

```

% -----
% function hfssAssignLumpedRLC(fid, Name, ObjName, iLStart, iLEnd,...
%                               R,L,C,Units)
%
% Description :
% -----
% Create the necessary VB Script to Assign a LumpedRLC.
%
% Parameters :
% -----
% fid      - file identifier of the HFSS script file.
% Name     - name of the lumped port (appears under 'Boundaries' in HFSS).
% ObjName  - name of the (sheet-like) object to which the lumped port is to
%            be assigned.
% iLStart  - (vector) starting point of the integration line. Specify as
%            [x, y, z].
% iLEnd    - (vector) ending point of the integration line. Specify as
%            [x, y, z].
% Units    - specify as 'meter', 'in', 'cm' (defined in HFSS).
%
% Note :
% -----
% only default Units is supported
% RLC_Units = {'ohm', 'nH', 'pF'};
% Example :
% -----
% fid = fopen('myantenna.vbs', 'wt');
% ...
% hfssAssignLumpedRLC(fid, 'LumpedRLC1', 'PIN1', iLStart, iLEnd, 'R_ohm', [], [], Units)

% Note: Support for variable input;
% -----

% -----
% This file is part of HFSS-MATLAB-API.
%
% HFSS-MATLAB-API 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 (at your option)
% any later version.
%
% HFSS-MATLAB-API 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
% Foobar; if not, write to the Free Software Foundation, Inc
% Place, Suite 330, ShangHai, CN Date: 20171222
% Copyright 2018, WoodyBuendia (woodybuendia@gmail.com)
%
% Added function
% 2019.05.25 at YanTai by woodybuendia@gmail.com

function hfssAssignLumpedRLC(fid, Name, ObjName, iLStart, iLEnd,...
                               R,L,C,Units)

```

```

%Type = 'R', Only Assign Resistance, L = [],C = [];
%Type = 'L', Only Assign Inductance, R = [],C = [];
%Type = 'C', Only Assign Capacitance, R = [],L = [];
%Type = 'RL', Assign Resistance and Inductance, C = [];
%Type = 'RC', Assign Resistance and Capacitance, L = [];
%Type = 'LC', Assign Inductance and Capacitance, R = [];
%Type = 'RLC',Assign Resistance and Inductance and Capacitance

%Please note input Units is distance unit,default is mm

% arguments processor.
if (nargin < 6)
    error('Not Enough Arguments !');
elseif (nargin < 7)
    L = [];
    C = [];
elseif (nargin < 8)
    C = [];
elseif (nargin < 9)
    Units = 'mm';
end;

% Setup default arguments.
RLC_Units = {'ohm','nH','pF'}; %only default Units is supported
Type_all = {'R','L','C','RL','RC','LC','RLC'};

%reshape input integration line, change class to cell
iLStart = reshape(iLStart, length(iLStart), 1);
if iscell(iLStart) ~= 1
    iLStart = mat2cell(iLStart,[1,1,1],1);
end

iLEnd = reshape(iLEnd, length(iLEnd), 1);
if iscell(iLEnd) ~= 1
    iLEnd = mat2cell(iLEnd,[1,1,1],1);
end

for i = 1:3
    if isnumeric(iLStart{i,1})
        iLStart{i,1} = [num2str(iLStart{i,1}), Units];
    end
    if isnumeric(iLEnd{i,1})
        iLEnd{i,1} = [num2str(iLEnd{i,1}), Units];
    end
end

RLC = {R;L;C};
for i = 1:3
    if isempty(RLC{i,1})
        Type(i) = 0;
    else
        Type(i) = 1;
        if isnumeric(RLC{i,1})
            RLC{i,1} = [num2str(RLC{i,1}), RLC_Units{i}];
        end
    end
end

% The usual fprintf stuff.
fprintf(fid, '\n');
fprintf(fid, 'Set oModule = oDesign.GetModule("BoundarySetup")\n');
fprintf(fid, 'oModule.AssignLumpedRLC Array("NAME:%s", _\n', Name);

```

```

fprintf(fid, '"Objects:=", Array("%s"), _\n', ObjName);
fprintf(fid, 'Array("NAME:CurrentLine", _\n');
fprintf(fid, '"Start:=", Array("%s", "%s", "%s"), _\n', ...
        iLStart{1,1}, iLStart{2,1}, iLStart{3,1});
fprintf(fid, ' "End:=", Array("%s", "%s", "%s"), _\n', ...
        iLEnd{1,1}, iLEnd{2,1}, iLEnd{3,1});
fprintf(fid, '"UseResist:=", ');
if ~Type(1)
    fprintf(fid, 'false, _\n');
else
    fprintf(fid, ' true, "Resistance:=", "%s", _\n',RLC{1});
end
fprintf(fid, '"UseInduct:=", ');
if ~Type(2)
    fprintf(fid, 'false, _\n');
else
    fprintf(fid, ' true, "Inductance:=", "%s", _\n',RLC{1});
end
fprintf(fid, '"UseCap:=", ');
if ~Type(3)
    fprintf(fid, 'false) \n');
else
    fprintf(fid, ' true, "Capacitance:=", "%s") \n',RLC{1});
end

```

错误使用 hfssAssignLumpedRLC (line 72)  
 Not Enough Arguments !