highlight.m

MATLAB, The Mathworks, Inc.

September 21, 2004

```
0001 function highlight(mfile,options,outfile)
0002 %HIGHLIGHT - Syntax Highlighting of Matlab M-files in HTML, LaTeX, RTF and XML.
0003 %  HIGHLIGHT(MFILE) takes an M-file MFILE in input and writes on disk an HTML
0004 %  file with the same basename ('foo.m' => 'foo.html') adding colored syntax
0005 %  highlighting on comment, string and Matlab keyword elements.
0006 %  HIGHLIGHT(MFILE,OPTIONS) with OPTIONS being string 'html', 'xhtml', 'tex',
0007 %  'rtf' or 'xml', allows to choose the output format between HTML, XHTML, LaTeX
0008 %  RTF or XML. The same rule is used to determine the name of the output file.
0009 %  HIGHLIGHT(MFILE,OPTIONS) allows to specify some options in a structure:
0010 %     options.type - Output file type
0011 %             [ {'html'} | 'tex' | 'rtf' | 'xml' | 'xhtml']
0012 %     options.tabs - Replace '\t' in source code by n white space
0013 %                            [ 0 ... {4}  ... n]
0014 %     options.linenb - Display line number in front of each line [ 0 | {1} ]
0015 %  HIGHLIGHT(MFILE,OPTIONS,OUTFILE) allows to specify the name of the output
0016 %  file. OUTFILE can also be a file handle. In that case, no header will be
0017 %  written, only the highlighted Matlab code will be sent to the OUTFILE stream.
0018
0019 %  Output file can be customized (font style, font color, font size, ...):
0020 %     o HTML: use CSS (Cascading Style Sheets) to define 'comment', 'string',
0021 %       'keyword', 'cont' and 'code' SPAN elements and 'mcode' PRE tag.
0022 %     o LaTeX: packages 'alltt' and 'color' are required, you can modify colors
0023 %       in defining colors 'string', 'comment' and 'keyword' using command
0024 %                       \definecolor{mycolor}{rgb}{a,b,c}
0025 %     o RTF: Colors are defined in the Color Table at the beginning of the
0026 %       document. See Rich Text Format Specifications for more details:
0027 %       <http://msdn.microsoft.com/library/en-us/dnrtfspec/html/rtfspec.asp>
0028 %     o XML: you will find the DTD of the resulting XML file in matlab.dtd
0029 %       You can then use XSL to transform your XML file in what you want.
0030 %       For example, mat2html.xsl transforms your XML file in HTML as it would
0031 %       be if you would have used highlight.m to to so.
0032 %       On Matlab 6.5, the command is:
0033 %               xslt highlight.xml mat2html.xsl highlight.html
0034 %       On Linux, using libxslt <http://xmlsoft.org/XSLT/>, the command is:
0035 %               xsltproc -o highlight.html mat2html.xsl highlight.xml
0036
0037 %  Copyright (C) 2003 Guillaume Flandin <Guillaume [at] artefact.tk>
0038 %  $Revision: 1.1 $Date: 2003/09/07 15:39:33 $
0039
0040 %  This program is free software; you can redistribute it and/or
0041 %  modify it under the terms of the GNU General Public License
0042 %  as published by the Free Software Foundation; either version 2
0043 %  of the License, or any later version.
0044 %
0045 %  This program is distributed in the hope that it will be useful,
```

```
0046 %  but WITHOUT ANY WARRANTY; without even the implied warranty of
0047 %  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
0048 %  GNU General Public License for more details.
0049 %
0050 %  You should have received a copy of the GNU General Public License
0051 %  along with this program; if not, write to the Free Software
0052 %  Foundation Inc, 59 Temple Pl. - Suite 330, Boston, MA 02111-1307, USA.
0053
0054 % Suggestions for improvement and fixes are always welcome, although no
0055 % guarantee is made whether and when they will be implemented.
0056 % Send requests to Guillaume [at] artefact.tk
0057
0058 %- Set up options
0059 error(nargchk(1,3,nargin));
0060
0061 opt = struct('type',   'html', ...
0062              'tabs',   4, ...
0063              'linenb', 1, ...
0064              'indent', 0);
0065 if nargin >= 2
0066     if isstruct(options)
0067         names = fieldnames(options);
0068         for i=1:length(names)
0069             opt = setfield(opt,names{i},getfield(options,names{i}));
0070         end
0071     elseif ischar(options)
0072         opt.type = options;
0073     else
0074         error('Bad input argument.');
0075     end
0076 end
0077 if strcmp(lower(opt.type),'latex'), opt.type = 'tex'; end
0078 if strcmp(lower(opt.type),'xml'), opt.linenb = 1; end
0079
0080 %- If no output filename is provided, one is chosen according to mfile
0081 if nargin < 3
0082     [pathstr, name, ext] = fileparts(mfile);
0083     outfile = fullfile(pathstr, [name, '.' lower(opt.type)]);
0084 end
0085
0086 %- If an output filename is provided, a standard header is created
0087 if ischar(outfile)
0088     outfid = fopen(outfile,'wt');
0089     if outfid == -1
0090         error(sprintf('Cannot open ',outfile));
0091     end
0092     feval([lower(opt.type) '_file_start'],outfid,mfile);
0093 %- Otherwise a file handle is provided
0094 else
0095     outfid = outfile;
0096 end
0097
0098 %- Open the Matlab mfile to be highlighted
0099 mfid = fopen(mfile,'rt');
0100 if mfid == -1
0101     error(sprintf('Cannot open ',mfile));
```

```
0102 end
0103
0104 %- Write the syntax highlighted mfile code in the output file
0105 write_highlighted_code(mfid,outfid,opt)
0106
0107 %- Close the Matlab mfile and potentially the output file
0108 fclose(mfid);
0109 if ischar(outfile),
0110     feval([lower(opt.type) '_file_end'],outfid);
0111     fclose(outfid);
0112 end
0113
0114 %=========================================================================================
0115 function write_highlighted_code(mfid,outfid,opt)
0116     matlabKeywords = getMatlabKeywords;
0117     mKeySort       = getMatlabKeywordsSorted;
0118     out_format     = feval([lower(opt.type) '_format']);
0119     strtok_delim   = sprintf(' \t\n\r(){}[]<>+-*~!|\\@&/.,:;="''%');
0120
0121     fprintf(outfid,out_format.pre_start);
0122     nbline = 1;
0123     nbblanks = 0;
0124     flagnextline = 0;
0125     while 1
0126         tline = fgetl(mfid);
0127         if ~ischar(tline), break, end
0128         tline = feval([lower(opt.type) '_special_char'],horztab(tline,opt.tabs));
0129         %- Display the line number at each line
0130         if opt.linenb
0131             fprintf(outfid,out_format.nb_line,nbline);
0132             nbline = nbline + 1;
0133         end
0134         %- Remove blanks at the beginning of the line
0135         if opt.indent
0136             tline = fliplr(deblank(fliplr(tline)));
0137         end
0138         nbblanks = nbblanks + flagnextline;
0139         flagnextline = 0;
0140         %- Split code into meaningful chunks
0141         splitc = splitcode(tline);
0142         newline = '';
0143         for i=1:length(splitc)
0144             if isempty(splitc{i})
0145             elseif splitc{i}(1) == ''''
0146                 newline = [newline sprintf(out_format.string,splitc{i})];
0147             elseif splitc{i}(1) == '%'
0148                 newline = [newline sprintf(out_format.comment,splitc{i})];
0149             elseif ~isempty(strmatch('...',splitc{i}))
0150                 newline = [newline sprintf(out_format.cont,'...')];
0151                 if ~isempty(splitc{i}(4:end))
0152                     newline = [newline sprintf(out_format.comment,splitc{i}(4:end))];
0153                 end
0154             else
0155                 %- Look for Matlab keywords
0156                 r = splitc{i};
0157                 stringcode = '';
```

```
0158                while 1
0159                    [t,r,q] = strtok(r,strtok_delim);
0160                    stringcode = [stringcode q];
0161                    if isempty(t), break, end;
0162                    isNextIncr  = any(strcmp(t,mKeySort.nextincr));
0163                    isNextIncr2 = any(strcmp(t,mKeySort.nextincr2));
0164                    isCurrDecr  = any(strcmp(t,mKeySort.currdecr));
0165                    isNextDecr  = any(strcmp(t,mKeySort.nextdecr));
0166                    isOther     = any(strcmp(t,mKeySort.other));
0167                    if isNextDecr % if strcmp(t,'end')
0168                        % 'end' is keyword or array subscript ?
0169                        rr = fliplr(deblank(fliplr(r)));
0170                        icomma = strmatch(',',rr);
0171                        isemicolon = strmatch(';',rr);
0172                        if ~(isempty(rr) | ~isempty([icomma isemicolon]))
0173                            isNextDecr = 0;
0174                        else
0175                            nbblanks = nbblanks - 1;
0176                            flagnextline = flagnextline + 1;
0177                        end
0178                        % TODO % false detection of a([end,1])
0179                    end
0180                    if isNextIncr, flagnextline = flagnextline + 1; end
0181                    if isNextIncr2, flagnextline = flagnextline + 2; end
0182                    if isNextDecr, flagnextline = flagnextline - 1; end
0183                    if isCurrDecr, nbblanks = nbblanks - 1; end
0184                    % if any(strcmp(t,matlabKeywords))
0185                    if isNextIncr | isNextIncr2 |isCurrDecr | isNextDecr | isOther
0186                        if ~isempty(stringcode)
0187                            newline = [newline sprintf(out_format.code,stringcode)];
0188                            stringcode = '';
0189                        end
0190                        newline = [newline sprintf(out_format.keyword,t)];
0191                    else
0192                        stringcode = [stringcode t];
0193                    end
0194                end
0195                if ~isempty(stringcode)
0196                    newline = [newline sprintf(out_format.code,stringcode)];
0197                end
0198            end
0199        end
0200        if ~opt.indent, nbblanks = 0; end
0201        %- Print the syntax highlighted line
0202        fprintf(outfid,out_format.line,[blanks(nbblanks*opt.tabs) newline]);
0203    end
0204    fprintf(outfid,out_format.pre_end);
0205
0206 %================================================================================
0207 %                                  HTML FORMAT                                   %
0208 %================================================================================
0209 function html_file_start(fid,mfile)
0210     fprintf(fid,[ ...
0211     '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"\n' ...
0212     '\t"http://www.w3.org/TR/REC-html40/loose.dtd">\n' ...
0213     '<html>\n<head>\n<title>%s</title>\n' ...
```

```
0214     '<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">\n' ...
0215     '<meta name="generator" content="highlight.m &copy; 2003 Guillaume Flandin">\n' ...
0216     '<style type="text/css">\n' ...
0217     '  .comment {color: #228B22;}\n' ...
0218     '  .string {color: #B20000;}\n' ...
0219     '  .keyword, .cont {color: #0000FF;}\n' ...
0220     '  .cont {text-decoration: underline;}\n' ...
0221     '  .code {color: #000000;}\n' ...
0222     '</style>\n' ...
0223     '</head>\n<body>\n'],mfile);
0224
0225 %--------------------------------------------------------------------------
0226 function html_file_end(fid)
0227     fprintf(fid,'\n</body>\n</html>');
0228
0229 %--------------------------------------------------------------------------
0230 function format = html_format
0231     format.string    = '<span class="string">%s</span>';
0232     format.comment   = '<span class="comment">%s</span>';
0233     format.code      = '%s'; %'<span class="code">%s</span>';
0234     format.keyword   = '<span class="keyword">%s</span>';
0235     format.cont      = '<span class="cont">%s</span>';
0236     format.pre_start = '<pre class="mcode">';
0237     format.pre_end   = '</pre>\n';
0238     format.nb_line   = '%04d ';
0239     format.line      = '%s\n';
0240
0241 %--------------------------------------------------------------------------
0242 function str = html_special_char(str)
0243     %- See http://www.w3.org/TR/html4/charset.html#h-5.3.2
0244     str = strrep(str,'&','&amp;');
0245     str = strrep(str,'<','&lt;');
0246     str = strrep(str,'>','&gt;');
0247     str = strrep(str,'"','&quot;');
0248
0249 %==========================================================================
0250 %                                  XHTML FORMAT                           %
0251 %==========================================================================
0252 function xhtml_file_start(fid,mfile)
0253     fprintf(fid,[ ...
0254     '<?xml version="1.0"?>\n' ...
0255     '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" ' ...
0256     '"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">\n' ...
0257     '<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">\n' ...
0258     '<head>\n<title>%s</title>\n' ...
0259     '<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />\n' ...
0260     '<meta name="generator" content="highlight.m &copy; 2003 Guillaume Flandin" />\n' ...
0261     '<style type="text/css">\n' ...
0262     '  .comment {color: #228B22;}\n' ...
0263     '  .string {color: #B20000;}\n' ...
0264     '  .keyword, .cont {color: #0000FF;}\n' ...
0265     '  .cont {text-decoration: underline;}\n' ...
0266     '  .code {color: #000000;}\n' ...
0267     '</style>\n' ...
0268     '</head>\n<body>\n'],mfile);
0269
```

```
0270 %--------------------------------------------------------------------------
0271 function xhtml_file_end(fid)
0272     fprintf(fid,'\n</body>\n</html>');
0273
0274 %--------------------------------------------------------------------------
0275 function format = xhtml_format
0276     format.string    = '<span class="string">%s</span>';
0277     format.comment   = '<span class="comment">%s</span>';
0278     format.code      = '%s'; %'<span class="code">%s</span>';
0279     format.keyword   = '<span class="keyword">%s</span>';
0280     format.cont      = '<span class="cont">%s</span>';
0281     format.pre_start = '<pre class="mcode">';
0282     format.pre_end   = '</pre>\n';
0283     format.nb_line   = '%04d ';
0284     format.line      = '%s\n';
0285
0286 %--------------------------------------------------------------------------
0287 function str = xhtml_special_char(str)
0288     %- See http://www.w3.org/TR/html4/charset.html#h-5.3.2
0289     str = strrep(str,'&','&amp;');
0290     str = strrep(str,'<','&lt;');
0291     str = strrep(str,'>','&gt;');
0292     str = strrep(str,'"','&quot;');
0293
0294
0295 %==========================================================================
0296 %                                  XML FORMAT                              %
0297 %==========================================================================
0298 function xml_file_start(fid,mfile)
0299     fprintf(fid,[ ...
0300     '<?xml version="1.0" encoding="iso-8859-1" ?>\n' ...
0301     '<!DOCTYPE mfile SYSTEM "matlab.dtd">\n' ...
0302     '<mfile name="%s">\n'],mfile);
0303
0304 %--------------------------------------------------------------------------
0305 function xml_file_end(fid)
0306     fprintf(fid,'</mfile>');
0307
0308 %--------------------------------------------------------------------------
0309 function format = xml_format
0310     format.string    = '<string>%s</string>';
0311     format.comment   = '<comment>%s</comment>';
0312     format.code      = '%s'; %'<code>%s</code>';
0313     format.keyword   = '<keyword>%s</keyword>';
0314     format.cont      = '<cont>%s</cont>';
0315     format.pre_start = '';
0316     format.pre_end   = '';
0317     format.nb_line   = '<line nb="%04d ">';
0318     format.line      = '%s</line>\n';
0319
0320 %--------------------------------------------------------------------------
0321 function str = xml_special_char(str)
0322     %- See http://www.w3.org/TR/REC-xml#sec-predefined-ent
0323     str = strrep(str,'&','&amp;');
0324     str = strrep(str,'<','&lt;');
0325     str = strrep(str,'>','&gt;');
```

```
0326     str = strrep(str,'"','&quot;');
0327     %str = strrep(str,'''','&apos;');
0328
0329 %==========================================================================
0330 %                                  LaTeX FORMAT                           %
0331 %==========================================================================
0332 function tex_file_start(fid,mfile)
0333     fprintf(fid,['\\documentclass[a4paper,10pt]{article}\n' ...
0334                  '    \\usepackage{alltt}\n' ...
0335                  '    \\usepackage{color}\n' ...
0336                  '    \\usepackage{fullpage}\n' ...
0337                  '    \\definecolor{string}{rgb}{0.7,0.0,0.0}\n' ...
0338                  '    \\definecolor{comment}{rgb}{0.13,0.54,0.13}\n' ...
0339                  '    \\definecolor{keyword}{rgb}{0.0,0.0,1.0}\n' ...
0340                  '    \\title{%s}\n' ...
0341                  '    \\author{\\textsc{Matlab}, The Mathworks, Inc.}\n' ...
0342                  '\\begin{document}\n' ...
0343                  '\\maketitle\n'],mfile);
0344
0345 %--------------------------------------------------------------------------
0346 function tex_file_end(fid)
0347     fprintf(fid,'\\end{document}');
0348
0349 %--------------------------------------------------------------------------
0350 function format = tex_format
0351     format.string    = '\\textcolor{string}{%s}';
0352     format.comment   = 'textcolor{comment}{%s}'; % '%' has been replaced by '\%'
0353     format.code      = '%s';
0354     format.keyword   = '\\textcolor{keyword}{%s}';
0355     format.cont      = '\\textcolor{keyword}{\\underline{%s}}';
0356     format.pre_start = '\\begin{alltt}\n';
0357     format.pre_end   = '\\end{alltt}\n';
0358     format.nb_line   = '%04d ';
0359     format.line      = '%s\n';
0360
0361 %--------------------------------------------------------------------------
0362 function str = tex_special_char(str)
0363     % Special characters: # $ % & ~ _ ^ \ { }
0364     str = strrep(str,'\','\(\backslash\)'); % $\backslash$ or \textbackslash or \verb+\+
0365     str = strrep(str,'#','\#');
0366     str = strrep(str,'$','\$');
0367     str = strrep(str,'%','\%');
0368     str = strrep(str,'&','\&');
0369     str = strrep(str,'_','\_');
0370     str = strrep(str,'{','\{');
0371     str = strrep(str,'}','\}');
0372     str = strrep(str,'^','\^{}');
0373     str = strrep(str,'~','\~{}'); % or \textasciitilde
0374
0375 %==========================================================================
0376 %                                  RTF FORMAT                             %
0377 %==========================================================================
0378 function rtf_file_start(fid,mfile)
0379     fprintf(fid,['{\\rtf1\\ansi\n\n' ...
0380                  '{\\fonttbl{\\f0\\fmodern\\fcharset0\\fprq1 Courier New;}}\n\n' ...
0381                  '{\\colortbl;\n' ...
```

```
0382                 '\\red0\\green0\\blue0;\n' ...
0383                 '\\red0\\green0\\blue255;\n' ...
0384                 '\\red0\\green255\\blue255;\n' ...
0385                 '\\red0\\green255\\blue0;\n' ...
0386                 '\\red255\\green0\\blue255;\n' ...
0387                 '\\red255\\green0\\blue0;}\n\n' ...
0388                 '{\\info{\\title %s}\n' ...
0389                 '{\\author HighLight.m Copyright 2003}\n' ...
0390                 '{\\creatim\\yr%s\\mo%s\\dy%s}' ...
0391                 '{\\*\\manager Guillaume Flandin}\n' ...
0392                 '{\\*\\company Artefact.tk}\n' ...
0393                 '{\\*\\hlinkbase http://www.artefact.tk/software/' ...textcolorcomment
0394                 'matlab/highlight/}}\n\n'],mfile,...
0395                 datestr(date,'yyyy'),datestr(date,'mm'),datestr(date,'dd'));
0396
0397 %------------------------------------------------------------------------------
0398 function rtf_file_end(fid)
0399     fprintf(fid,'}');
0400
0401 %------------------------------------------------------------------------------
0402 function format = rtf_format
0403     format.string    = '{\\cf6 %s}';
0404     format.comment   = '{\\cf4 %s}';
0405     format.code      = '{%s}';
0406     format.keyword   = '{\\cf2 %s}';
0407     format.cont      = '{\\cf2 %s}';
0408     format.pre_start = '{\\f0\\fs16{';
0409     format.pre_end   = '}}';
0410     format.nb_line   = '{%04d }';
0411     format.line      = '%s\n\\par ';
0412
0413 %------------------------------------------------------------------------------
0414 function str = rtf_special_char(str)
0415     str = strrep(str,'\','\\');
0416     str = strrep(str,'{','\{');
0417     str = strrep(str,'}','\}');
0418
0419 %==============================================================================
0420 %                                 MATLAB KEYWORDS                             %
0421 %==============================================================================
0422 function matlabKeywords = getMatlabKeywords
0423     %matlabKeywords = iskeyword; % Matlab R13
0424     matlabKeywords = {'break', 'case', 'catch', 'continue', 'elseif', 'else',...
0425                       'end', 'for', 'function', 'global', 'if', 'otherwise', ...
0426                       'persistent', 'return', 'switch', 'try', 'while'};
0427
0428 %------------------------------------------------------------------------------
0429 function mKeySort = getMatlabKeywordsSorted
0430     mKeySort.nextincr = {'for', 'while', 'if', 'else', 'elseif', ...
0431                          'case', 'otherwise', 'try', 'catch'};
0432     mKeySort.nextincr2 = {'switch'};
0433     mKeySort.currdecr = {'else', 'elseif', 'case', 'otherwise', 'catch'};
0434     mKeySort.nextdecr = {'end'};
0435     mKeySort.other    = {'break', 'continue', 'function', 'global', ...
0436                          'persistent', 'return'};
0437
```

```
0438 %==========================================================================
0439 %                              HANDLE TAB CHARACTER                        %
0440 %==========================================================================
0441 function str = horztab(str,n)
0442     %- For browsers, the horizontal tab character is the smallest non-zero
0443     %- number of spaces necessary to line characters up along tab stops that are
0444     %- every 8 characters: behaviour obtained when n = 0.
0445     if n > 0
0446         str = strrep(str,sprintf('\t'),blanks(n));
0447     end
0448
0449 %==========================================================================
0450 %                              MATLAB CODE PARSER                          %
0451 %==========================================================================
0452 function splitc = splitcode(code)
0453     %Split a line of  Matlab code in string, comment and other
0454
0455     %- Label quotes in {'transpose', 'beginstring', 'midstring', 'endstring'}
0456     iquote = findstr(code,'''');
0457     quotetransp = [double('_''.)}]') ...
0458                    double('A'):double('Z') ...
0459                    double('0'):double('9') ...
0460                    double('a'):double('z')];
0461     flagstring = 0;
0462     flagdoublequote = 0;
0463     jquote = [];
0464     for i=1:length(iquote)
0465         if ~flagstring
0466             if iquote(i) > 1 & any(quotetransp == double(code(iquote(i)-1)))
0467                 % => 'transpose';
0468             else
0469                 % => 'beginstring';
0470                 jquote(size(jquote,1)+1,:) = [iquote(i) length(code)];
0471                 flagstring = 1;
0472             end
0473         else % if flagstring
0474             if flagdoublequote | ...
0475                (iquote(i) < length(code) & strcmp(code(iquote(i)+1),''''))
0476                 % => 'midstring';
0477                 flagdoublequote = ~flagdoublequote;
0478             else
0479                 % => 'endstring';
0480                 jquote(size(jquote,1),2) = iquote(i);
0481                 flagstring = 0;
0482             end
0483         end
0484     end
0485
0486     %- Find if a portion of code is a comment
0487     ipercent = findstr(code,'%');
0488     jpercent = [];
0489     for i=1:length(ipercent)
0490         if isempty(jquote) | ...
0491            ~any((ipercent(i) > jquote(:,1)) & (ipercent(i) < jquote(:,2)))
0492             jpercent = [ipercent(i) length(code)];
0493             break;
```

```
0494         end
0495     end
0496
0497     %- Find continuation punctuation '...'
0498     icont = findstr(code,'...');
0499     for i=1:length(icont)
0500         if (isempty(jquote) | ...
0501             ~any((icont(i) > jquote(:,1)) & (icont(i) < jquote(:,2)))) & ...
0502             (isempty(jpercent) | ...
0503             icont(i) < jpercent(1))
0504             jpercent = [icont(i) length(code)];
0505             break;
0506         end
0507     end
0508
0509     %- Remove strings inside comments
0510     if ~isempty(jpercent) & ~isempty(jquote)
0511         jquote(find(jquote(:,1) > jpercent(1)),:) = [];
0512     end
0513
0514     %- Split code in a cell array of strings
0515     icode = [jquote ; jpercent];
0516     splitc = {};
0517     if isempty(icode)
0518         splitc{1} = code;
0519     elseif icode(1,1) > 1
0520         splitc{1} = code(1:icode(1,1)-1);
0521     end
0522     for i=1:size(icode,1)
0523         splitc{end+1} = code(icode(i,1):icode(i,2));
0524         if i < size(icode,1) & icode(i+1,1) > icode(i,2) + 1
0525             splitc{end+1} = code((icode(i,2)+1):(icode(i+1,1)-1));
0526         elseif i == size(icode,1) & icode(i,2) < length(code)
0527             splitc{end+1} = code(icode(i,2)+1:end);
0528         end
0529     end
0530
0531 %=========================================================================
0532 %                          MODIFIED VERSION OF STRTOK                     %
0533 %=========================================================================
0534 function [token, remainder, quotient] = strtok(string, delimiters)
0535 % Modified version of STRTOK to also return the quotient
0536 % string = [quotient token remainder]
0537 %STRTOK Find token in string.
0538 %   STRTOK(S) returns the first token in the string S delimited
0539 %   by "white space".   Any leading white space characters are ignored.
0540 %
0541 %   STRTOK(S,D) returns the first token delimited by one of the
0542 %   characters in D.  Any leading delimiter characters are ignored.
0543 %
0544 %   [T,R] = STRTOK(...) also returns the remainder of the original
0545 %   string.
0546 %   If the token is not found in S then R is an empty string and T
0547 %   is same as S.
0548 %
0549 %   Copyright 1984-2002 The MathWorks, Inc.
```

```
0550 %   $Revision: 5.14 $  $Date: 2002/04/09 00:33:38 $
0551
0552 token = []; remainder = []; quotient = string;
0553
0554 len = length(string);
0555 if len == 0
0556     return
0557 end
0558
0559 if (nargin == 1)
0560     delimiters = [9:13 32]; % White space characters
0561 end
0562
0563 i = 1;
0564 while (any(string(i) == delimiters))
0565     i = i + 1;
0566     if (i > len), return, end
0567 end
0568 start = i;
0569 while (~any(string(i) == delimiters))
0570     i = i + 1;
0571     if (i > len), break, end
0572 end
0573 finish = i - 1;
0574
0575 token = string(start:finish);
0576
0577 if (nargout >= 2)
0578     remainder = string(finish + 1:length(string));
0579 end
0580
0581 if (nargout == 3 & start > 1)
0582     quotient = string(1:start-1);
0583 else
0584     quotient = [];
0585 end
```