**Test Cases**

# **Introduction**

## **Purpose**

This document documents test procedures and results for Unified Code Count Tool (UCC) Matlab language support. Requirements, installation procedures and usage instructions are documented in UCC User Manual and Release Notes.

## **References**

## **Definitions, Acronyms and Abbreviations**

# **Test Cases**

## **Test Case <1>**

### ***Test Objectives***

The main objective of this test is to verify that the UCC is counting Matlab Data Analysis keywords.

### ***Test Description***

*Execute UCC with the input Matlab code file as followed*

-dir <source files directory> UCC\_Matlab\_dataanalysis\_Keywords\_Test.m

*Verify UCC output file according to the Pass/Fail criteria.*

### ***Pre-conditions***

*a) The UCC executables is accessible.*

*b) Test directories must be writable*

*c) Matlab source file is accessible.*

### ***Post-conditions***

1. *UCC complete execution without error.*
2. *Output files, MATLAB\_outfile.csv and outfile\_cplx.csv, are created.*

### ***Dependencies***

*See the Pre-condition fro any dependencies.*

### ***Assumptions and Constraints***

*None.*

### ***Input Specifications***

*Matlab code file: UCC\_Matlab\_dataanalysis\_Keywords\_Test.m*

|  |
| --- |
| %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  %  % TEST IDENTIFICATION: <dataanalysis\_1>  %  %  % TEST DESCRIPTIONS:  % This file contains test cases for  % Matlab Data Analysis keywords UCC software.  %  % KEYWORD #OF USAGE  % ---------------------  %  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    brush 1  cumprod 1  cumsum 1  linkdata 1  prod 1  sort 2  sortrows 1  sum 1  corrcoef 1  cov 1  max 1  mean 1  median 1  min 1  mode 1  std 1  var 1  conv 2  conv2 1  convn 1  deconv 1  detrend 1  filter 1  filter2 1  interp1 1  interp2 1  interp3 1  interpn 1  mldivide \ 1, mrdivide / 1  polyfit 1  polyval 1  abs 1  angle 1  cplxpair 1  fft 1  fft2 1  fftn 1  fftshift 1  fftw 1  ifft 1  ifft2 1  ifftn 1  ifftshift 1  nextpow2 1  unwrap 1  cumtrapz 1  del2 1  diff 1  gradient 1  polyder 1  polyint 1  trapz 1  append (timeseries) 1  get (timeseries) 1  getdatasamples (timeseries) 1  getdatasamplesize (timeseries) 1  getqualitydesc (timeseries) 1  getsamples (timeseries) 1  isempty (timeseries) 1  length (timeseries) 1  plot (timeseries) 1  set (timeseries) 1  size (timeseries) 1  timeseries 9  tsdata.event 1  tstool 1  addsample (timeseries) 1  ctranspose (timeseries) 1  delsample (timeseries) 1  detrend (timeseries) 1  filter (timeseries) 1  getabstime (timeseries) 1  getinterpmethod (timeseries) 1  getsampleusingtime (timeseries) 1  idealfilter (timeseries) 1  resample (timeseries) 1  setabstime (timeseries) 1  setinterpmethod (timeseries) 1  synchronize (timeseries) 1  transpose (timeseries) 1  vertcat (timeseries) 1  addevent 1  delevent 1  gettsafteratevent 1  gettsafterevent 1  gettsatevent 1  gettsbeforeatevent 1  gettsbeforeevent 1  gettsbetweenevents 1  iqr (timeseries) 1  max (timeseries) 1  mean (timeseries) 1  median (timeseries) 1  min (timeseries) 1  std (timeseries) 1  sum (timeseries) 1  var (timeseries) 1  get (tscollection) 1  isempty (tscollection) 1  length (tscollection) 1  plot (timeseries) 1  set (tscollection) 1  size (tscollection) 1  tscollection 1  tstool 1  addsampletocollection 1  addts 1  delsamplefromcollection 1  getabstime (tscollection) 1  getsampleusingtime (tscollection) 1  gettimeseriesnames 1  horzcat (tscollection) 1  removets 1  resample (tscollection) 1  setabstime (tscollection) 1  settimeseriesnames 1  vertcat (tscollection) 1    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  % TEST CASE: <dataanalysis\_1>  %  % TEST DESCRIPTIONS:  % Simple usages of data analysis keywords  %  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  %this matlab program counts the number of occurences of data analysis keywords in Matlab    brush on;    cumprod(1:5);    cumsum(1:5);    linkdata on;    M = magic(3);  B = prod(M);    B = sort(B);    B = sortrows(B);    x = randn(30,4);  x(:,4) = sum(x,2);  [r,p] = corrcoef(x);    A = [-1 1 2 ; -2 3 1 ; 4 0 3];  v = diag(cov(A));    X = [2 8 4; 7 3 9];  max(X,[],1);    A = [1 2 3; 3 3 6; 4 6 8; 4 7 7];  mean(A);    median(A) ;    C = min(A);    C = mode(A);    C = std(A);    c = var(A);    w = conv(u,v);    A = rand(3);  B = rand(4);  C = conv2(A,B);    C = convn(A,B);    u = [1 2 3 4]  v = [10 20 30]  c = conv(u,v);  [q,r] = deconv(c,u);    sig = [0 1 -2 1 0 1 -2 1 0];  trend = [0 1 2 3 4 3 2 1 0];  x = sig+trend;  y = detrend(x,'linear',5);    Y = filter2(h,X);    x = 0:10;  y = sin(x);  xi = 0:.25:10;  yi = interp1(x,y,xi);    [X,Y] = meshgrid(-3:.25:3);  Z = peaks(X,Y);  [XI,YI] = meshgrid(-3:.125:3);  ZI = interp2(X,Y,Z,XI,YI);    [x,y,z,v] = flow(10);  [xi,yi,zi] = meshgrid(.1:.25:10, -3:.25:3, -3:.25:3);  vi = interp3(x,y,z,v,xi,yi,zi);    VI = interpn(V,ntimes);    A = magic(3);  b = [1;2;3]  x=A\b;  x = A/b;  x = (0: 0.1: 2.5)';  y = erf(x);  p = polyfit(x,y,6);    p = polyval(x,y,6);  abs(-5);    theta = angle(Z);    c = complex(a,b);    B = cplxpair(A);    Y = fft(X);    Y = fft2(X);    Y = fftn(X);    Y = fftshift(X);    t=0:.001:5;  x = sin(2\*pi\*50\*t)+sin(2\*pi\*120\*t);  y = x + 2\*randn(size(t));  Y = fftw(y,1458);    y = ifft(X);    Y = ifft2(X);    Y = ifftn(X);    ifftshift(X);    A = [1 2 3 4 5 9 519]  nextpow2(A);    Q = unwrap(P);    Y = [0 1 2; 3 4 5];  cumtrapz(Y,1);    L = del2(U);    v = -2:0.2:2;  [x,y] = meshgrid(v);  z = x .\* exp(-x.^2 - y.^2);  [px,py] = gradient(z,.2,.2);    X = sort(rand(1,101)\*pi);  Y = sin(X);  Z = trapz(X,Y);    ts1 = timeseries(rand(5,1),[1 2 3 4 5]);  ts2 = timeseries(rand(5,1),[6 7 8 9 10]);  ts3 = append(ts1, ts2)    ts1 = timeseries(rand(5,1),[1 2 3 4 5], 'Name', 'MyTimeseries');  get(ts1);    ts = timeseries(rand(5,1),[1 2 3 4 5]);  samples = getdatasamples(ts, [2 3]);    count\_ts = timeseries(count,[1:24],'Name','VehicleCount');  getdatasamplesize(count\_ts);    ts = timeseries([3; 4.2; 5; 6.1; 8], 1:5, [1; 0; 1; 0; 1]);  ts.QualityInfo.Description = {'good' 'bad'};  getqualitydesc(ts);    ts = timeseries(rand(5,1),[1 2 3 4 5]);  ts1 = getsamples(ts, ts.time([2 3]));    isempty(ts1);    length(ts1);    plot(ts1);    ts1 = timeseries(rand(5,1),[1 2 3 4 5]);  set(ts1, 'Name', 'mytimeseries');    size(ts1);    tsdata.event('AMCommute',8);    tstool();    ts = ts.addsample('Time',3,'Data',420);    ctranspose(ts1);    ts = timeseries(rand(5,1),[10 20 30 40 50]);  ts1 = delsample(ts,'Index', 1);    ts1 = detrend(ts, method);    filter\_count = filter(count1, b, a);    ts.TimeInfo.StartDate = '10/27/2005 07:05:36';  getabstime(ts)    getinterpmethod(ts1);    getsampleusingtime(ts1);    idealfilter(ts1);    resample(ts1);    setabstime(ts1);    setinterpmethod(ts1);    setuniformtime(ts1);    synchronize(ts1);    transpose(ts1);    vertcat(ts1);    addevent(count1, e1);    ts = delevent(ts,'test');    ts1 = gettsafteratevent(ts,event);    ts1 = gettsafterevent(ts,event);    ts1 = gettsatevent(ts,event);    ts1 = gettsbeforeatevent(ts,event);    tsc = addsampletocollection(tsc,'time',3.5,'acceleration',10,'speed',{5 1});    tsc = addts(tsc, ts2);    tsc = delsamplefromcollection(tsc,'Index',N);    tsc = tscollection(timeseries([3 6 8 0 10]));    getabstime(tsc);    tsc2=getsampleusingtime(tsc1,Time) ;    names = gettimeseriesnames(tsc);    tsc=horzcat(tsc1,tsc2);    tsc=removets(tsc,Name);    tsc=resample(tsc,Time);    tsc = setabstime(tsc,Times);    tsc=settimeseriesnames(tsc,old,new);    tsc=vertcat(tsc1,tsc2); |
|  |
|  |
|  |

### ***Expected Output Specifications***

*File MATLAB\_outfile.csv should contain the following information*

*a) Total Physical SLOC is xxx*

*b) Total Logical SLOC is xxx*

*File outfile\_cplx.csv should contain the following information*

*a) Only 1 Matlab source file is processed and file name is the correction input file according to the input specification.*

### ***Pass/Fail Criteria***

*If the results match those of the Expected Output Specification, the test has passed.* Otherwise the test has failed.

### ***Test Results***

*[Insert test result – test will be rerun before final delivery]*

## **Test Case <2>**

### ***Test Objectives***

The main objective of this test is to verify that the UCC is counting Matlab GUI development keywords.

### ***Test Description***

*Execute UCC with the input Matlab code file as followed*

-dir <source files directory> UCC\_Matlab\_GUI.m

*Verify UCC output file according to the Pass/Fail criteria.*

### ***Pre-conditions***

*a) The UCC executables is accessible.*

*b) Test directories must be writable*

*c) Matlab source file is accessible.*

### ***Post-conditions***

1. *UCC complete execution without error.*
2. *Output files, MATLAB\_outfile.csv and outfile\_cplx.csv, are created.*

### ***Dependencies***

*See the Pre-condition fro any dependencies.*

### ***Assumptions and Constraints***

*None.*

### ***Input Specifications***

*Matlab code file: UCC\_Matlab\_GUI.m*

|  |
| --- |
| %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  %  % TEST IDENTIFICATION: <GUI\_1>  %  %  % TEST DESCRIPTIONS:  % This file contains test cases for  % Matlab Data Analysis keywords UCC software.  %  % KEYWORD #OF USAGE  % ---------------------  %  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    dialog 1  errordlg 1  export2wsdlg 1  helpdlg 1  inputdlg 1  listdlg 1  msgbox 1  printdlg 1  printpreview 1  questdlg 1  uigetdir 1  uigetfile 1  uigetpref 1  uiopen 1  uiputfile 1  uisave 1  uisetcolor 1  uisetfont 1  waitbar 1  warndlg 1  guidata 1  guihandles 1  movegui 1  openfig 1  setpixelposition 1  getappdata 1  ginput 1  guidata 1  guide 1  inspect 1  isappdata 1  rmappdata 1  setappdata 1  uigetpref 1  uisetpref 1  waitfor 1  waitforbuttonpress 1  menu 1  uibuttongroup 1  uicontextmenu 1  uicontrol 1  uimenu 1  uipanel 1  uipushtool 1  uitable 1  uitoggletool 1  uitoolbar 1  findall 1  findfigs 1  findobj 1  gcbf 1  gcbo 1  align 1  getpixelposition 1  listfonts 1  movegui 1  selectmoveresize 1  setpixelposition 1  textwrap 1  uisetcolor 1  uisetfont 1  uistack 1  addpref 1  getpref 1  ispref 1  rmpref 1  setpref 1  uigetpref 1  uiresume 1  uisetpref 1  uiwait 1  waitfor 1  waitforbuttonpress 1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  % TEST CASE: <GUI\_1>  %  % TEST DESCRIPTIONS:  % Simple usages of GUI Development keywords  %  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  % Test case to check for the occurences of Matlab GUI development program.  out = dialog('WindowStyle', 'normal', 'Name', 'My Dialog');  errordlg('File not found','File Error');  A = randn(10,1);  checkLabels = {'Save sum of A to variable named:', 'Save mean of A to variable named:'};  varNames = {'sumA','meanA'};  items = {sum(A),mean(A)};  export2wsdlg(checkLabels,varNames,items,'Save Sums to Workspace');  helpdlg('Choose 10 points from the figure','Point Selection');  prompt = {'Enter matrix size:','Enter colormap name:'};  dlg\_title = 'Input for peaks function';  num\_lines = 1;  def = {'20','hsv'};  answer = inputdlg(prompt,dlg\_title,num\_lines,def);  d = dir;  str = {d.name};  [s,v] = listdlg('PromptString','Select a file:','SelectionMode','single','ListString',str);  h = msgbox("Hi");  printdlg();  printpreview();  button = questdlg('qstring','title');  uigetdir(matlabroot,'MATLAB Root Directory')  [FileName,PathName] = uigetfile('\*.m','Select the MATLAB code file');  pref\_value = uigetpref(group,pref,title,question,pref\_choices);  uiopen('filename');  [file,path] = uiputfile('animinit.m','Save file name');  h = 365;  g = 52;  uisave({'h','g'},'var1');  c = uisetcolor(...,'dialogTitle');  h = text(.5,.5,'Figure Annotation');  uisetfont(h,'Update Font');  h = waitbar(0,'Please wait...');  warndlg('Pressing OK will clear memory','!! Warning !!');  data = guidata(a);  handles = guihandle;  movegui(handle,'north');  openfig('filename.fig');  f = figure('Position',[300 300 300 200]);  p = uipanel('Position',[.2 .2 .6 .6]);  h1 = uicontrol(p,'Style','PushButton','Units','normalized', 'String','Push Button','Position',[.1 .1 .5 .2]);  pos1 = getpixelposition(h1);  setpixelposition(h1,pos1 + [10 10 25 25]);  values = getappdata(h);  [x,y] = ginput(4);  guide('filename.fig');  h = actxserver('excel.application');  inspect(h);  isappdata(h,name);  rmappdata(h,name);  setappdata(h,'name',value);  uigetpref();  uisetpref();  waitfor(h,'PropertyName');  k = waitforbuttonpress();  choice = menu('Choose a color','Red','Blue','Green');  h = uibuttongroup('visible','off','Position',[0 0 .2 1]);  hcmenu = uicontextmenu();  uicontrol('Style', 'popup','String', 'hsv|hot|cool|gray','Position', [20 340 100 50], 'Callback', @setmap);  f = uimenu('Label','Workspace');  hp = uipanel('Title','Main Panel','FontSize',12, 'BackgroundColor','white', 'Position',[.25 .1 .67 .67]);  hpt = uipushtool(ht,'CData',icon,'TooltipString','uipushtool','ClickedCallback','disp(''Hello World!'')');  f = figure('Position',[200 200 400 150]);  dat = rand(3);  cnames = {'X-Data','Y-Data','Z-Data'};  rnames = {'First','Second','Third'};  t = uitable('Parent',f,'Data',dat,'ColumnName',cnames,'RowName',rnames,'Position',[20 20 360 100]);  h = figure('ToolBar','none');  ht = uitoolbar(h);  a = rand(16,16,3);  htt = uitoggletool(ht,'CData',a,'TooltipString','Hello');  findall();  findfigs();  align(h, 'distribute','bottom');  fig = gcbf;  h = gcbo;  getpixelposition();  listfonts();  movegui();  selectmoveresize();  setpixelposition();  textwrap();  uisetcolor();  uisetfont();  addpref();  getpref();  uistack(h) ;  uiwait();  uiresume(h);  ispref('mytoolbox','version');  rmpref();  setpref();  uisetpref();  uiresume();  set(gca,'ButtonDownFcn','selectmoveresize'); |
|  |
|  |
|  |

### ***Expected Output Specifications***

*File MATLAB\_outfile.csv should contain the following information*

*a) Total Physical SLOC is xxx*

*b) Total Logical SLOC is xxx*

*File outfile\_cplx.csv should contain the following information*

*a) Only 1 Matlab source file is processed and file name is the correction input file according to the input specification.*

### ***Pass/Fail Criteria***

*If the results match those of the Expected Output Specification, the test has passed.* Otherwise the test has failed.

### ***Test Results***

*[Insert test result – test will be rerun before final delivery]*