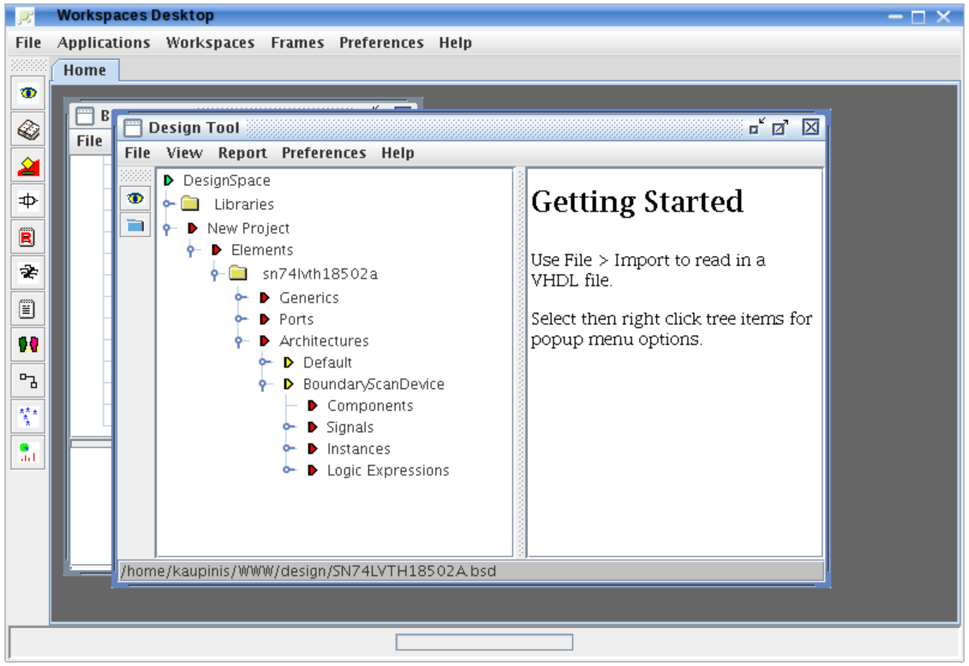
eightolives.com

Scripting in the Workspaces Desktop

William_Kaupinis@eightolives.com
March 16, 2010

Introduction



Use File > Import to read in a VHDL file.

Select then right click tree items for popup menu options.

/home/kaupinis/WWW/design/SN74LVTH18502A.bsd

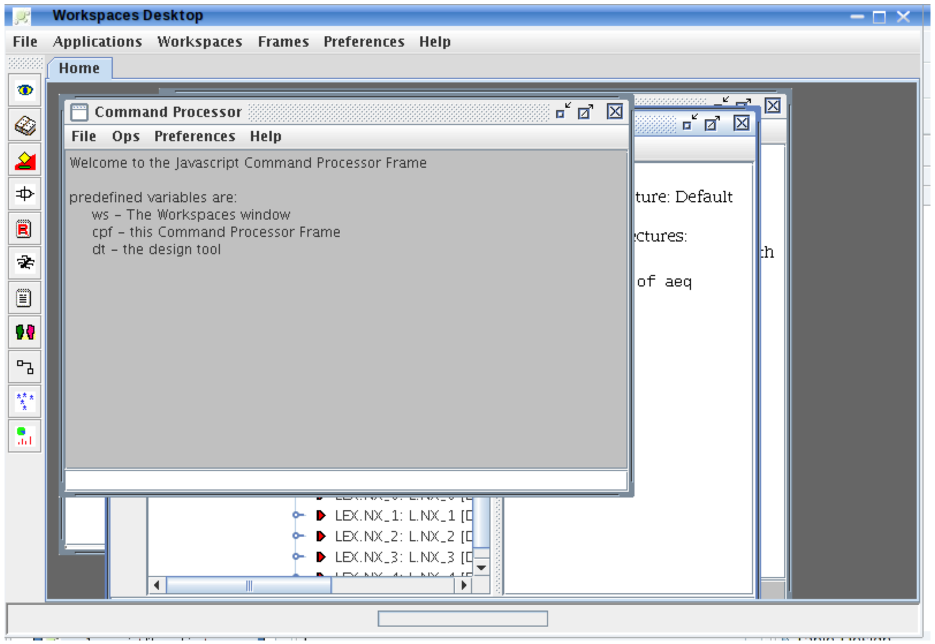
Includes Bookmark tree, Design Tool, Editor, Browser, Requirements Tracker, Process Tracker, Bug Tracker

- Workspaces Desktop is a Java-based GUI with a tool set helpful in digital design
- Scripting using ECMA Javascript allows customizable flexibility

Why scripting?

- Scripting gives you the power to create custom operations on a design or document
- Library of script functions can add new functionality as required
- Scripts can automate operations and tests

Start the tool


Workspaces Desktop
File Applications Workspaces Frames Preferences Help
Home
Command Processor
File Ops Preferences Help

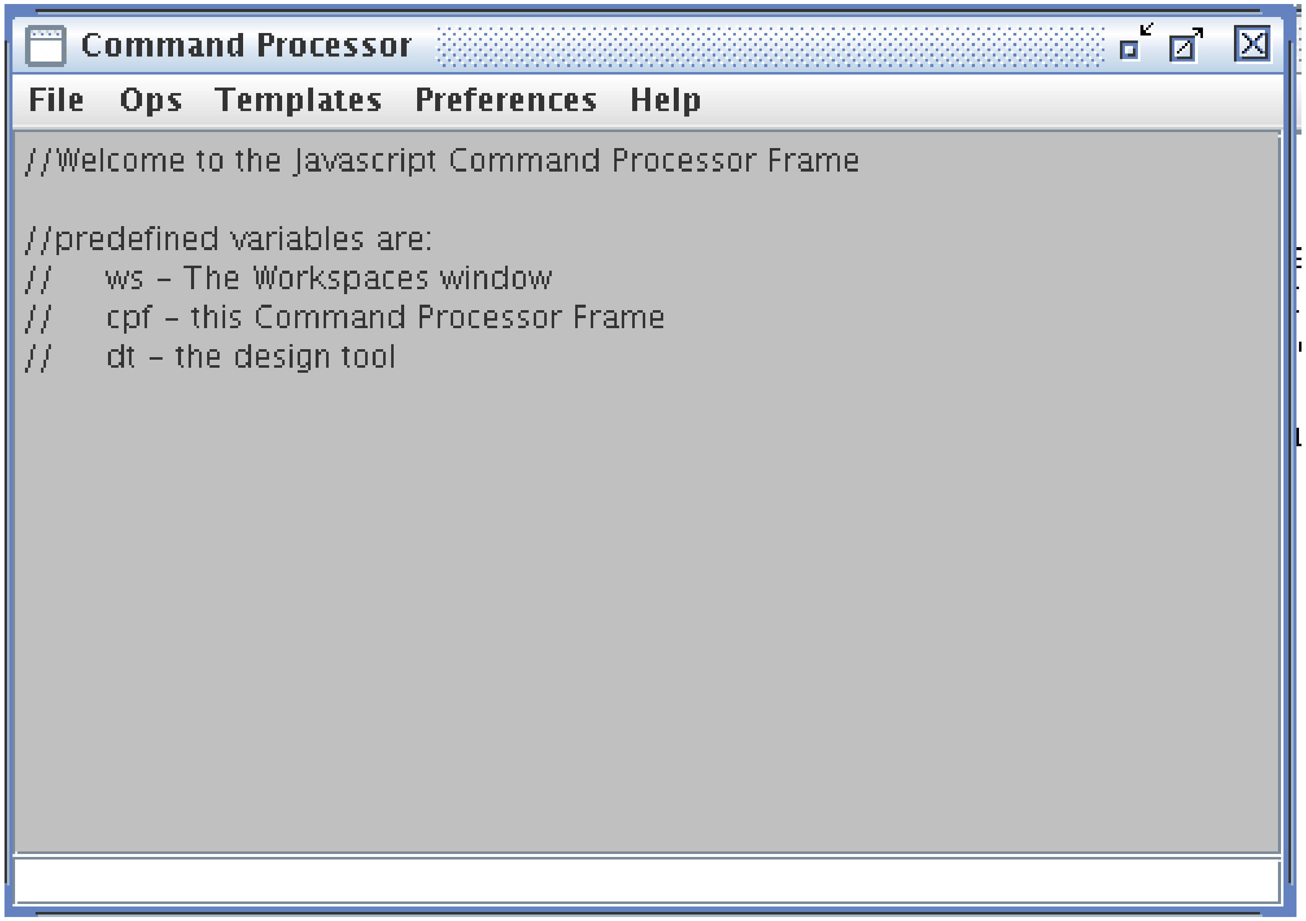
```
Welcome to the Javascript Command Processor Frame

predefined variables are:
    ws - The Workspaces window
    cpf - this Command Processor Frame
    dt - the design tool
```

- The Command Processor tool is your link to scripting
- Invoke the tool from the Workspaces Applications menu or from the Design Tool View menu
- Predefined objects are the gateways into the design.

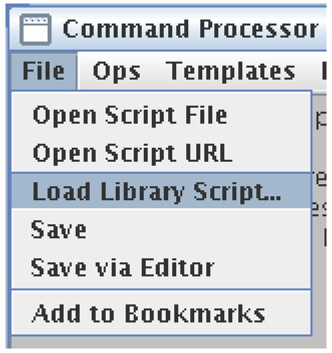
eightolives.com

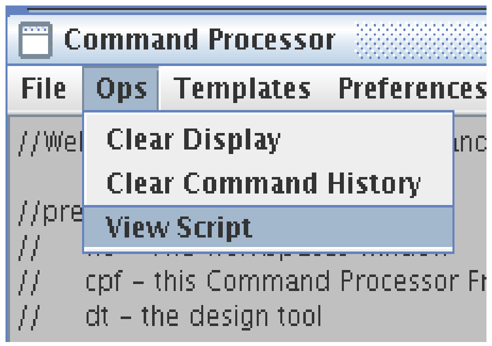
The Command Processor

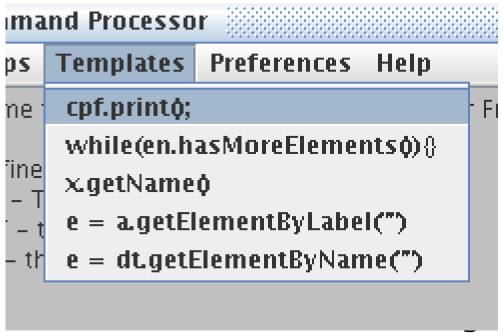

Command Processor

```
//Welcome to the Javascript Command Processor Frame

//predefined variables are:
//     ws - The Workspaces window
//     cpf - this Command Processor Frame
//     dt - the design tool
```







```
cpf.print();
while(en.hasMoreElements()){
x.getName()
e = a.getElementByLabel("")
e = dt.getElementByName("")
```

- Enter commands in the bottom text field or use menu options
- Up/Down arrows scroll through command history

Javascript likes Objects

- Designs are represented by an Object Oriented (OO) Hardware API
- Workspaces tools are written in OO Java
- Javascript can access both these API structures

- Typical format is:
  object.function(arg1, arg2)
- Useful examples:

```
dt = ws.openItem(“design.vhd”);
ed = ws.openEditor(“a.txt”);
we = dt.getWorkingElement();
cpf.print(“Entity name is “ + we.getName());
ports = we.getPorts(); // a Vector
K = ports.size(); // size of the vector
p = ports.elementAt(i); // element of vector
```

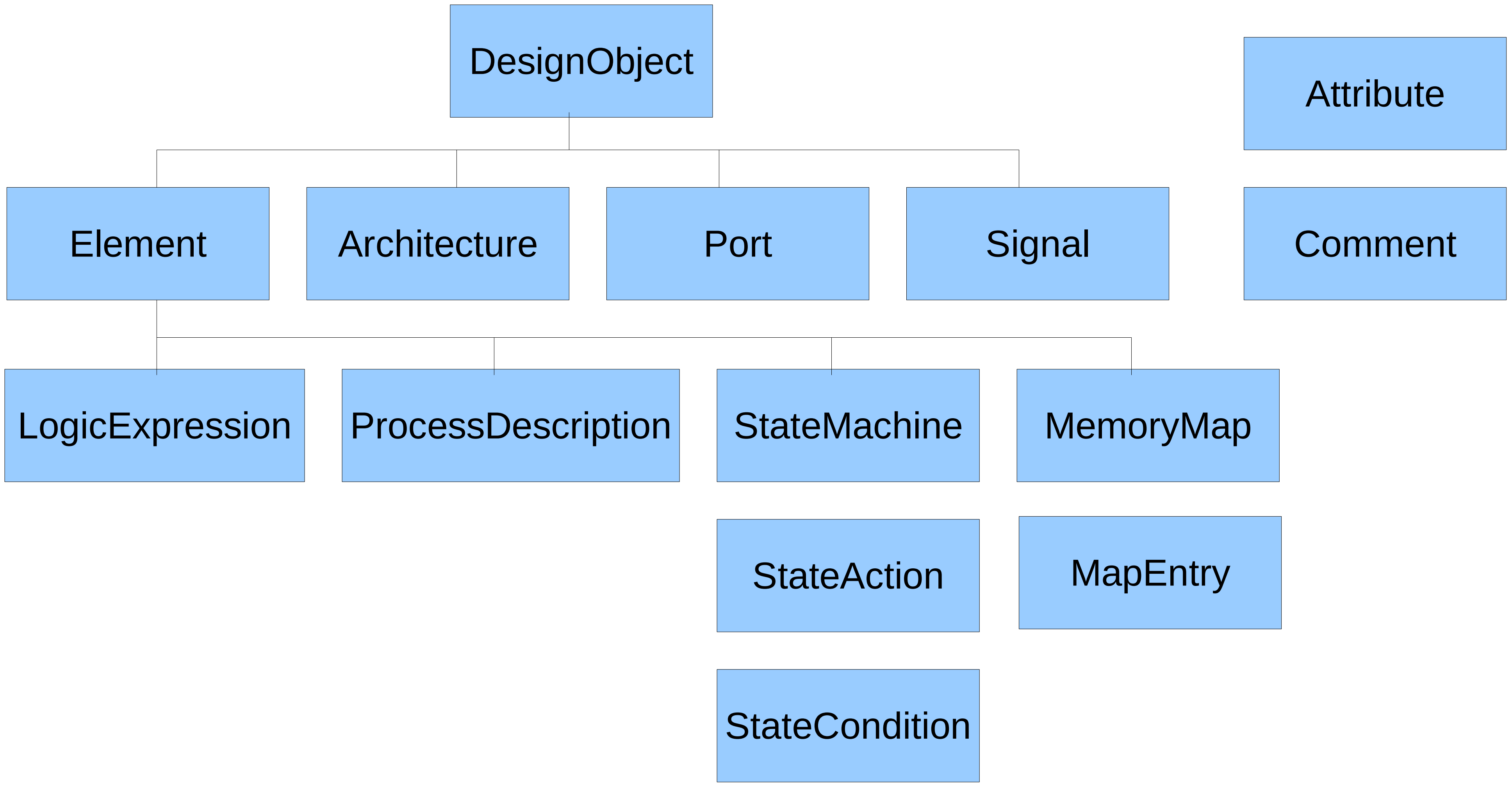
Example: Listing I/Os

```
// This script lists the I/O Ports of a design component in an Editor window.
print("------- This is a demo file: test.js  ---------\n"); // prints the message to Java console or command line
dt = ws.openItem("c:/temp/sprite.vhd"); // brings up the Design Tool dt; ws is a pre-defined object representing the Workspaces Desktop
we = dt.getWorkingElement(); // an Element represents a component or VHDL design in the Hardware API
cpf.print("Working element is " + we.getName() + "\n"); // prints the message to the Command Processor window
s = we.getName() + "\n" +"\nList of Ports:\n";  // s is a String in which we will accumulate our results for printout
v = we.getPorts(); // v is a Vector (collection) of I/O Ports in the Element
I = 0; // I is an integer
k = v.size(); // k, an integer, is set to the number of Ports in Vector v
while(i < k) // a loop
   {
   p = v.elementAt(i);  // let p be one of the Ports
   s += p.getName() + " - " + p.getCommentAsString() + "\n"; // get its name and associated comment
   i += 1;
   } // end of loop
ed = ws.openEditor("results.txt");  // open an Editor window in the Workspaces GUI
ed.setText(s); // put the results String s in the editor
```

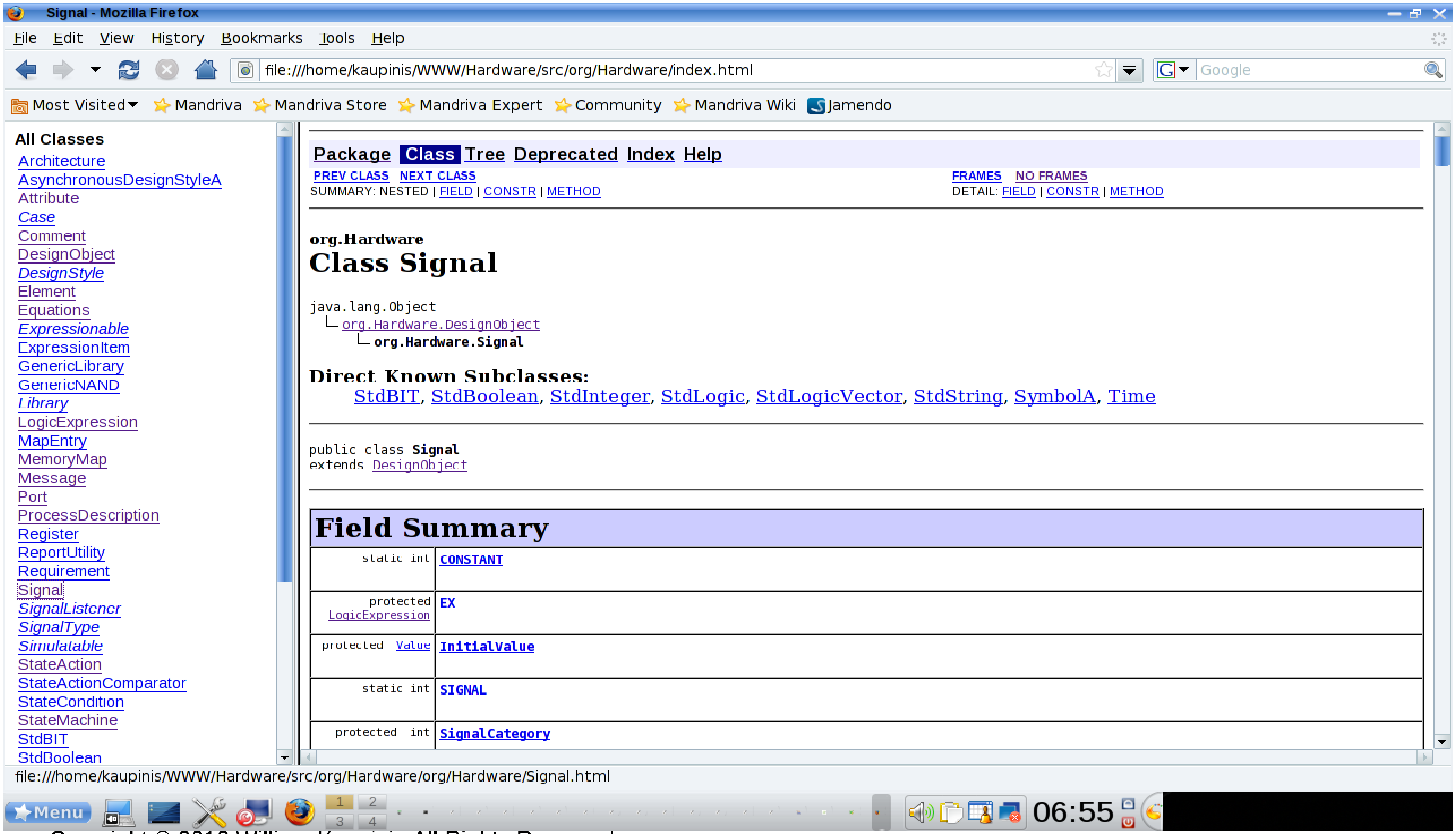
Example: Functions

```
// example function definition
function shallCounter(sb)
{
i = 0;
count = 0;
j = sb.indexOf(“shall”);
while(j != -1)
    {
    count +=  1;
    j = sb.indexOf(“shall, j+1);
    }
return(count);
}
```

Key Hardware API Objects



Javdoc pages define the APIs


Signal - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///home/kaupinis/WWW/Hardware/src/org/Hardware/index.html
Google
Most Visited Mandriva Mandriva Store Mandriva Expert Community Mandriva Wiki Jamendo
Package Class Tree Deprecated Index Help
PREV CLASS NEXT CLASS
FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD
DETAIL: FIELD | CONSTR | METHOD
file:///home/kaupinis/WWW/Hardware/src/org/Hardware/org/Hardware/Signal.html
Menu
06:55

All Classes

Architecture
AsynchronousDesignStyleA
Attribute
Case
Comment
DesignObject
DesignStyle
Element
Equations
Expressionable
ExpressionItem
GenericLibrary
GenericNAND
Library
LogicExpression
MapEntry
MemoryMap
Message
Port
ProcessDescription
Register
ReportUtility
Requirement
Signal
SignalListener
SignalType
Simulatable
StateAction
StateActionComparator
StateCondition
StateMachine
StdBIT
StdBoolean

org.Hardware

Class Signal

```
java.lang.Object
  └ org.Hardware.DesignObject
      └ org.Hardware.Signal
```

Direct Known Subclasses:
StdBIT, StdBoolean, StdInteger, StdLogic, StdLogicVector, StdString, SymbolA, Time

```
public class Signal
extends DesignObject
```

Field Summary

static int	CONSTANT
protected LogicExpression	EX
protected Value	InitialValue
static int	SIGNAL
protected int	SignalCategory

For More Information

- Check the tutorials at http://www.eightolives.com/tutorials.htm
  - Workspaces Desktop Tool Overview
  - Modeling Hardware in Java
- Read the Workspaces Desktop Users Manual
- ECMA Javascript info at

  http://www.ecma-international.org/publications/standards/Ecma-262.htm