highlight.m

Matlab, The Mathworks, Inc.

September 21, 2004

```
function highlight(mfile,options,outfile)
%HIGHLIGHT - Syntax Highlighting of Matlab M-files in HTML, LaTeX, RTF and XML.
%  HIGHLIGHT(MFILE) takes an M-file MFILE in input and writes on disk an HTML
%  file with the same basename ('foo.m' => 'foo.html') adding colored syntax
%  highlighting on comment, string and Matlab keyword elements.
%  HIGHLIGHT(MFILE,OPTIONS) with OPTIONS being string 'html', 'xhtml', 'tex',
%  'rtf' or 'xml', allows to choose the output format between HTML, XHTML, LaTeX
%  RTF or XML. The same rule is used to determine the name of the output file.
%  HIGHLIGHT(MFILE,OPTIONS) allows to specify some options in a structure:
%     options.type - Output file type
%             [ {'html'} | 'tex' | 'rtf' | 'xml' | 'xhtml']
%     options.tabs - Replace '\t' in source code by n white space
%                            [ 0 ... {4}  ... n]
%     options.linenb - Display line number in front of each line [ 0 | {1} ]
%  HIGHLIGHT(MFILE,OPTIONS,OUTFILE) allows to specify the name of the output
%  file. OUTFILE can also be a file handle. In that case, no header will be
%  written, only the highlighted Matlab code will be sent to the OUTFILE stream.

%  Output file can be customized (font style, font color, font size, ...):
%     o HTML: use CSS (Cascading Style Sheets) to define 'comment', 'string',
%       'keyword', 'cont' and 'code' SPAN elements and 'mcode' PRE tag.
%     o LaTeX: packages 'alltt' and 'color' are required, you can modify colors
%       in defining colors 'string', 'comment' and 'keyword' using command
%                      \definecolor{mycolor}{rgb}{a,b,c}
%     o RTF: Colors are defined in the Color Table at the beginning of the
%       document. See Rich Text Format Specifications for more details:
%       <http://msdn.microsoft.com/library/en-us/dnrtfspec/html/rtfspec.asp>
%     o XML: you will find the DTD of the resulting XML file in matlab.dtd
%       You can then use XSL to transform your XML file in what you want.
%       For example, mat2html.xsl transforms your XML file in HTML as it would
%       be if you would have used highlight.m to to so.
%       On Matlab 6.5, the command is:
%               xslt highlight.xml mat2html.xsl highlight.html
%       On Linux, using libxslt <http://xmlsoft.org/XSLT/>, the command is:
%               xsltproc -o highlight.html mat2html.xsl highlight.xml

%  Copyright (C) 2003 Guillaume Flandin <Guillaume [at] artefact.tk>
%  $Revision: 1.1 $Date: 2003/09/07 15:39:33 $

%  This program is free software; you can redistribute it and/or
%  modify it under the terms of the GNU General Public License
%  as published by the Free Software Foundation; either version 2
%  of the License, or any later version.
%
%  This program is distributed in the hope that it will be useful,
```

```
%  but WITHOUT ANY WARRANTY; without even the implied warranty of
%  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
%  GNU General Public License for more details.
%
%  You should have received a copy of the GNU General Public License
%  along with this program; if not, write to the Free Software
%  Foundation Inc, 59 Temple Pl. - Suite 330, Boston, MA 02111-1307, USA.

% Suggestions for improvement and fixes are always welcome, although no
% guarantee is made whether and when they will be implemented.
% Send requests to Guillaume [at] artefact.tk

%- Set up options
error(nargchk(1,3,nargin));

opt = struct('type',   'html', ...
             'tabs',   4, ...
             'linenb', 1, ...
             'indent', 0);
if nargin >= 2
    if isstruct(options)
        names = fieldnames(options);
        for i=1:length(names)
            opt = setfield(opt,names{i},getfield(options,names{i}));
        end
    elseif ischar(options)
        opt.type = options;
    else
        error('Bad input argument.');
    end
end
if strcmp(lower(opt.type),'latex'), opt.type = 'tex'; end
if strcmp(lower(opt.type),'xml'), opt.linenb = 1; end

%- If no output filename is provided, one is chosen according to mfile
if nargin < 3
    [pathstr, name, ext] = fileparts(mfile);
    outfile = fullfile(pathstr, [name, '.' lower(opt.type)]);
end

%- If an output filename is provided, a standard header is created
if ischar(outfile)
    outfid = fopen(outfile,'wt');
    if outfid == -1
        error(sprintf('Cannot open ',outfile));
    end
    feval([lower(opt.type) '_file_start'],outfid,mfile);
%- Otherwise a file handle is provided
else
    outfid = outfile;
end

%- Open the Matlab mfile to be highlighted
mfid = fopen(mfile,'rt');
if mfid == -1
    error(sprintf('Cannot open ',mfile));
```

```
end

%- Write the syntax highlighted mfile code in the output file
write_highlighted_code(mfid,outfid,opt)

%- Close the Matlab mfile and potentially the output file
fclose(mfid);
if ischar(outfile),
    feval([lower(opt.type) '_file_end'],outfid);
    fclose(outfid);
end

%===============================================================================
function write_highlighted_code(mfid,outfid,opt)
    matlabKeywords = getMatlabKeywords;
    mKeySort       = getMatlabKeywordsSorted;
    out_format     = feval([lower(opt.type) '_format']);
    strtok_delim   = sprintf(' \t\n\r(){}[]<>+-*~!|\\@&/.,:;="''%');

    fprintf(outfid,out_format.pre_start);
    nbline = 1;
    nbblanks = 0;
    flagnextline = 0;
    while 1
        tline = fgetl(mfid);
        if ~ischar(tline), break, end
        tline = feval([lower(opt.type) '_special_char'],horztab(tline,opt.tabs));
        %- Display the line number at each line
        if opt.linenb
            fprintf(outfid,out_format.nb_line,nbline);
            nbline = nbline + 1;
        end
        %- Remove blanks at the beginning of the line
        if opt.indent
            tline = fliplr(deblank(fliplr(tline)));
        end
        nbblanks = nbblanks + flagnextline;
        flagnextline = 0;
        %- Split code into meaningful chunks
        splitc = splitcode(tline);
        newline = '';
        for i=1:length(splitc)
            if isempty(splitc{i})
            elseif splitc{i}(1) == ''''
                newline = [newline sprintf(out_format.string,splitc{i})];
            elseif splitc{i}(1) == '%'
                newline = [newline sprintf(out_format.comment,splitc{i})];
            elseif ~isempty(strmatch('...',splitc{i}))
                newline = [newline sprintf(out_format.cont,'...')];
                if ~isempty(splitc{i}(4:end))
                    newline = [newline sprintf(out_format.comment,splitc{i}(4:end))];
                end
            else
                %- Look for Matlab keywords
                r = splitc{i};
                stringcode = '';
```

```
                while 1
                    [t,r,q] = strtok(r,strtok_delim);
                    stringcode = [stringcode q];
                    if isempty(t), break, end;
                    isNextIncr  = any(strcmp(t,mKeySort.nextincr));
                    isNextIncr2 = any(strcmp(t,mKeySort.nextincr2));
                    isCurrDecr  = any(strcmp(t,mKeySort.currdecr));
                    isNextDecr  = any(strcmp(t,mKeySort.nextdecr));
                    isOther     = any(strcmp(t,mKeySort.other));
                    if isNextDecr % if strcmp(t,'end')
                        % 'end' is keyword or array subscript ?
                        rr = fliplr(deblank(fliplr(r)));
                        icomma = strmatch(',',rr);
                        isemicolon = strmatch(';',rr);
                        if ~(isempty(rr) | ~isempty([icomma isemicolon]))
                            isNextDecr = 0;
                        else
                            nbblanks = nbblanks - 1;
                            flagnextline = flagnextline + 1;
                        end
                        % TODO % false detection of a([end,1])
                    end
                    if isNextIncr, flagnextline = flagnextline + 1; end
                    if isNextIncr2, flagnextline = flagnextline + 2; end
                    if isNextDecr, flagnextline = flagnextline - 1; end
                    if isCurrDecr, nbblanks = nbblanks - 1; end
                    % if any(strcmp(t,matlabKeywords))
                    if isNextIncr | isNextIncr2 |isCurrDecr | isNextDecr | isOther
                        if ~isempty(stringcode)
                            newline = [newline sprintf(out_format.code,stringcode)];
                            stringcode = '';
                        end
                        newline = [newline sprintf(out_format.keyword,t)];
                    else
                        stringcode = [stringcode t];
                    end
                end
                if ~isempty(stringcode)
                    newline = [newline sprintf(out_format.code,stringcode)];
                end
            end
        end
        if ~opt.indent, nbblanks = 0; end
        %- Print the syntax highlighted line
        fprintf(outfid,out_format.line,[blanks(nbblanks*opt.tabs) newline]);
    end
    fprintf(outfid,out_format.pre_end);

%===============================================================================
%                                  HTML FORMAT                                 %
%===============================================================================
function html_file_start(fid,mfile)
    fprintf(fid,[ ...
    '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"\n' ...
    '\t"http://www.w3.org/TR/REC-html40/loose.dtd">\n' ...
    '<html>\n<head>\n<title>%s</title>\n' ...
```

```
    '<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">\n' ...
    '<meta name="generator" content="highlight.m &copy; 2003 Guillaume Flandin">\n' ...
    '<style type="text/css">\n' ...
    '   .comment {color: #228B22;}\n' ...
    '   .string {color: #B20000;}\n' ...
    '   .keyword, .cont {color: #0000FF;}\n' ...
    '   .cont {text-decoration: underline;}\n' ...
    '   .code {color: #000000;}\n' ...
    '</style>\n' ...
    '</head>\n<body>\n'],mfile);

%-------------------------------------------------------------------------------
function html_file_end(fid)
    fprintf(fid,'\n</body>\n</html>');

%-------------------------------------------------------------------------------
function format = html_format
    format.string     = '<span class="string">%s</span>';
    format.comment    = '<span class="comment">%s</span>';
    format.code       = '%s'; %'<span class="code">%s</span>';
    format.keyword    = '<span class="keyword">%s</span>';
    format.cont       = '<span class="cont">%s</span>';
    format.pre_start = '<pre class="mcode">';
    format.pre_end    = '</pre>\n';
    format.nb_line    = '%04d ';
    format.line       = '%s\n';

%-------------------------------------------------------------------------------
function str = html_special_char(str)
    %- See http://www.w3.org/TR/html4/charset.html#h-5.3.2
    str = strrep(str,'&','&amp;');
    str = strrep(str,'<','&lt;');
    str = strrep(str,'>','&gt;');
    str = strrep(str,'"','&quot;');

%===============================================================================
%                                  XHTML FORMAT                                %
%===============================================================================
function xhtml_file_start(fid,mfile)
    fprintf(fid,[ ...
    '<?xml version="1.0"?>\n' ...
    '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" ' ...
    '"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">\n' ...
    '<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">\n' ...
    '<head>\n<title>%s</title>\n' ...
    '<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />\n' ...
    '<meta name="generator" content="highlight.m &copy; 2003 Guillaume Flandin" />\n' ...
    '<style type="text/css">\n' ...
    '   .comment {color: #228B22;}\n' ...
    '   .string {color: #B20000;}\n' ...
    '   .keyword, .cont {color: #0000FF;}\n' ...
    '   .cont {text-decoration: underline;}\n' ...
    '   .code {color: #000000;}\n' ...
    '</style>\n' ...
    '</head>\n<body>\n'],mfile);
```

```
%----------------------------------------------------------------------------
function xhtml_file_end(fid)
    fprintf(fid,'\n</body>\n</html>');

%----------------------------------------------------------------------------
function format = xhtml_format
    format.string    = '<span class="string">%s</span>';
    format.comment   = '<span class="comment">%s</span>';
    format.code      = '%s'; %'<span class="code">%s</span>';
    format.keyword   = '<span class="keyword">%s</span>';
    format.cont      = '<span class="cont">%s</span>';
    format.pre_start = '<pre class="mcode">';
    format.pre_end   = '</pre>\n';
    format.nb_line   = '%04d ';
    format.line      = '%s\n';

%----------------------------------------------------------------------------
function str = xhtml_special_char(str)
    %- See http://www.w3.org/TR/html4/charset.html#h-5.3.2
    str = strrep(str,'&','&amp;');
    str = strrep(str,'<','&lt;');
    str = strrep(str,'>','&gt;');
    str = strrep(str,'"','&quot;');


%============================================================================
%                                 XML FORMAT                                 %
%============================================================================
function xml_file_start(fid,mfile)
    fprintf(fid,[ ...
    '<?xml version="1.0" encoding="iso-8859-1" ?>\n' ...
    '<!DOCTYPE mfile SYSTEM "matlab.dtd">\n' ...
    '<mfile name="%s">\n'],mfile);

%----------------------------------------------------------------------------
function xml_file_end(fid)
    fprintf(fid,'</mfile>');

%----------------------------------------------------------------------------
function format = xml_format
    format.string    = '<string>%s</string>';
    format.comment   = '<comment>%s</comment>';
    format.code      = '%s'; %'<code>%s</code>';
    format.keyword   = '<keyword>%s</keyword>';
    format.cont      = '<cont>%s</cont>';
    format.pre_start = '';
    format.pre_end   = '';
    format.nb_line   = '<line nb="%04d ">';
    format.line      = '%s</line>\n';

%----------------------------------------------------------------------------
function str = xml_special_char(str)
    %- See http://www.w3.org/TR/REC-xml#sec-predefined-ent
    str = strrep(str,'&','&amp;');
    str = strrep(str,'<','&lt;');
    str = strrep(str,'>','&gt;');
```

```
    str = strrep(str,'"','&quot;');
    %str = strrep(str,'''','&apos;');

%==============================================================================
%                                  LaTeX FORMAT                               %
%==============================================================================
function tex_file_start(fid,mfile)
    fprintf(fid,['\\documentclass[a4paper,10pt]{article}\n' ...
                 '    \\usepackage{alltt}\n' ...
                 '    \\usepackage{color}\n' ...
                 '    \\usepackage{fullpage}\n' ...
                 '    \\definecolor{string}{rgb}{0.7,0.0,0.0}\n' ...
                 '    \\definecolor{comment}{rgb}{0.13,0.54,0.13}\n' ...
                 '    \\definecolor{keyword}{rgb}{0.0,0.0,1.0}\n' ...
                 '    \\title{%s}\n' ...
                 '    \\author{\\textsc{Matlab}, The Mathworks, Inc.}\n' ...
                 '\\begin{document}\n' ...
                 '\\maketitle\n'],mfile);

%------------------------------------------------------------------------------
function tex_file_end(fid)
    fprintf(fid,'\\end{document}');

%------------------------------------------------------------------------------
function format = tex_format
    format.string    = '\\textcolor{string}{%s}';
    format.comment   = 'textcolor{comment}{%s}'; % '%' has been replaced by '\%'
    format.code      = '%s';
    format.keyword   = '\\textcolor{keyword}{%s}';
    format.cont      = '\\textcolor{keyword}{\\underline{%s}}';
    format.pre_start = '\\begin{alltt}\n';
    format.pre_end   = '\\end{alltt}\n';
    format.nb_line   = '%04d ';
    format.line      = '%s\n';

%------------------------------------------------------------------------------
function str = tex_special_char(str)
    % Special characters: # $ % & ~ _ ^ \ { }
    str = strrep(str,'\','\(\backslash\)'); % $\backslash$ or \textbackslash or \verb+\+
    str = strrep(str,'#','\#');
    str = strrep(str,'$','\$');
    str = strrep(str,'%','\%');
    str = strrep(str,'&','\&');
    str = strrep(str,'_','\_');
    str = strrep(str,'{','\{');
    str = strrep(str,'}','\}');
    str = strrep(str,'^','\^{}');
    str = strrep(str,'~','\~{}'); % or \textasciitilde

%==============================================================================
%                                   RTF FORMAT                                %
%==============================================================================
function rtf_file_start(fid,mfile)
    fprintf(fid,['{\\rtf1\\ansi\n\n' ...
                 '{\\fonttbl{\\f0\\fmodern\\fcharset0\\fprq1 Courier New;}}\n\n' ...
                 '{\\colortbl;\n' ...
```

```
                '\\red0\\green0\\blue0;\n' ...
                '\\red0\\green0\\blue255;\n' ...
                '\\red0\\green255\\blue255;\n' ...
                '\\red0\\green255\\blue0;\n' ...
                '\\red255\\green0\\blue255;\n' ...
                '\\red255\\green0\\blue0;}\n\n' ...
                '{\\info{\\title %s}\n' ...
                '{\\author HighLight.m Copyright 2003}\n' ...
                '{\\creatim\\yr%s\\mo%s\\dy%s}' ...
                '{\\*\\manager Guillaume Flandin}\n' ...
                '{\\*\\company Artefact.tk}\n' ...
                '{\\*\\hlinkbase http://www.artefact.tk/software/' ...textcolorcomment
                'matlab/highlight/}}\n\n'],mfile,...
                datestr(date,'yyyy'),datestr(date,'mm'),datestr(date,'dd'));

%------------------------------------------------------------------------------
function rtf_file_end(fid)
    fprintf(fid,'}');

%------------------------------------------------------------------------------
function format = rtf_format
    format.string    = '{\\cf6 %s}';
    format.comment   = '{\\cf4 %s}';
    format.code      = '{%s}';
    format.keyword   = '{\\cf2 %s}';
    format.cont      = '{\\cf2 %s}';
    format.pre_start = '{\\f0\\fs16{';
    format.pre_end   = '}}';
    format.nb_line   = '{%04d }';
    format.line      = '%s\n\\par ';

%------------------------------------------------------------------------------
function str = rtf_special_char(str)
    str = strrep(str,'\','\\');
    str = strrep(str,'{','\{');
    str = strrep(str,'}','\}');

%==============================================================================
%                                 MATLAB KEYWORDS                             %
%==============================================================================
function matlabKeywords = getMatlabKeywords
    %matlabKeywords = iskeyword; % Matlab R13
    matlabKeywords = {'break', 'case', 'catch', 'continue', 'elseif', 'else',...
                      'end', 'for', 'function', 'global', 'if', 'otherwise', ...
                      'persistent', 'return', 'switch', 'try', 'while'};

%------------------------------------------------------------------------------
function mKeySort = getMatlabKeywordsSorted
    mKeySort.nextincr = {'for', 'while', 'if', 'else', 'elseif', ...
                         'case', 'otherwise', 'try', 'catch'};
    mKeySort.nextincr2 = {'switch'};
    mKeySort.currdecr = {'else', 'elseif', 'case', 'otherwise', 'catch'};
    mKeySort.nextdecr = {'end'};
    mKeySort.other    = {'break', 'continue', 'function', 'global', ...
                         'persistent', 'return'};

```

```
%===============================================================================
%                              HANDLE TAB CHARACTER                             %
%===============================================================================
function str = horztab(str,n)
    %- For browsers, the horizontal tab character is the smallest non-zero
    %- number of spaces necessary to line characters up along tab stops that are
    %- every 8 characters: behaviour obtained when n = 0.
    if n > 0
        str = strrep(str,sprintf('\t'),blanks(n));
    end

%===============================================================================
%                               MATLAB CODE PARSER                              %
%===============================================================================
function splitc = splitcode(code)
    %Split a line of  Matlab code in string, comment and other

    %- Label quotes in {'transpose', 'beginstring', 'midstring', 'endstring'}
    iquote = findstr(code,'''');
    quotetransp = [double('_''.)}]') ...
                   double('A'):double('Z') ...
                   double('0'):double('9') ...
                   double('a'):double('z')];
    flagstring = 0;
    flagdoublequote = 0;
    jquote = [];
    for i=1:length(iquote)
        if ~flagstring
            if iquote(i) > 1 & any(quotetransp == double(code(iquote(i)-1)))
                % => 'transpose';
            else
                % => 'beginstring';
                jquote(size(jquote,1)+1,:) = [iquote(i) length(code)];
                flagstring = 1;
            end
        else % if flagstring
            if flagdoublequote | ...
               (iquote(i) < length(code) & strcmp(code(iquote(i)+1),''''))
                % => 'midstring';
                flagdoublequote = ~flagdoublequote;
            else
                % => 'endstring';
                jquote(size(jquote,1),2) = iquote(i);
                flagstring = 0;
            end
        end
    end

    %- Find if a portion of code is a comment
    ipercent = findstr(code,'%');
    jpercent = [];
    for i=1:length(ipercent)
        if isempty(jquote) | ...
           ~any((ipercent(i) > jquote(:,1)) & (ipercent(i) < jquote(:,2)))
            jpercent = [ipercent(i) length(code)];
            break;
```

```
        end
    end

    %- Find continuation punctuation '...'
    icont = findstr(code,'...');
    for i=1:length(icont)
        if (isempty(jquote) | ...
            ~any((icont(i) > jquote(:,1)) & (icont(i) < jquote(:,2)))) & ...
            (isempty(jpercent) | ...
            icont(i) < jpercent(1))
            jpercent = [icont(i) length(code)];
            break;
        end
    end

    %- Remove strings inside comments
    if ~isempty(jpercent) & ~isempty(jquote)
        jquote(find(jquote(:,1) > jpercent(1)),:) = [];
    end

    %- Split code in a cell array of strings
    icode = [jquote ; jpercent];
    splitc = {};
    if isempty(icode)
        splitc{1} = code;
    elseif icode(1,1) > 1
        splitc{1} = code(1:icode(1,1)-1);
    end
    for i=1:size(icode,1)
        splitc{end+1} = code(icode(i,1):icode(i,2));
        if i < size(icode,1) & icode(i+1,1) > icode(i,2) + 1
            splitc{end+1} = code((icode(i,2)+1):(icode(i+1,1)-1));
        elseif i == size(icode,1) & icode(i,2) < length(code)
            splitc{end+1} = code(icode(i,2)+1:end);
        end
    end

%===============================================================================
%                          MODIFIED VERSION OF STRTOK                          %
%===============================================================================
function [token, remainder, quotient] = strtok(string, delimiters)
% Modified version of STRTOK to also return the quotient
% string = [quotient token remainder]
%STRTOK Find token in string.
%   STRTOK(S) returns the first token in the string S delimited
%   by "white space".   Any leading white space characters are ignored.
%
%   STRTOK(S,D) returns the first token delimited by one of the
%   characters in D.  Any leading delimiter characters are ignored.
%
%   [T,R] = STRTOK(...) also returns the remainder of the original
%   string.
%   If the token is not found in S then R is an empty string and T
%   is same as S.
%
%   Copyright 1984-2002 The MathWorks, Inc.
```

```
%   $Revision: 5.14 $  $Date: 2002/04/09 00:33:38 $

token = []; remainder = []; quotient = string;

len = length(string);
if len == 0
    return
end

if (nargin == 1)
    delimiters = [9:13 32]; % White space characters
end

i = 1;
while (any(string(i) == delimiters))
    i = i + 1;
    if (i > len), return, end
end
start = i;
while (~any(string(i) == delimiters))
    i = i + 1;
    if (i > len), break, end
end
finish = i - 1;

token = string(start:finish);

if (nargout >= 2)
    remainder = string(finish + 1:length(string));
end

if (nargout == 3 & start > 1)
    quotient = string(1:start-1);
else
    quotient = [];
end
```