%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Test cases for Matlab Functions

% Section : EXTERNAL INTERFACES

% 1. Standard Libraries

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[res,st] = calllib('shrlibsample','addStructByRef',struct);

m = libfunctions('shrlibsample', '-full');

libfunctions shrlibsample -full;

libfunctionsview('shrlibsample');

ibisloaded('shrlibsample');

pv = libpointer('int16Ptr', v);

sc = libstruct('c\_struct', sm);

loadlibrary shrlibsample shrlibsample.h alias lib;

loadlibrary('libmx', hfile, 'mfilename', 'mxproto');

unloadlibrary shrlibsample

% many variations

class(obj);

obj = class(s,'class\_name');

names = fieldnames(s);

names = fieldnames(obj);

names = fieldnames(obj, '-full');

import containers.Map;

import java.util.Enumeration java.lang.String;

inspect(h);

tf = isjava(A);

javaaddpath({

'C:\Work\Java\Curvefit\Test', ...

'C:\Work\Java\mywidgets.jar'});

dblArray = javaArray ('java.lang.Double', 4, 5);

javachk('awt', component);

javaclasspath({

'C:\Work\Java\ClassFiles', ...

'C:\Work\JavaTest\curvefit.jar', ...

'C:\Work\JavaTest\timer.jar', ...

'C:\Work\JavaTest\patch.jar'});

javaMethod('isNaN','java.lang.Double',2.2);

javaMethodEDT('add',v,'string');

strObj = javaObject('java.lang.String','hello');

f = javaObjectEDT('javax.swing.JFrame', 'New Title');

javarmpath('C:\Work\Java\mywidgets.jar');

methods('MException');

methodsview java.awt.MenuItem;

usejava('awt');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% .NET

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

enableNETfromNetworkDrive;

asm = NET.addAssembly('System.Windows.Forms');

% Netassemblyclass

arr =NET.convertArray(A, 'System.Int32');

strArray = NET.createArray('System.String', 3);

dblLst = NET.createGeneric('System.Collections.Generic.List', ...

{'System.Double'}, 10);

NET.disableAutoRelease(comObj);

A = NET.enableAutoRelease(obj);

vpType = NET.GenericClass('System.Collections.Generic.KeyValuePair',...

'System.Int32', 'System.String');

a = NET.invokeGenericMethod(obj, ...

'myGenericSwapMethod', ...

{'System.Double', 'System.Double'}, ...

5, 6);

NET.setStaticProperty('MyTestObject.MyClass.myStaticProperty', 5);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% ACTX

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

cal = actxcontrol('mscal.calendar', [0 0 600 500], f);

list = actxcontrollist;

[h, info] = actxcontrolselect;

h = actxGetRunningServer('progid');

e = actxserver ('Excel.Application');

h.addproperty('Position');;

h.delete;

delete(h);

h.deleteproperty('propertyname')

deleteproperty(h,'propertyname')

state = enableservice('AutomationServer',true);

info = h.eventlisteners;

info = eventlisteners(h);

S = h.events;

S = events(h);

result = h.Execute('command');

result = Execute(h, 'command');

result = invoke(h, 'Execute', 'command');

a = h.Feval('strcat', 1, 'hello', ' world');

names = fieldnames(obj, '-full');

V = h.get;

V = h.get('propertyname');

str = h.GetCharArray('varname', 'workspace');

str = GetCharArray(h, 'varname', 'workspace');

MReal = h.GetFullMatrix('M','base',zeros(5),zeros(5));

C2 = h.GetVariable('C1','base');

D = h.GetWorkspaceData('varname','workspace');

D = GetWorkspaceData(h,'varname','workspace');

inspect([h1,h2]);

customlist = h.interfaces;

customlist = interfaces(h);

c1 = h.invoke('ICalc1');

S = h.invoke('methodname', arg1, arg2);

tf = h.iscom;

tf = iscom(h);

tf = h.isevent('eventname');

tf = isevent(h, 'eventname');

tf = h.isinterface;

tf = isinterface(h);

tf = h.ismethod('methodname');

tf = ismethod(h, 'methodname');

tf = h.isprop('propertyname');

tf = isprop(h, 'propertyname');

h.load('mwsample');

h.MaximizeCommandWindow;

MaximizeCommandWindow(h);

methods('classname');

methodsview classname;

methodsview(object);

h.MinimizeCommandWindow;

MinimizeCommandWindow(h);

pos = h.move([50 50 200 200]);

h.propedit;

propedit(h);

h.PutCharArray('Foo', 'base', chArr);

h.PutFullMatrix('varname','workspace',xreal,ximag);

PutFullMatrix(h,'varname','workspace',xreal,ximag);

MatLab.PutWorkspaceData("A","base",data);

h.Quit;

Quit(h);

invoke(h, 'Quit');

h.registerevent(eventhandler);

registerevent(h, eventhandler);

h.release;

release(h);

h.save('filename');

save(h, 'filename');

h.set('pname1', value1, 'pname2', value2);

h.unregisterallevents;

unregisterallevents(h);

h.unregisterevent(eventhandler);

unregisterevent(h, eventhandler);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% WEBSERVICES

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

response = callSoapService(...

'http:%test/soap/services/LibraryCatalog',...

'urn:LibraryCatalog#getAuthor',...

message);

createClassFromWsdl('pathto\_myWebService');

message = createSoapMessage(...

'urn:LibraryCatalog',...

'getAuthor',...

{'In the Fall'},...

{'nameToLookUp'},...

{'{http:%www.w3.org/2001/XMLSchema}string'},...

'rpc');

author = parseSoapResponse(response);

% SERIALPORTDEVICES

clear obj;

delete(obj);

obj;

disp(obj);

fclose(obj);

tline = fgetl(obj);

[tline,count] = fgetl(obj);

[tline,count,msg] = fgetl(obj);

tline = fgets(obj);

[tline,count] = fgets(obj);

[tline,count,msg] = fgets(obj);

fopen(s);

fprintf(obj,'format','cmd','mode');

A = fread(obj,size,'precision');

A = fscanf(obj,'format',size);

fwrite(obj,A,'precision','mode');

out = get(obj,'PropertyName');

% unusual use of instrcallback

set(s,'OutputEmptyFcn',@instrcallback);

instrcallback(obj,event);

out2 = instrfind({'Port','BaudRate'},{'COM2',4800});

obj = instrfindall(props,vals);

isvalid(sarray);

length(obj);

load MyObject s2;

readasync(s);

record(s,'on');

save('MySerial2','s');

save MySerial1 s;

serialbreak(obj);

serialbreak(obj,time);

set(obj,PN,PV);

props = set(obj,'PropertyName');

[m,n] = size(obj);

m = size(obj,dim);

stopasync(obj);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%