HyperLynx® 3D EM

Library Manual

Software Version 15.2

Contractor/manufacturer is:
Mentor Graphics Corporation
8005 S.W. Boeckman Road, Wilsonville, Oregon 97070-7777.
Telephone: 503.685.7000
Toll-Free Telephone: 800.592.2210
Website: www.mentor.com
SupportNet: supportnet.mentor.com/
Contact Your Technical Writer: supportnet.mentor.com /doc_feedback_form

IE3DLibrary User's Manual

Release 15.0

Who should use this manual?

- This manual documents the key points of using the IE3DLIBRARY, the 2nd layout editor of IE3D™. It is good for all users of IE3D package.
- It is a supplemental document to the IE3D User's Manual. You should read the IE3D User's Manual and get familiar with IE3D's 1st layout editor MGRID first.
- IE3DLIBRARY is actually built on top of the basic geometry modeling capabilities of MGRID. Getting familiar with MGRID first will help you better understand IE3DLIBRARY.
- IE3DLIBRARY is much easier to use than MGRID because there are few commands involved.
- IE3DLIBRARY should be used in conjunction with MGRID because they are complementary.

Introduction to IE3D

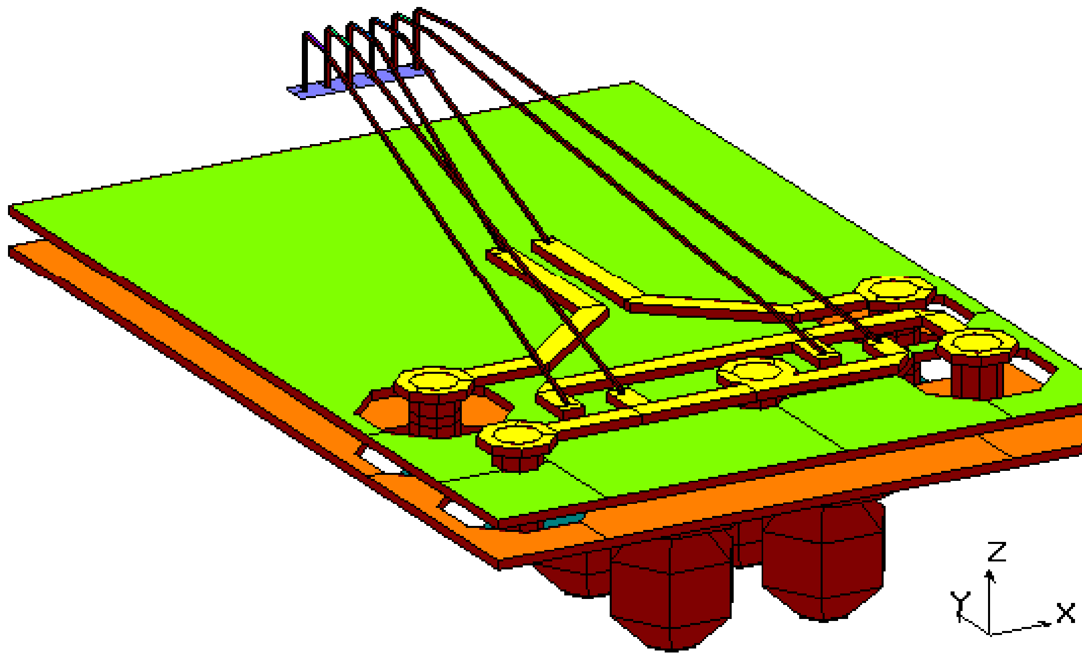
- IE3D is a powerful full-wave EM design package for all aspects of high frequency applications.
- It is based upon 3D integral equation, method of moment for high accuracy and high efficiency full-wave EM simulations.
- It is not just for planar structures. It can also handle full-3D structures elegantly.
- It is not limited by uniform grids and shape of the structures.
- It is much more capable, accurate, efficent and flexible than other EM simulators.

Applications of IE3D

- **Microwave circuits and MMICs.**
- **RF circuits, LTCC circuits, RFICs and RFIDs.**
- **Microwave, RF and wireless antennas.**
- **RF MEMS.**
- **HTS filters.**
- **Electronic packaging and signal integrity.**
- **PCB, MCM, SoP and SiP**
- **EMC and EMI**
- **Many other low to high frequency structures.**

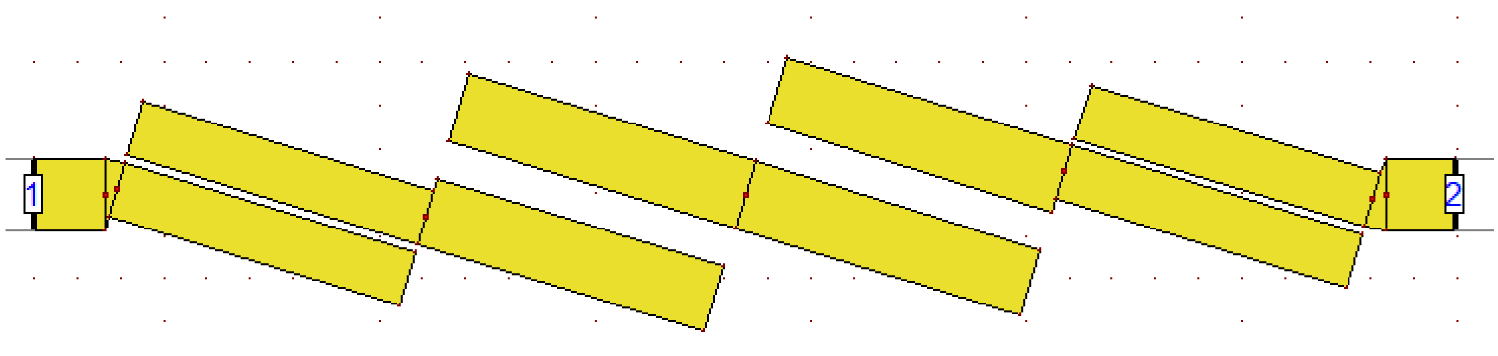
IE3D Dealing with Structure

- **IE3D is solving the current distribution on the metallic parts of a structure. We need to describe the structure.**



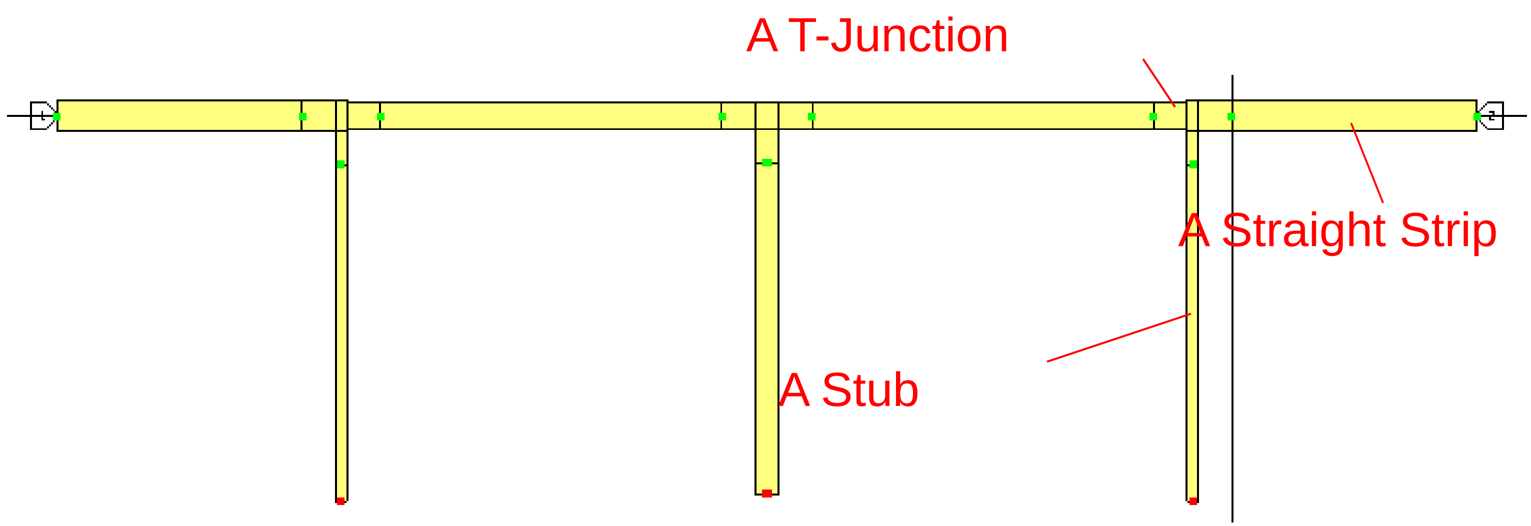
Polygon-Based Layout Editor MGRID

- MGRID has been the standard layout editor for IE3D. It allows a user to create a structure as a set of polygons. A polygon is described as a set of vertices. You have full control over the structure because you can access the polygons and vertices on MGRID. You can change any detail of the structure. However, it is inconvenient to parameterize a structure. For example, it is not simple to change the gap width or the strip length and maintain the symmetry of the filter in the following picture.



Object-Based Layout Editor IE3DLIBRARY

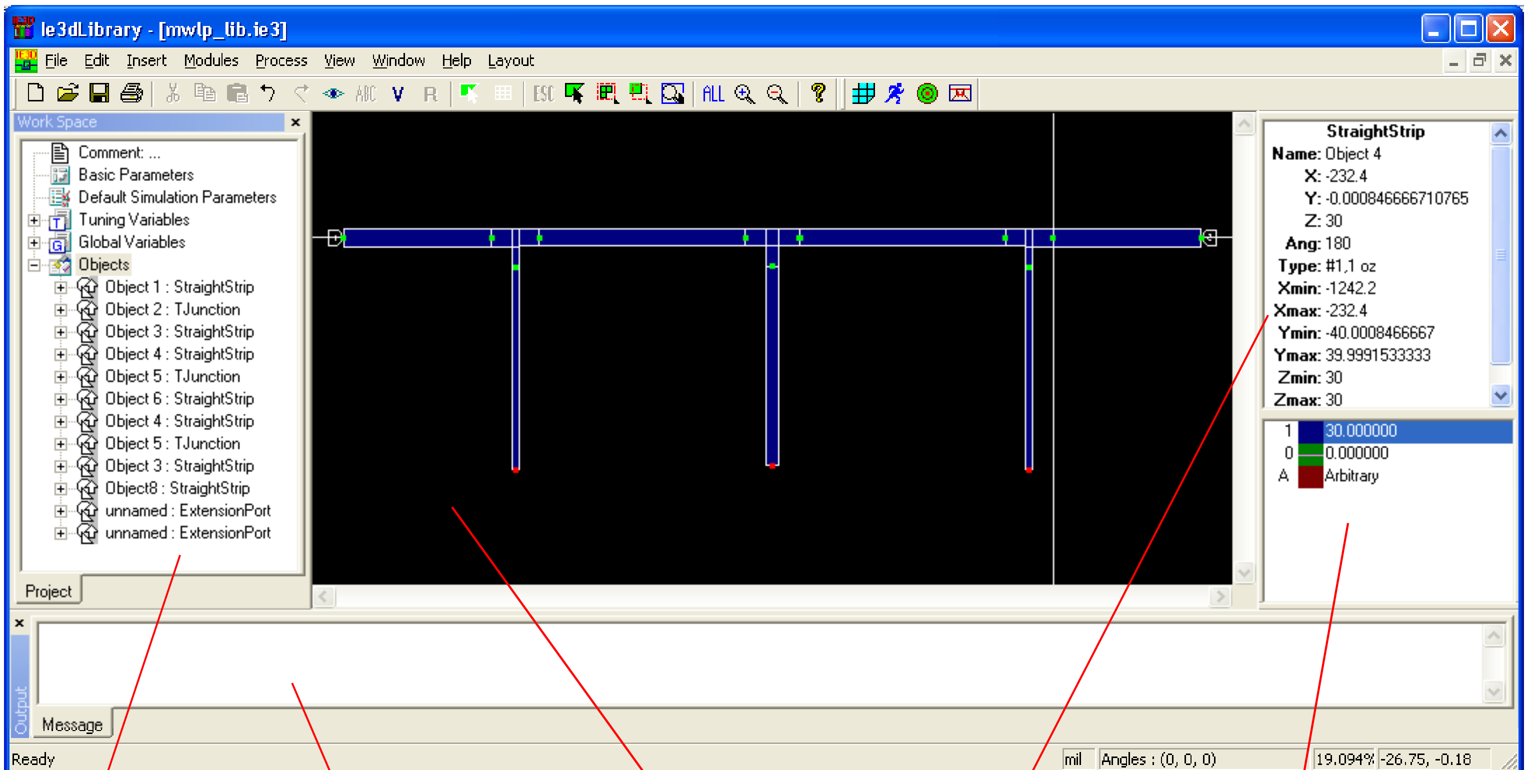
- **IE3DLIBRARY is a layout editor based upon objects. It contains a library of different objects of different shapes. The basic shape of each object is pre-defined while the dimensions can be user-defined. The objects are put together to create a structure can't be found from the library. IE3DLIBRARY is a layout editor working in the same way as schematic editor,**



MGRID vs. IE3DLIBRARY

MGRID	IE3DLIBRARY
Work on lower-level basic elements of polygons and vertices.	Work on high level objects such as T-junctions, circles and spirals.
You have full-control over all the detail of a structure because you can change the locations of the vertices.	You usually have limited control over the sizes and shapes of objects.
Relatively difficult and tedious to parameterize structures for tuning and optimization.	Easy to parameterize structures. Easy to change structures through parameters.
Suitable to complicated structures with small details.	Suitable to structures with fixed basic objects but different dimensions.
The shapes can be modeled are unlimited.	The shapes can be modeled are limited by the library. Implementation of Boolean objects expands this capability.
Less likely to make mistake because what you see is what you get.	Easier to make mistake especially on ports because polygons from objects can become overlapped and ports may be removed in the polygon creation process.
More difficult to correct structure because you may need to change many polygons and vertices to correct one mistake.	Easier to correct mistake because you can still modify the parameters of objects and the polygons will be recreated.

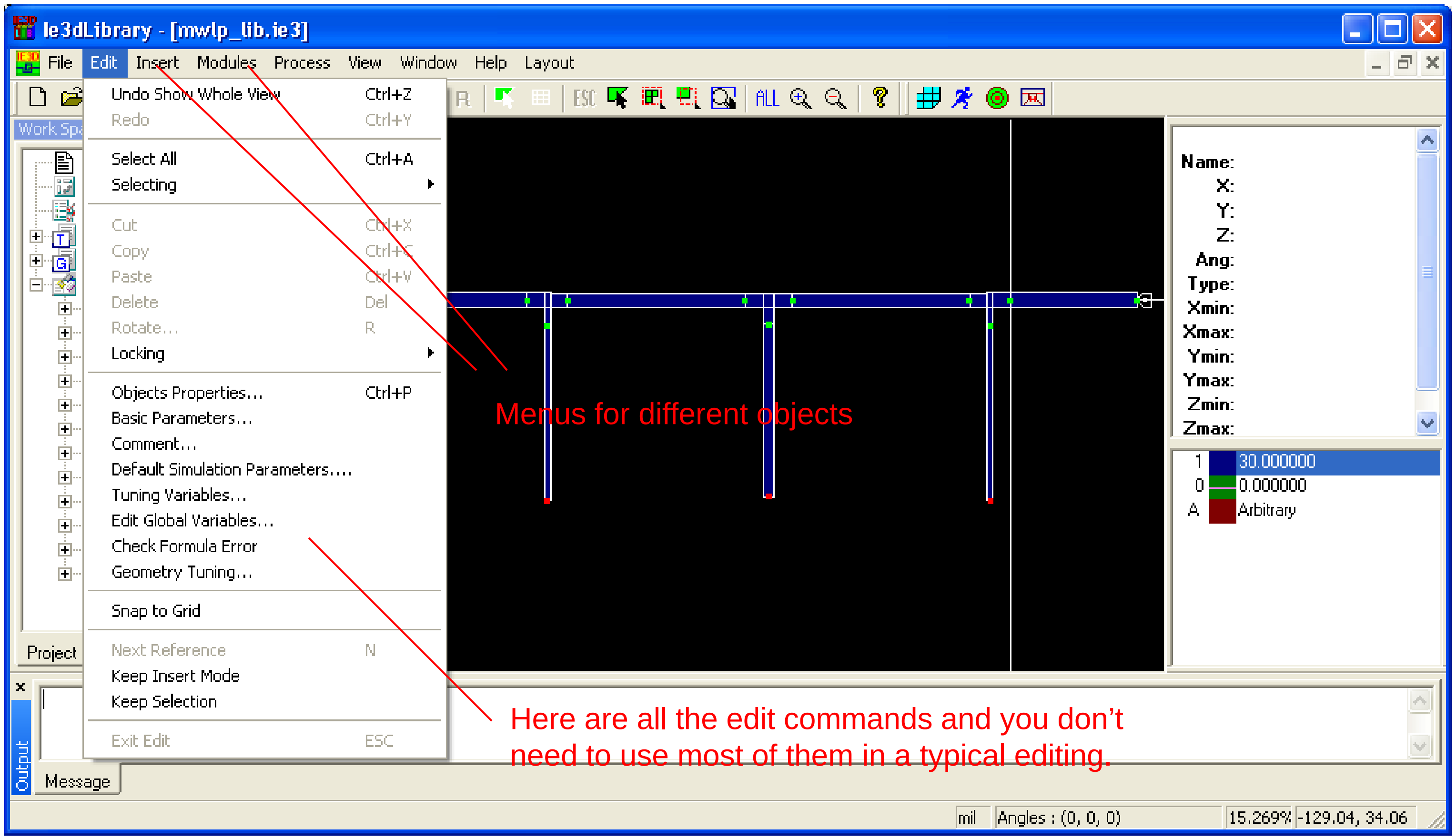
IE3DLIBRARY Window Configuration (1)



Workspace window Window | Information Window | Top View Window | Object Properties Window | Layer

- It is an MDI interface with multiple windows for one document.

IE3DLIBRARY Windows Configuration (2)



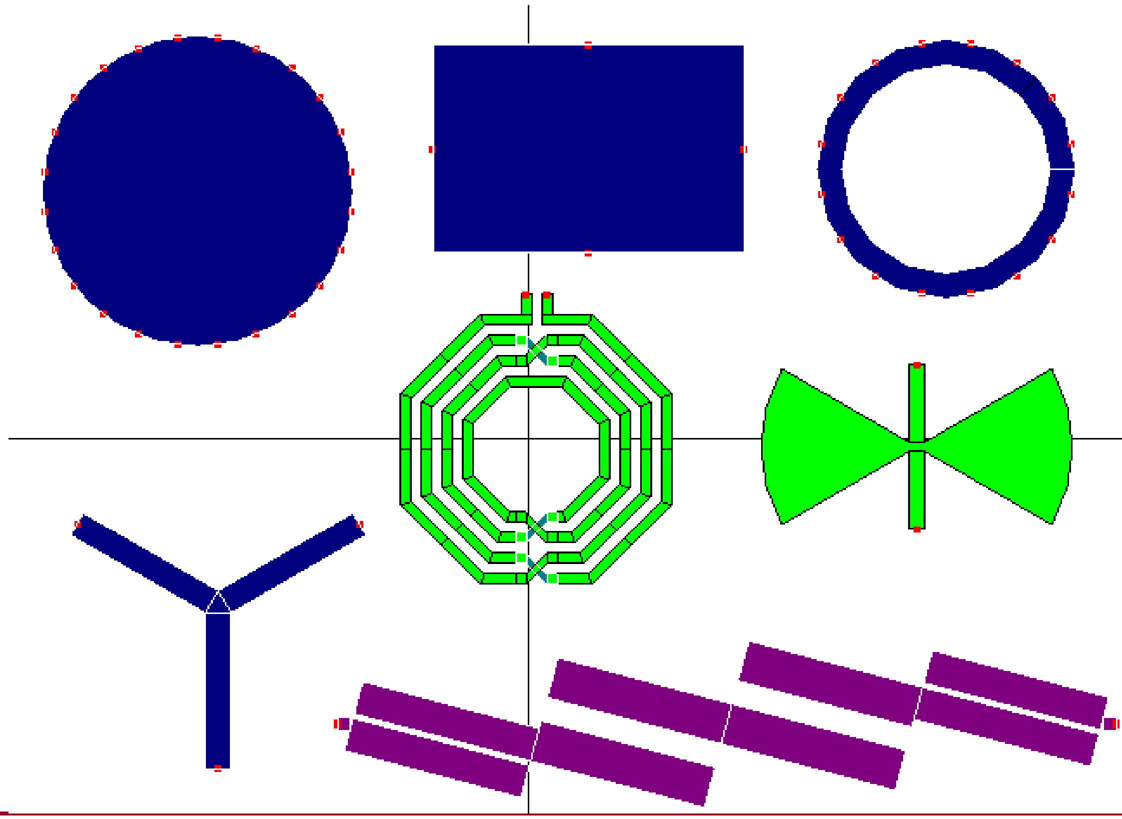
- There are few commands. Most of the menu items are for objects.
- There are just some simple rules on objects.
- Your task is to create what you want using the objects and following the rules.

Organization of This User's Guide

- Because there are few commands in using IE3DLIBRARY, we will focus on the concepts and rules in handling different objects. We will not focus on the step-by-step tutorial. You will find the use of IE3DLIBRARY is quite different from the polygon-based layout editor MGRID. In some sense, IE3DLIBRARY should be easier to use while MGRID may allow more detailed editing.
- Most objects and shapes discussed here can be found in the directory: C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\ie3d\ie3dlibrary_examples.
- The file extension for IE3DLIBRAY is .ie3. You can use IE3DLIBRARY to open the .ie3 files and see what they are.
- The meanings of the structures are self-explanatory in the file names.

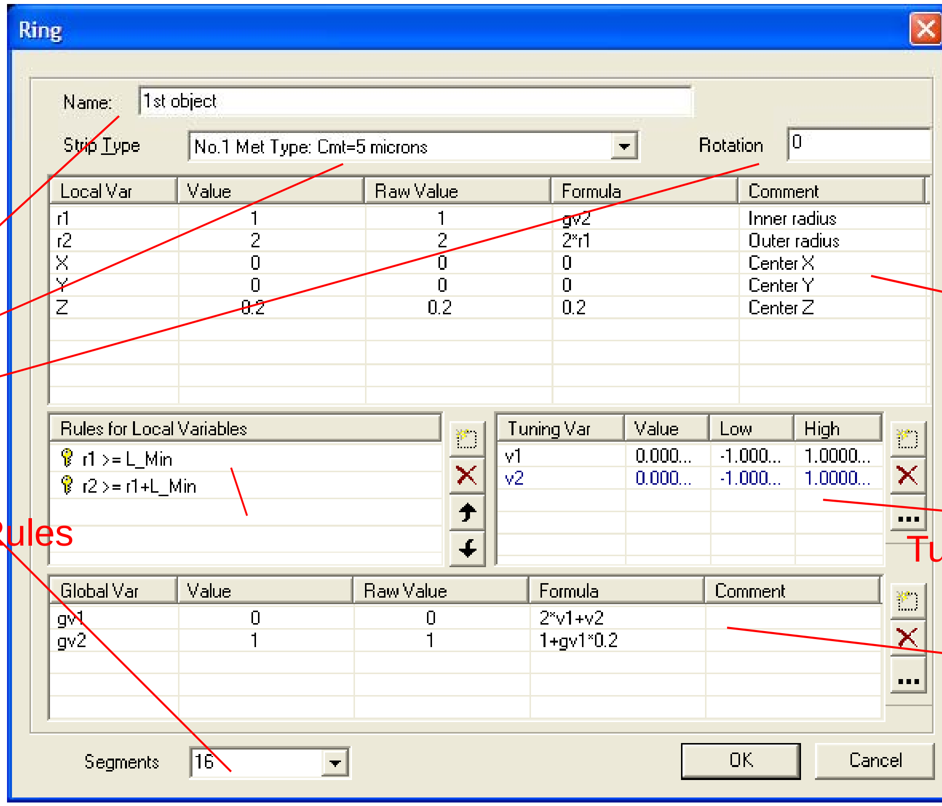
Objects of IE3DLIBRARY

- **There are tens of different objects and all the objects are parameterized. Building a structure is to create it by putting together different parameterized objects.**



The Common Properties of a Typical Object

- **They contain Tuning Variables, Global Variables, Local Variables, Rules and non-equation based local parameters. Actually, Tuning Variables and Global Variables**



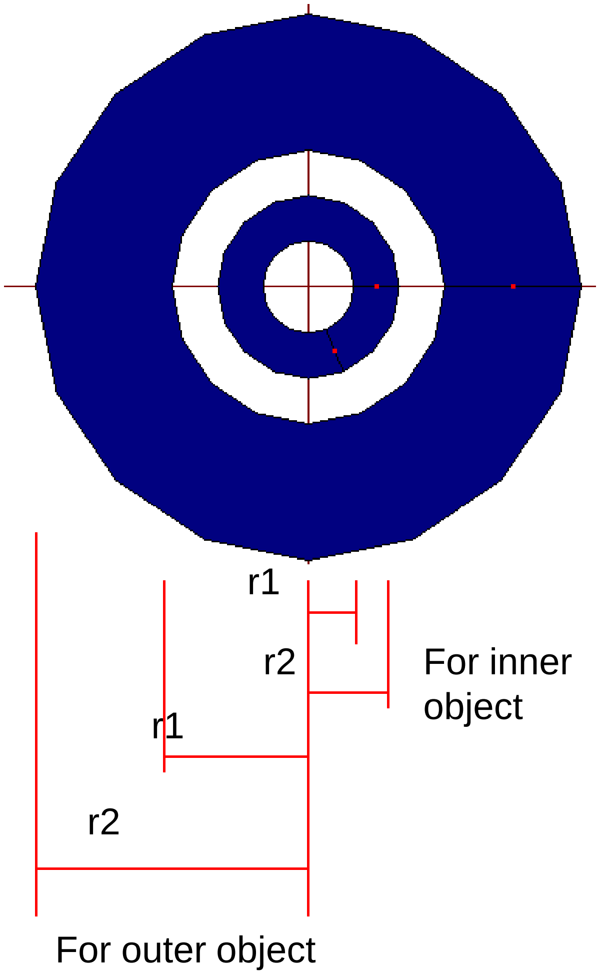
Variables in Parameterization

Variable Type	Scope	Explanation
Tuning Variables	Accessible by global variables and all objects	They are changed by optimizers or the users in FastEM Tuning. A tuning variable can’t be defined as an equation of other tuning variables or global variables.
Global Variables	Accessible by global variables and all objects.	They are intermediate variables. A global variable can be an equation of tuning variables and other global variables.
Local Variables	Accessible by other local variables of the same object.	They belong to a specific object. Each object has its own local variables. A local variable of an object can be an equation of tuning variables, global variables, and other local variables of the same object. However, the equation can’t contain local variables of another object.

Rules for Local Variables will be discussed later.

Example of Parameterization

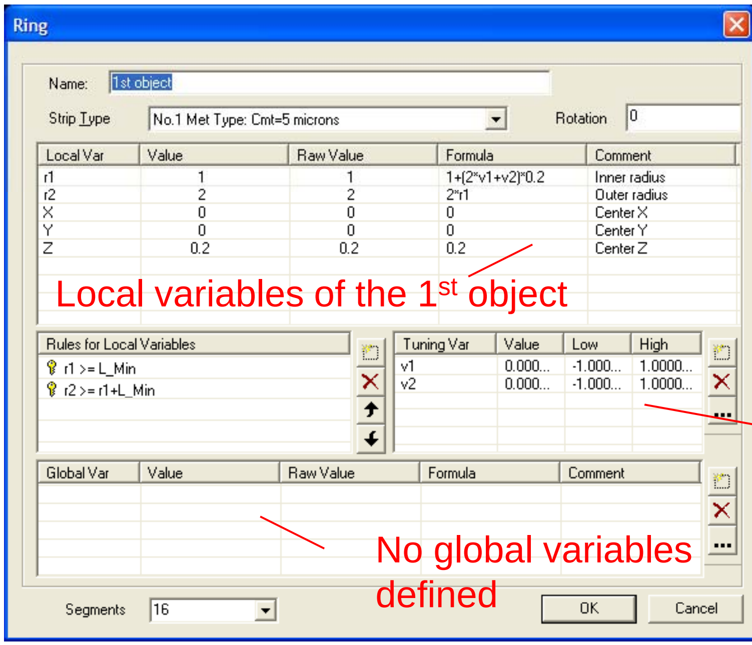
- Assume we have a structure with two rings, one inside another one. Assume we have two tuning variables: v1 and v2. v1 from -1 to 1 and v2 from -1 to 1.
- Assume the inner radius of the inner object is: r1 = 1 + (2*v1+v2)*0.2, and the outer radius of inner object is: r2 = 2*r1.
- Assume the inner radius of the outer object is: r1 = [1+(2*v1+v2)*0.2]*3, and the outer radius of outer object is: r2 = 2 * r1.
- It does not match any single object in IE3DLIBRARY. However, we can construct it as two ring objects. Each object has r1 and r2 variables.



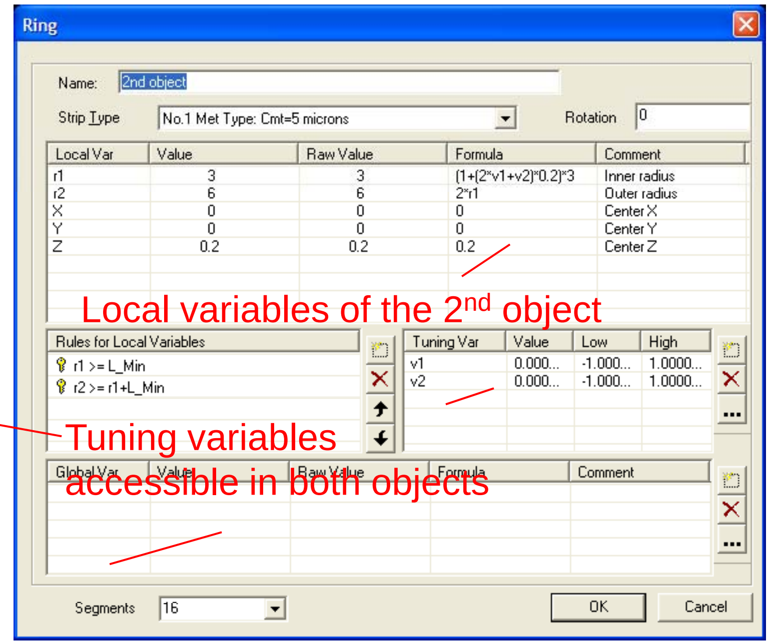
Two Rings Built Without Global Variables

- **File in: .\ie3d\ie3dlibrary_examples\two_rings_no_gva.ie3:**

The 1st Object



The 2nd object



r1 = 1 + (2 * v1 + v2) * 0.2

r2 = 2 * r1

r1 = (1 + (2 * v1 + v2) * 0.2) * 3

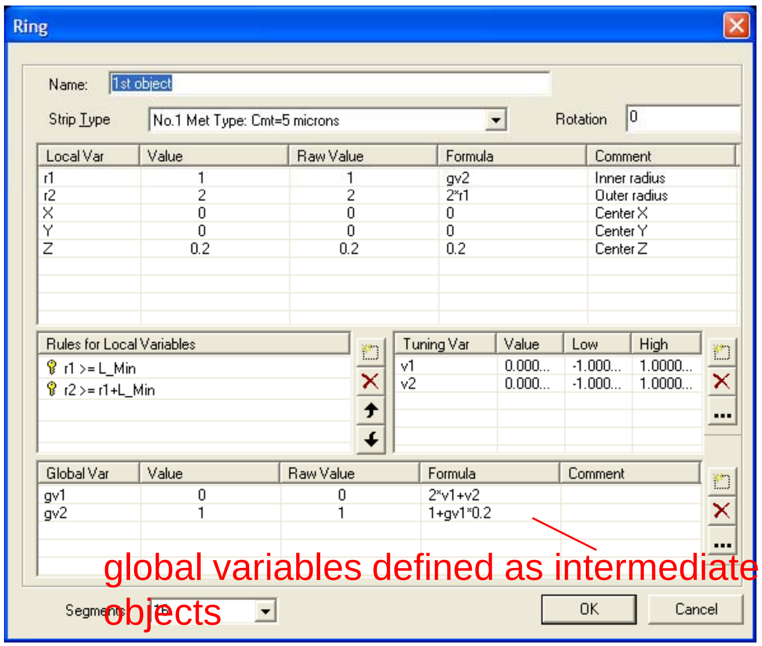
r2 = 2 * r1

Local variables are accessing the tuning variables directly!

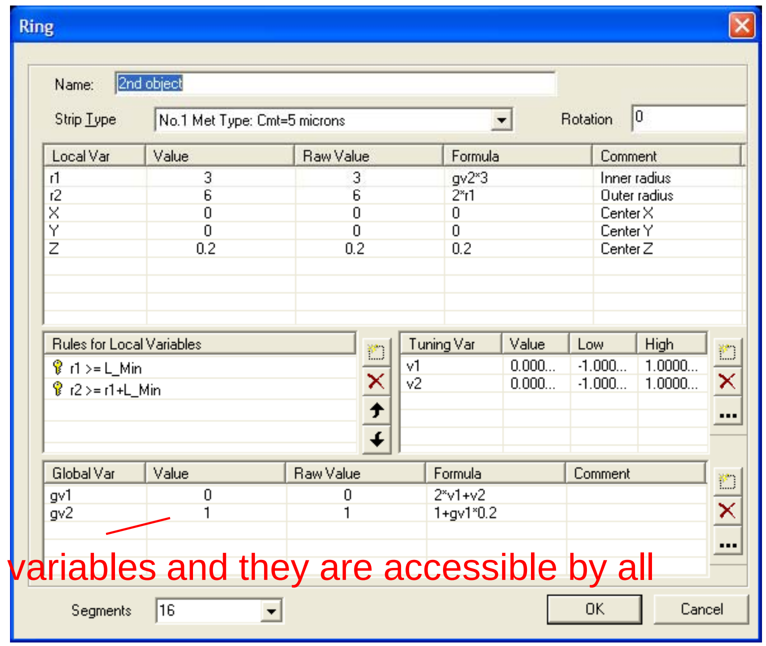
Two Rings Built With Global Variables

- **File in: .\ie3d\ie3dlibrary_examples\two_rings_with_gva.ie3:**

The 1st Object



The 2nd object



global variables defined as intermediate variables and they are accessible by all objects

r1 = gv2

r2 = 2 * r1

gv1 = 2 * v1 + v2

gv2 = 1 + gv1 * 0.2

r1 = gv2 * 3

r2 = 2 * r1

Local variables do not access the tuning variables directly! They are equations of global variables.

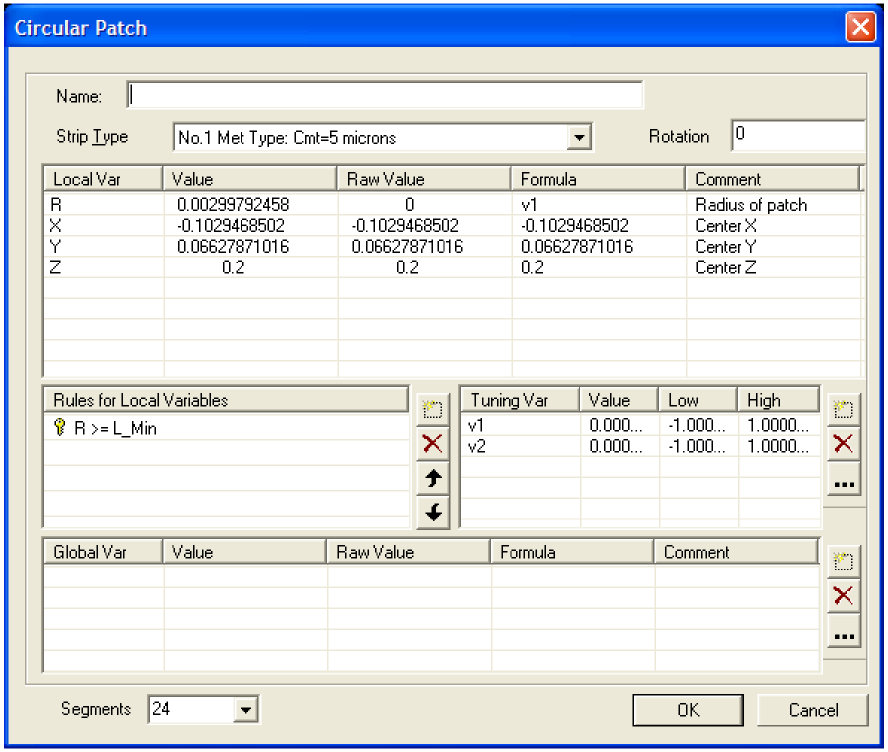
Why do we need global variables?

- Global variables are intermediate variables. It may not be necessary to use them.
- Use global variables make the equations of local variables of each object simpler because you have may same formulas for local variables of different objects.
- It is a good habit not to use tuning variables directly in the equation of a local variable. The reason is that a structure may have many variables you want to tune. We may have to tune selected variables once at a time. We may have to change the tuning variables during the tuning and optimization process. If we define a local variable as an equation of tuning variables directly, we will have to change the equations when we change tuning variables. If we define all local variables as equations of global variables and global variables as equations of tuning variables, we only need to change the global variables when we change tuning variables. It is much easier to change global variables than local variables because the scopes of global variables are for all objects.

Local Variables and Rules

- **A local variable has a Name, a Value, a Raw Value, a Formula (or equation) and a Comment.**

Example: .\ie3d\ie3dlibrary_examples\one_tiny_circular_patch.ie3 defines a small circular patch with radius R = v1, and v1 is the tuning variable from -1 to 1. The Formula of R is “v1”. The Raw Value is 0 when v1 = 0. However, the true value of R is the Value or 0.00299792458. “0.00299782458” is basically L_Min. Why R takes L_Min value. The reason is that there is a rule “R >= L_Min” defined in the Rules for Local Variables. There is a key on the left of the rule meaning this rule is an internally defined rule. You can not change it even though you can add other rules in the Rules for Local Variables section. The radius of a circle can not be equal to 0 and we force it not to be smaller than the L_Min. Every project has an L_Min and it is defined in Edit->Tuning Variables. Rules have order. The later rules will supersede earlier rules. The internal rules are always the last ones because they can’t be violated.

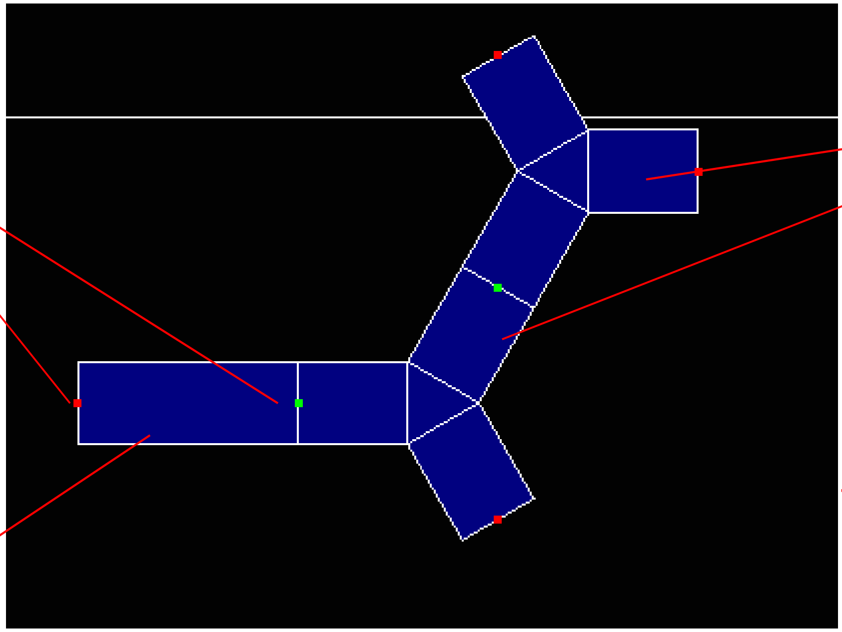


Connection of Objects

- **Each object may have some terminals. The terminals are the handles of the objects. You can grab an object by a handle and snap it to another object.**

The 2 terminals of the straight object. The red marker means it is not connected to other object and the green marker means it is connected to another object

A Straight Strip Object



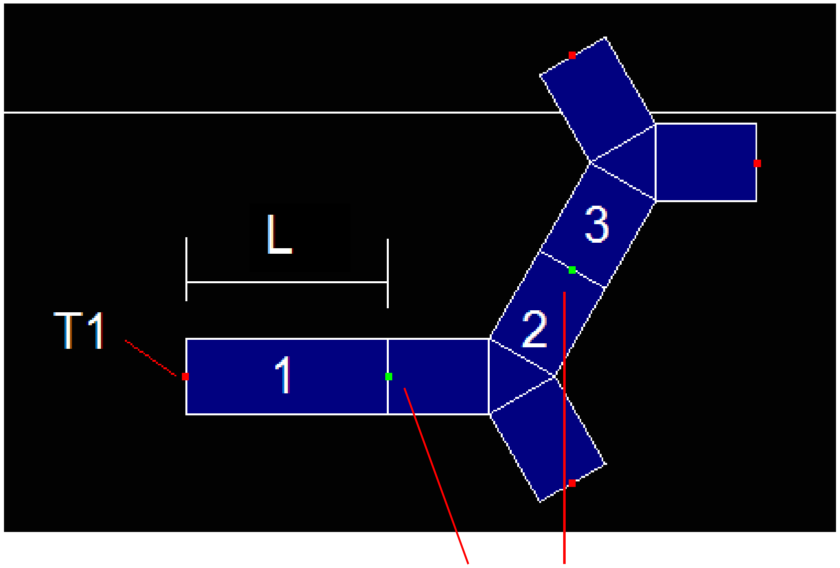
Two Y-Junction Objects

No markers on the common edges of polygons belonging to the same object.

Note: Red marker on an edge between 2 ore more polygons on MGRID means a connection between the polygons.

Connections Kept in Tuning and Optimization

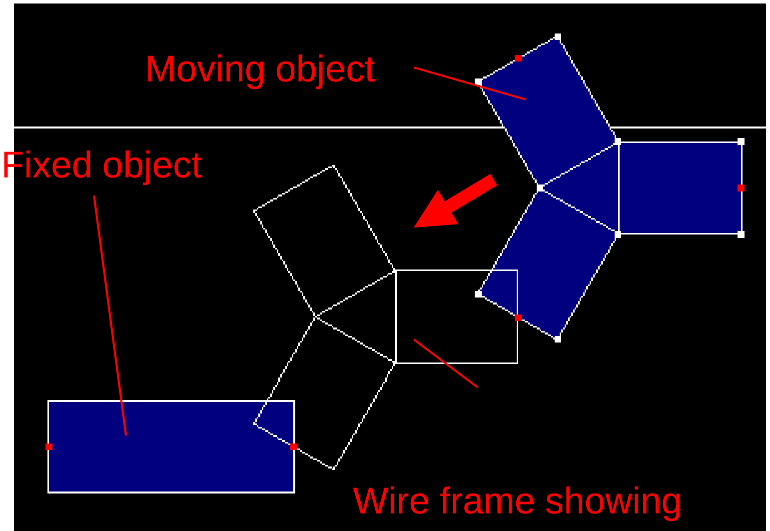
- **The orders of objects are critical in positioning the objects. Later objects always locate themselves based upon previous connected objects.**
- **For the objects 1, 2 and 3 in the picture, the terminal T1 is fixed. When the L is changing with tuning variables, objects 2 and 3 will be re-locating with keep the connections between the objects 1, 2 and 3.**



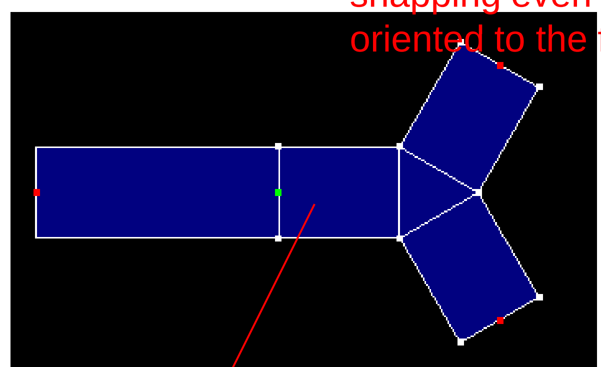
Objects 2 and 3 re-locating with changing L to keep the connections between the 3 objects.

Connection and Snapping of Objects

- **You can grab a terminal of an object and move it toward some fixed object(s). IE3DLIBRARY will detect the closest terminal of the fixed object(s) and snap to it when the distance is close enough.**
- **You can disable snapping by holding the mouse left button down while you are moving an object in case you don't want the object to snap to other objects.**
- **Snapping will happen only when the two terminals are on the same z-coordinates at the current state.**
- **The z-coordinates of objects can be equation-based. The two z-coordinates can be different during tuning and optimization. A connecting via will be created automatically between the two z-coordinates to guarantee the connection.**



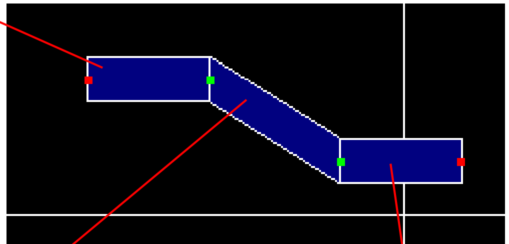
Wire frame showing snapping even not re-oriented to the final angle.

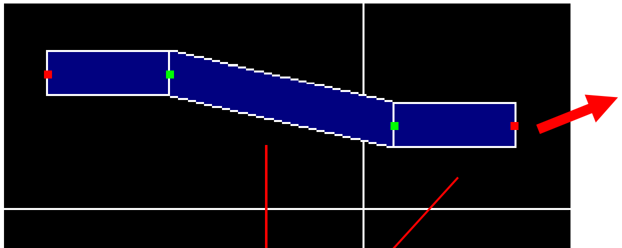


Moving object is moved, re-oriented and fixed with connection after you release the left mouse button.

Independent Objects and Dependent Objects

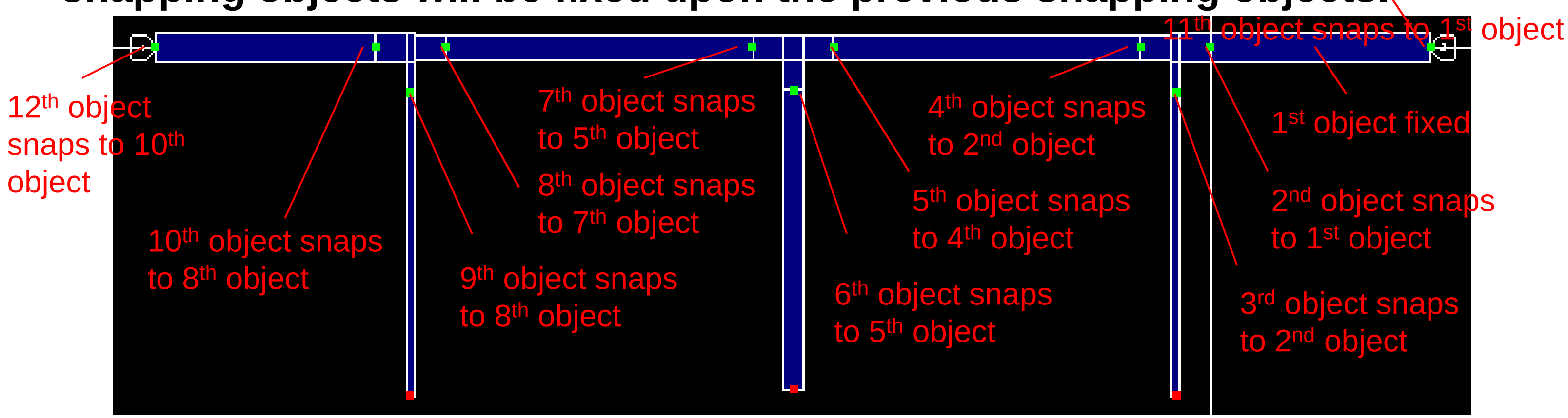
- **Independent objects do not change shape when relocating.**
- **Dependent objects do not exist alone. They need to be defined on independent objects. They may change shapes when they are relocating with the independent objects that they are connected to.**





Snapping and Order

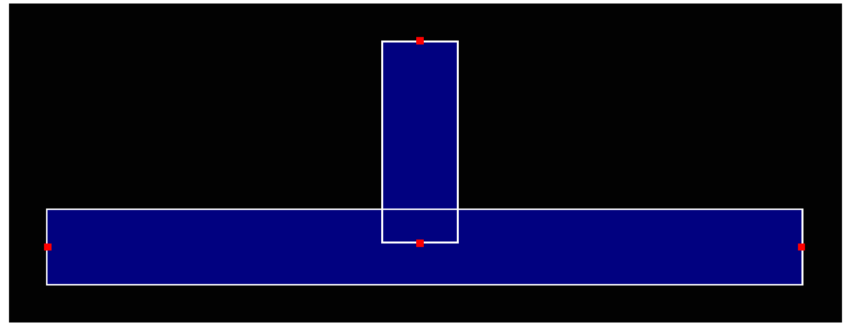
- The order of the objects is important to the creation of final structure.
- An independent object can snap to another independent object on terminals. When such thing happens, the order of the involved objects will be shuffled. Basically, the location of the snapped object will be fixed first, the snapping object will fix its location and orientation based upon the snapped object.
- An independent object may not need to snap to other objects. In case it snap, it can only snap to one independent object . It can’t snap to more than one object.
- A dependent object can snap to independent objects only. It can’t snap to other dependent objects.
- The structure creation process is like a train. The 1st object is fixed, then the snapping objects will be fixed based upon the snapped objects. The next snapping objects will be fixed upon the previous snapping objects.



Overlapping of Objects

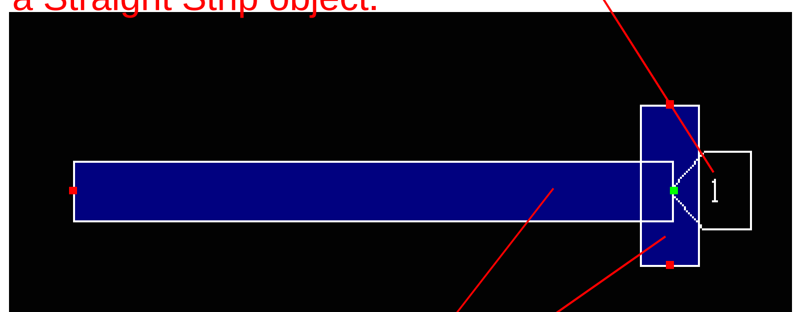
- Overlapped objects will be automatically merged on IE3DLIBRAY.

- Please avoid overlapping on edges where ports are defined. Otherwise, the ports may become invalid and IE3DLIBRARY should issue you an error message during simulation or optimizaiton.



Two Straight Strip objects overlapping. They will be considered to be connected in IE3DLIBRARY.

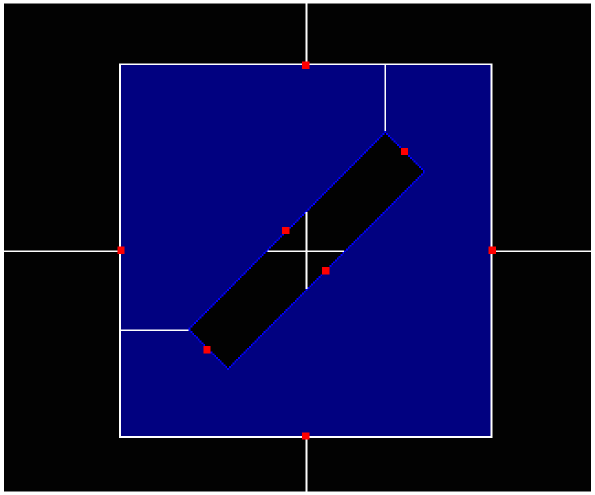
An Extension Port object is defined on a terminal of a Straight Strip object.



Two Straight Strip objects overlapping on the terminal where an Extension Port is defined. It will cause problem in simulation and optimization.

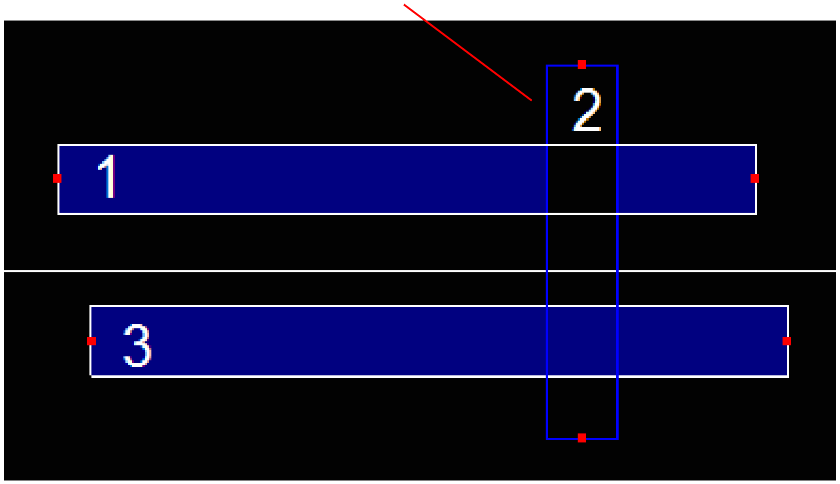
Boolean Objects for New Shapes

- Boolean objects applied to other objects may create new shapes not found in the library.
- Orders of objects are critical in Boolean operations.



A slotted square patch is created with a Rectangular Patch and a slanted Rectangular Patch with Boolean properties.

Boolean object displayed as Blue wire frames



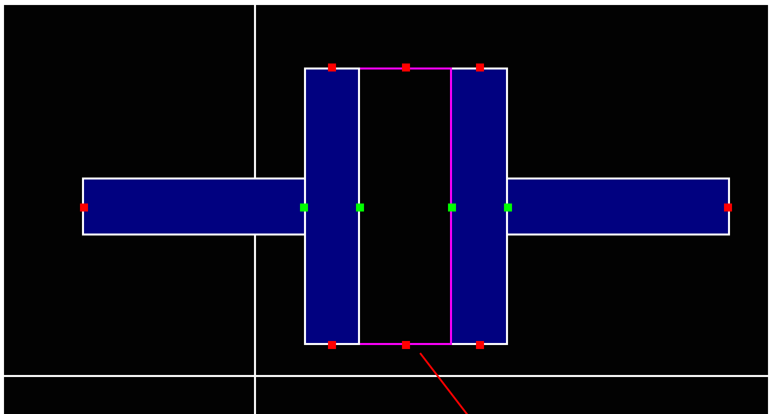
Boolean object 2 cuts a slot on object 1 while it does not cut a slot on object 3 because object 3 is behind object 2 in order.

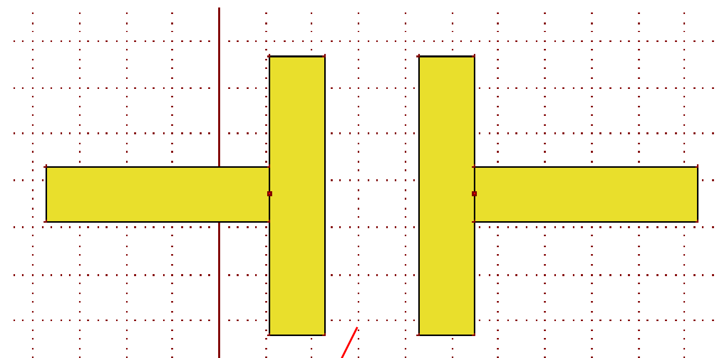
Void Objects for Locating Other Objects

- **Void objects do not appear in the final simulations. However, they are used to locate other objects and they can also be parameterized.**

Schematic-layout processing

Final layout for simulation

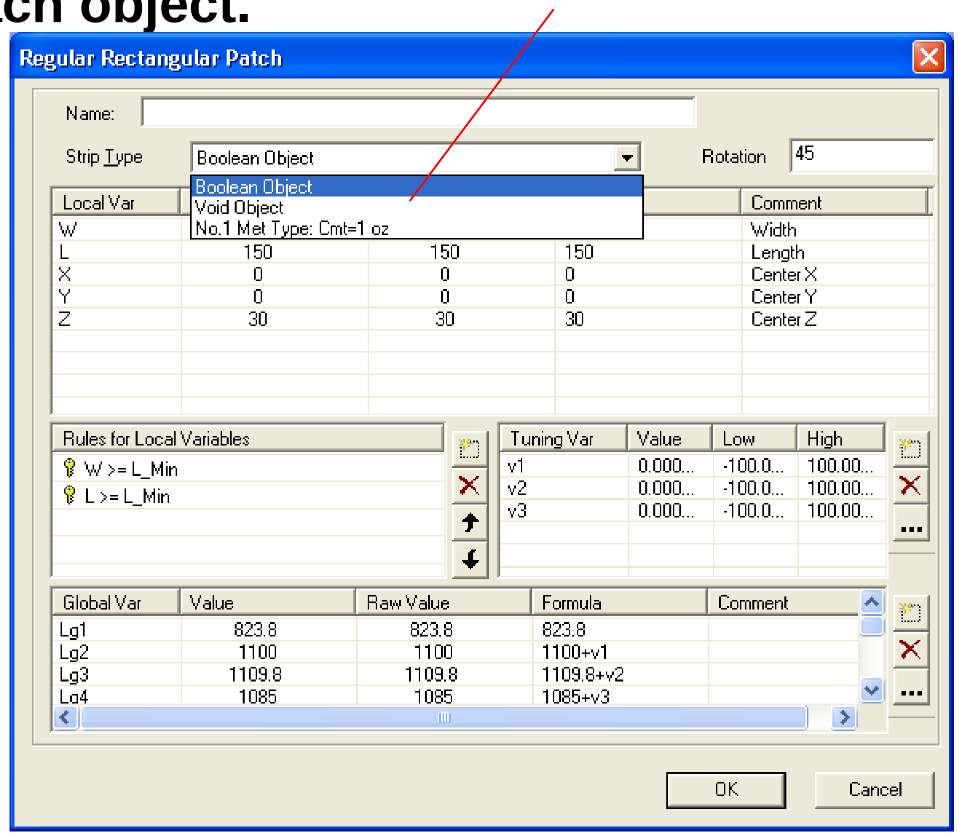




A Void object is displayed in violet color in the Schematic-Layout editor while it does not appear in the final geometry for simulation. It helps to keep the relative location of the other objects in the parameterization

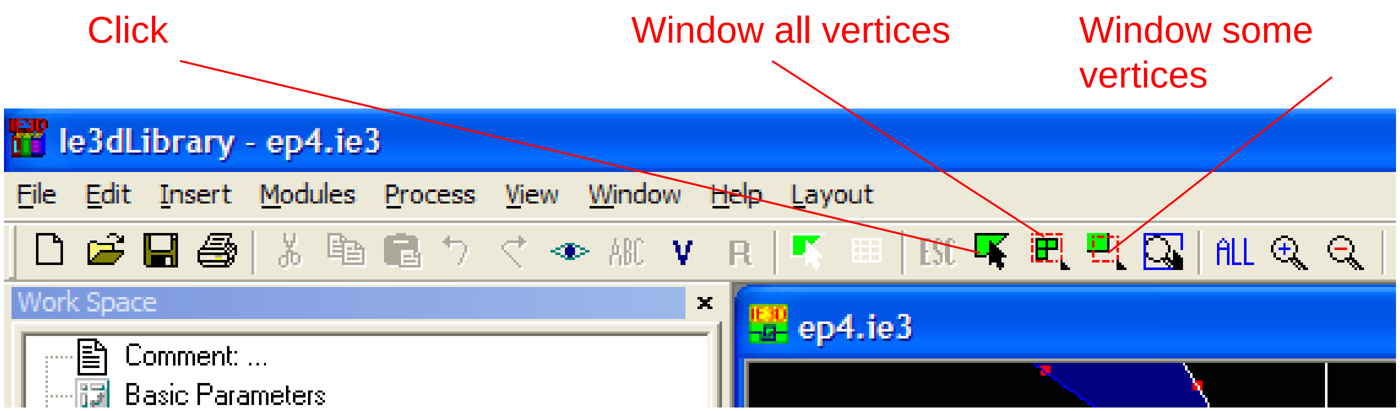
Easy to Define Boolean and Void Objects

- **You can specify whether an object is Boolean object, Void object or regular object in the Strip Type combo in the Properties dialog of each object.**
  - Some objects can't be Boolean objects. Some can't be Void objects. You will see the choices are not in the list if the object is not allowed to be a Boolean object or a Void object.
  - You can define multiple metallic strip types in Basic Parameters and select one for an object.



Selection and Editing of Objects

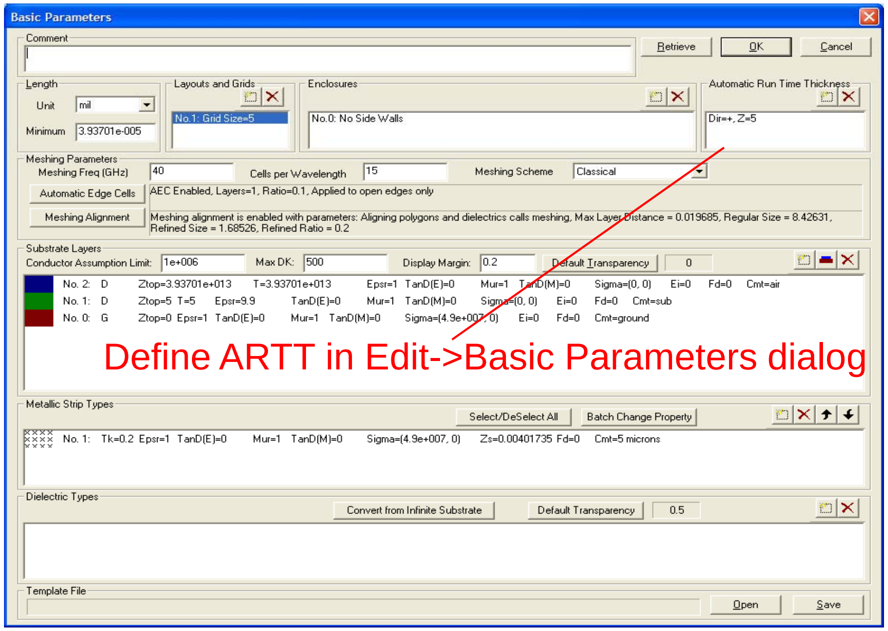
- There are multiple ways to select one or more objects. They are in 3-different modes.
  - Click at the object you want. This is the simplest way. To select one more object, you can hold SHIFT button and click again.
  - Window all the vertices of the objects to select them. It allows you to select objects when they are overlapped.
  - Window some vertices of the objects. It allows you to select objects when they are overlapped.
- You can also select objects in Edit->Objects Properties dialog. Future improvement will allow you to select objects in the Workspace window.



Automatic Run-Time Thickness Modeling

- If you have used IE3D/MGRID, you should have known that you can create high accuracy thickness model using the Grow Thickness… commands.
- You can also use Automatic Run-Time Thickness (ARTT) feature in Basic Parameters on IE3D/MGRID to get the run-time thickness model. ARTT allows you to build the models as thin polygons while the model at run-time is thickness model. On MGRID, you can select Process->Create Run-Time Thickness Model to see how the thickness model looks. You can also simulate the structure and save the current distribution file. You will be able to see the thickness model in post-processing of the current distribution on MGRID.

  - On IE3DLIBRARY, thin model is default. However, you can define thickness model for specific z-coordinates on Edit->Basic Parameters.
  - Please note that you should avoid defining a layer with tunable z-coordinate as a thickness layer. At this time, ARTT will apply to layer with fixed z. If you have a layer with changing z-coordinate controlled by tuning variables, you will get thickness layer when the z-coordinate matches one of the z-coordinates defined in the ARTT section of the Basic Parameters.

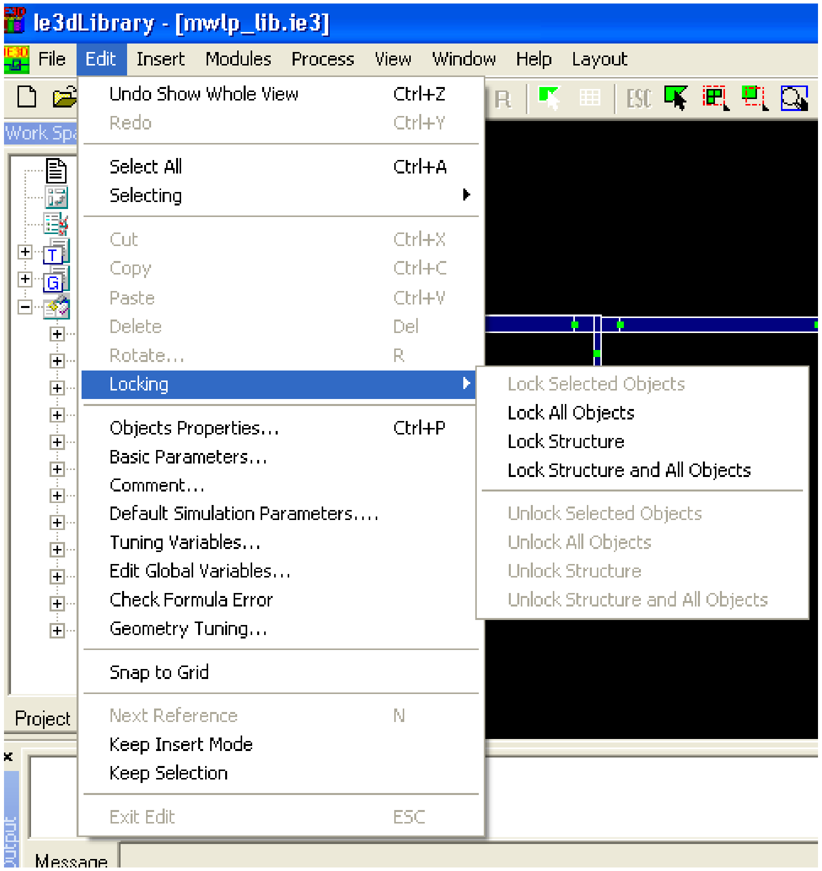


What results can you get?

- After you build a structure and define some ports on it, you can simulate it in the same way as you simulate layouts on MGRID.
- You can obtain the s-, y-, z-parameters, current distribution and radiation patterns. You can do near field calculation and visualization. You can also find the frequency dependent and frequency independent equivalent circuit.
- Current distribution and near field visualization will be on cells displayed on MGRID. You are allowed to calculate and visualize near field and far field based upon current distribution
- Starting from V12, the s-parameters are also appended in the .ie3 file and you can display s-parameters on IE3DLIBRARY.
- You can optimize the structure on IE3D.
- You can also create FastEM data for real-time full-wave EM tuning and optimization.
- You can also select an independent object and simulate its performance alone.

Locking of Structures

- A structure consists of a number of objects. You can lock a structure or some selected objects of the structure.
- When a structure is simulated, the s-parameter results are stored inside the file and the structure will become locked automatically. Unlocking or modifying the structure will cause removal of the s-parameters.

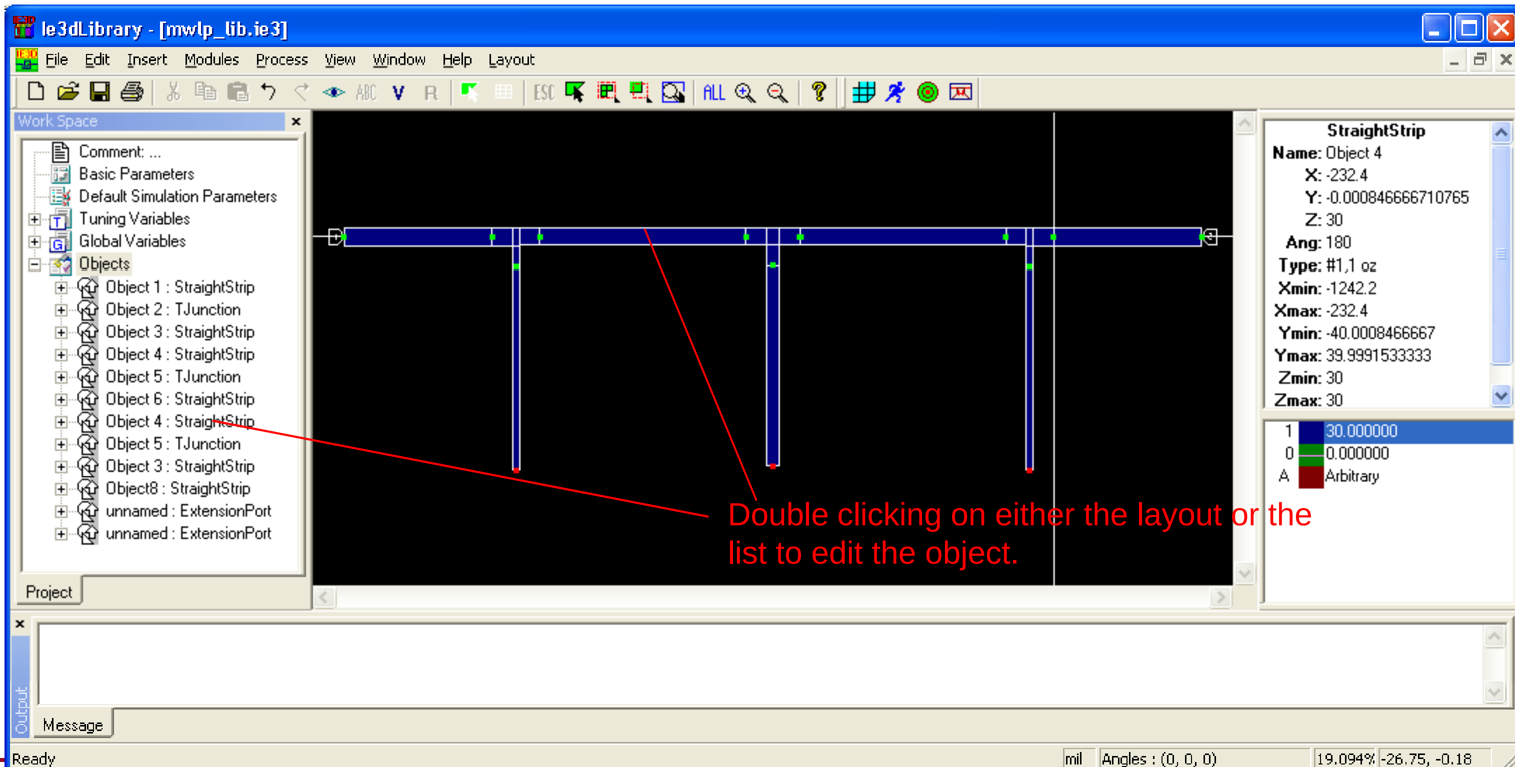


Locking of Objects

- You can also selected objects and lock the selected objects.
- Locking an object will not allow you to edit the parameters of the object. However, you can rotate and move the objects as you like. The objects may still adjust the location and angle due to snapping.
- Locking an object even does not mean the shape of the object is locked. It is possible the shape may be changed due to changes in the Tuning Variables and the Global Variables.

Editing of Objects

- You can double-click at the object in the Top View window or the Objects List on the Workspace window to bring out the dialog to edit an object.
- You can also select some object(s) on the layout and go to Edit->Objects Properties to edit the objects.



Selection and Moving of Objects

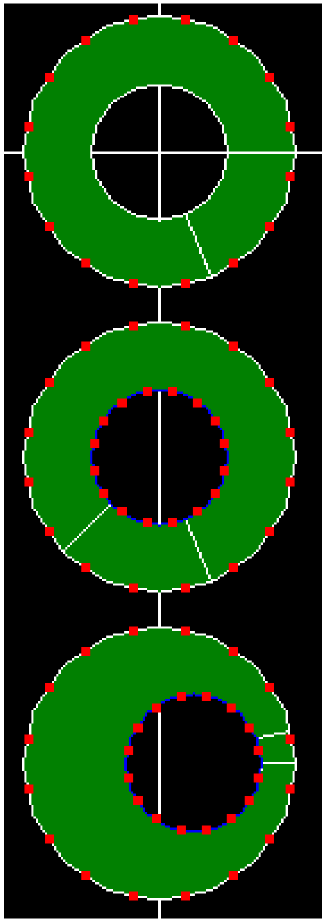
- **You can select an object by clicking at it on the Top View.**
- **You can hold down SHIFT or CTRL button and click at more objects to select multiple objects.**
- **You can move the objects by dragging the selected objects.**
- **When you are dragging the objects, the mouse cursor will be on one of the terminals of an object and you can snap it to a terminal of any fixed objects as long as the terminal is not connected. Not connected terminals are shown with red dots.**
- **To avoid snapping, yon can hold SHIFT button.**
- **You can also use the arrow keys on the key pad to move the objects. No snapping will happen if you use arrow keys.**
- **You can also change the location and rotation angle of a structure inside the properties dialog of an object.**

Flexibility and Error Handling

- **Various schemes are carefully planned and implemented to ensure maximum flexibility in parameterizing structures.**
- **Rules for local objects ensure the dimensions of objects will not go out of range in tuning and optimization.**
- **Due to high flexibility in tuning and optimizing objects, it is impossible to ensure a structure to be valid in all cases. There are possibilities that a structure can become invalid even all the rules are followed. The most frequently encountered problems are related to ports because ports have strong requirements on the polygons where the ports are defined.**
- **You can always open the internally created structure in MGRID to see how the polygons are. The internally created structure contains the polygons to be simulated in the simulation engine.**

Flexibility and Wise Use of IE3DLIBRARY

- **Again, IE3DLIBRARY is very flexible. There are alternative ways to build the same structure. Please try to use it wisely and creatively. For example, you want to build a ring structure. You can use use a Ring object to create it. You can also use 2 Circular Patch objects of different radiuses to build it with the 2nd smaller circle as a Boolean object. It is more convenient to use the 1st way or the Ring object when the 2 circles of the ring are co-centered. If the 2 circles are not co-centered or the ring does not have uniform width, you have to use the 2nd way to build it.**



Constant width ring built as a Ring object.

Constant width ring built as 2-Circular Patch objects with the smaller one as Boolean object.

Non-uniform width ring built as 2-Circular Patch objects with the smaller one as Boolean object.

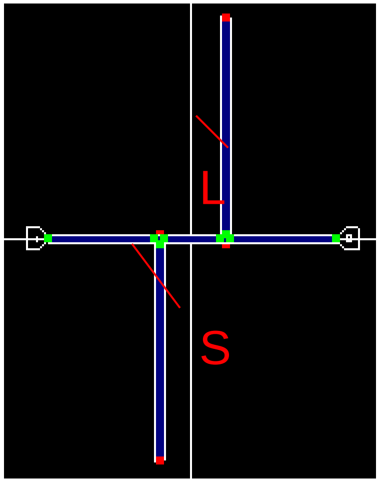
IE3DLIBRARY Project Examples

- In the following pages, we will try to document some simple examples of using IE3DLIBRARY. You can learn the various techniques in using the different objects and schemes in IE3DLIBRARY in creating structures. All structures are simple ones for illustrating the concepts only. You can apply IE3DLIBRARY to much more complicated structures.
- Since there are not many commands involved, we will not document the steps of creating the example structures. We will just provide some ideas and explanations for some specific considerations in creating the objects.
- All the example files are stored in C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\ie3d\ie3dlibrary_examplesdirectory. Please try to open the files and see how we parameterize the structures.

Double-T Structure (1)

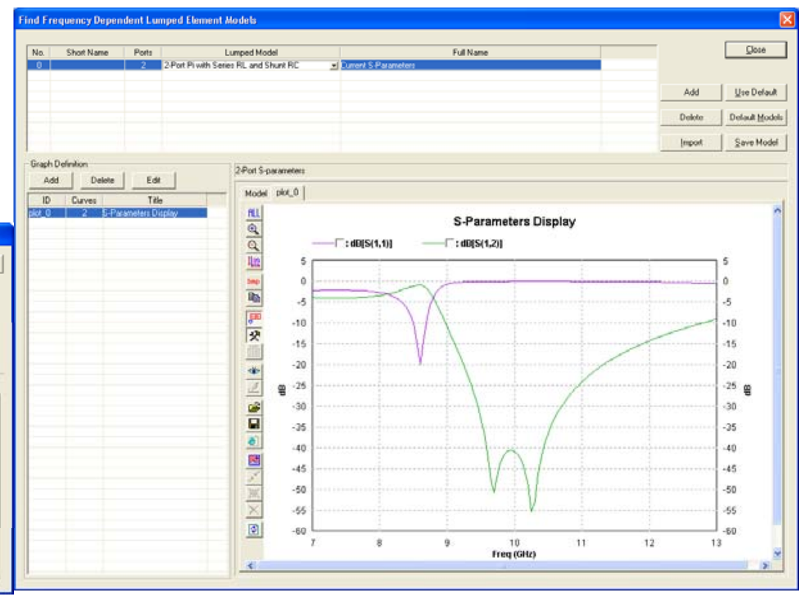
- **This is a famous structure found by designers in TI in late 1980s. The files are created in: c:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\ie3d\ie3dlibrary_examples\double_tee.ie3 and double_tee_FastEM .ie3. Both structures are the identical except the 1st one contains simulation result while the 2nd one contains FastEM Design Kit data for EM tuning and optimization. The structure is parameterized with the separation S = 30 + 5 v1 (mils) with v1 = [-1, 1] and the stub length L = 115 + 20 v2 (mils) with v2 = [-1, 1].**

- The structure is created with two Rectangular Patch objects for the T-junctions and four Straight Strip objects for the two stubs and the two feed lines. They are all snapped together with two Extension ports defined at both ends.
- The double_tee.ie3 is simulated from 7 to 13 GHz with 121 frequency points and v1 and v2 are both set to 0.
- The double_tee_FastEM.ie3 is simulated in the same frequency range with 11 points for both v1 and v2 from –1 to 1. The points of v1 and v2 are defined in the FastEM Design and EM Tuning section of the Simulation Setup dialog.

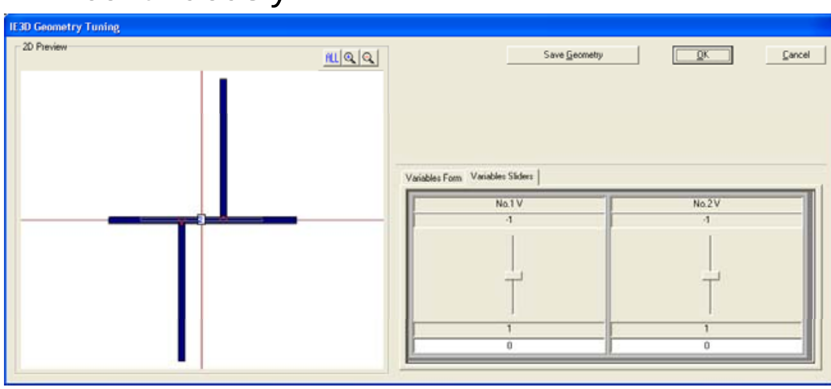


Double-T Structure (2)

- Open double_tee.ie3 file. This file contains FastEM data for s-parameters visualization.
- Select Process->S-Parameters and Lumped Equivalent Circuit command. The Find Frequency Dependent Lumped Element Model dialog comes up. By default, you get the 2-Port Series RL and Shunt RC model. You can change to other model. However, any simple lumped model may not be appropriate for such a structure.
- Please select Add button in Graph Definition group. Select S-Parameters. Select dB[S(1,1)] and dB[S(1,2)] for a graph. A graph called “Plot_0” is created for you. Please select “Plot_0” in the display tabs to show the S-parameters. Close the dialog.
- Select Edit->Geometry Tuning. IE3DLIBRARY will warn you that FastEM data will be lost if you continue. You can select YES. It is ok to remove the FastEM data as long as you will not save the change into the same file.

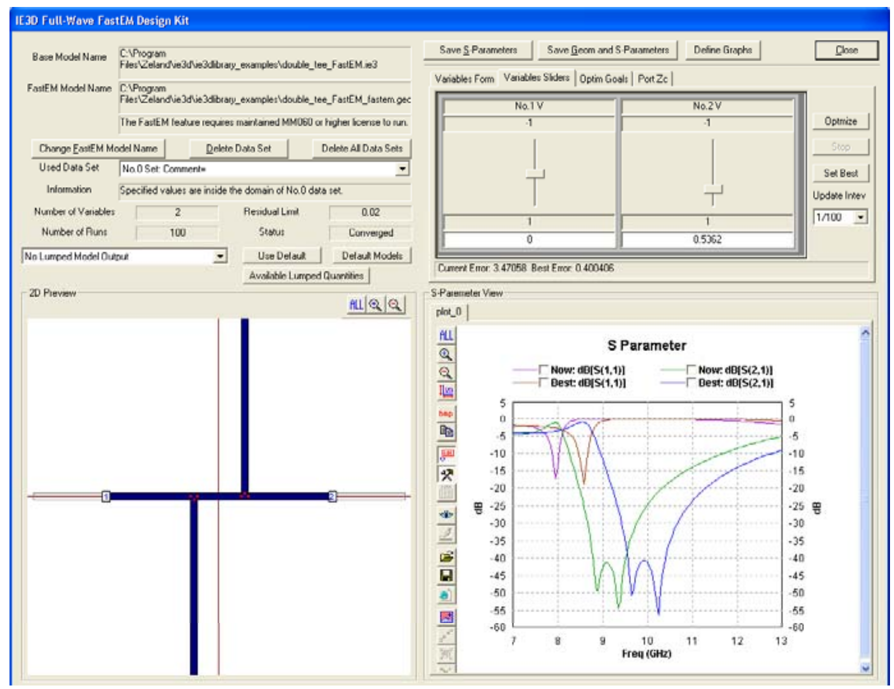


- Select the Variable Sliders in Geometry Tuning dialog. Slide the bars and see how they change the shapes of the structure continuously.

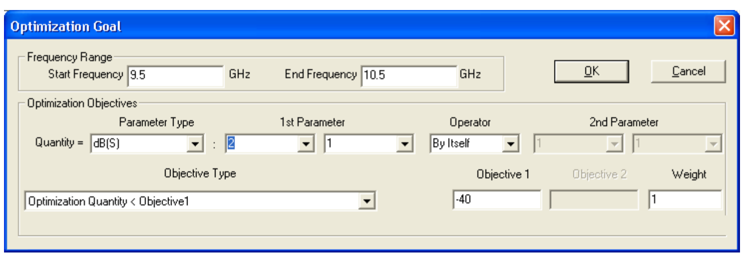


Double-T Structure (3)

- **Open double_tee_FastEM.ie3 file. This file contains FastEM data for tuning and optimization because many combinations of different V1 and V2 values are simulated.**
- **Select Process->Full-Wave EM Design using FastEM Design Kit to open the dialog. The V1 and V2 are set to 0 and the dB[S(1,1)], dB[S(1,2)], Ang[S(1,1)] and Ang[S(1,2)] are shown in the default “Plot_0”. Please select Define Graph. Select “Plot_0” and uncheck Ang[S(1,1)] and Ang[S(1,2)] for a plot showing dB[S(1,1) and dB[S(1,2)]. Select Port Zc tab to make sure the Zc = 50 (by default).**
- **Select Optim Goals tab and Insert button to define a goal as: dB[S(2,1)] < - 40 from 9.5 to 10.5 GHz.**
- **Select Variable Sliders tab and slide the bars. It will capture the best result for your goals while you slide the bars. Select the Optim button. It will optimize your structure for the best result with your defined goals in real-time. After the optim, you can Set to the Best for the design. You should save the change into file if you want to keep the best results.**

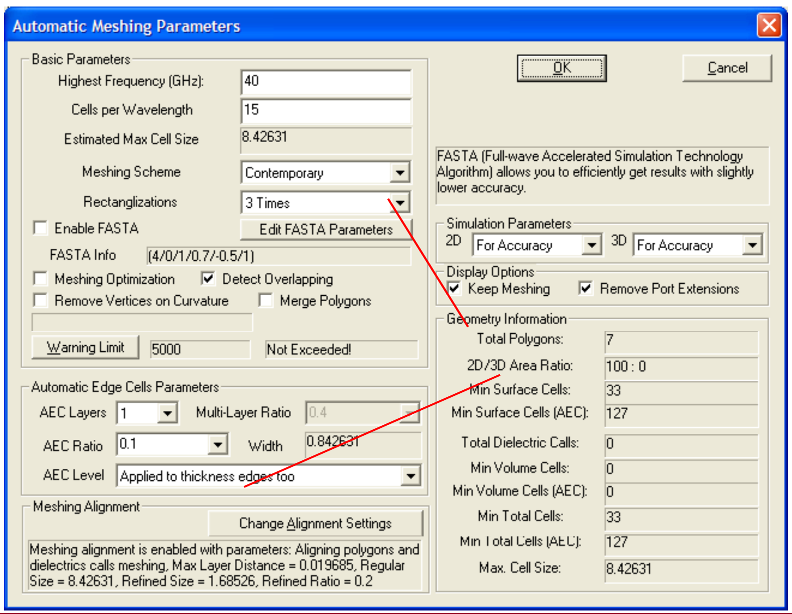


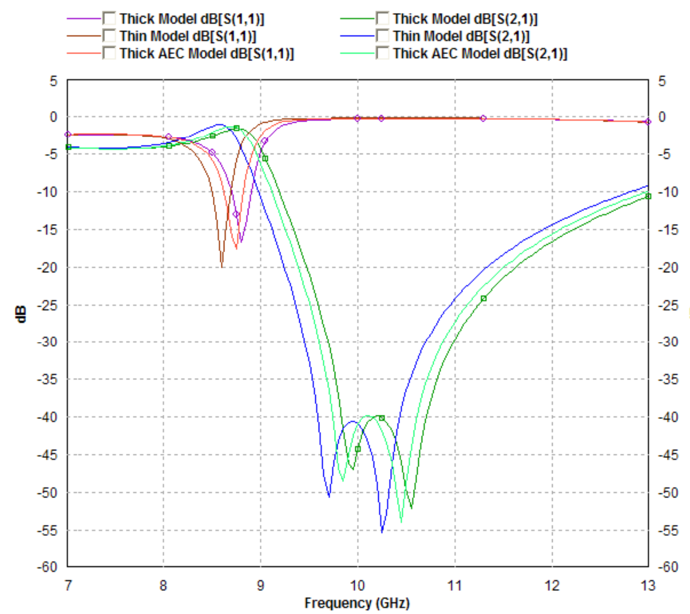
- You can select Save Geom and S-Param to save the polygon based .geo file and the FastEM based s-parameters.. The FastEM based s-parameters are normally good. However, please do simulate the saved structure again if you want to make sure it is accurate. Select Close to send back the optimized object-based geometry to IE3DLIBRARY for further processing.



Double-T Structure with Thickness

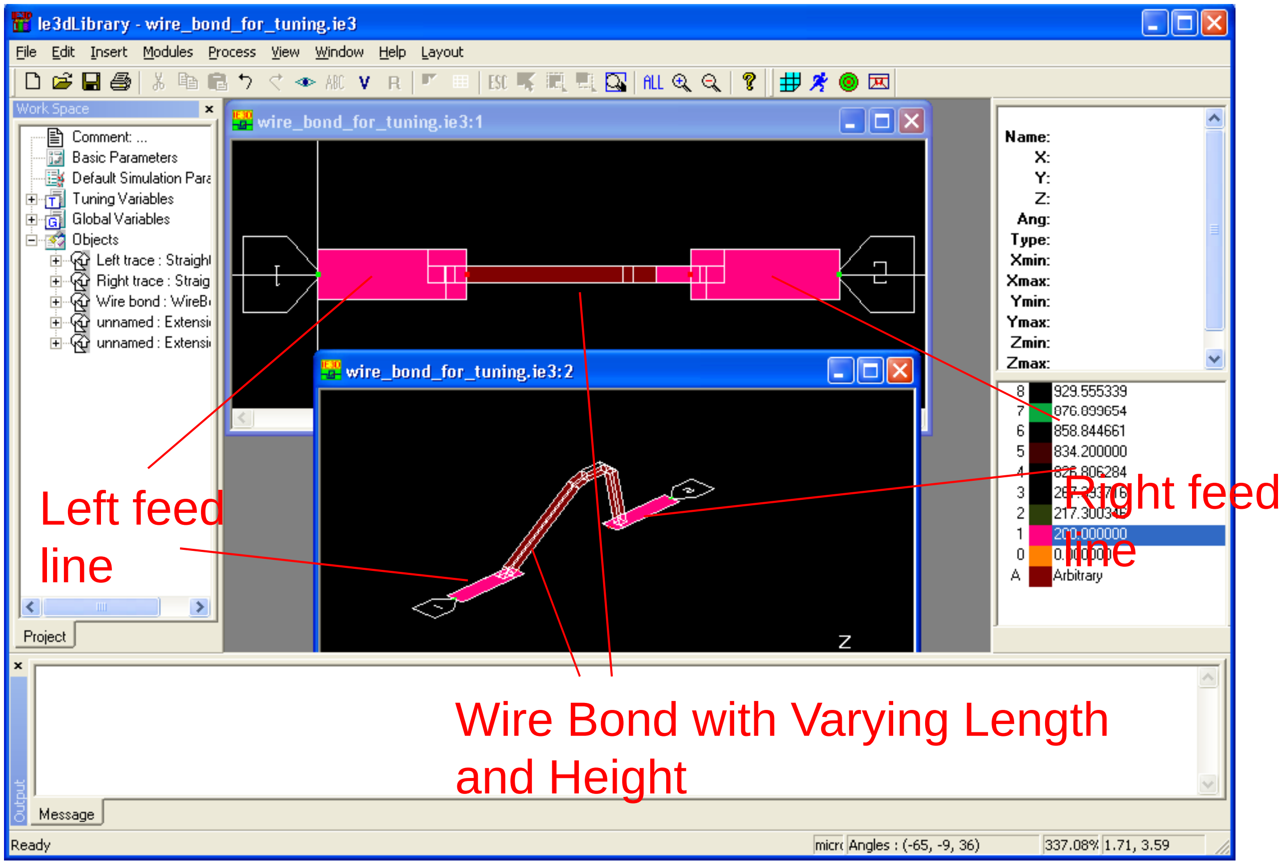
- Saved in file double_tee_tk.ie3 is the thickness model of the double_tee.ie3 structure. Almost everything is the same as the double_tee.ie3. The only difference is in Edit->Basic Parameters dialog. In double_tee_tk.ie3, we have defined the z = 5 mils with thickness growing in +z direction. The trace width of the structure is 5 mils while the thickness is 0.2 mils. Normally, you may not need to use thickness model because the thickness is still much smaller than 10% of the strip thickness. Using ARTT may improve the accuracy by capturing some structure effect of the thickness. However, by default, edge cells will not be created on thickness trace by default. If the thickness is not very big, using thickness model without edge cells may degrade accuracy for some other aspects.
- Saved in file: double_tee_tk_aec.ie3 is the thickness model with Contemporary meshing and AEC Level = Applied to Thickness Edges Too. This model should yield the best accuracy. However, it will require the most computational resources. Normally, we suggest users to use thin model with AEC when the thickness is small, use thick model without edge cells when the thickness is large enough. For best accuracy, we should use the thickness model with Contemporary Meshing and AEC Level = Applied to Thickness Edges Too.





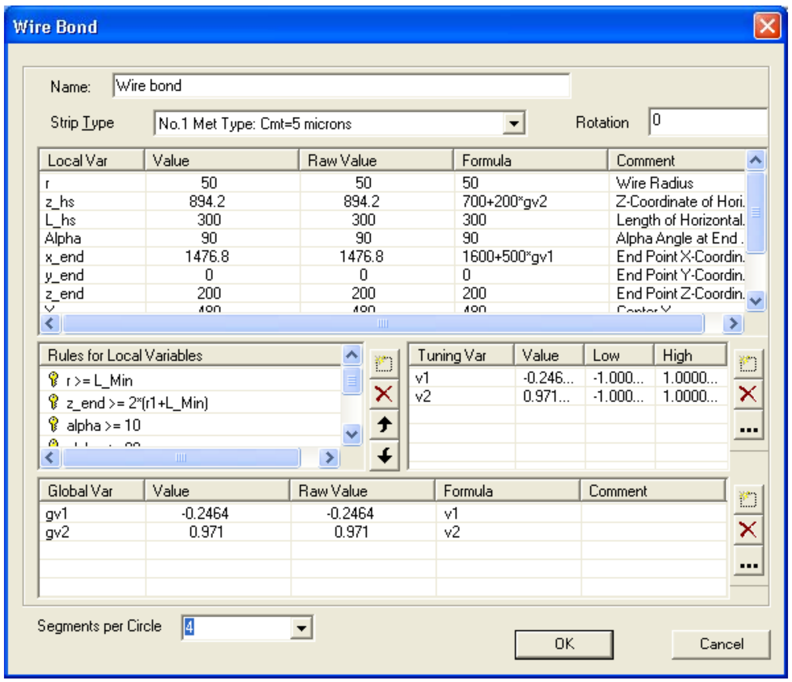
A Wire Bond with Tunable Length and Height (1)

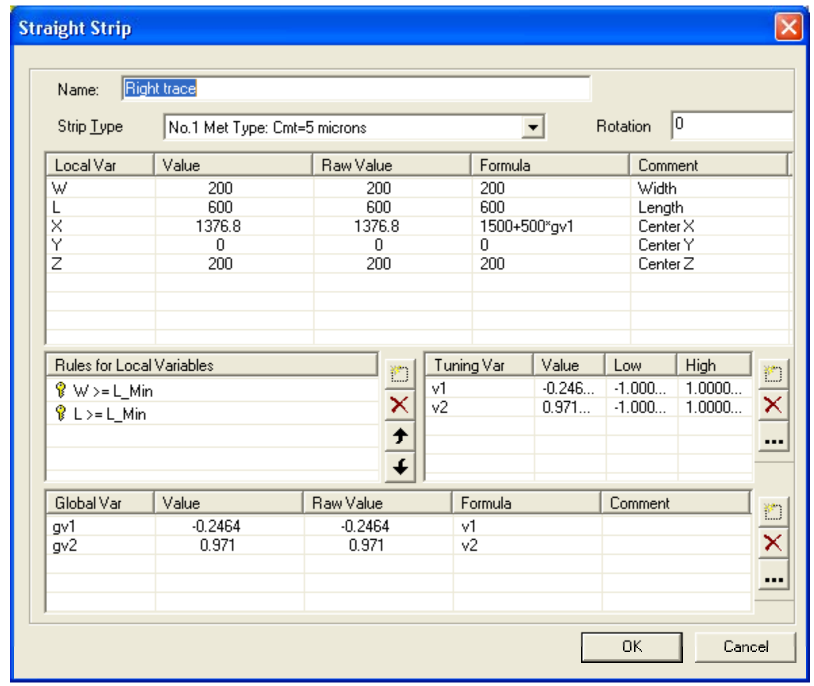
- Wire bonds are used widely in MMIC, RFIC and digital circuits. We will demonstrate how we can parameterize a wire bond structure and obtain a FastEM model on it. Shown below is a wire bond built with 5 objects saved in C:\MentorGraphics\<latest_release> IE3D\SDD_HOME\IE3D\ie3d\ie3dlibrary_examples\wire_bond_for_tuning.ie3 with FastEM data created.



A Wire Bond with Tunable Length and Height (2)

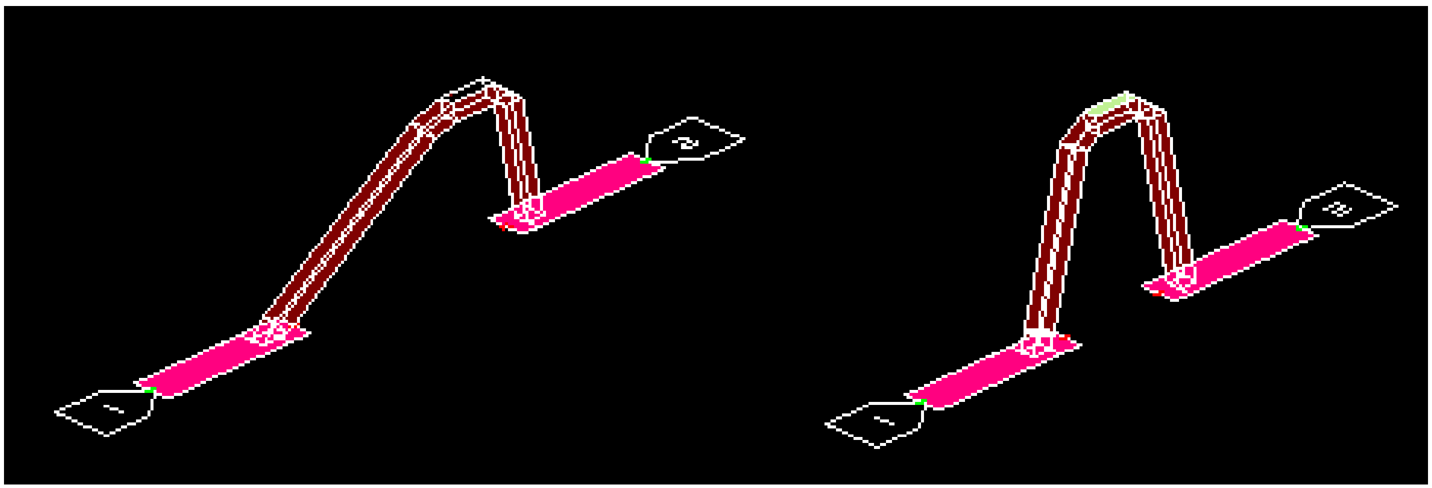
- **Left feed line is a fixed Straight Strip object of 200 um width and 600 um long on z = 200 um. The Wire Bond object is tunable with one end at X = 480 um and the other end at X = 1600 + 500 * gv1 (um). The horizontal segment of the wire bond is at Z = 700 + 200 * gv2. The radius of the wire bond is 50 um and the pads are at Z = 200 um. The gv1 and gv2 are the global variables. They are defined as v1 and v2, and v1 and v2 are the tuning variables. The right feed line is also a Straight Strip object of the same size as the left feed line. However, its X-coordinate is defined as: X = 1500 + 500 * gv1 in order to be moving with the 2nd end of the wire bond.**





A Wire Bond with Tunable Length and Height (3)

- **IE3D V12 is the 1st time we allow EM tuning and optimization of dimensions in the z-direction. If you go to Edit->Geometry Tuning, you can slide the bars and tune the geometry. At the time of this writing, 3D view is not available in the Geometry Tuning and FastEM Tuning while it will be implemented soon. You can use 3D view on MGRID or IE3DLIBRARY to see the changed wire bond geometry. When the wire bond is changing in size and shape, the connection between the wire bond and the feed lines are guaranteed to be electrically connected in IE3DLIBRAY. This is one big step ahead from the IE3D V11 in handling geometry on IE3D. Again, the project contains FastEM Design Kit data. If you do any Geometry Tuning, please remember not to save the project. Please reload the project for FastEM Design in the next.**

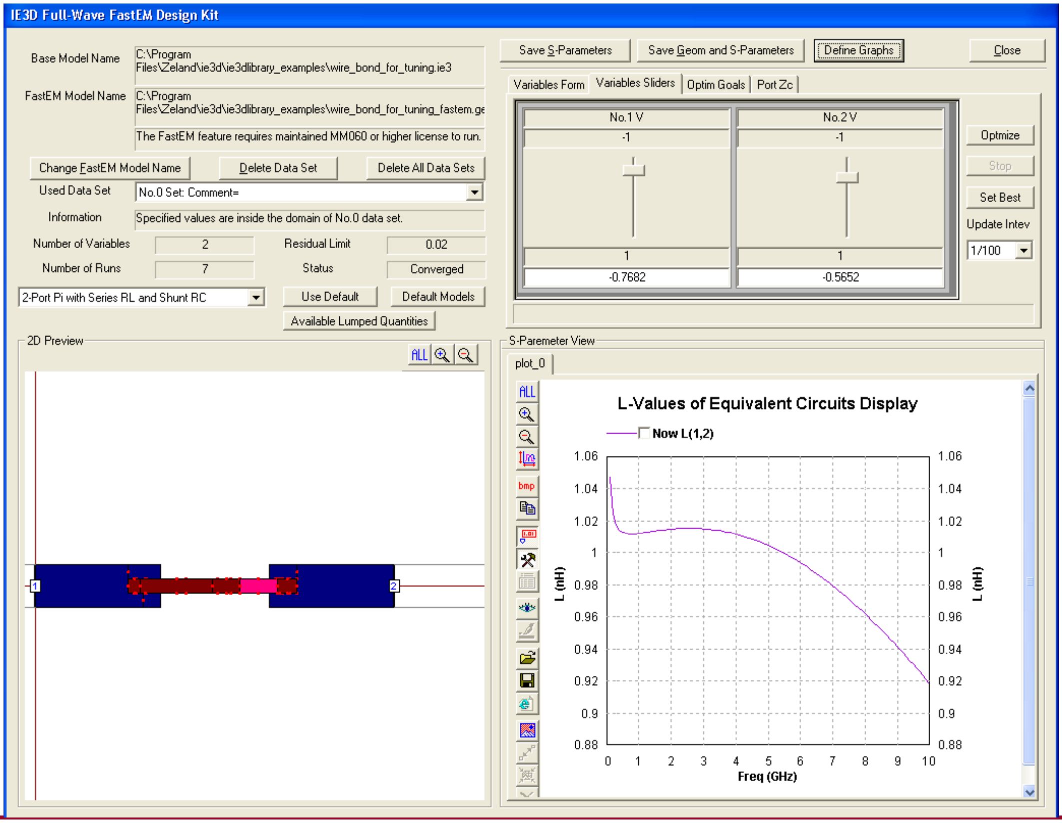


Wire bond of with different lengths and heights created from the same IE3DLIBRARY with different settings of the tuning variables.

A Wire Bond with Tunable Length and Height (4)

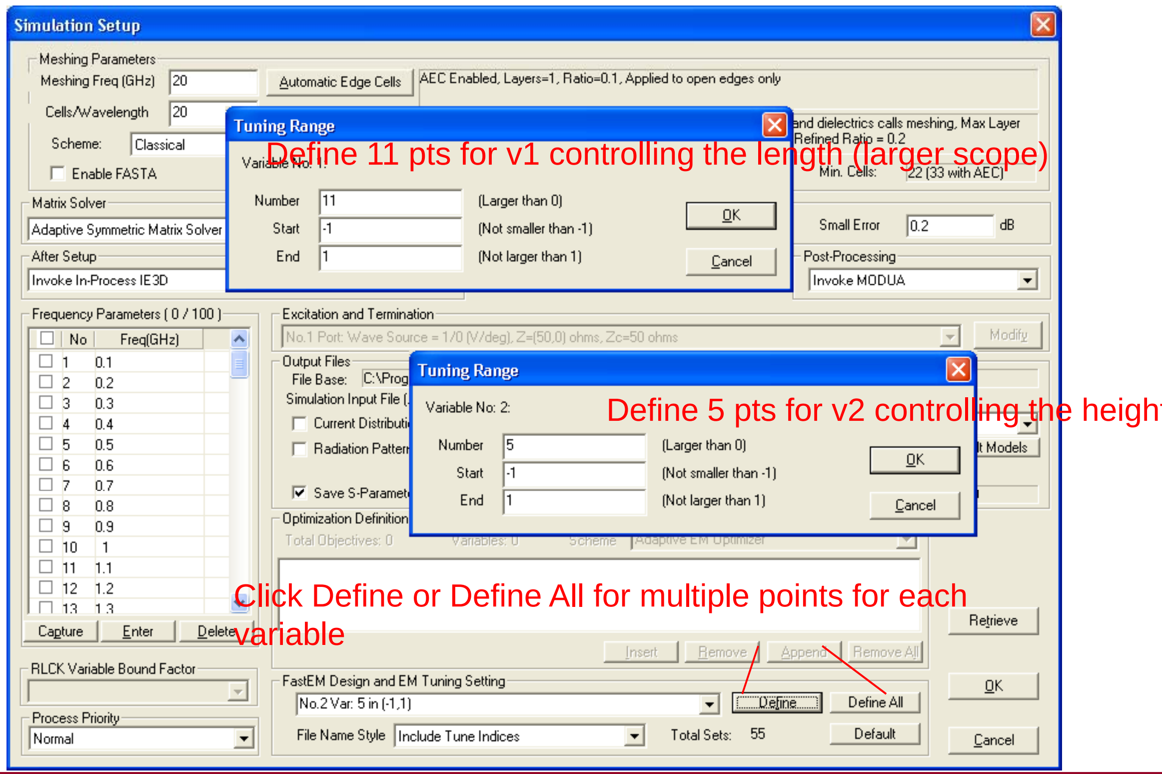
- **Reload the project into IE3DLIBRARY. Select Process->Full-Wave EM Design Using FastEM Design Kit command. IE3DLIBRARY will show you the dialog. By default, s-parameters are displayed. Please select “2-Port Series RL and Shunt RC” for lumped model. Select Define Graph and remove the s-parameter plot. Please define a plot with”L(1,2)”. Select OK to get back to FastEM dialog. Please select the Variable Sliders. Slide the bars and you can see the real-time tuning of L and the geometry.**

- If you define some goals, you can find the best results by either manual tuning or automatic optimization.
- You can select Save Geom and S-Parameters to save the geometry and s-parameters with the specified settings for the v1 and v2. The s-parameters are based upon the FastEM model extracted from IE3D full-wave EM simulation. It should be accurate. However, if you want to make sure, you can always re-simulate the saved geometry to see whether the FastEM results are good.
- You can also plot and optimize Q-value of the wire bond.



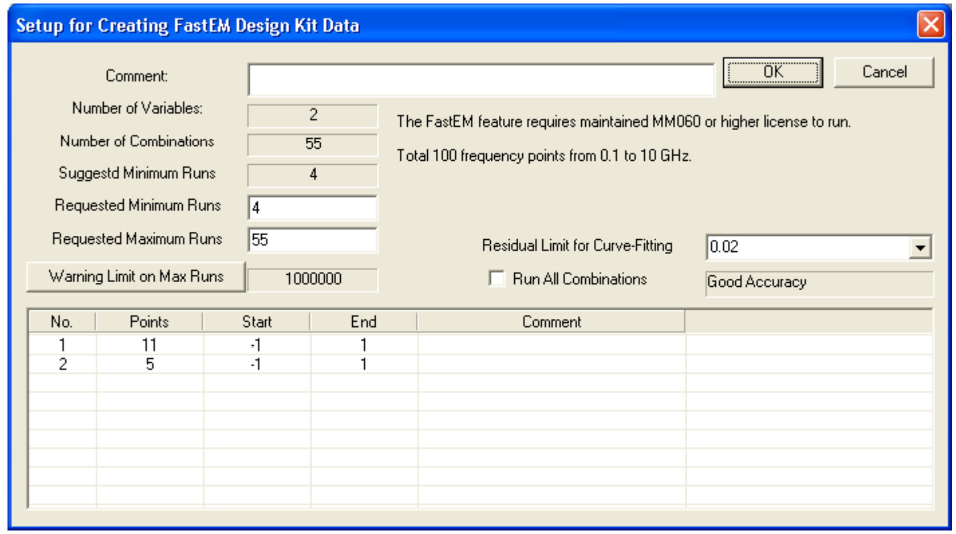
A Wire Bond with Tunable Length and Height (5)

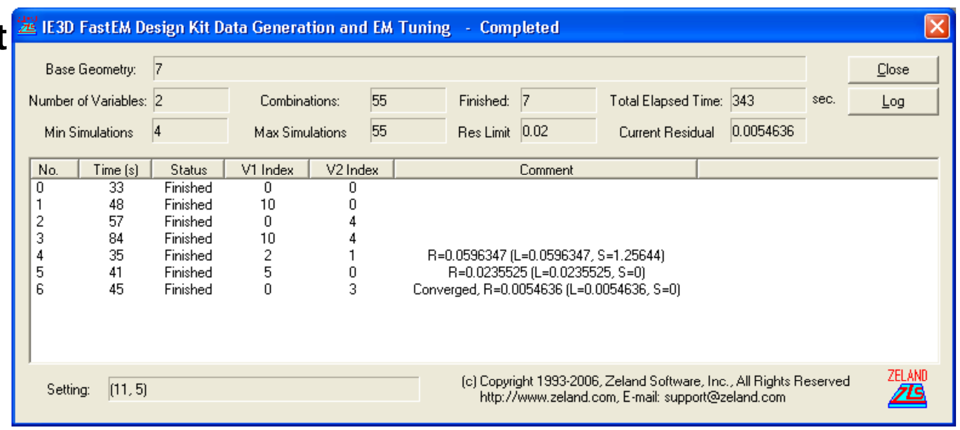
- **We will use this example to demonstrate how to prepare FastEM data after optimization variables are defined. Please pay attention to the FastEM dialog. We are using the No.0 data set and it is the only one. We can create multiple FastEM data sets for the structure if we don't delete them. You can delete some FastEM data sets from the dialog. Let's try to do it. On the FastEM dialog showing the wire bond, please select Delete All Data Sets. IE3DLIBRARY will issue you a warning. Please select Yes to remove the data. FastEM dialog will be closed and it resumes to IE3DLIBRARY window. Please select Process->Simulate.**
  - Enter the frequency range as 100 points from 0.1 to 10 GHz. Select Define or Define All in FastEM section. Define 11 pts for the V1 controlling the length. Length has larger scope and we should define more points. Define 5 points for the V2 controlling the height.
  - After we define the frequency points and the multiple points for each variable, we can select OK and IE3DLIBRARY will prompt you the information about FastEM data preparation.



A Wire Bond with Tunable Length and Height (6)

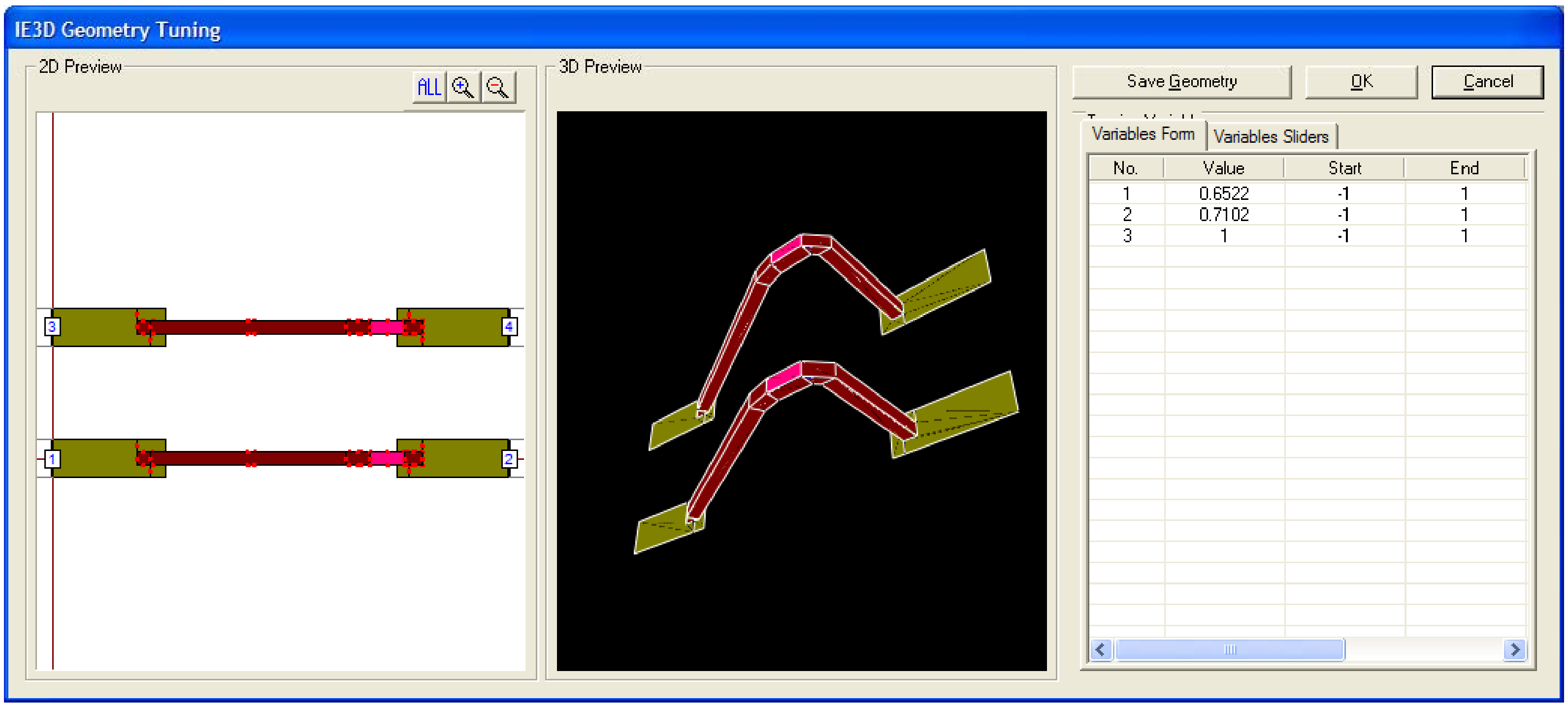
- All information about variables and FastEM is shown in the dialog (on the right). There are 2 variables with 11 pts for v1 and 5 points for v2. There are total 55 combinations. You can specify minimum and maximum runs FastEM. In case, FastEM detects convergence during the preparation, it will automatically stop the preparation without running the rest of the simulations. High accuracy should be guaranteed at those combinations IE3D didn't run. Select OK to run it.
- The IE3D FastEM Design Kit Data Generation and Tuning data comes up. It will invoke IE3D engine to run the different combinations. For this particular example, it finishes the preparation for 7 combinations and it detects convergence. The process takes about a few minutes. For more complicated structures, it usually runs more combinations and it may take much longer time. Select Close. IE3DLIRBARY will save the FastEM data and go to FastEM tuning and optimization directly. We already demonstrate the tuning and optimization earlier and we will not repeat here.





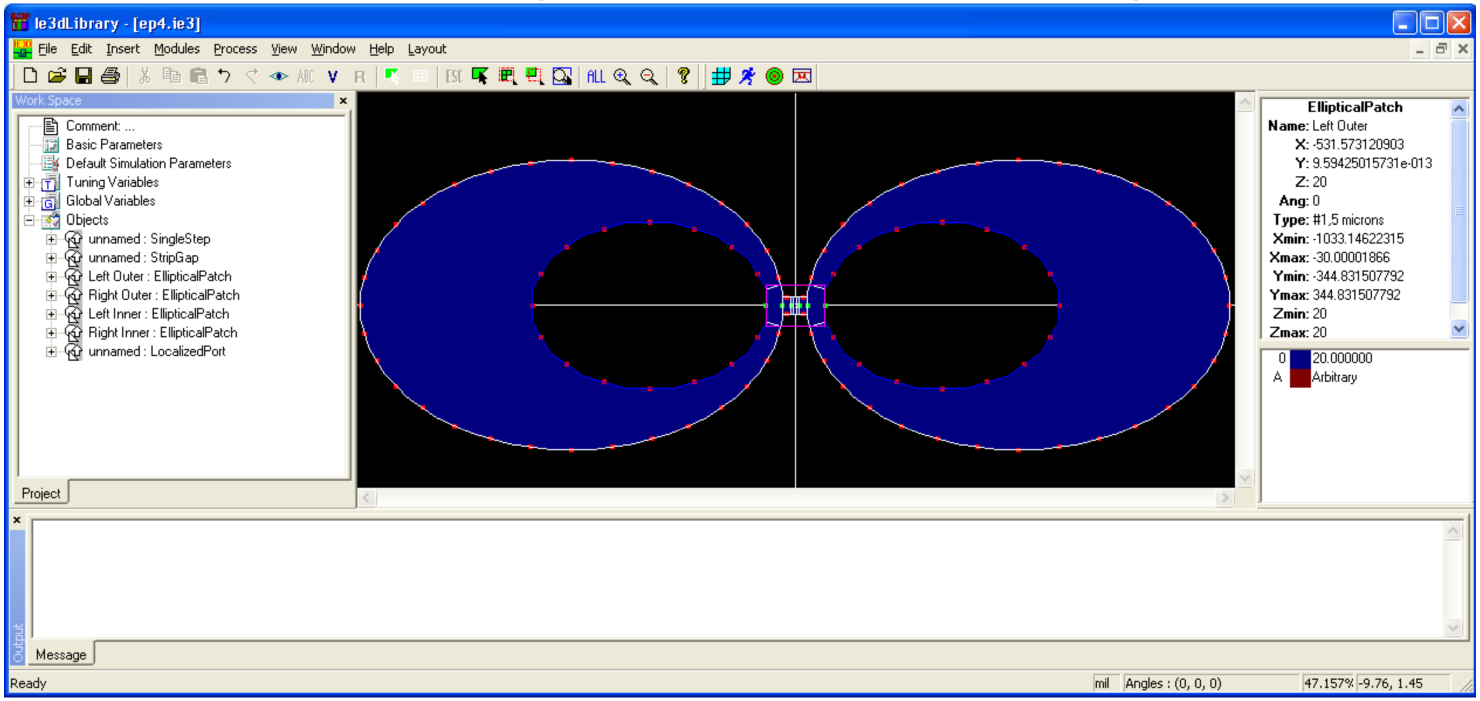
Parameterized Coupled Wire Bonds

- Saved in C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\ie3d\ie3dlibrary _examples\wire_bonds_for_tuning.ie3 is a pair of coupled wire bonds. The length, height and separation of the wire bonds are tunable. FastEM data is created on it.



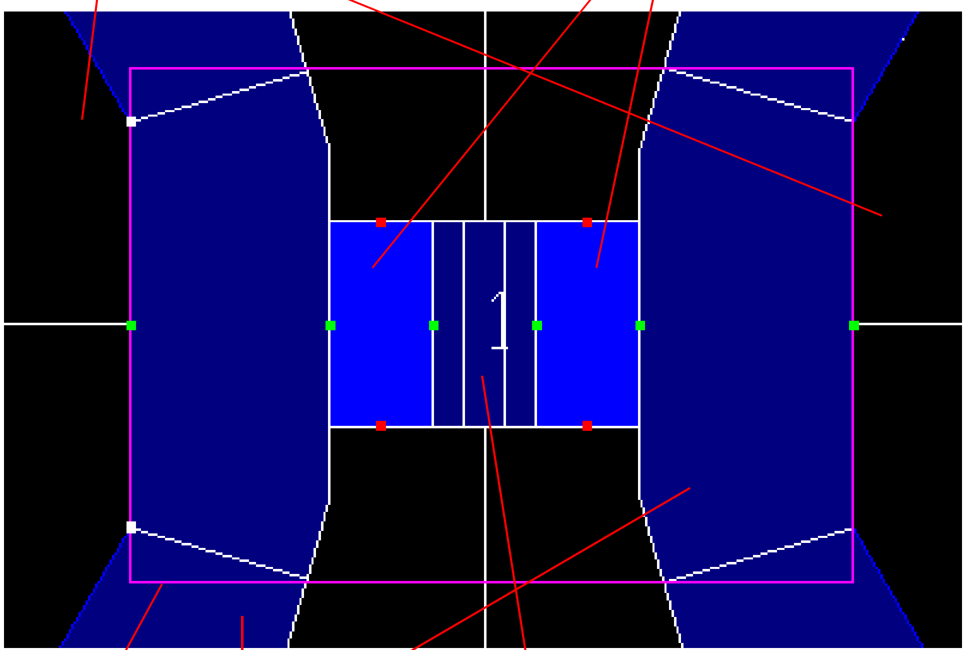
A UWB Antenna with Elliptical Shapes (1)

- Elliptical shaped patch antennas are known to yield wideband frequency behavior. However, it is not a simple task to design a half-wavelength antenna achieving S(1,1) < - 10 dB from 3.1 to 10.6 GHz This example, saved in C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\ie3d\ie3dlibrary_ examples\ep4.ie3, demonstrate how we can achieve it using IE3D FastEM. This example also demonstrate how we can use Boolean objects and void objects to parameterize complicated structure for FastEM tuning and synthesis.
- The antenna is a modified dipole with 2 elliptical poles. We try to dig 2 elliptical holes inside the 2 patches. We are going to optimize the dimensions of the major axis and minor axis of elliptical patches In the meantime, we will also adjust the size and the ratio between the two axes of the inner elliptical shapes. When the variables are changing, we would like to keep each pair of elliptical shapes to be identical. We would also want to keep the minimum distance between each pair to be constant.



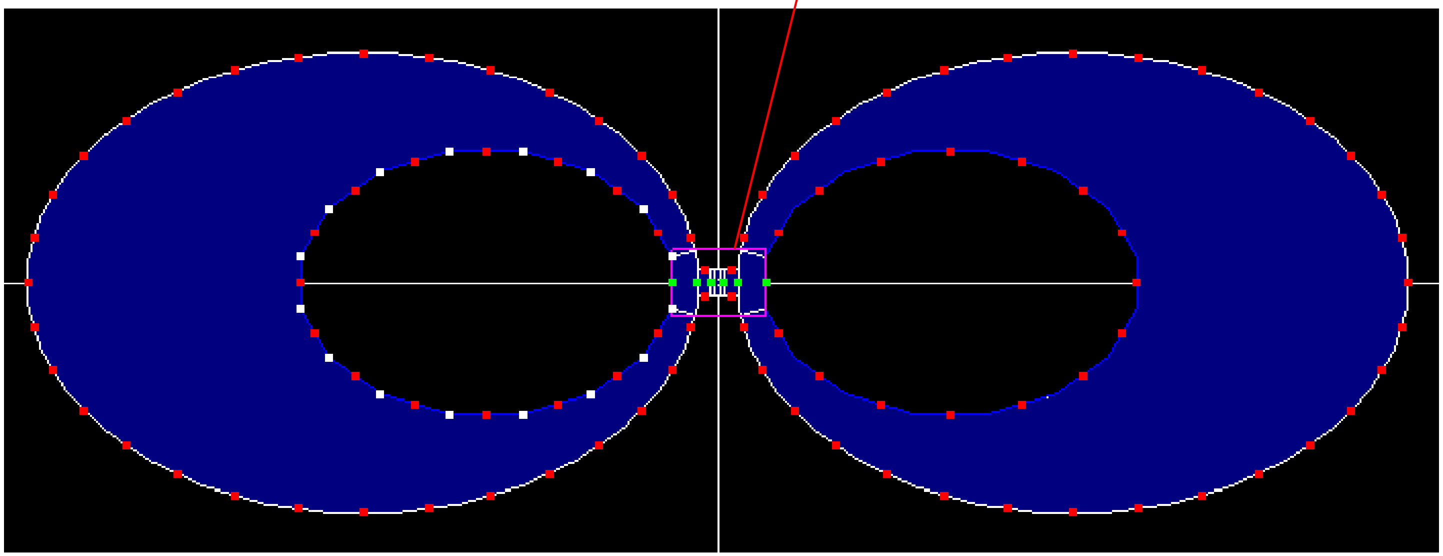
A UWB Antenna with Elliptical Shapes (2)

- **Our 1st step is to choose the right objects. For the 4 elliptical shapes, we can use the Insert->Patch Objects->Elliptical Patch. For the feed between the 2 poles, we will use two objects to create it. One is a Strip Gap and one is a Localized Port. Please see the picture below. The Strip Gap object is purposely colored with light blue in the illustration to identify different objects. The 2 outer terminals of the Strip Gap object are connected to the two larger elliptical shapes. The 2 inner terminals of the Strip Gap object are connected to the Localized Port object.**
  - A Localized Port object is a dependent object defined by 2 terminals. When the terminals are connected to other independent objects, its shape will be fixed. The Localized Port object in IE3DLIBRARY corresponds to either the V-Localized port or H-Localized port depending upon the relative locations of the 2 terminals which can be on different level.
  - When we try to create the port and the 2 elliptical patches. We first create the Strip Gap object. Then, we connect a Localized Port to the inner terminals of the Strip Gap. Finally, we connect the 2 elliptical patches to the outer terminals.



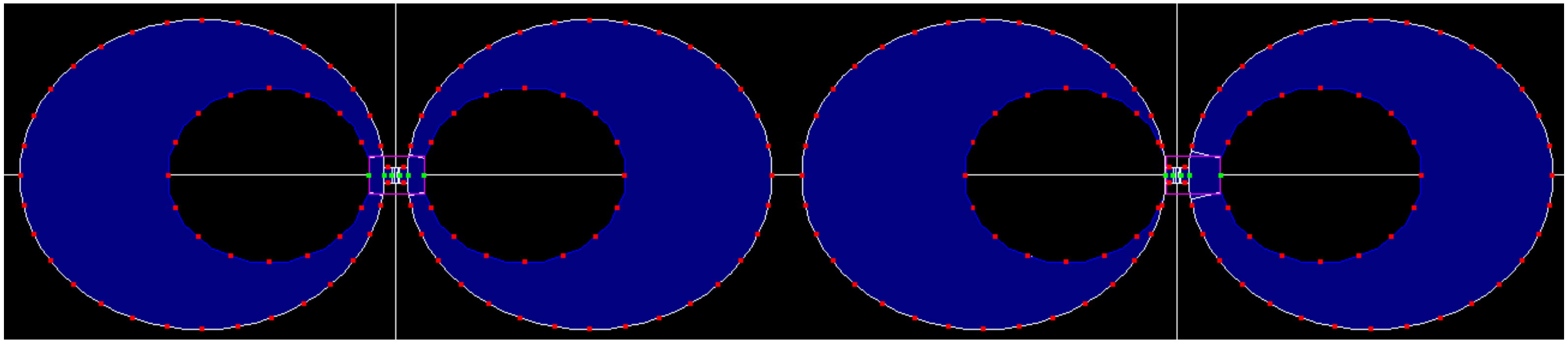
A UWB Antenna with Elliptical Shapes (3)

- **How do we create the 2 elliptical holes with fixed minimum distance between them? There are many ways. Following is a simple way.**
- **We create a Straight Strip object on the same layer as the 2 bigger Elliptical Patch objects. It will be used to locate the 2 elliptical holes.However, we don’t want the Straight Strip object there in the final layout. We define its metallic property as Void Object.**
- **Then, we create the 2 smaller Elliptical Patch objects and connect them to both ends of the void Straight Strip objects. Please remember to define the 2 smaller Elliptical Patch objects as Boolean Object in the metallic strip property. Please check the 4 elliptical objects, their dimensions are equations of 3 global variables while global variables are basically the tuning variables. Again, avoid referencing tuning variables directly in local variables will give you maximum flexibility.**



A UWB Antenna with Elliptical Shapes (4)

- Please be very careful about the order of the objects. We create the Strip Gap object first. We locate it at the center of the origin. Then, we connect the 2 big Elliptical Patch objects to both sides of the Strip Gap object. We define the Localized Port on both internal terminals of the Strip Gap object. It forms a typical dipole like antenna with 2 elliptical arms.
- When we create the elliptical holes, we should first create the void object at the center. Then, connect the 2 Boolean Elliptical Objects to both ends. This is the correct order. If we build one Boolean object first, connect the void object to it and connect the other Boolean object to the void object, we may get unexpected results when we change the tuning variables. Shown below are the cases. All the objects are identical. Object the order of the 1st Boolean object and the void object are swapped. For the correct case on the left, the void object is fixed. The Boolean objects are connected to it. When the variables are changing, the Boolean objects are changing while the void object is not changed. For the incorrect case on the right, the 1st Boolean object is fixed while the void object is connected to it. When tuning variables are changing, the shapes of the 1st Boolean object is changing. The terminal where the void object is connected to is also changing. The void object's location will be changing and it will cause the non-symmetrical shape below.
- You may not see the incorrect shape unless you change the variables. It is a good habit to tune the geometry in Edit->Geometry Tuning before we do any FastEM preparation on the structure.

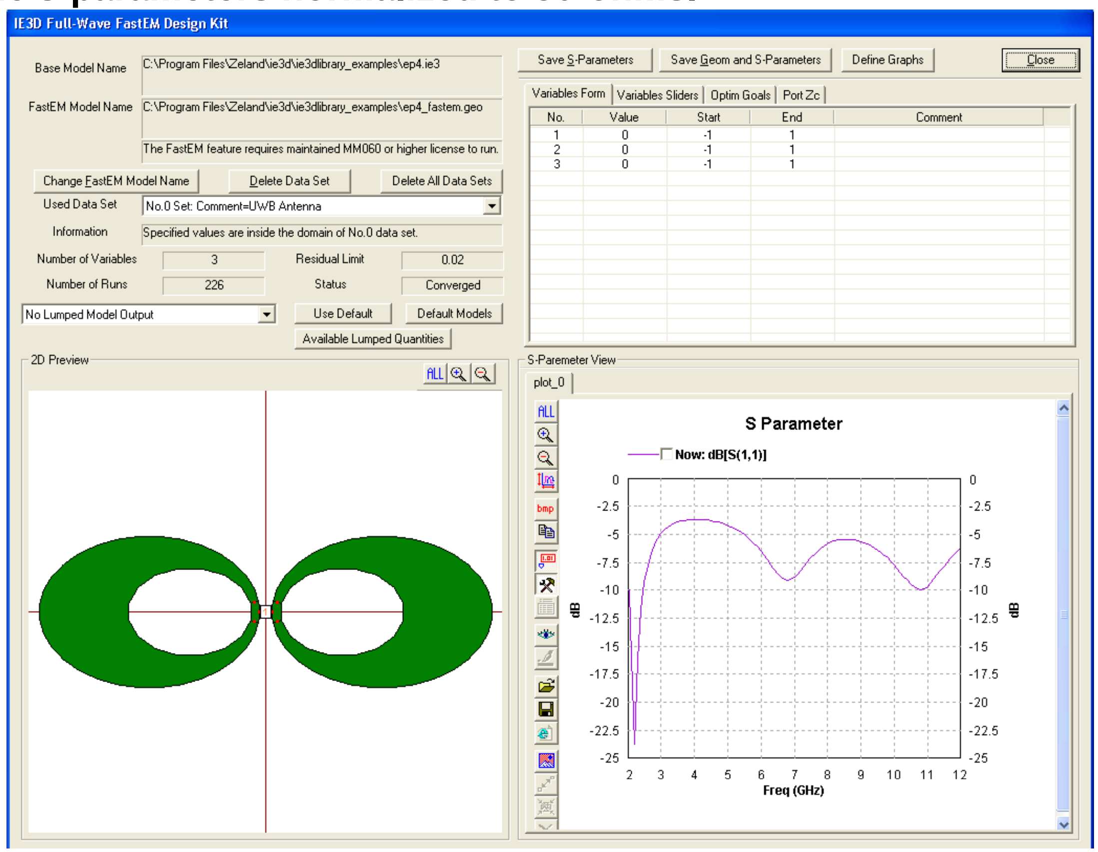


Correct Shape due to Correct Order

Incorrect Shape due to Incorrect Order.

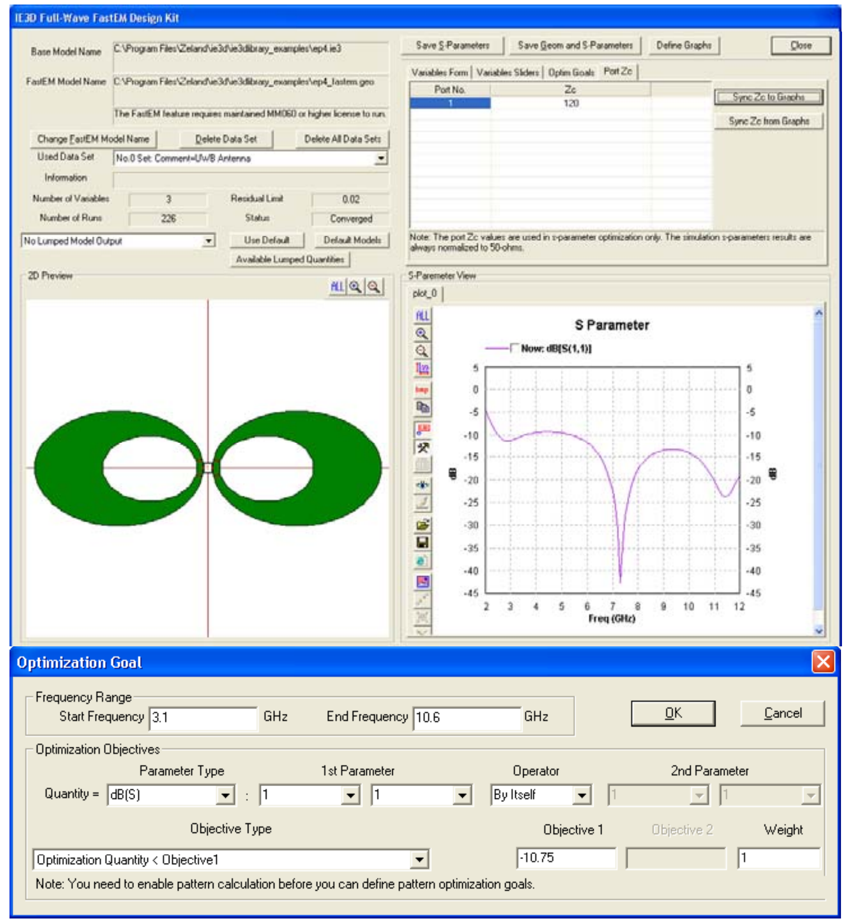
A UWB Antenna with Elliptical Shapes (5)

- **We are going to demonstrate the FastEM design. The FastEM data is already prepared and saved in the file. It takes some hours to prepare it. You may have done some editing on the file. Please don't save the change and reload the ep4.ie3 file to do the FastEM optimization.**
- **Select Process->Full-Wave EM Design Using FastEM Design Kit to bring up the dialog. The default dialog is showing the s-parameters normalized to 50-ohms.**
  - This antenna has wider bandwidth at around 120-ohms. We can go to Define Graph to change the normalization Zc to 120-ohms. Since we are going to optimize the antenna and set the goals based upon Zc = 120, we can also do it in the Port Zc tab next to the Optim Goal tab.
  - Select Port Zc tab. Change the Zc of Port 1 to 120-ohms. Please remember that the Zc defined here will only affect the s-parameters in tuning and optimization. They will not affect the graphs because each graph can be displaying s-parameters of different Zc Please select Sync Zc to Graphs in Port Zc tab to sync them in a 1-time action.



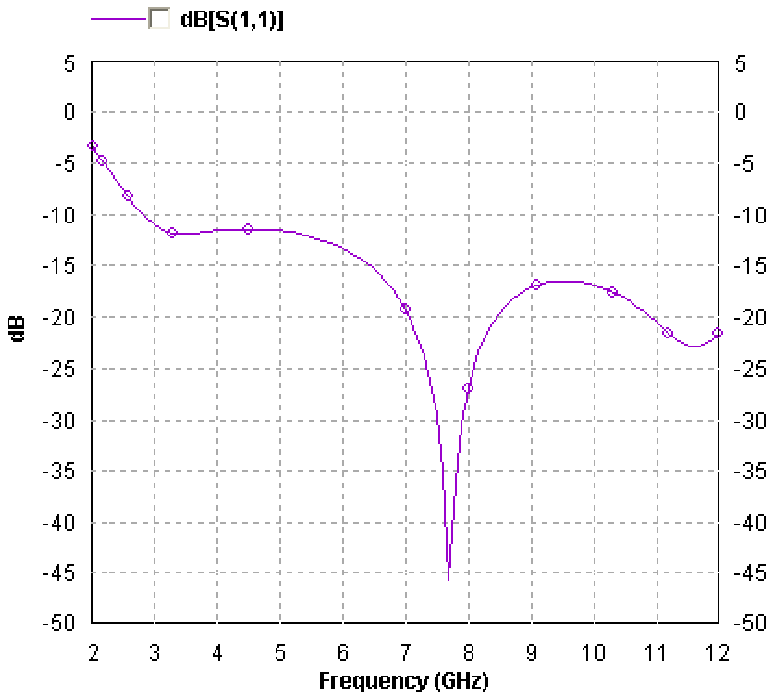
A UWB Antenna with Elliptical Shapes (6)

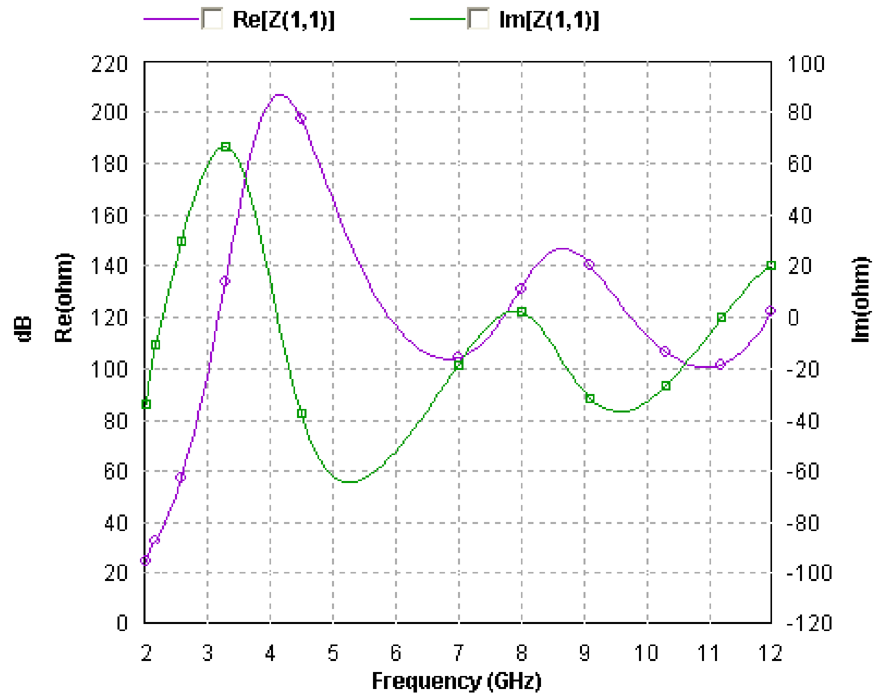
- **After you sync the Zc to graphs, you will see the displayed s-parameters changed. It is still the same data internally. They are just normalized to different Zc.**
- **As you can see, even the s-parameters are normalized to Zc = 120, the dB[S(1,1)] is still not below –10 dB in the frequency range of 3.1 to 10.6 GHz. We need to define the goals and optimize the antenna.**
  - Please select Optim Goals tab. Please select Insert button. Define the goal as dB[S(1,1)] < -10.75 from 3.1 to 10.6 GHz. Select OK. This is the only goal we want.
  - Select the Variable Sliders tab. Slide the bars and we can see how the antenna structure and its s-parameters change with the sliding bars. Select Optimize button. FastEM should be able to find the goal in a short time.
  - For the particular FastEM data in this file, there are some ripples which might be due to not enough points in the variables. Please select Save Geom and S-Parameters to create the optimized IE3D geometry. It will be opened on MGRID. Please re-simulate it on MGRID to check its result. The default s-parameters are normalized to 50-ohms. Please remember to re-normalize it the 120-ohms. You will see the truly high accuracy IE3D results on the optimized structure is very good.



A UWB Antenna with Elliptical Shapes (7)

- **The IE3D verified results of the antenna with great UWB performance.**

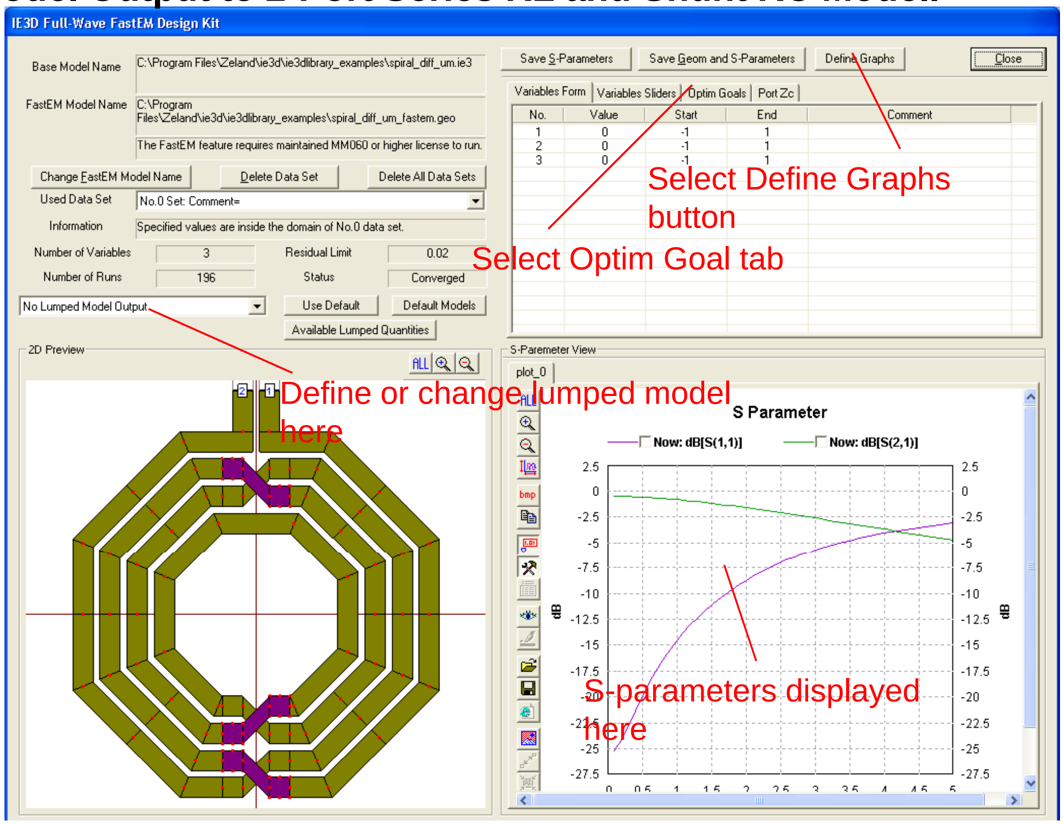




A Spiral Inductor for Synthesis (1)

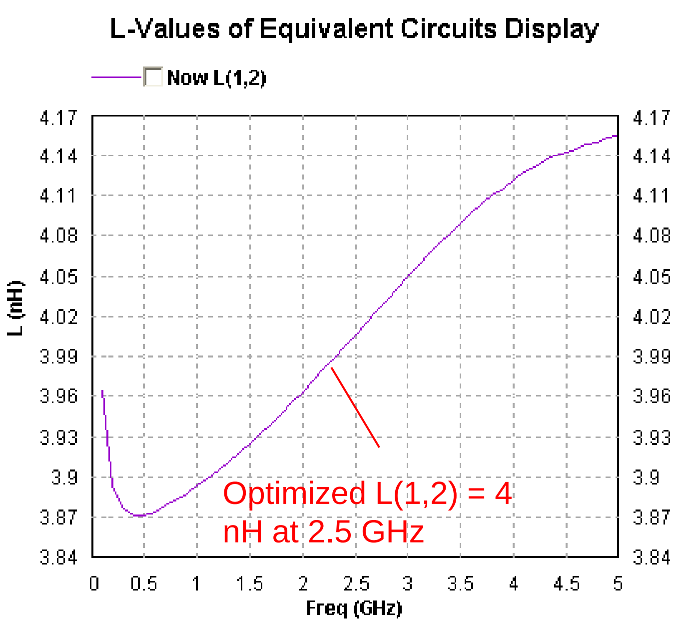
- Please open: C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\ie3d\ie3dlibrary _examples\ spiral_diff_um.ie3. It is a differential spiral created from the SpiralSyn module of IE3DLIBRARY. Please select Process->Full-wave EM Design Using FastEM Design Kit to open the FastEM dialog.
- We are going to demonstrate how we can tune and optimize the L and Q values of the inductor.
- Please change No Lumped Model Output to 2-Port Series RL and Shunt RC Model.

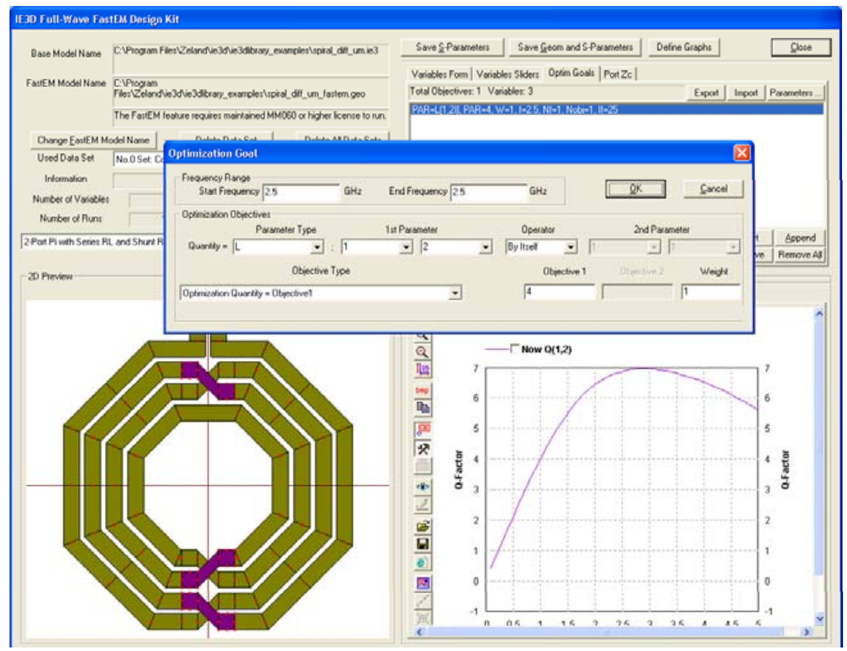
  - Select Define Graph button to define more graphs. Select Add button and select Q-Values of Equivalent Circuit in Graph Type. Select OK. Check Q(1,2) for quantities. Select OK to define Plot_1.
  - Select Add button and select L-Values of Equivalent Circuit in Graph Type. Select OK. Check L(1,2) for quantities. Select OK to define Plot_2.
  - Select Close to get back to FastEM dialog. Select Plot_1 tab to display Q-values and Plot_2 to display L-values.



A Spiral Inductor for Synthesis (2)

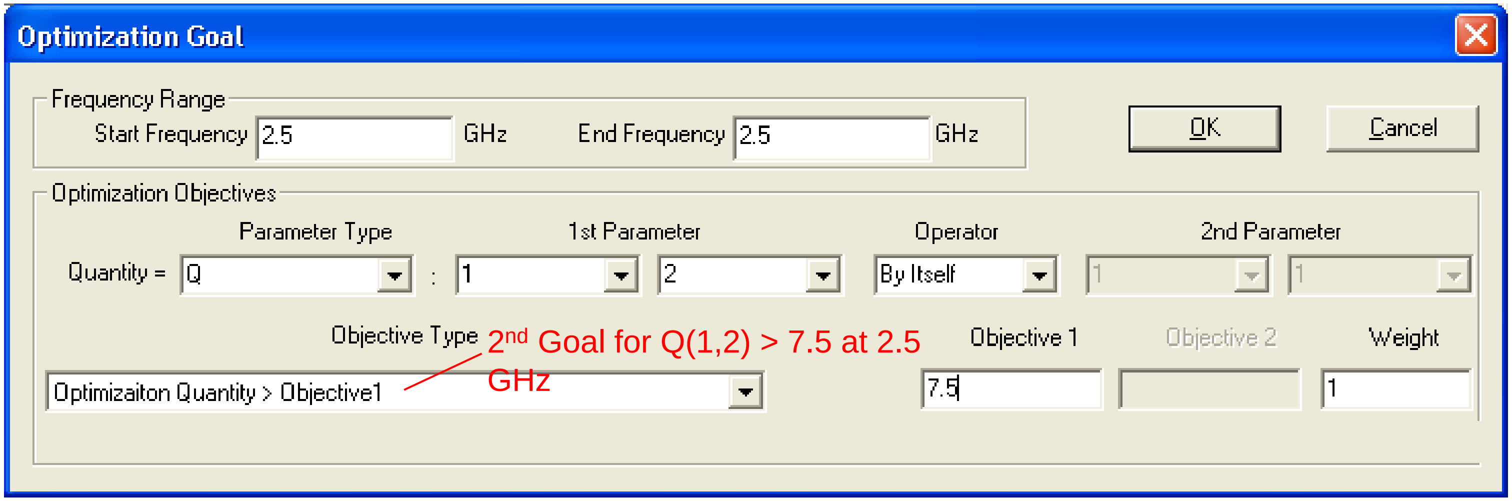
- Select Optim Goals tab. Select Insert button in Optim Goals tab. Define Start Freq = 2.5 GHz and End Freq = 2.5 GHz. Define the 1st goal as L(1,2) = 4 nH at 2.5 GHz. Select OK to add the goal into the list.
- We can define more goals if we like. Let's see how it works with one goal. Select Variable Sliders tab. Slide the bars. FastEM tries to find the variable values to match L(1,2) = 4 nH at 2.5 GHz.
- Select Optimize button in Variable Sliders tab. FastEM should be able to find the point for L(1,2) = 4 nH at 2.5 GHz in seconds. We have 3 variables to control the size, the strip width and the strip gap. There should be many points where we can achieve L(1,2) = 4 nH at 2.5 GHz. At our optimized point, the Q-value is about 6.7. We will demonstrate how we can optimize both L and Q simultaneously.





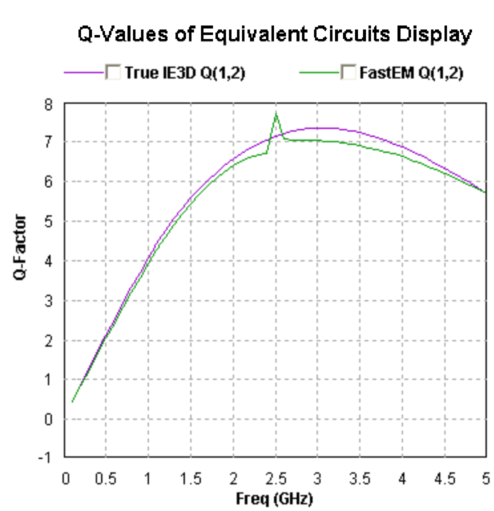
A Spiral Inductor for Synthesis (3)

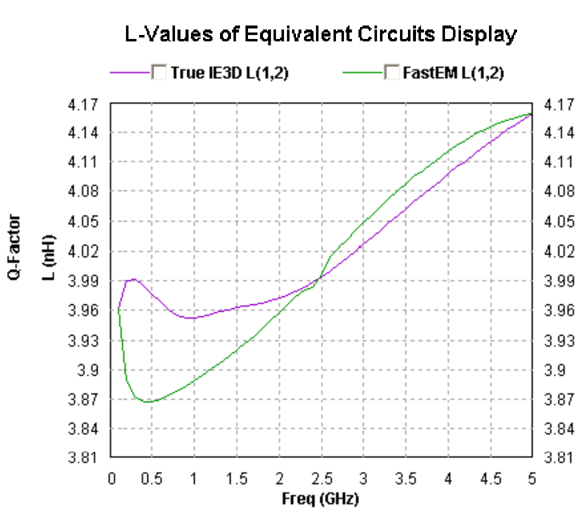
- Select Save Geom and S-Parameters to save the current geometry first.
- Select Optim Goals. Add another goal as Q(1,2) > 7.5 at 2.5 GHz. We don't know how big Q can go. We just enter some value and see how much it can do. Certainly, we should not enter a value much larger than the possible goal. Otherwise, it will try to balance the requirement for L(1,2) = 4 nH and goal for Q(1,2). It is possible the optimized value of L(1,2) may not be close to 4 nH because the other goal on Q(1,2) may draw too much attention. You can use weight to balance between them.
- Try to optimize it again for L(1,2) = 4 nH and Q(1,2) > 7.5 at 2.5 GHz. FastEM will finish the optimization process in a short time.
- Please save the optimized geometry and the FastEM s-parameters. Please save the optimized geometry into another file and simulate the optimized geometry for the true IE3D result. The reason for saving the file into another name is that we want to keep the FastEM result and we will compare the true IE3D results and FastEM result to see how good they are.

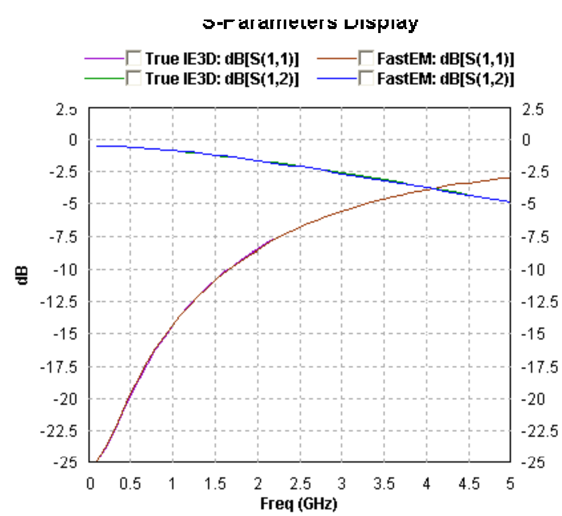


A Spiral Inductor for Synthesis (4)

- **You can compare the s-parameters, L and Q-values of the true IE3D simulation results and the FastEM results on MODUA (command: Process->General Lumped Equivalent Circuit). The comparison is shown in the Figures below. As you can see, the s-parameters match perfectly. There are some slight difference in the L and Q curves. The difference looks large while it is small because of the small range for the values. There is a small peak in the Q-value of the FastEM data. This should be the “noise” in extracting the FastEM signature. L and Q values are much more sensitive than the s-parameters. As they are, the FastEM predicted L and Q values are quite good even though they may not be perfect. That is also the reason why you are always suggested to perform the IE3D simulation on the tuned or optimized structure especially when the designs are critical.**
- **IE3D Fast EM can also be used to tune or optimize transformers and capacitors in RFIC. We will not discuss any example here. Interested users can try it.**

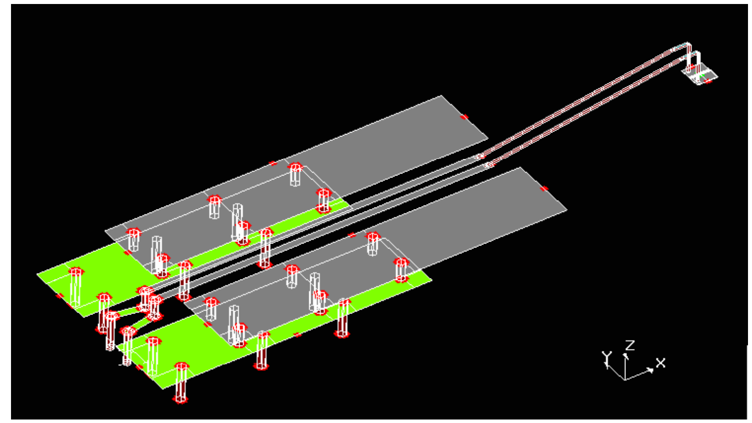


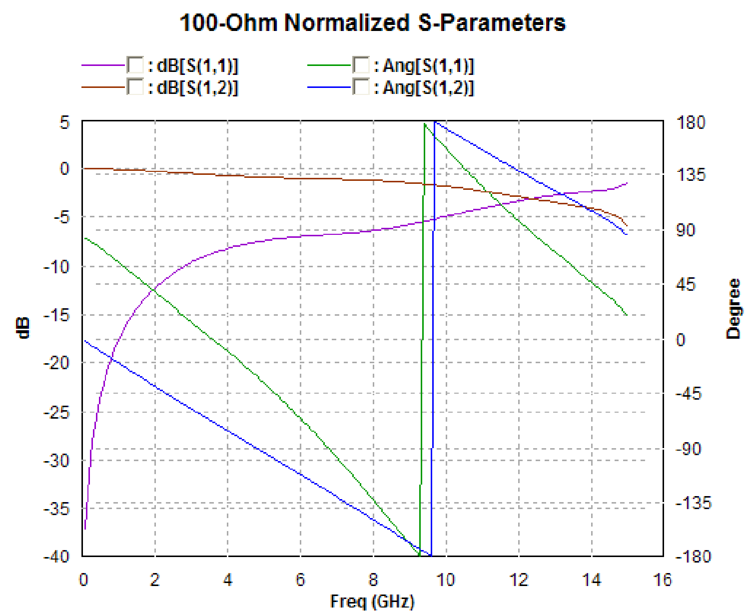




A Differential Pair on PCB and Packaging

- Saved in C:\MentorGraphics\<latest_release>IE3D\SDD_ HOME\IE3D\ie3d \ie3dlibrary_examples\pcb.ie3 is a typical differential pair including thickness traces, solder balls, vias, finite ground planes and wire bonds. The solder balls from z = 0 to 500 are modeled as some Cylindrical Vias objects.
- The structure is not parameterized even though it is not difficult to parameterize it. The most important feature to demonstrate in this example is the Boolean operations capabilities of IE3DLIBRARY. Experienced IE3D users should know that it is not easy to change the dimensions and coordinates on the polygon-based layout editor MGRID. On MGRID, Boolean operations are done on polygons. The dimensions and coordinates of polygons must be valid numbers when Boolean operations are performed. They can't be equations. For such a reason, it is very difficult to change the dimensions and coordinates after an advanced geometry operation involving Boolean operations. On IE3DLIBRARY, we can change the dimensions and coordinates of vias, wire bonds and other geometry parameters anytime.
- IE3DLIBRARY basically tries to re-build the vias and wire bonds and other things after every change you make. It may involve much computational effort. This example involves 26 vias and 2 wire bonds. It may be slow during the operation of geometry modeling and other editing process when re-building of the geometry is involved. It is more convenient to select, copy, paste and re-locating objects inside Edit->Objects Properties.





Suggestions on Using FastEM

- **Try to use fewer variables if possible. The number of simulations involved in a FastEM data preparation increases very fast with the number of variables.**
- **Try to limit the range of each variable and try to use fewer grid points for each variable. Again, the number of simulations will increase very fast with the grid points. If you have one variable, you can define 20 or even 100 grid points for the variable. If you have two variabels, you can define 10 or even 20 grid points for each variable. If you have three variables, try not to use more than 10 grid points for each variable. Normally, please try to limit the Requested Maximum Runs to be within 1000. When the number is getting big, it will be very slow.**
- **Try to use fewer frequency points. Advanced curve-fitting schemes are used to detect the convergence of sampling points. Too many frequency points will increase the RAM requirement and make it tougher for a complete convergence. It may increase the simulation time significantly.**
- **Try to reduce the electrical length of the structure you want to extract the FastEM model. Smaller electrical length will make the convergence much faster.**

Summary

- The object-based layout editor IE3DLIBRARY is quite different from the polygon-based layout editor MGRID.
- It is used like a schematic editor and it does not need many commands. It should be easier to use than MGRID.
- IE3DLIBRARY is well-planed and implemented with flexibility and ease of use in parameterization.
- Geometry editing on IE3DLIBRARY is basically building a script graphically. The script can be changed anytime. Internal re-running of the script allows re-creation of the geometry with different parameters.
- IE3D FastEM is implemented into IE3DLIBRARY make the combination very powerful for creating full-wave EM models, performing full-wave EM tuning, optimization and synthesis.
- MGRID and IE3DLIBRARY are complementary. You can achieve some goals much easier on MGRID and you can achieve some other goals much easier on IE3DLIBRARY. MGRID and IE3DLIBRARY can work together for the maximum flexibility. MGRID contains matured technology supporting the advanced features on IE3DLIBRARY. IE3DLIBRARY allows high level geometry modeling and much improvement will be done in the near future.

Third-Party Information

This section provides information on open source and third-party software that may be included in the HyperLynx 3D EM product.

- This software application may include Boost version 1.42.0 third-party software. Boost version 1.42.0 is distributed under the terms of the Boost Software License version 1.0 and is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the license for the specific language governing rights and limitations under the license. You can view a copy of the license at: <your_Mentor_Graphics_documentation_directory>/legal/ boost_1.0.pdf. Portions of this software may be subject to the Boost Artistic License. You can view a copy of the Boost Artistic License at: <your_Mentor_Graphics_documentation_directory>/legal/boost_artistic_2000.pdf. Portions of this software may be subject to Apache version 2.0 license. You can view a copy of the Apache version 2.0 License at: <your_Mentor_Graphics_documentation_directory>/legal/apache_2.0.pdf. Portions of this software may be subject to jQuery which is dual licensed under the MIT license or the GNU General Public License. We elect to distribute jQuery under the MIT license. Boost version 1.42.0 may be subject to the following copyrights:

  © 2001-2003 William E. Kempf

  © 2001-2003 David Moore, William E. Kempf

  Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. William E. Kempf makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

  © 1994, 2002 Hewlett-Packard Company

  Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Hewlett-Packard Company makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

  © 1996, 1997, 1998, 1999 Silicon Graphics Computer Systems, Inc.

  Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Silicon Graphics makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

  © 2001 Ronald Garcia

  Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Ronald Garcia makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

  © 2001 Jeremy Siek

  Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Silicon Graphics makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

  © 2000 Jeremy Siek, Lie-Quan Lee, and Andrew Lumsdaine

  Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation. We make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.