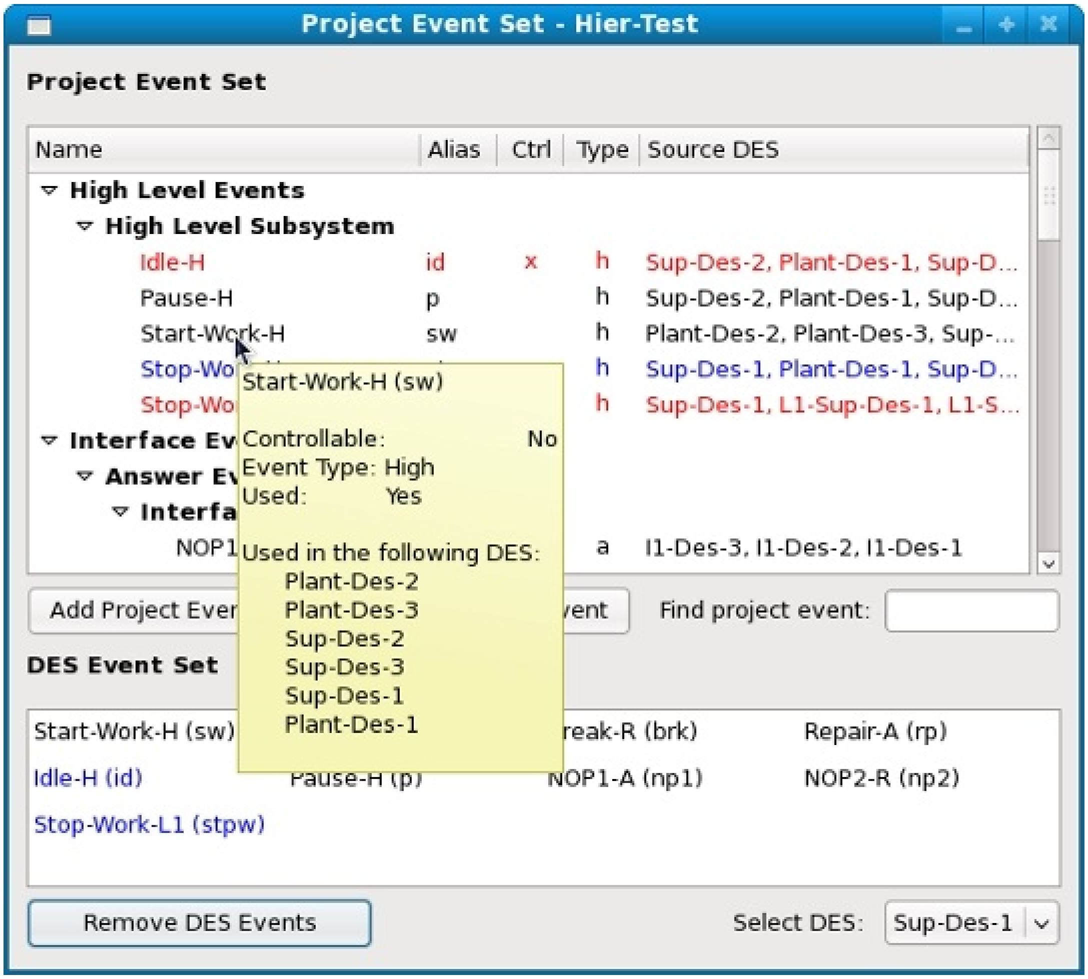


Name	Alias	Ctrl	Type	Source DES
▿ High Level Events				
▿ High Level Subsystem				
Idle-H	id	x	h	Sup-Des-2, Plant-Des-1, Sup-D...
Pause-H	p		h	Sup-Des-2, Plant-Des-1, Sup-D...
Start-Work-H	sw		h	Plant-Des-2, Plant-Des-3, Sup-...
Stop-Work-H	stpw	x	h	Sup-Des-1, Plant-Des-1, Sup-D...
Stop-Work-L1	stpw	x	h	Sup-Des-1, L1-Sup-Des-1, L1-S...
▿ Interface Events				
▿ Answer Events				
▿ Interface 1				
NOP1-I1-A	np1		a	I1-Des-3, I1-Des-2, I1-Des-1

Start-Work-H (sw)	Stop-Work-H (stpw)	Break-R (brk)	Repair-A (rp)
Idle-H (id)	Pause-H (p)	NOP1-A (np1)	NOP2-R (np2)
Stop-Work-L1 (stpw)			

The Event Manager in the HPE is based on the FPE version. However the rules governing the project event set and its construction are more complex. Subsystem can only contain default events and interface events (Request, Answer, and Low data events) from eligible interfaces. Interfaces can only contain request, answer or low-data events. A high-level subsystem is allowed to contain interface events from any interface but a low-level subsystem is only allowed to contain interface events from the interface it implements. The default events each subsystem contains must be pair-wise disjoint from those of other subsystems. Every interface event must belong to exactly one interface.

When any of these rules are violated, the tooltip that is displayed when hovering over a project event will not only display the event properties but also any errors associated with the event. An event that has conflicts is displayed in red. An example of such tooltip is shown below.



Project Event Set - Hier-Test

Project Event Set

Name	Alias	Ctrl	Type	Source DES
High Level Events				
High Level Subsystem				
Idle-H	id	x	h	Sup-Des-2, Plant-Des-1, Sup-D...
Pause-H	p		h	Sup-Des-2, Plant-Des-1, Sup-D...
Start-Work-H	sw		h	Plant-Des-2, Plant-Des-3, Sup-...
Stop-Wo			h	Sup-Des-1, Plant-Des-1, Sup-D...
Stop-Wo			h	Sup-Des-1, L1-Sup-Des-1, L1-S...
Interface Ev				
Answer Ev				
Interfa				
NOP1			a	I1-Des-3, I1-Des-2, I1-Des-1

Start-Work-H (sw)

Controllable: No
Event Type: High
Used: Yes

Used in the following DES:
Plant-Des-2
Plant-Des-3
Sup-Des-2
Sup-Des-3
Sup-Des-1
Plant-Des-1

Find project event:

DES Event Set

Start-Work-H (sw) reak-R (brk) Repair-A (rp)
Idle-H (id) Pause-H (p) NOP1-A (np1) NOP2-R (np2)
Stop-Work-L1 (stpw)

Select DES: Sup-Des-1

HPE: Subsystems

1. Add a Subsystem

2. Rename a Subsystem

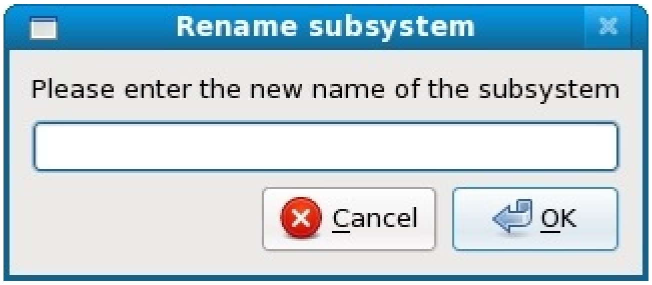
3. Delete a Subsystem

Add a Subsystem

From an HPE window, select the **Subsystems** menu, hover over the **Add** submenu, then click on the type of subsystem you wish to add. The toolbar also has buttons to add an **Interface** , or **Low-level** subsystem. This will open a dialog box prompting you for the new subsystem's properties. If you are adding a Low-level subsystem, you must specify the interface to which it will belong from the drop box. Enter the name of your new subsystem and click **OK**.

Rename a Subsystem

From an HPE window, select the Subsystem you wish to rename by highlighting it in the "Project Structure" list. Then click on the **Subsystems** menu and select **Rename**. This will open a dialog box similar to the one below.

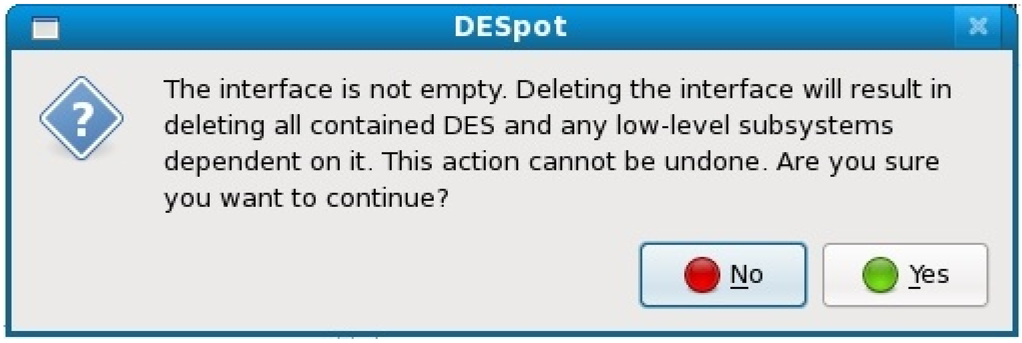

Rename subsystem

Please enter the new name of the subsystem

Enter the new name for the Subsystem and click **OK**.

Delete a Subsystem

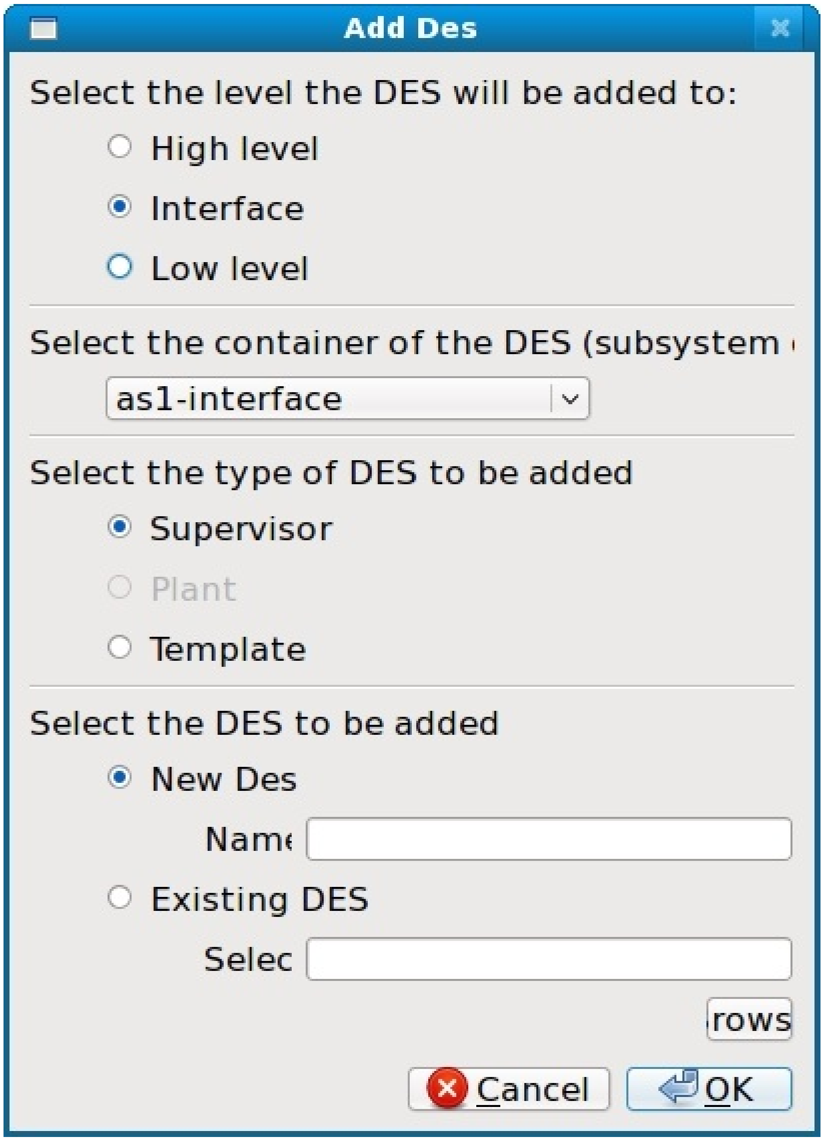
From an HPE window, select the Subsystem you wish to delete by highlighting it in the "Project Structure" list. Then click on the **Subsystems** menu and select **Delete**. Please note that when you delete a subsystem, everything under that subsystem will also be deleted. This means that if you are deleting a Low-level subsystem, all DES in the subsystem will also be deleted. If you are deleting an interface, the Low-level subsystem attached to it and all DES in the Low-level will also be deleted. A message box will popup warning you about this action.



The interface is not empty. Deleting the interface will result in deleting all contained DES and any low-level subsystems dependent on it. This action cannot be undone. Are you sure you want to continue?

HPE: Add DES

From the HPE menu bar, select **DES | Add DES...** or from the toolbar, click on the **Add DES** button . This will open the dialog box shown below.



Select the level the DES will be added to:

○ High level

● Interface

○ Low level

Select the container of the DES (subsystem

as1-interface

Select the type of DES to be added

● Supervisor

○ Plant

○ Template

Select the DES to be added

● New Des

Name

○ Existing DES

Selec

Use the first radio button to select whether the DES will be added to the **High level**, **Interface** or **Low level**. Use the drop down box to select whether the DES will be part of a **Subsystem** or **Interface**. Use the second radio button to select whether the DES will be of type **Plant** or **Supervisor**. In the text box, type the name of your new DES, or click Browse, to open an existing DES.

HPE Tools

Project Tools

Please Note: Many of the project tools for an HISC system are very similar to those for a Flat system and will treat an HISC project as a flat project (for eg. nonblocking). The

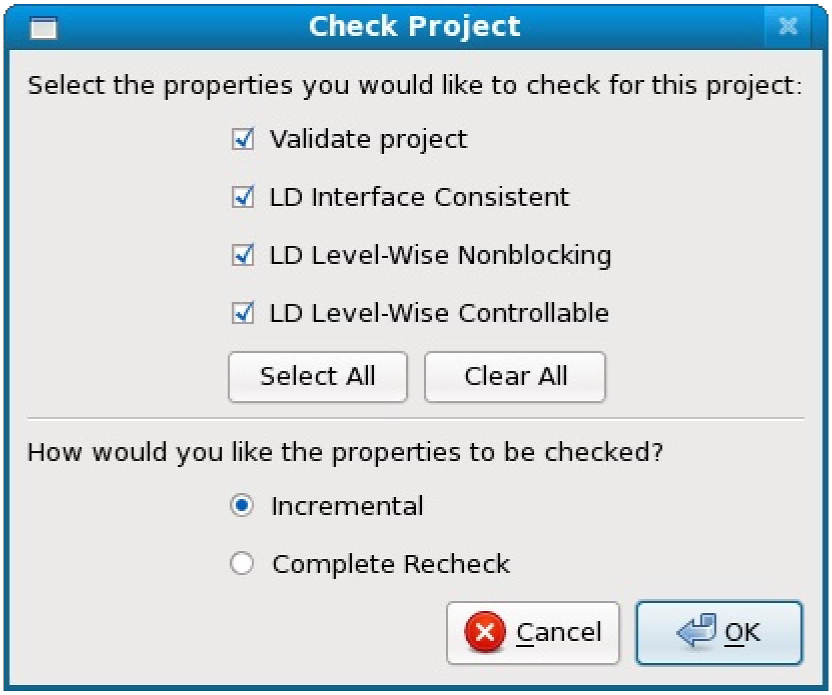
HISC Specific Tools are unique to HISC projects and treat the project as a hierarchical system.

Verify Project Integrity

From the **Tools** menu, select **Verify Project Integrity** or from the toolbar, click on the Verify Project Integrity button ✓. This will run a series of checks to verify that the project is Valid. If the project fails any of these checks, a message box will display the number of errors and warnings, and the Output Tab will provide a list of errors and warnings. Note that most algorithms (nonblocking, controllability, sync, etc) will refuse to run on an invalid project.

Check Project

From the **Tools** menu, select **Check Project** or from the toolbar, click on the Check Project button. This will open the following window.



Check Project

Select the properties you would like to check for this project:

☑ Validate project

☑ LD Interface Consistent

☑ LD Level-Wise Nonblocking

☑ LD Level-Wise Controllable

How would you like the properties to be checked?

● Incremental

○ Complete Recheck

From here you can choose which properties to check. An **Incremental** check will check all the properties that you have selected and that have either not been checked, or have failed during the last check. A **Complete Recheck** will check all the properties you have selected regardless of current status.

For HISC check project, each LD algorithm is run concurrently in a separate thread. If you have a three CPU or higher system, you should see around a three-fold speedup over running these algorithms sequentially. However, you will also see around a three-fold increase in memory as the threads don't share data structures.

Clean Project

From the **Tools** menu, select **Clean Project**. This will discard any data saved from a previous check and "clean" the project and DES of any precomputed information. This means that you will have to re-check the various properties of your project in order to have each property valid again. If a project was created by an earlier version of DESpot and is producing odd results, doing a "Clean Project" may solve the issue.

Nonblocking

From the **Tools** menu, select **Nonblocking** or from the toolbar, click on the Check Nonblocking button . This will run an automata based algorithm to check that the synchronous product of your project is nonblocking.

Controllabiltiy

From the **Tools** menu, select **Controllabiltiy** or from the toolbar, click on the Check Controllability button . This will run an automata based algorithm to verify that the synchronous product of your project supervisor DES is controllable for the synchronous product of your plant DES. Note that the algorithm assumes that your closed-loop system is constructed using the synchronous product.

Project Meet

From the **Tools** menu, select **Project Meet** or from the toolbar, click on the Meet button ||M. **Please Note:** This will create a DES from the synchronous product of all project DES. In order to perform a meet operation, the event sets of all the DES in the project must be identical.

Project Sync

From the **Tools** menu, select **Project Sync** or from the toolbar, click on the Sync button ||S. This will create a DES from the synchronous product of all project DES.

Project Synthesis

From the **Tools** menu, select **Project Synthesis**. This will create a supremal, controllable, nonblocking supervisor for the project. It will treat your project like a flat project to do this.
Please note: This feature is not yet implemented.

System Simulation

From the **Tools** menu, select **System Simulation** or from the toolbar, click on the System Simulation button . This will open the DES System Simulation window. Please note that the project must be Valid before the simulator can be used.

HISC-Specific Tools

DESpot now also supports multi-level HISC systems as described in HILL's paper referenced below. Multi-level systems are represented as a tree structure.

As per the definitions in Hill's paper, all the nodes other than the root node and leaf nodes (Nodes which do not have nodes below them) are treated as both high and low-levels for the purpose of checking multi-level HISC properties.

For the interface consistent property, DESpot checks the multi-level interface consistent property from Hill. It does not check Hill's multi-level weak interface consistent property.

R.C. Hill, J.E.R. Cury, M.H. de Queiroz,D.M. Tilbury, and S. Lafortune, "Multi-level hierarchical interface-based supervisory control," *Automatica,* vol 46, no 7, pp 1152-1164, Jul. 2010.

Check LD Interface

From the **Tools** menu, select **Check LD Interface**. This will verify that the selected interface is an LD interface.

Check Subsystem LD I-Consist

From the **Tools** menu, select **Check Subsystem LD I-Consist**. This will verify that the selected subsystem is LD interface consistent.

Check Subsystem LD Nonblocking

From the **Tools** menu, select **Check Subsystem Nonblocking**. This will verify that the selected subsystem is LD level-wise nonblocking.

Check Subsystem LD Controllable

From the **Tools** menu, select **Check Subsystem LD Controllable**. This will verify that the selected subsystem is LD level-wise controllable.

LD Interface Consistent

From the **Tools** menu, select **LD Interface Consistent** or from the toolbar, click on the LD Interface Consistent button . This will verify that the project is LD interface consistent.

LD Level-Wise Nonblocking

From the **Tools** menu, select **LD Level-Wise Nonblocking** or from the toolbar, click on the LD Level-Wise Nonblocking button . This will verify that the project is LD level-wise nonblocking.

LD Level-Wise Controllability

From the **Tools** menu, select **LD Level-Wise Controllability** or from the toolbar, click on the LD Level-wise Controllability button . This will verify that the project is LD level-wise controllable.

Level-Wise Synthesis

From the **Tools** menu, select **Level-Wise Synthesis**. This will use the HISC structure to construct supremal, controllable, nonblocking level-wise supervisors for the project. **Please note: This feature is not yet implemented.**

Export to BDDhisc

To export a project to BDDhisc, open the **Project** menu and select **Export to BDDhisc**. DESpot will ask you for the filename and filepath you wish to save your BDDhisc project file. Click **Save** to finish the operation. Your project file and all required subsystem and DES files will now be created in the BDDhisc format. For a HISC project to be eligible for export, every uncontrollable event must belong to at least one plant DES.

When exporting multiple projects, it is highly recommended to save each project in a separate directory. This will prevent files from being overwritten by different projects. If you must export multiple projects to the same directory, your projects must have different names for all the high-level and low-level subsystems. Exporting creates files based on the names of these subsystems and will overwrite previously saved files if you export multiple projects with subsystems that have the same name.

BDD Tools

BDD Tools perform most of the same checks as found in Tools, however the algorithms used benefit from the use of binary decision diagrams (BDD), where as the regular Tools use automata-based algorithms (i.e. algorithms that use explicit lists of states and transitions). These tools use the BDD package BuDDy which can be found at http://sourceforge.net/projects/buddy/ and includes documentation for the package.

BDDhisc Algorithms

The BDDhisc algorithms can be accessed from the **BDD Tools** menu of an HPE window. These algorithms include all the standard HISC checks, HISC level-wise synthesis, as well as flat (treats HISC project as if it was a flat project) nonblocking and controllability checks.

The BDDhisc algorithms now support verification of HISC systems that include low data events. They also now contain experimental level-wise synthesis algorithms for HISC systems containing LD events.

When the user selects to do an HISC level-wise synthesis, a dialogue pops up to allow the user to select between different output file formats. If the user selects "DES," the synthesis result will be saved in a "*_sup.txt" text file, one per subsystem. This can produce extremely large files for large examples.

If the user selects "BDD," the synthesis results will be saved in "*.bdd" and "*.dot" files, one for each controllable event. The "*.bdd" is the output format used by BuDDy, and "*.dot" is the file format used by the Graphviz program (http://www.graphviz.org/). They represent the BDD predicate which evaluates to true only for those states of the indicated subsystem at which the event should be enabled. Each event is only given in terms of a single component, as the enablement of an event in HISC is uniquely determined by a single subsystem and relevant interfaces. For more information, see:

Raoguang Song and Ryan J. Leduc, "Symbolic synthesis and verification of hierarchical interface-based supervisory control," Proc. of 8th International Workshop on Discrete Event Systems, pages 419-426, Ann Arbor, USA, July, 2006.

BDDsd Algorithms

The BDDsd algorithms can be accessed from the **BDD Tools** menu of a flat project window. They include nonblocking, untimed controllability, TDES controllability, SD controllability, proper time behavior, S-singular prohibitable behavior, plant completeness, and activity-loop-free checks. Currently, BDDsd requires that each DES in the project must contain a controllable event called **tick**, with alias **t**. This restriction does not apply if you are only doing a nonblocking or untimed controllabilty check.

DESpot does not currently allow one to directly specify which events are forcible in a timed DES. Instead it uses the sampled-data supervisory control assumption that the set of forcible events is equal to the set of prohibitable (controllable non-tick) events.

For more information, see:

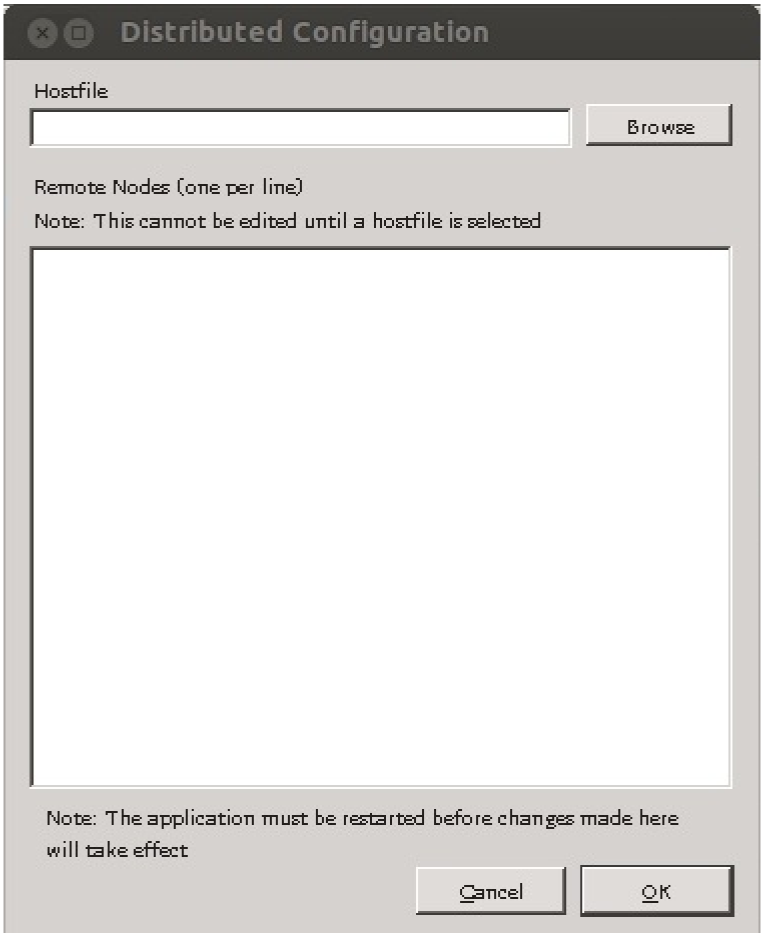
Ryan Leduc and Yu Wang, "Sampled-data supervisory control," in Proc. 10th International Workshop on Discrete Event Systems (WODES'10), pages 353-359, Berlin, Germany, Aug 2010.

Distributed Tools

This feature is only supported for Linux installations. PLEASE NOTE: It is currently broken. A version of DESpot was release where the slave program attempts to run a GUI element, causing it to crash.

Distributed Configuration

From within the Project Editor, select the **Distributed Tools** menu, then click on **Distributed Configuration**. This will open the **Distributed Configuration window**.



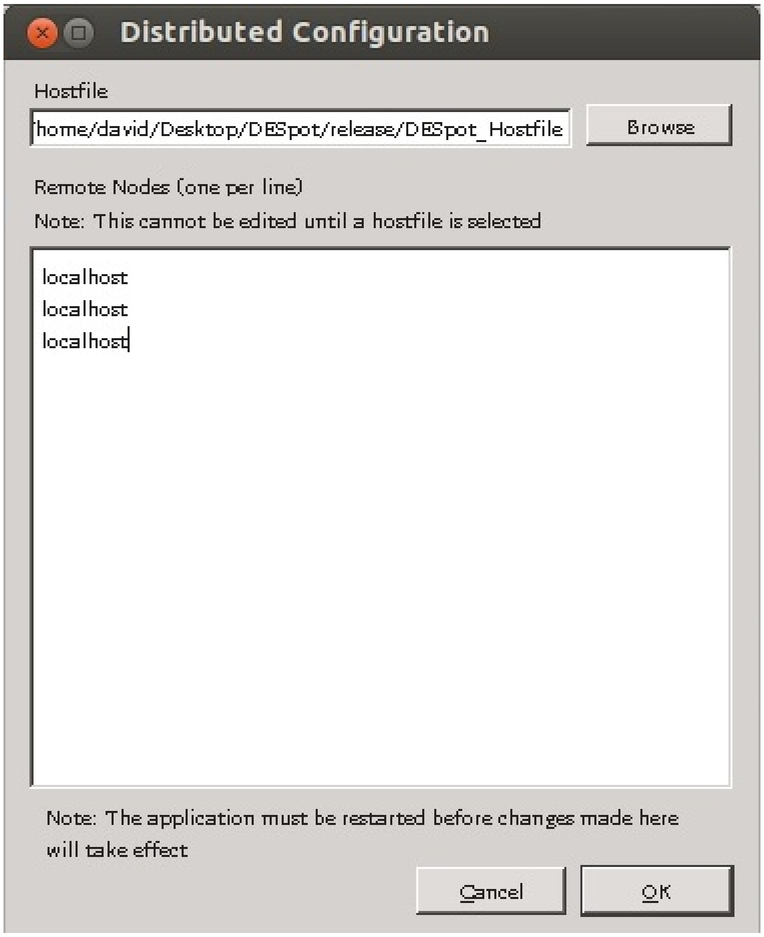
Note: This cannot be edited until a hostfile is selected

Note: The application must be restarted before changes made here will take effect

In this window, you are able to select your hostfile, containing the remote nodes you wish to use, as well as edit the hostfile. The lower text box, used for editing the host file, will not be enabled until a hostfile is selected.

To select the hostfile, click on the browse button at the top. This will open a file browser where you can navigate to the desired hostfile. Please note that the seleted hostfile must match the default hostfile specified in the mca-params.conf file created during installation.

Once you select your hostfile, its path will be loaded into the text box at the top, and its contents will be loaded into the text box at the bottom. DESpot will remember your selected hostfile between sessions, so you will not need to re-select it each time you run DESpot. It is assumed that the hostfile will always contain **localhost** as the first line (corresponding to the host used for the master process), and so this line will not be displayed in the lower text box. Your window will look something like this



Distributed Configuration

Hostfile

/home/david/Desktop/DESpot/release/DESpot_Hostfile

Remote Nodes (one per line)

Note: This cannot be edited until a hostfile is selected

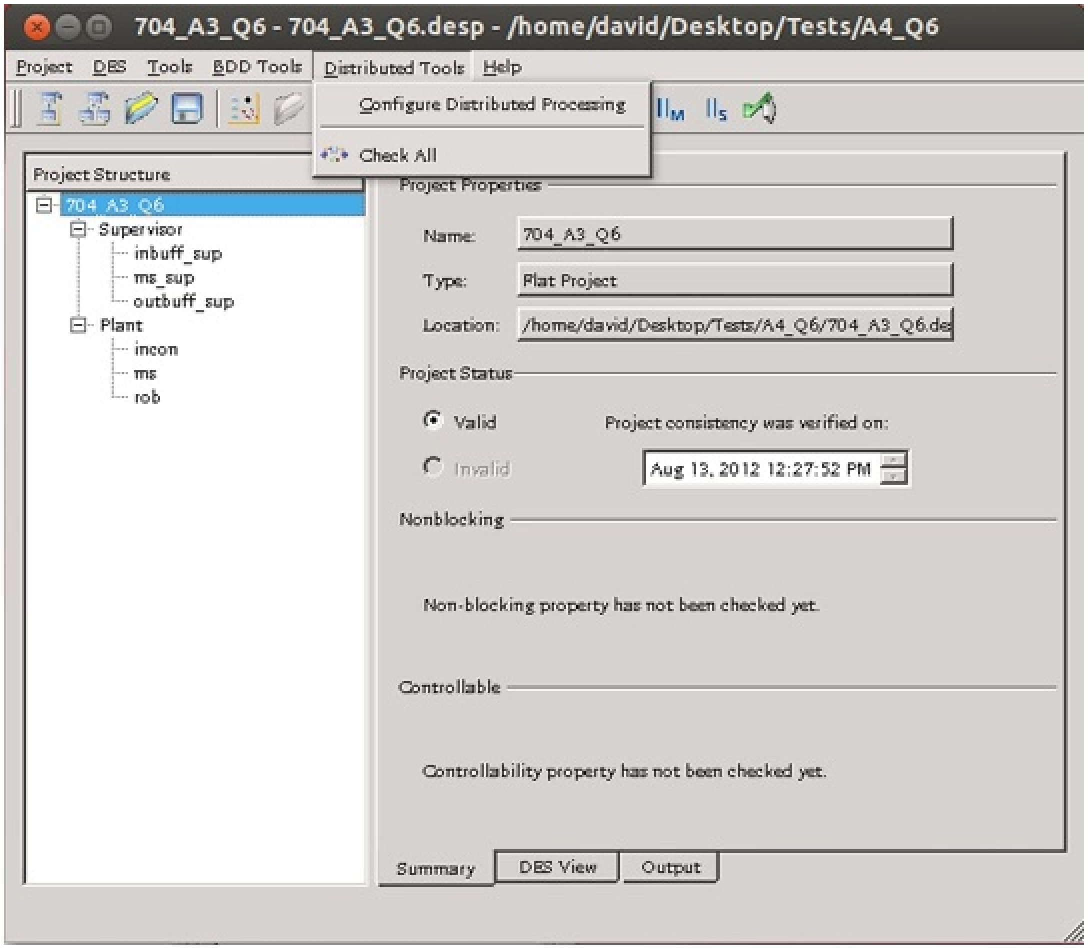
```
localhost
localhost
localhost
```

Note: The application must be restarted before changes made here will take effect

You are now able to edit the contents of the specified hostfile. You must enter each remote node you wish to use on a separate line. If you wish to have multiple slave processes started on the same node, simply add that node on multiple lines as demonstrated in the image above.

To save your changes, click **Ok**. Please note that you must restart DESpot after making any changes to your hostfile.

Distributed Flat Project Checks



Project Properties

Name: 704_A3_Q6

Type: Flat Project

Location: /home/david/Desktop/Tests/A4_Q6/704_A3_Q6.de

Project Status

● Valid

○ Invalid

Project consistency was verified on: Aug 13, 2012 12:27:52 PM

Nonblocking

Non-blocking property has not been checked yet.

Controllable

Controllability property has not been checked yet.

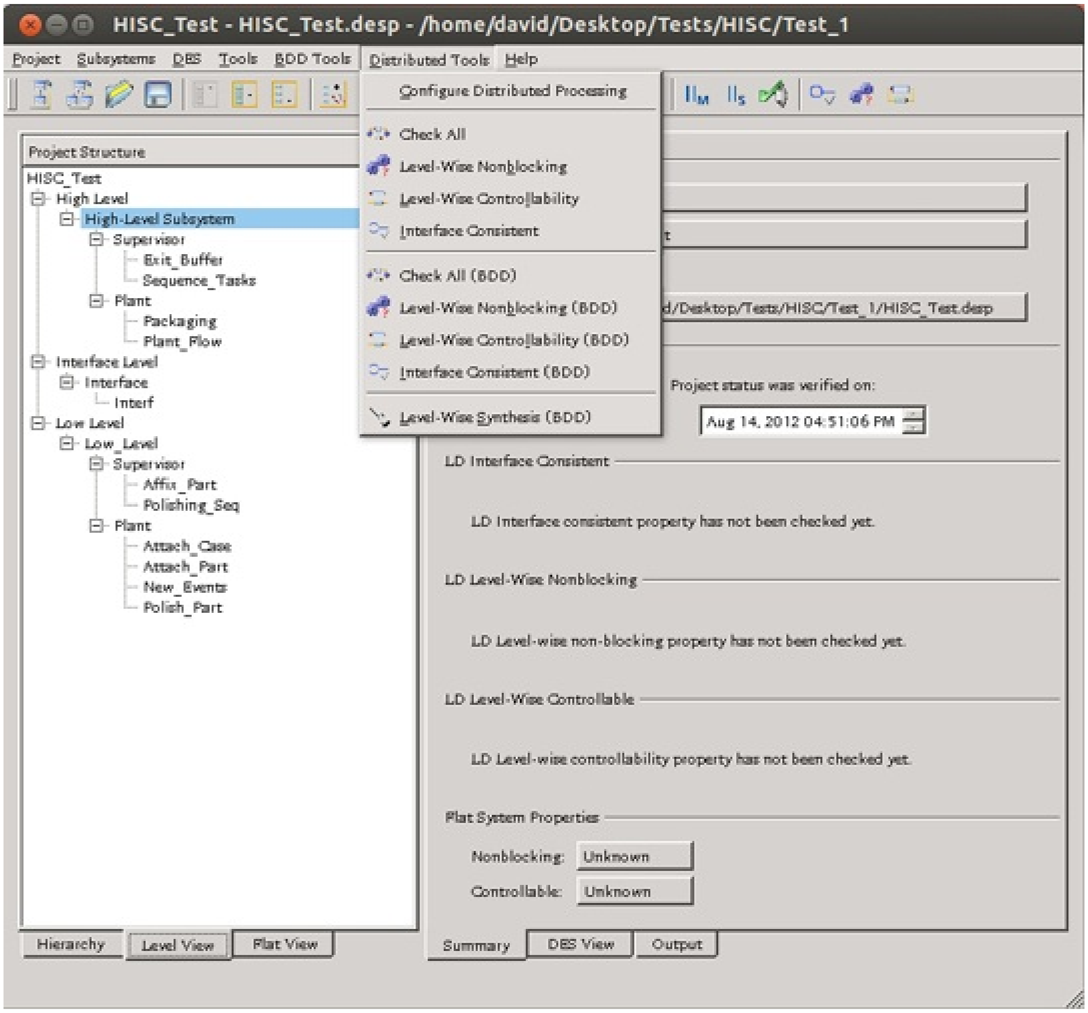
Distributed Configuration must be completed before Distributed Flat Project Checks may be performed.

Flat projects can only perform a Distributed **Check All**. This check will run the project through the **Non-Blocking** and **Controllability** algorithms utilizing, at most, 2 remote nodes.

Please note that projects must be validated and any changes saved before Distributed Computations can be performed.

The output follows the same format as the non-distributed version.

Distributed HISC Project Checks



Project status was verified on: Aug 14, 2012 04:51:06 PM

LD Interface consistent property has not been checked yet.

LD Level-wise non-blocking property has not been checked yet.

LD Level-wise controllability property has not been checked yet.

Nonblocking: Unknown

Controllable: Unknown

Distributed Configuration must be completed before Distributed Flat Project Checks may be performed.

HISC projects can perform considerably more distributed operations than flat projects. You may perform **Level-Wise Non-Blocking** checks, **Level-Wise Controllability** checks, **Level-Wise Interface Consistency** checks, and project wide **Check All** computations, as either standard DES or BDD systems. In additional, you may also perform distributed BDD **Synthesis**.

Unlike the Distributed Flat Project Checks, these checks will attempt to use all of the available nodes specified in your hostfile.

Please note that projects must be validated and any changes saved before Distributed Computations can be performed.

The output follows the same format as the non-distributed DES and BDD checks.

Save a Project

From the Project Editor window of the project you wish to save, select the **Project** menu, then click **Save Project** or **Save Project As...**. This will open up a file browser, where you can choose the location and file name of your project. Located on the top right corner of the file browser, the **Create Folder** button allows one to make a new subdirectory to save the project into.
Please Note:
The project file name is used for identifying your project on disk system whereas the project name is used for identifying your project within DESpot.

Save Project

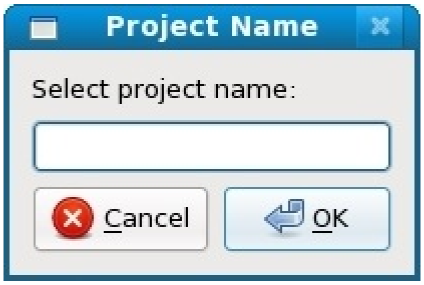
Use the **Save Project** option when you have made changes to a previously saved project and you wish to save the changes using the same project file name. The file browser will only appear when you are saving a new project.

Save Project As...

Use the **Save Project As...** option when you wish to save the project under a new file name. This will keep the original project file unmodified and save a new copy of the project under the new file name.

Set Project Name

From a Project Editor window, select the **Project** menu then click on **Set Name...** This will open the following dialog box.



Enter a new name for the project and click **OK**. Use this option to change the name of your project.

Please Note: The project name is used for identifying your project in DESpot and is separate from the project file name. The file name identifies your project on disk.

Print Project

From a Project Editor window select the **Project** menu, then select **Print to file...** This will translate your project into a text file (*.txt) and save it in the file path of your choice. The text file will include full details on the project as well as all the DES the project contains.

Edit DES

DESpot offers both a Tabular and Graphical DES editor. By default, the graphical editor is used on **Edit DES** commands. To change this, Select the **DES** menu and use the radio button to change the default editor.

From a Project Editor window, highlight the DES you wish to edit by clicking on it.

From the **DES** menu, select **Edit DES** or click on the **Edit DES** button from the toolbar. You can also simply double-click on the DES you wish to edit and the appropriate editor will open the DES for editing.

Remove a DES from a Project

From a Project Editor window, highlight the DES you wish to remove by clicking on it. Then from the menu bar, select **DES | Remove DES** or click on the **Remove DES** button on the toolbar. This will remove the selected DES from the project. If the DES is a template, then it will also remove all instantations belonging to the template as well.

DES Editor

1. Tabular Editor

2. Graphical Editor

3. DES Tools

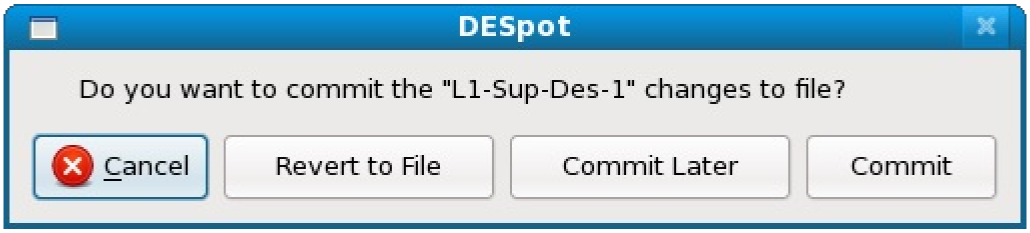
4. Save DES

5. Set DES Name

6. Set DES Type

7. Print DES

Please note that when you make a change to a DES in a project using a DES editor, the change is made immeditately to the project's DES data structure and is seen by the entire project. If you try to close the editor without saving the DES, the dialog below pops up.

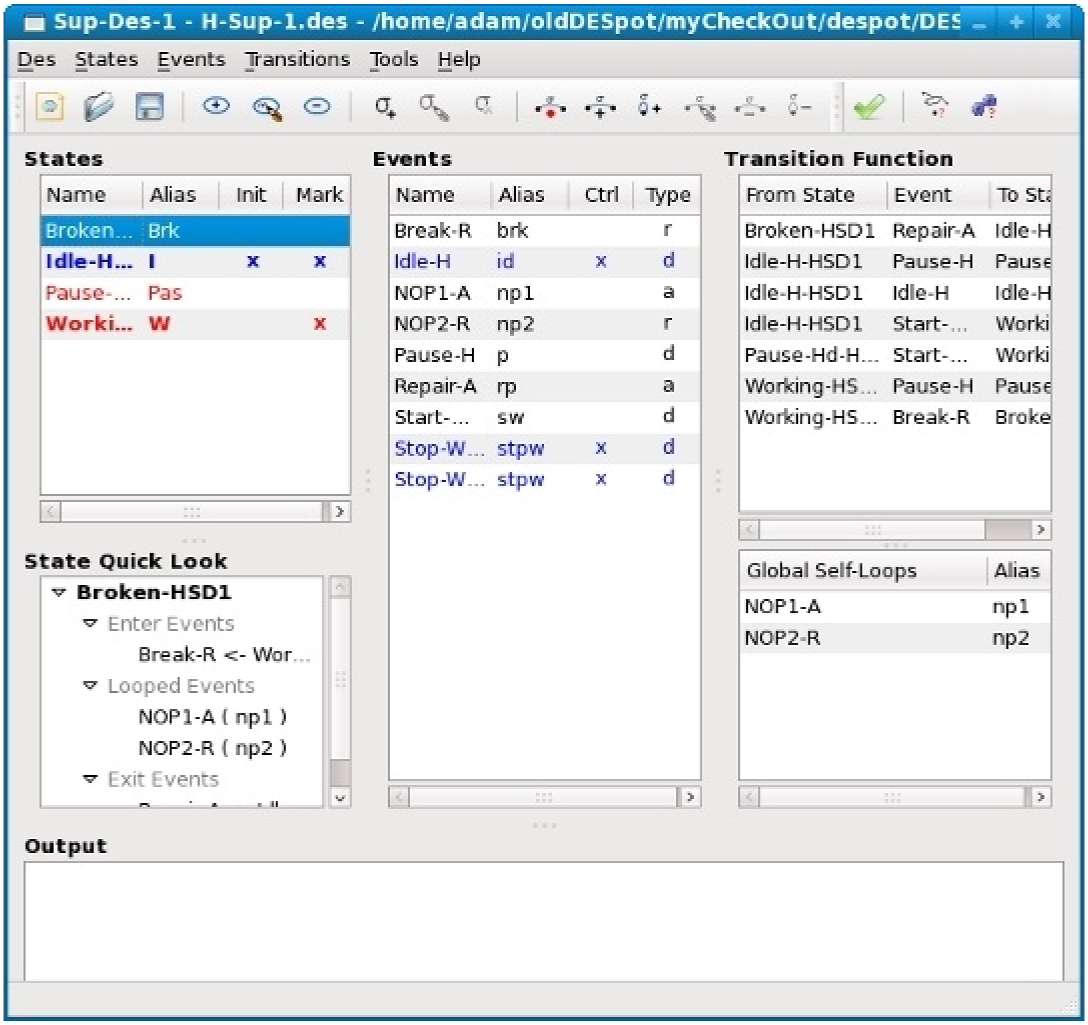


Do you want to commit the "L1-Sup-Des-1" changes to file?

Click on **Cancel** to return to the editor, **Revert To File** to discard unsaved changes and reload the DES from its file, **Commit Later** to keep the unsaved changes but not save changes to disk, and **Commit** to save the changes to disk.

Tabular Editor

When a DES is created or opened in the **Tabular Editor**, the following window is displayed.


Sup-Des-1 - H-Sup-1.des - /home/adam/oldDESpot/myCheckOut/despot/DES
Des States Events Transitions Tools Help
States
Events
Transition Function
State Quick Look
Output

Name	Alias	Init	Mark
Broken...	Brk		
Idle-H...	I	x	x
Pause-...	Pas		
Worki...	W		x

Name	Alias	Ctrl	Type
Break-R	brk		r
Idle-H	id	x	d
NOP1-A	np1		a
NOP2-R	np2		r
Pause-H	p		d
Repair-A	rp		a
Start-...	sw		d
Stop-W...	stpw	x	d
Stop-W...	stpw	x	d

From State	Event	To Sta
Broken-HSD1	Repair-A	Idle-H
Idle-H-HSD1	Pause-H	Pause
Idle-H-HSD1	Idle-H	Idle-H
Idle-H-HSD1	Start-...	Worki
Pause-Hd-H...	Start-...	Worki
Working-HS...	Pause-H	Pause
Working-HS...	Break-R	Broke

Global Self-Loops	Alias
NOP1-A	np1
NOP2-R	np2

- Broken-HSD1
  - Enter Events
    - Break-R <- Wor...
  - Looped Events
    - NOP1-A (np1)
    - NOP2-R (np2)
  - Exit Events

The DES Tabular Editor window allows you to create and edit DES as well as save them into DES files (*.des). Through the editor, you can validate a DES and can verify different DES properties such as reachability and nonblocking.

The Tabular Editor window has 4 main parts:

- The state editor composed of two panels: the state list and the state quick look. The state list simply lists all the states and their properties. The state quick look pane shows the enter-events, loops and exit-events for the currently selected state.
- The event editor lists all the events in the event set of the DES.
- The transition editor is split into two parts: the top part shows the regular transitions; the bottom parts shows transitions that are self-looped at every state in the DES.

In addition, the Tabular editor also has an output panel where the output of the different algorithms is shown.

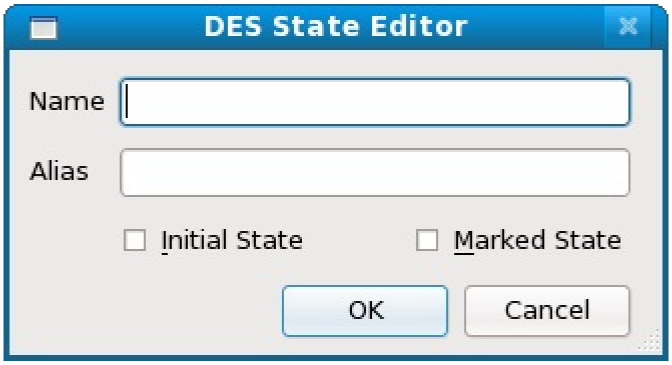
To create a DES, in general, you must first create a state, then an event and then connect that state to another state in the form of a transition. Note that states, events, and transitions offer access to related functionality via context menus.

Context menus are opened by right clicking on the object.

TAB Editor: States

TAB Editor: Creating a State

From the Tabular Editor window, select the **States** menu, then click **Add State**, or from the toolbar click on the **Add State** button ⊕. This will open the **DES State Editor** dialog box.



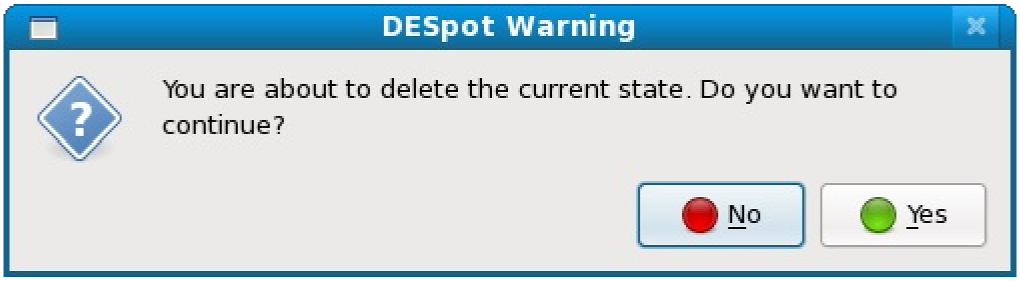
Enter the *Name* and *Alias* of the state in the text boxes. The *Name* is used to identify the state and the *Alias* is a short-form name for the state that is used when display space is limited. Use the check boxes to indicate whether the state will be an *Initial* and/or a *Marked* state. Once you have entered the information, click **OK** and the state will be added to the DES. The new state will appear in the **States** pane.

TAB Editor: Editing a State

From the Tabular Editor window, select the state you wish to edit in the **States** pane. Next, open the **States** menu and select **Change State** or click on the **Change State** button on the toolbar . This will reopen the **DES State Editor** dialog box. Edit the state properties you wish to change and click **OK**. This will update the state properties in the **States** pane. You can also edit a state by double-clicking on the desired property of the state in the state list.

TAB Editor: Deleting a State

From the Tabular Editor window, select the state you wish to delete in the **States** pane. Then open the **States** menu and select **Delete State** or click on the **Delete State** button on the toolbar . This will display a warning message about the state deletion. Note that deleting a state will also remove any transitions that contained that state.


DESpot Warning

You are about to delete the current state. Do you want to continue?

Confirm your request to delete by clicking **OK** and the state will be removed from the DES.

TAB Editor: Initial State

To toggle the **Initial** property of a state, select the state you wish to change and then open the **States** menu and select **Initial State**. Alternatively, you can open the DES State Editor to change the **Initial** property.

Please Note: Only one state at a time can have the **Initial** property.

TAB Editor: Marked State

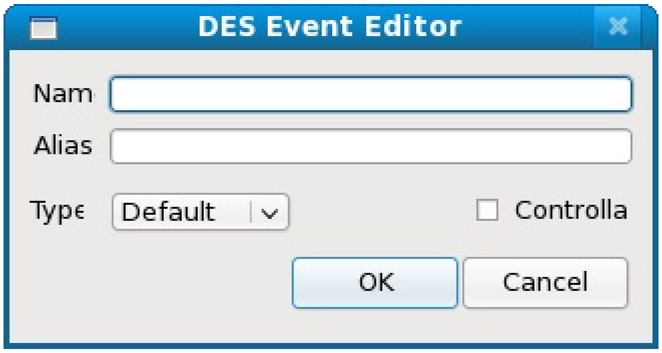
To toggle the **Marked** property of a state, select the state you wish to change and then open the **States** menu and select **Marked State**. Alternatively, you can open the DES State Editor to change the **Marked** property.

TAB Editor: Events

1. Create Event
2. Change Event
3. Delete Event
4. Controllable Event
5. Event Type

TAB Editor: Creating an Event

From the Tabular Editor window, select the **Events** menu, then click **Add Event**, or from the toolbar click on the **Add Event** button σ+. This will open the **DES Event Editor** dialog box.



Enter the *Name* and *Alias* of the event in the text boxes. The *Name* is used to identify the event and the *Alias* is a short-form name for the event that is used when display space is limited. Select the *Type* of the event using the drop-down box.

There are four *Types* of events that can be added to a DES:

- Default: Used for DES in a flat project or for a DES that are part of a high-level or low-level subsystem (corresponding to a high-level or low-level event, as appropriate) in an HISC project.
- Answer: Used for answer events in DES belonging to an HISC project.
- Request: Used for request events in DES belonging to an HISC project.
- Low Data: Used for low data events in DES belonging to an HISC project.

Lastly, indicate whether you want the event to be *Controllable* or not using the checkbox. Once you have entered all the event information, click **OK** and the event will be added to the DES. The new event will appear in the **Events** pane.

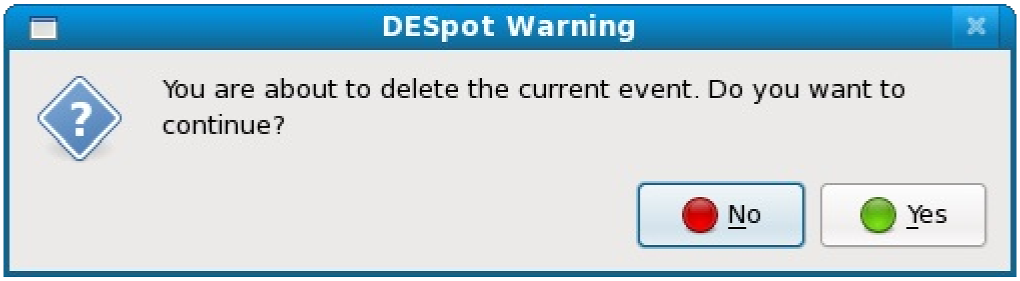
TAB Editor: Editing an Event

From the Tabular Editor window, select the event you wish to edit in the **Events** pane. Next, open the **Events** menu and select **Change Event** or click on the **Change Event** button on the toolbar . This will reopen the **DES Event Editor** dialog box. Edit the event properties you wish to change and click **OK**. This will update the event properties in the **Events** pane. You can also edit an event by double-clicking on the desired property of the event in the event list.

TAB Editor: Deleting an Event

From the Tabular Editor window, select the event you wish to delete in the **Events** pane. Then open the **Events** menu and select **Delete Event** or click on the **Delete**

Event button on the toolbar . This will display a warning message about the event deletion.



DESpot Warning

You are about to delete the current event. Do you want to continue?

Confirm your request to delete by clicking **OK** and the event will be removed from the DES. Deleting the event will also remove any transitions that contain the event.

TAB Editor: Controllable Event

To toggle the **Controllable** property of an event, select the event you wish to change and then open the **Events** menu and select **Controllable Event**. Alternatively, you can open the DES Event Editor to change the **Controllable** property.

TAB Editor: Event Type

To toggle the **Type** property of an event, select the event you wish to change and then open the **Events** menu and then the **Set Type** submenu. Now select the new **Type**. Alternatively, you can open the DES Event Editor to change the **Type** property.

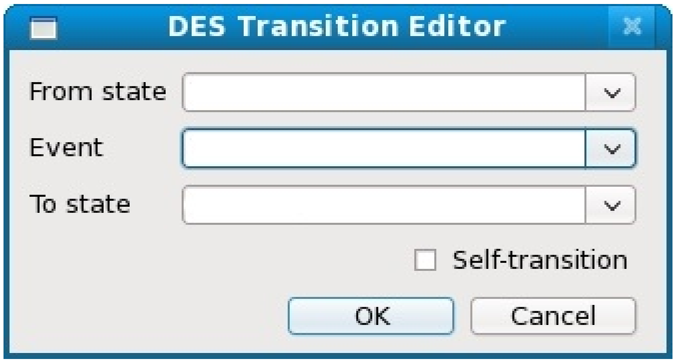
TAB Editor: Transitions

4. Create Global Self-Loop

5. Delete Global Self-Loop

TAB Editor: Creating a Transition

From the Tabular Editor window, select the **Transitions** menu, then click **Add Transition**, or from the toolbar click on the **Add Transition** button. This will open the **DES Transition Editor** dialog box.



Using the **From state** drop-down box, select the state that the transition will start at, once the transition occurs.
Using the **Event** drop-down box, select the event that will execute the transition.
Using the **To State** drop-down box, select the state that the DES will move to when the transition occurs.

By checking the **Self-transition** box, a self loop will be created using the selected **From State** and **Event**.

Another way to create a transition is by using the **Record Transition** function. From The **Transitions** menu select **Record Transitions** or from the toolbar click the **Record Transitions** button. This will start recording transitions based on the states and events you click. The **Output** pane shows you what part of the transition you need to select next. Once the **Record Transitions** function has started, in the **States** pane, click on the *Start State* for your transition. Next, click on the *Event* in the **Events** pane that you want to execute your transition. Finally, select the *End State* from the **States** pane. Once you select these three items, you will notice a new transition will be added to the **Transitions** pane. Be aware that you can continue clicking on states and events to create more transitions. To exit the **Record Transitions** function, click the **Record Transitions** button again.

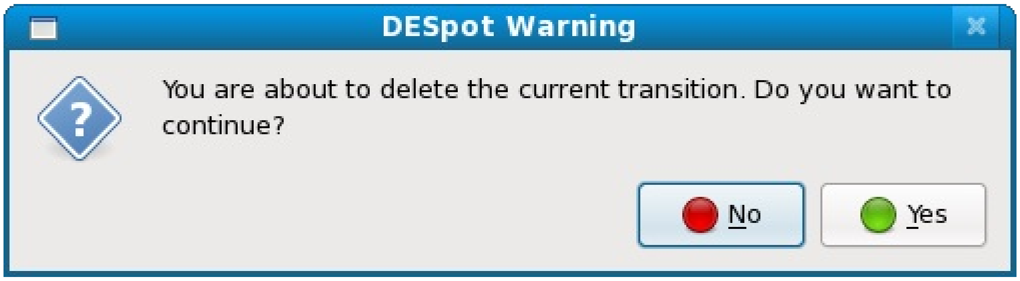
Please Note: DESpot only allows the creation of deterministic DES, thus the editor will not allow the creation of a transitions leaving a given state if there already exists another transition leaving the same state with the same event.

TAB Editor: Editing a Transition

From the Tabular Editor window, select the transition you wish to edit in the **Transition** pane. Then open the **Transition** menu and select **Change Transition** or click on the **Change Transition** button on the toolbar . This will reopen the **DES Transition Editor** dialog box. Edit the transition properties you wish to change and click **OK**. This will update the transition properties in the **Transition** pane. You can also edit a transition by double-clicking on the desired property of the transition in the transition list.

TAB Editor: Deleting a Transition

From the Tabular Editor window, select the transition you wish to delete in the **Transition** pane. Then open the **Transitions** menu and select **Delete Transition** or click on the **Delete Transition** button on the toolbar . This will display a warning message about the transition deletion.



DESpot Warning

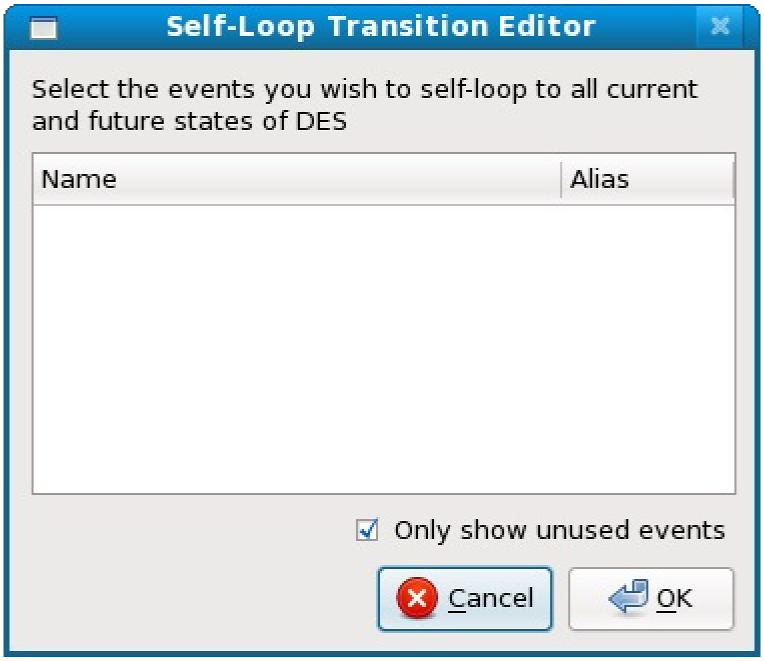
You are about to delete the current transition. Do you want to continue?

Confirm your request to delete by clicking **OK** and the transition will be removed from the DES.

TAB Editor: Adding a Global Self-Loop

From the Tabular Editor window, select the **Transitions** menu, then click **Add Global**

Self-Loop, or from the toolbar click on the **Add Global Self-Loop** button σ+ . This will open the **Self-Loop Transition Editor** dialog box.



Self-Loop Transition Editor

Select the events you wish to self-loop to all current and future states of DES

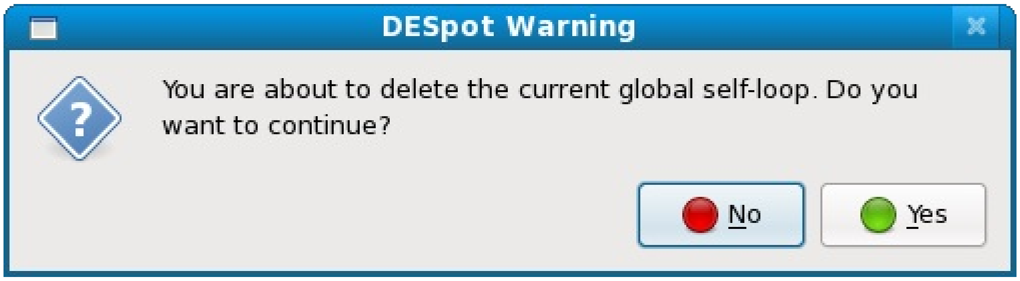
Name	Alias

☑ Only show unused events

By default, the window only shows events that have not been used. This can be changed by unchecking the **Only show unused events** checkbox. Select the event you wish to use from the list and click **OK**. The global self-loop will be added to the **Global Self-Loops** pane. This will create a self-loop for every state of the DES using the selected event.

TAB Editor: Deleting a Global Self-Loop

From the [Tabular Editor](#) window, select the global self-loop you wish to delete from the**Global Self-Loops** pane. Next, open the**Transitions** menu and click **Delete Global Self-Loop**, or from the toolbar click on the **Delete Global Self-Loop** button σ− . This will display a warning message about the transition deletion.

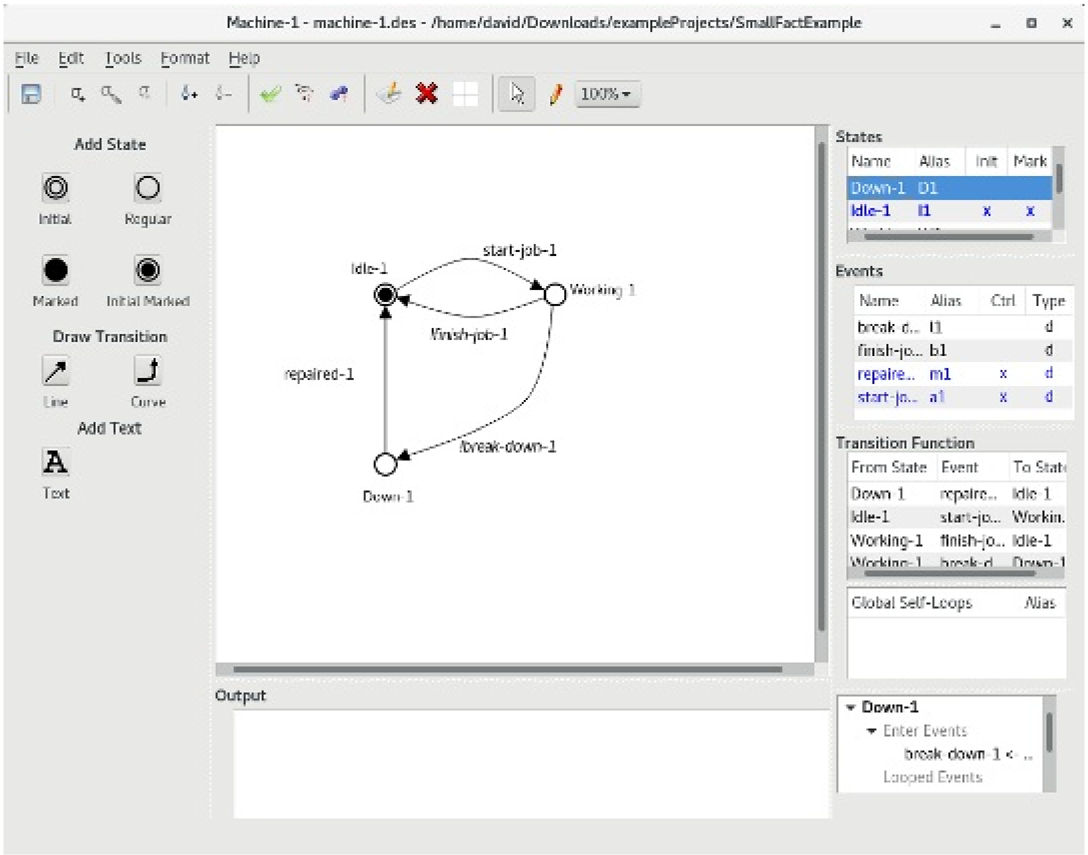


You are about to delete the current global self-loop. Do you want to continue?

Confirm your request to delete by clicking **OK** and the global self-loop will be removed from the DES.

Graphical Editor

When a DES is created or opened in the **Graphical Editor**, the following window is displayed.


Machine-1 - machine-1.des - /home/david/Downloads/exampleProjects/SmallFactExample

States

Name	Alias	Init	Mark
Down-1	D1		
Idle-1	I1	x	x

Events

Name	Alias	Ctrl	Type
break-d...	I1		d
finish-jo...	b1		d
repaire...	m1	x	d
start-jo...	a1	x	d

Transition Function

From State	Event	To State
Down-1	repaire...	Idle-1
Idle-1	start-jo...	Workin...
Working-1	finish-jo...	Idle-1
Working-1	break-d...	Down-1

Global Self-Loops	Alias

- Down-1
  - Enter Events
    - break-down-1 <- ..
  - Looped Events

The DES Graphical Editor is very similar to the Tabular Editor in that it allows you to create and edit DES as well as save them into DES files (*.des). The main difference is that in the graphical version, you design the DES' layout graphically as well. An optional Grid is provided to help align various DES components. Through the

Graphical Editor, you can validate a DES and verify its properties.

If a DES is opened that has no graphical display information, the editor will display the DES in a default fashion. The user can then reposition the DES' components as desired and save the results.

The Graphical Editor window has 4 main parts:

- The right hand side contains the **States**, **Events**, **Transitions**, **Global Self-Loops** and **State Quick Look** panes that are found in the Tabular Editor. One difference is that you can't double-click on an element to edit them in the Graphical Editor.
- The **Output** panel is located at the bottom of the window which displays feedback from the various algorithms and checks.
- The left side contains tool buttons used to **Add DES Components** like states and transitions. When you are finished adding a DES component, click on the Select mode to return to normal editing. The **Text** button allows you to add a text field to the window for documentation purposes. Simply click on the **Text** button and then click on the place in the editor window where you want the text to appear. To start entering text, triple-click on the text box.

  Note: The **Text** button is currently broken. You can create a text item, but it is not currently being saved with the DES.
- The main area is where the DES is displayed graphically and where you can interact with the DES. Uncontrollable events are indicated in the diagram by showing the event's name in italics and by adding an "!" before the event's name.

Another added feature of the Graphical Editor is that you can save an Image of your DES to disk.

It is possible to move the entire DES at once by clicking outside the DES and dragging the rubber band image around the entire DES. Next, click on a state and drag the DES to the desired new position. Then, click outside of the DES to deselect.

GRPH Editor: Editing Modes

When using the Graphical Editor, there are several modes. Two of these modes are for editing existing DES elements, and the remaining are for adding elements to the DES.

The first editing mode is **Selection** mode. It is used for selecting and moving components of the DES.

The second is **Transition Shape Editor** mode It is used for adding or moving inflection points on a transition line or curve.

To enter either of these modes, click on the appropriate button in the toolbar:

GRAPH Editor: States

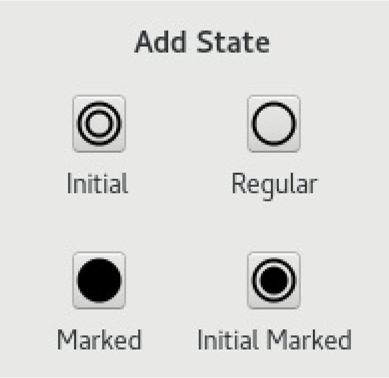
1. Create State

2. Edit State

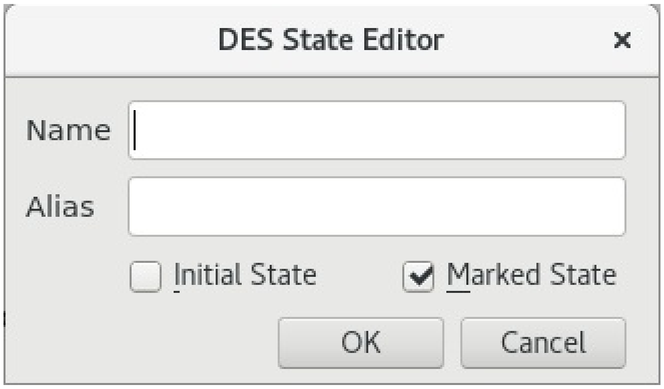
3. Delete State

GRAPH Editor: Creating a State

To add a state to your DES, click the button corresponding to the type of state to add in the toolbox on the left:



Next click in the DES editor workspace where you wish to place the state. This will open the **DES State Editor** dialog box.



DES State Editor

Name

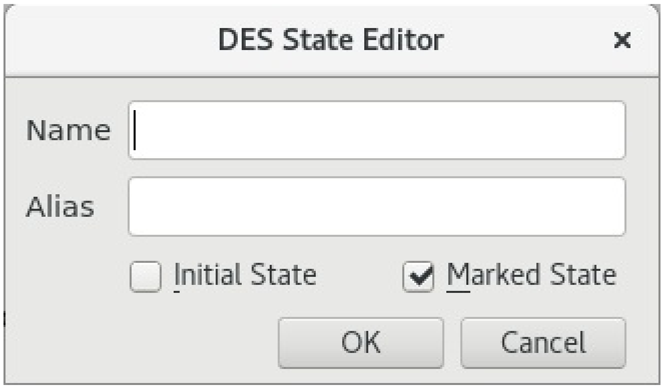
Alias

☐ Initial State ☑ Marked State

Enter the state *name* and *alias* in the text boxes. If you selected the wrong type of state you can change it by clicking the checkboxes for the *initial* and *marked* properties. Click **OK** and your new state will be added to the DES. You will notice your new state listed in the states pane.

GRAPH Editor: Editing a State

To edit a state make sure you are in Selection mode. Then select the state you wish to edit and click on the **Edit** button on the toolbar. Alternatively, right-click on the state and select **Edit**. This will reopen the **DES State Editor** dialog box.



DES State Editor

Name

Alias

☐ Initial State ☑ Marked State

Change the properties you wish and click **OK**.

To move a state to a different graphical position in the DES, make sure that you are in Selection mode. Click on the state you wish to move and drag it to a different position in the DES. This will also move the **State Name Label**. You can move just the **State Name Label** by clicking and dragging the label to a new position in the DES.

Notice when you have a state selected, the **State Name Label** will also be highlighted with a dotted border.

GRAPH Editor: Deleting a State

To delete a state make sure you are in Selection mode.

Select the state you wish to delete and click the **Delete** button on the toolbar ✖. Alternatively, right-click on the state and select **Delete** or simply press the *Delete* key when the state is selected. You will notice the state will be removed from the DES editor workspace and from the **States** pane.

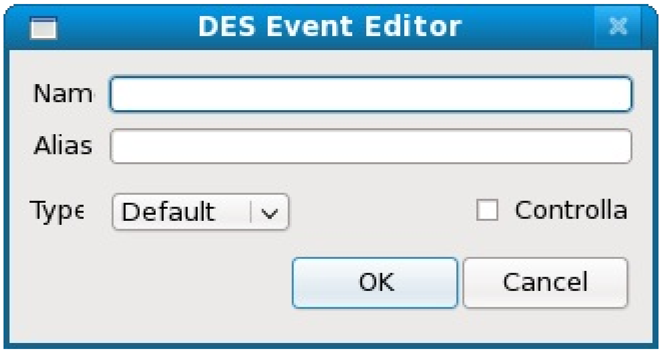
Please Note: Deleting a state will delete all transitions to and from that state.

GRAPH Editor: Events

1. Create Event
2. Change Event
3. Delete Event
4. Controllable Event
5. Event Type

GRAPH Editor: Creating an Event

From the Graphical Editor window, select the **Edit** menu, then click **Add Event**, or from the toolbar click on the **Add Event** button σ+. This will open the **DES Event Editor** dialog box.



Enter the *Name* and *Alias* of the event in the text boxes. Select the *Type* of the event using the drop-down box.

There are four *Types* of events that can be added to a DES:

- Default: Used for DES in a flat project or for a DES that are part of a high-level or low-level subsystem (corresponding to a high-level or low-level event, as appropriate) in an HISC project
- Answer: Used for answer events in DES belonging to an HISC project.
- Request: Used for request events in DES belonging to an HISC project.
- Low Data: Used for low data events in DES belonging to an HISC project.

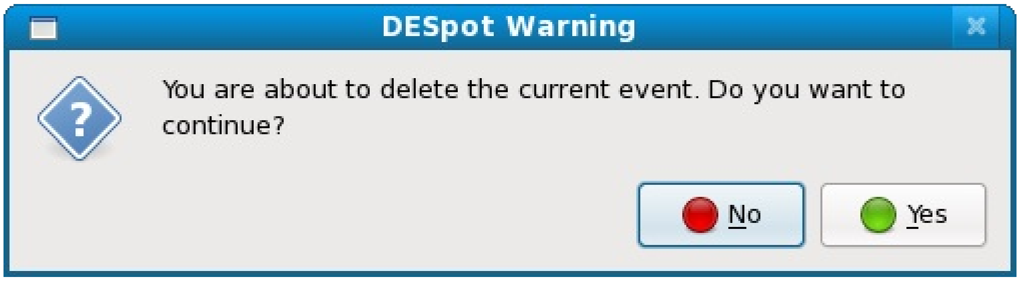
Lastly, indicate whether you want the event to be *Controllable* or not using the checkbox. Once you have entered all the event information, click **OK** and the event will be added to the DES. The new event will appear in the **Events** pane.

GRAPH Editor: Editing an Event

From the Graphical Editor window, select the event you wish to edit in the **Events** pane. Next, open the **Events** menu and select **Change Event** or click on the **Change Event** button on the toolbar . This will reopen the **DES Event Editor** dialog box. Edit the event properties you wish to change and click **OK**. This will update the event properties in the **Events** pane.

GRAPH Editor: Deleting an Event

From the Graphical Editor window, select the event you wish to delete in the **Events** pane. Then open the **Edit** menu and select **Delete Event** or click on the **Delete Event** button on the toolbar . This will display a warning message about the event deletion.


DESpot Warning

You are about to delete the current event. Do you want to continue?

Confirm your request to delete by clicking **OK** and the event will be removed from the DES.

Please Note: Deleting an event will delete all transitions associated with that event.

GRAPH Editor: Controllable Event

To toggle the **Controllable** property of an event, select the event you wish to change and then open the **Edit** menu and select **Controllable Event**. Alternatively, you can open the DES Event Editor to change the **Controllable** property.

GRAPH Editor: Event Type

To toggle the **Type** property of an event, select the event you wish to change and then open the **Edit** menu and then the **Set Type** submenu. Now select the new **Type**. Alternatively, you can open the DES Event Editor to change the **Type** property.

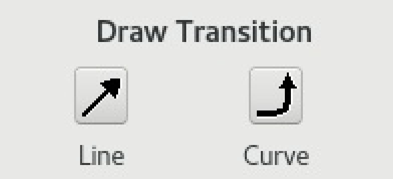
GRAPH Editor: Transitions

1. Create Transition
2. Edit Transition
3. Delete Transition
4. Create Global Self-Loop

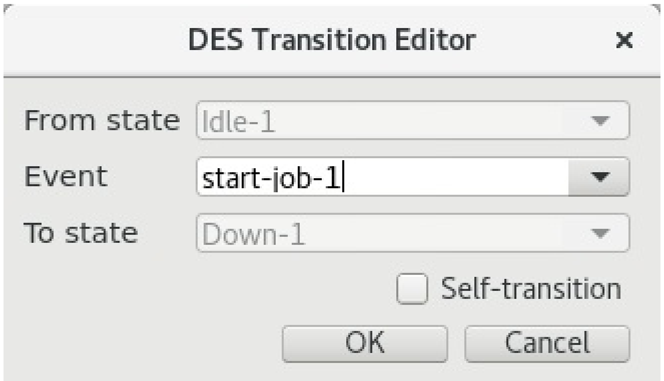
5. Delete Global Self-Loop

GRAPH Editor: Creating a Transition

To add a transition to your DES, first make sure you have added the events for the transition. Then click either the line or curve transition buttons in the toolbox on the left.



To start a transition, click and hold on the *start* or *from state* of your transition and drag your mouse cursor to the *stop* or *to state*. You will notice a transition line follow your mouse cursor until you release your mouse click at the *stop* state. Now the transition line will connect the two states. If you only click on your *start* state or drag your transition line to the same *start* and *stop* state you will create a self-loop. Once you place your transition line, the **DES Transition Editor** dialog box will appear.



From state: Idle-1

Event: start-job-1

To state: Down-1

☐ Self-transition

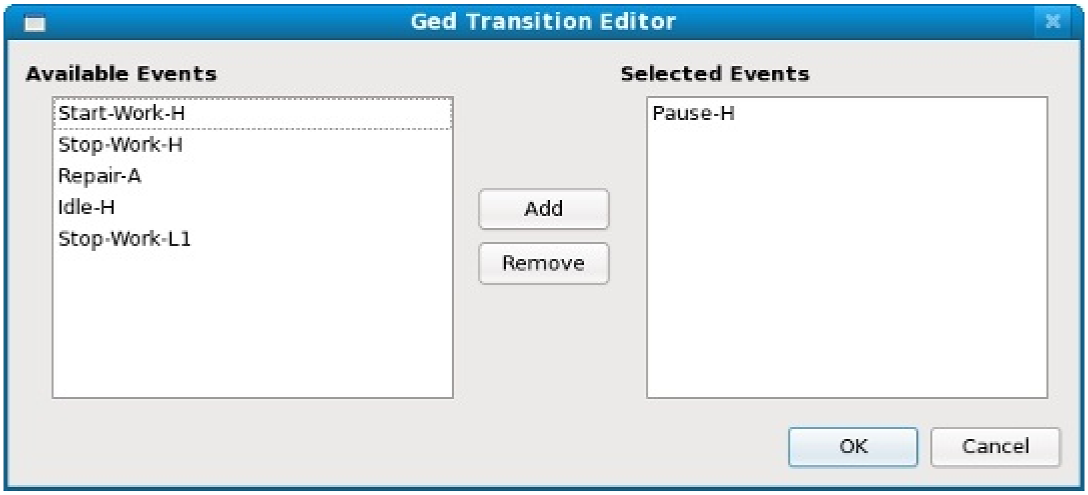
This is where you select the event for the transition. If you click the **Self-transition** checkbox, a self-loop will be created and your *To state* will now match your *From state*. When you are done selecting the event, click **OK** and the transition will be added to the **Transition** pane. You will also notice a new label appear next to your transition line with the event name you selected.

GRAPH Editor: Editing a Transition

Editing Events

To edit a transition in your DES, make sure you have Selection mode enabled. Select the transition you wish to edit by clicking on the transition line in your DES.

Now click on the **Edit** button in the toolbar . Alternatively, right-click on the transition in your DES and click **Edit** from the popup menu. This will open the **Ged Transition Editor** window.



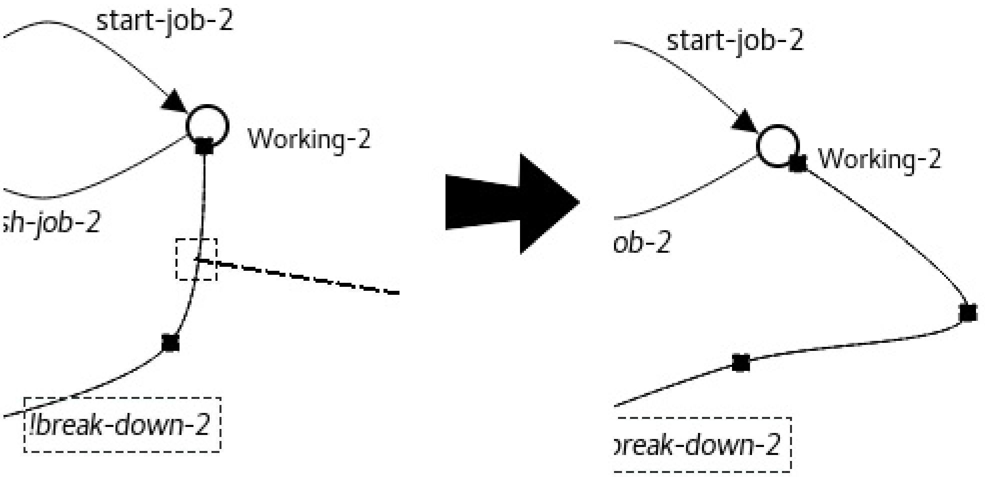
From this window you can add and remove events from the transition you are editing. To add an event to the transition, select the event in the **Available Events** pane and click **Add**. This will move the event to the **Selected Events** pane. To remove an event from the transition, select the event in the **Selected events** pane and click **Remove**. This will move the event to the **Available Events** pane. If you wish to change the *start* or *stop* state for the transition, you must first Delete the transition then Create a new transition with the proper states.

You can also click on the transition label and drag it to a new position. If unsure which label belongs to your transition, clicking on a transition arrow will cause its label to be highlighted with a dotted border.

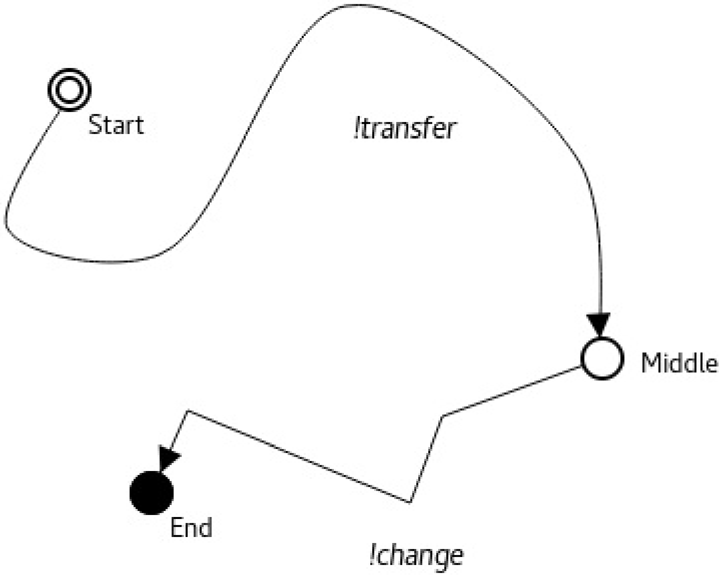
Editing the Transition's Shape

To edit the graphical representation of the transition (i.e the transition line) first make sure you have Transition Shape Editor mode enabled.

Click and drag a point on the existing transition line or curve to a new space in the DES editor. When you release the mouse, an intermediary point will be added to the transition representation.



If the transition is represented by a line, the line will be split into two segments, connected at the point where the left mouse button was released. If the transition is represented by a curve, then the curve will go smoothly through the added point. You can add as many new vertices to the transition representation as you would like using the same method.



If you are unsatisfied with your changes, you can right-click on the transition line and select **Default Shape**. This will reset the line back to the original shape.

To rotate a self-loop around its state, click on the circle and drag to a new position. Editing a transition shape is especially useful when you have more than one transition terminating at the same state.

Changing Between Lines and Curves

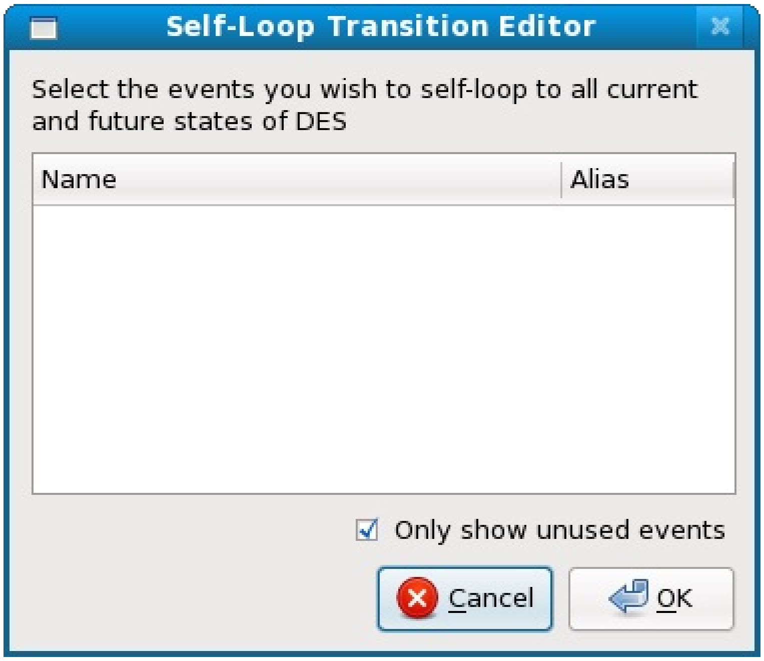
If you would like to switch a particular transition from using curves to using lines for its representation (or vice-versa), enter Selection mode . Then, right click on the transition representation that you would like to change. Select the **Toggle** Curvature option in the context menu.

GRAPH Editor: Deleting a Transition

To delete a transition from your DES, make sure you have Selection mode enabled. Select the transition you wish to remove by clicking on its transition line. Now click the **Delete** button on the toolbar ✖. Alternatively, you can right-click on the transition line and select **Delete** or simply press the *Delete* key when the transition is selected. You will notice the transition will be removed from the DES editor workspace and from the **Transition** pane.

GRAPH Editor: Adding a Global Self-Loop

From the Graphical Editor window open the **Edit** menu and select **Add Global Self-Loop** or from the toolbar click on the **Add Global Self-Loop** button . This will open the **Self-Loop Transition Editor** dialog box.



Self-Loop Transition Editor

Select the events you wish to self-loop to all current and future states of DES

Name	Alias

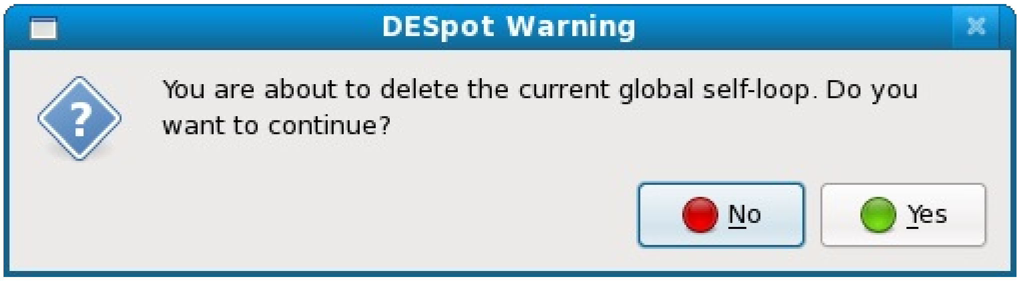
☑ Only show unused events

By default, the window only shows events that have not been used. This can be changed by unchecking the **Only show unused events** checkbox. Select the event you wish to use from the list and click **OK**. The global self-loop will be added to the **Global Self-Loops** pane. This will create a self-loop for every state of the DES using the selected event.

Please Note: Global self-loops are not displayed graphically in order to save space and clutter in the DES editor workspace.

GRAPH Editor: Deleting a Global Self-Loop

From the Graphical Editor window, select the global self-loop you wish to delete from the**Global Self-Loops** pane. Next, open the**Edit** menu and click **Delete Global Self-Loop**, or from the toolbar click on the **Delete Global Self-Loop** button σ− . This will display a warning message about the transition deletion.



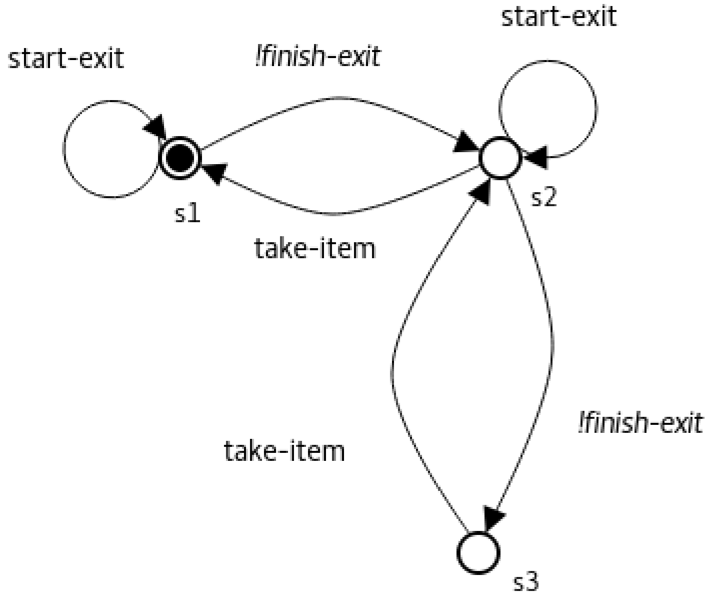
DESpot Warning

You are about to delete the current global self-loop. Do you want to continue?

Confirm your request to delete by clicking **OK** and the global self-loop will be removed from the DES.

GRAPH Editor: Exporting DES to Image File

To save an image of your DES, open the **File** menu and select **Export to image**. This will open a file browser. Choose the location you wish to save your DES image and choose the file type of your image, either Encapsulated PostScript (*.eps), PostScript (*.ps), or Portable Network Graphic (*.png). Below is an example of a DES that has been exported to a PNG image.

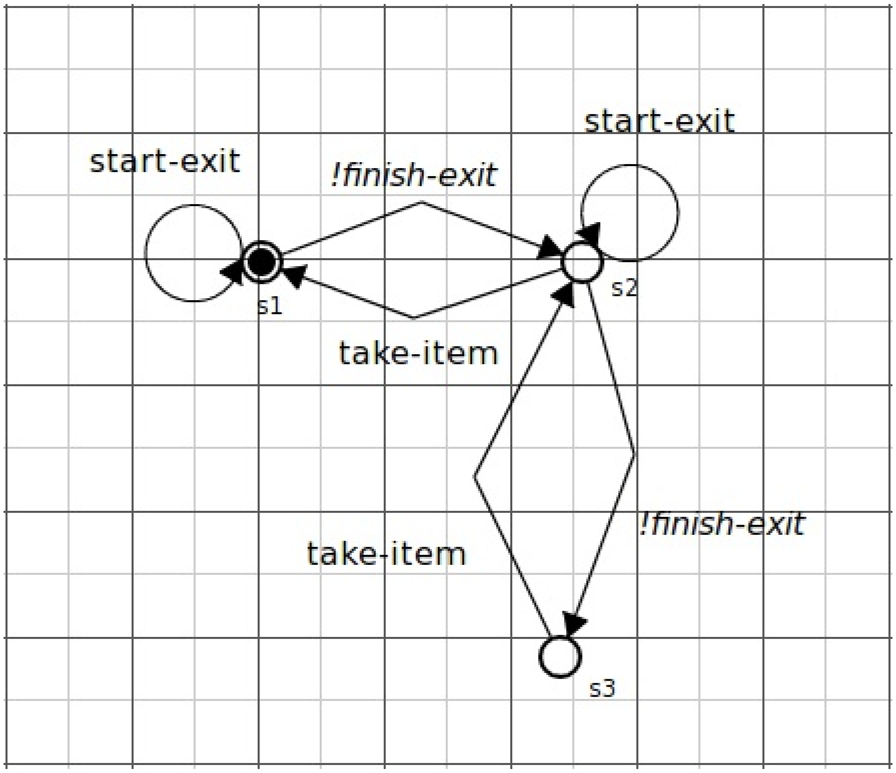


You can choose to export images for all DES in your project by selecting **Export All** from the **File** menu, then clicking on the desired image type (*.eps, *.ps or *.png). An image of each DES will be created in a subdirectory called `images`. Please note that only images that have been created or edited in the graphical editor will be exported as images. This will export all DES, including DES that are instantiations of templates.

You can also export images for all template DES in the project. Select **File**, **Export All Template**, then select the format that you want to export as. It follows the same process as exporting all DES, placing each image in the subfolder `images`. This option will only export images of template DES, not any instantations or any manually created DES.

GRAPH Editor: Grid

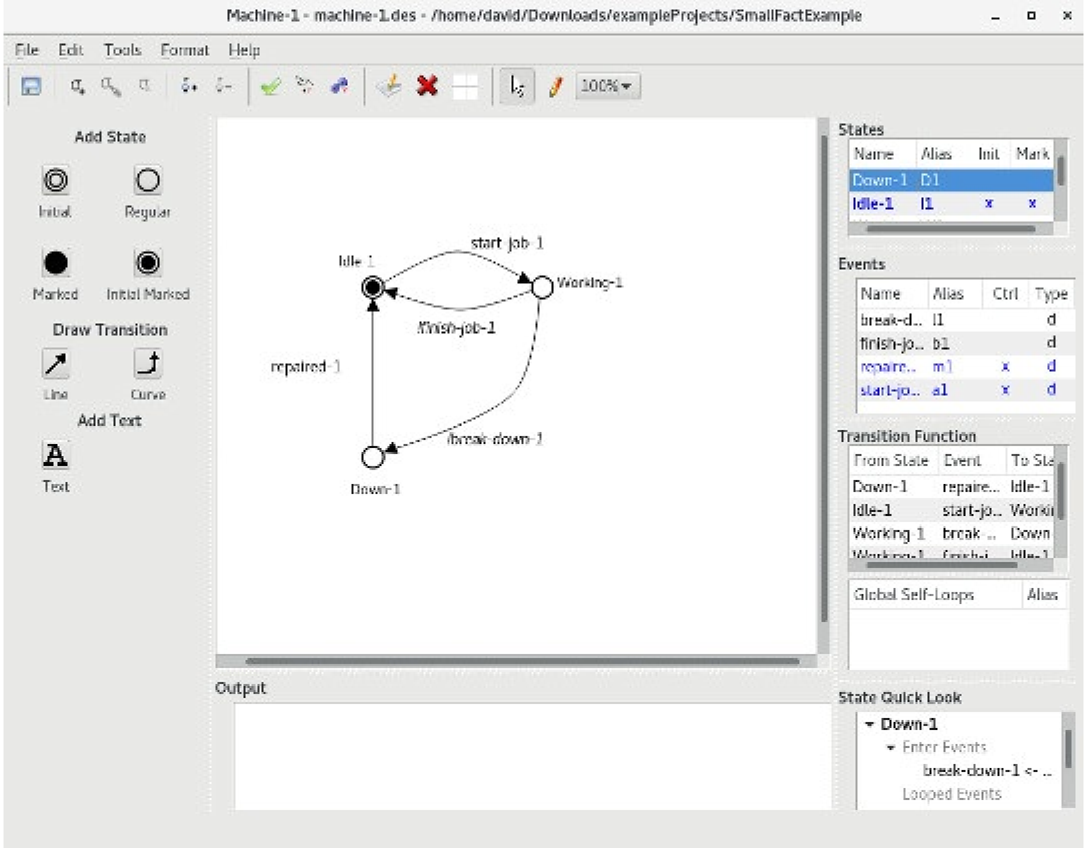
The Grid is a background image that can be turned on the help with the alignment of DES components. To turn on the Grid, open the **Format** menu and select **Show Grid** or click on the **Grid** button on the toolbar .
Please Note: The Grid will not be included in a DES Image export.
Below is a DES image with the Grid enabled.



GRAPH Editor: Zoom

To change the Zoom of a DES workspace, select the percentage zoom you wish from the **Zoom** drop-down box on the toolbar. 100%
Below is a DES at 100% zoom and 150% zoom.

Please Note: Zoom only affects the display of the DES. It does not affect exporting the DES to an image file.

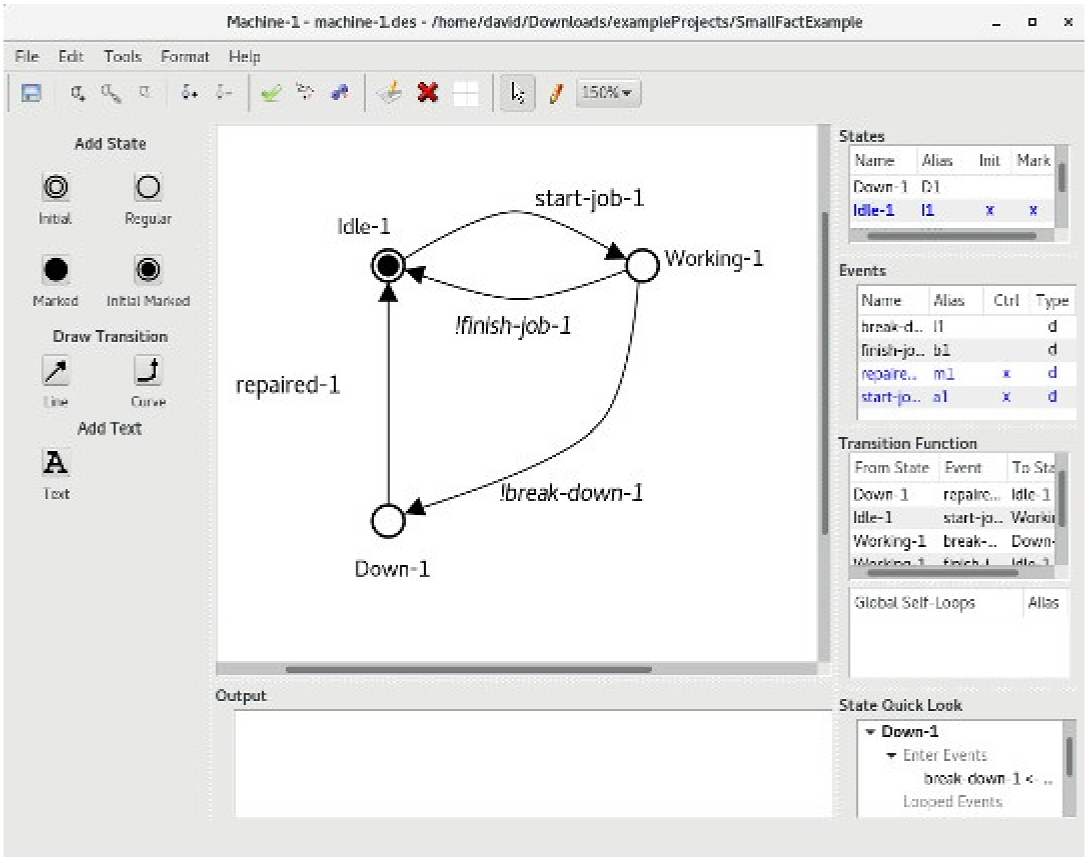


Name	Alias	Init	Mark
Down-1	D1		
Idle-1	I1	x	x

Name	Alias	Ctrl	Type
break-d...	l1		d
finish-jo...	b1		d
repaire...	m1	x	d
start-jo...	a1	x	d

From State	Event	To Sta
Down-1	repaire...	Idle-1
Idle-1	start-jo...	Worki
Working-1	break-...	Down
Working-1	finish-j...	Idle-1

Global Self-Loops	Alias



States

Name	Alias	Init	Mark
Down-1	D1		
Idle-1	I1	x	x

Events

Name	Alias	Ctrl	Type
break-d...	l1		d
finish-jo...	b1		d
repaire...	m1	x	d
start-jo...	a1	x	d

Transition Function

From State	Event	To Sta
Down-1	repaire...	Idle-1
Idle-1	start-jo...	Worki
Working-1	break-...	Down-
Working-1	finish-J	Idle-1

Global Self-Loops	Alias

DES Tools

Verify DES Integrity

From a DES Editor window, open the **Tools** menu and select **Verify DES Integrity** or click on the **Verify DES Integrity** button ✓ on the toolbar. This will verify the DES is Consistent.

Check Reachability

From a DES Editor window, open the **Tools** menu and select **Check Reachability** or click on the **Check Reachability** button on the toolbar. This will check that every state is reachable from the initial state.

Check Nonblocking

From a DES Editor window, open the **Tools** menu and select **Check Nonblocking** or click on the **Check Nonblocking** button on the toolbar. This will check that every reachable state is coreachable (i.e. a path exists from the state to a marked state).

DES Consistency

A DES is consistent if it meets the following criteria:

1. The name of the DES and the state and event labels it contains do not contain invalid characters.
2. Each state in the DES has a unique name within the DES statespace.
3. The DES must have one and only one initial state.
4. The DES must have at least one marked state.
5. The DES must have at least one event.
6. Each event in DES has a unique name (label) within the DES event space.
7. The DES must be deterministic. This means a single initial state and no two transitions leaving the same state with the same event label.
8. The DES must be reachable. This means that there must be a path from the initial state to every state in the DES.
9. The DES must have at least one transition.
10. If the DES is of type Interface, each event in DES must be either request, answer or low data.
11. Only template DES can contain name, state and event labels that contain '%' characters.
12. Every variable that appears in a state or event label inside a template DES, must also appear in the name of the DES.

Save a DES

From a Tabular Editor or Graphical Editor window of the DES you wish to save, open the **DES** menu then click on **Save** or **Save Duplicate** or click on the **Save** button 💾 on the toolbar. This will open up a file browser (for **Save**, only for first usage), where you can choose the location and file name of your DES.
Please Note:
The DES file name is used for identifying your DES file on disk and the DES name is used for identifying your DES within DESpot.

Save

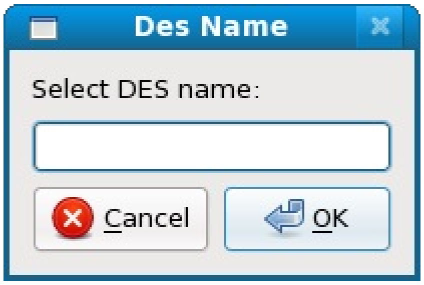
Use the **Save** option when you have made changes to a previously saved DES and you wish to save the changes using the same DES file name.

Save Duplicate

Use the **Save Duplicate** option when you wish to save the DES under a new file name. This will keep the original DES file unmodified and save a new copy of the DES under the new file name. After choosing a filename, a dialogue will pop up allowing you to change the DES name.

Set DES Name

From a Tabular Editor window of the DES you wish to change the name, open the **DES** menu. From a Graphical Editor window open the **File** menu. Next click **Set Name...** This will open the following dialog box.



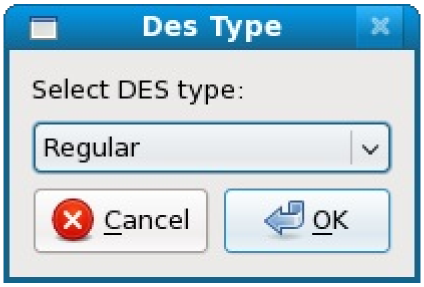
Select DES name:

Enter the new name for the DES and click **OK**.
Use this option to change the name of your DES.

Please Note: The DES name is used for identifying your DES in DESpot and is separate from the DES file name. The file name identifies your DES on disk.

Set DES Type

From a Tabular Editor window of the DES you wish to change the type, open the **DES** menu. From a Graphical Editor window open the **File** menu. Next click **Set Type...** This will open the following dialog box.



Select DES type: Regular

Enter the new type for the DES and click **OK**.
Use this option to change your DES type.

Print DES

From a Tabular Editor window of the DES you wish to print, open the **DES** menu. From a Graphical Editor window open the **File** menu. Next click **Print to file...** This will translate your DES into a text file (*.txt) and save it in the file path of your choice.

Simulator

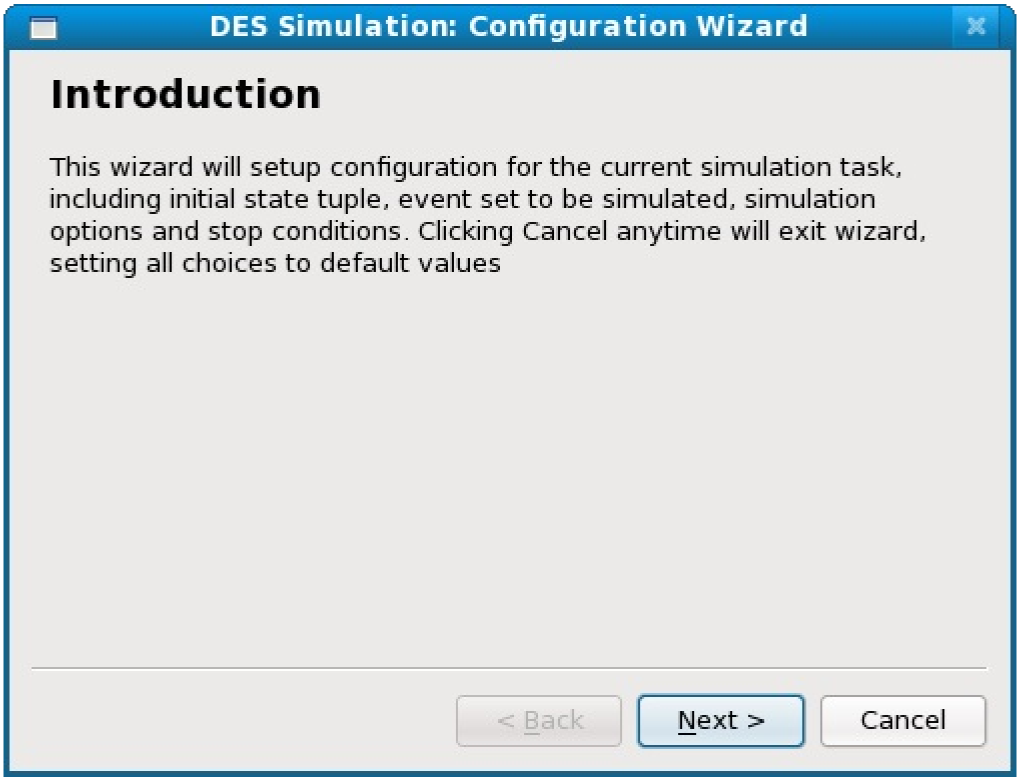
For help on opening the Simulator, please refer to the **Tools** page of the Flat or HISC project editor windows.

The DES Simulator is a great addition to DESpot because it allows the user to simulate transitions occurring in order to see how a project of DES will react and function. It allows the user to troubleshoot errors in a project by simulating a specific sequence or group of events. Furthermore, a project's robustness can be checked by having a simulation run for a long period of time.

When the simulator is started, the Simulator Configuration Wizard will open to help you configure the various properties needed to run a simulation. Once you have completed the wizard, you are now ready to start simulating your project.

Configuration Wizard

When the Simulator is started, the **Configuration Wizard** also opens. This wizard is used to configure the various properties needed to run a simulation.


DES Simulation: Configuration Wizard

Introduction

This wizard will setup configuration for the current simulation task, including initial state tuple, event set to be simulated, simulation options and stop conditions. Clicking Cancel anytime will exit wizard, setting all choices to default values

The first property to choose is the **Simulation Type** and **Event Set Type**. There are three different simulation types. The first is to simulate a true hierarchical HISC model. The second is to simulate a hierarchical project as if it were a flat model (if you have not already verified that an HISC project is LD Interface Consistent or if it is a multi-level project, only this option will be available). The last simulation type is for simulating a flat model. The Event set type can either be a specific set of events, or a predetermined sequence of events. You will almost always want to choose **Simulate an Event Set**.

The next step is to choose a starting state tuple. The default option is the initial state of all the DES. Double click on a DES' state to change that DES to a different state.

If you chose to simulate an event set, after choosing your initial state tuple, you will need to choose your event set. The default event set is all available events (recommended). You can click on **Edit Event Set** to select fewer events. When editing the event set you can add and remove events from the event set on the right using the respective buttons. The event list on the left is all the available events.

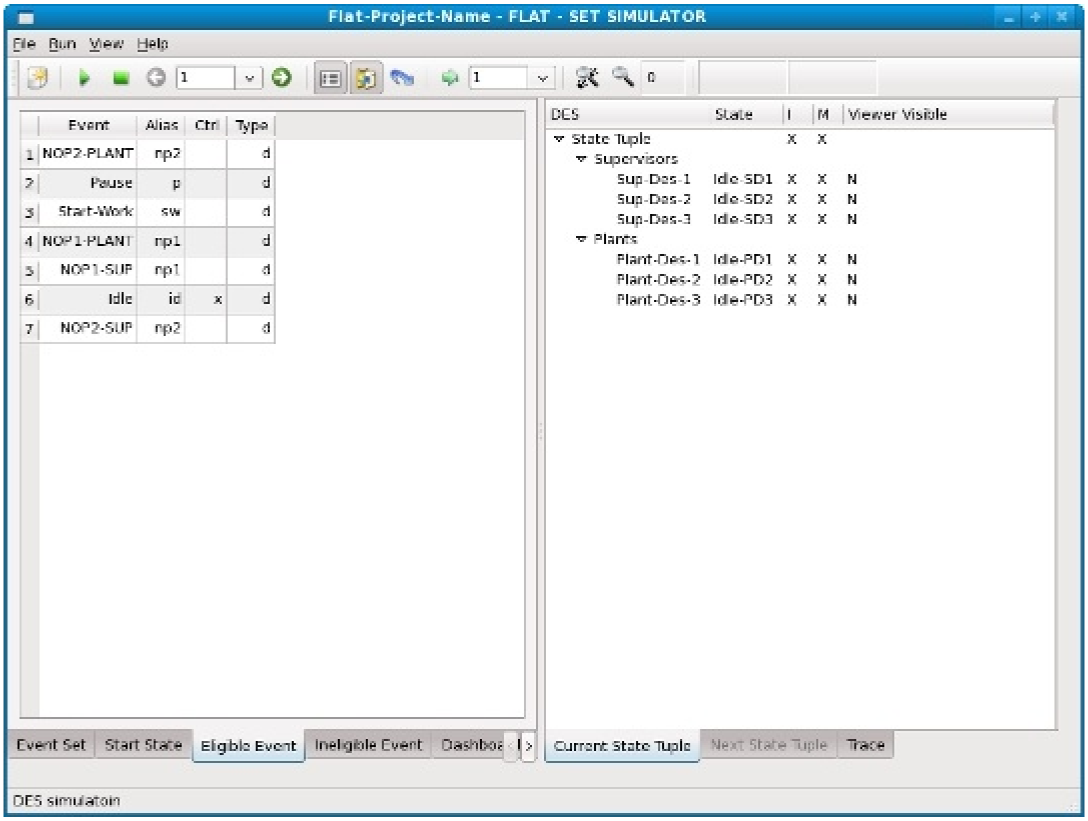
If you chose to simulate an event sequence, after choosing your initial state tuple, you will need to choose your event sequence. The default sequence is all the events in order. Click on **Edit Event Sequence** to change this. Use the buttons to add and remove events from the event sequence on the right. You can also move and event to a different position in the sequence by selecting it and clicking the **Move Up** and **Move Down** buttons. The event list on the left is all the available events.

Lastly choose a stop condition. This is a condition that will stop the simulation. By default, there is no stop condition. You can have the simulation stop after a specified number of steps, when a marked state is reached, or when an initial state is reached.

If you wish to access the wizard again later you can do so by opening it from the **File** menu in the Simulator window.

SIM: Main Window

When you complete the Configuration Wizard, you will be redirected to the Simulator window which will look something like this.



	Event	Alias	Ctrl	Type
1	NOP2-PLANT	np2		d
2	Pause	p		d
3	Start-Work	sw		d
4	NOP1-PLANT	np1		d
5	NOP1-SUP	np1		d
6	Idle	id	x	d
7	NOP2-SUP	np2		d

DES	State	I	M	Viewer Visible
State Tuple		X	X	
Supervisors				
Sup-Des-1	Idle-SD1	X	X	N
Sup-Des-2	Idle-SD2	X	X	N
Sup-Des-3	Idle-SD3	X	X	N
Plants				
Plant-Des-1	Idle-PD1	X	X	N
Plant-Des-2	Idle-PD2	X	X	N
Plant-Des-3	Idle-PD3	X	X	N

The Simulator window contains a number of tools as well as a multitude of information you can use to help simulate your project. The four main areas of the window are the Menu, Toolbar, Events pane, and DES pane.

The Menu has three directories, File, Run, and View. The File menu is where you can Start a new simulation, reopen the Configuration Wizard, Open an existing simulation, Save your current simulation, or close the simulator. The Run menu is where you can Start and Stop your simulation, step Forward and Backward through the simulation, and toggle the Trace, Fast Mode, and History. The View menu has an option to toggle between the Running view and the History view as well as open the Trace Viewer and Graphical Viewer.

The toolbar is divided into six sections. The first section contains a shortcut button to start a new simulation. The second and third sections contain everything found in the Run Menu. The drop down box in the second section allows you to select how many events will take place consecutively when you step forward or backward in your simulation.

The next section contains buttons to toggle showing History and Trace, or not showing them to speed up the simulation. There is also a button to turn Fast Mode simulation on or off (default is off).

The next section contains a **Go To** button as well as a drop down box. Use these to go to a specific time in the history of the simulation. For example, if you have simulated 1436 events (i.e your current step is 1436), you could **Go To** any place in time from step 1-1435. Please note that jumping to an earlier step will discard all later transitions.

The fifth section contains shortcut buttons for the Trace and Graphical viewers as well as the very important **Current Step**. The Current Step is the amount of events simulated consecutively from your start state. The last section contains the previous and next events. The previous event is the event that brought the simulation from the last step to the current step. The next event is the event that will bring the simulation to the next step. The next event is only "known" when in History mode.

The Events pane contains several tabs which are used to display different information about the events in the simulation. This includes the Dashboard tab. Please see the Event Pane page for more information.

The DES pane has two tabs to show current and next state tuples and a third tab to display trace information. Please see the States Pane page for more information.

SIM: Starting a New Simulation

To start a new simulation, click on **System Simulation** from a project window. From a simulation window, open the **File** menu and click **New** or click on the **New** button on the toolbar. This will prompt the option to save the current simulation if you have made changes to it.

SIM: Opening an Existing Simulation

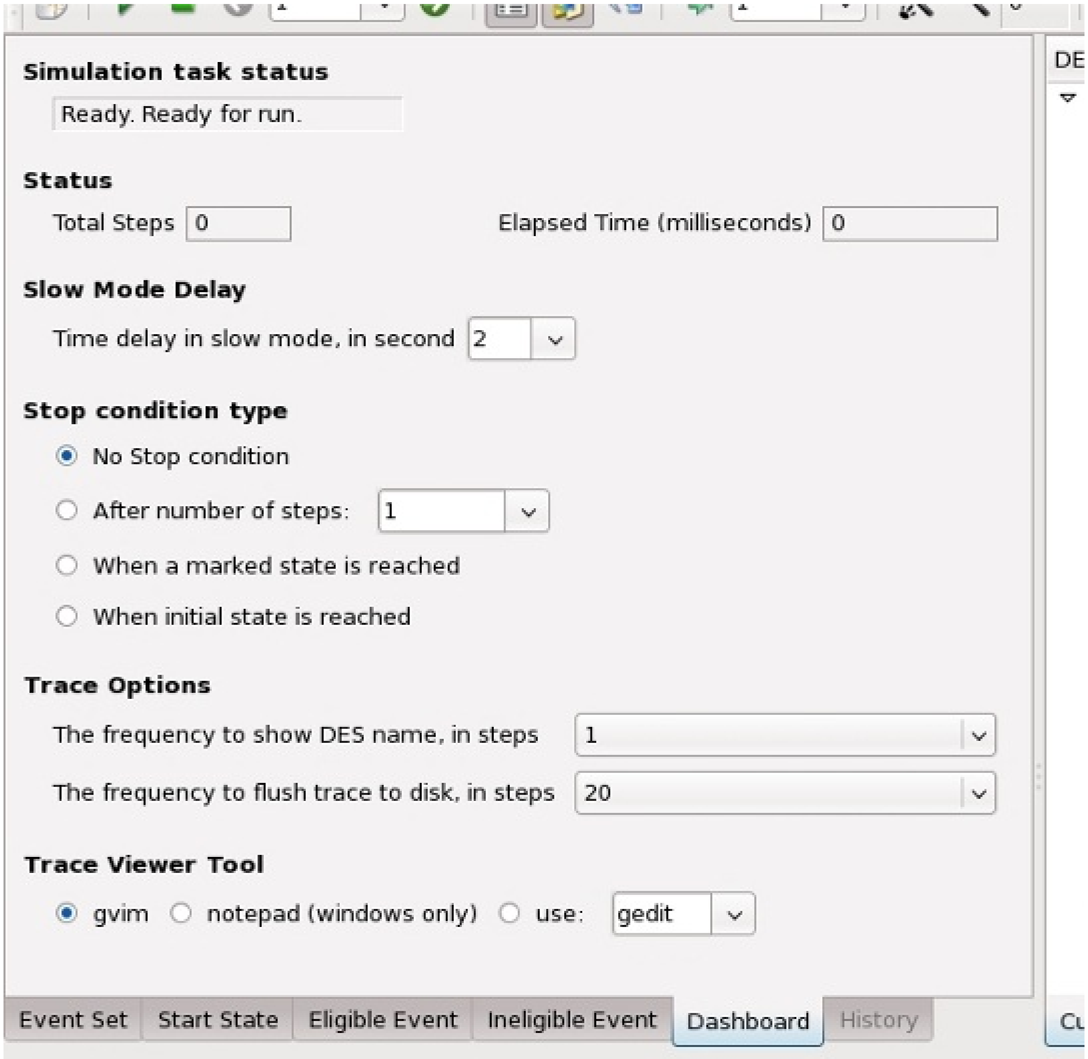
To open a simulation you have previously saved (*.sim extension), from a simulation window, open the **File** menu and click **Open**. This will prompt the option to save the current simulation if you have made changes to it.

SIM: Saving a Simulation

To save your simulation, open the **File** menu and click on **Save** or **Save As...**. The **Save** option saves your simulation under the file name you have already created. If you are saving your simulation for the first time, a file dialog will appear where you can specify the file name and file path. If you would like to save your simulation under a new file name or file path, use the **Save as** option. The simulation is stored as a *.sim file. This file includes the start state used and the sequence of events generated.

SIM: Events Pane

The events pane contains six tabs. The first two tabs contain static information. The next three contain information related to events when in Running View and the last tab is used when in History View.


Simulation task status

Ready. Ready for run.

Status

Total Steps: 0 — Elapsed Time (milliseconds): 0

Slow Mode Delay

Time delay in slow mode, in second: 2

Stop condition type

- ● No Stop condition
- ○ After number of steps: 1
- ○ When a marked state is reached
- ○ When initial state is reached

Trace Options

The frequency to show DES name, in steps: 1

The frequency to flush trace to disk, in steps: 20

Trace Viewer Tool

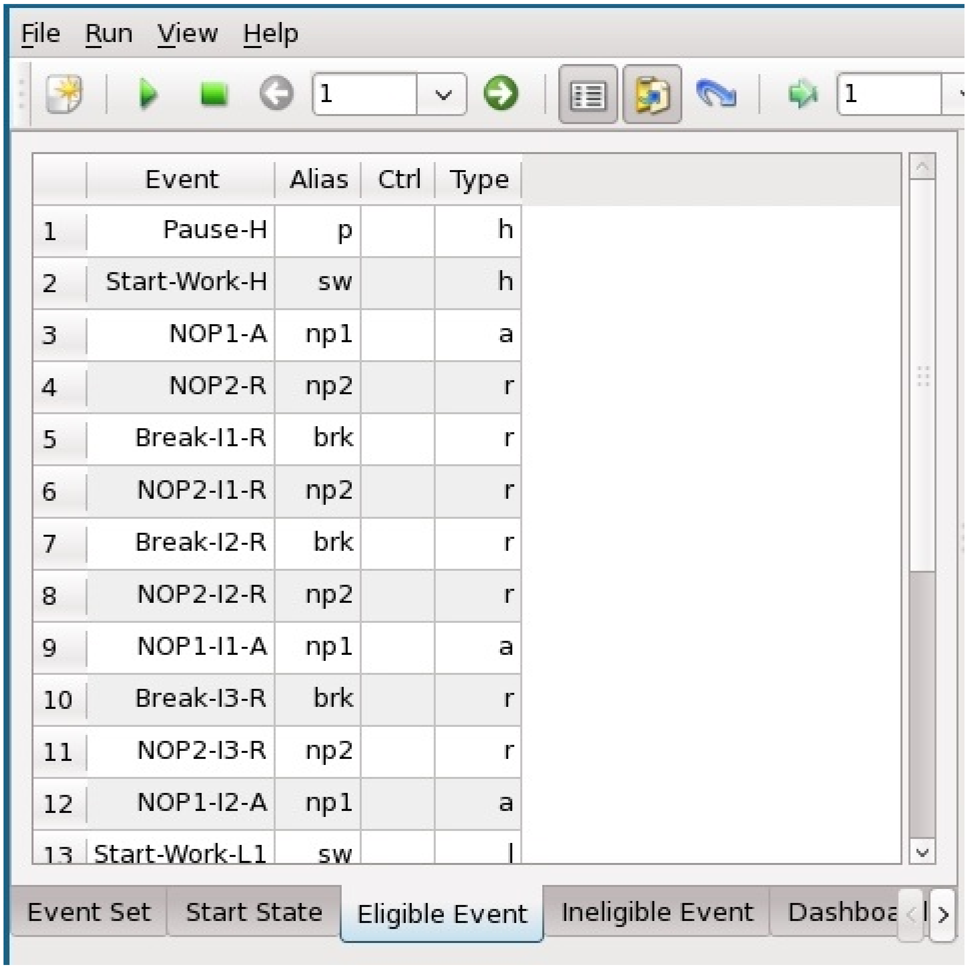
● gvim ○ notepad (windows only) ○ use: gedit

The first tab displays the event set or sequence (depending on what you have chosen in the wizard). If you are using an event set, all the events in the set are displayed. If you are simulating an event sequence, each event is displayed in order of the sequence you have chosen. Two columns display the name and alias of each event, the second last column indicates each event's controllable property and the last column indicated the type of event. The event types are displayed in shorthand to save space. High-Level and Low-Level events are represented by the letters 'h' and 'l' respectively. Request, Answer and low data events are represented by 'r', 'a' and 'ld' respectively.

The second tab displays your chosen start state. This tab lists all the DES in your simulation, the state at which they start the simulation, and whether that state is Initial or Marked. The DES are displayed in a hierarchical form for easy reference.

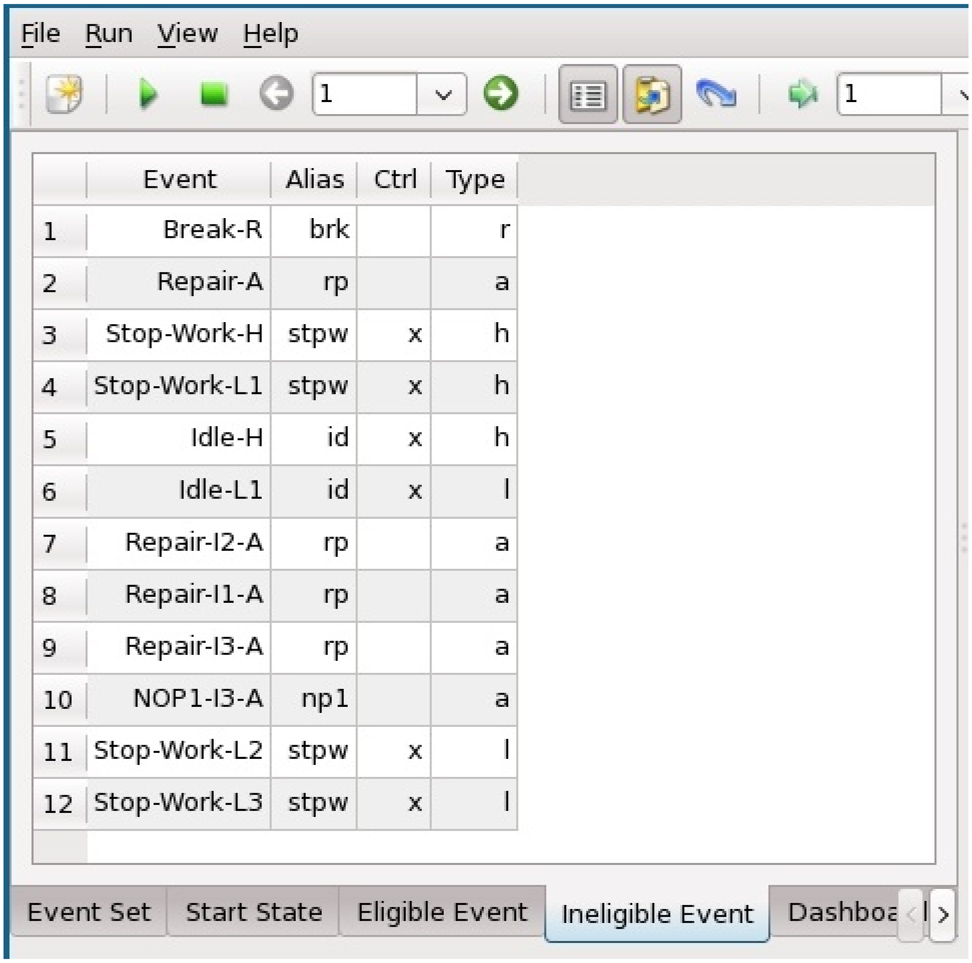
Running View

When Running View is enabled, the next three tabs are available and display information. To enable Running View, open the **View** menu and select **Running View**. The first of the three tabs displays a list of **Eligible Events**. These are the events that can take place given the state of the simulation. If you click on one of the eligible events, you can see the result of that event happening by looking in the Next State Tuple tab of the States Pane. Note that if you click on an event in the eligible tab, then click on the Step Forward button, then this transition will be simulated. Clicking on Step Forward with no event selected will cause a random eligible event to be simulated.



	Event	Alias	Ctrl	Type
1	Pause-H	p		h
2	Start-Work-H	sw		h
3	NOP1-A	np1		a
4	NOP2-R	np2		r
5	Break-I1-R	brk		r
6	NOP2-I1-R	np2		r
7	Break-I2-R	brk		r
8	NOP2-I2-R	np2		r
9	NOP1-I1-A	np1		a
10	Break-I3-R	brk		r
11	NOP2-I3-R	np2		r
12	NOP1-I2-A	np1		a
13	Start-Work-L1	sw		l

The next tab displays all the remaining events or the **Ineligible Events**. Clicking on an event in the ineligible tab will cause all DES in the "Current State Tuple" tab that are blocking the event, to be highlighted in **Red**.

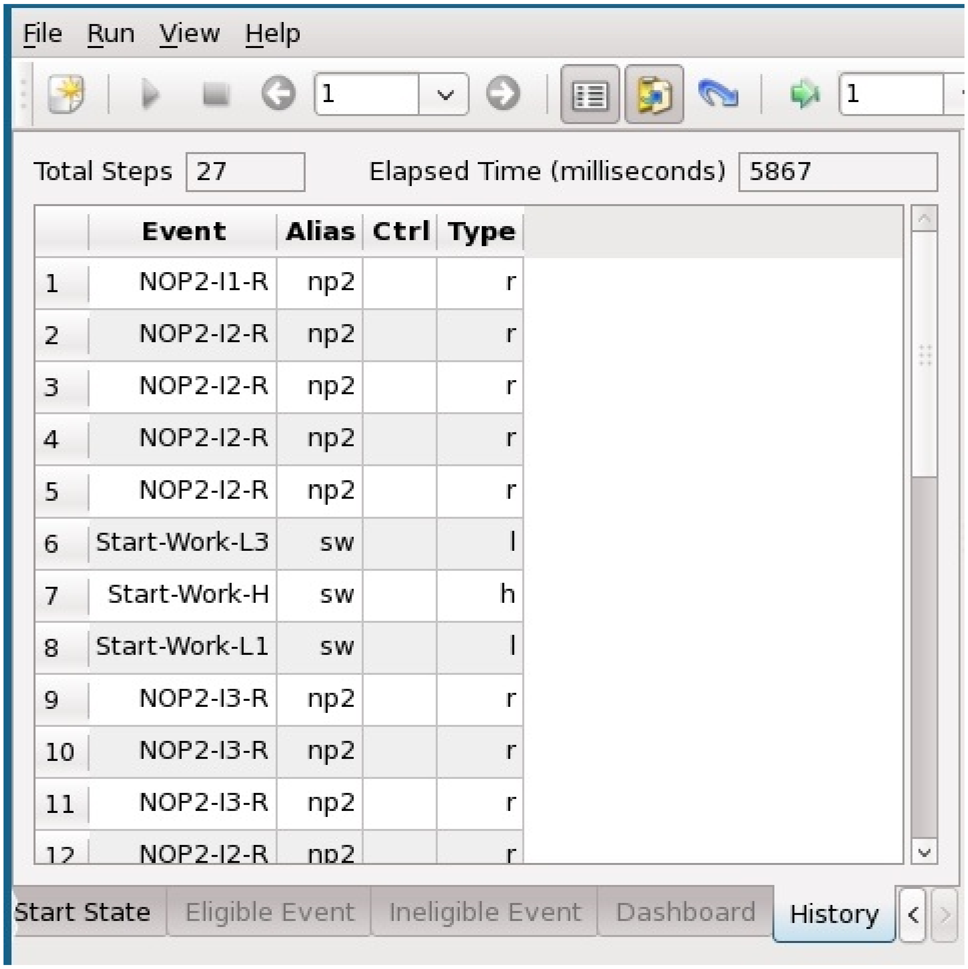


	Event	Alias	Ctrl	Type
1	Break-R	brk		r
2	Repair-A	rp		a
3	Stop-Work-H	stpw	x	h
4	Stop-Work-L1	stpw	x	h
5	Idle-H	id	x	h
6	Idle-L1	id	x	l
7	Repair-I2-A	rp		a
8	Repair-I1-A	rp		a
9	Repair-I3-A	rp		a
10	NOP1-I3-A	np1		a
11	Stop-Work-L2	stpw	x	l
12	Stop-Work-L3	stpw	x	l

The last tab is the **Dashboard**. This tab gives you more options for the simulation. It displays the current task status as well as simulation status. The **Slow Mode** option is the time delay used when running a simulation. Set it longer or shorter to give you more or less time to watch the simulation taking place. The **Stop Condition** can also be changed in this tab. The **Frequency to show DES names** is the amount of steps that will occur before the DES names will be displayed in the Trace. The **Frequency to flush trace to disk** is the amounts of steps that will occur before the trace is cleared and the trace is written to disk. For more information on the Trace please visit the States Pane page. The **Trace Viewer Tool** is the external program used when opening the Trace Viewer.

History View

When History View is enabled, the History tab becomes available and the eligibility of events tabs as well as the dashboard are disabled.



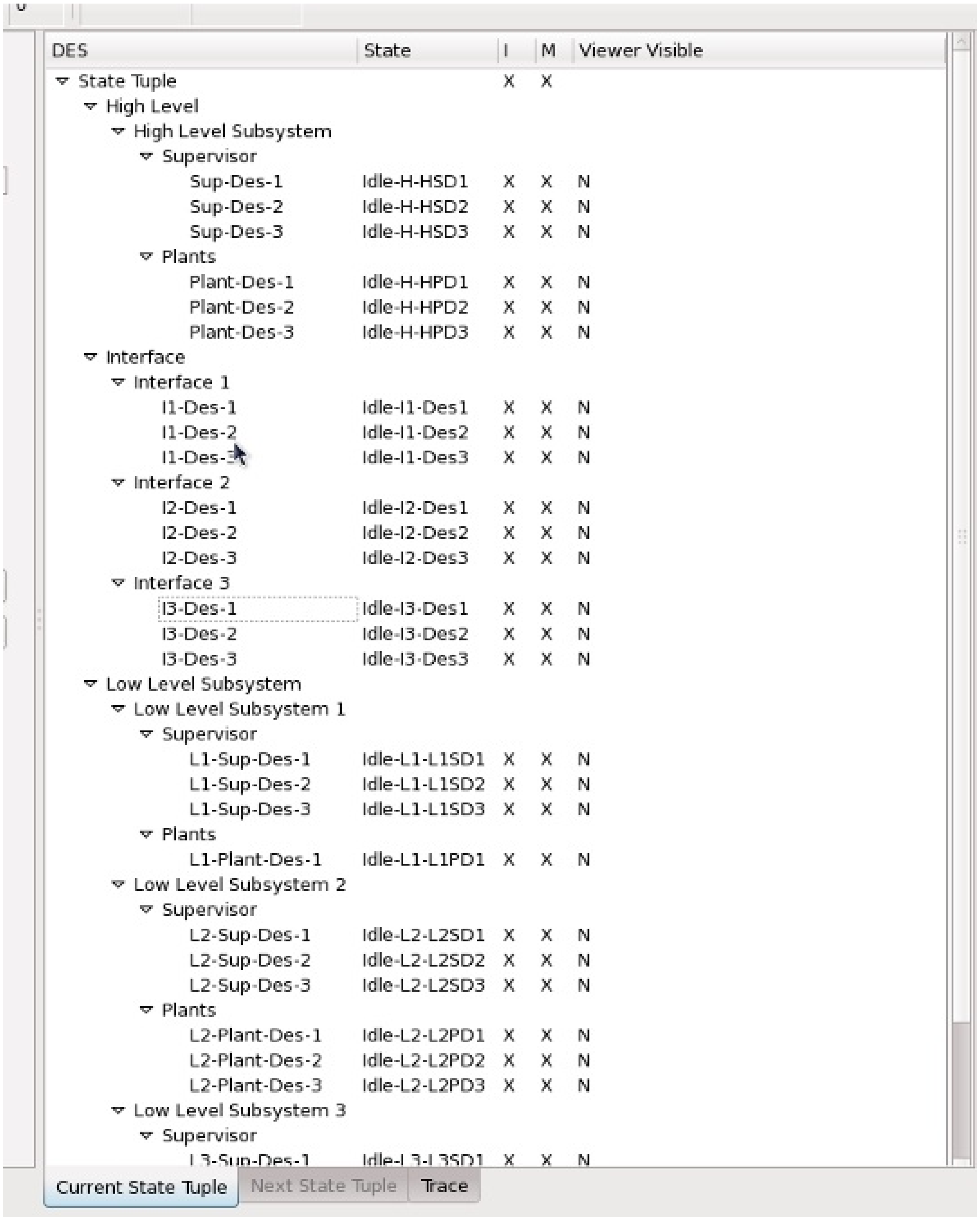
Total Steps 27   Elapsed Time (milliseconds) 5867

	Event	Alias	Ctrl	Type
1	NOP2-I1-R	np2		r
2	NOP2-I2-R	np2		r
3	NOP2-I2-R	np2		r
4	NOP2-I2-R	np2		r
5	NOP2-I2-R	np2		r
6	Start-Work-L3	sw		l
7	Start-Work-H	sw		h
8	Start-Work-L1	sw		l
9	NOP2-I3-R	np2		r
10	NOP2-I3-R	np2		r
11	NOP2-I3-R	np2		r
12	NOP2-I2-R	np2		r

To enable History View, open the **View menu** and select **History View**. The History tab shows the events that have taken place from the start of the simulation in sequential order. It shows each event's name & alias and the controllable and event type properties. At the top of the pane, the total step count is shown. This is the number of events that have occurred since the simulation started (without backtracking). It also displays the total time elapsed in milliseconds. When you are in History View, you cannot simulate events, but you can go back to a particular step in time and follow the events and the DES as they change states. Clicking on an event in the History tab causes the "Current State Tuple" and "Next State Tuple" tab to update to show the state the project is at before the event occurs, and the state reached when the event occurs, respectively. **Please Note:** History View is unavailable when the History Mode is disabled.

SIM: States Pane

The States Pane contains three tabs.



DES	State	I	M	Viewer Visible
State Tuple		X	X	
High Level				
High Level Subsystem				
Supervisor				
Sup-Des-1	Idle-H-HSD1	X	X	N
Sup-Des-2	Idle-H-HSD2	X	X	N
Sup-Des-3	Idle-H-HSD3	X	X	N
Plants				
Plant-Des-1	Idle-H-HPD1	X	X	N
Plant-Des-2	Idle-H-HPD2	X	X	N
Plant-Des-3	Idle-H-HPD3	X	X	N
Interface				
Interface 1				
I1-Des-1	Idle-I1-Des1	X	X	N
I1-Des-2	Idle-I1-Des2	X	X	N
I1-Des-3	Idle-I1-Des3	X	X	N
Interface 2				
I2-Des-1	Idle-I2-Des1	X	X	N
I2-Des-2	Idle-I2-Des2	X	X	N
I2-Des-3	Idle-I2-Des3	X	X	N
Interface 3				
I3-Des-1	Idle-I3-Des1	X	X	N
I3-Des-2	Idle-I3-Des2	X	X	N
I3-Des-3	Idle-I3-Des3	X	X	N
Low Level Subsystem				
Low Level Subsystem 1				
Supervisor				
L1-Sup-Des-1	Idle-L1-L1SD1	X	X	N
L1-Sup-Des-2	Idle-L1-L1SD2	X	X	N
L1-Sup-Des-3	Idle-L1-L1SD3	X	X	N
Plants				
L1-Plant-Des-1	Idle-L1-L1PD1	X	X	N
Low Level Subsystem 2				
Supervisor				
L2-Sup-Des-1	Idle-L2-L2SD1	X	X	N
L2-Sup-Des-2	Idle-L2-L2SD2	X	X	N
L2-Sup-Des-3	Idle-L2-L2SD3	X	X	N
Plants				
L2-Plant-Des-1	Idle-L2-L2PD1	X	X	N
L2-Plant-Des-2	Idle-L2-L2PD2	X	X	N
L2-Plant-Des-3	Idle-L2-L2PD3	X	X	N
Low Level Subsystem 3				
Supervisor				
L3-Sup-Des-1	Idle-L3-L3SD1	X	X	N

Current State Tuple

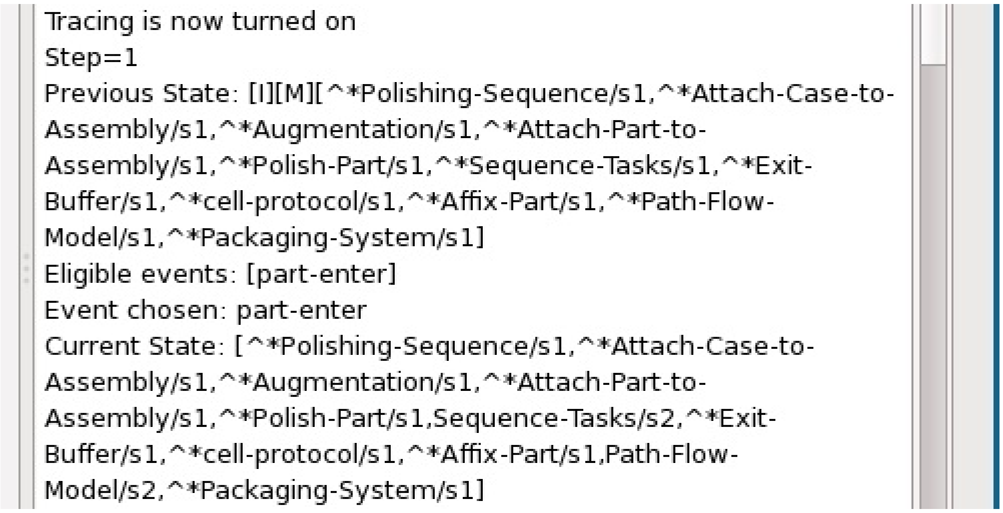
This tab displays all the DES that are being simulated and what their current states are, that is, the current states at the current step. They are listed in a hierarchical form for easier viewing. The DES names are listed followed by their current state. The next two columns indicate whether the current state is an Initial state and whether it is a marked state. The last column indicates if the DES Viewer is open or not for that DES. Double-clicking on a DES will open a Graphical Viewer for that DES, if the DES contains graphical information (i.e. The DES has been edited in the Graphical DES Editor). As the simulation proceeds, the viewer will update.

Next State Tuple

This tab is enabled when in History View or when an event is selected in the Eligible Events tab. When stepping back through a simulation, the Next State Tuple displays all the DES and their states they will change to when the next event occurs. This is only available in History View because the event has already taken place so we know what the Next State Tuple will be. As with the Current State Tuple tab, the Next State Tuple tab displays all the DES in a hierarchical form, followed by their next state, based on the event that has already occurred, and the Initial and Marked properties of that state.

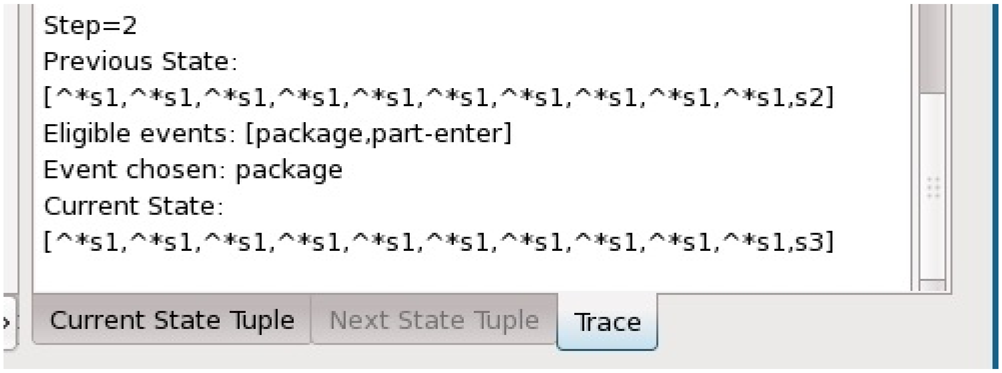
Trace

The Trace is a quick way to follow a simulation in a textual form. It first lists all the information related to the simulation including information such as event set and stop condition. When simulating, it lists first the step number, then the previous state tuple, the eligible events, the event chosen and lastly the current state tuple. When listing the state tuples, you may notice that sometimes it displays all the DES names followed by their states and sometimes it only lists states. This is a shorthand feature of the trace to make it easier to read. To change the **Frequency to show DES names** and for other simulation options please see the Dashboard tab of the Events Pane. For example, if the frequency is set to 1, the DES names will display every time. If it is set to 4, it will only display the DES names every fourth step. You will also notice when running a long simulation that the trace will appear to reset after a number of steps. This happens when the **Frequency to flush trace to disk** count has been met. To save memory and space, the trace is written to a file and cleared from the window at a user defined interval. This frequency can also be changed in the Dashboard tab. To read the state tuples a shorthand is used to save space. Here is a sample trace.



```
Tracing is now turned on
Step=1
Previous State: [I][M][^*Polishing-Sequence/s1,^*Attach-Case-to-
Assembly/s1,^*Augmentation/s1,^*Attach-Part-to-
Assembly/s1,^*Polish-Part/s1,^*Sequence-Tasks/s1,^*Exit-
Buffer/s1,^*cell-protocol/s1,^*Affix-Part/s1,^*Path-Flow-
Model/s1,^*Packaging-System/s1]
Eligible events: [part-enter]
Event chosen: part-enter
Current State: [^*Polishing-Sequence/s1,^*Attach-Case-to-
Assembly/s1,^*Augmentation/s1,^*Attach-Part-to-
Assembly/s1,^*Polish-Part/s1,Sequence-Tasks/s2,^*Exit-
Buffer/s1,^*cell-protocol/s1,^*Affix-Part/s1,Path-Flow-
Model/s2,^*Packaging-System/s1]
```

As you can see this is the first step in the simulation (Step=1). The first thing displayed in the Previous State is **[I][M]**. This indicates all the DES are at **I**nitial and **M**arked states. Next is the DES names. Preceding each DES name there is a **^** and a ***** symbol. The **^** indicates the DES is at an Initial state. The ***** indicates the DES is at a Marked state. Next is the DES name followed by a **/** and the state name. This information is then separated by a comma **,** and repeated for each DES. The next image shows the trace without the DES names.



```
Step=2
Previous State:
[^*s1,^*s1,^*s1,^*s1,^*s1,^*s1,^*s1,^*s1,^*s1,^*s1,s2]
Eligible events: [package,part-enter]
Event chosen: package
Current State:
[^*s1,^*s1,^*s1,^*s1,^*s1,^*s1,^*s1,^*s1,^*s1,^*s1,s3]
```

As you can see, the same information is present except for the DES names. Notice that the **[I]** and **[M]** are not shown. This is because at least one DES is not at an Initial or Marked state. In this case, the last DES. At this state tuple two events are eligible. The event chosen is 'package' which causes the last DES to to change states. You can see this because all the states are the same from the previous state to the current state except for the last DES.

Clicking on the Trace Viewer button brings up a text editor allowing the entire trace to be viewed at once.

SIM: Simulation Modes

There are three different modes you can toggle when running a simulation. By default, the Trace and History are enabled, and Fast Mode is disabled.

Toggle Trace

When Trace is enabled, the Trace output will function as described in the Trace tab of the States Pane. When Trace is disabled, it will not be shown in the Trace tab or in the Trace Viewer.

Toggle Fast Mode

By default, this mode is disabled. This is also known as **Slow Mode**. When simulating several events simultaneously Slow Mode pauses between each event to give the user time to view the changes taking place within the DES. This time delay can be changed in the Dashboard tab. When you enable fast mode, no time delay is used and the simulation runs at full speed.

Toggle History

When History is disabled, you can only run a simulation forward in time and cannot step back. This also disables the History View. Once History is disabled for a simulation, it cannot be re enabled.

SIM: Running a Simulation

Once you have setup your simulation with all the options of your choosing, you are ready to start simulating. There are two ways to run a simulation, automatically or manually. To run a simulation manually, select the **Step Size** from the first drop-down box on the toolbar
9
. Then use the **Step Forward** and **Step Backward** buttons to advance the simulation by the chosen number of events. By default, the next transition to occur is randomly selected from the Eligible Event list. If you wish a specific event to occur next, click on the event in the eligible event list, then click on Step Forward.

To run a simulation automatically, simply push the **Run Simulation** button on the toolbar. The simulation will run fast or slow depending on if Fast Mode is enabled. If it is not enabled, the simulation will step based on the time delay you have selected in the Dashboard. The simulation will run until you reach your stop condition. If you have no stop condition, your simulation will run indefinitely. If you wish to stop your simulation at any point in time, press the **Stop** button on the toolbar. A notification will appear indicating the user has stopped the simulation. All the buttons to run a simulation can also be found in the **Run** menu.

If you have simulated a large number of events but wish to go back to a specific step, it can cause strain to manually click the **Step Backward** button numerous times. This is why there is another drop-down box on the toolbar. If you know the specific step you wish to navigate to, simply enter it in the drop-down box
126
and click the **Go To** button on the toolbar.

Debugging DES in DESpot

DESpot provides the following features to debug errors in discrete event systems.

- Counter Example Simulation
- Counter Example Running Modes
- Debugging in Multiple Sessions
- Error State Simulation

- Non Counter Example Algorithm List

Counter Example Simulation(CES)

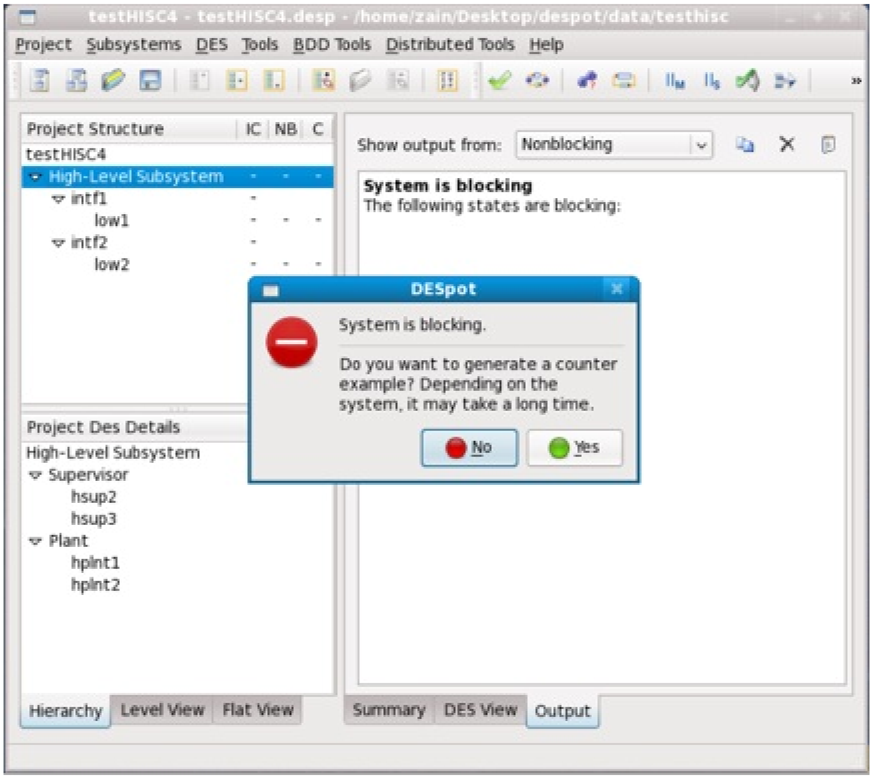
For debugging complicated errors, DESpot provides the feature of generating a counter example. This is a short sequence of events inducing a traversal of the system's synchronous product from the initial state to the error state. This provides a mechanism to pinpoint the state where the error occurred. An incremental approach is adopted wherein the very first error generated is used for debugging. Its removal allows the users to move to the next error, assuming it is still relevant. Based on the project type, choose the appropriate topic for a step-by-step procedure to generate a counter example.

Warning: at this time, counter example generation does not support multi-level HISC projects.

- Flat Counter Example Simulation
- Hierarchical Counter Example Simulation

Flat System CES

1. When an algorithm fails, the counter example dialog box is displayed. Click on the ● Yes button to generate the counter example. If you click on the ● No button, no counter example will be generated, but the error state is still recorded.

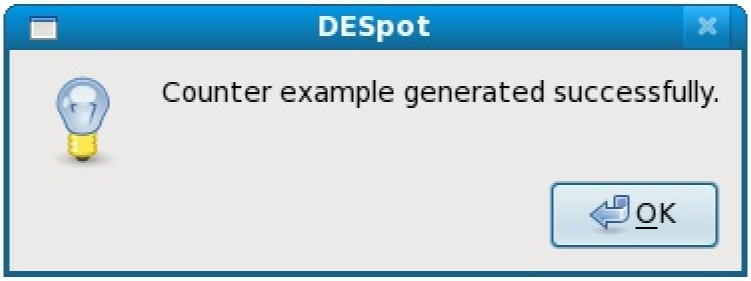

testHISC4 - testHISC4.desp - /home/zain/Desktop/despot/data/testhisc
Project Subsystems DES Tools BDD Tools Distributed Tools Help
Project Structure IC NB C
testHISC4
High-Level Subsystem
intf1
low1
intf2
low2
Project Des Details
High-Level Subsystem
Supervisor
hsup2
hsup3
Plant
hplnt1
hplnt2
Hierarchy Level View Flat View
Show output from: Nonblocking
Summary DES View Output
DESpot
No Yes

System is blocking
The following states are blocking:

System is blocking.

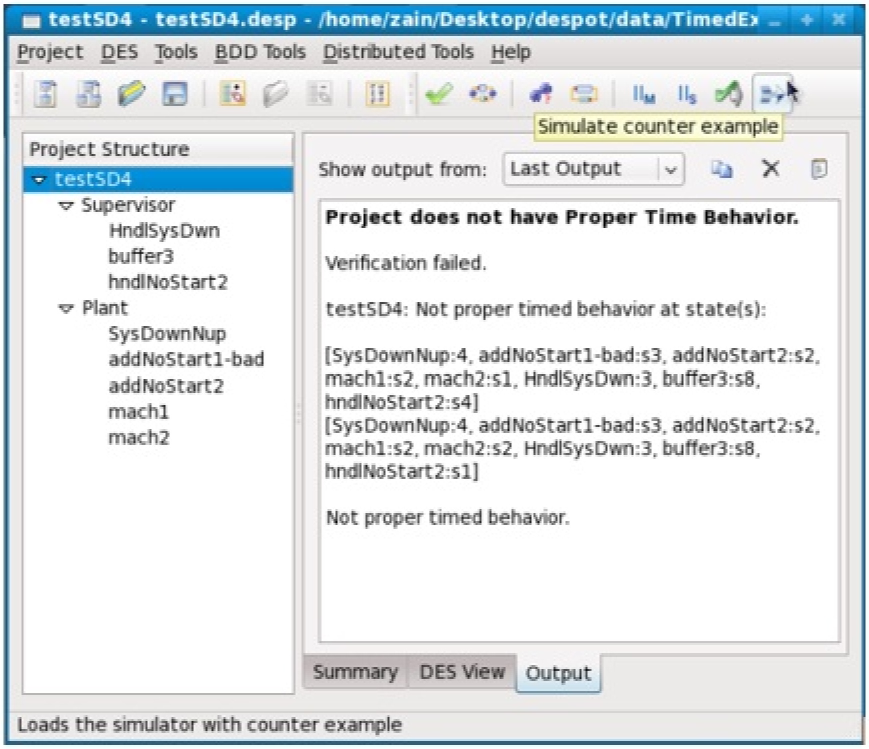
Do you want to generate a counter example? Depending on the system, it may take a long time.

2. The counter example algorithm completion dialog then appears. Click on the OK button.



Counter example generated successfully.

3. Now click on the button. If you generated a counter example, this is loaded into the simulator. Otherwise, the simulator opens with the system started at the recorded error state.


testSD4 - testSD4.desp - /home/zain/Desktop/despot/data/TimedEx
Project DES Tools BDD Tools Distributed Tools Help
Simulate counter example
Project Structure
testSD4
Supervisor
HndlSysDwn
buffer3
hndlNoStart2
Plant
SysDownNup
addNoStart1-bad
addNoStart2
mach1
mach2
Show output from: Last Output
Summary DES View Output
Loads the simulator with counter example

```
Project does not have Proper Time Behavior.

Verification failed.

testSD4: Not proper timed behavior at state(s):

[SysDownNup:4, addNoStart1-bad:s3, addNoStart2:s2,
mach1:s2, mach2:s1, HndlSysDwn:3, buffer3:s8,
hndlNoStart2:s4]
[SysDownNup:4, addNoStart1-bad:s3, addNoStart2:s2,
mach1:s2, mach2:s2, HndlSysDwn:3, buffer3:s8,
hndlNoStart2:s1]

Not proper timed behavior.
```

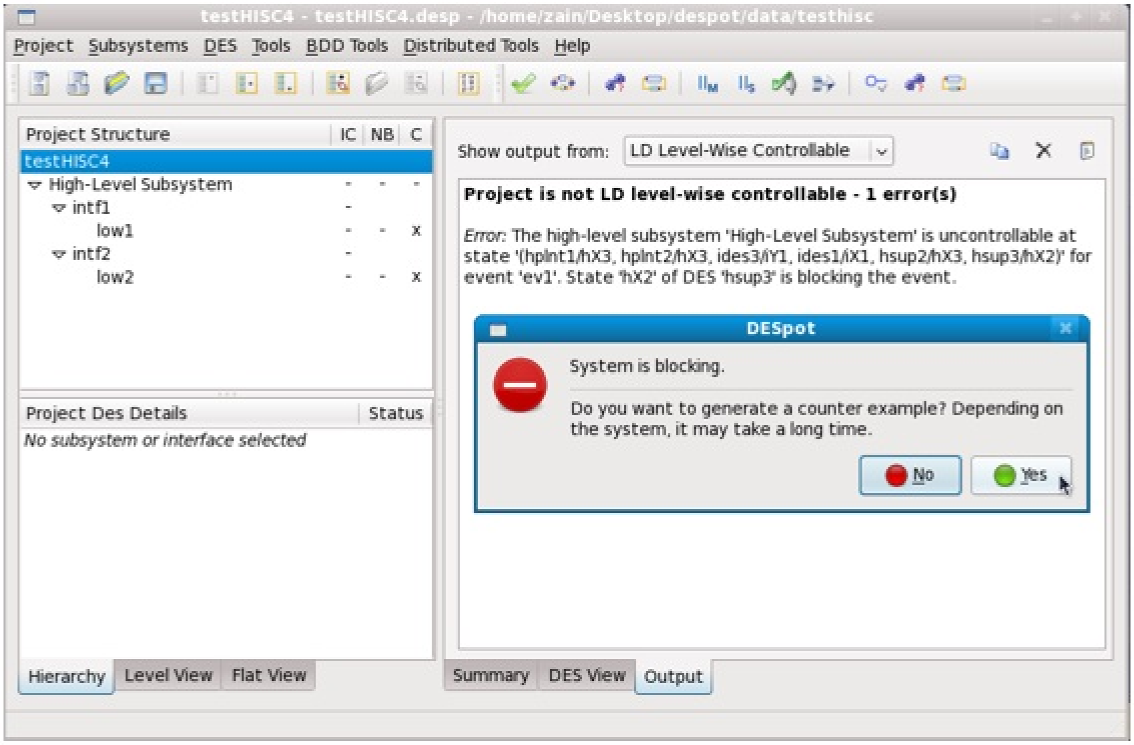
OR

From the menu bar, select **Tools | Debug simulation**

For information on running a counter example simulation, see Running Modes

Hierarchical System CES

1. When an algorithm fails, the counter example dialog box is displayed. Click on the ● Yes button to generate the counter example. If you click on the ● No button, no counter example will be generated, but the error state is still recorded.


testHISC4 - testHISC4.desp - /home/zain/Desktop/despot/data/testhisc
Project Subsystems DES Tools BDD Tools Distributed Tools Help
Project Structure IC NB C
testHISC4
High-Level Subsystem - - -
intf1 -
low1 - - x
intf2 -
low2 - - x
Project Des Details Status
Hierarchy Level View Flat View
Summary DES View Output
No
Yes

Show output from: LD Level-Wise Controllable

Project is not LD level-wise controllable - 1 error(s)

Error: The high-level subsystem 'High-Level Subsystem' is uncontrollable at state '(hpInt1/hX3, hpInt2/hX3, ides3/iY1, ides1/iX1, hsup2/hX3, hsup3/hX2)' for event 'ev1'. State 'hX2' of DES 'hsup3' is blocking the event.

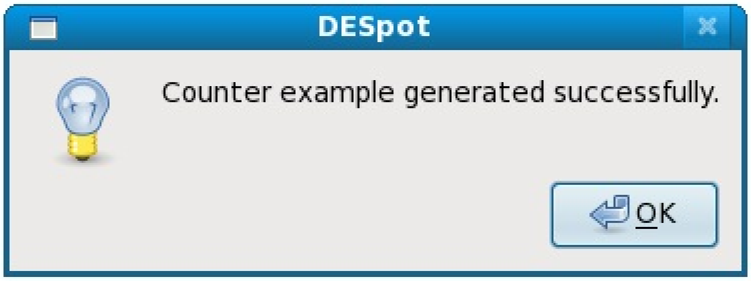
No subsystem or interface selected

DESpot

System is blocking.

Do you want to generate a counter example? Depending on the system, it may take a long time.

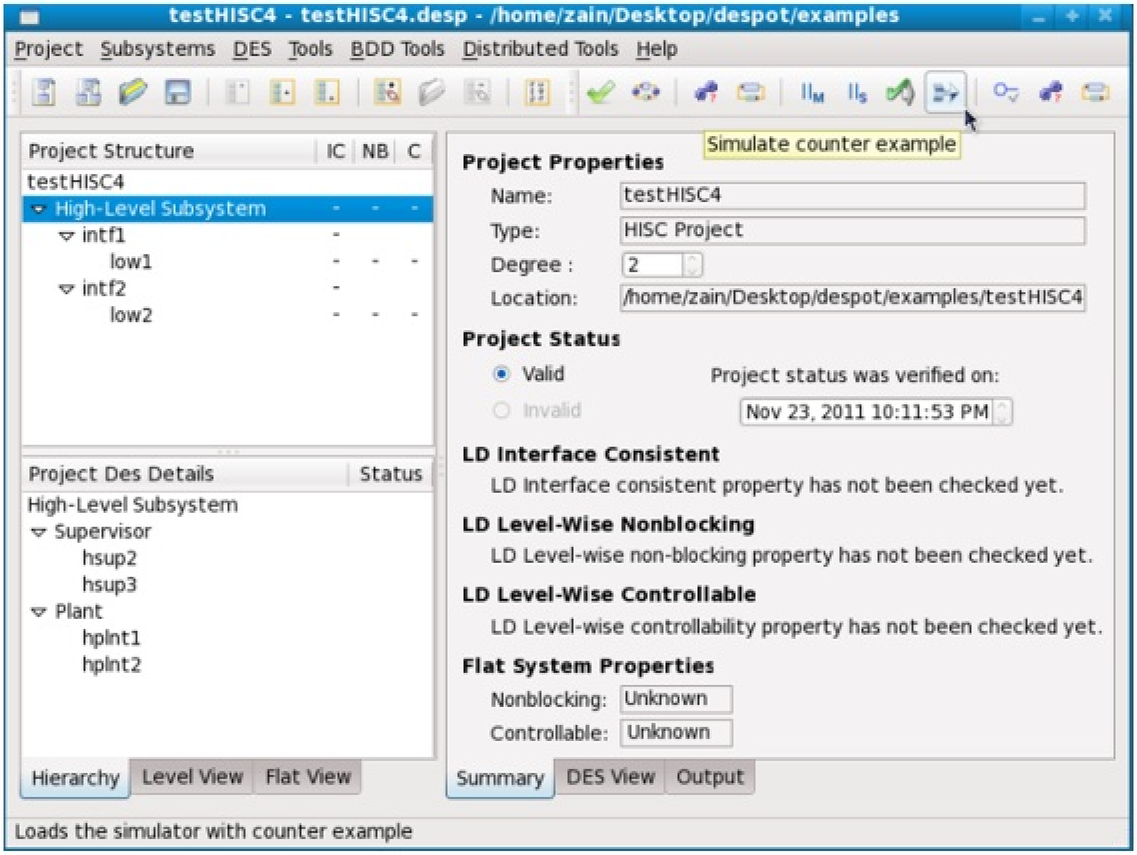
2. The counter example algorithm completion dialog then appears. Click on the OK button.



Counter example generated successfully.

3. Now click on the button. If you generated a counter example, this is loaded into the simulator. Otherwise, the simulator opens with the system started at the recorded error state.

As HISC conditions are per component, so are the error states and counter examples. When you enter the simulator, only the component that failed and its interfaces are loaded, and they are treated as a flat project.



Project Structure	IC	NB	C
testHISC4			
High-Level Subsystem	-	-	-
intf1	-		
low1	-	-	-
intf2	-		
low2	-	-	-

Project Des Details	Status
High-Level Subsystem	
Supervisor	
hsup2	
hsup3	
Plant	
hplnt1	
hplnt2	

Project Properties

Name: testHISC4

Type: HISC Project

Degree : 2

Location: /home/zain/Desktop/despot/examples/testHISC4

Project Status

● Valid

○ Invalid

Project status was verified on: Nov 23, 2011 10:11:53 PM

LD Interface Consistent

LD Interface consistent property has not been checked yet.

LD Level-Wise Nonblocking

LD Level-wise non-blocking property has not been checked yet.

LD Level-Wise Controllable

LD Level-wise controllability property has not been checked yet.

Flat System Properties

Nonblocking: Unknown

Controllable: Unknown

Loads the simulator with counter example

OR

From the menu bar, select **Tools | Debug simulation**

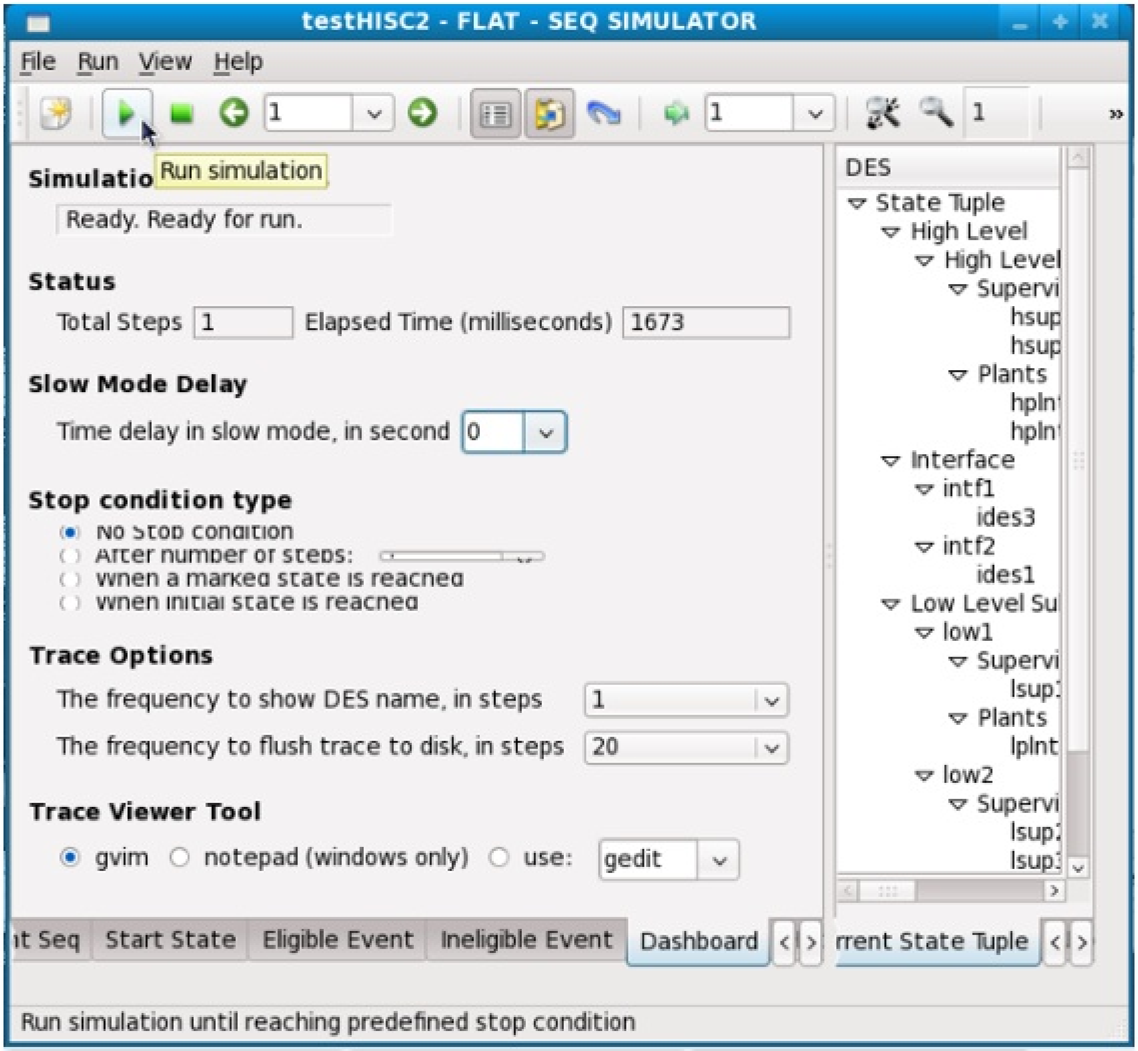
For information on running a counter example simulation, see Running Modes

Counter Example Running Modes

MODE 1: Forward simulation

Press ⬅ button to move back, and ➡ button to move forward.

1. This is the default mode. The simulation is run in slow mode with a 2 second delay between transitions. To run in this mode click on ▶ button.



Simulatio

Ready. Ready for run.

Status

Total Steps 1 Elapsed Time (milliseconds) 1673

Slow Mode Delay

Time delay in slow mode, in second 0

Stop condition type

● No Stop condition
○ After number of steps:
○ When a marked state is reached
○ When initial state is reached

Trace Options

The frequency to show DES name, in steps 1

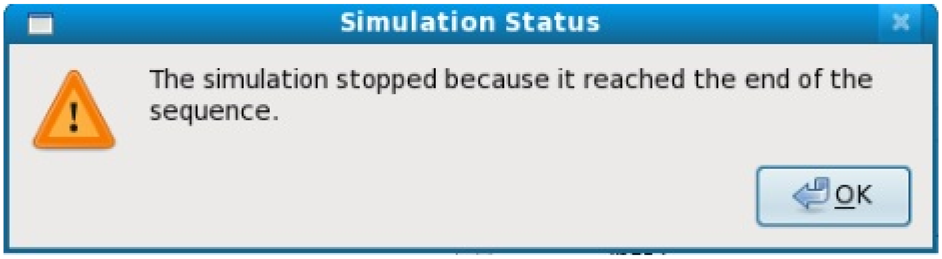
The frequency to flush trace to disk, in steps 20

Trace Viewer Tool

● gvim ○ notepad (windows only) ○ use: gedit

Run simulation until reaching predefined stop condition

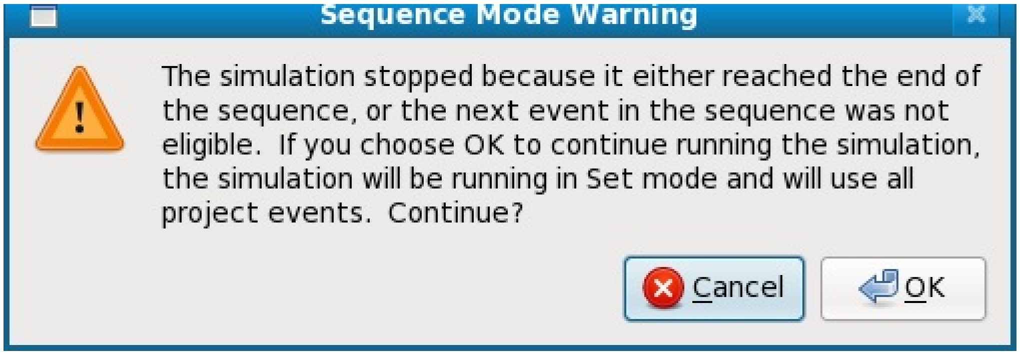
2. When the complete counter example has been run, the sequence completion dialog pops up. At this point the system is at the error state. Pressing the forward button will cause the set mode dialog to pop up and the counter example information will be lost. If the initial state is the error state itself, the sequence completion dialog pops up the moment the simulator is evoked. This is the normal expected behavior as in this case the counter example is the empty sequence.


Simulation Status

The simulation stopped because it reached the end of the sequence.

MODE 2: Backtracking

From the dashboard set the delay to 0 seconds and then click on the ▶ button to quickly move to the error state.
Note: The debugging information for counter example simulation is only retained as long as the user stays within the event sequence mode. The user can freely step forward and backward within this mode. If the user goes beyond the last event in the sequence, the event set mode dialog will pop up. Once the user enters the event set mode, the counter example information will be lost if the user steps backward in the simulation.


Sequence Mode Warning

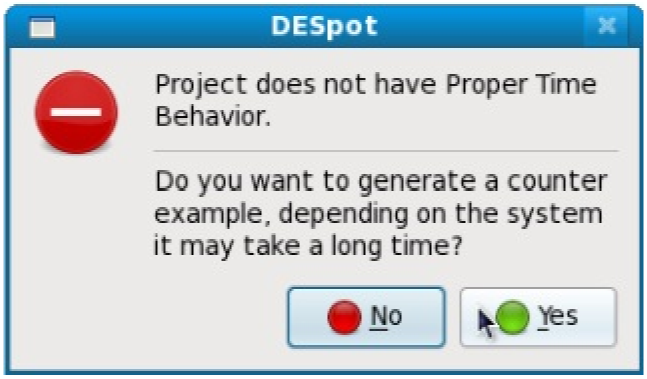
The simulation stopped because it either reached the end of the sequence, or the next event in the sequence was not eligible. If you choose OK to continue running the simulation, the simulation will be running in Set mode and will use all project events. Continue?

Error State Simulation

To facilitate the incremental removal of errors, the first error state the algorithm encounters can be loaded into the simulator. This provides a graphical view of the error, which provides a better understanding of the error. This aids in its removal.

Simulating an error state:

1. For the counter example dialog, press ● No button.



Project does not have Proper Time Behavior.

Do you want to generate a counter example, depending on the system it may take a long time?

2. From the menu bar select **Tools | Debug simulation**

Loading a Counter Example Simulation for a Subsystem

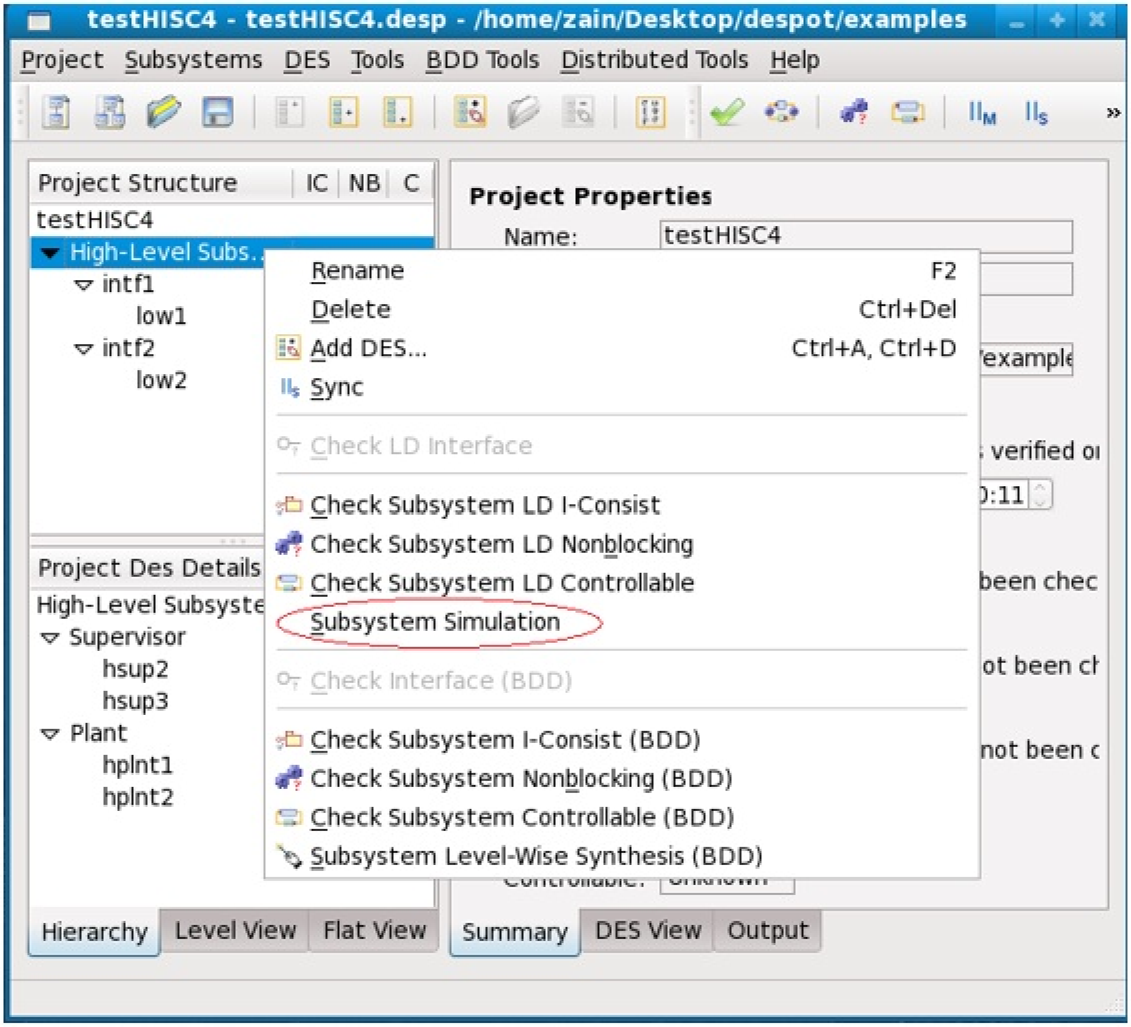
1. From the DESpot main window, open the project for which you want to load the simulation.

2. Select the subsystem for which you want to load the simulation by clicking on it.

3. From the menu bar, click **Tools | Subsystem Simulation**

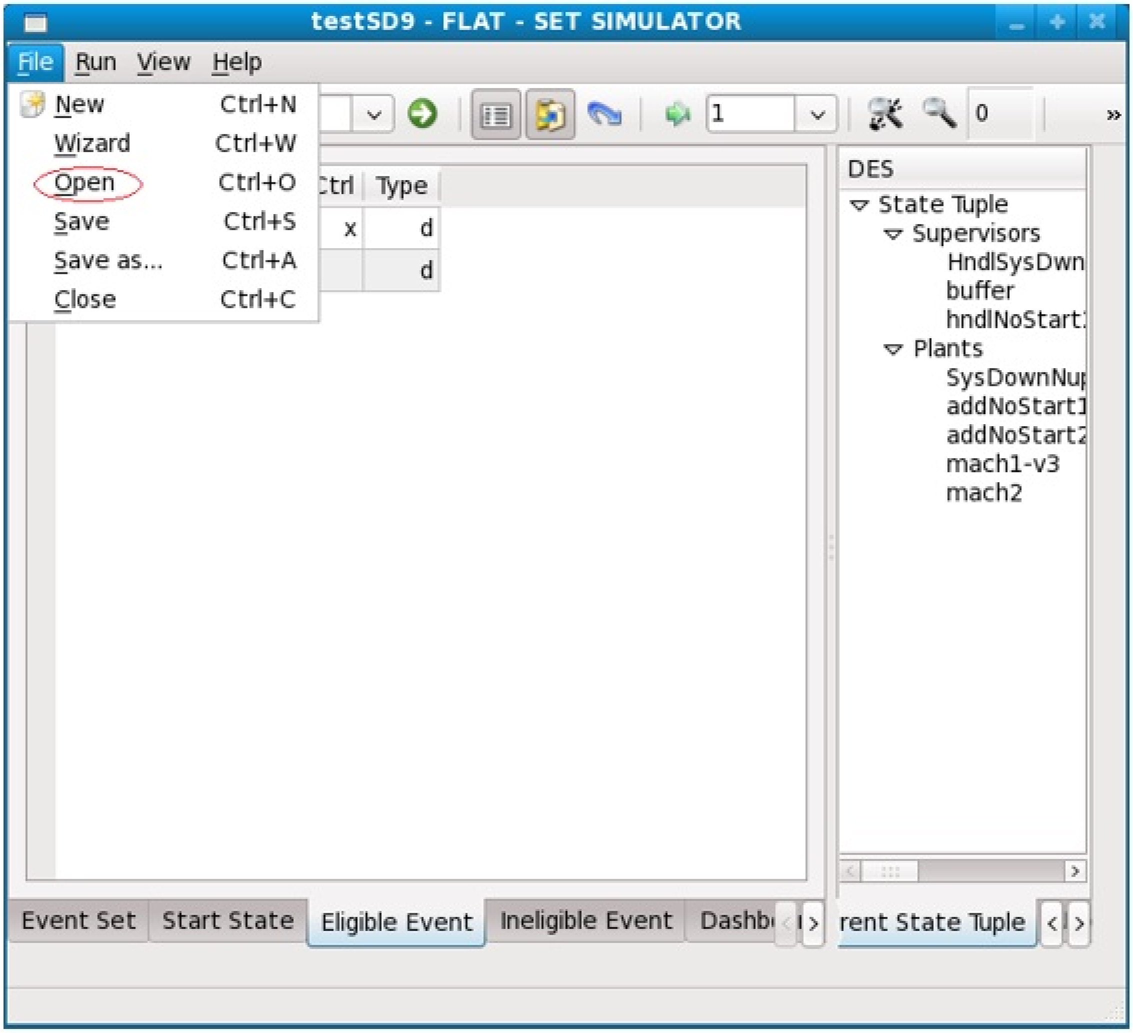
OR

Right click the subsystem and from the context menu click "Subsystem Simulation".


testHISC4 - testHISC4.desp - /home/zain/Desktop/despot/examples
Project Subsystems DES Tools BDD Tools Distributed Tools Help
Project Structure IC NB C
testHISC4
High-Level Subs...
intf1
low1
intf2
low2
Project Des Details
High-Level Subsyste
Supervisor
hsup2
hsup3
Plant
hplnt1
hplnt2
Hierarchy Level View Flat View
Project Properties
Name: testHISC4
example
verified o
0:11
been chec
ot been ch
not been c
Controllable: Unknown
Summary DES View Output
Rename F2
Delete Ctrl+Del
Add DES... Ctrl+A, Ctrl+D
Sync
Check LD Interface
Check Subsystem LD I-Consist
Check Subsystem LD Nonblocking
Check Subsystem LD Controllable
Subsystem Simulation
Check Interface (BDD)
Check Subsystem I-Consist (BDD)
Check Subsystem Nonblocking (BDD)
Check Subsystem Controllable (BDD)
Subsystem Level-Wise Synthesis (BDD)

3. The configuration wizard pops up. Click on the Cancel button.

4. From the menu bar, click **File | Open**



5. Select the file and click on the Open button.

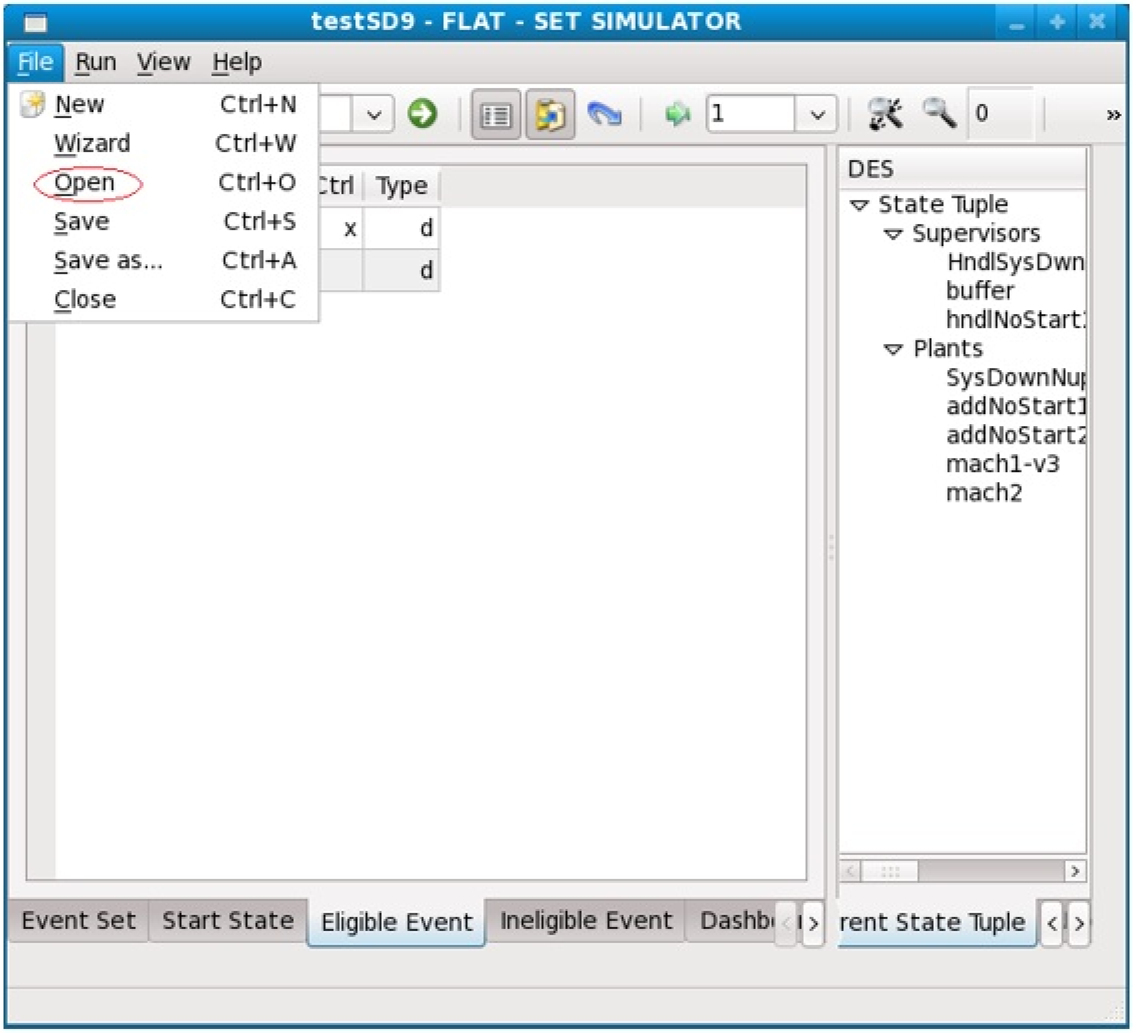
Loading a Counter Example Simulation for a Complete System

1. From the DESpot main window, open the project for which you want to load the simulation.

2. Click on the button.

3. The configuration wizard pops up. Click on the Cancel button.

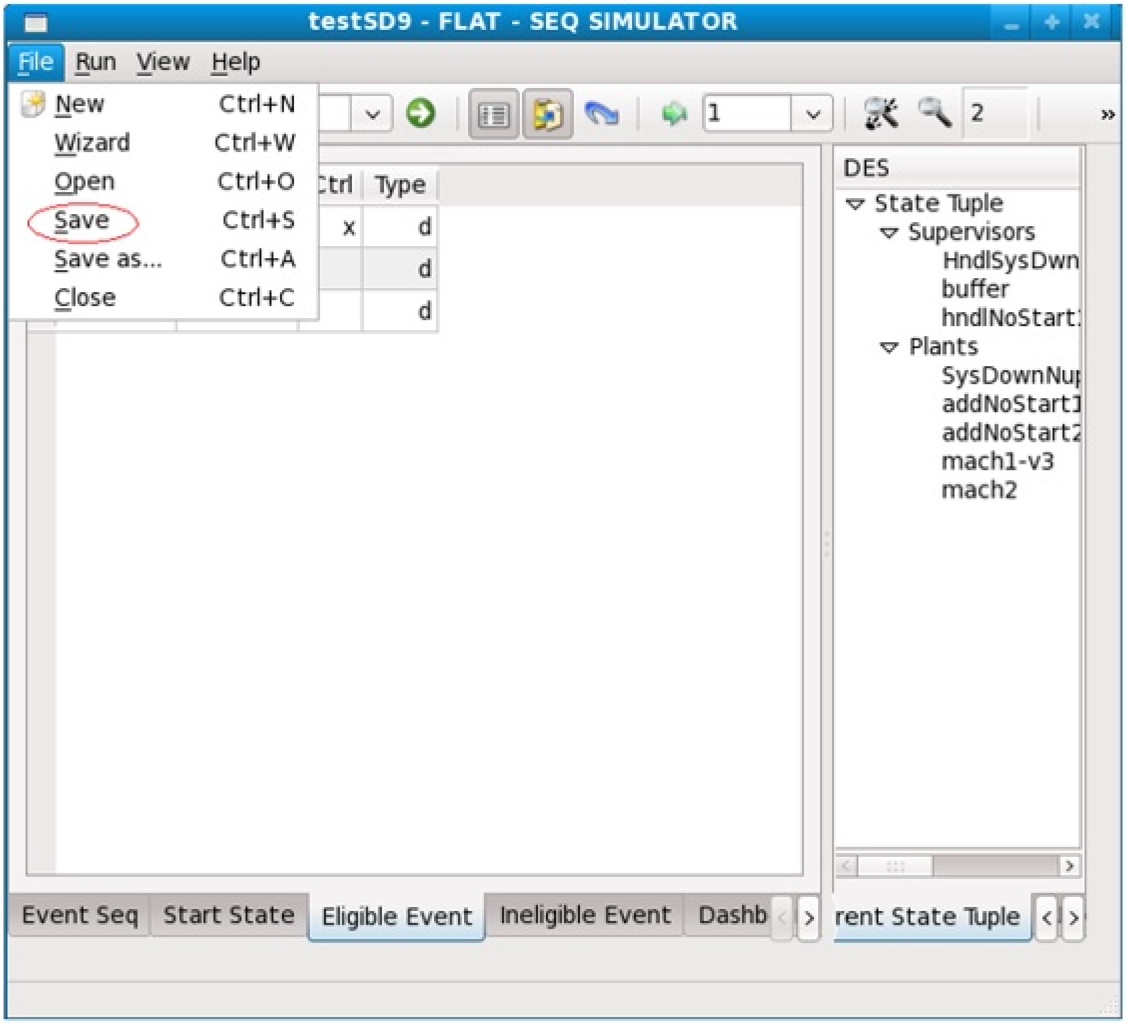
4. From the menu bar, click **File | Open**



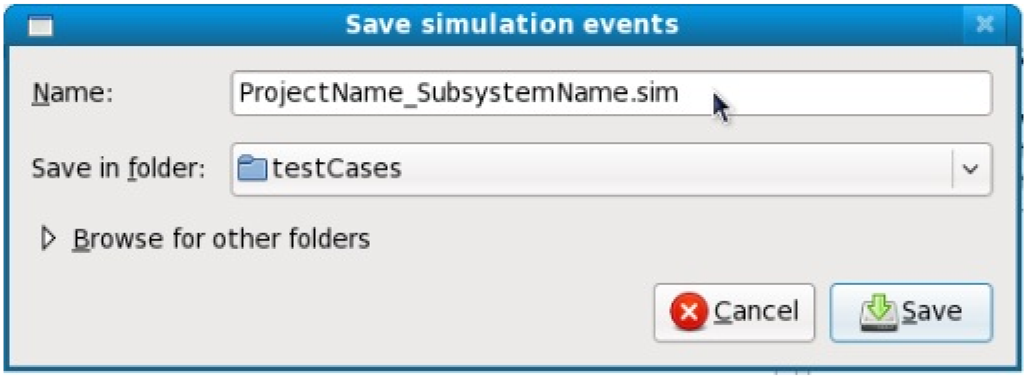
5. Select the file and click on the Open button.

Saving a Counter Example Simulation

1. From the menu, click **File | Save**



2. In the save simulation dialog that pops up, enter the name for the simulation. It is recommended that you enter the project name and subsystem name (if this is a HISC project) to aid in identifying the error subsystem when loading the simulation.



Save simulation events

Name: ProjectName_SubsystemName.sim

Save in folder: testCases

3. Press the Save button.

Debugging in Multiple Sessions

In order to allow debugging over multiple sessions, users are provided with the features to save their entire simulation work and resume later from the exact state.

Click on the appropriate topic for a step-by-step procedure:
Saving a Simulation
Loading a Simulation for a Complete System
Loading a Simulation for a Subsystem

Algorithms that do not Generate a Counter Example

1. Project integrity check
2. Project meet
3. Project sync
4. Project synthesis
5. BDD meet
6. BDD sync product
7. BDD synthesis
8. Level wise synthesis
9. BDD subsystem level wise synthesis
10. Check LD interface
11. All distributed algorithms

Moreover, no counter example is generated in the following scenarios:

1. Every algorithm first checks the project integrity, if the integrity fails the algorithm is not run; so no counter example is generated.

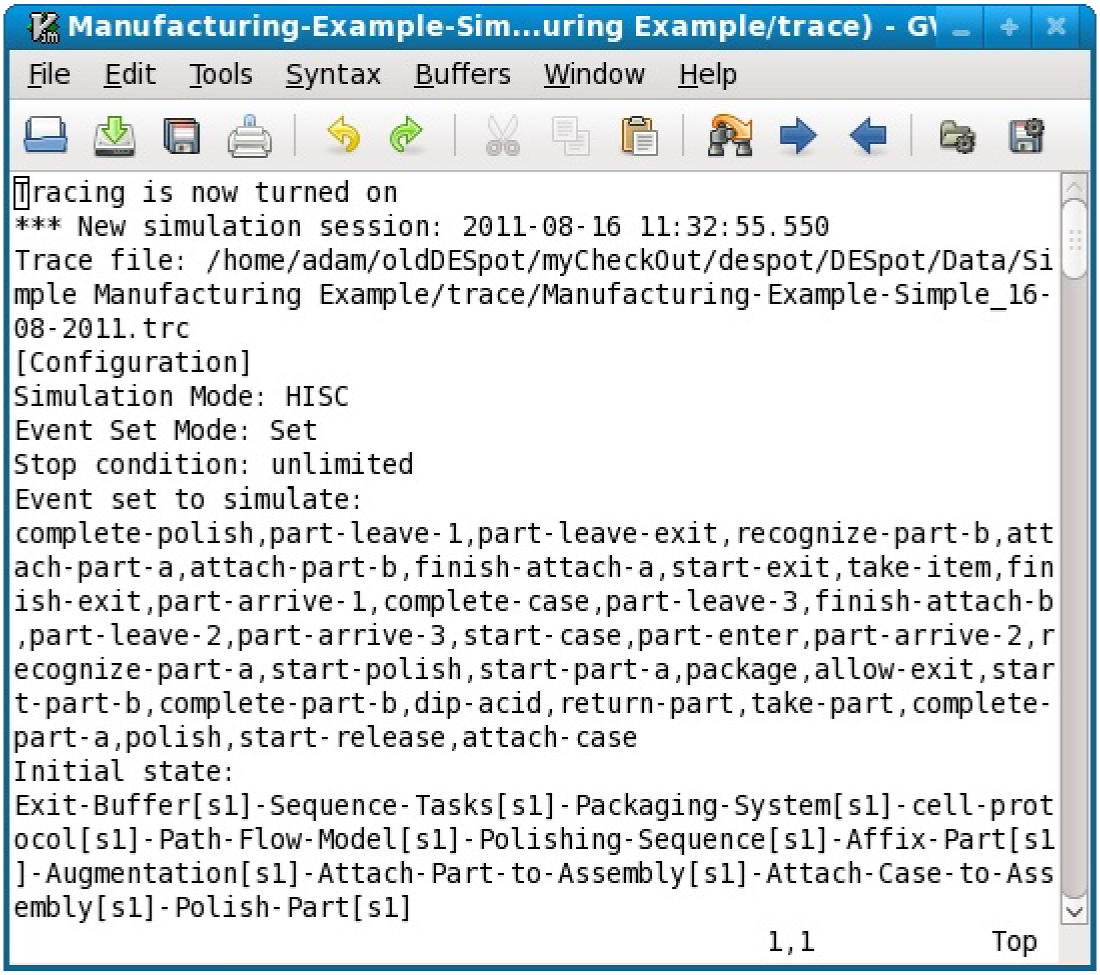
2. If the interface consistency check fails at point 2, no counter example is generated.

3. The SD controllability check and the S Singular prohibitive behaviour check are currently combined, if the result of these tests is not determined, no counter example is generated.

SIM: Simulation Viewers

Trace Viewer

The Trace Viewer is used to display the trace file in an external viewer tool such as gvim or notepad. To open the Trace Viewer, open the **View** menu and select **Trace Viewer**. You can also click on the **Trace Viewer** button on the toolbar



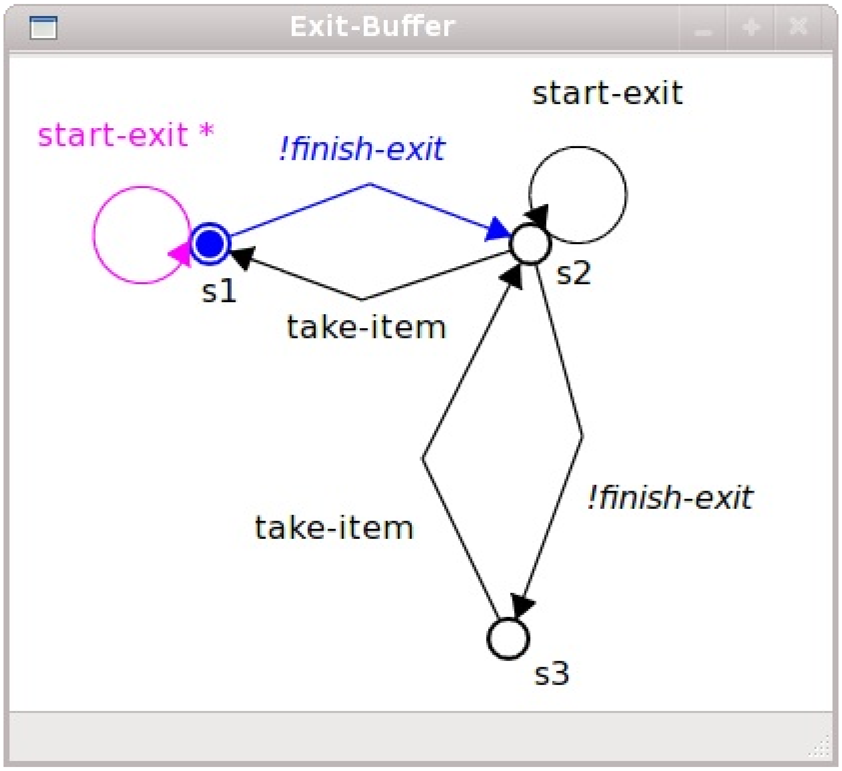
```
Tracing is now turned on
*** New simulation session: 2011-08-16 11:32:55.550
Trace file: /home/adam/oldDESpot/myCheckOut/despot/DESpot/Data/Si
mple Manufacturing Example/trace/Manufacturing-Example-Simple_16-
08-2011.trc
[Configuration]
Simulation Mode: HISC
Event Set Mode: Set
Stop condition: unlimited
Event set to simulate:
complete-polish,part-leave-1,part-leave-exit,recognize-part-b,att
ach-part-a,attach-part-b,finish-attach-a,start-exit,take-item,fin
ish-exit,part-arrive-1,complete-case,part-leave-3,finish-attach-b
,part-leave-2,part-arrive-3,start-case,part-enter,part-arrive-2,r
ecognize-part-a,start-polish,start-part-a,package,allow-exit,star
t-part-b,complete-part-b,dip-acid,return-part,take-part,complete-
part-a,polish,start-release,attach-case
Initial state:
Exit-Buffer[s1]-Sequence-Tasks[s1]-Packaging-System[s1]-cell-prot
ocol[s1]-Path-Flow-Model[s1]-Polishing-Sequence[s1]-Affix-Part[s1
]-Augmentation[s1]-Attach-Part-to-Assembly[s1]-Attach-Case-to-Ass
embly[s1]-Polish-Part[s1]
```

Depending on your selected Trace Viewer Tool, the trace will open with that program. To select a different Trace Viewer Tool, see the Dashboard tab of the Events Pane. When a Trace Viewer is open, you will see all the trace that has been written to the file since you started your simulation. Depending on your program, you may need to refresh the file as you simulate.

Graphical Viewer

The Graphical Viewer is used to watch a simulation taking place as it happens graphically, using the graphical representation of each DES. To open the Graphical Viewer, open the **View** menu and select **Graphical Viewer**. You can also click on the **Graphical Viewer** button on the toolbar 🔍. This will shift the simulator window to the upper left corner of the screen and, if no DES are currently displayed, it will open a viewer window for each in a tiled fashion. If a single DES viewer is open, it will close all viewer windows.

If you would like to open a single DES in the Graphical Viewer, double-click on the DES of your choice in the **Current State Tuple** tab of the States Pane.



The Graphical Viewer uses different colours to show the simulation's evolution. A DES's states and transitions are defaulted to the colour **Black** except when they are taking part in the current step of the simulation in some way. A state gets coloured **Blue** when it is the current state in that DES. A transition also gets coloured **Blue** when it is on the list of eligible events for the current step. You may notice you will not always have a **Blue** transition for a given DES. This is because you may not always have an eligible event to change the state of your DES. When an event occurs that changes the state of the DES you are viewing, the transition arrow will flash **Blue** for a moment and the current state will change. At this point you will notice another colour. When a DES has changed state, the transition arrow that brought the DES into that state will change to **Pink** and the transition name label will also be **Pink** with a *. It's also possible to have multiple events on the same transition arrow. Only the event labels that are eligible or previous will be coloured. If an event is selflooped, it could be an eligible event and the last event at the same time. In this case, the transition arrow body is **Blue**, and the transition arrow head is **Pink**. The name label will be **Blue**, and the star **Pink**. You could have a selfloop with two events, with one eligible and one last. In this case the arrow will be **Blue** with **Pink** arrowhead and the two event labels will be coloured appropriately.

Template DES

A template DES is a regular DES with one or more parameter variables in its name and in the name of some or all of its state and event labels. This new type of DES allows pairs of percent symbols (%) in the DES name, state names and event names. The variable between two '%' symbols can be assigned values by instantiating the template. Every variable that appears in a state name or event name inside the DES, must also appear in the name of the DES.

A template DES can then be instantiated by specifying whether it is to be a plant or supervisor, and specifying the values the templates' variables can take on. For each distinct value assigned to a variable (or tuple of values assigned to a set of variables), one DES (called an **instantiated DES**) will be created and added to the project by replacing every instance of the variable (including the enclosing '%') with the specified value. This allows the user to easily specify multiple DES that are identical up to relabelling.

NOTE: Template DES are treated as separate from the project and are ignored by the algorithms. Only the instantiated DES, which are created on the fly from the templates and instantiation information, are visible to the verification algorithms. Also, instantiated DES can be viewed but not edited. Instead, the template must be edited and then the instantiation must be regenerated.

In a flat project, instantiation DES can be created as a plant or supervisor. An instantiation name must be unique to the project.

In an HISC project, a template can be added to any subsystem or interface. The same template DES can appear in more than one subsystem or interface. Instantiation DES can be created as part of a subsystem as a plant or supervisor, or as part of an interface as a supervisor. An instantiation name must be unique within a given subsystem or interface, but is not required to be unique to the entire project.

We can check all the related properties for the instantiations by running existing algorithms. Template DES can be saved and loaded for further work. However, the instantiations cannot be saved separately. DESpot only stores instantiation information in the project file. The instantiations are generated immediately when the project loads.

DESpot provides the following template features:

- Instantiate Templates
- Edit Instantiations
- Delete Instantiations

- Regenerate Instantiations

Instantiate Template

The purpose of template DES is to automate the creation of a group of DES that are identical up to relabeling. The idea is to create a single template DES with one or more template variables, and then to use the template to create the actual DES to be used in verification algorithms by assigning a range of values to the template variables.

The user specifies values for the template variables, and then DESpot creates a new DES identical to the template DES but with all template variables replaced.

There are two methods to assign values to template variables in order to create instantiations:

Range Option
Tuple Option

In a flat project, the name of instantiations must be unique.

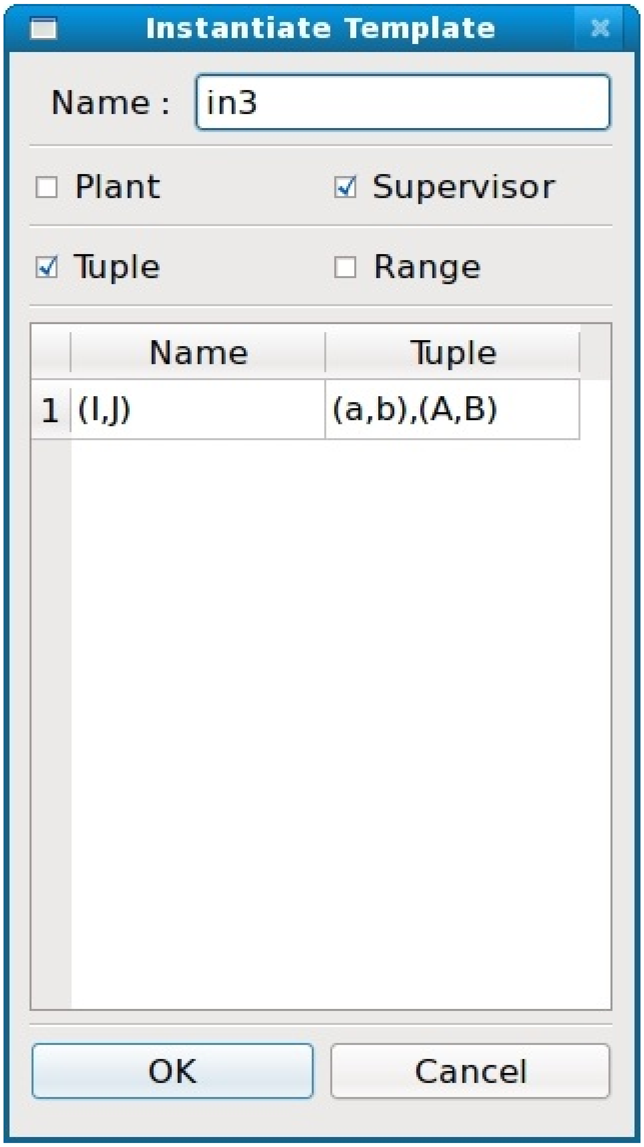
In an HISC project, instantiations can only be added to the subsystem or interface that the template belongs to. The names of instantiations must be unique in the same subsystem or interface. However, they can be duplicated in different subsystems or interfaces. The instantiation information can be viewed in both Hierachy View and Level View in the project editor. They are displayed below the template DES. When instantiating a template to an interface, only supervisors can be specified.

Tuple Option

The tuple method assigns tuples explicitly to the template variables when a template DES contains more than one variable. For example, if we wanted (i,j)=(1,a), (2,b) only, assigning the range on the previous page would give us two unwanted DES. In the tuple method, we simply specify the tuples that we want for our variables. For both the range and tuple method, an instantiation must assign a value to all template variables in a template.

For the tuple method, each member of the tuple can be a string containing numbers, letters (upper and lower case), “-“, and “_”. A valid set of tuples would be (i,j)=(1,a), (AS1,b-z).

The following window shows an example.



Instantiate Template

Name : in3

□ Plant ☑ Supervisor

☑ Tuple □ Range

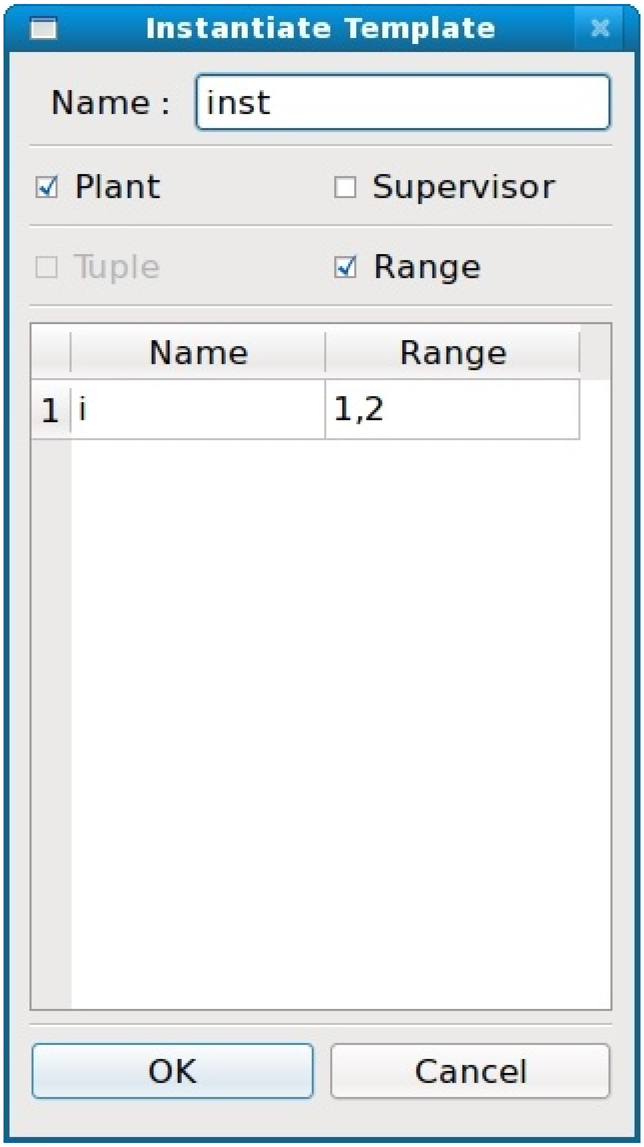
	Name	Tuple
1	(I,J)	(a,b),(A,B)

Range Option

The range method assigns a range to each template variable. For example, if a template contains variables %i% and %j%, we could assign i=1,2 and j=a,b. This will create four DES (ie produces (i,j)=(1,a), (1,b), (2,a), (2,b)).

For the range method, variables can be assigned a number or a letter (upper or lower case) or a combination of letters and numbers by using a comma to separate entries, and using “..” to specify a sequence. A valid range would be i=1,3..5,7aw,a..c. This is equivalent to i=1,3,4,5,7aw,a,b,c.

The following window shows an example.



Instantiate Template

Name : inst

☑ Plant □ Supervisor

□ Tuple ☑ Range

	Name	Range
1	i	1,2

Edit Instantiation

By double clicking the instantiation's name in the project editor, or selecting the ‘Edit instantiation information’ option by right clicking, the user can modify the instantiation information.

The instantiation template window will load the previously entered information and fill in the various dialogue values. The user can then modify these values. After clicking ‘ok’, the existing instantiated DES will be deleted, and new ones with the new information will be created.

Delete Instantiation

By right clicking and selecting "Delete Instantiation," an instantiation can be deleted. This will delete the instantiation information as well as the associated instantiated DES.

Regenerate Instantiation

To guarantee the consistency of template DES and their instantiations, we want to make sure the two stay in synchronization. When a user modifies a template DES that has been instantiated, they can use "regenerate instantiation" to regenerate all instantiations for that template DES.

Regression Test Suite

The Regression Test Suite is designed to allow future developers for DESpot to make changes to existing algorithm code and test the result obtained from the improved algorithm against the original algorithm. The Regression Test Suite will provide a pass or fail result for the algorithm the user tests on a project, which confirms whether the new implementation of the algorithm is providing correct results. The Regression Test Suite also allows users to edit a project in the project editor and run all algorithms on the project instantly by having the project open in a Regression Test Editor as well.

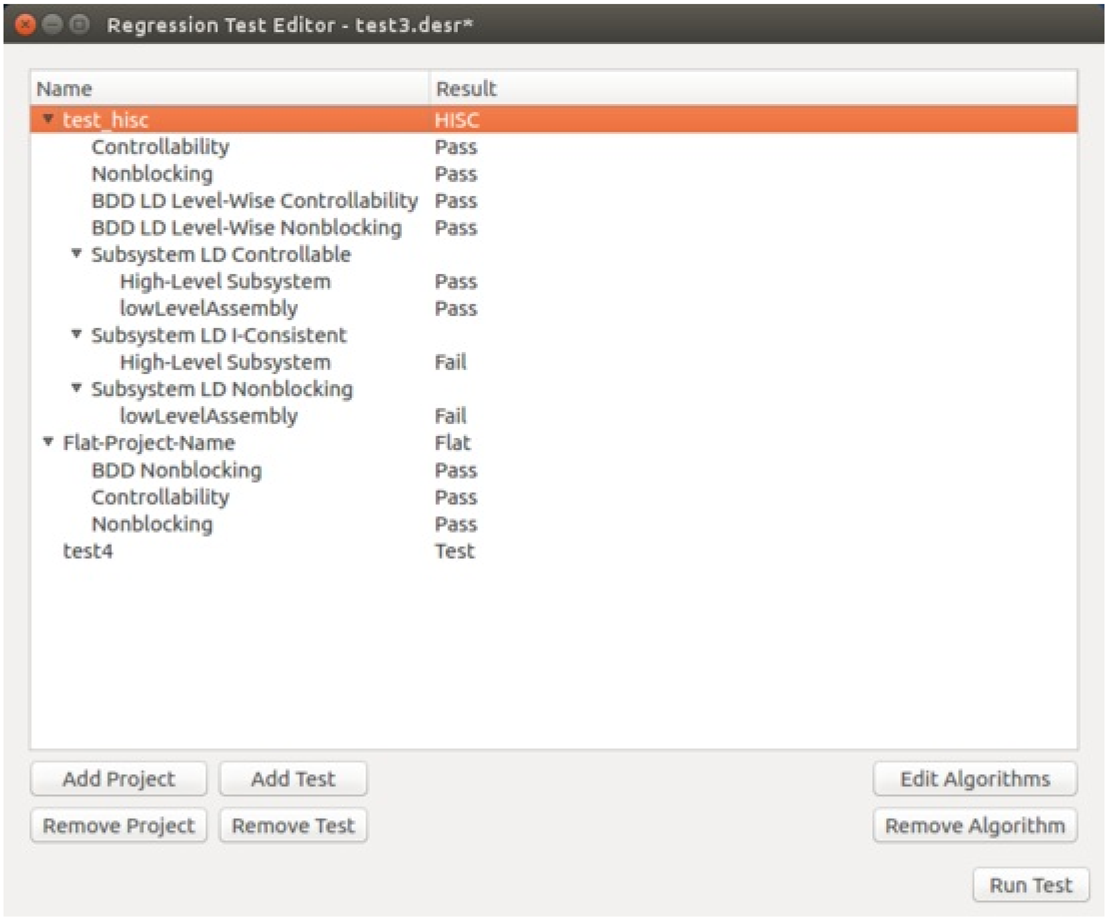
1. Regression Test Editor
2. Algorithm Editor
3. Subsystem Editor
4. Save a Regression Test
5. Regression Test Output

Regression Test Editor Window

The Regression Test Editor allows you to add or remove projects, tests, algorithms, and subsystems using the pushbuttons or matching menu options found in the **Tools** menu. The **Tools** menu also includes a **Clear Algorithms** and **Clear All** option, which remove all algorithms and subsystems from the selected project and clears all items from the Regression Test Editor respectively.

Subsystems are treated like algorithms so to remove a selected subsystem or algorithm the **Remove Algorithm** option is used. Algorithms are added to projects using the Algorithm Editor and subsystems are added to subsystem-specfic algorithms with the Subsystem Editor. These editors can also remove algorithms and subsystems from a project.

The Regression Test Editor's first column displays the name of the project, test, algorithm, or subsystem. The second column displays the project type, test label, algorithm result, and subsystem result. Please Note: only projects may have algorithms added to them, tests are a collection of projects, algorithms, and tests that can be saved and modified.

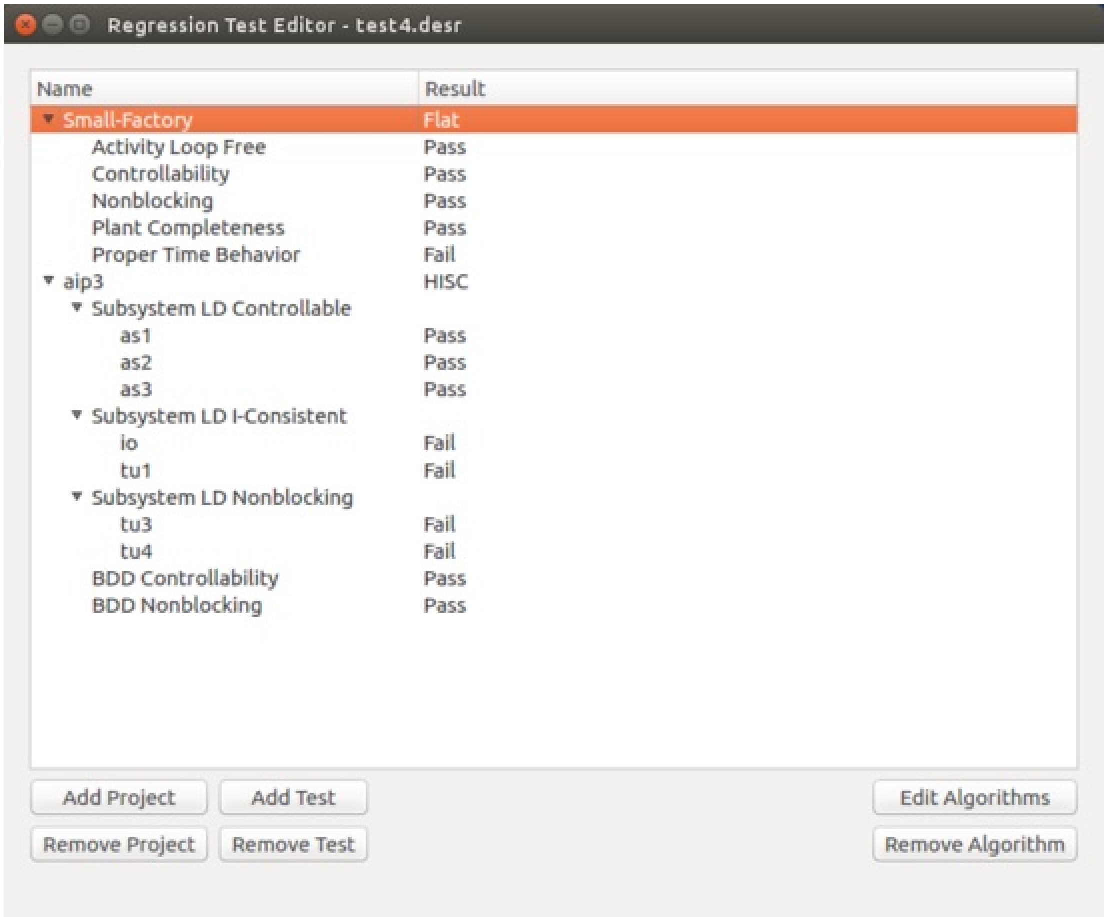


Name	Result
▾ test_hisc	HISC
Controllability	Pass
Nonblocking	Pass
BDD LD Level-Wise Controllability	Pass
BDD LD Level-Wise Nonblocking	Pass
▾ Subsystem LD Controllable	
High-Level Subsystem	Pass
lowLevelAssembly	Pass
▾ Subsystem LD I-Consistent	
High-Level Subsystem	Fail
▾ Subsystem LD Nonblocking	
lowLevelAssembly	Fail
▾ Flat-Project-Name	Flat
BDD Nonblocking	Pass
Controllability	Pass
Nonblocking	Pass
test4	Test

To edit a previously saved test, double-click on the test to open an "edit-only"

Regression Test Editor window. This editor provides the same functionality as the original editor, expect it will not allow you to execute the test. It will also not allow you to create a new Regression Test (this option is greyed out in the **File** menu).

If you double-click a test within the "edit-only" editor, another "edit-only" editor will be displayed for that test. Similarly, double-clicking a project or non-subsystem-specific algorithm will display the Algorithm Editor for that project. Double-clicking a subsystem-specific algorithm or subsystem will display the Subsystem Editor for that algorithm.



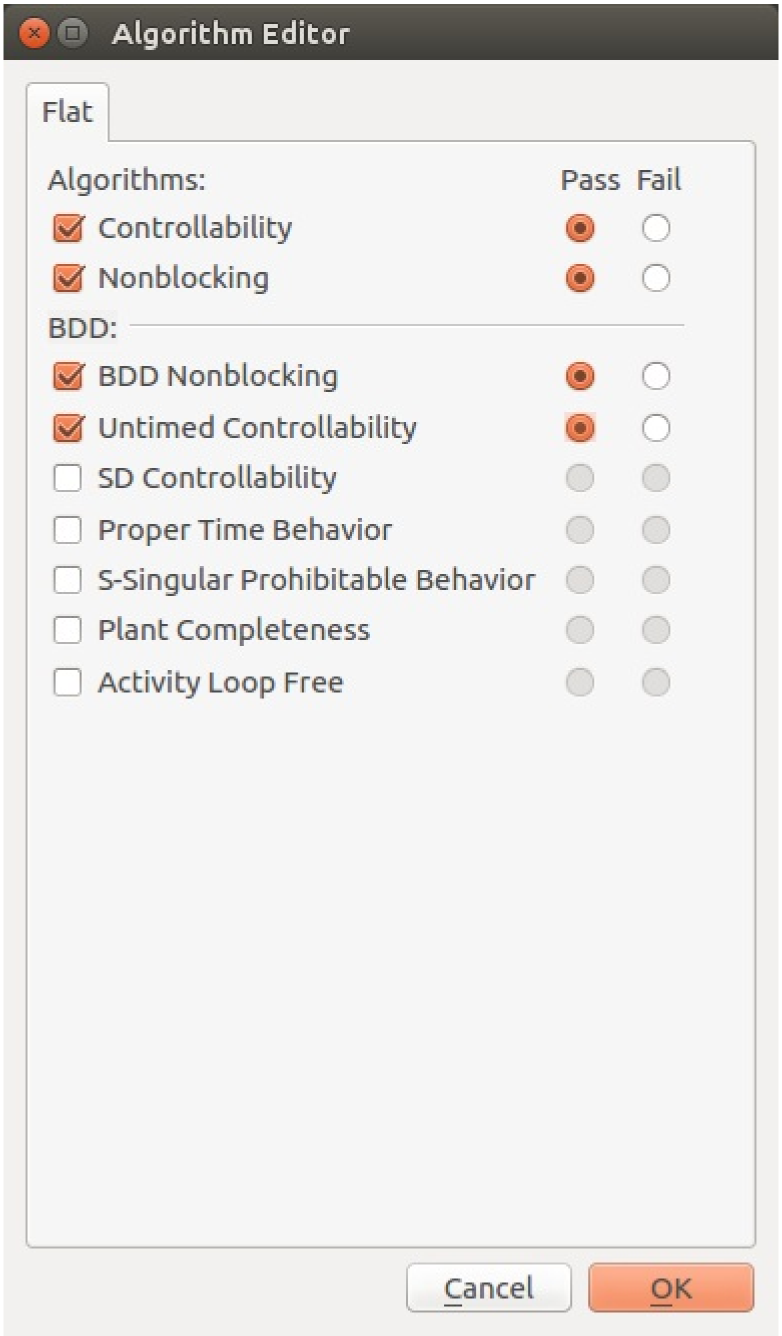
Name	Result
▾ Small-Factory	Flat
Activity Loop Free	Pass
Controllability	Pass
Nonblocking	Pass
Plant Completeness	Pass
Proper Time Behavior	Fail
▾ aip3	HISC
▾ Subsystem LD Controllable	
as1	Pass
as2	Pass
as3	Pass
▾ Subsystem LD I-Consistent	
io	Fail
tu1	Fail
▾ Subsystem LD Nonblocking	
tu3	Fail
tu4	Fail
BDD Controllability	Pass
BDD Nonblocking	Pass

Click the **Run Test** button or select the **Run | Run Test** menu option to get results for your chosen projects and algorithms. The results are displayed in the Regression Test Output window. When a test or project is added to the Regression Test Editor, only its name and type attributes are stored in the editor; the project and test data is not loaded until **Run Test** is clicked. This allows you to edit a project with DESpot's Project Editor or edit a test with the Regression Test Editor and save the changes to disk before running the test, and have updated information to run the algorithms on. Please Note: if you are editing the same test in two separate Regression Test Editors, the one that is saved most recently will be the version used in both cases.

Before any algorithms are run on a project, an integrity check is completed to ensure the project passes. If the project fails the integrity check, none of the algorithms are run and the output will be labelled with "Integrity Error".

Algorithm Editor Window

To display the Algorithm Editor for a selected project, click **Edit Algorithms** from the form's pushbuttons or the **Tools | Edit Algorithms** menu option, or double-click on a project or a non-subsystem-specific algorithm. The Algorithm Editor allows you to choose which algorithms to add to a project and select the result the algorithm is expected to return. For an algorithm to be added to the project it must have a result selected. The Algorithm Editor for a Flat project contains all relevant standard and BDD algorithms that return a pass or fail result.



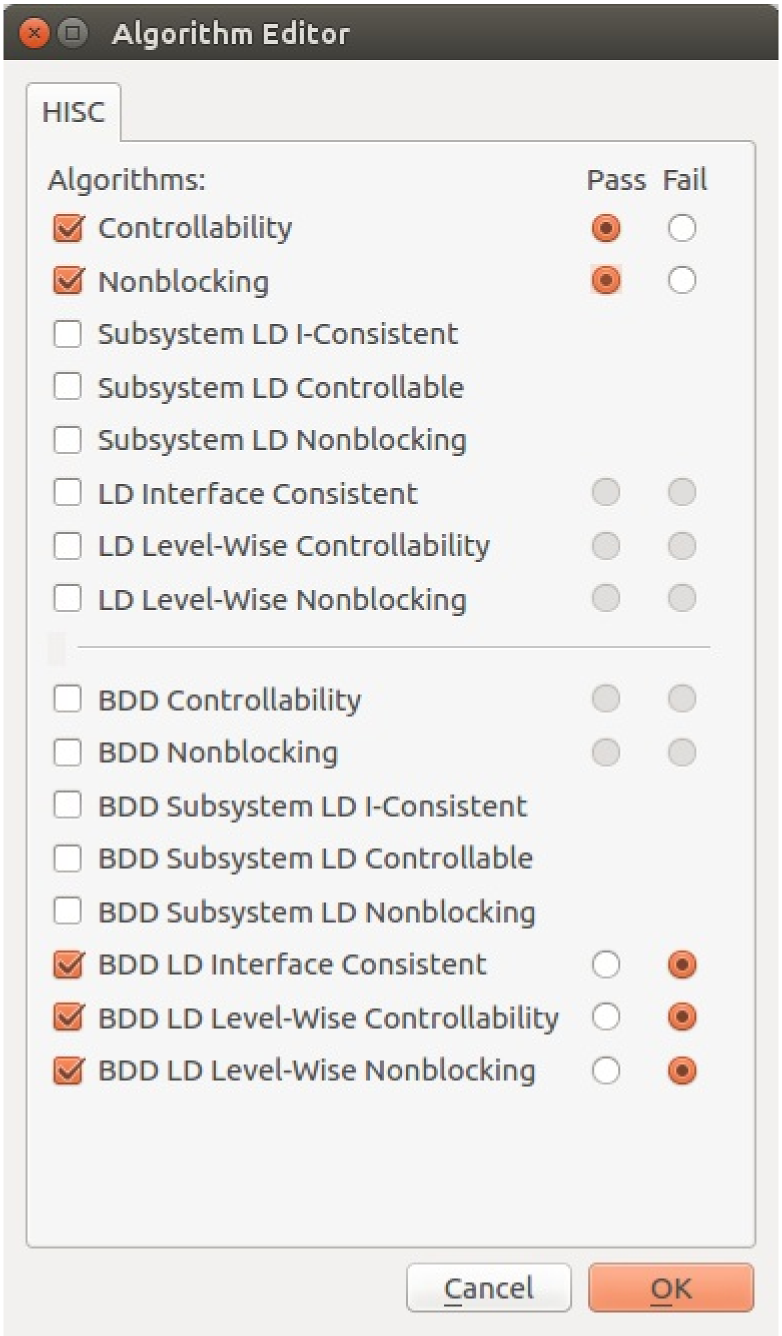
Algorithms: Pass Fail

- ☑ Controllability ● ○
- ☑ Nonblocking ● ○

BDD:

- ☑ BDD Nonblocking ● ○
- ☑ Untimed Controllability ● ○
- □ SD Controllability ○ ○
- □ Proper Time Behavior ○ ○
- □ S-Singular Prohibitable Behavior ○ ○
- □ Plant Completeness ○ ○
- □ Activity Loop Free ○ ○

The Algorithm Editor for a HISC project also contains all relevant standard and BDD algorithms that return a pass or fail result. The HISC algorithms include subsystem-specific algorithms. The subsystem-specific algorithms do not have results associated with them but if one is selected, the Subsystem Editor is displayed, which allows a result to be chosen for each subsystem.

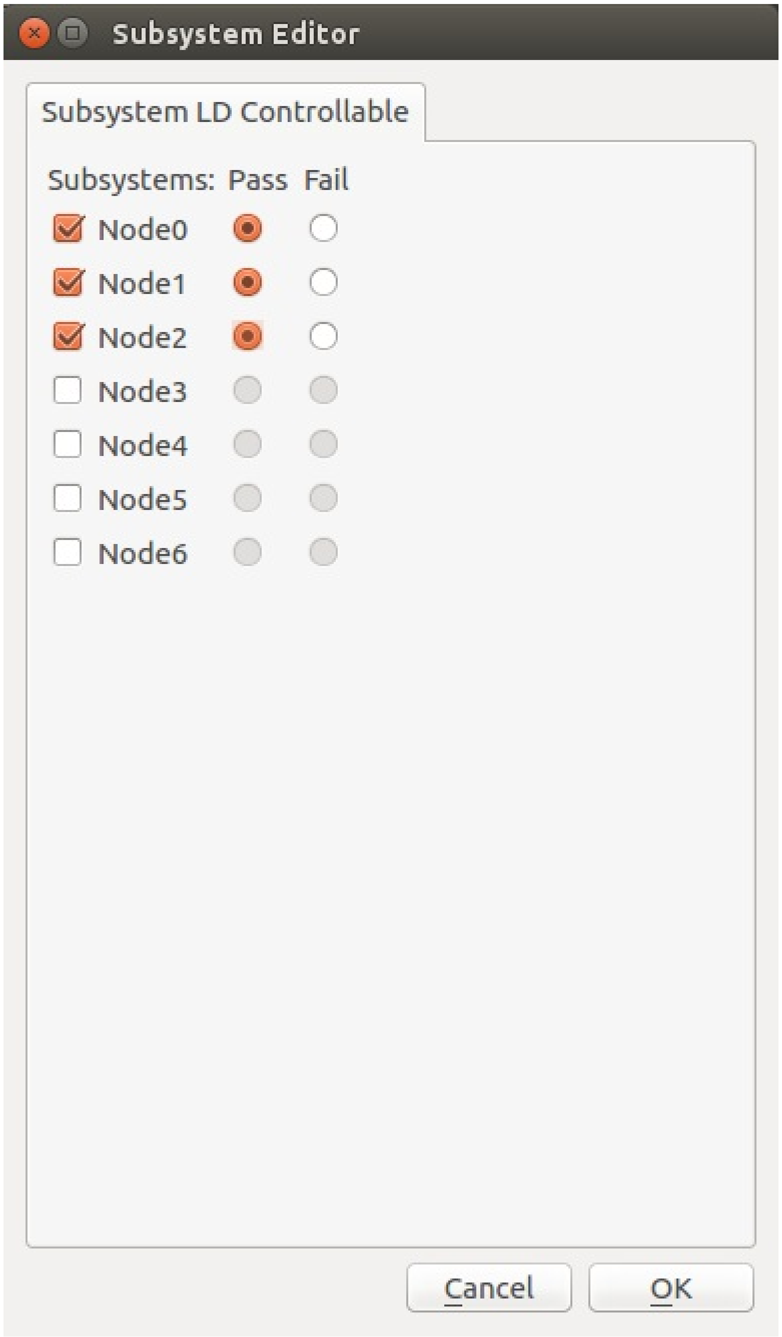


Algorithms:	Pass	Fail
☑ Controllability	●	○
☑ Nonblocking	●	○
☐ Subsystem LD I-Consistent		
☐ Subsystem LD Controllable		
☐ Subsystem LD Nonblocking		
☐ LD Interface Consistent	○	○
☐ LD Level-Wise Controllability	○	○
☐ LD Level-Wise Nonblocking	○	○
☐ BDD Controllability	○	○
☐ BDD Nonblocking	○	○
☐ BDD Subsystem LD I-Consistent		
☐ BDD Subsystem LD Controllable		
☐ BDD Subsystem LD Nonblocking		
☑ BDD LD Interface Consistent	○	●
☑ BDD LD Level-Wise Controllability	○	●
☑ BDD LD Level-Wise Nonblocking	○	●

The Algorithm Editor also allows you to remove algorithms from a project. Once the project has algorithms assigned to it, the Algorithm Editor will populate itself with those algorithms each time it is opened. An algorithm can be removed by unchecking it from the Algorithm Editor or by using the **Remove Algorithm** option. However, unchecking all algorithms from the Algorithm Editor will have no affect. To clear all algorithms from a selected project, use the **Tools | Clear Algorithms** menu option.

Subsystem Editor Window

The Subsystem Editor is displayed only if you select a subsystem-specfic algorithm from the Algorithm Editor and select **Ok** or if you double-click on a subsystem-specfic algorithm, or its subsystems, from the Regression Test Editor.



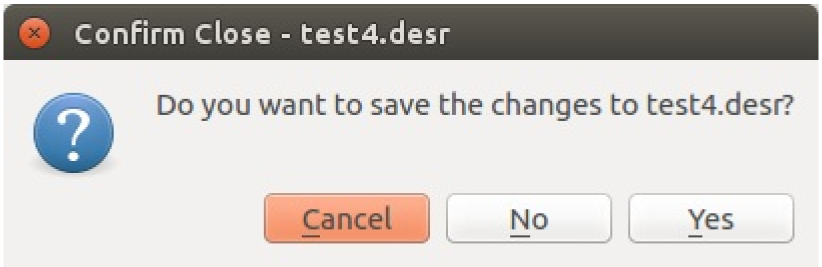
Subsystems:	Pass	Fail
☑ Node0	●	○
☑ Node1	●	○
☑ Node2	●	○
□ Node3	○	○
□ Node4	○	○
□ Node5	○	○
□ Node6	○	○

The window displays the current algorithm you are selecting subsystems for. Only the subsystems selected with a pass or fail result will be added to the Regression Test Editor. Like the Algorithm Editor, the Subsystem Editor will populate itself with the algorithm's assigned subsystems. Subsystems can be removed by unchecking them from the Subsystem Editor or by using the **Remove Algorithm** option. If you choose to remove a subsystem-specific algorithm from a project, all subsystems associated with the algorithm will also be removed.

Save a Regression Test

In either the Regression Test Editor or "edit-only" Regression Test Editor, you may save a test to file, allowing you to open it again to modify the contents and rerun the algorithms. Tests are saved with the extension ".desr". To save a test, press **Ctrl+S** or select the **File | Save** menu option. The **File | Save As** menu option is also available for saving tests to a different name.

The editor's window title displays the test name if one has been saved and "*.desr" if it is a new test. The "*" symbol will appear after a saved test's name in the window title if the test has been modified without saving. In this case, if you close the window or DESpot's Main Window, a close confirmation dialog will appear prompting you if the test should be closed without saving, saved before closing, or the close operation should be cancelled.


Confirm Close - test4.desr

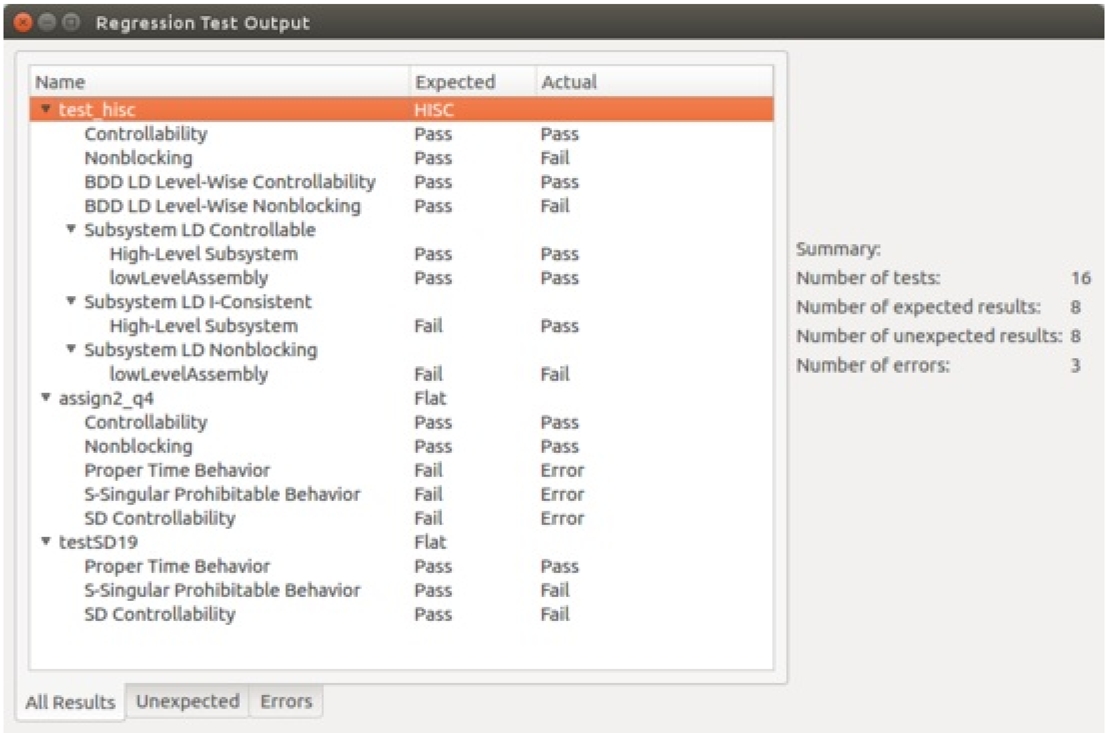
Do you want to save the changes to test4.desr?

Tests may be saved recursively, i.e. test1.desr contains test2.desr and visa versa, however test3.desr may not contain test3.desr, this will cause an error message when attempting to save.

Regression Test Output Window

The Regression Test Output window provides you with all the projects, algorithms, and expected results you selected along with the actual results obtained by running the algorithms on the projects. The window is separated into 4 sections: the **All Results**, **Unexpected**, and **Errors** tabs, and the **Summary**.

The **All Results** tab displays a list of all projects, each with the algorithms run on them. The third column is the **Actual** results column, which can display "Pass", "Fail", "Error", or "Integrity Error" for each algorithm or subsystem. The Regression Test Suite only runs an algorithm on a project once, so if tests containing the same information were run, only one copy of the results are included.

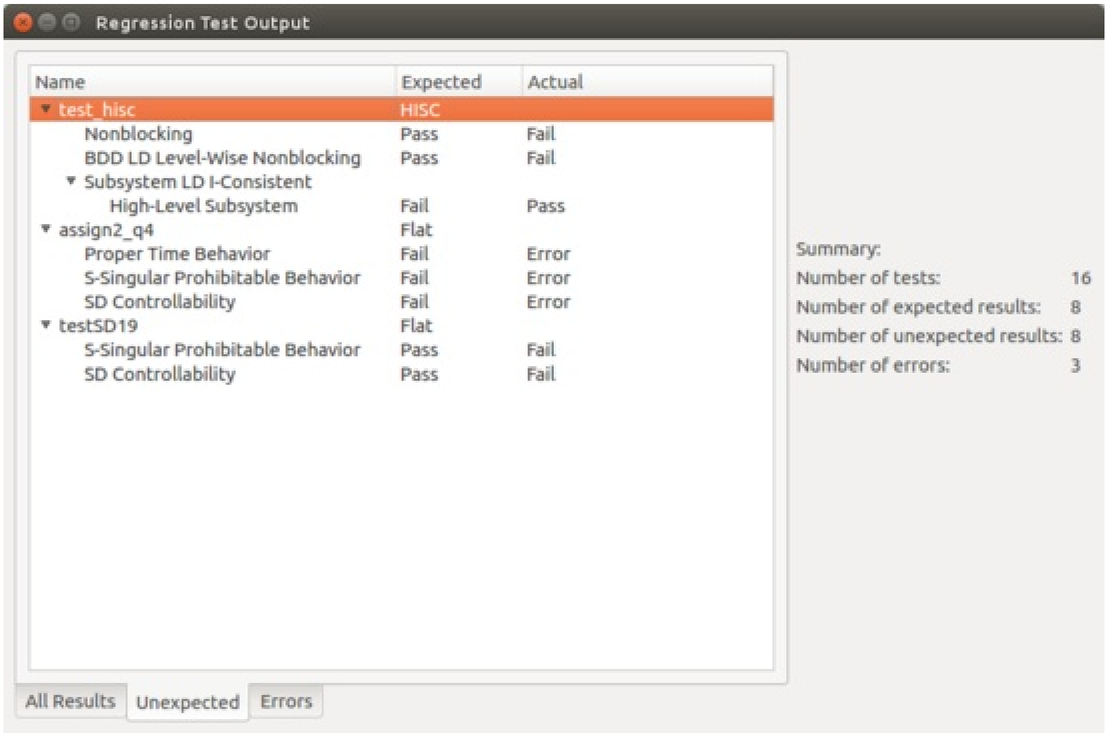


Name	Expected	Actual
▾ test_hisc	HISC	
Controllability	Pass	Pass
Nonblocking	Pass	Fail
BDD LD Level-Wise Controllability	Pass	Pass
BDD LD Level-Wise Nonblocking	Pass	Fail
▾ Subsystem LD Controllable		
High-Level Subsystem	Pass	Pass
lowLevelAssembly	Pass	Pass
▾ Subsystem LD I-Consistent		
High-Level Subsystem	Fail	Pass
▾ Subsystem LD Nonblocking		
lowLevelAssembly	Fail	Fail
▾ assign2_q4	Flat	
Controllability	Pass	Pass
Nonblocking	Pass	Pass
Proper Time Behavior	Fail	Error
S-Singular Prohibitable Behavior	Fail	Error
SD Controllability	Fail	Error
▾ testSD19	Flat	
Proper Time Behavior	Pass	Pass
S-Singular Prohibitable Behavior	Pass	Fail
SD Controllability	Pass	Fail

Summary:

Number of tests:	16
Number of expected results:	8
Number of unexpected results:	8
Number of errors:	3

If an algorithm's expected result is different than the result obtained by the Regression Test, it is added to the **Unexpected** tab. This tab also includes algorithms that have experienced errors. The **Unexpected** tab provides a concise view of where problems may lie in modified DESpot code or in a projects implementation.

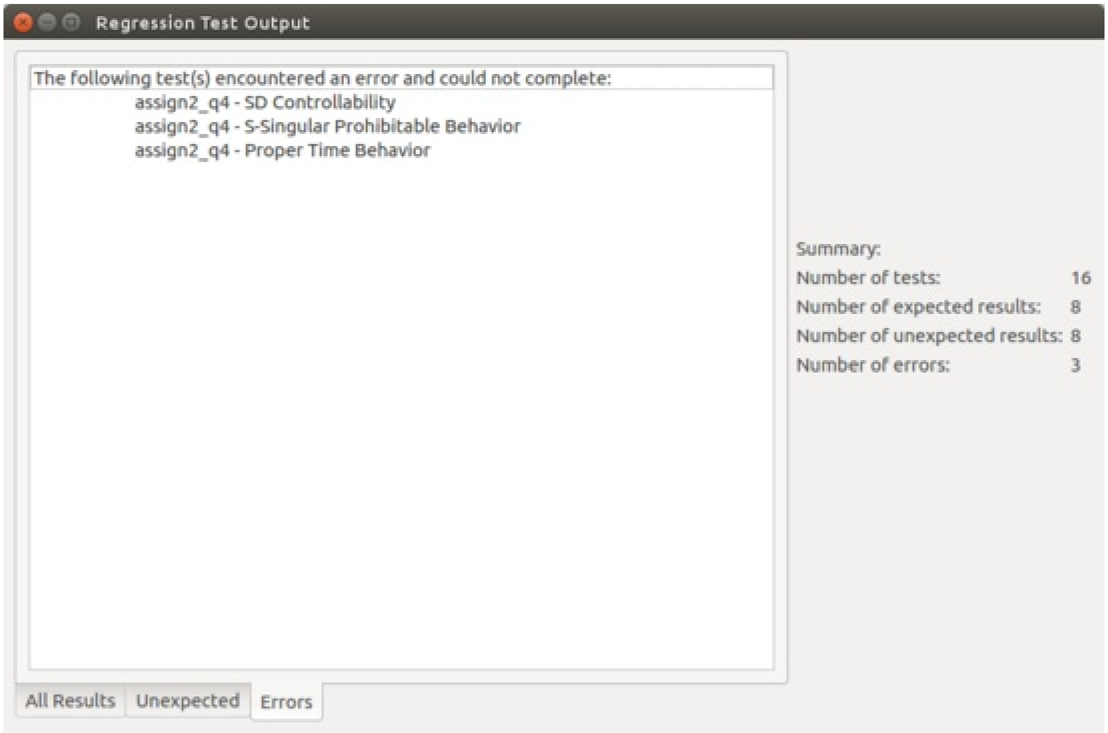


Name	Expected	Actual
▾ test_hisc	HISC	
Nonblocking	Pass	Fail
BDD LD Level-Wise Nonblocking	Pass	Fail
▾ Subsystem LD I-Consistent		
High-Level Subsystem	Fail	Pass
▾ assign2_q4	Flat	
Proper Time Behavior	Fail	Error
S-Singular Prohibitable Behavior	Fail	Error
SD Controllability	Fail	Error
▾ testSD19	Flat	
S-Singular Prohibitable Behavior	Pass	Fail
SD Controllability	Pass	Fail

Summary:

Number of tests:	16
Number of expected results:	8
Number of unexpected results:	8
Number of errors:	3

The **Errors** tab filters the algorithms that recieved an "Error" or "Integrity Error" result from the test. To obtain more detail on why the error occurred, the algorithm could be run on the project in DESpot's Project Editor. If a project fails the integrity check, all remaining algorithms for the project will not be run and instead labeled with "Integrity Error".



```
The following test(s) encountered an error and could not complete:
        assign2_q4 - SD Controllability
        assign2_q4 - S-Singular Prohibitable Behavior
        assign2_q4 - Proper Time Behavior
```

```
Summary:
Number of tests:               16
Number of expected results:    8
Number of unexpected results:  8
Number of errors:              3
```

The **Summary** on the right hand side of the window provides a quick-look at the testing results. If the Regression Test Editor that created the Regression Test Output is closed, the output window will also be closed.

If a project's saved location has changed and the Regression Test Editor is unable to locate the project, the project will be labelled with "Project not found" in the **All Results** and **Errors** tabs.