

Handle Graphics Introduction and Quick Reference
Copyright Dr. C. S. Tritt, 11/27/11 (v. 1.1)

This document provides a very brief introduction to programmatically changing the appearance of Matlab plots using handle graphics. The appearance of graphical objects, including plots, in Matlab is controlled by property values of handle graphics objects. Each of these objects is associated with a unique scalar value called the object's *handle*. These handle values are used with the *get* function to retrieve property values and with the *set* function to set property values for the particular object.

Objects can contain other objects (their children), or be contained within other objects (their parents). The highest level Matlab objects are figures. Figures can contain axes and it is these axes that are commonly thought of as plots. Matlab maintains handles to a particular figure and a set of axes that graphic commands act on by default. These are known as the current figure and current axes. These handles can be retrieved using the *gcf* (get current figure) and *gca* (get current axes) functions, respectively. Axes properties of particular interest include the handles of the plot *title*, *x* and *y* axis labels, and *children* (typically the plotted data (lines)).

Here are two scripts that demonstrate the use of this approach and these functions. The first script, *plotMake.m*, simply creates a plot for later modification. The plot includes two lines, a title and axes labels.

```
% Make a plot for later modification using handle graphics.
% Created by Dr. C. S. Tritt, 11/15/10 (v. 1.0)

clear; close all; clc; % Start clean.

% Generate the data.

x = 0:0.1:10;
y1 = x.^2/4.0;
y2 = 4.0*x.^(0.5);

% Plot the data and decorate the plot.

plot(x, y1, x, y2);
title 'My Title';
xlabel 'My X Label';
ylabel 'My Y Label';
```

The second script, *plotMod.m*, conditionally modifies this plot. It can be run multiple times and changes selected plot properties back and forth between two distinct appearances.

```
% plotMod.m
%
% This script uses handle graphics to modify the appearance of the current
% axis. Note that for clarity, more variables than necessary are used in
% this code. In general, the handles and properties could be retrieved and
```

```

% used immediately rather than saved and used just once on a following
% line.
%
% Created by Dr. C. S. Tritt, 11/15/10 (v. 1.0)

clear, clc; % Clear the workspace and command window. Leave figure open.

% Get the necessary handles and store these in convenient variables.

hAxes = gca(); % Get the handle of the current axes.
pAxes = get(hAxes); % Get the properties of the axes.
hTitle = pAxes.Title; % Store the handle of the title.
hXLabel = pAxes.XLabel; % Store the handle of the X label.
hYLabel = pAxes.YLabel; % Store the handle of the X label.
hLine = pAxes.Children; % Store the line handles (children of the axes).

% Conditionally modify the title and label colors.

tColor = get(hTitle, 'Color');
if sum(tColor) == 0 % if black.
    set(hTitle, 'Color', [1, 0, 0])
else
    set(hTitle, 'Color', [0, 0, 0])
end

xlColor = get(hXLabel, 'Color');
if sum(xlColor) == 0 % if black.
    set(hXLabel, 'Color', [0, 1, 0])
else
    set(hXLabel, 'Color', [0, 0, 0])
end

ylColor = get(hYLabel, 'Color');
if sum(ylColor) == 0 % if black.
    set(hYLabel, 'Color', [0, 0, 1])
else
    set(hYLabel, 'Color', [0, 0, 0])
end

% Conditionally modify the line width. Don't assume a number of lines.

for n = 1:length(hLine) % each line...
    width = get(hLine(n), 'LineWidth'); % Get the current width.
    if width < 3 % if narrow, make it wider.
        width = width + 3;
    else % make it narrower.
        width = width - 3;
    end
    set(hLine(n), 'LineWidth', width); % This actually changes the line.
end

```

The following is a list of some common graphical objects and their properties of particular interest. The data types (classes) of the properties are also listed.

Axes Properties

Color	1x3 double
LineWidth	double
Title	double (handle)
XLabel	double (handle)
YLabel	double (handle)
Children	2x1 double (handle)

Title Properties

Color	1x3 double
FontAngle	char
FontName	char
FontSize	double
FontUnits	char
FontWeight	char

XLabel and YLabel Properties

Color	1x3
FontAngle	char
FontName	char
FontSize	double
FontUnits	char
FontWeight	1x6

Axes Children (typically lines) Properties

Color	1x3 double
LineStyle	char
LineWidth	double
LineMarker	char
MarkerSize	double

Here are descriptions of some of the properties that can be specified.

Color

The color of the item of interest. Can generally be specified as a three element RGB vector or using color name or abbreviation string.

FontAngle

Select italic or normal font. This property selects the character slant for axes text. A value of *normal* specifies a non-italic font, whereas, *italic* and *oblique* specify italic font.

FontName

A string, such as *Courier* or the string *FixedWidth*. Specifies the name of the font to use for the text object. To display and print properly, this must be a font that your system supports. The default font is *Helvetica*.

FontSize

An integer specifying the font size to use for axes labels and titles, in units determined by the *FontUnits* property. The default point size is 12 and the maximum allowable font size depends on your OS.

FontUnits

A string specifying the units used to interpret the *FontSize* property. Valid values include *points* (the default), *normalized*, *inches*, *centimeters* and *pixels*. When set to *normalized*, MATLAB interprets the value of *FontSize* as a fraction of the height of the axes. For example, a normalized *FontSize* of 0.1 sets the text characters to a font whose height is one tenth of the axes' height.

Note that if you set both the *FontSize* and the *FontUnits* in one function call, you must set the *FontUnits* property first so that MATLAB can correctly interpret the specified *FontSize*.

FontWeight

Use to specify *bold* or *normal* font. The character weight for the text. Valid values are *normal* (the default), *bold*, *light* and *demi*.

LineStyle

A string specifying the line style. Supported line styles are: '-' (solid line, the default), '--' (dashed line), ':' (dotted line), '-.' (dash-dot line), and 'none' (no line). Use *LineStyle none* when you want to place a marker at each point but do not want the points connected with a line (see the Marker property).

LineWidth

A scalar value specifying the width of a line object. Specify this value in points (1 point = 1/72 inch). The default value is 0.5 points.

Marker

A string specifying the symbol displayed at data points. Supported markers are: '+' (plus sign), 'o' (circle), '*' (asterisk), '.' (point), 'x' (cross), 'square' or 's' (square), 'diamond' or 'd' (diamond), '^' (upward-pointing triangle), 'v' (downward-pointing triangle), '>' (right-pointing triangle), '<' (left-pointing triangle), 'pentagram' or 'p' (five-pointed star), 'hexagram' or 'h' (six-pointed star) or 'none' (no marker, the default).

MarkerSize

A scalar value specifying the size of the marker, in points. The default value is six points (1 point = 1/72 inch). MATLAB draws the point marker (specified by the '.' symbol) at one-third the specified size.