Download Submission

Highlights from assert.m

assert(varargin) assert(testCondition)

View all files

assert.m

by Gerald Dalley
15 Feb 2006 (Updated 07 Nov 2006)

Assert function that expands like the C macro assert

assert(varargin)

```
function assert(varargin)
%assert(testCondition)
%   Evaluates the test condition in the caller's context.  If the result is
%   false, an error is thrown indicating that the test failed.
%   TESTCONDITION should be a string.
%
%assert(truthValue)
%assert(truthValue,errMsg)
%   If TRUTHVALUE is zero, an error is thrown.  The text of the error
%   contains ERRMSG if supplied.  This version of ASSERT was made to
%   promote compatibility with Kevin Murphy's Bayes Net Toolbox (i.e. he
%   uses this syntax).  TRUTHVALUE must be a logical or a number.
%
%EXAMPLES:
%   assert('1==2')      ==> assertion error generated
%   assert(1==2)        ==> assertion error generated (BNT syntax)
%   assert 1==2;        ==> assertion error generated
%   x=1; assert('x==1') ==> no error
%   x=1; assert x==1;   ==> no error
%   assert('{a==3')     ==> error generated by Matlab (bad assert code)
%
%by Gerald Dalley

% Make ourselves compatible with Kevin Murphy's Bayes Net toolbox.
if (ismember(nargin,[1 2]) && ...
    (islogical(varargin{1}) || isnumeric(varargin{1})))
  if nargin<2, msg = ''; else, msg = varargin{2}; end
  if ~varargin{1}
    error('assertion violated: %s', msg);
  end
  return
end

% Standard string-based usage
for i=1:nargin
    if (~ischar(varargin{i}))
        error('Type HELP ASSERT for usage');
    end
end

testCondition = join(' ', varargin);
if (~evalin('caller', testCondition))
    error(['ASSERT FAILED: ' testCondition]);
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function s = join(d,varargin)
% This JOIN function is an inlined version of Gerald Dalley's one posted at the
% Matlab Central website.  It is placed here as a convenience to users that
% have not downloaded it.

if (isempty(varargin)),
    s = '';
else
    if (iscell(varargin{1}))
        s = join(d, varargin{1}{:});
    else
        s = varargin{1};
    end

    for ss = 2:length(varargin)
        s = [s d join(d, varargin{ss})];
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function test
% Copy-and-paste the below code into the command window to test this
% function.

clear all;

worked = 0; try, assert('1==2'); catch, worked = 1; end
if (~worked), error('Test failed'); end

worked = 1; try, x=1; assert('x==1'); catch, worked = 0; end
if (~worked), error('Test failed'); end

worked = 0; try, assert(1==2); catch, worked = 1; end
if (~worked), error('Test failed'); end

worked = 0; try, assert('{a==3'); catch, worked = 1; end
if (~worked), error('Test failed'); end
```

Contact us