Table of Contents

Ansys

2025 R2

POWERING INNOVATION THAT DRIVES HUMAN ADVANCEMENT

EMIT Help

ANSYS, Inc.
Southpointe
2600 Ansys Drive
Canonsburg, PA 15317
ansysinfo@ansys.com
https://www.ansys.com
(T) 724-746-3304
(F) 724-514-9494

Release 2025 R2
July 2025

ANSYS, Inc. and ANSYS Europe, Ltd. are UL registered ISO 9001:2015 companies.

1 - Welcome to EMIT Help

The Electromagnetic Interference Toolkit (EMIT) design type in Ansys Electronics Desktop provides system simulation for predicting and mitigating radio frequency interference (RFI) in electronic devices. EMIT provides:

- A framework for managing all of the information that is required to define complex radio frequency (RF) environments
- A powerful computation engine for computing interference throughout the entire scenario including higher-order effects such as intermodulation products
- Dynamic diagnostic and result views for rapid analysis of the simulation results to quickly and intuitively identify the "root-cause" of all interference problems and to visualize interference signal paths

EMIT is designed to work with different types of input data that describe the RF environment to be modeled, thus enabling RFI simulations to be performed with the data that is available to the analyst.

EMIT's power as a simulation tool for predicting RFI between RF systems lies in its unique multi-fidelity approach to modeling complex scenarios using whatever information is available. Lower fidelity models in EMIT for transmitters (Txs) and receivers (Rxs) are based on typically available high-level radio specifications, such as maximum spurious emission levels (for Txs), minimum discernible signal (for Rxs), and RF & IF filter bandwidths and frequencies. EMIT uses these RF system specifications to create radio models that are then combined with wideband antenna-to-antenna coupling data obtained from HFSS or HFSS 3D Layout to compute the interference metrics (EMI Margin, Sensitivity, Desense, and Availability) for the complete system. The interference metrics provide the analyst with an evaluation of the level of interference in each Rx as well as the root cause of the interference that is easily identified using EMIT's integrated diagnostic tools.

As more information is obtained about the performance of the Tx and Rx systems, it can be introduced into the existing lower fidelity EMIT models in order to yield more accurate representations of the radios in EMIT. That, in turn, leads to more accurate EMI Margin predictions. At the highest level of fidelity, EMIT can use models derived from measurements that characterize the Tx and Rx systems over a wide bandwidth, or from simulations of the entire radio performed in sophisticated nonlinear RF circuit simulators like Ansys Nexxim. These types of high fidelity models capture completely the complex wideband behavior of the radios, leading to highly accurate EMI Margin predictions within EMIT.

Detailed information on the underlying methodology and equations used by EMIT can be found in the EMIT Theory section of the help.

Migrating from EMIT Classic

> **Important:**
>
> The Import from Classic EMIT feature will be deprecated at release 2026 R1. To ensure a smooth transition, please convert all projects to the AEDT format prior to that time.

EMIT has existed as a standalone product ("EMIT Classic") for many years. To help facilitate the transition to EMIT in EDT, several toolkits have been created that will import EMIT Classic projects and libraries into the new format. For details on using these toolkits, see: Importing EMIT Classic Projects and Libraries.

Tutorials

Several tutorials have been designed to introduce users to basic RFI concepts and to help familiarize them with EMIT's functionality. These tutorials start with the creation of simple RFI scenarios and progressively add more complex capabilities and features.

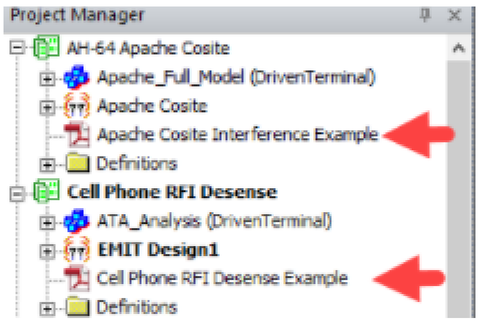
1. **Tutorial 1** – Users will learn how to create an EMIT Design, how to create a simple scenario with one transmitter and one receiver, and how to analyze the RFI between the systems. The resulting RFI will be mitigated using a transmit filter.
2. **Tutorial 2** – Builds on Tutorial 1 by adding multiple channels and random sampling to the systems and introduces users to more interactive results analysis features, such as Availability.
3. **Tutorial 3** – Expands upon the prior tutorials to include multiple transmitters and receivers. Users will also learn about EMIT's amplifier model and the N-to-1 analysis mode. These features combine to allow users to model intermodulation products generated from transmitter-to-transmitter coupling.
4. **Tutorial 4** – This is a standalone tutorial that demonstrates the use of CAD in EMIT and explores some of EMITs parametric coupling models.
5. **Tutorial 5** – This is a standalone tutorial that demonstrates how to create a receiver susceptibility model. The susceptibility model will be defined based on the details of the WiFi 6 standard.
6. **Tutorial 6** – Builds on Tutorial 4 to further explore the EMIT Elements Library and learn how EMIT's simulation engine handles larger scenarios.
7. **Tutorial 7** – This is a standalone tutorial that demonstrates how to create a detailed EMIT radio model from a typical manufacturer's specification sheet.
8. **Tutorial 8** – This is a standalone tutorial that demonstrates an IoT module and explores personal library models and EMIT's Desense results metrics.

The Examples directory for EMIT also includes the following pre-built examples:

- AH-64 Apache Cosite
- Cell Phone RFI Desense

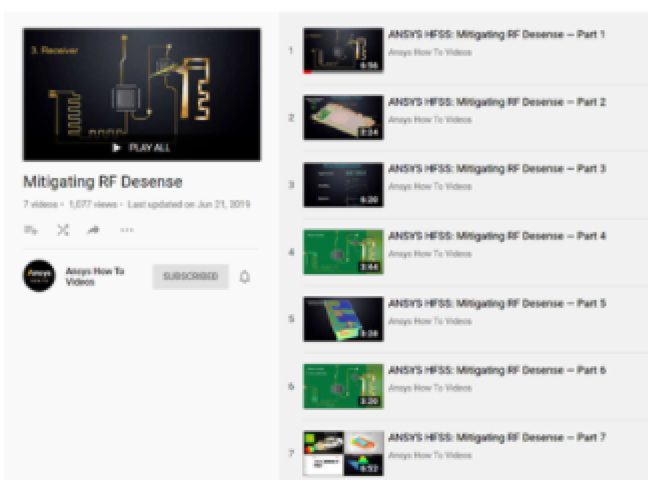
To access these files from within Electronics Desktop:

1. Click **File** > **Open Examples** and open the EMIT folder.
2. Select either *AH-64 Apache Cosite.aedt* or *Cell Phone RFI Desense.aedt*.
   After you load the project, the example is shown in the Project Manager, along with a detailed PDF guide that is accessible from the Project Manager window:



Additional Resources

An EMIT-specific video tutorial titled *Mitigating RF Desense* can be accessed from the Ansys YouTube channel. This tutorial corresponds to the material provided in the Examples installation directory (C:\Program Files\ANSYS Inc\v252\AnsysEM\Examples\EMIT\Tutorials). The IoT_ Board_Desense.pdf document in this directory also provides an overview of this example.



EMIT Native File Formats

The following are EMIT's native file formats:

- RF System Characteristic Files
- Spectral Profile File Format
- Antenna Pattern File Format

RF System Characteristic File

Emit can import the following types of files describing transmitter emissions and receiver susceptibility:

- Raw Channel Data (*.dlxch)
- Spectral Component File (*.meas)

Raw Channel Data

Raw channel data is information from a standard channel measurement system that EMIT can import as Measured Channel data. Currently, only spectrum analyzer data for a transmitter is supported. The data is supplied to EMIT in text files using the *.dlxch file format. When *.dlxch files are imported, the data is automatically converted and stored as a *.meas file in the project directory.

Raw channel data files contain frequency-power pairs with the frequency specified in Hz and the power specified in dBm. The file consists of a header and two columns of data that must be separated by a tab or a space (not a comma). A brief header is necessary to specify information about the channel and the measurement configuration. The data is arranged in two columns (frequency and power) after the header. The header info should be in the following form:

```
@ DlxDataType = "TxSpecAnData"
@ ChannelFreq = 100e6
@ SpecAnResBw = 30e3
@ Begin Data
```

Each line in the header is as follows:

@ **DlxDataType** – defines the type of data in the file. For transmitter data from a spectrum analyzer, this should be "TxSpecAnData".

@ **ChannelFreq** – defines the frequency of the measured transmit channel, in Hz.

@ **SpecAnResBw** – defines the resolution bandwidth of the spectrum analyzer when the transmitter measurement was performed. Note that a uniform frequency spacing and resolution bandwidth must be used across all data points in the file.

@ **Begin Data** – the actual data begins after this line.

> **Note:**
>
> The header lines are *case sensitive*. Comments can be included in the header section of the file by preceding the comment with a "# ". Any lines beginning with a "# " are ignored by EMIT.

An example of a raw channel data file is shown below:

```
# Comment
@ DlxDataType = "TxSpecAnData"
@ ChannelFreq = 144e6
@ SpecAnResBw = 30e3
@ BeginData
# Freq (Hz)     Power (dBm)
 1000000.000000 -96.70
 1030006.743088 -87.83
 1060013.486177 -89.63
 1090020.229265 -90.08
 1120026.972353 -88.72
 ...
 ...
```

Spectral Component File

A spectral component file contains information to model transmit channel emissions or receive channel susceptibility. The file is distinct from Raw Channel Data in that spectral characteristics (such as transmitter harmonics, receiver mixer products, spurious responses, etc.) are identified and specified as distinct entities. This additional structure allows EMIT to properly address broadband noise, compute intermodulation, and extrapolate the measured data to other un-measured channels. The spectral component file format uses XML syntax.

The first line specifies the document type as:

```
<!DOCTYPE DlxXML>
```

MeasurementGroup and Measurement Elements

The remainder of the document is contained within the root element, **MeasurementGroup**. The root has a single child element, **Measurement**, which must specify an attribute Version="1.0". Note that only a single Measurement is supported in Spectral Component Files loaded by EMIT. The Measurement element must have three child elements: TxOrRx, InputParameters, and Result.

```
<MeasurementGroup>
 <Measurement Version="1.0">
```

```
  ...body of document...
 </Measurement>
</MeasurementGroup>
```

Measurement Child Elements (TxOrRx, InputParameters, Result)

The first child element of Measurement has the tag **TxOrRx** and specifies whether the Spectral Component Data is for a Transmitter or a Receiver. For a Transmitter, the TxOrRx element will be:

```
<TxOrRx>Tx</TxOrRx>
```

For a Receiver, the TxOrRx element will be:

```
<TxOrRx>Rx</TxOrRx>
```

The second child element of Measurement has the tag **InputParameters**. The InputParameters element has one child, which specifies the frequency of the channel represented by the file. The frequency is specified in Hz. For a 100 MHz channel the InputParameters element is:

```
<InputParameters>
 <CenterFrequency>100.0e6</CenterFrequency>
</InputParameters>
```

The third and final child element of Measurement is the **Result** element. The child elements of Result differ for transmit and receive channels.

Transmit Channel Result Element

A transmit channel Result has:

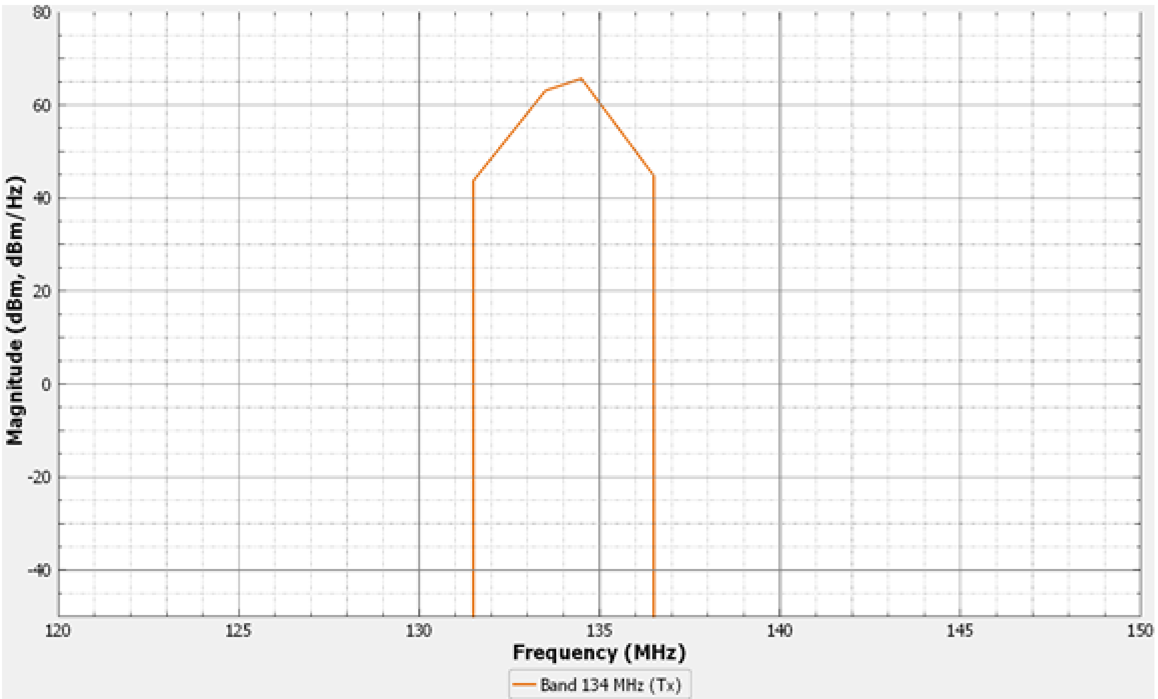
- Zero or one Fundamental element
- Zero or more Harmonic elements
- Zero or more Spur elements
- Zero or one BroadbandNoise element

The **Fundamental** element specifies the spectral content at the channel CenterFrequency. In version 1.0 of the format, the Fundamental element has a single attribute specifying the fundamental amplitude in dBm. A 10 watt transmitter would have the following Fundamental element:

```
<Fundamental Amplitude="40.0"/>
```

In version 2.0 of the format, the Fundamental element is expanded to have zero or more child **Point** elements. The Point elements are used to define the shape of the fundamental and contain two attributes: RelFreq and Amp. The RelFreq attribute specifies the frequency as a delta from the channel frequency and the Amp specifies the power at this relative frequency in dBm. The snippet below shows the fundamental definition for a 134 MHz channel using version 2.0 and its corresponding plot.

```
<Fundamental>
 <Point RelFreq="-2500000.0" Amp="43.6713"/>
 <Point RelFreq="-500000.0" Amp="63.0888"/>
 <Point RelFreq="500000.0" Amp="65.6478"/>
 <Point RelFreq="2500000.0" Amp="44.8015"/>
</Fundamental>
```



A **Harmonic** element specifies nonlinear artifacts in the transmit spectrum that occur at integer multiples of the channel frequency.

In version 1.0, this element has three attributes: Order, AboveNoise, and Amplitude. The first attribute specifies the harmonic order. This is an integer starting with 2 (the 2nd harmonic at 2x the channel CenterFrequency). The second attribute specifies whether the harmonic was detected at a level above the measurement noise. If no peak is detected above the measurement noise when searching for a particular harmonic, this value can be set to false. The harmonic amplitude would then be specified at the measurement noise level as a conservative estimate. In the EMIT GUI, there is an option for disabling harmonics below the noise floor which takes advantage of this information. The final attribute is the amplitude, specified in dBm. A second harmonic that was at -60 dBm and above the measured noise level would be specified as:

```
<Harmonic Order="2" AboveNoise="true" Amplitude="-60.0"/>
```

In version 2.0 of the format, the Harmonic element is expanded to have zero or more child Point elements. The Point elements are used to define the shape of the harmonic and contain two attributes: RelFreq and Amp. The RelFreq attribute specifies the frequency as a delta from the harmonic center frequency (Order*CenterFrequency) and the Amp specifies the power at this relative frequency in dBm. The Amplitude attribute of the Harmonic element is ignored if any Point elements are defined for the harmonic.

```
<Harmonic AboveNoise="true" Order="2">
 <Point RelFreq="-200.0e3" Amp="-70.0"/>
 <Point RelFreq="-100.0e3" Amp="-70.0"/>
 <Point RelFreq="100.0e3" Amp="-70.0"/>
 <Point RelFreq="200.0e3" Amp="-70.0"/>
</Harmonic>
```

A **Spur** element specifies narrowband spectral components at non-harmonic frequencies.

In version 1.0, Spur is defined by width, frequency, and amplitude attributes. All Spur elements defined by version 1.0 of the format are assumed to have a rectangular shape. The width is the bandwidth of the spur in Hz. The frequency is the center frequency of the spur and is also specified in Hz. The amplitude is specified in dBm. A spur that is 25 kHz wide, centered at 80 MHz, and has an amplitude of -70 dBm would be specified as:

```
<Spur Width="25e3" Frequency="80e6" Amplitude="-70.0"/>
```

In version 2.0 of the format, the Spur element is expanded to have zero or more child Point elements. The Point elements are used to define the shape of the spur and contain two attributes: Freq and Amp. The Freq attribute specifies the absolute frequency of a point in the spur and the Amp specifies the power at this frequency in dBm.

```
<Spur>
 <Point Freq="563371000" Amp="-40.0"/>
 <Point Freq="563372000" Amp="-34.0"/>
 <Point Freq="563374000" Amp="-39.0"/>
</Spur>
```

A **BroadbandNoise** element specifies the broadband transmitter noise characteristics. The noise floor is modeled as a collection of frequency-amplitude points. Keep in mind that EMIT will linearly interpolate between points and extrapolate the first point amplitude down to 1 Hz and the final point amplitude up to 100 GHz. Include sufficient data to account for this behavior. Each point is a child element of BroadbandNoise and has amplitude and frequency attributes. The amplitude is in dBm/Hz and the frequency is in Hz. A broadband trace that is -100 dBm/Hz at 100 MHz and falls with a slope of 1 dB per MHz to a floor of -130 dBm/Hz is specified as:

```
<BroadbandNoise>
 <Point Amp="-130.0" Freq="70e6"/>
 <Point Amp="-100.0" Freq="100e6"/>
 <Point Amp="-130.0" Freq="130e6"/>
</BroadbandNoise>
```

> **Note:**
>
> There are no changes in the Broadband specification between versions 1.0 and 2.0 of the format.

Receive Channel Result Element

A receive channel Result element has:

- Zero or one RxSelectivity element
- Zero or one IntermediateFreq element
- Zero or more MixerProduct elements
- Zero or more Spur elements
- Zero or one RxSaturation element

Features that are left unspecified will be modeled using measured data for other channels in the band or the band's parametric model.

An **RxSelectivity** element specifies the receiver susceptibility for the receive channel and adjacent channels. The selectivity is modeled as a collection of frequency-amplitude points. EMIT will linearly interpolate between points. Each point is a child element of RxSelectivity and contains frequency and amplitude attributes. The amplitude is the susceptibility and is specified in dBm. The frequency is specified in Hz. The selectivity for a 100 MHz receive channel with approximately 100 kHz bandwidth, -110 dBm susceptibility in-channel, and a 40 dB change in susceptibility in the adjacent channels is specified as:

```
<RxSelectivity>
 <Point Amp="-70.0" Freq="99.8e6"/>
 <Point Amp="-110.0" Freq="99.9e6"/>
 <Point Amp="-110.0" Freq="100.1e6"/>
 <Point Amp="-70.0" Freq="100.2e6"/>
</RxSelectivity>
```

An **IntermediateFreq** element specifies the susceptibility at the receiver IF frequency and is modeled as a receive spur at that frequency. This element requires only an amplitude attribute. The frequency is determined from parameters specified in the Band. The Mixer Product model must be enabled for the intermediate frequency specification to take effect. A receive channel with -60 dBm susceptibility at the intermediate frequency is specified as:

```
<IntermediateFreq Amplitude="-60.0"/>
```

In version 2.0 of the format, the IntermediateFreq element is expanded to have zero or more child Point elements. The Point elements are used to define the shape of the fundamental and contain two attributes: RelFreq and Amp. The RelFreq attribute specifies the frequency as a delta from the channel's intermediate frequency and the Amp specifies the power at each frequency in dBm. The snippet below shows the IntermediateFreq definition.

```
<IntermediateFreq>
 <Point Amp="-84.0" RelFreq="-0.2e6"/>
 <Point Amp="-77.0" RelFreq="-0.1e6"/>
 <Point Amp="-76.0" RelFreq="0.1e6"/>
 <Point Amp="-81.0" RelFreq="0.2e6"/>
</IntermediateFreq>
```

MixerProduct elements specify the susceptibility of receiver mixer products. These elements are specified by the LOOrder (integer), RFOrder (integer) and Amplitude (in dBm). The sum and difference products are differentiated by the sign of the RFOrder. In order for these spurs to be included in the receive channel model, the Mixer Product model must be enabled in the Band. Note that one of the mixer products is the tuned frequency. It is acceptable to define the susceptibility of the in-channel mixer product. However, if the RxSelectivity is also defined, it will override the mixer product definition of the in-channel region. A mixer product due to the third harmonic of some interference mixing with the second harmonic of the local oscillator whose difference frequency falls in the receiver intermediate bandwidth is specified as:

```
<MixerProduct LOOrder="2" RFOrder="-3" Amplitude="-60.0"/>
```

In version 2.0 of the format, the MixerProduct element is expanded to have zero or more child Point elements. The LOOrder and RFOrder for the parent MixerProduct element are still required. The Point elements are used to define the shape of the fundamental and contain two attributes: RelFreq and Amp. The RelFreq attribute specifies the frequency as a delta from the channel's intermediate frequency and the Amp specifies the power at each frequency in dBm. The snippet below shows the IntermediateFreq definition.

```
<MixerProduct LOOrder="2" RFOrder="-3">
 <Point Amp="-99.0" RelFreq="-0.5e6"/>
 <Point Amp="-96.0" RelFreq="-0.2e6"/>
 <Point Amp="-95.0" RelFreq="0.2e6"/>
 <Point Amp="-101.0" RelFreq="0.5e6"/>
</MixerProduct>
```

A **Spur** element specifies narrowband spectral components at non-mixer product frequencies. In version 1.0 a Spur is defined by Width, Frequency, and Amplitude attributes. All spurs defined by version 1.0 of the format are assumed to have a rectangular shape. The width is the bandwidth of the spur in Hz. The frequency is the center frequency of the spur and is also specified in Hz. The amplitude is specified in dBm. A spur that is 25 kHz wide, centered at 80 MHz, and has an amplitude of -70 dBm would be specified as:

```
<Spur Width="25e3" Frequency="80e6" Amplitude="-70.0"/>
```

In version 2.0 of the format, the Spur element is expanded to have zero or more child Point elements. The Point elements are used to define the shape of the spur and contain two attributes: Freq and Amp. The Freq attribute specifies the absolute frequency of a point in the spur and the Amp specifies the power at this frequency in dBm.

```
<Spur>
 <Point Freq="563371000" Amp="-40.0"/>
 <Point Freq="563372000" Amp="-34.0"/>
 <Point Freq="563374000" Amp="-39.0"/>
</Spur>
```

An **RxSaturation** element specifies the receiver susceptibility outside of the otherwise-defined narrowband regions (i.e., Mixer Products, RxSelectivity, IF Frequency). The saturation is modeled as a collection of frequency-amplitude points. EMIT will linearly interpolate between points and extrapolate the amplitude of the lowest frequency point down to 1 Hz and the amplitude of the highest frequency point up to 100 GHz. Each point is a child element of RxSaturation and has an amplitude and frequency attribute. The amplitude is the susceptibility and is specified in dBm. The frequency is specified in Hz. The saturation for a receiver which is -20 dBm over a 140-150 MHz tuning range and rises to +10 dBm over a 50 MHz span outside this region is defined as:

```
<RxSaturation>
 <Point Amp="10.0" Freq="90.0e6"/>
 <Point Amp="-20.0" Freq="140.0e6"/>
 <Point Amp="-20.0" Freq="150.0e6"/>
 <Point Amp="10.0" Freq="200.0e6"/>
</RxSaturation>
```

> **Note:**
>
> There are no changes in the Saturation specification between versions 1.0 and 2.0 of the format.

Example Files

An example Spectral Component File version 1.0 for a 144 MHz transmitter is shown below.

```
<!DOCTYPE DlxXML>
<MeasurementGroup>
  <Measurement Version="1.0">
    <TxOrRx>Tx</TxOrRx>
    <InputParameters>
      <CenterFrequency>144000000</CenterFrequency>
    </InputParameters>
    <Result>
      <BroadbandNoise>
        <Point Amp="-133.7897" Freq="1000000.0"/>
        <Point Amp="-134.1818" Freq="9760000.0"/>
        <Point Amp="-120.0549" Freq="17200000.0"/>
        <Point Amp="-120.1689" Freq="34530000.0"/>
        <Point Amp="-126.2221" Freq="42260000.0"/>
        <Point Amp="-129.4344" Freq="50140000.0"/>
        <Point Amp="-125.5133" Freq="60850000.0"/>
        <Point Amp="-120.9997" Freq="66320000.0"/>
        <Point Amp="-116.6689" Freq="80010000.0"/>
        <Point Amp="-117.2021" Freq="89070000.0"/>
        <Point Amp="-119.3128" Freq="95480000.0"/>
        <Point Amp="-117.0206" Freq="107100000.0"/>
        <Point Amp="-118.3513" Freq="118620000.0"/>
        <Point Amp="-115.5927" Freq="129710000.0"/>
        <Point Amp="-107.7436" Freq="134830000.0"/>
        <Point Amp="-105.7393" Freq="138730000.0"/>
        <Point Amp="-95.6713" Freq="142720000.0"/>
        <Point Amp="-95.0888" Freq="144910000.0"/>
        <Point Amp="-105.6478" Freq="155790000.0"/>
        <Point Amp="-104.8015" Freq="163270000.0"/>
        <Point Amp="-107.9786" Freq="167490000.0"/>
        <Point Amp="-116.1507" Freq="174540000.0"/>
        <Point Amp="-117.3791" Freq="181120000.0"/>
        <Point Amp="-128.8414" Freq="194980000.0"/>
        <Point Amp="-133.9816" Freq="204790000.0"/>
        <Point Amp="-136.8578" Freq="217250000.0"/>
        <Point Amp="-136.0068" Freq="744910000.0"/>
        <Point Amp="-136.4791" Freq="1499990000.0"/>
      </BroadbandNoise>
      <Fundamental Amplitude="45.35"/>
      <Harmonic Order="2" AboveNoise="true" Amplitude="-59.61"/>
      <Harmonic Order="3" AboveNoise="true" Amplitude="-64.69"/>
      <Harmonic Order="4" AboveNoise="true" Amplitude="-51.59"/>
      <Harmonic Order="5" AboveNoise="true" Amplitude="-73.83"/>
      <Harmonic Order="6" AboveNoise="true" Amplitude="-77.58"/>
      <Harmonic Order="7" AboveNoise="false" Amplitude="-95.7675"/>
      <Harmonic Order="8" AboveNoise="true" Amplitude="-82.89"/>
      <Harmonic Order="9" AboveNoise="true" Amplitude="-85.28"/>
      <Harmonic Order="10" AboveNoise="true" Amplitude="-82.11"/>
      <Spur Width="10000" Frequency="45900000" Amplitude="-77.12"/>
      <Spur Width="20000" Frequency="98105000" Amplitude="-66.04"/>
      <Spur Width="20000" Frequency="143315000" Amplitude="-43.84"/>
      <Spur Width="10000" Frequency="144120000" Amplitude="-45.58"/>
      <Spur Width="20000" Frequency="144685000" Amplitude="-42.31"/>
      <Spur Width="20000" Frequency="189895000" Amplitude="-73"/>
    </Result>
  </Measurement>
</MeasurementGroup>
```


An example Spectral Component File version 2.0 for a 134 MHz transmitter is shown below.

```xml
<MeasurementGroup>
  <Measurement Version="2.0">
    <TxOrRx>Tx</TxOrRx>
    <InputParameters>
      <CenterFrequency>134000000</CenterFrequency>
    </InputParameters>
    <Result>
      <BroadbandNoise>
        <Point Freq="1000000.000000000000000" Amp="-133.7897"/>
        <Point Freq="17200000.000000000000000" Amp="-120.0549"/>
        <Point Freq="20680000.000000000000000" Amp="-118.0598"/>
        <Point Freq="42260000.000000000000000" Amp="-126.2221"/>
        <Point Freq="50140000.000000000000000" Amp="-129.4344"/>
        <Point Freq="60850000.000000000000000" Amp="-125.5133"/>
        <Point Freq="66320000.000000000000000" Amp="-120.9997"/>
        <Point Freq="80010000.000000000000000" Amp="-116.6689"/>
        <Point Freq="89070000.000000000000000" Amp="-117.2021"/>
        <Point Freq="127470000.000000000000000" Amp="-116.9772"/>
        <Point Freq="129710000.000000000000000" Amp="-115.5927"/>
        <Point Freq="134830000.000000000000000" Amp="-107.7436"/>
        <Point Freq="138730000.000000000000000" Amp="-105.7393"/>
        <Point Freq="155790000.000000000000000" Amp="-105.6478"/>
        <Point Freq="163270000.000000000000000" Amp="-104.8015"/>
        <Point Freq="167490000.000000000000000" Amp="-107.9786"/>
        <Point Freq="174540000.000000000000000" Amp="-116.1507"/>
        <Point Freq="181120000.000000000000000" Amp="-117.3791"/>
        <Point Freq="356080000.000000000000000" Amp="-136.4226"/>
        <Point Freq="372800000.000000000000000" Amp="-137.7641"/>
        <Point Freq="439300000.000000000000000" Amp="-137.8487"/>
        <Point Freq="865740000.000000000000000" Amp="-136.1594"/>
        <Point Freq="930120000.000000000000000" Amp="-137.3747"/>
        <Point Freq="1286750000.000000000000000" Amp="-137.8334"/>
        <Point Freq="1499990000.000000000000000" Amp="-136.4791"/>
      </BroadbandNoise>
      <Fundamental>
        <Point RelFreq="-2500000.0" Amp="43.6713"/>
        <Point RelFreq="-500000.0" Amp="63.0888"/>
        <Point RelFreq="500000.0" Amp="65.6478"/>
        <Point RelFreq="2500000.0" Amp="44.8015"/>
      </Fundamental>
      <Harmonic AboveNoise="true" Order="2">
        <Point RelFreq="-200000.0" Amp="-70.0"/>
        <Point RelFreq="-100000.0" Amp="-40.0"/>
        <Point RelFreq="100000.0" Amp="-40.0"/>
        <Point RelFreq="200000.0" Amp="-70.0"/>
      </Harmonic>
      <Harmonic AboveNoise="true" Amplitude="-64.69" Order="3"/>
      <Harmonic AboveNoise="true" Amplitude="-51.59" Order="4"/>
      <Harmonic AboveNoise="true" Amplitude="-73.83" Order="5"/>
      <Spur>
        <Point Freq="92720000.000000000000000" Amp="55.6713"/>
        <Point Freq="94910000.000000000000000" Amp="55.0888"/>
        <Point Freq="105790000.000000000000000" Amp="57.6478"/>
        <Point Freq="103270000.000000000000000" Amp="57.8015"/>
        <Point Freq="106790000.000000000000000" Amp="59.6478"/>
        <Point Freq="109270000.000000000000000" Amp="53.8015"/>
      </Spur>
      <Spur Frequency="48900000" Amplitude="-77.12" Width="10000"/>
    </Result>
  </Measurement>
</MeasurementGroup>
```

An example Spectral Component File version 2.0 for a 138 MHz receiver is shown below.

```xml
<!DOCTYPE D1xXML>
<MeasurementGroup>
  <Measurement Version="2.0">
    <TxOrRx>Rx</TxOrRx>
    <InputParameters>
      <CenterFrequency>138000000</CenterFrequency>
    </InputParameters>
    <Result>
      <RxSelectivity>
        <Point Amp="0" RelFreq="-80.0e6"/>
        <Point Amp="-35" RelFreq="-30.0e6"/>
        <Point Amp="-68" RelFreq="-10.0e6"/>
        <Point Amp="-98" RelFreq="-1.0e6"/>
        <Point Amp="-127" RelFreq="-0.5e6"/>
        <Point Amp="-127" RelFreq="0.5e6"/>
        <Point Amp="-98" RelFreq="1.0e6"/>
        <Point Amp="-68" RelFreq="10.0e6"/>
        <Point Amp="-35" RelFreq="30.0e6"/>
        <Point Amp="0" RelFreq="80.0e6"/>
      </RxSelectivity>
      <IntermediateFreq>
        <Point Amp="-84.0" RelFreq="-0.2e6"/>
        <Point Amp="-77.0" RelFreq="-0.1e6"/>
        <Point Amp="-76.0" RelFreq="0.1e6"/>
        <Point Amp="-81.0" RelFreq="0.2e6"/>
      </IntermediateFreq>
      <MixerProduct LOOrder="1" RFOrder="1">
        <Point Amp="-99" RelFreq="-.5e6"/>
        <Point Amp="-96" RelFreq="-.2e6"/>
        <Point Amp="-95" RelFreq=".2e6"/>
        <Point Amp="-101" RelFreq=".5e6"/>
      </MixerProduct>
      <MixerProduct LOOrder="1" Amplitude="-46.012649" RFOrder="-1"/>
      <MixerProduct LOOrder="1" Amplitude="-48.026500" RFOrder="-2"/>
      <MixerProduct LOOrder="2" RFOrder="2">
        <Point Amp="-89" RelFreq="-.7e6"/>
        <Point Amp="-86" RelFreq="-.3e6"/>
        <Point Amp="-85" RelFreq=".3e6"/>
        <Point Amp="-91" RelFreq=".7e6"/>
      </MixerProduct>
      <MixerProduct LOOrder="2" Amplitude="-37.337989" RFOrder="-2"/>
      <RxSaturation>
        <Point Amp="0.45" Freq="6e7"/>
        <Point Amp="0.1" Freq="7e7"/>
        <Point Amp="0.2" Freq="8e7"/>
        <Point Amp="0.17" Freq="9e7"/>
        <Point Amp="0" Freq="1e8"/>
        <Point Amp="0.1" Freq="2e8"/>
        <Point Amp="0.15" Freq="3e8"/>
        <Point Amp="0.05" Freq="4e8"/>
        <Point Amp="0.3" Freq="5e8"/>
        <Point Amp="0.45" Freq="6e8"/>
        <Point Amp="0.1" Freq="7e8"/>
        <Point Amp="0.2" Freq="8e8"/>
        <Point Amp="0.17" Freq="9e8"/>
      </RxSaturation>
      <Spur>
        <Point Amp="-50.0" Freq="543371000"/>
        <Point Amp="-40.0" Freq="563371000"/>
        <Point Amp="-38.0" Freq="583371000"/>
        <Point Amp="-52.0" Freq="603371000"/>
      </Spur>
      <Spur Frequency="80e6" Width="25e3" Amplitude="-70.0"/>
      <Spur Frequency="120e6" Width="31e3" Amplitude="-65.0"/>
    </Result>
  </Measurement>
</MeasurementGroup>
```

Spectral Profile File Format (*.dlxsp)

Spectral profiles are used to define the transmission characteristics of Components (filters, cables, etc.) in EMIT. This data is supplied to EMIT in text files using the *.dlxsp file format. Note that outboard components can also be specified in EMIT by using a Touchstone file.

Spectral profile files contain frequency amplitude pairs with the amplitude specified in dB. The file consists of a header and two columns of data that must be separated by a tab or a space (not a comma). The data is arranged in two columns (frequency and magnitude) after the header.

The header info should be in the following form:

```
@ DlxDataType = SpectralProfile
@ FrequencyUnit = megahertz
@ Begin Data
```

Each line in the header is as follows:

@ DlxDataType – defines the type of data in the file. For spectral profiles this should be 'SpectralProfile'.

@ FrequencyUnit – defines the frequency units used in the file. Valid units are hertz, kilohertz, megahertz or gigahertz.

@ Begin Data – the actual data begins after this line.

> **Note:**
>
> The header lines are *case sensitive* and the amplitude values are specified in dB. Comments can be included in the header section of the file by beginning the comment with a "# ". Any lines beginning with a "# " are ignored by EMIT.

An example of a spectral profile file is shown below:

```
# Example EMIT Spectral Profile File

# Blank lines or lines beginning with # are ignored

@ DlxDataType = SpectralProfile
@ FrequencyUnit = hertz
@ Begin Data

2.0             0
2.00E+07        -0.02
4.00E+07        -4.76E-02
6.00E+07        -0.04
8.00E+07        -0.01
1.00E+08        -0.01
1.20E+08        -0.05
1.40E+08        0
1.60E+08        -0.45
1.80E+08        -2.98
2.00E+08        -6.23
```

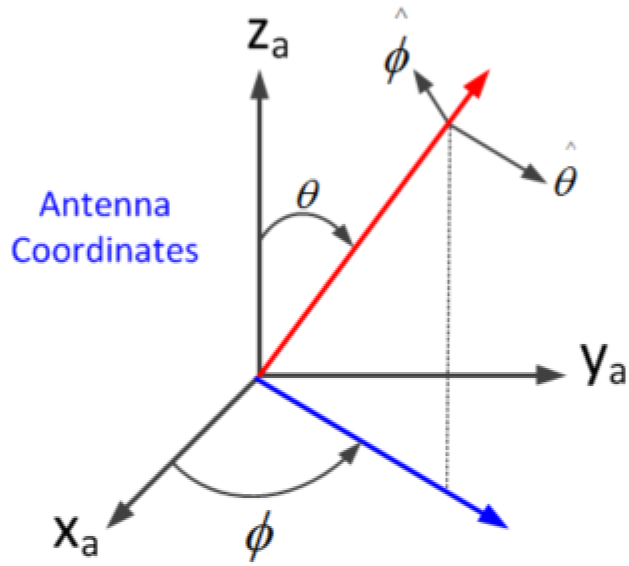
Antenna Pattern File Format (*.patdat)

EMIT supports the Ansys antenna pattern file format for importing 3D antenna radiation patterns that can be used by EMIT for computing antenna-to-antenna coupling. This file format has been developed by Ansys to be a general format for 3D antenna patterns and as such there are some aspects of the file that are not relevant to EMIT. In particular, the renorm keyword is not needed by EMIT and should either be set to false or be omitted from the *.patdat file. If EMIT encounters a *.patdat file with the renorm keyword set to true, it will be ignored.

The Ansys native antenna pattern file format defines the far-field radiation pattern of an antenna over the entire 4π steradians of a sphere. Pattern samples are entered over a two-dimensional angular sampling domain versus spherical coordinates, θ and ϕ. Frequency dependent antennas may also be characterized by introducing a third sampling dimension for frequency.

Pattern samples in the *.patdat file are interpreted by EMIT in gain scale. By convention, gain is a scalar quantity. In order to support phase information, the gain values in the *.patdat file are entered as field-scale complex (real and imaginary parts), such that their magnitude square is power-scale gain. If complex g = field-scale gain (as in the *.patdat file), then conventional power-scale gain G = |g|^2, and gain on a dBi scale is given by 10log10G.

Since antennas have polarization, gain levels must be specified for the two orthogonal polarizations, $\hat{\theta}$ and $\hat{\phi}$. The diagram below defines θ, ϕ, and the two associated polarization vectors that are implicit in the table.



EMIT interprets the values in the *.patdat file as true antenna gain and uses these values in the EMI margin calculations. For this reason, the renorm flag is ignored if present in the *.patdat file and no normalization of the values to directivity is performed. The Ansys antenna pattern format is documented in the header comments in the example below.

```
# The header of the .patdat format antenna pattern file is based on keys. Key lines can appear in any order.
# Blank lines and lines beginning with '#' are ignored.
#
# "renorm" key line is optional. Default is no renormalization. When switched on, it forces tables levels to be
#renormalized to directivity levels. When switched off, table values are taken as already scaled in units
# of linear complex directivity or gain, such that their MAGNITUDE SQUARED is conventional Directivity or Gain. Their
# interpretation as Directivity, Gain, or some other quantitiy is up to the user.
#
# renorm true
#
# Key lines specifying a sweep, such as a theta sweep, include a parameter for the number of steps in the sweep (e.g.,
# <num_theta_step>). The number of sample points is the number of steps PLUS ONE.
#
# "freq" key line only required for multi-frequency antenna descriptions. If not included, then the pattern table is
#assumed to be mono-frequency and provide valid data for all frequencies.
#
# freq <freq_start_ghz> <freq_stop_ghz> <num_freq_step>
# If the following line is decommented, it would imply that the table samples frequency from 1 to 4 GHz in 30 steps
# (i.e., 31 samples per angle).
# freq 1.0 4.0 30
#
# "theta" key line is required. theta is measured in degrees from the z-axis of the antenna's local coordinate system.
# theta <theta_start_deg> <theta_stop_deg> <num_theta_step>
# <theta_start_deg> and <theta_stop_deg> MUST be 0, 180, respectively.

theta 0. 180. 90

# "phi" key line is required. phi is measured in degrees from the x-axis of the antenna's local coordinate system, such
# that the y-axis at +90 deg.
# phi <phi_start_deg> <phi_stop_deg> <num_phi_step>
# <phi_start_deg> and <phi_stop_deg> MUST be 0, 360, respectively.

phi 0. 360. 180

# "table_start" key line is required. It marks the end of the header and the start of pattern table data.

table_start

# In the table, each line provides the complex directivity in the theta- and phi-polarization for one aspect angle and one
# frequency.
# Frequency loops within theta angle, which loops within phi angle.
# Rl{theta-pol} Im{theta-pol} Rl{phi-pol} Im{phi-pol}
 8.309304e-02 -2.974831e+00 0.000000e+00 0.000000e+00
-4.107189e-02 -2.164177e+00 0.000000e+00 0.000000e+00
 1.697901e-01 2.048769e+00 0.000000e+00 0.000000e+00
#
# Contains 91*181 = 16471 lines of data
# Rest of table clipped in this template file
```

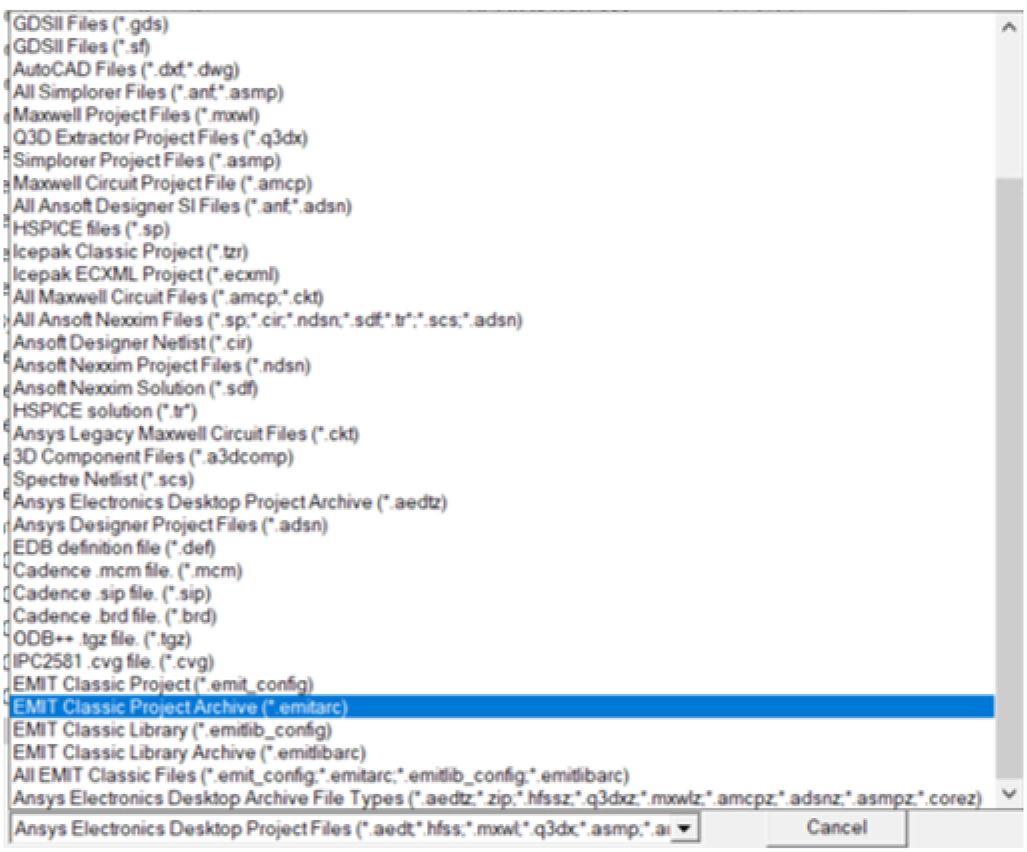
Importing EMIT Classic Projects and Libraries

> **Important:**
>
> The Import from Classic EMIT feature will be deprecated at release 2026 R1. To ensure a smooth transition, please convert all projects to the AEDT format prior to that time.

EMIT Classic projects and libraries can be imported into Ansys Electronics Desktop (AEDT) providing users with new schematic capabilities and tighter integration with other Ansys simulation tools such as HFSS.

To import an EMIT Classic project, do the following:

1. Click **File > Open**, and then change the **Files of type** field to **EMIT Classic Project (*.emit_config)** or **EMIT Classic Project Archive (*.emitarc)**.
2. Navigate to the desired project file, select it, and click **Open**. The project is then automatically converted to the AEDT format.



Similarly, EMIT Classic libraries can be imported by:

1. Click **File > Open**, and then change the **Files of type** field to **EMIT Classic Library (*.emitlib_config)** or **EMIT Classic Library Archive (*.emitlibarc)**.
2. Navigate to the desired library file, select it, and click **Open**. The library components are then automatically converted to the AEDT format and stored in a user's **PersonalLib** directory.

All project components and settings are transferred during the import with a few minor exceptions that are not currently supported:

- Multiplexers with >5 ports are not currently supported in AEDT.
- S Parameter Components with >3 ports are not currently supported in AEDT.
- RF Systems with multiple configurations - only a single configuration can be imported into each design. Users are prompted during the import to specify which configuration they want for each RF System and can create multiple designs within a single AEDT project.
- Passive Noise and Thermal Noise are automatically set to their default values and cannot be specified in AEDT.
- RF Systems stored in an EMIT Classic library are saved as individual designs in the **PersonalLib** directory and cannot be viewed from the library tree. To use these designs, users must select **File > Open** and select the design after navigating to the PersonalLib directory. The designs can be copied and pasted into other designs and/or projects using Ctrl+C and Ctrl+V.

EMIT Getting Started Guides

EMIT documentation includes the following Getting Started Guides:

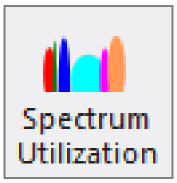
- *Getting Started with EMIT*
- *Tutorial 1*
- *Tutorial 2*
- *Tutorial 3*
- *Tutorial 4*
- *Tutorial 5*
- *Tutorial 6*
- *Tutorial 7*
- *Tutorial 8*

EMIT Toolkits

EMIT contains several built-in toolkits that provide additional workflows to ease simulation setup and analysis:

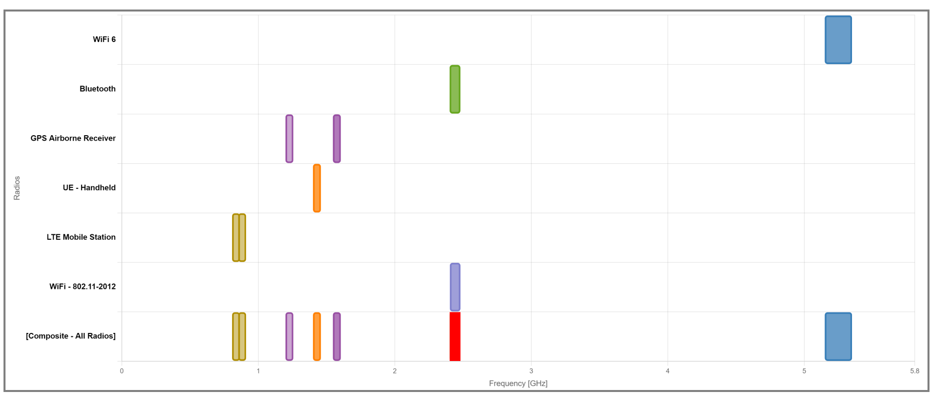
- Spectrum Utilization
- Emitter Group
- MIPI C-PHY Emitter Group

Spectrum Utilization



The Spectrum Utilization toolkit performs a pre-analysis of all radios and emitters in a project. It quickly computes and plots the transmit and receive bands of all systems in a project to help identify the most likely interference sources. The plots can also be used to quickly determine whether there is spectrum available for adding new systems to the environment.

The example below shows the potential for interference between the Bluetooth and WiFi-802.11-2012 radios.

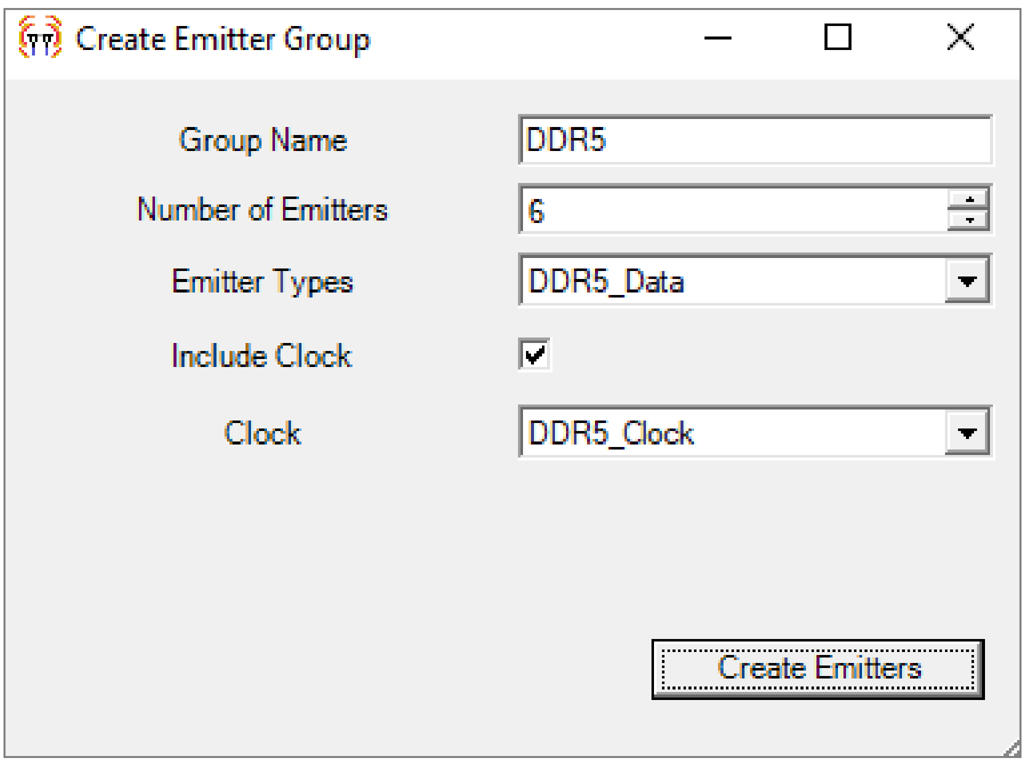


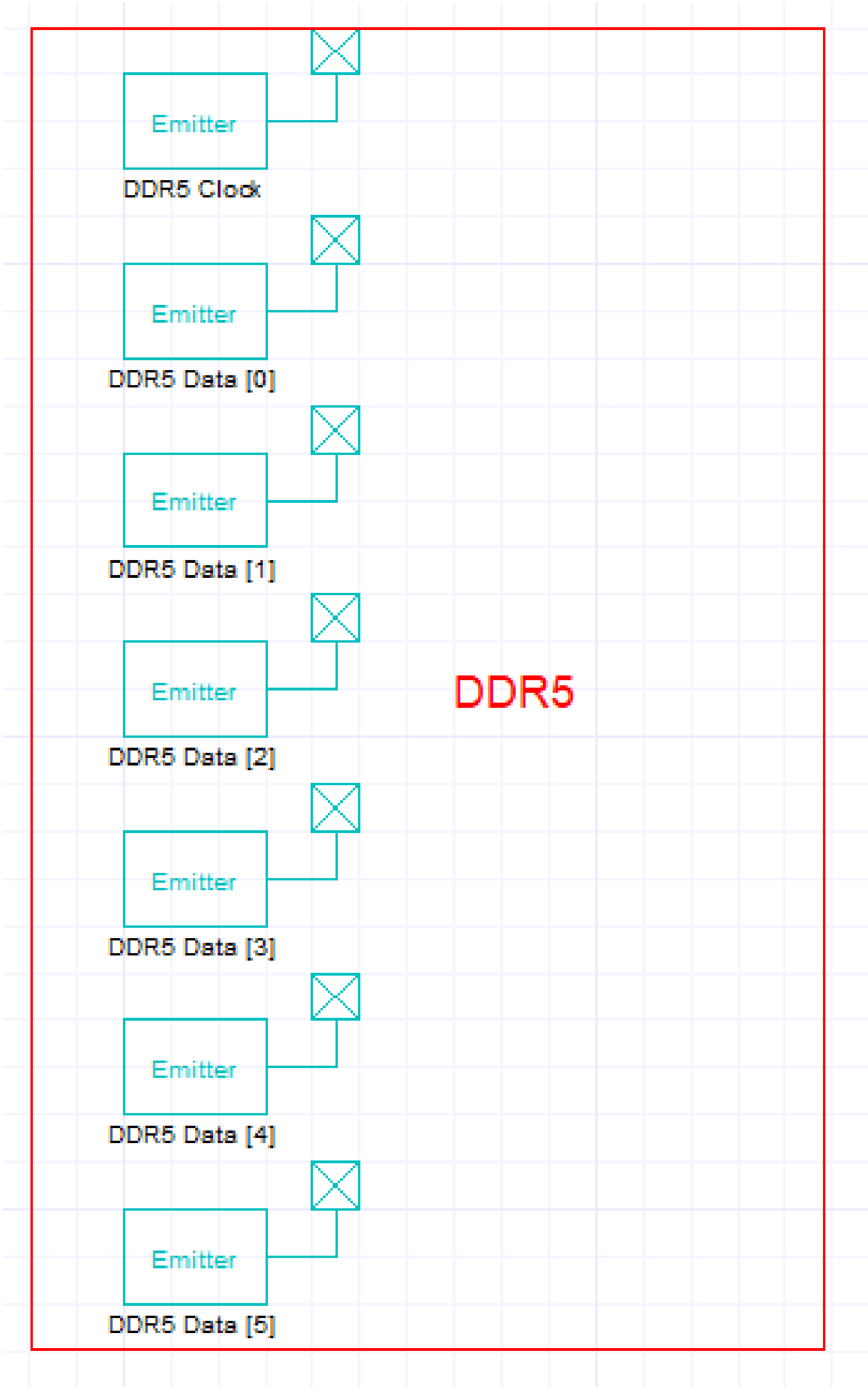
Note:

The Spectrum Utilization toolkit only analyzes the operational bands of each radio or emitter. It does not include any harmonics or spurs or the effects of broadband noise.

Emitter Group

The **Emitter Group** toolkit enables users to quickly add multiple emitters with identical characteristics to a project. The emitter type is selected from EMIT's built-in library models and a corresponding clock model for the standard can also be added. For example, the parameters below will model a DDR5 system with 6 data lines and 1 clock line. Once added to the schematic, the emitters can be connected to any coupling models and individually modified as usual.

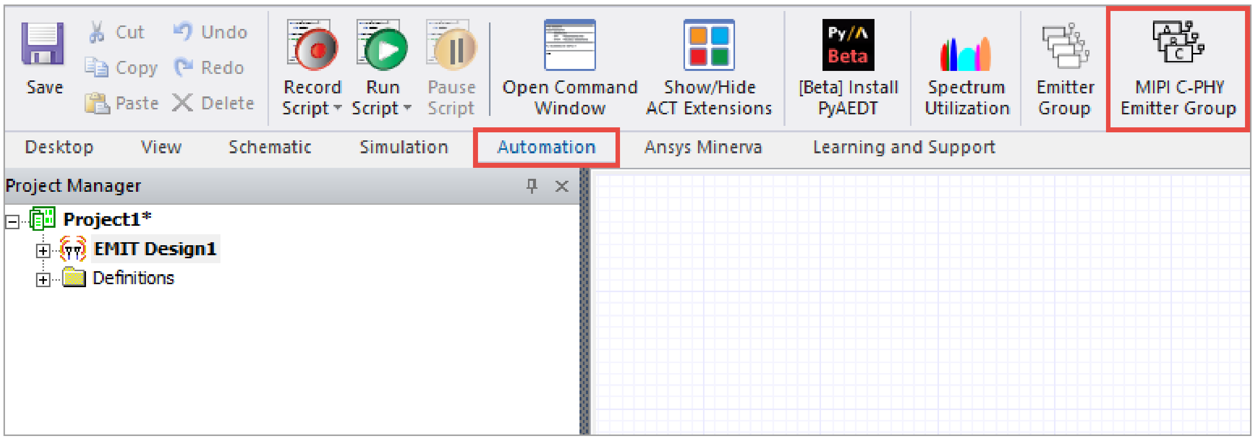



Emitter
DDR5 Clock
Emitter
DDR5 Data [0]
Emitter
DDR5 Data [1]
Emitter
DDR5 Data [2]
DDR5
Emitter
DDR5 Data [3]
Emitter
DDR5 Data [4]
Emitter
DDR5 Data [5]

MIPI C-PHY Emitter Group

The MIPI C-PHY Emitter Group toolkit automates the workflow for creating emitters based on the MIPI C-PHY specification.

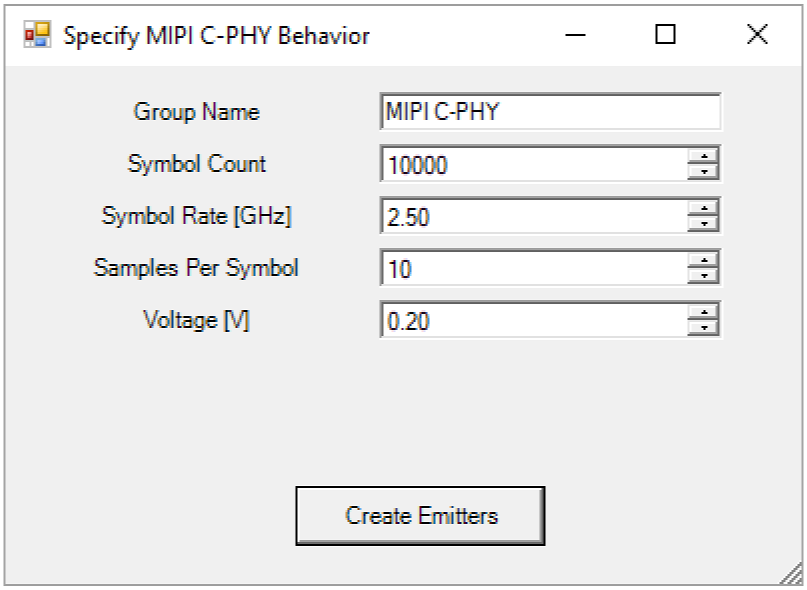
To use the toolkit, navigate to the **Automation** tab within an EMIT design and select the **MIPI C-PHY Emitter Group** toolkit.



This opens the **Specify MIPI C-PHY Behavior** window, with some key parameters. These parameters define the spectrum that is generated by the toolkit and used for each emitter:

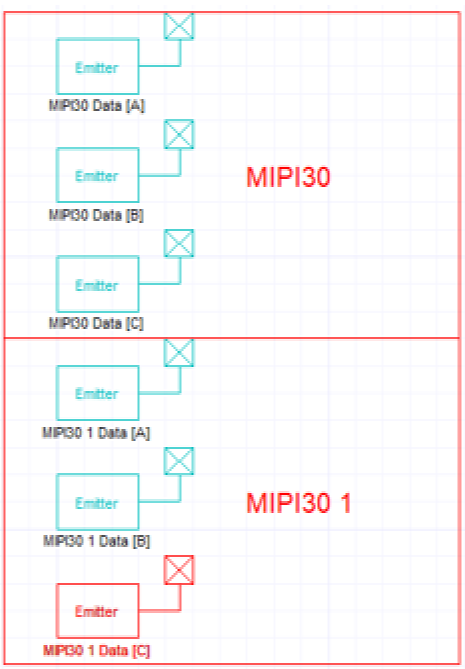
- Group Name
- Symbol Count
- Symbol Rate [Hz]
- Samples Per Symbol
- Voltage [V]

The window launches with default values to help guide the user.

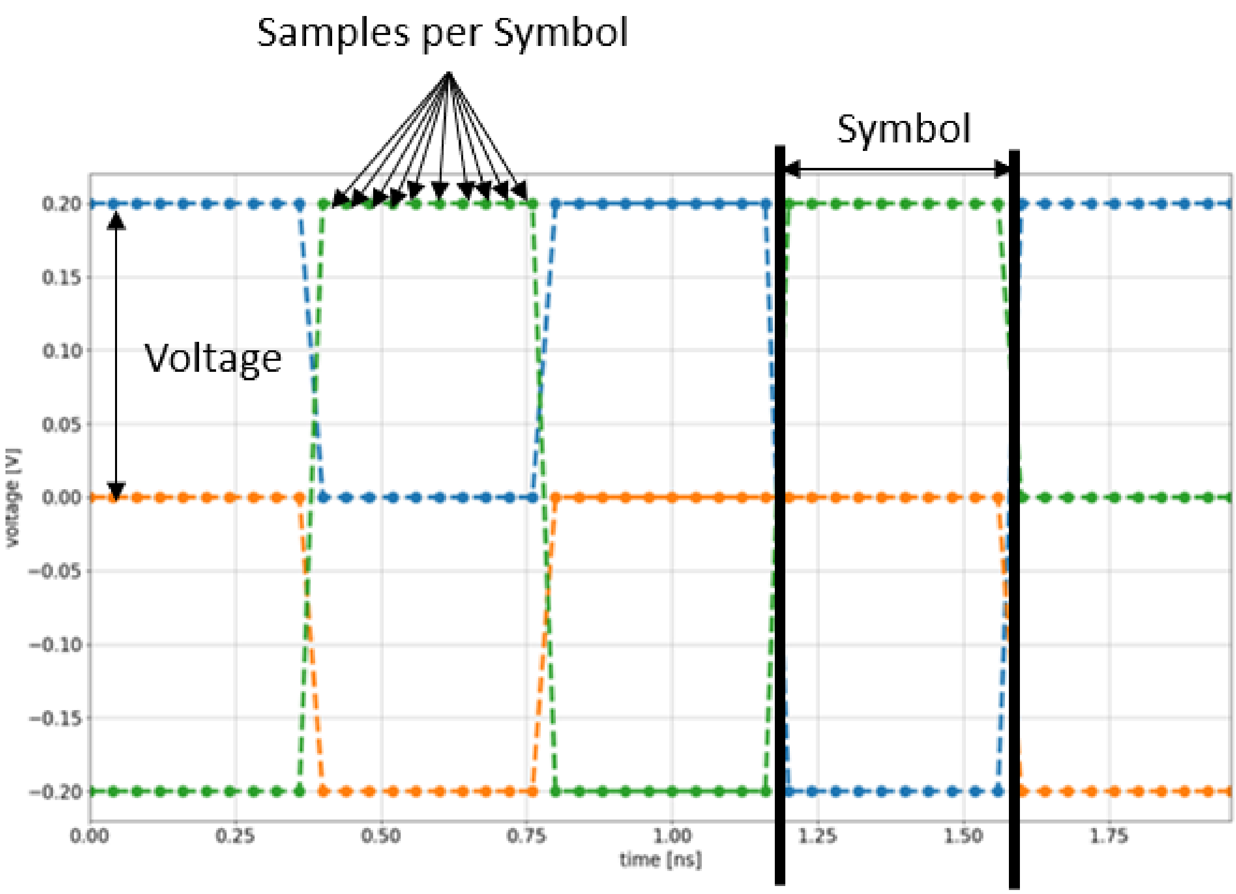


Note:

If the **Group Name** specified matches an existing emitter, the group name will be modified by adding a space and incremental number to the group name, as shown below.



After the MIPI C-PHY emitters are added to the EMIT design, the resulting waveform can be examined by double-clicking on any of the emitters (or by right-clicking and selecting **Configure**).

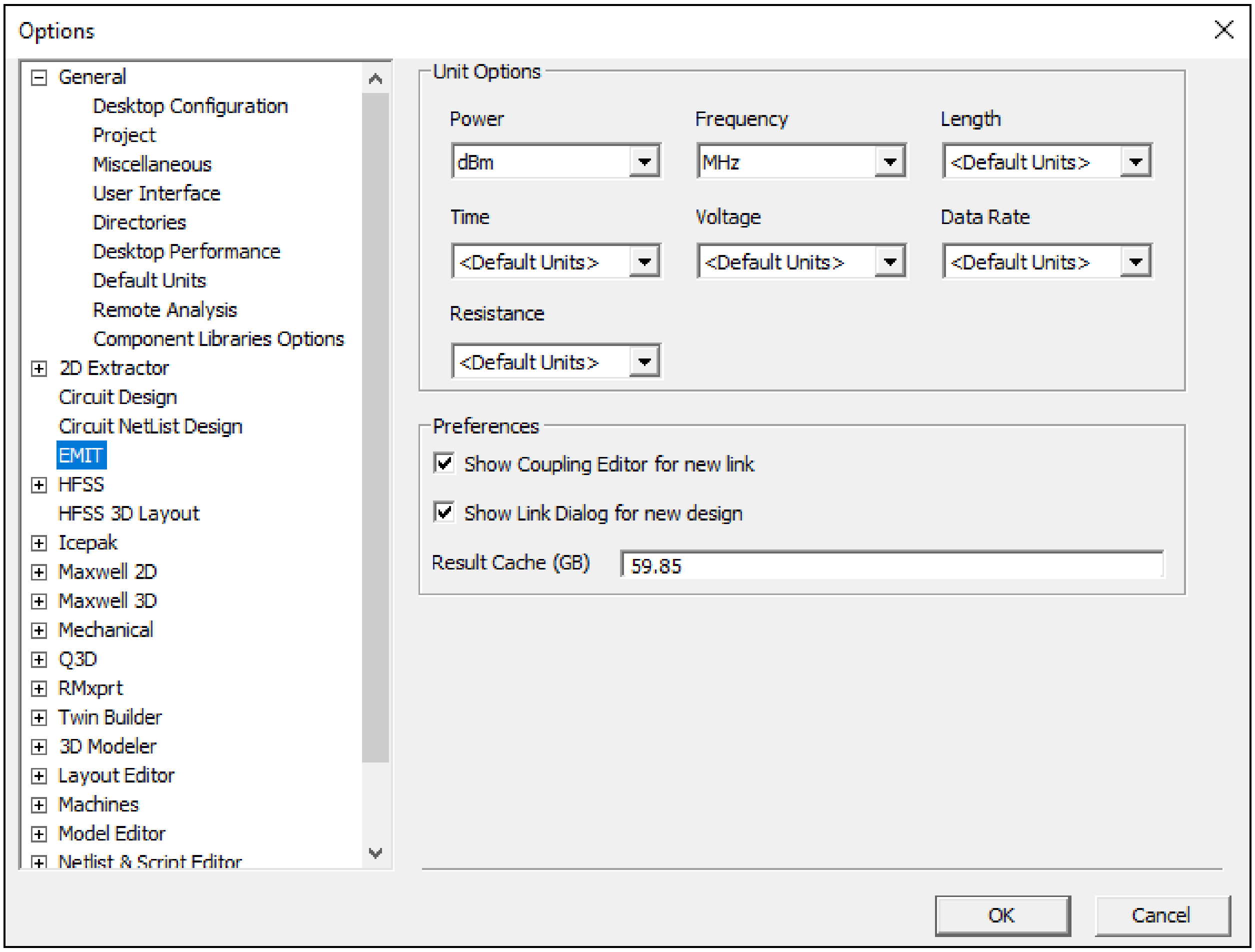


Each created emitter is linked to a text file specifying the time (seconds) and voltage levels (volts) of the signals. The number of data points, N = (Symbol Count) * (Samples per Symbol). During each symbol, the signals remain constant for a number of samples, specified by Samples per Symbol and the duration of each symbol is determined by the Symbol Rate (Hz). The duration in time, T = 1 / (Symbol Rate).

The data generated and stored in these text files implements the MIPI C-PHY signal, conformal to the description in section 4.1 of the MIPI C-PHY, version 2.1. The state transitions between the 6 MIPI C-PHY symbols is enforced by randomly switching to one of the other 5 symbols. The randomness of the generated signal means that running the script with the same variables may not result in the exact same output.

EMIT Options

EMIT uses the default units in **Tools** > **Options** > **General** > **Default Units** unless you configure global changes to EMIT options by making changes in **Tools** > **Options** > **EMIT**.



Unit Options set the units to be used for Power, Frequency, Length, Time, Voltage, Data Rate, and Resistance in EMIT designs. When set to *<Default Units>*, EMIT uses the units specified under **General** > **Default Units**. When specified, EMIT options override the general units for all EMIT designs.

By default, EMIT designs use *dBm* for Power because it is a more suitable unit for use in EMIT.

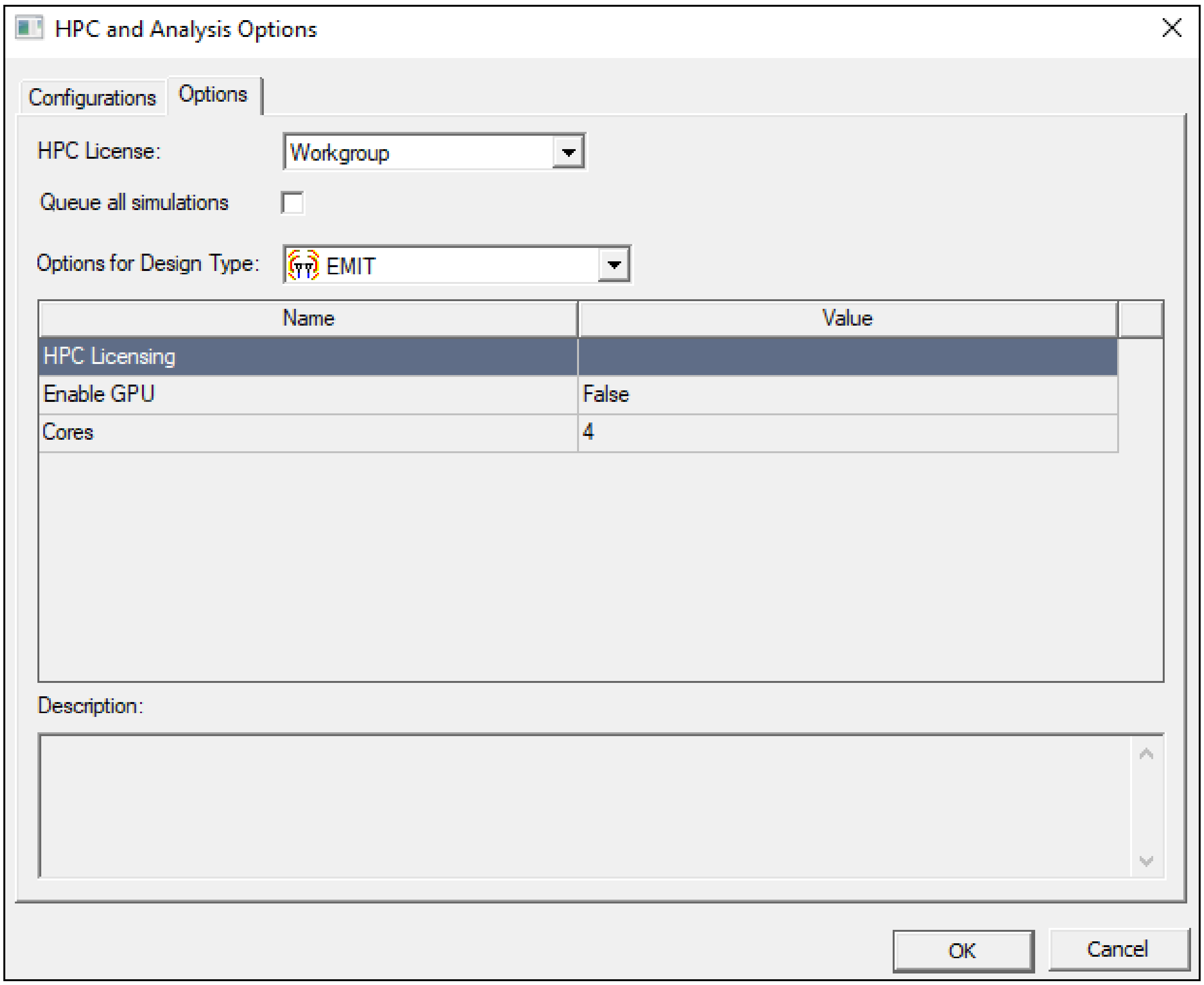
Preferences control settings within the EMIT design:

- **Show Coupling Editor for new link** – if selected, the Coupling Editor automatically opens each time a new coupling link is added to the EMIT design.

- **Show Link Dialog for new design** – if selected, when a new EMIT design is added to a project with an existing linkable (HFSS or HFSS 3D Layout) design, the Add Coupling Link dialog automatically opens. This enables the link(s) to be quickly added to the EMIT design.
- **Result Cache (GB)** – sets the limit on the amount of memory used to store channel level results. EMIT will store channel level results in RAM until this limit is reached and then the oldest results will be deleted.

EMIT HPC and Analysis Options

EMIT's analysis settings can be configured by going to **Tools** > **Options** > **HPC and Analysis Options** and clicking the **Options** tab.



Enable GPU: Allows EMIT to use available GPUs for EMIT solves.

Cores: Sets the number of CPU cores to be used for EMIT solves. Min:1, Max: 128

Performance and GPU Settings

Most EMIT solves will benefit from the use of a GPU; however, there are a few notable limitations:

- Tunable Filters prevent EMIT from creating batches of work that run well on the GPU. Any interactions between systems where either system includes a Tunable Filter will be solved using the CPU only.
- Bands with fewer than 16 channels are solved using the CPU only.
- The power flow through components in the RF chain and amplifier intermodulation are both computed on the CPU. As a result, the following situations may also result in the simulation relying more heavily on the CPU and will not see as substantial of a performance boost from the GPU:
    - Systems pairs with multiple amplifiers.
    - Systems with multiple antennas or multiple 3+ port components.
    - Radios, RF components, or coupling modeled with dense frequency sampling.

Finding Information in the Online Help

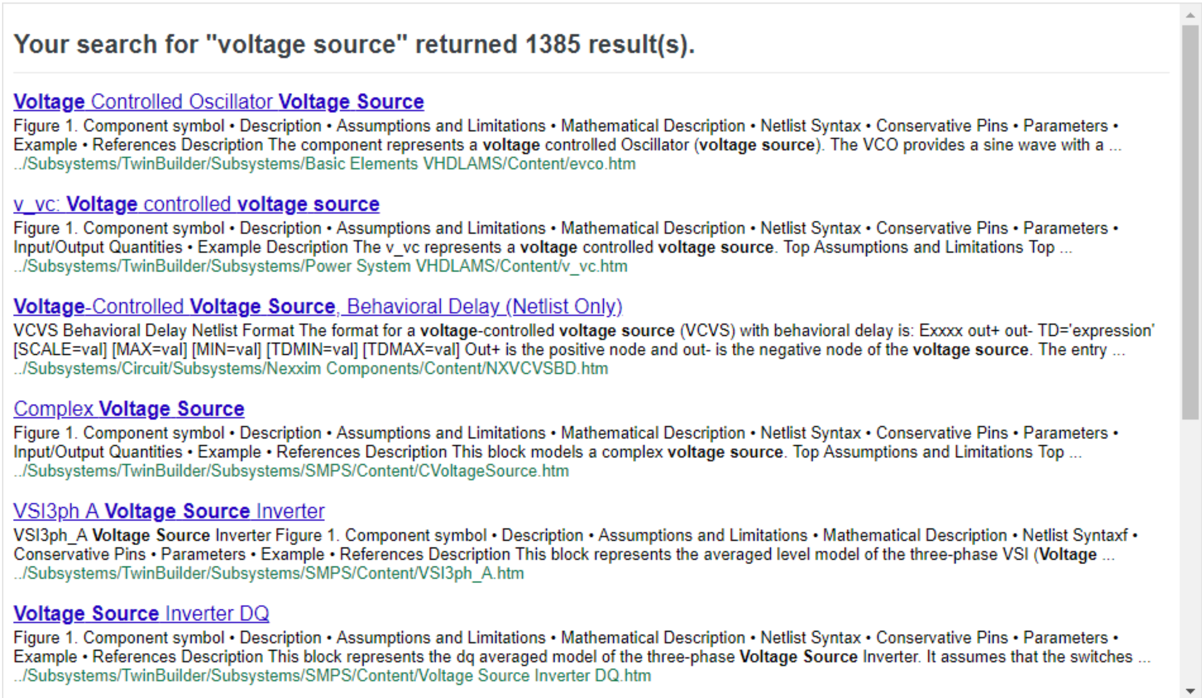
The help system provides different ways to find information and navigate quickly:

- Press **F1** on any open dialog box to open the relevant help topic.
- *A hierarchical table of contents* – Contents – Browse through the table of contents, expand entries, and close entries. Click an entry to see it in the content area.
- *A full text search* – To locate occurrences of a word or phrase that may be contained in the help, use the search function.

> **Note:**
>
> Ansys Electronics Desktop Student includes access to PDF documentation only.

Using the Search Function in the Help

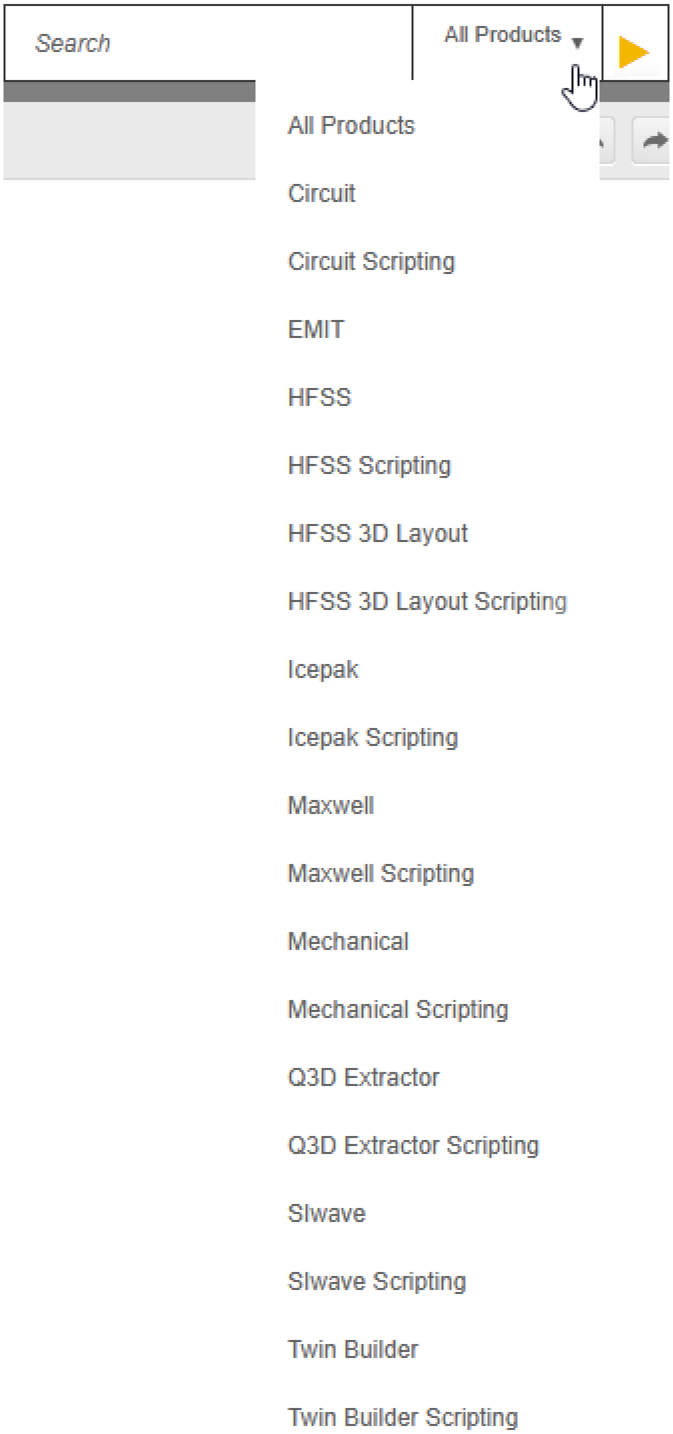
When you enter words or strings to search for in the help, the search engine lists all topics in which any of the words occur. For example, if you enter “voltage source” without the quotation marks, the results show all topics that contain “voltage” or “source.”



This method probably provides more hits than you want. The search function in the help provides several methods for making searches more specific.

Performing a Basic Search

1. Type the words or string in the search box.
   - If you are searching within the full Electronics help system, you see a search box that includes a drop-down filter for specifying a product, a product's scripting guide, or searching across all products. When you change the filter, the results dynamically reflect the selected filter.

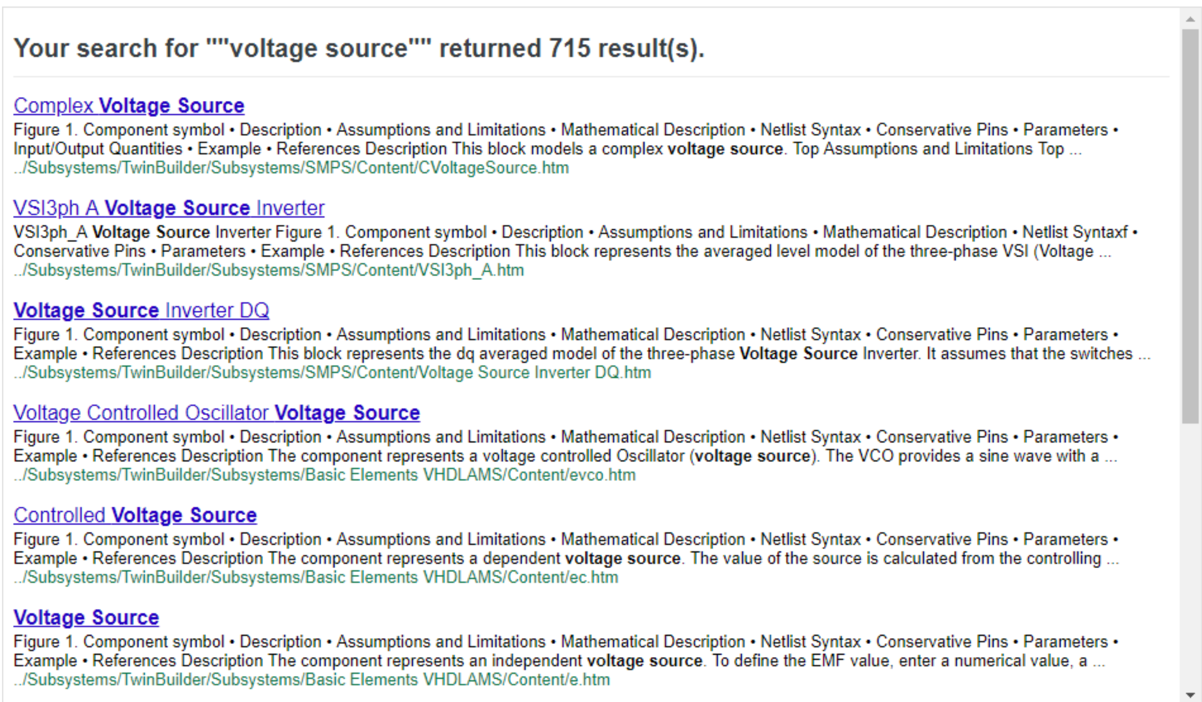


- If you accessed the help for a specific product by pressing **F1** in Electronics Desktop, the search box will have preselected that product. You can change the selection to a different product, a product's scripting guide, or all products.

2. Click on the topic you want in the results list.
    - Some topics in different products share the same title. When searching "All Products," check the URL below each link. The URL indicates the product under which the topic falls.
    - After clicking a topic, if you want to view a different topic, click your browser's back button to return to the results list.
    - To turn off highlighting on the page you are viewing, click the Remove Highlights icon ().

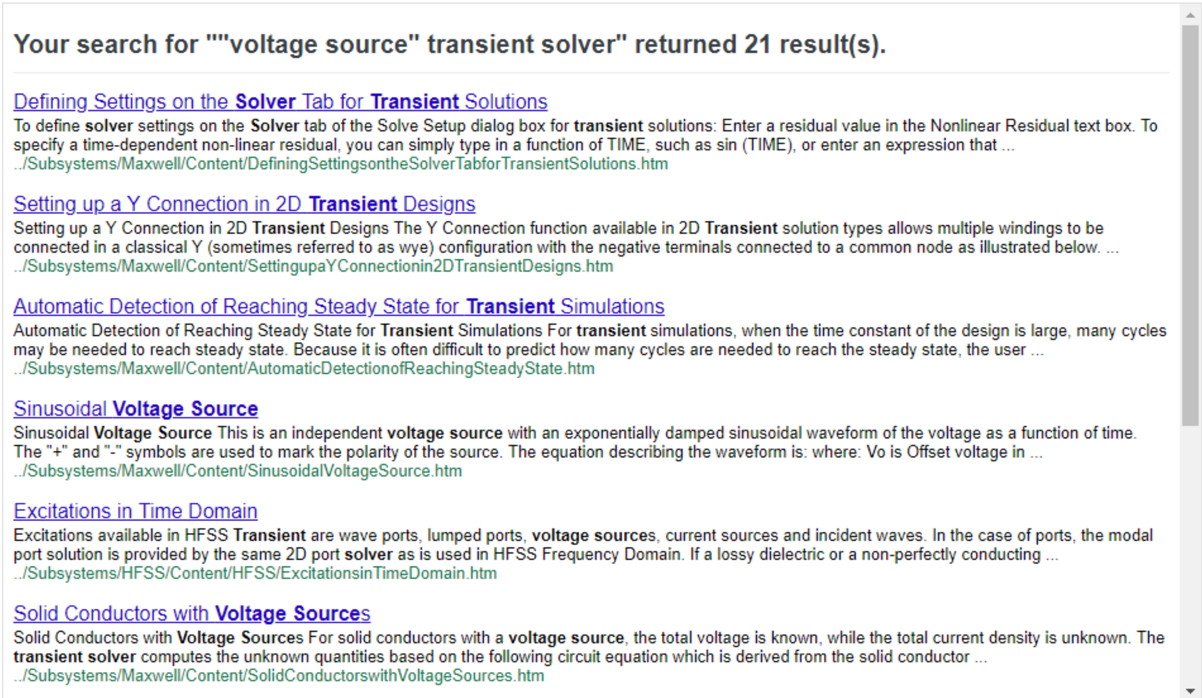
Searching with Quotation Marks

If you enter “voltage source” with quotation marks, the results show all topics that include the phrase.



As you can see, this returns far fewer results than the basic search.

To further limit the results, you can enter additional words, such as: “voltage source” transient solver



Note:

- Searches are not case sensitive, so you can type your search in uppercase or lowercase characters.
- You may search for any combination of letters (a-z) and numbers (0-9).
- Punctuation marks (period, colon, semicolon, comma, hyphen) are ignored during a search.
- When searching for a file name with an extension, group the entire string in quotation marks (for example, "filename.ext").

Using Boolean Operators

Note:

Functionality featured in the example(s) in this section applies to multiple design types.

You can also use boolean operators to affect the number of topics listed.

Operator (s)	Usage	Example(s)
AND	Lists all topics that contain all of the terms.	Net AND Selection

<table>
<tr><th>Operator (s)</th><th>Usage</th><th>Example(s)</th></tr>
<tr><td>+
&</td><td></td><td>Net + Selection
Net & Selection</td></tr>
<tr><td>OR
|</td><td>Lists all topics that contain any of the terms.</td><td>Net OR Selection
Net | Selection</td></tr>
<tr><td>NEAR</td><td>Lists all topics that contain the terms near the other terms.</td><td>Net NEAR Selection</td></tr>
<tr><td>NOT
!
^</td><td>Lists all topics that contain the first term but not the second.</td><td>Net NOT Selection
Net ! Selection
Net ^ Selection</td></tr>
</table>

Use parentheses to group terms and operators. For example:

- solver AND (Circuit) NOT HFSS NEAR dynamic
- solver AND (HFSS OR Circuit) NOT (Q3D OR 2d) NEAR dynamic
- “dynamic link” ! (HFSS | circuit)

> **Important:**
>
> Because the characters +, &, |, !, and ^ are used as operators, you cannot search for them in the help. Doing so will result in an error.

Help Conventions

Please take a moment to review how instructions and other useful information are presented in this documentation.

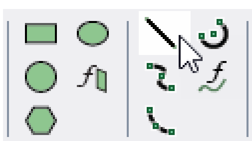
- Procedures are presented as numbered lists. A single bullet indicates that the procedure has only one step.
- Bold type is used for the following:
  - Keyboard entries that should be typed in their entirety exactly as shown. For example, “**copy file1**” means you must type the word **copy**, then type a space, and then type **file1**.
  - On-screen prompts and messages, names of options and text boxes, and menu commands. Menu commands are often separated by greater than signs (>). For example, “click **HFSS** > **Excitations** > **Assign** > **Wave Port**.”
  - Labeled keys on the computer keyboard. For example, “Press **Enter**” means to press the key labeled **Enter**.

- Italic type is used for the following:
  - Emphasis
  - The titles of publications
  - Keyboard entries when a name or a variable must be typed in place of the words in italics. For example, "**copy** *filename*" means you must type the word **copy**, then type a space, and then type the name of the file.
- The plus sign (+) is used between keyboard keys to indicate that you should press the keys at the same time. For example, "Press Shift+F1" means to press the **Shift** key and, while holding it down, press the **F1** key also. You should always depress the modifier key or keys first (for example, Shift, Ctrl, Alt, or Ctrl+Shift), continue to hold it/them down, and then press the last key in the instruction.

Accessing Commands: *Ribbons, menu bars,* and *shortcut menus* are three methods that can be used to see what commands are available in the application.

- The *Ribbon* occupies the rectangular area at the top of the application window and contains multiple tabs. Each tab has relevant commands that are organized, grouped, and labeled. An example of a typical user interaction is as follows:

  "Click **Draw** > **Line**"

  This instruction means that you should click the **Line** command on the **Draw** ribbon tab. An image of the command icon, or a partial view of the ribbon, is often included with the instruction.

- The *menu bar* (located above the ribbon) is a group of the main commands of an application arranged by category such File, Edit, View, Project, etc. An example of a typical user interaction is as follows:

  "On the **File** menu, click the **Open Examples** command" means you can click the **File** menu and then click **Open Examples** to launch the dialog box.

- Another alternative is to use the *shortcut menu* that appears when you click the right-mouse button. An example of a typical user interaction is as follows:

  "Right-click and select **Assign Excitation> Wave Port**" means when you click the right-mouse button with an object face selected, you can execute the excitation commands from the shortcut menu (and the corresponding sub-menus).

Getting Help: Ansys Technical Support

For information about Ansys Technical Support, go to the Ansys corporate Support website, http://www.ansys.com/Support. You can also contact your Ansys account manager in order to obtain this information.

All Ansys software files are ASCII text and can be sent conveniently by e-mail. When reporting difficulties, it is extremely helpful to include very specific information about what steps were taken or what stages the simulation reached, including software files as applicable. This allows more rapid and effective debugging.

Help Menu

To access help from the Help menu, click **Help** and select from the menu:

- **[product name] Help** – opens the contents of the help. This help includes the help for the product and its *Getting Started Guides*.
- **[product name] Scripting Help** – opens the contents of the *Scripting Guide*.
- **[product name] Getting Started Guides** – opens a topic that contains links to Getting Started Guides in the help system.

Context-Sensitive Help

To access help from the user interface, press **F1**. The help specific to the active product (design type) opens.

You can press **F1** while the cursor is pointing at a menu command or while a particular dialog box or dialog box tab is open. In this case, the help page associated with the command or open dialog box is displayed automatically.

Getting Help from Ansys Technical Support

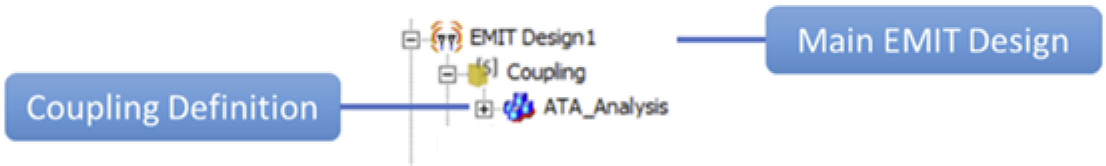
To contact Ansys technical support staff in your geographical area, please log on to the Ansys corporate website, ansys.com/support. You can also contact your Ansys EM account manager in order to obtain this information.

E-mail can work well for technical support. All Ansys EM software files are ASCII text and can be sent conveniently by email. When reporting difficulties, it is extremely helpful to include very specific information about what steps were taken or what stages the simulation reached. This allows more rapid and effective debugging.

2 - EMIT Design Environment

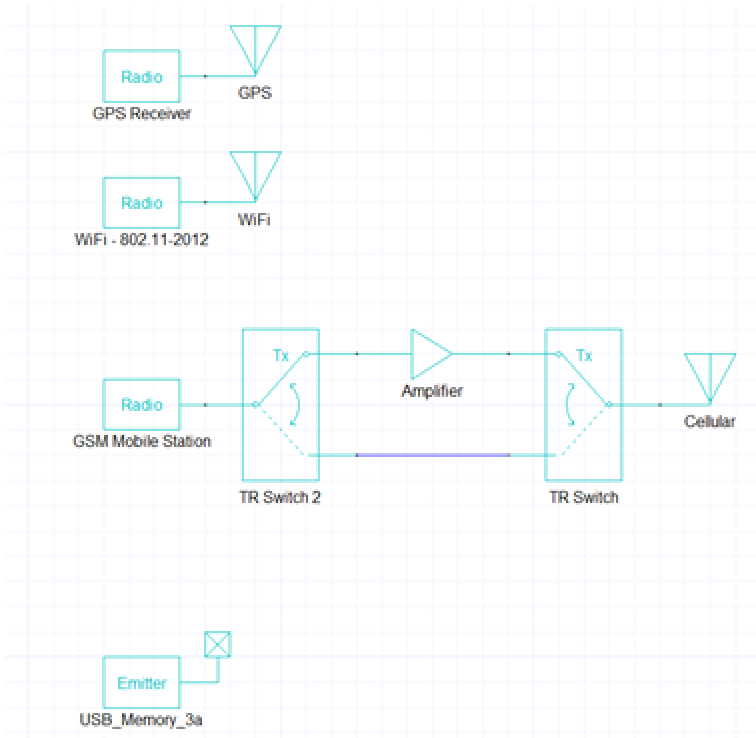
The EMIT design environment makes use of the schematic editor in the Ansys Electronics Desktop and provides an interface for creating and editing EMIT designs.

An EMIT design consists of two elements as shown below.



Main EMIT Design

The main EMIT design is created/edited using the schematic editor to configure antennas, emitters, radios, and other RF components that are to be part of the design. In the main EMIT design, RF system blocks and antennas are connected together to configure the wireless scenario to be analyzed. The arrangement of the radios and emitters in the EMIT design schematic editor are used to order how they appear in the analysis & results window. The figure below shows an example EMIT design schematic that includes three radios, three antennas, one emitter, and other RF components (two TR switches and an amplifier).



Related Topics

Schematic Editor

Coupling Definition

The wideband coupling between all antennas and emitters in an EMIT design is defined via links to an HFSS design or by using one of the built-in EMIT coupling models defined via the Coupling Editor.

Related Topics

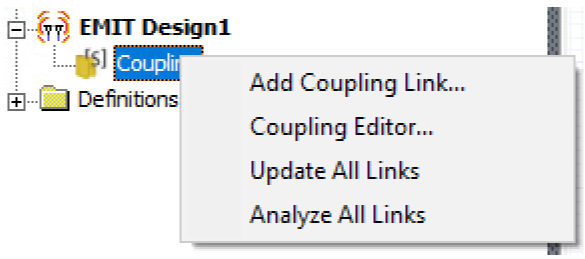
Coupling

Coupling Editor

3 - Coupling

The Coupling setup in EMIT provides definitions for how antennas and emitters in an EMIT design interact (couple) with each other. Coupling between all antennas and emitters in the EMIT design must be specified in order to perform the simulation. However, EMIT uses default values when specific coupling models are not explicitly defined.

All EMIT designs contain a single Coupling node. Right-clicking on this node in the Project Manager provides the following options:



Add Coupling Link allows you to add a link to HFSS and HFSS 3D Layout designs to use for coupling in the EMIT design.

Coupling Editor brings up the Coupling Editor where all of the coupling between antennas and emitters in the EMIT design can be set up and visualized.

Update All Links updates the port list, setups, sweeps, and associated coupling data for all linked designs.

Analyze All Links runs all unsolved HFSS and HFSS 3D Layout simulations, and updates the port list, setups, sweeps, and associated coupling data for all linked designs.

> **Note:**
> Make sure the HFSS and HFSS 3D Layout designs are in the same project as the EMIT design.

You can also access the **Coupling Editor** via the **Coupling Editor** icon on the **Simulation** ribbon menu of the EMIT design: 

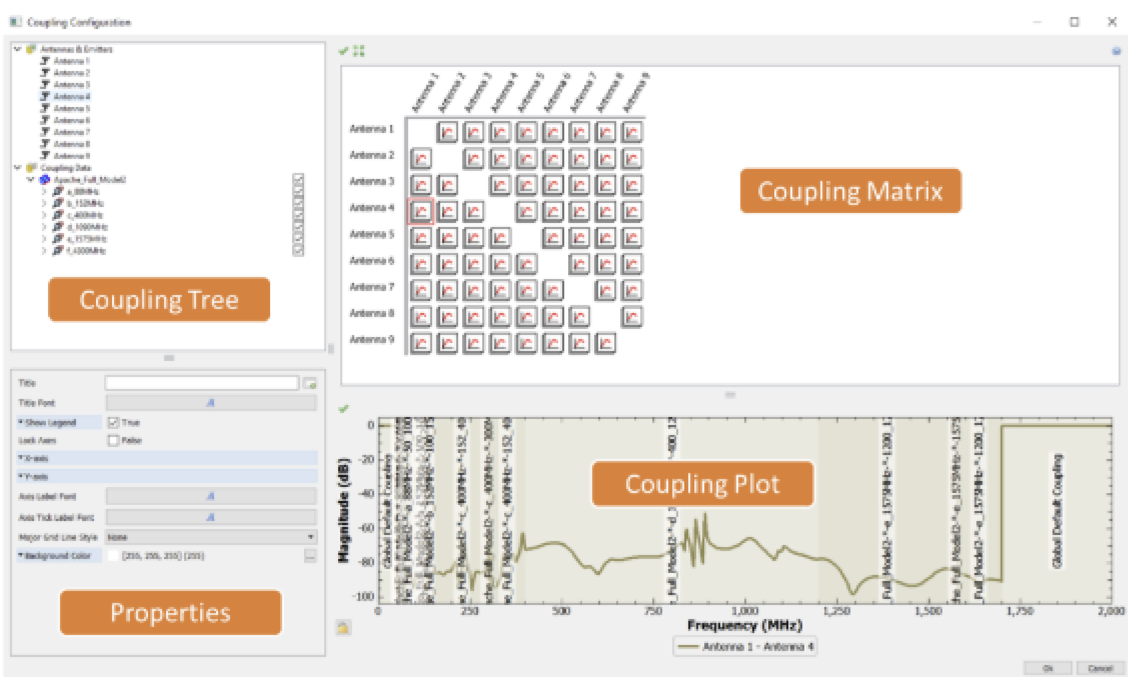
Related Topics

- Coupling Editor
- Coupling Links

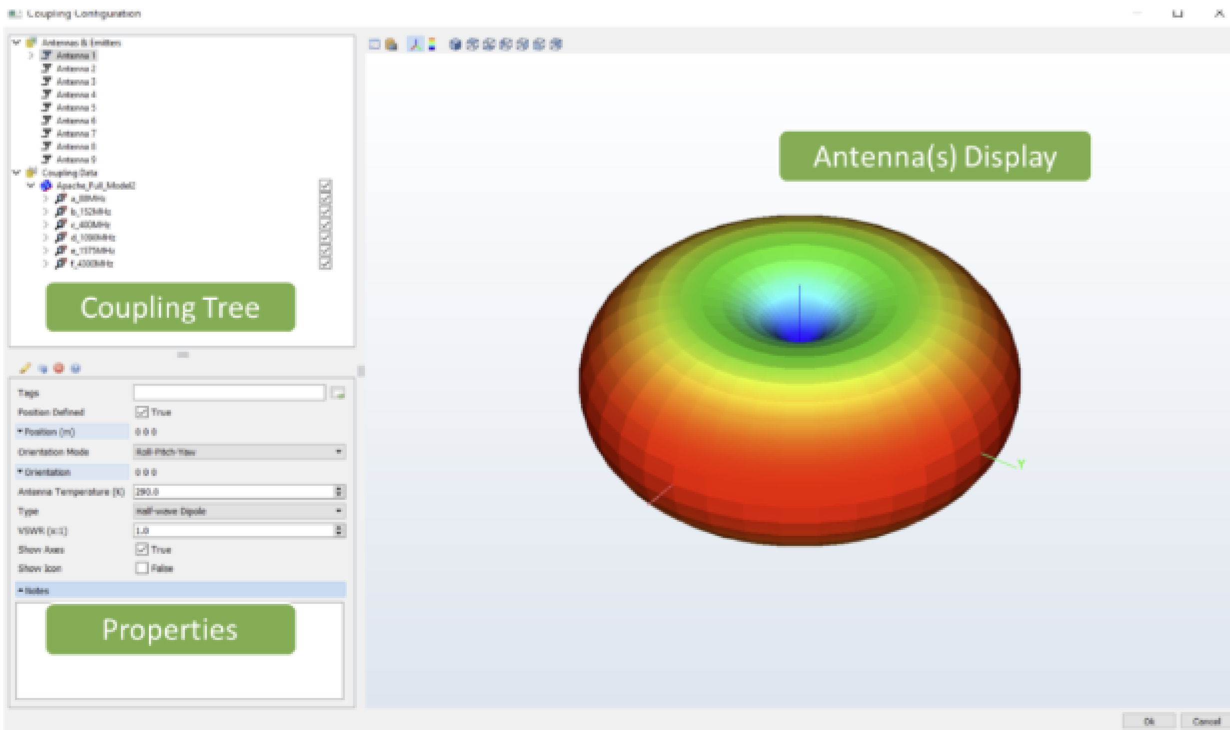
Coupling Editor

The EMIT Coupling Editor provides a single multi-pane multi-function window that allows the user to manage and visualize all of the antenna, emitter and coupling associated with an EMIT design. The Coupling Editor is opened via the right-click menu on the Coupling folder in the Project Manager or by double-clicking it. You can also access the **Coupling Editor** via the **Coupling Editor** icon on the **Simulation** ribbon menu of the EMIT design: 

The contents of each pane in the Coupling Editor depends on the current selection in the dialog box. When an item from the Coupling section of the tree is selected, the dialog looks like the image below, showing information on the selected coupling.

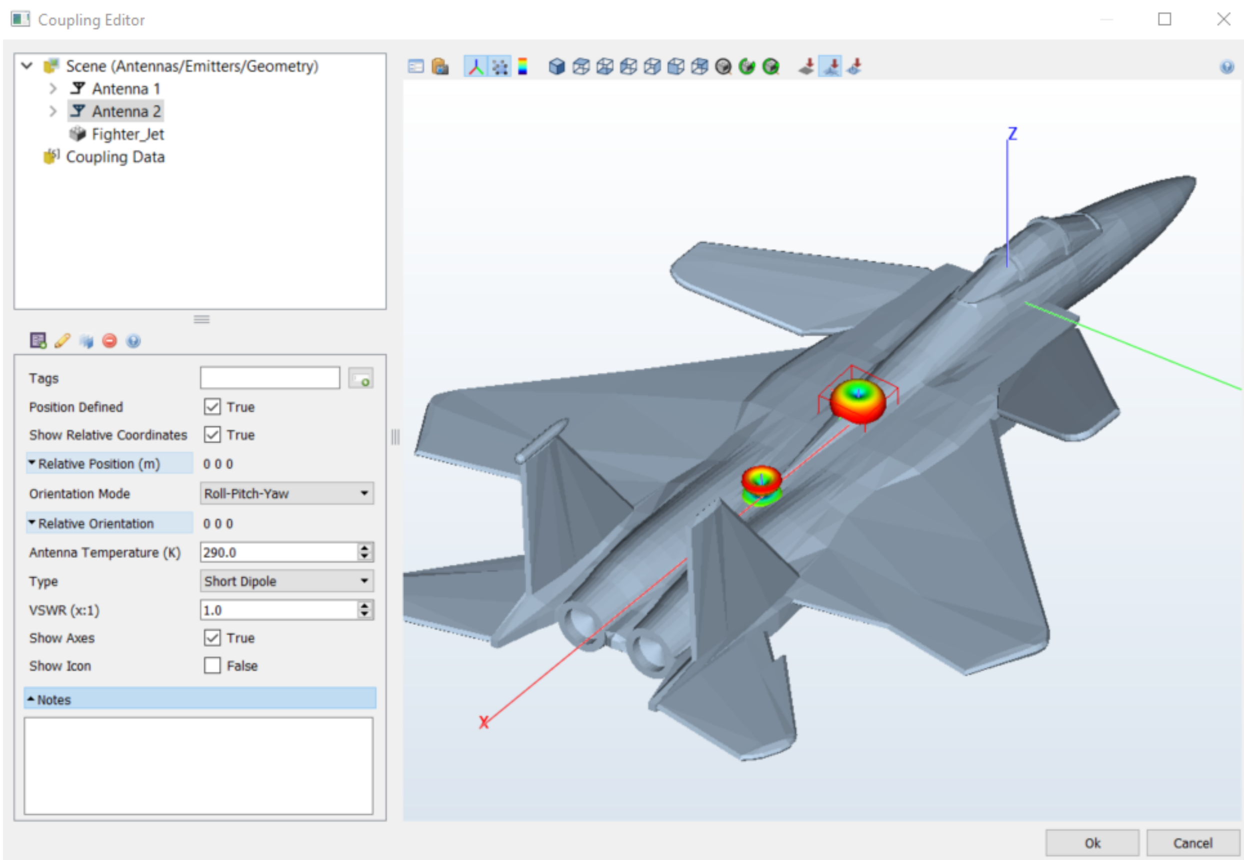


When an item from the Antennas & Emitters section is selected, the dialog shows the antennas' locations and patterns (if defined) as shown below. Selecting one of the antennas displays that antenna's pattern (if defined) and selecting the parent Antennas & Emitters shows all of the antennas (if defined) in the entire design. Note that since Emitters do not contain pattern or position information, these are not shown in the 3D display.



3D Window

The **3D window** in the Coupling Editor is shown when the Antennas & Emitters folder or any individual antenna or emitter is selected in the Coupling Tree. It displays the locations and patterns of all antennas that have a position defined. If an individual antenna is selected, then that antenna is highlighted with a red bounding box in the 3D window.



3D Window Actions

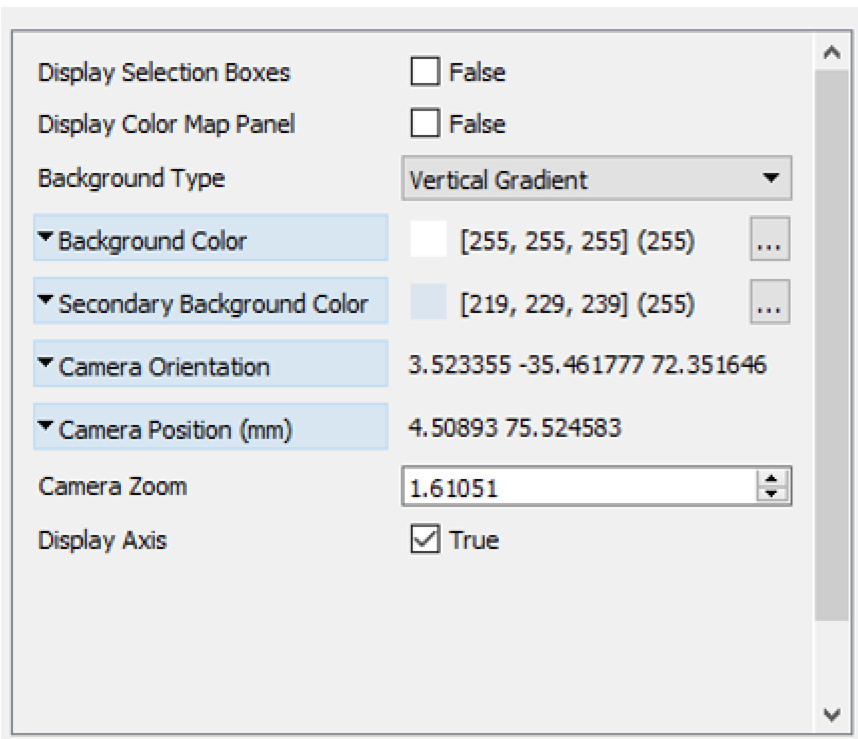
The table below summarizes keyboard/mouse actions available in each 3D display window. To use them, the mouse must be positioned in the 3D window.

Change View Angle (Rotate)	Left-button Drag Middle-button Drag Alt + (Arrow Keys)
Pan (Translate)	Shift + (Left-button Drag) Ctrl + (Middle-button Drag) Ctrl + (Arrow Keys)

Zoom In/Out	Ctrl + Shift + (Left-button Drag) Ctrl + (+/- Keys) Mac: Option + Shift + (Left-button Drag) Shift + (Middle-button Drag) Scroll Wheel Up/Down Page Up/Page Down Keys
Place Antenna at Current Mouse Location	With antenna node selected: Ctrl + (Left-button)
Restore Default View	Left-button Double Click Z Key

3D Window Property Panel

The 3D window property panel, accessible by clicking the Setting icon in the 3D window toolbar, contains settings to control the look of the 3D window in the viewing area.



Display Selection Boxes

Toggle on/off display of bounding box outlines to highlight the currently selected node and its children. When enabled, any objects displayed in the 3D window corresponding to the currently selected node or its children are drawn with an encompassing wireframe bounding box.

Display Color Map Panel

Toggle on/off display of the color map panel on the right side of the 3D window.

Background Type

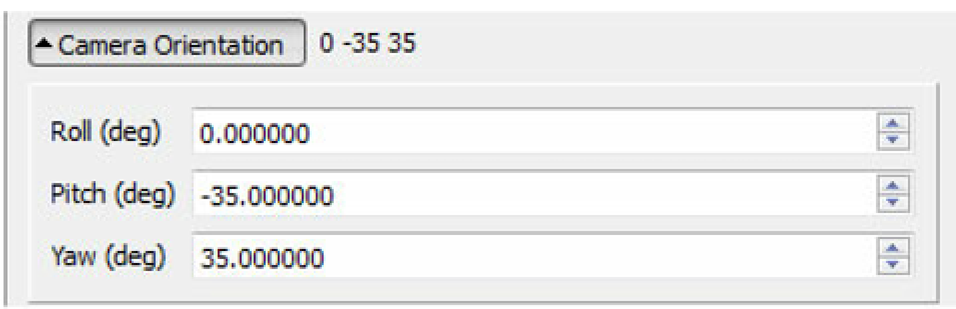
Select the type of color gradient for the background of the window. By default, the 3D window has a purple background with a vertical gradient from dark to light purple. The available choices are solid (no gradient), radial gradient, horizontal gradient, and vertical gradient.

Background Color / Secondary Background Color

Set the background color of the 3D window. Expand the panel folder to set the red-green-blue (RGB) levels for the background color. Click the button to choose the color from a palette. For gradient background types, a secondary color must also be configured.

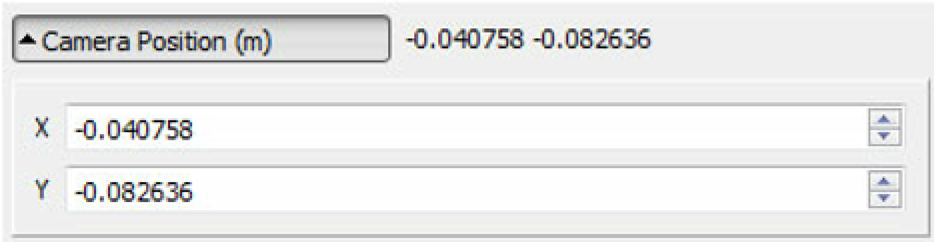
Camera Orientation

Set camera (view) angle in terms of Roll, Pitch, and Yaw angles. It is usually easiest to modify the view angle interactively with mouse drag operations in the 3D window. However, when more precise or repeatable control is needed, the view angle can be set through direct numeric entry by expanding the panel and setting the Roll-Pitch-Yaw angles. The meaning of Roll, Pitch, and Yaw is illustrated in the diagram below, which indicates how the displayed global coordinate axes will rotate relative to screen (window) coordinates (that is, the camera - you, the observer - effectively rotates in the opposite direction). The ordering of rotations is Roll, then Pitch, then Yaw about the global axes, such that Roll rotation always occurs about the axis pointing out of the screen and Yaw rotation always occurs about the current z-axis of the global coordinate system.



Camera Position

Set the 2D translation of the scene in screen (window) coordinates. It is usually easiest to pan the scene left/right and up/down interactively with mouse drag operations in the 3D window. However, when more precise or repeatable control is needed, the scene position (opposite of camera position) can be set through direct numeric entry by expanding the panel and setting its X and Y values.



Increasing X causes the scene to shift to the right (that is, the camera moves left). Increasing Y causes the scene to shift up.

Camera Zoom

Set the zoom (magnification) factor of the scene. It is usually easiest to zoom-in and out interactively using the mouse scroll wheel or various mouse/keyboard action substitutes while the mouse is pointed somewhere in the window. However, you can also control the zoom factor in the 3D window node property panel by adjusting the zoom control. The default zoom level is 1, meaning that the entire displayable content of the 3D window can be seen with some margin around the edges of the view window. This can be adjusted from 0.1 to 1 million, allowing close focus on small details. When adjusting the zoom, the center point of the view window remains fixed. Hence, to focus on some small part in the 3D window, first pan the view so that the part is in the center of the view window, and then increase the zoom factor.

Display Axis

Toggle on/off display of all global and local coordinate system axes in the 3D window

3D Window Toolbar

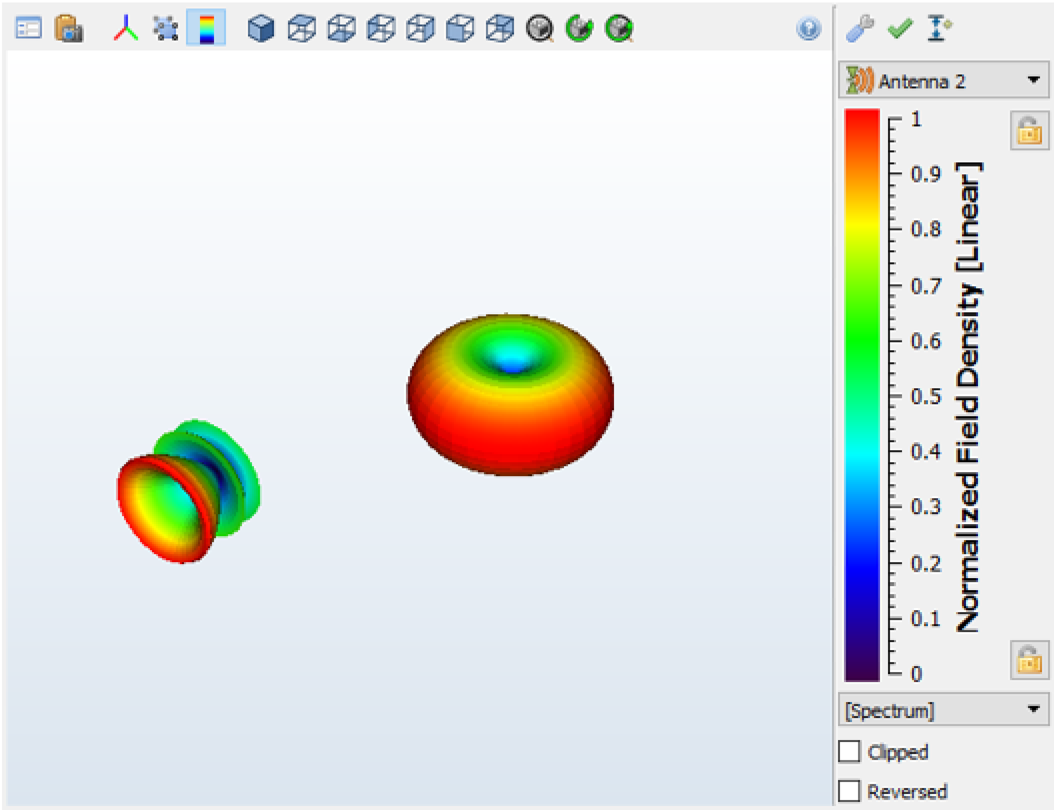
The 3D window toolbar provides the functions in the table below (left to right as they appear in the toolbar).

Settings	Shows the 3D window property panel to control the look of the 3D window in the viewing area.
Copy Snapshot	Copies an image of the 3D window to the clipboard.
Display Axis	Toggle on/off display of all global and local coordinate system axes.
Display Bounding Box	Toggle on/off display of the color bar panel in the 3D window.
Show Color Map	Toggle on/off display of the color bar panel n the 3D window.
Default View	Reset the 3D window to its default view settings
Top View	Set the 3D window to view displayed objects from the top.
Bottom View	Set the 3D window to view displayed objects from the bottom.

Right View	Set the 3D window to view displayed objects from the right.
Left View	Set the 3D window to view displayed objects from the left.
Front View	Set the 3D window to view displayed objects from the front.
Back View	Set the 3D window to view displayed objects from the back.
Zoom to Node	Focus the 3D window on the selected node by moving the camera position and zooming on the contents of the node.
Return Around Node	Set the currently selected node coordinates to be the center of rotation for the current 3D window.
Zoom to and Rotate Around Node	Perform the two previous actions, Zoom and set the Rotate Point on the selected node.
Place Object	Place antenna at selected point (Antenna node must be selected in the Coupling Editor Tree).
Place Object Tangent to the Surface	Place antenna at selected point with its z-axis perpendicular to the surface (Antenna node must be selected in the Coupling Editor Tree).
Place Object Oriented toward the Camera	Place antenna at selected point with its z-axis pointing toward you (Antenna node must be selected in Coupling Editor Tree).

Color Bar Panel

The color bar panel appears on the right side of the 3D window when the **Show Color Map** button is toggled on in the 3D window toolbar.



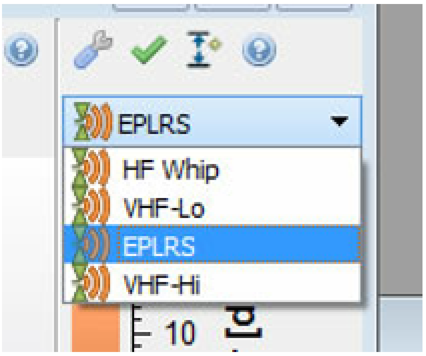
Color Bar Toolbar

The following options are available in the toolbar above the color bar:

Configure Extents	Select node that owns this map to access/adjust color bar numeric extents.
Default Scale	Revert min/max extents to those of the data.
Round Scale	Change min/max extents to round numbers.

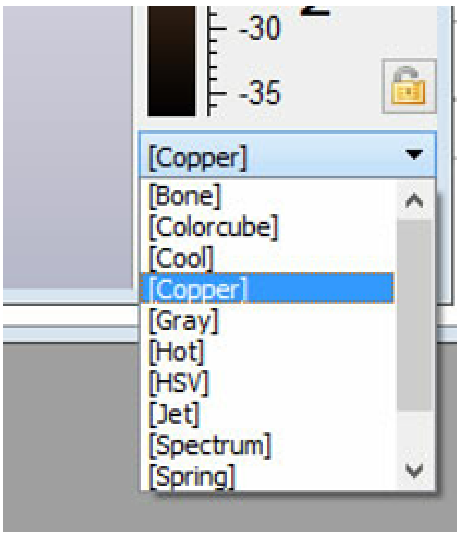
Color Map Selector

Selects the antenna for the color map to be displayed.



Color Scheme Selector

The Color Scheme Selector, located immediately below the vertical color bar, is used to select the color palette for mapping real-number data to colors. The default color scheme is Spectrum. Several other color schemes are built in, including popular ones like Gray, HSV, and Jet.



There are two aspects to mapping data to colors. The first is the available spectrum of colors, which are arranged along a single dimension in the color bar. This spectrum is determined by the choice made in the Color Scheme Selector. The second is the range of real numbers, minimum to maximum, that are linearly mapped into the selected one-dimensional spectrum. This range is indicated by the numbered axis to the right of the color bar. The range extents can be modified in two ways:

- Dragging the color bar up or down using the mouse. The detailed behavior depends on the states of the lower and upper Color Map Locks.
- Entering the Minimum/Maximum color map values in the node property panel under the Color Map panel folder. This panel folder is not in the color bar panel itself. Rather, it is in the property panel of the Pattern node of the Antenna that is currently selected in the Color Map Node Selector. The relevant property panel can be exposed by clicking the Configure Extents button in the color bar toolbar.

Color Map Locks

The locked/unlocked /  toggle buttons appearing to the right of the labeled color bar (one at the top and one at the bottom) control whether the Maximum or Minimum extent of the color bar is held fixed when vertically dragging the color bar up or down with the mouse. For example, if the color bar represents a decibel (dB) scale, and both locks are unlocked, then dragging with the mouse simultaneously adjusts the color bar Minimum and Maximum by the same amount; the dynamic range (dB span) of the color bar remains fixed. Alternatively, if the color bar represents a linear field magnitude scale, it usually makes sense to lock the lower extent at zero and leave the upper extent unlocked. That way, when you drag the color bar, only the upper extent is modified. In general, the locks give you the flexibility to control how the color bar responds to drag operations.

The color bar lock toggles are mirrored next to the Minimum and Maximum settings in the corresponding Color Map panel in the relevant node's property panel. That is, the locks can be toggled either in the Color Bar panel of the 3D window or in the Color Map panel of the node, and changes in their state remain synchronized.

Locking an extent prevents it from automatically updating due to changes in the selected polarization or field component being represented by the color-scale rendering of the result data. However, even when a color bar extent is locked, it can still be changed by direct numeric entry in the Color Map panel.

Clipped

This toggle button appears near the bottom of the color bar panel. When toggled on, scaled result data that fall below the current minimum of the color map are not rendered. This is equivalent to rendering these values with complete transparency.

Reversed

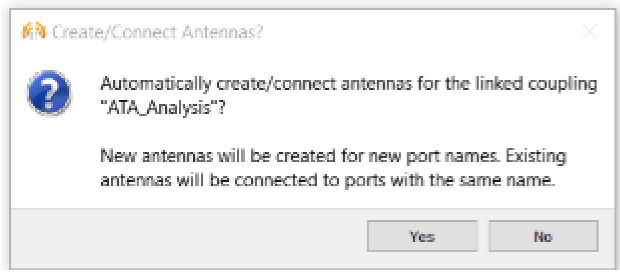
This toggle button appears at the bottom of the color bar panel. When toggled on, the colors in the color bar are vertically inverted. For example, if the selected Color Scheme Selector is Spectrum, the Minimum color map value is mapped to red instead of dark purple.

Coupling Editor

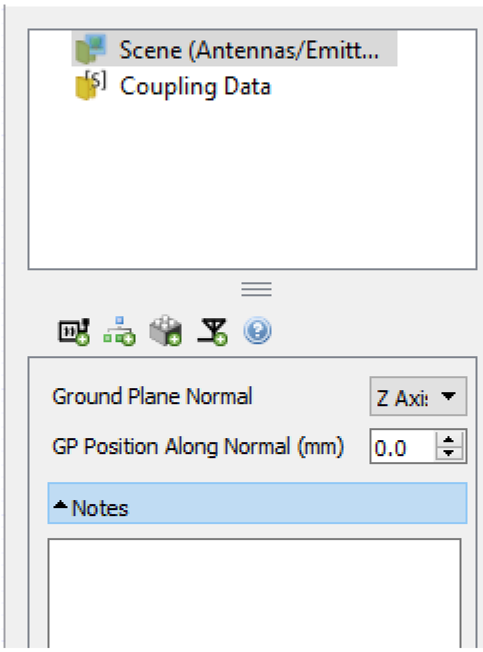
Scene (Antennas/Emitters/Geometry)

The Scene node in the Coupling tree contains all of the antennas, emitters, and geometry that are defined in an EMIT design. Antennas, emitters, and geometry can be added via the right-click menu of the Scene (Antennas/Emitters/Geometry) node. They can also be added from applicable coupling types in the Coupling Editor. Any antennas or emitters in the Coupling Editor automatically appear in the schematic and vice versa. Note that geometry (that is, CAD models) is displayed only in the Coupling Editor.

When you add an HFSS link to an EMIT design, the Coupling Editor opens and you are prompted to add new antennas for each of the HFSS ports if desired as shown below.



Selecting the Scene (Antennas/Emitters/Geometry) shows the properties in the figure below.



Antennas, emitters, and geometry can be added from the properties pane by clicking the icons for those actions at the properties pane.

The properties are defined in this pane:

Ground Plane Normal: Sets the direction of the normal to the ground plane. For example, a Ground Plane Normal along the Z Axis configures the ground plane (not shown in the 3D Scene) parallel to the X-Y plane.

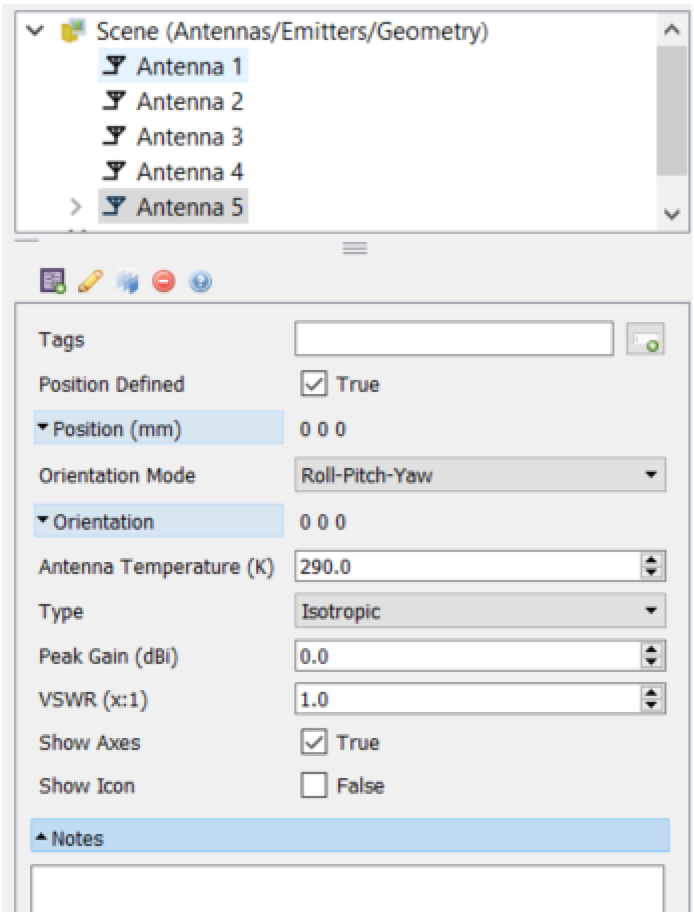
GP Position Along Normal: Sets the offset of the ground plane from the global origin.

Notes: Expand the Notes panel folder to view or edit scene-level user notes. The notes are stored with the project.

Antenna Properties

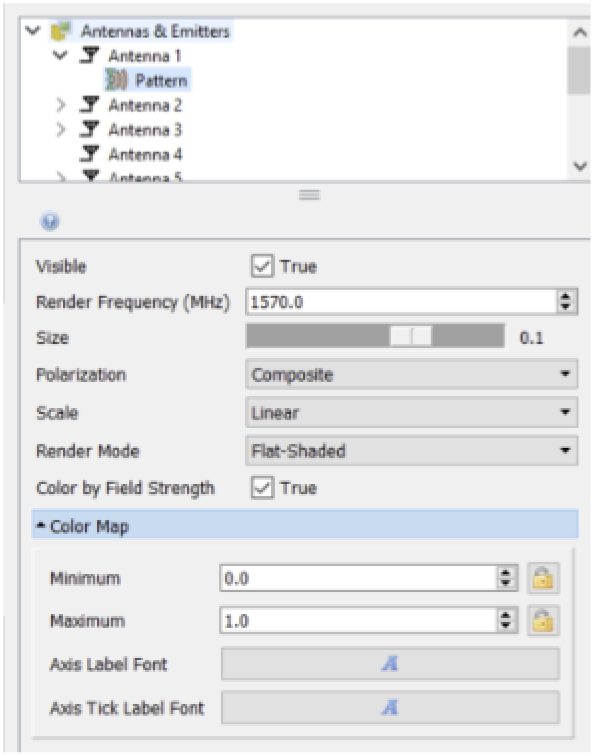
Selecting an antenna in the coupling tree shows the properties for an antenna in the property pane. (Note that antenna nodes can be moved in the tree by dragging and dropping.)

The options available in the Antenna property panel are described in the Antennas section.



Pattern Properties

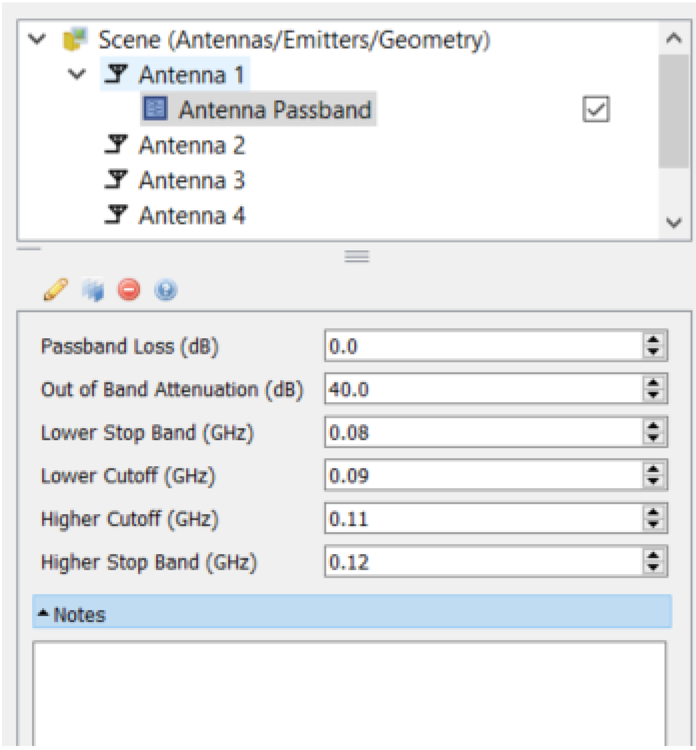
When a position is defined for an antenna, there is a Pattern node as a child of the Antenna node in the tree with properties available in the property pane as seen in the figure below.



The properties available are explained in the Pattern section.

Passband properties

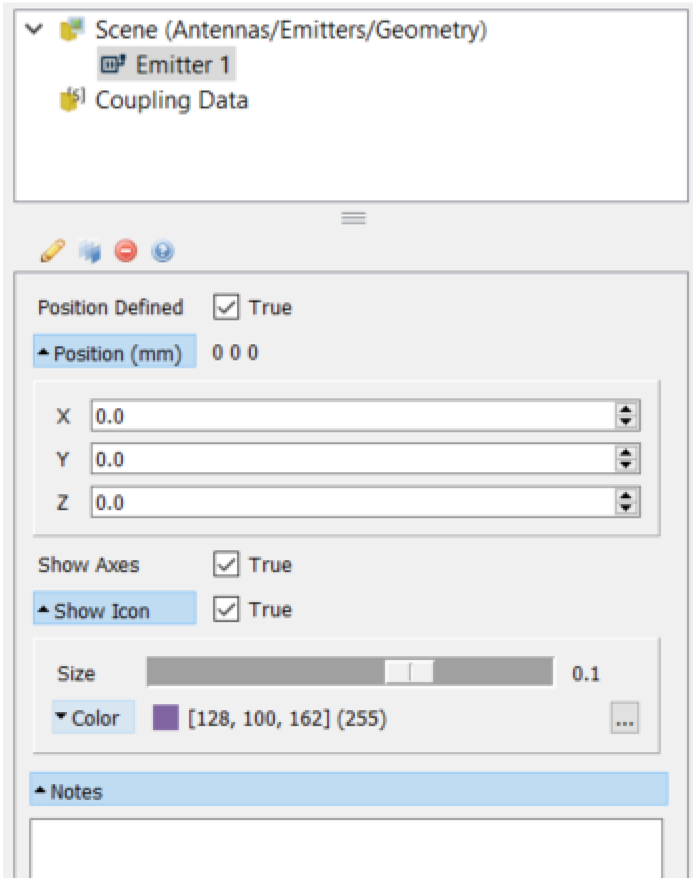
When a Passband is defined for an antenna, there is an Antenna Passband node shown as a child of the Antenna node in the tree with properties available in the property pane as seen in the figure below.



The properties available are explained in the Antenna Passband section.

Emitter Properties

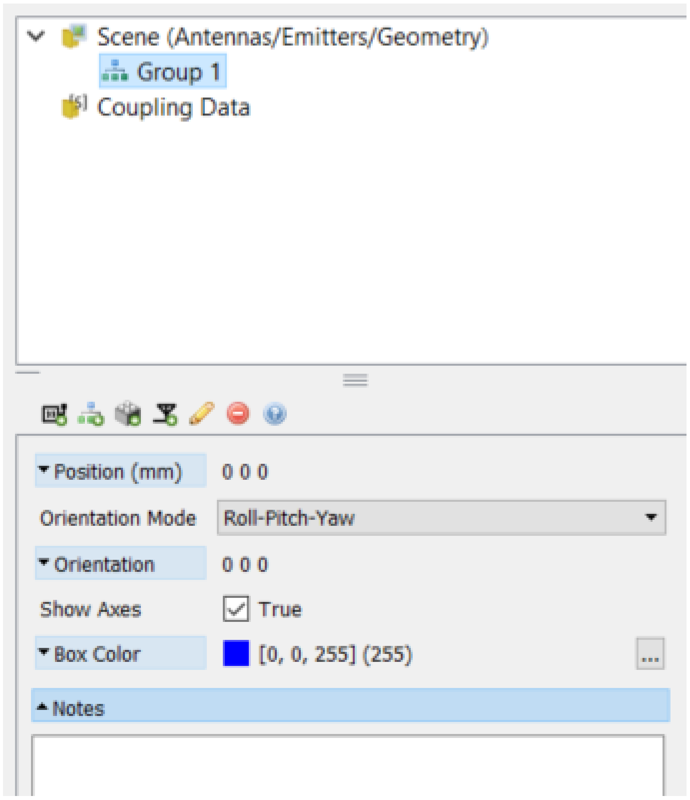
Selecting an emitter in the coupling tree shows the properties for the emitter in the property pane. The only options available in the Emitter Property panel are for setting the position of the emitter and visualizing the emitter in the Scene.



Group Properties

Selecting a Group in the coupling tree shows the properties for the Group in the property pane. The Group nodes can be moved in the tree by dragging and dropping them.

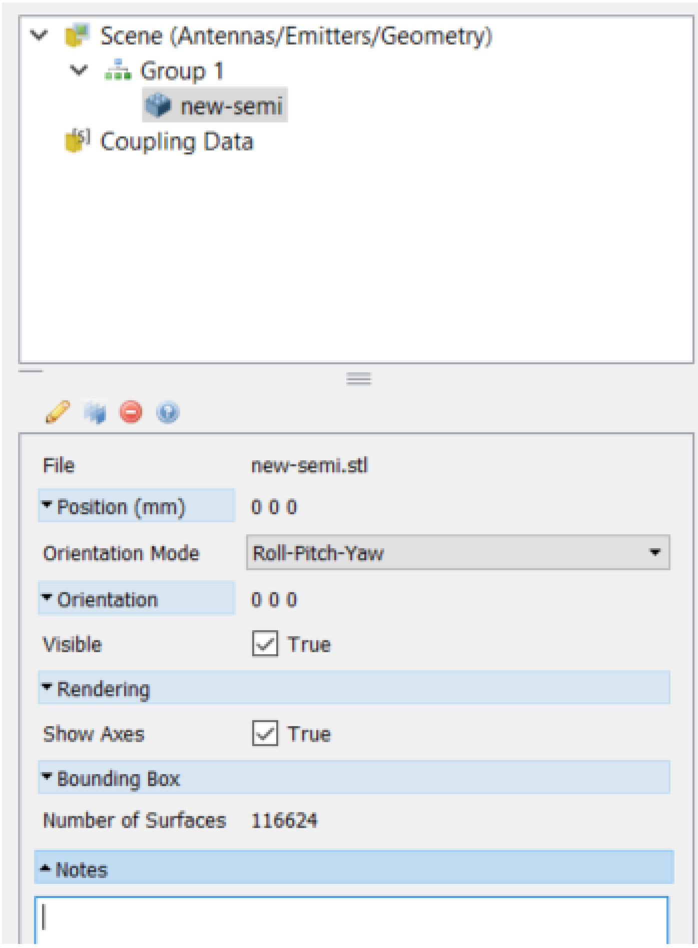
The options available in the Group property panel are described in the Group section.



Geometry Properties

Selecting a Geometry in the coupling tree shows the properties for the Geometry in the property pane. The Geometry nodes can be moved in the tree by dragging and dropping them.

The options available in the Geometry property panel are described in the Geometry section.

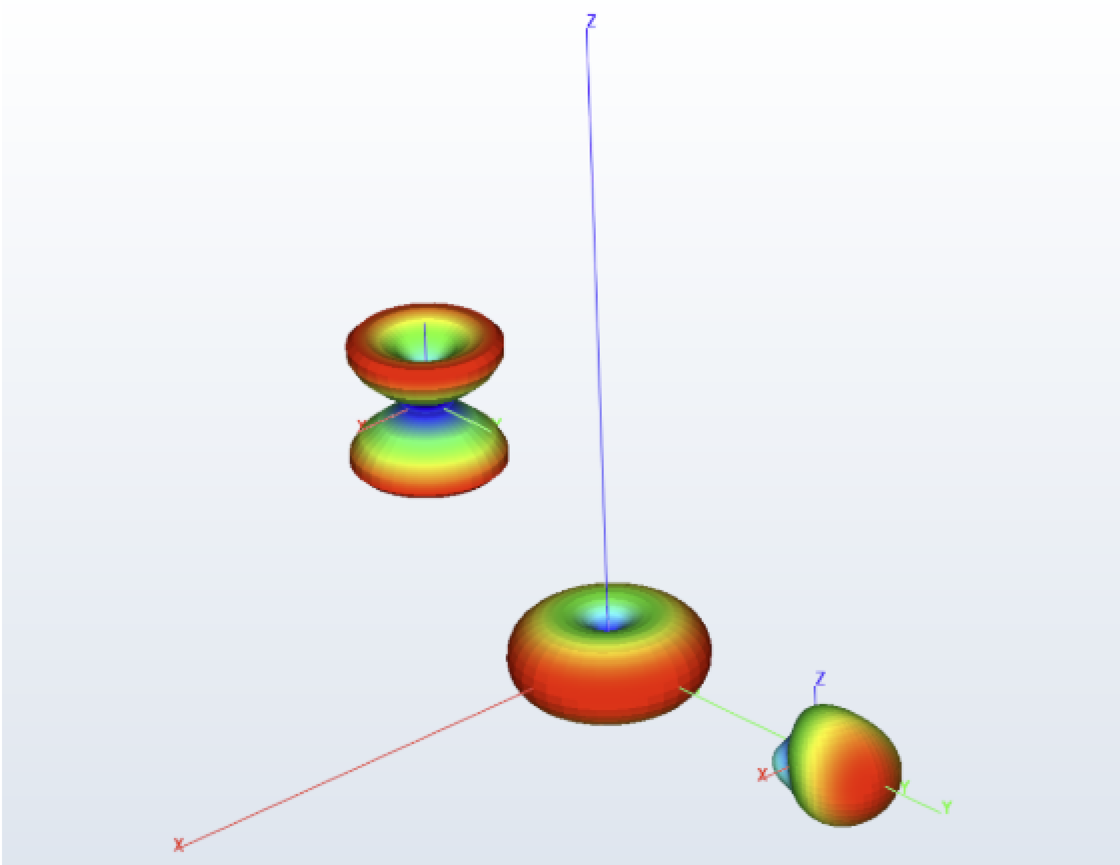


Visualizing Antennas

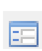
When the Scene (Antennas/Emitters/Geometry) node or any single Antenna/Emitter/Group/Pattern/Geometry node is selected in the Coupling tree, a 3D visualization window is displayed. The pattern (and/or icon) is displayed for antennas/emitters that have a position defined.

When an Antenna or Pattern node is selected, only the pattern/icon for that antenna is displayed. When the Antennas & Emitters node is selected, the patterns/icons for all antennas are displayed in their defined positions.

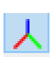
An example of a 3D scene consisting of three antennas with positions and patterns defined is show below.



The following actions are accessible via the toolbar on the 3D window:

Settings Displays visualization settings for the 3D window in the properties pane.

Copy Snapshot Copies a snapshot of the 3D window to the system clipboard.

Display Axis  Toggle on/off display of the axis in the 3D window.

Display Axis Labels Toggle on/off display of the axis labels (x, y, z) in the 3D window.

> **Warning:**
> With some operating system and GPU combinations, toggling on the axis labels may result in a crash.

Display Bounding Boxes Toggle on/off display of bounding boxes to highlight the currently selected node.

Show Color Map Toggle on/off display of the color bar panel in the 3D window.

Default View Reset the 3D window to its default view settings.

Top View Set the 3D window to view displayed objects from the top.

Bottom View Set the 3D window to view displayed objects from the bottom.

Left View Set the 3D window to view displayed objects from the left.

Right View Set the 3D window to view displayed objects from the right.

Front View Set the 3D window to view displayed objects from the front.

Back View Set the 3D window to view displayed objects from the back.

Zoom to Node Focus the 3D window on the selected node by moving the camera position and zooming on the contents of the node.

Rotate around Node Set the currently selected node coordinates to be the center of rotation for the current 3D window.

Zoom to and Rotate around Node Perform the two previous actions, Zoom and set the Rotation Point, on the selected node.

Place Object Place antenna at selected point (Antenna node must be selected in the Scene tree).

Place Object Tangent to Surface Place antenna at selected point with its z-axis perpendicular to the surface (Antenna node must be selected in the Scene tree).

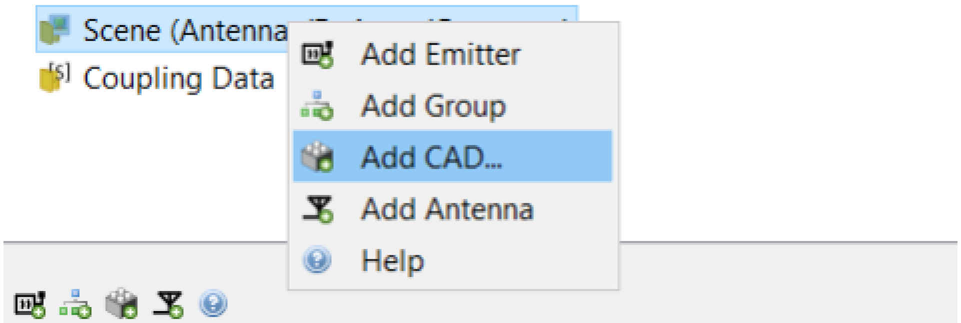
Place Object Oriented toward Camera Place antenna at selected point with its z-axis pointing toward you (Antenna node must be selected in the Scene tree).

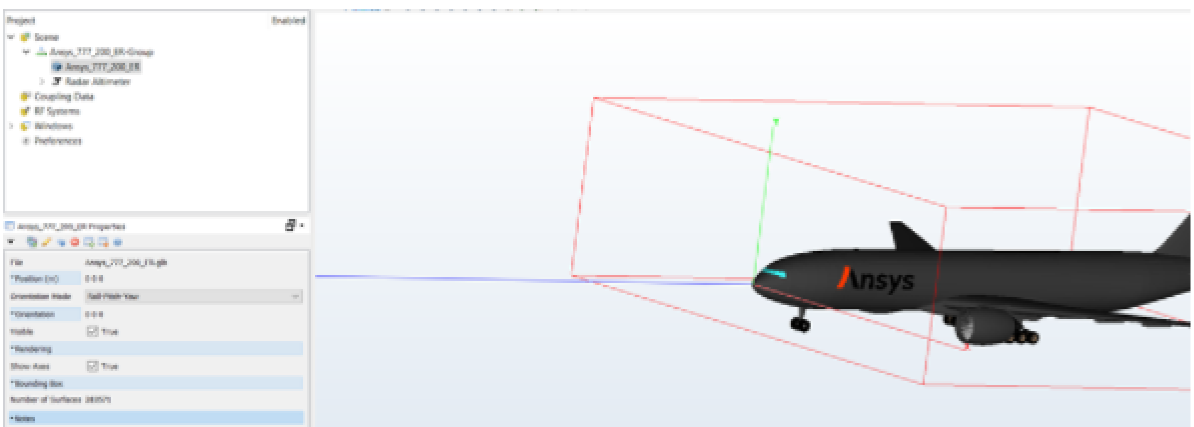
Geometry

Groups

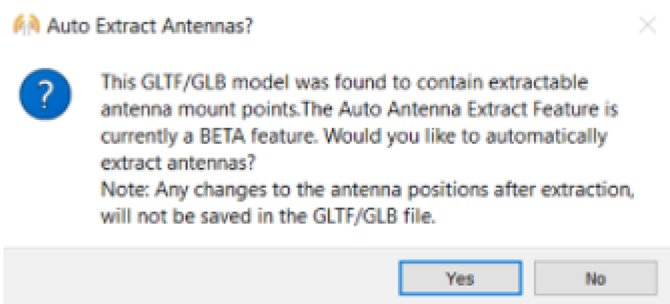
Geometry

Geometry is added to an EMIT design from the Coupling Editor dialog box. On the Scene (Antennas/Emitters/Geometry) node or any Group node, right-click and select **Add CAD** or select the **Add CAD** icon in the property panel.





Within Ansys Discovery there is the ability to export models that have special attach points as GLTF/GLB models. EMIT can support the import of GLTF/GLB models. EMIT's GLTF/GLB import also supports the coloring of individual primitives, by grabbing the base color factors assigned to any primitive but cannot utilize texture packages. As the GLTF/GLB model is being loaded, EMIT scans for any attach points. If a GLTF/GLB model does not have any attach points to convert into antennas, the model is added to the parent node as a regular CAD node.

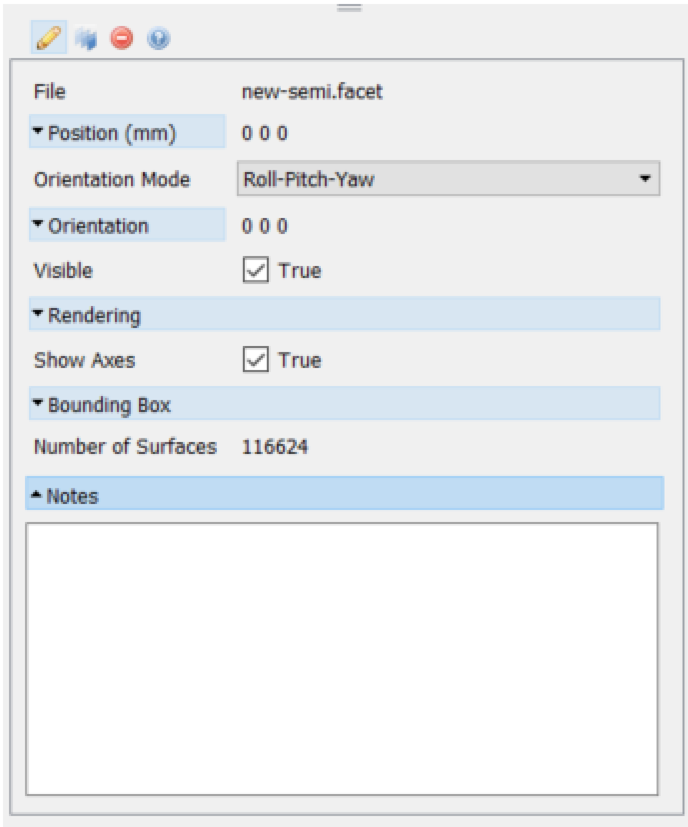


If there are attach points defined within the GLTF/GLB model, EMIT asks if you want to automatically create antennas for each attach point. If you agree, the CAD model and the extracted antennas are grouped together under the parent node.

> **Note:**
> If a GLTF/GLB model contains any primitives that are not designed to be rendered as triangles, a CADNode warning is displayed.

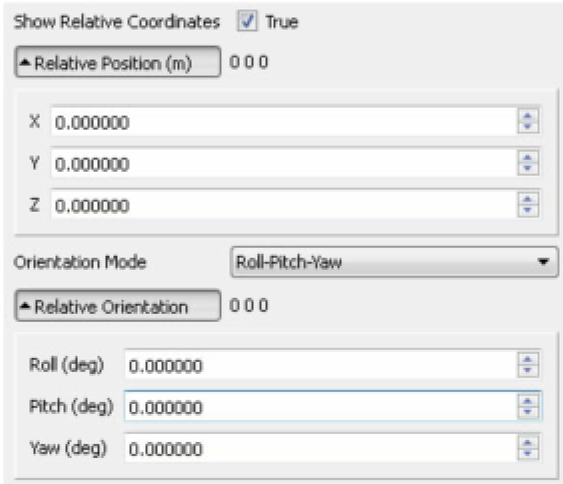
> **Note:**
> Any changes to the antenna positions after extraction are not saved in the GLTF/GLB file.

Each geometry node has the following properties defined in the Property panel.

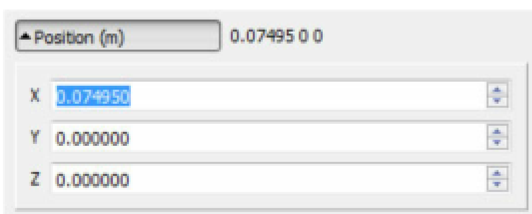


File: Displays the name of the imported CAD file.

Show Relative Coordinates: Toggle on/off display of Position and Orientation relative to placement coordinates instead of parent-node coordinates. Note that this setting is not available until after the CAD model has been interactively placed for the first time. When switched on, the *Position* and *Orientation* panel folders are renamed to *Relative Position* and *Relative Orientation*. They then indicate the CAD model position and orientation (that is, the CAD node coordinate system) relative to the placement coordinates. When switched back off, these panel folders revert to their original labels and indicate the CAD model coordinate system relative to the coordinate system of its parent node.



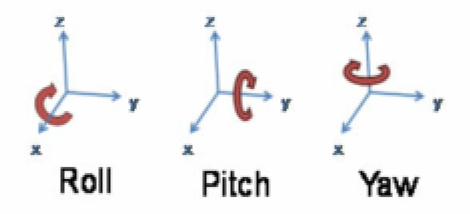
Position: Set the position (origin) of the CAD node coordinate system. In its collapsed state, the Position panel folder shows the current location (in meters by default) of the CAD model in the scene with respect to the coordinate system of its parent node. Expand the Position panel folder to change the position. Alternatively, you can interactively place the CAD model on surfaces of the platform CAD model using point-and-click operation. It is recommended after point-and-click placement that you fine-tune the position of the CAD model by directly editing its Position.



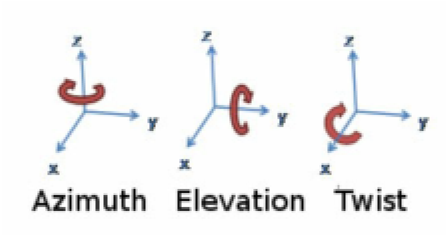
Orientation Mode: Select convention for specifying coordinate system orientation (order of rotations). The orientation of a node's coordinate system is uniquely specified by three left-handed rotation angles, one about each of the principal axes. EMIT supports two different conventions that differ by the order in which rotations are applied (both conventions are left-handed). When two or more rotations are required, one of these two conventions will often be more convenient than the other. In the following description, "reference coordinates" means one of two things, depending on the state of the **Show Relative Coordinates** toggle:

Show Relative Coordinates	**Reference Coordinates**
False (Off)	Parent-node coordinate system
True (On)	Placement coordinate system
(Not shown in panel)	Parent-node coordinate system

Roll-Pitch-Yaw: The order of rotations is Roll, then Pitch, then Yaw. Hence, Roll is always about the reference coordinates' x-axis, while Yaw is always about the current antenna coordinates' z-axis. The arrows represent the rotation of the local coordinate system relative to reference coordinates.



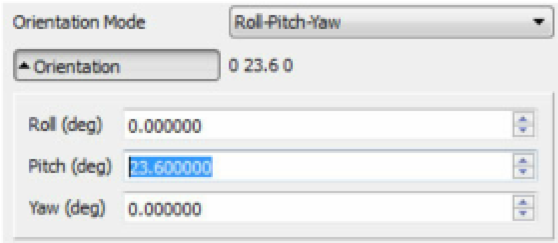
Azimuth-Elevation-Twist: The order of rotations is Azimuth, then Elevation, then Twist. Hence, Azimuth is always about the reference coordinates' z-axis, while Twist is always about the current antenna coordinates' x-axis. The arrows represent the rotation of the local coordinate system relative to reference coordinates.



Rotation Definitions

Name	**Axis**			**Sense**		**Order Applied**		
	+X	+Y	+Z	right-handed	left-handed	1st	2nd	3rd
Roll	•				•	•		
Pitch		•			•		•	
Yaw			•		•			•
Azimuth			•		•	•		
Elevation		•			•		•	
Twist	•				•			•

Orientation: Set the orientation of the CAD node coordinate system. The orientation of the node's coordinate system is uniquely defined by three rotation angles. These are either in Roll-Pitch-Yaw or Azimuth-Elevation-Twist convention, depending on the current selection of Orientation Mode.



Rendering: Expand the **Rendering** panel folder to modify how the CAD model is displayed. Note the Rendering panel folder is not visible if the model visibility is toggled off.

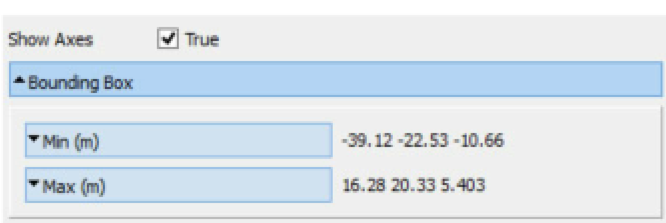
Render Mode: Set whether CAD model is rendered with flat-shading, wire-frame, wire frame with hidden line removal, or as an outline.

Color: Select the rule used for coloring surfaces of the CAD model.

Mesh Quality: Set the meshing resolution for display of curved-surface CAD models. This rendering control is only shown when a curved-surface CAD model is loaded (that is, an IGES model). Available choices are low, medium (default), and high. The mesh quality does not influence EMIT simulations, only how the CAD model is displayed. When the mesh quality is changed, EMIT re-tessellates the CAD model into flat polygons. Depending on the CAD model, this may cause a noticeable delay. A progress bar is shown for this process.

Show Axes: Toggle on/off display of the CAD model coordinate axes.

Bounding Box: Expand the Bounding Box panel folder to display the corner coordinates of the rectangular bounding box that encompasses the entire CAD model. Among other things, this is useful in verifying that the CAD model is scaled in meters. The corners are presented in the local (implicit) coordinate system of the CAD model, which is the local coordinate system of the CAD-Model node. The bounding box is the smallest rectangular box encompassing the CAD model with sides that are aligned with the three principal axes of its coordinate system.

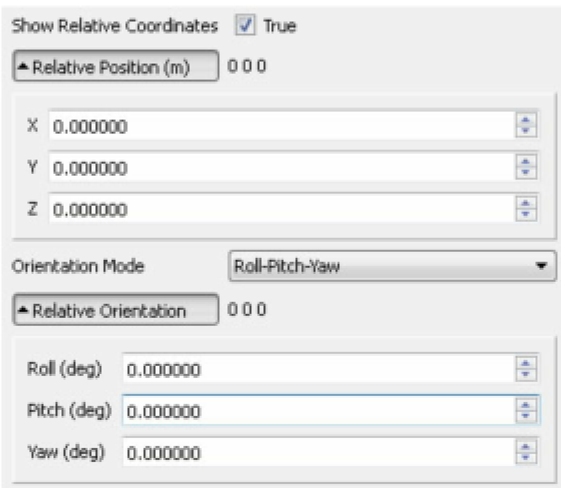


Notes: The Notes field provides a text area for you to describe the CAD model. The notes are stored with the project.

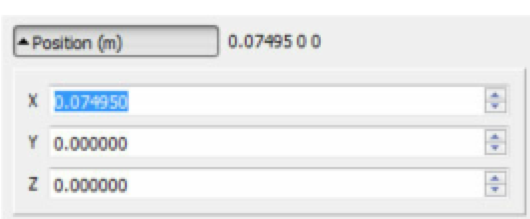
Groups

A Group is added to an EMIT design from the Coupling Editor. Each Group has the following properties defined in the Properties window.

Show Relative Coordinates: Toggle on/off display of Position and Orientation relative to placement coordinates instead of parent-node coordinates. Note that this setting is not available until after the Group has been interactively placed for the first time. When switched on, the *Position* and *Orientation* panel folders are renamed to *Relative Position* and *Relative Orientation*. They then indicate the Group position and orientation (that is, the Group node coordinate system) relative to the placement coordinates. When switched back off, these panel folders revert to their original labels and indicate the Group coordinate system relative to the coordinate system of its parent node.



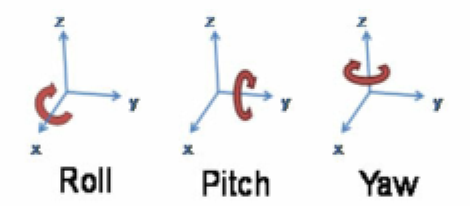
Position: Set the position (origin) of the Group node coordinate system. In its collapsed state, the Position panel folder shows the current location (in meters by default) of the Group in the scene with respect to the coordinate system of its parent node. Expand the Position panel folder to change the position. Alternatively, you can interactively place the Group on surfaces of the platform CAD model using point-and-click operation. It is recommended after point-and-click placement that you fine-tune the position of the Group by directly editing its Position.



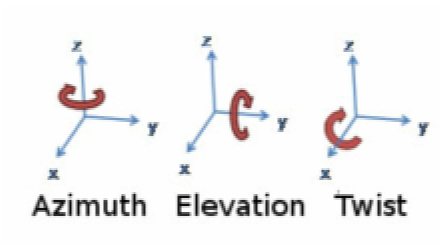
Orientation Mode: Select convention for specifying coordinate system orientation (order of rotations). The orientation of a node's coordinate system is uniquely specified by three left-handed rotation angles, one about each of the principal axes. EMIT supports two different conventions that differ by the order in which rotations are applied (both conventions are left-handed). When two or more rotations are required, one of these two conventions will often be more convenient than the other. In the following description, "reference coordinates" means one of two things, depending on the state of the Show Relative Coordinates toggle:

Show Relative Coordinates	Reference Coordinates
False (Off)	Parent-node coordinate system
True (On)	Placement coordinate system
(Not shown in panel)	Parent-node coordinate system

Roll-Pitch-Yaw: The order of rotations is Roll, then Pitch, then Yaw. Hence, Roll is always about the reference coordinates' x-axis, while Yaw is always about the current antenna coordinates' z-axis. The arrows represent the rotation of the local coordinate system relative to reference coordinates.



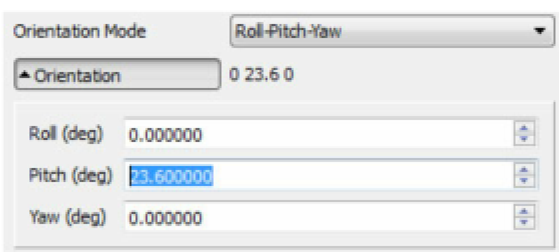
Azimuth-Elevation-Twist: The order of rotations is Azimuth, then Elevation, then Twist. Hence, Azimuth is always about the reference coordinates' z-axis, while Twist is always about the current antenna coordinates' x-axis. The arrows represent the rotation of the local coordinate system relative to reference coordinates.



Rotation Definitions

Name	Axis			Sense		Order Applied		
	+X	+Y	+Z	right-handed	left-handed	1st	2nd	3rd
Roll	•				•	•		
Pitch		•			•		•	
Yaw			•		•			•
Azimuth			•		•	•		
Elevation		•			•		•	
Twist	•				•			•

Orientation: Set the orientation of the Group node coordinate system. The orientation of the node's coordinate system is uniquely defined by three rotation angles. These are either in Roll-Pitch-Yaw or Azimuth-Elevation-Twist convention, depending on the current selection of Orientation Mode.



Show Axes: Toggle on/off display of the CAD model coordinate axes.

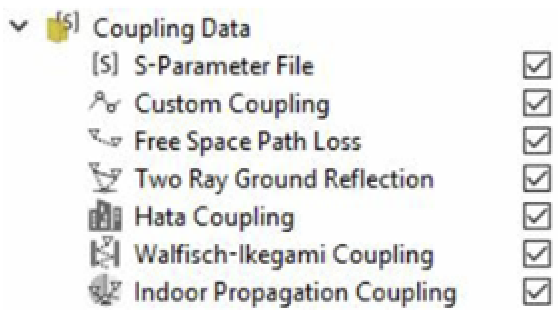
Box Color: Sets the color rendering properties of the group bounding box.

Notes: The Notes field provides a text area for you to describe the CAD model. The notes are stored with the project.

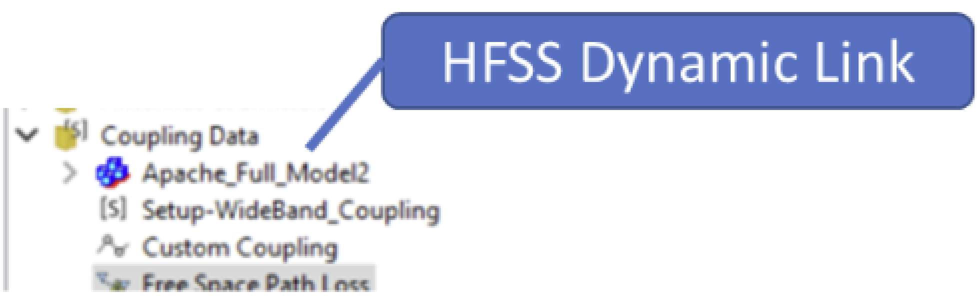
Coupling Data

All EMIT projects contain a single Coupling Data node. In EMIT, Coupling Data provides different models for taking into account antenna-to-antenna (ATA) coupling for cosite interference predictions. The Coupling Data node is a parent node whose children are the different coupling models available in the project.

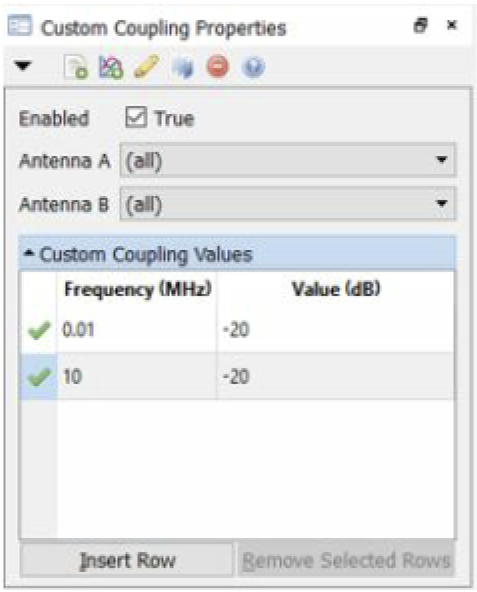
EMIT has seven different models available for treating antenna-to-antenna coupling as shown below. In order to use a model in a specific project, it has to be added to the project. The antennas in the project can then be assigned to one or more of the Coupling Models. If an antenna pair is assigned to multiple models, then EMIT will determine the coupling over a specific frequency range based on the priority of the models, which is determined by the order of the Coupling Model nodes in the tree under the Coupling Data node. The models are listed from lowest to highest priority and they can be re-arranged via drag-and-drop.

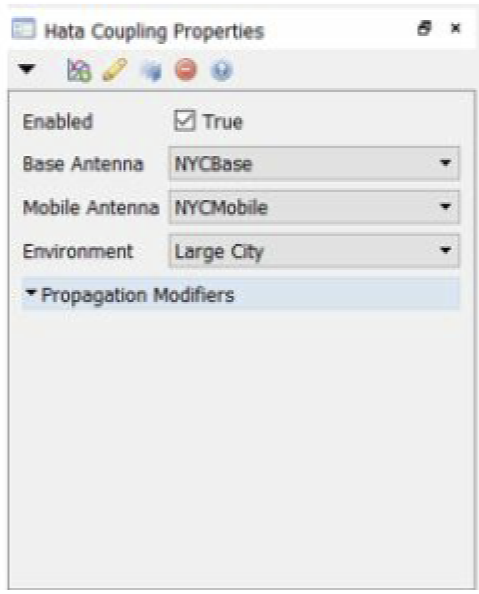


In addition to these models, coupling can also be defined via links to HFSS or HFSS 3D Layout models and these will also appear in the Coupling Data.

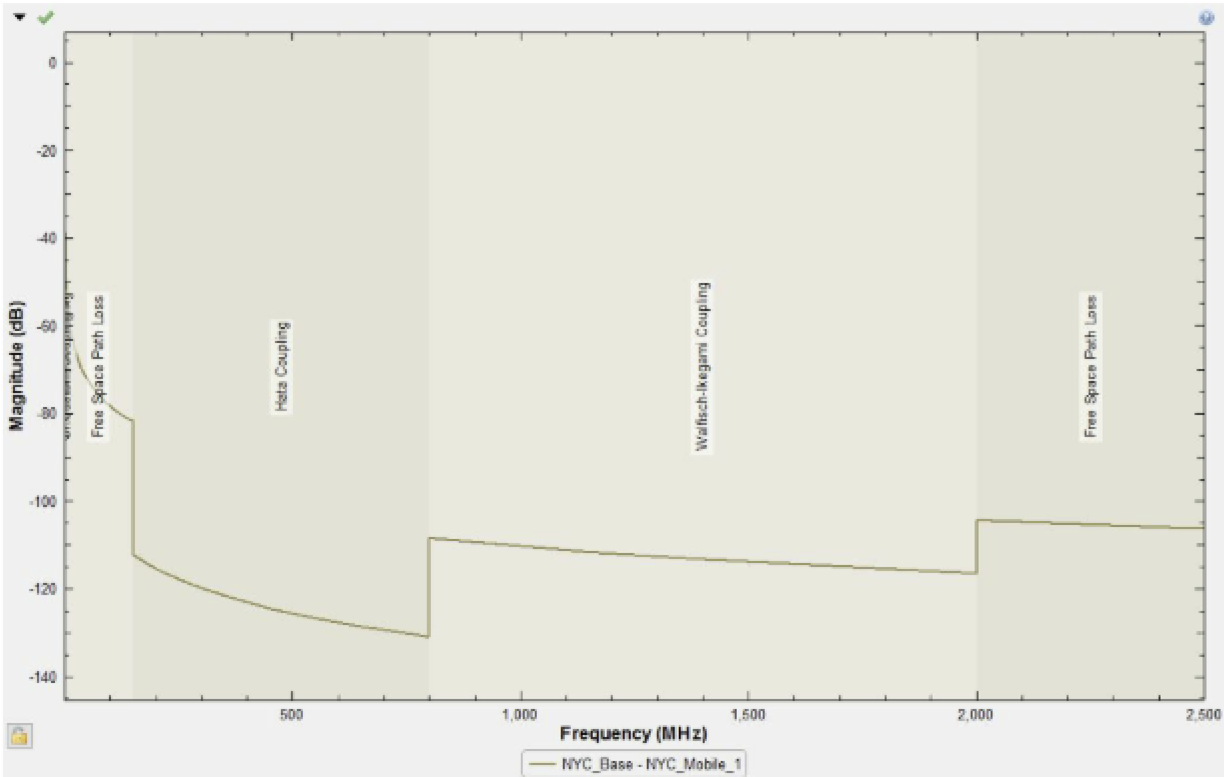


For each Coupling Model, you must specify which antenna pairs are applicable. Antenna Tags are extremely useful for this as they allow multiple antennas to be grouped together and then specify a Coupling Model for the entire group. The Hata Coupling property panel below illustrates how to apply the Hata Coupling Model to two groups of antennas tagged "NYCBase" and "NYCMobile" respectively. Similarly, the Custom Coupling model in the figure is applied to all of the antennas in the project. Note that since the Custom Coupling node is above the Hata Coupling node in the project tree (figure above), the Hata Coupling model has the higher priority and thus will be applied over the model's valid frequency range.

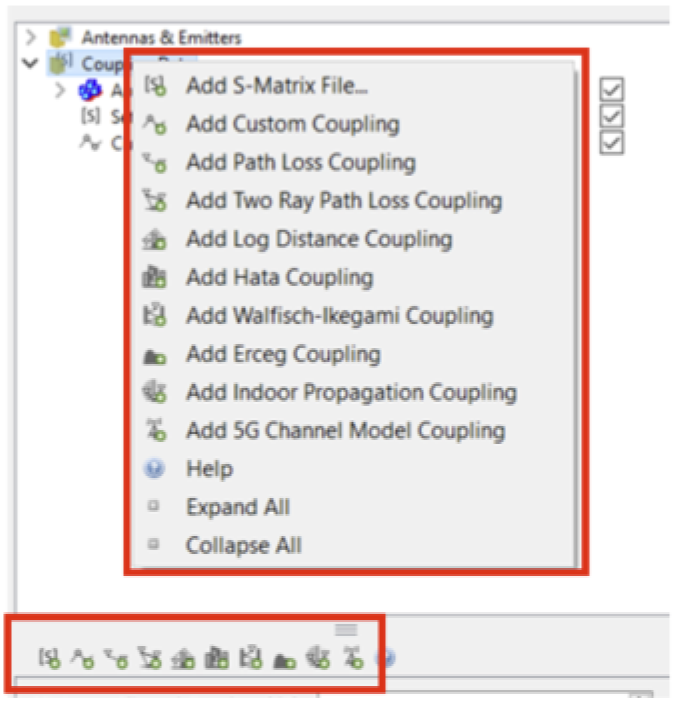




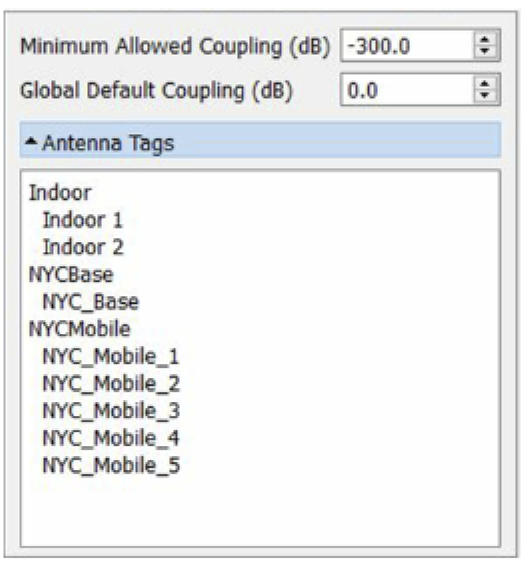
Partial Coupling: EMIT automatically combines all applicable Coupling Models for an antenna pair into a single broadband antenna-to-antenna coupling plot. Using the priorities defined above, the figure below shows the coupling between a NYC Mobile antenna and the NYC Base Station antenna. The antennas are not assigned to the Indoor Propagation model and thus the Indoor Propagation model is not used for any frequencies. The next highest priority coupling model is the Walfisch-Ikegami model which is used over the model's entire valid frequency range (800-2000 MHz). Next, the Hata Coupling model is applied over its valid frequency range (150-2000 MHz). Note that the valid range for the Hata model overlaps with that of the Walfisch-Ikegami model. Since the Walfisch-Ikegami model is specified to have a higher priority, it is used for the overlapping frequencies (800-2000 MHz) and the Hata model is used only over the non-overlapping portion of its range (150-800 MHz). Finally, the Free Space Path Loss model is used and since this is valid over all frequencies above the near field cutoff, it is applied over the remaining frequencies.



With the exception of HFSS links (which are created in the main AEDT project manager tree), new coupling models to be used are added via the Coupling Data node right-click menu or via the toolbar buttons in the Coupling Data properties pane. Details for each of the coupling models can be found here.



The Coupling Data properties are defined as follows:



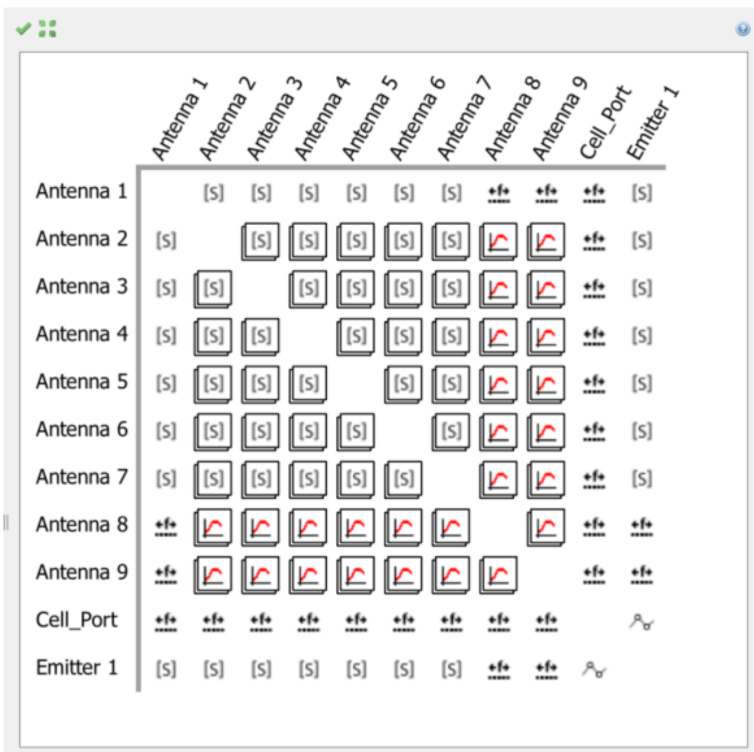
Minimum Allowed Coupling: Sets a lower limit for the antenna-to-antenna coupling computed for all antenna pairs in the project.

Global Default Coupling: Sets a default value for the antenna-to-antenna coupling. The Global Default Coupling value will be used if an antenna pair does not have any other coupling model specified at particular frequency (or range of frequencies).

Antenna Tags: Unique strings that allow multiple antennas to be grouped together. The Antenna Tags field in the Coupling Data node lists all of the tags that have been defined for the project. In the example above, the "Indoor" tag was assigned to two antennas ("Indoor 1" and "Indoor 2"), the "NYCBase" tag was assigned to a single antenna named "NYC_Base" and the "NYCMobile" tag was assigned to 5 antennas ("NYC_Mobile_1" through "NYC_Mobile_5").

Coupling Matrix

EMIT's Coupling Matrix window provides a compact way for viewing and manipulating coupling in an EMIT project. The Coupling Matrix presents the complete scene in a matrix format with all antennas/emitters listed both horizontally and vertically. The intersection of two antennas/emitters represents coupling between the antenna/emitter pair.

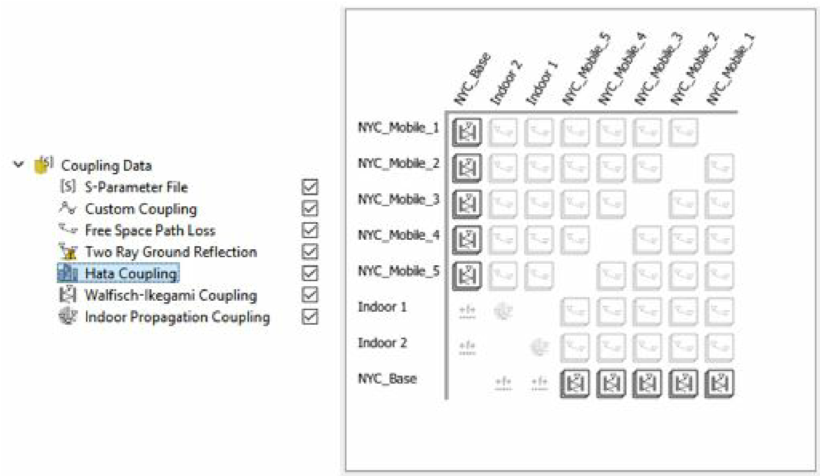


The icons in the matrix represent the type of coupling model that EMIT is using for the analysis of that antenna/emiter pair. If an antenna pair has more than one coupling type associated with it, then the icons will appear "stacked" with the icon for the highest priority coupling model shown. As noted in the Coupling Model section, the priority of the coupling models is determined by the order of the models under the Coupling Data node. The coupling models are shown with the lowest priority model at the top and increasing in priority as you go down the tree. The Coupling Model nodes can all be rearranged via drag-and-drop.

If an antenna/emitter pair has more than one coupling type associated with it, then the icons will appear "stacked" with the icon for the highest priority coupling model shown. As noted in the Coupling Model section, the priority of the coupling models is determined by the order of the models under the Coupling Data node. The coupling models are shown with the lowest priority model at the top and increasing in priority as you go down the tree. The Coupling Model nodes can all be rearranged via drag-and-drop.

Blank entries in the Coupling Matrix matrix represent self-interactions for which coupling is not computed.

The Coupling Matrix will update based on the selected Coupling Model node. For example, the figure below shows the Antenna Coupling Matrix when the Hata Coupling model is selected in the project tree. The coupling icons for the antennaemitter pairs that use the Hata Coupling model are highlighted, while the other coupling models are grayed out. Note that the coupling icon still shows the highest priority coupling model for each antenna pair and thus the highlighted icons still show the Walfisch-Ikegami Coupling model.



The following mouse actions are available in the Coupling Matrix window.

Scroll Wheel: Zoom in/out.

Shift-Left-Drag: In matrix area: Pans the matrix entries in the view window.

Click on Matrix Square: Plot the coupling for the antenna pair.

The following actions are accessible via the toolbar on the Antenna Coupling Matrix window.

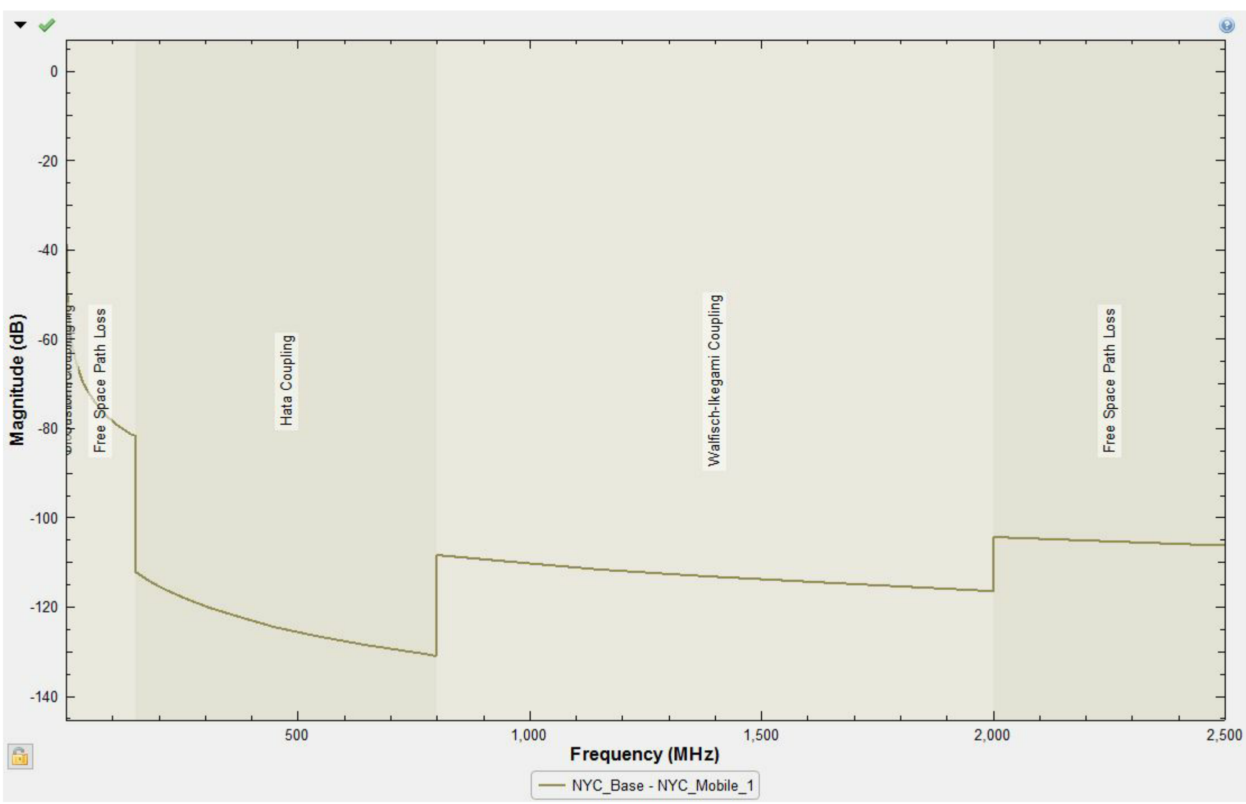
Reset View: Reset zoom and pan levels to default values to display entire Antenna Coupling Matrix.

Fit to Window: Sizes the current Antenna Coupling Matrix view to fill the entire

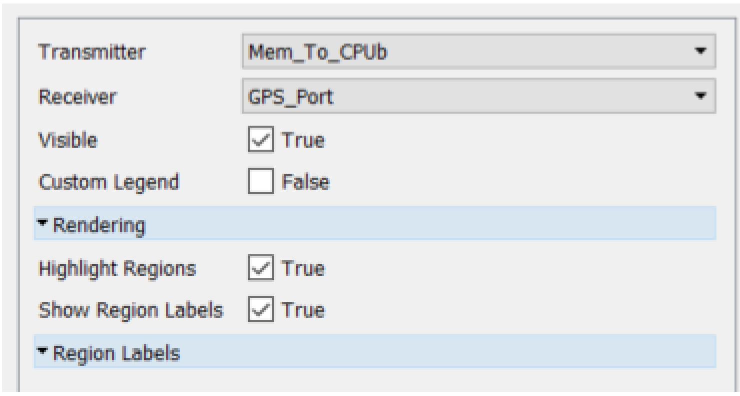
Coupling Plot

Clicking on a square in the Coupling Matrix results in a plot of the wideband coupling associated with the selected antenna/emitter pair being displayed in the Coupling Plot pane of the Coupling Editor. This is the composite coupling that EMIT assembles from all available coupling data & models and represents the coupling that will be used in the simulation for the selected antenna/emitter pairs.

Since the coupling for any antenna/emitter pairs can be derived from multiple sources depending on the frequency, the Coupling Plot shows what the source of the coupling data is for different segments of the wideband frequency range of interest. For example, in the coupling plot below the coupling data is assembled using a combination of Free Space Path Loss, Hata and Walfisch-Ikegami models. The frequency ranges over which the particular coupling model is used are identified in the plot with alternating color bands and text labels (the visibility of which can be toggled on/off).



Clicking on the trace label once at the bottom of the plot shows the Coupling Plot specific properties.



Transmitter/Receiver: Plots the coupling data associated with the specified Transmitter and Receiver antenna/emitter. Equivalent to selecting a square in the Coupling Matrix.

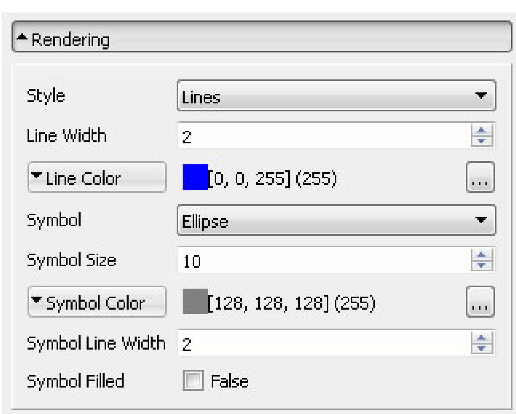
Visible: Toggles whether the trace is displayed in the Plot.

Custom Legend: This is a toggle panel folder. When switched on, Custom Legend becomes a panel folder. Expand this panel folder to specify the custom text of the trace in the plot legend. You can click the button to the right of the entry field to insert a variety of symbols (encoded as rich text). The plot legend appears at the bottom of the plot window.

> **Tip:** If you are having issues seeing the full legend in the plot window after changing a custom legend label, try adjusting the size of the plot window.

> **Tip:** HTML tags may be added to the legend text to control the font and formatting.

Rendering: The Rendering panel folder contains settings for configuring how the plot trace is drawn: line style, line width, line color, symbol type, symbol size, symbol color, symbol line width, and symbol fill. For plots with many traces, there many options for differentiating them.



Style: Specify the trace line style: None, Lines, Dotted, etc.

Line Width: Specify the line thickness.

Line Color: Set the line color. Expand this panel folder to set the red-green-blue (RGB) levels for the line color. Click the icon button to choose the color from a palette.

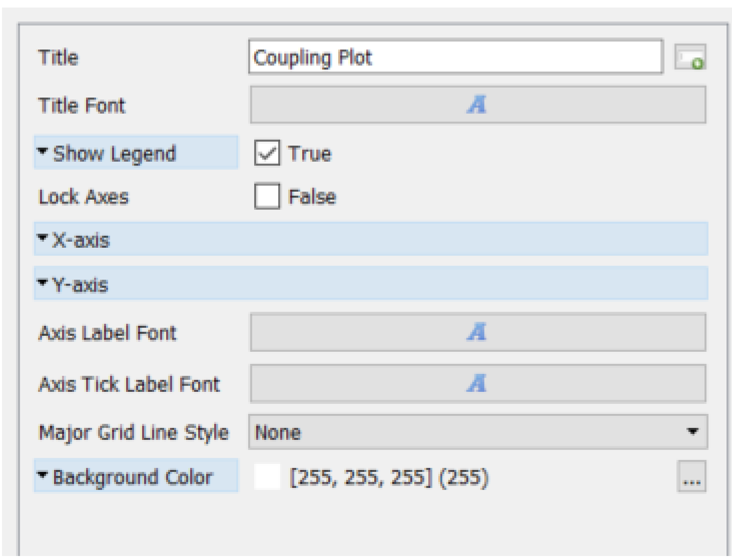
Symbol: Select a marker symbol to be placed at every sample point: NoSymbol (default), Ellipse, Rect, etc.

Highlight Regions: Toggles on/off the color bands identifying coupling regions in the plot.

Show Region Labels: Toggles on/off the text labels identifying coupling regions in the plot.

Region Labels: Provides options for the appearance of the text labels identifying coupling regions in the plot.

By clicking on the trace label twice at the bottom of the plot, the plots Properties will be displayed in the property pane of the Coupling Editor as shown below.



Title: Set the title text displayed at the top of the plot. While editing the Title, you can click the button to the right of the entry field to insert a variety of symbols (encoded as rich text) at the current cursor position.

Tip: If you clear the title text and the title displayed at the top of the plot area does not completely clear, enter a single space to make it explicitly blank.

Tip: HTML tags may be added to this text. Supported tags include <h1> through <h5> header tags, for bold, <i> for italics, <sup> and <sub> for superscript and subscript, and many others. The Title Font property is often easier to use, but HTML tags allow more flexibility, such as mixed font text, adding newlines via <br>, and changing colors as in

. Most HTML tags require an ending tag with the slash '/' character so that the modified text is encapsulated. For example,

Bold <i>bold-italic</i>^{superscript}

generates **Bold *bold-italic*** **superscript**

Title Font: Brings up a dialog to change the font for the title text displayed at the top of the plot.

Show Legend Toggle Panel Folder: Toggle on/off trace legend at the bottom of the plot. When switched on, Show Legend becomes a panel folder that can be expanded to show additional settings.

Legend Font: Brings up a dialog to change the font for the legend text

X-Axis: Expand the X-Axis panel folder to modify the horizontal axis extents and tick marks of the plot.

X-Axis Min, Max: Set the extents of the horizontal axis

Max Major Ticks (X): Control the number of major tick-mark intervals along the X-axis.The plot has both major and minor tick-marks. The major tick-marks are longer and numerically labeled. Max Major Ticks (X) sets the maximum number of major intervals along the horizontal axis, including partial intervals at the beginning or end of the axis extents. The plot axis management logic may choose a fewer number of intervals, but it will not exceed this limit.

Max Minor Ticks (X): Control the number of minor tick-mark intervals along the X-axis. The plot has both major and minor tick-marks. The minor tick-marks are shorter and not numerically labeled. Max Minor Ticks (X) sets the maximum number of minor tick-mark intervals between major ticks along the horizontal axis. The plot axis management logic may choose a fewer number of intervals, but it will not exceed this limit.

Log Scale: Set to true to use a logarithmic scale on the plot axis.

Y-Axis: Expand the Y-Axis panel folder to modify the horizontal axis extents and tick marks of the plot.

Y-Axis Min, Max: Set the extents of the horizontal axis

Max Major Ticks (Y): Control the number of major tick-mark intervals along the Y-axis.The plot has both major and minor tick-marks. The major tick-marks are longer and numerically labeled. Max Major Ticks (Y) sets the maximum number of major intervals along the horizontal axis, including partial intervals at the beginning or end of the axis extents. The plot axis management logic may choose a fewer number of intervals, but it will not exceed this limit.

Max Minor Ticks (Y): Control the number of minor tick-mark intervals along the Y-axis. The plot has both major and minor tick-marks. The minor tick-marks are shorter and not numerically labeled. Max Minor Ticks (Y) sets the maximum number of minor tick-mark intervals between major ticks along the horizontal axis. The plot axis management logic may choose a fewer number of intervals, but it will not exceed this limit.

Axis Label Font: Brings up a dialog to change the font for the x and y axis labels

Axis Tick Label Font: Brings up a dialog to change the font for the numbers next to the axis ticks

Major Grid Line Style: Set the line style for plot area grid lines extending from the major tick-marks. The default is None, meaning no major tick-mark grid lines.

Major Grid Color: Set the color used for major tick-mark grid lines

This setting becomes available when both major and minor grid lines are switched on.

Background Color: Set the background color of the Plot window. Expand this panel folder to set the red-green-blue (RGB) levels for the background color. Click the icon button to choose the color from a palette.

Coupling Models

The following coupling models are available:

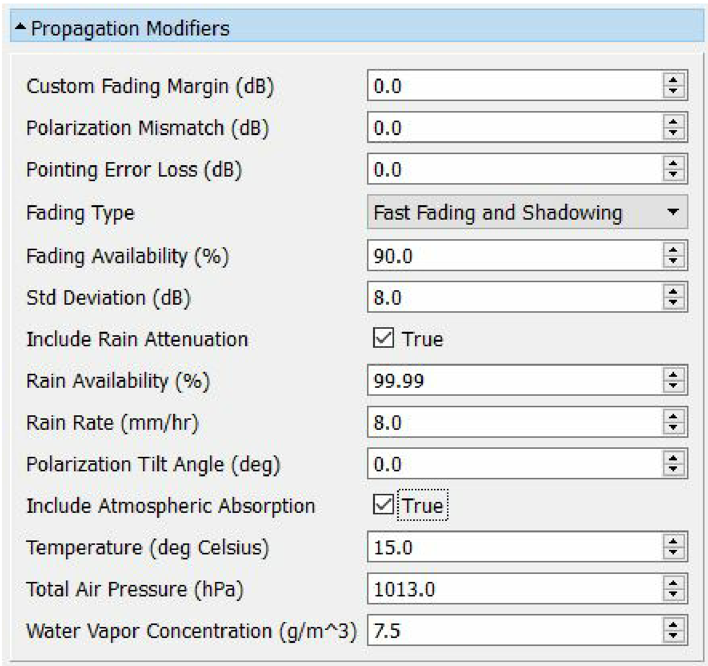
- General Parameters
- S-Parameter
- Custom
- Path Loss
- Two Ray Path Loss
- Log Distance
- Hata
- Walfisch-Ikegami
- Erceg
- Indoor Propagation
- 5G Channel Model

General Parameters

These are general parameters that can be added to any of the Path Loss coupling models in EMIT. This includes all Coupling Models **except** for the S-Parameter and Custom coupling models. By default, these parameters are not applied to the various coupling models.

Propagation Modifiers: Add margins to the path loss coupling model propagation loss to account for uncertainties in the propagation environment. Most of the modifiers are constant for all frequencies (i.e. they add the same amount of margin to the coupling at every frequency).

The exceptions are the Rain Attenuation and Atmospheric Absorption modifiers in which the amount of additional attenuation is a function of frequency.



Custom Fading Margin: Adds a specified margin to the path loss coupling to account for uncertainties in the propagation environment.

Polarization Mismatch: Adds a specified margin to the path loss coupling to account for losses due to a mismatch in the polarization of the Tx and Rx antennas.

Pointing Error Loss: Adds a specified margin to the path loss coupling to account for errors in aligning the main beams of the Tx and Rx antennas. Pointing error loss is more significant in highly directional antennas separated by a large distance.

- **Fading Type:** Specifies the fading environment for the coupling model. Options include:
- **Fast Fading:** Fading due to multipath and modeled as a Rayleigh distribution.
- **Shadowing:** Shadowing or "slow fading" is typically due to the motion of one or both terminals and is the result of variations in the terrain over time. This uncertainty is expressed as a random amount of shadowing loss and is modeled as a Gaussian random variable with zero mean and a specified standard deviation.

Fading Availability: The availability (or reliability) of a communications link. The higher the availability, the greater the margin needs to be.

Std Deviation: The standard deviation representing the shadowing loss.

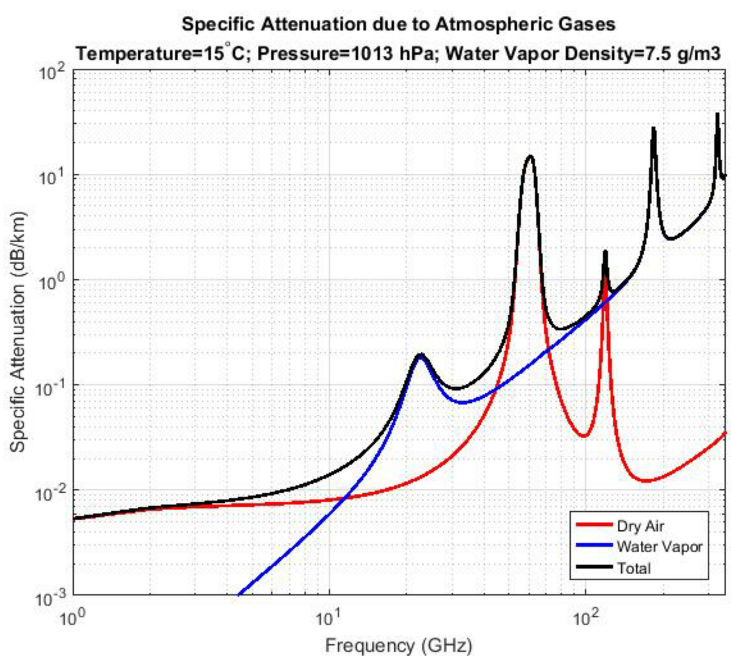
Include Rain Attenuation: If true, EMIT computes the additional attenuation due to rain for all frequencies. EMIT's rain attenuation model is based on the recommendations of ITU-R P838-3.

Rain Availability: The availability (reliability) of the communications link. The procedure specified in ITU-R P838-3 requires that the availability is in the range 99%-99.999%.

Rain Rate: The rain rate (mm/Hr) at the desired location. The rain rate specified should correspond to the location's rain rate that is exceeded 0.01% of the time. Rain rate values for most locations can be found in ITU-R P.837-6.

Polarization Tilt Angle: Polarization tilt angle relative to horizontal. For example, 45 degrees for circular polarization, 0 degrees for horizontal polarization, and 90 degrees for vertical polarization.

Include Atmospheric Absorption: If true, EMIT computes the additional attenuation due to atmospheric absorption for all frequencies. EMIT's atmospheric absorption model is based on the recommendations of ITU-R P676-11.



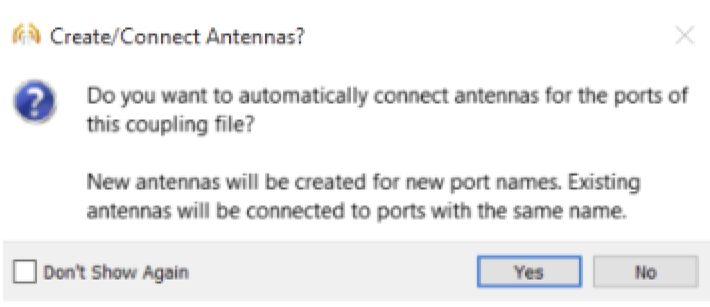
Temperature: The temperature (degrees Celsius) at the location of interest. If the antennas are at different locations, then an average value can be used.

Total Air Pressure: The total air pressure (hPa) at the location of interest. If the antennas are at different locations, then an average value can be used.

Water Vapor Concentration: The water vapor concentration (g/m3) at the location of interest. If the antennas are at different locations, then an average value can be used.

S-Parameter

The S-Parameter coupling model in EMIT uses an N-port scattering matrix for antenna coupling. The S-matrix is obtained from a Touchstone File supplied by the user. To add an S-matrix to the project, right-click on the Couplings node and select Add S-Matrix File. EMIT shows the following prompt:

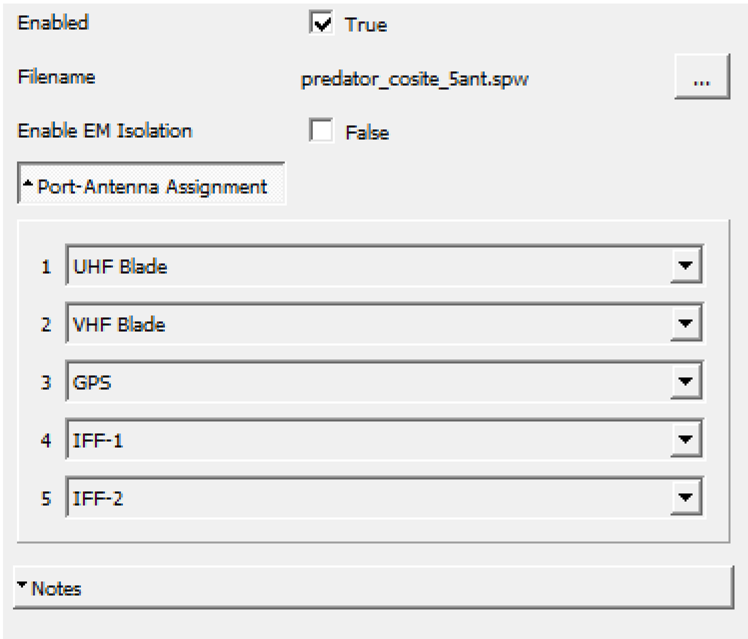


Select **Yes** to automatically create N antennas in the scene and associate them with the N ports of the imported S-matrix. Move the antennas to their intended position using the individual antennas' property panels. If you select **No**, EMIT does not create any antennas, allowing you to associate the S-matrix with antennas that already exist in the scene or create new antennas manually.

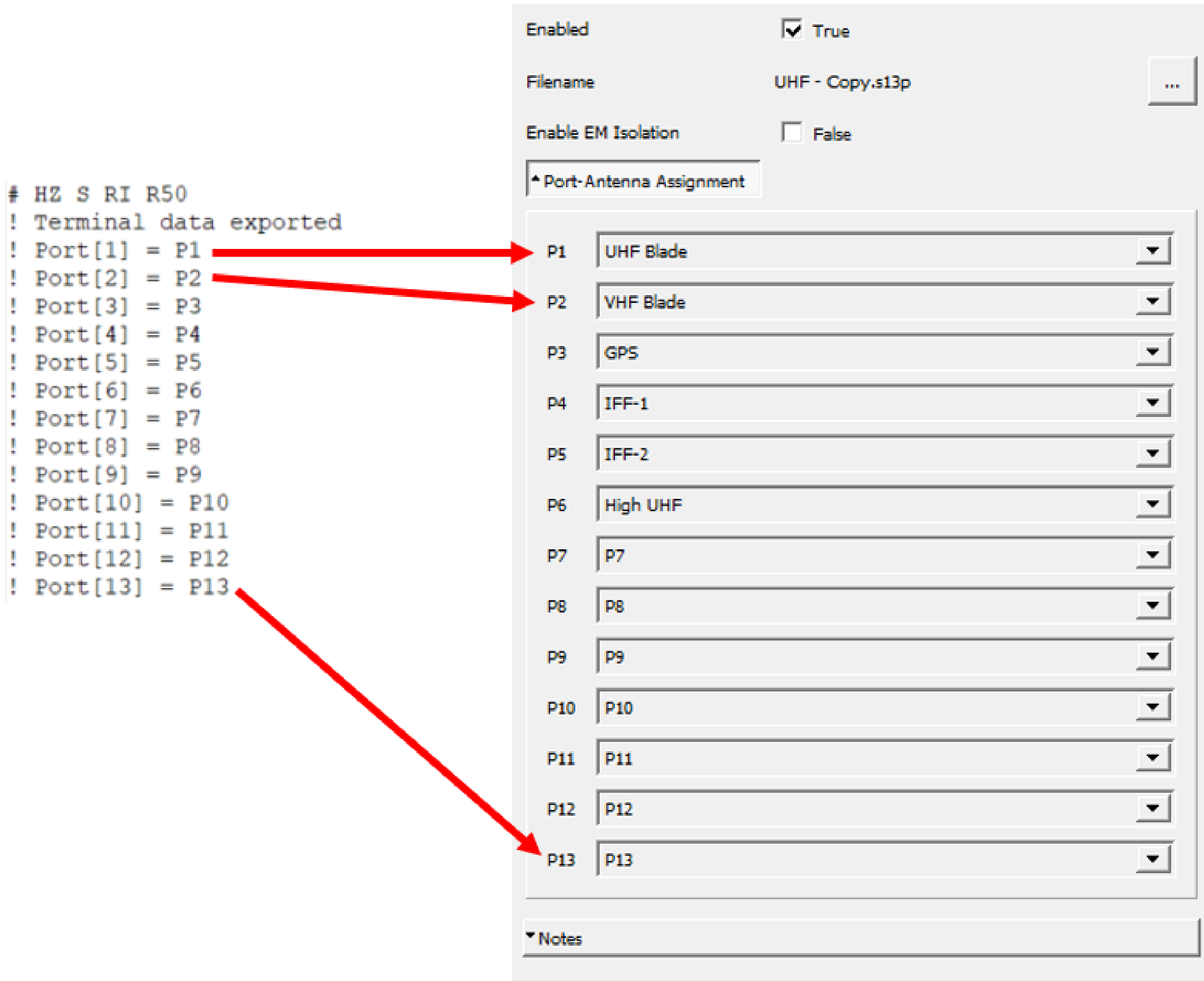
After reading in the Touchstone file, EMIT creates an S-Parameter node using the Touchstone filename (without extension) as the node name. Since EMIT implicitly assumes a 50-Ohm impedance for all RF Systems, all Touchstone files are normalized to a 50-Ohm reference impedance. Multiple Touchstone files can be imported into an EMIT project, each resulting in a separate S-Parameter coupling node.

Imported S Parameter files are considered the highest fidelity coupling model in EMIT and are automatically used by EMIT when they are available for a specified antenna pair. EMIT supports N-port S parameter files, and each file can be toggled on/off within EMIT.

The following options are available from the S-parameter property panel:



EMIT will also use optionally defined port labels. The port labels must be defined in the comment lines of the Touchstone file as shown below. The comment lines must start with a "!" and the ports are then specified by "Port[X] = [port label]" where *X* is an integer specifying the port number and [port label] is a string label. The labels can also be used as the antenna names (P7-P13 below) or they can just be used for the port label (P1-P6 below).



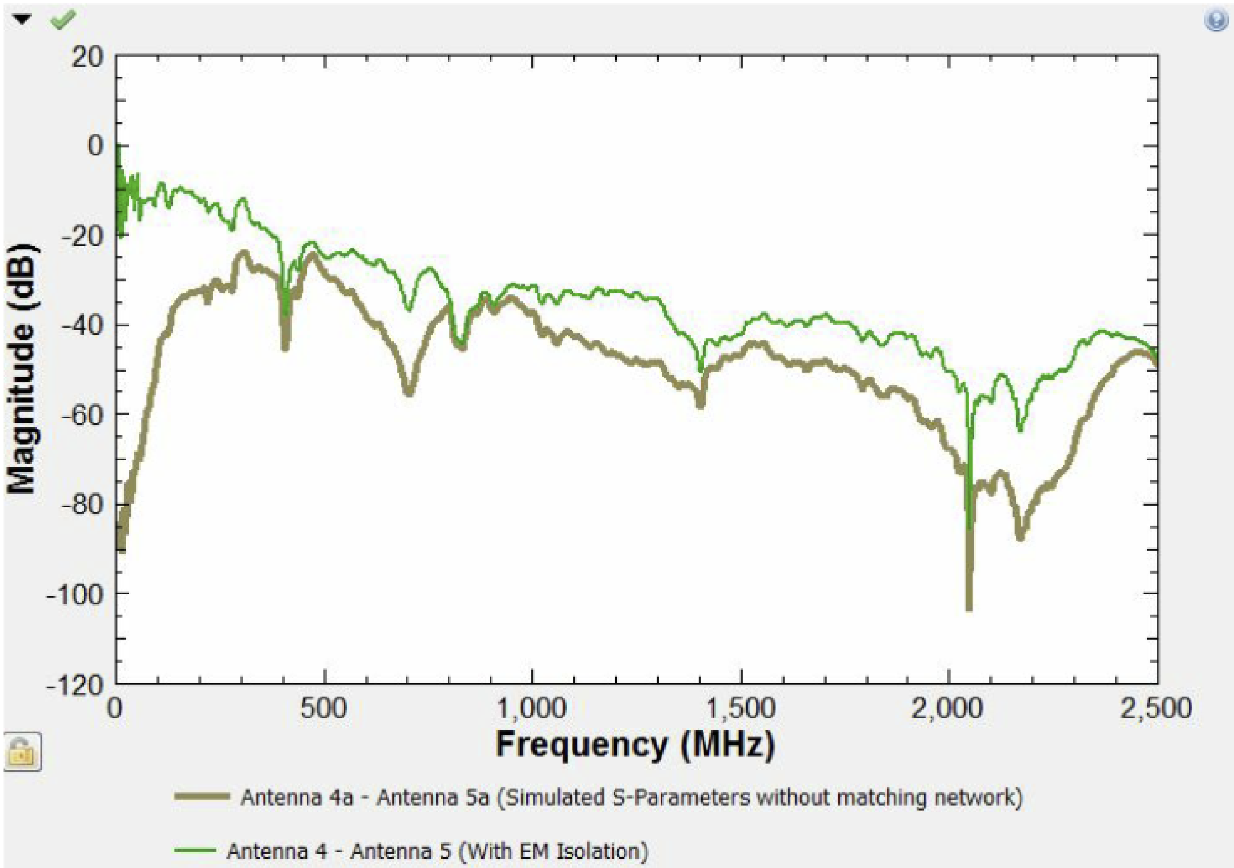
Enabled: Toggle on/off the use of an individual S parameter coupling file. If disabled, EMIT will not use the S parameter file for any simulations.

Filename: The name of the Touchstone file containing the S-parameter coupling data.

Enable EM Isolation: When enabled, EMIT computes the worst-case coupling, that is, “EM Isolation,” for imported coupling data to account for matching networks. The resulting coupling is computed assuming a conjugate match for each antenna pair at each frequency. This provides a worst-case coupling estimate in cases where the S-parameter data may not include the effects of matching networks. (ref. J. Rahola , Bandwidth potential and electromagnetic isolation: Tools for analyzing the impedance behavior of antenna systems, Proceedings of the EuCAP 2009 conference, Berlin, March 23-27, 2009).

The figure below shows a comparison between simulated S-parameter coupling data before and after EM Isolation is applied. Note that for all frequencies, the coupling computed via EM

Isolation is equal to or worse than the initial data which is expected since it assumes a perfect antenna match at each frequency.



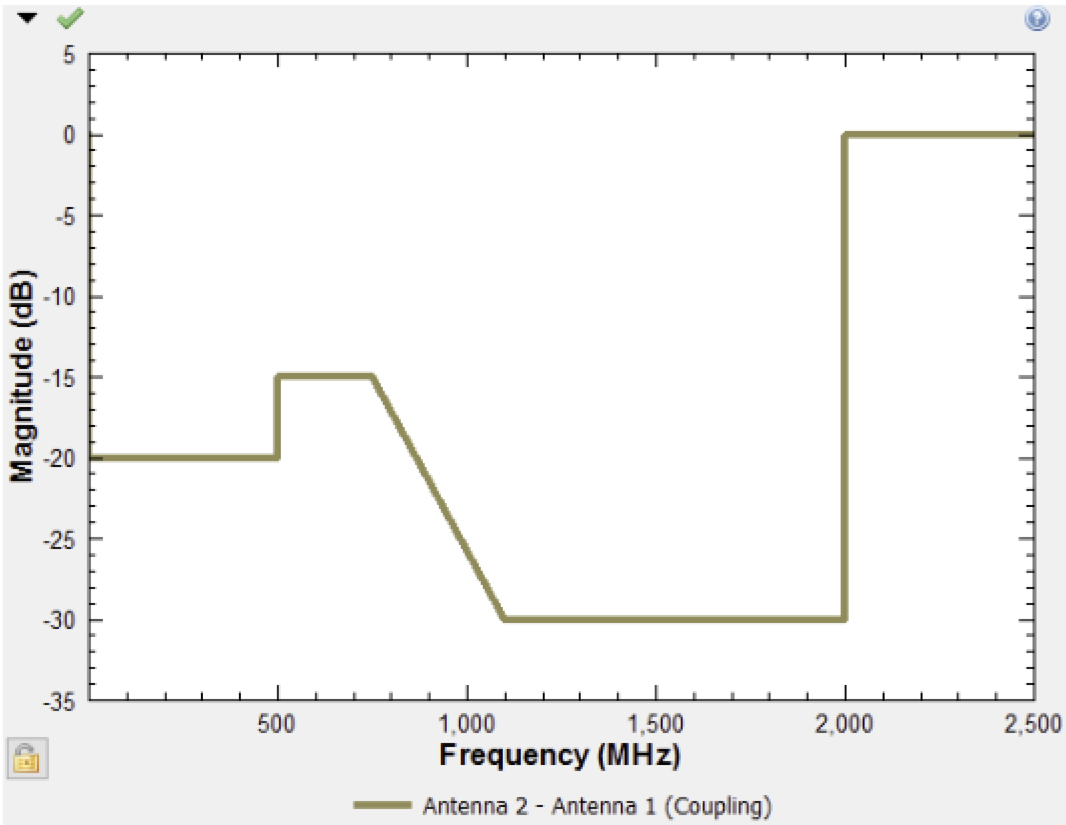
Ports: The N ports of the S-matrix need to be 'connected' to the antennas with which they are associated in the EMIT scene. Under the Ports folder in the S-Parameter property panel, a list of ports will appear as shown in the figure above (the figure above is for a 5-port S-matrix). A pull-down menu next to each port number will allow an antenna in the scene to be 'connected' to the corresponding port of the S-matrix. If EMIT created antennas for each port when the Touchstone file was imported then these associations will have been made automatically. Otherwise, the pull-down menus should be used to make the associations.

It is not required that all ports be associated with an antenna. For example, a valid EMIT scenario may only need to use a subset of the N-port S-matrix to define the coupling between antennas.

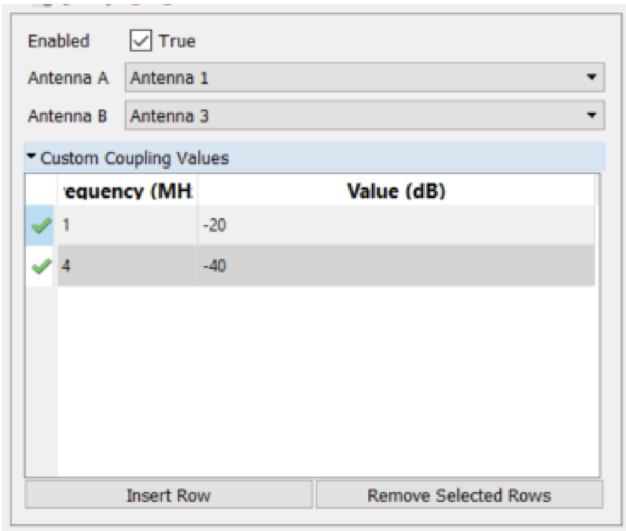
Notes: The Notes field provides a text area for the user to enter a description of the Scattering Matrix file. The notes are stored with the project.

Custom Coupling

Custom Coupling allows multiple constant coupling values to be defined over unique ranges for an antenna pair. This is done on a piece-wise linear nature, with a "fixed value" for the coupling defined at each frequency of interest. In this way, users can quickly approximate the in-band and out-of-band regions for the antennas and perform a rough cosite analysis prior to obtaining any detailed antenna-to-antenna coupling data.



There is no limit to the number of Custom Coupling nodes that can be specified for a project. This provides a lot of flexibility in terms of how each Custom Coupling node is defined since each Custom Coupling node can then be assigned a different priority. For example, two Custom Coupling nodes and a Path Loss Coupling node can be defined for a project with the Custom Coupling nodes specifying the coupling at "low" and "high" frequencies where the Path Loss Coupling model may be less accurate for the given application. Additionally, antenna tags can be used to group one or more antennas so that a single Custom Coupling node can be used across multiple antenna-to-antenna interactions. The following options are available from the Custom Coupling property panel:



Enabled: Toggle on/off the use of the Custom Coupling node. If disabled, EMIT will not use this Custom Coupling node for any simulations.

Antenna A: Specify the first antenna in the Custom Coupling node's antenna pair. Note that "(all)" can be used to assign all of the project's antennas to the coupling model and that antenna tags can also be used to specify a group of antennas. If Antenna A = 'Group A' and Antenna B = 'Group B', then all combinations of antennas from Group A to Group B or from Group B to Group A will use the coupling model

Antenna B: Specify the second antenna in the Custom Coupling node's antenna pair. Note that "(all)" can be used to assign all of the project's antennas to the coupling model and that antenna tags can also be used to specify a group of antennas. If Antenna A = 'Group A' and Antenna B = 'Group B', then all combinations of antennas from Group A to Group B or from Group B to Group A will use the coupling model.

Custom Coupling Values: The Custom Coupling Values table allows users to specify the coupling in a piece-wise linear nature. Each Custom Coupling node allows the users to specify the coupling at any number of frequency points. EMIT then linearly interpolates between the specified frequency/attenuation pairs to create a wide-band coupling trace over the node's frequency range (i.e. from the minimum frequency specified in the node to the highest frequency specified).

Path Loss Coupling

The Path Loss coupling model computes the coupling between antennas based on the distance between the antennas according to the following equation:

$$Path\ Loss\ (dB) = 20log_{10}\left(\frac{4\pi d}{\lambda}\right)$$

where

$$d = distance\ between\ antennas$$
$$\lambda = free\ space\ wavelength$$

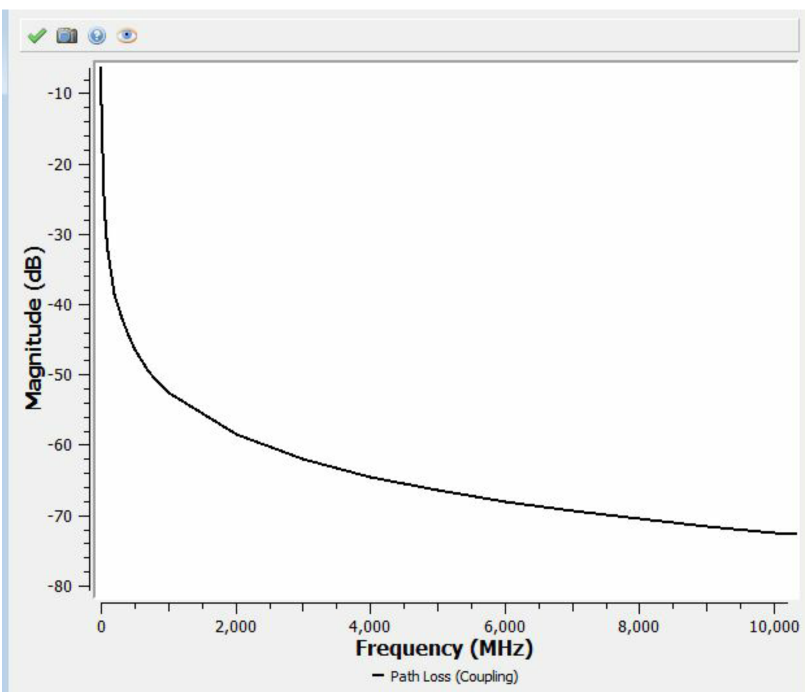
The antenna locations are defined via the property panel for each antenna. The figure below shows the path loss as a function of frequency for two antennas separated by a distance of 10 m.

The Path Loss model is only valid if the antenna's position has been defined and for frequencies for which the antennas are in the far field. These frequencies are defined using the common far field/near field boundary:

$$Far\ Field\ Freq = \frac{c}{2\pi d}$$

$$c = speed\ of\ light\ [^m/_s]$$

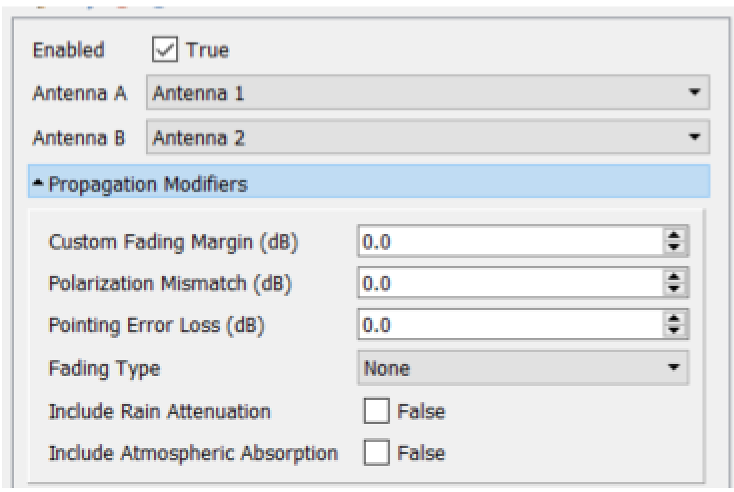
$$d = distance\ between\ antennas$$



The Path Loss model will also include the antenna gains along the line-of-sight (LOS) path between the two antennas. Note that EMIT does not account for any physical objects in a scene and thus will compute the LOS gain between two antennas even if they are physically blocked (e.g. opposite sides of a fuselage) if the Path Loss Coupling model is used. In scenarios such as these, it is recommended to use a full wave (HFSS) or asymptotic solver (SBR+) to obtain the coupling and then to import that data into EMIT as a S-matrix coupling file.

The Path Loss model also contains (optional) Propagation Modifiers. These enable margins (e.g. due to rain attenuation, atmospheric absorption, shadowing, etc) to be added to the "base" value of coupling computed by the coupling model.

The following options are available from the **Path Loss Coupling** property panel:



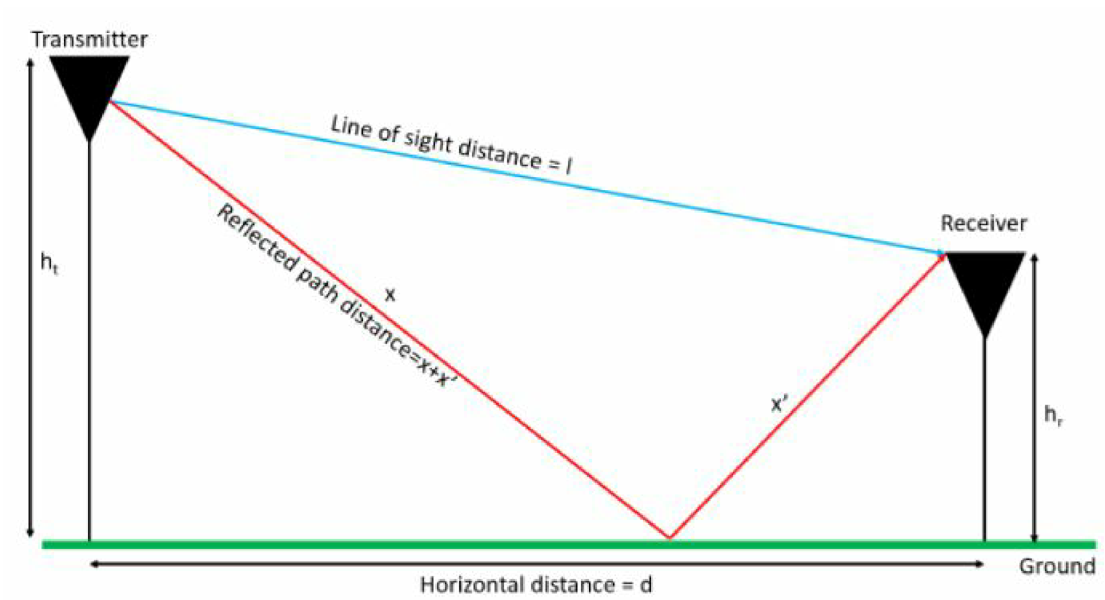
Enabled: Toggle on/off the use of the Path Loss Coupling node. If disabled, EMIT will not use this Path Loss Coupling node for any simulations.

Antenna A: Specify the first antenna in the Path Loss Coupling node's antenna pair. Note that "(all)" can be used to assign all of the project's antennas to the coupling model and that antenna tags can also be used to specify a group of antennas. If Antenna A = 'Group A' and Antenna B = 'Group B', then all combinations of antennas from Group A to Group B will use the coupling model.

Antenna B: Specify the second antenna in the Path Loss Coupling node's antenna pair. Note that "(all)" can be used to assign all of the project's antennas to the coupling model and that antenna tags can also be used to specify a group of antennas. If Antenna A = 'Group A' and Antenna B = 'Group B', then all combinations of antennas from Group A to Group B will use the coupling model.

Two Ray Path Loss Coupling

The Two Ray Path Loss coupling model assumes that there are two dominant signal paths between the Tx and Rx antennas. These are the line-of-sight (LOS) path and a reflected path as shown in the figure below.



The two signals combine, constructively and destructively, at the receive antenna resulting in peaks and nulls in the received power. The coupling between the antennas is a function of the frequency and the distance between the antennas and can be computed according to the following equation:

$$L = \left[\frac{\lambda}{4\pi}\right]^2 \left| \frac{\sqrt{G_l}}{l} + \frac{R\sqrt{G_r}e^{-j\Delta\varphi}}{x+x'} \right|^2$$

$G_l = G_a G_b$ in the LOS direction

$G_r = G_c G_d$ in the reflected direction

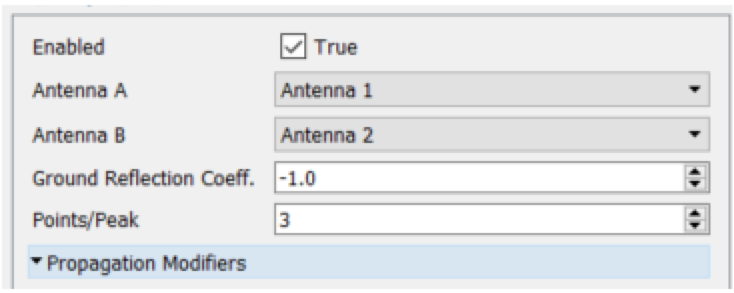
R is the ground reflection coefficient

$\Delta\varphi = 2\pi\left(\frac{x+x'-l}{\lambda}\right)$ is the phase difference between the 2 signals

$$x + x' - l = \sqrt{(h_t + h_r)^2 + d^2} - \sqrt{(h_t - h_r)^2 + d^2}$$

The Two Ray Path Loss coupling model is typically used to model propagation environments with minimal multipath fading and a large separation between the antennas. These conditions result in the two dominant propagation paths shown above.

The following options are available from the Two Ray Path Loss Coupling property panel:



Enabled: Toggle on/off the use of the Two Ray Path Loss Coupling node. If disabled, EMIT does not use this Two Ray Path Loss Coupling node for any simulations.

Antenna A: Specify the first antenna in the Two Ray Path Loss Coupling node's antenna pair. Note that "(all)" can be used to assign all of the project's antennas to the coupling model and that antenna tags can also be used to specify a group of antennas. If Antenna A = 'Group A' and Antenna B = 'Group B', then all combinations of antennas from Group A to Group B will use the coupling model.

Antenna B: Specify the second antenna in the Two Ray Path Loss Coupling node's antenna pair. Note that "(all)" can be used to assign all of the project's antennas to the coupling model and that antenna tags can also be used to specify a group of antennas. If Antenna A = 'Group A' and Antenna B = 'Group B', then all combinations of antennas from Group A to Group B will use the coupling model.

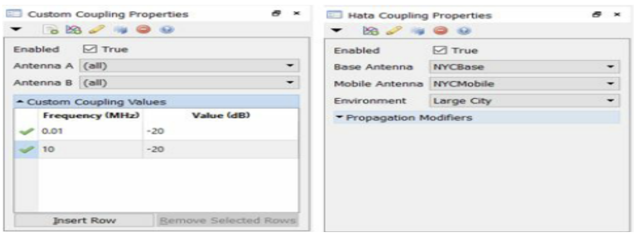
Path Loss Coefficient: The Path Loss coefficient is a function of the material properties of the reflecting surface. For small angles, the reflection coefficient is -1 as there is a 180 degree phase change on reflection.

Points/Peak: The two ray Path Loss coupling model has many peaks and nulls due to the constructive and destructive interference resulting from the reflected signal. The Points/Peak parameter specifies the number of frequency points used to model the peaks in the coupling. Increasing the number of points will improve the accuracy, but may also increase the simulation time.

Hata Coupling

The Hata Coupling model (with the COST 231 extension) is an empirical model developed to estimate the propagation loss in various complex environments. As an empirical model, it is derived from measurements at different frequencies across multiple sites. The Hata Coupling model is particularly popular for estimating median path loss in macrocellular systems.

The Hata Coupling model provides an estimate of the median path loss as a function of carrier frequency, base station and mobile station antenna heights, and the distance between the base station and mobile station. The Hata Coupling model is applicable over the following range of parameters:



The Hata coupling model defines the median path loss in an urban environment by:

$$PL_{urban}(dB) = 69.55 + 26.16 log_{10} f - 13.82 log_{10} h_b + \left(44.9 - 6.55 log_{10} h_b\right) log_{10} d - a(h_m)$$

$a(h_m)$: mobile station antenna correction factor

For a large city with dense building clutter and narrow streets, the mobile station antenna correction factor is:

$$a(h_m) = 8.29\left[log_{10}(1.54 h_m)\right]^2 - 1.1 \qquad f \leq 200\ MHz$$

$$a(h_m) = 3.20\left[log_{10}(11.75 h_m)\right]^2 - 4.97 \qquad f > 200\ MHz$$

For a small or medium size city, where the building clutter density is smaller, the mobile station antenna correction factor is:

$$a(h_m) = \left(1.11 log_{10} f - 0.7\right) h_m - \left(1.56 log_{10} f - 0.8\right)$$

For a suburban area, the same mobile station antenna correction factor used for small/medium cities is applicable, but the median path loss is modified to be:

$$PL_{suburban}(dB) = PL_{urban} - 2\left[log_{10}\left(\frac{f}{28}\right)\right]^2 - 5.4$$

For a rural area, the same mobile station antenna correction factor used for small/medium cities is also applicable, but the median path loss is modified to be:

$$PL_{rural}(dB) = PL_{urban} - 4.78\left[log_{10}f\right]^2 + 18.33log_{10}f - 40.98$$

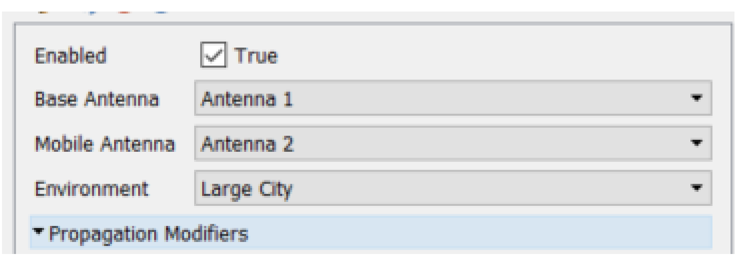
The European Cooperation in the field of Scientific and Research (COST) group extended the original model developed by Hata to also cover propagation environments in the 1500-2000 MHz band. This was necessary due to the popularity of cellular PCS deployments and subsequent cellular enhancements. The other model parameters listed above are still applicable to the COST extension. The median path loss for the COST-231 Hata model is:

$$PL\,(dB) = 46.3 + 33.9log_{10}f - 13.82log_{10}h_b + \left(44.9 - 6.55log_{10}h_b\right)log_{10}d - a\left(h_m\right) + C_f$$

$$a\left(h_m\right) = \left(1.11log_{10}f - 0.7\right)h_m - \left(1.56log_{10}f - 0.8\right)$$

For metropolitan centers the correction factor Cf=3 dB and for small/medium sized cities and suburban centers with medium tree density the correction factor Cf=0 dB. This model is also recommended by the WiMAX Forum for system simulations and network planning of macrocellular systems in both urban and suburban areas for mobility applications.

The following options are available from the Hata Coupling property panel:



Enabled: Toggle on/off the use of the Hata Coupling node. If disabled, EMIT will not use this Hata Coupling node for any simulations.

Base Antenna: Specify the base station antenna(s) in the Hata Coupling node's antenna pair. Note that "(all)" can be used to assign all of the project's antennas to the coupling model and that antenna tags can also be used to specify a group of antennas. If Base Antenna = 'Group A' and

Mobile Antenna = 'Group B', then all combinations of antennas from Group A to Group B will use the coupling model.

Mobile Antenna: Specify the mobile antenna(s) in the Hata Coupling node's antenna pair. Note that "(all)" can be used to assign all of the project's antennas to the coupling model and that antenna tags can also be used to specify a group of antennas. If Base Antenna = 'Group A' and Mobile Antenna = 'Group B', then all combinations of antennas from Group A to Group B will use the coupling model.

Environment: Specifies the propagation environment where the antennas are located.

Walfisch-Ikegami Coupling:

The Walfisch-Ikegami Coupling model is an empirical model developed to estimate the propagation loss in various complex environments. As an empirical model, it is derived from measurements at different frequencies across multiple sites. While the Hata Coupling model is suitable for macrocellular environments, it is not recommended for smaller cells with radii less than 1 km. The Walfisch-Ikegami Coupling model, on the other hand, is recommended for modeling these microcellular environments. The model assumes an urban environment with several characteristics of the environment specified as inputs. Diffraction is assumed to be the main mode of propagation and the model is valid over the following range of parameters:

- 800 MHz $\leq$ frequency $\leq$ 2000 MHz
- 4 m $\leq h_b \leq$ 50 m (base station antenna height)
- 1 m $\leq h_m \leq$ 3 m (mobile station antenna height)
- 0.2 km $\leq$ distance $\leq$ 5 km

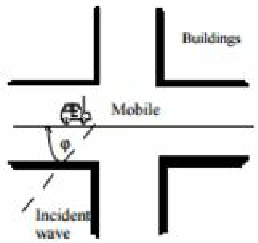
The Walfisch-Ikegami model distinguishes between the non-line-of-sight (NLOS) and the LOS cases. For the NLOS case, the loss is composed of the terms free space loss, Lfs, multiple screen diffraction loss, Lmsd, and roof top to street diffraction and scatter loss Lrts:

$$PL_{NLOS} = L_{fs} + L_{rts} + L_{msd} \quad for\ L_{rts} + L_{msd} > 0$$

$$PL_{NLOS} = L_{fs} \quad for\ L_{rts} + L_{msd} \leq 0$$

$$L_{fs}(dB) = 32.4 + 20log_{10}\left(\frac{d}{km}\right) + 20log_{10}\left(\frac{f}{MHz}\right)$$

The roof top to street diffraction and scatter loss term describes the coupling of the wave propagating along the multiple screen path into the street where the mobile station is located. In COST 231, the following street orientation was developed and the Walfisch-Ikegami equations for Lrts are specified:



$$L_{rts} = -16.9 - 10log_{10}\frac{w}{m} + 10log_{10}\frac{f}{MHz} + 20log_{10}\frac{\Delta h_m}{m} + L_{Ori}$$

$$L_{Ori} = -10 + 0.354\frac{\varphi}{deg} \qquad for\ 0^\circ \le \varphi < 35^\circ$$

$$L_{Ori} = 2.5 + 0.075\left(\frac{\varphi}{deg} - 35\right) \quad for\ 35^\circ \le \varphi < 55^\circ$$

$$L_{Ori} = 4.0 - 0.114\left(\frac{\varphi}{deg} - 55\right) \quad for\ 55^\circ \le \varphi < 90^\circ$$

$$\Delta h_m = h_{roof} - h_m$$

$$\Delta h_b = h_b - h_{roof}$$

The heights of buildings and their spatial separations along the direct radio path are modelled by absorbing screens for the determination of Lmsd:

$$L_{msd} = L_{bsh} + k_a + k_d log_{10} \frac{d}{km} + k_f log_{10} \frac{f}{MHz} - 9 log_{10} \frac{b}{m}$$

$$L_{bsh} = -18 log_{10} \left(1 + \frac{\Delta h_b}{m}\right) \quad for\ h_b > h_{roof}$$

$$L_{bsh} = 0 \quad for\ h_b \le h_{roof}$$

$$k_a = 54 \quad for\ h_b > h_{roof}$$

$$k_a = 54 - 0.8 \frac{\Delta h_b}{m} \quad for\ d \ge 0.5\ km\ and\ h_b \le h_{roof}$$

$$k_a = 54 - 0.8 \frac{\Delta h_b}{m} \frac{d}{0.5 km} \quad for\ d < 0.5\ km\ and\ h_b \le h_{roof}$$

$$k_d = 18 \quad for\ h_b > h_{roof}$$

$$k_a = 18 - 15 \frac{\Delta h_b}{h_{roof}} \quad for\ h_b \le h_{roof}$$

$$k_f = -4 + 0.7 \left(\frac{\frac{f}{MHz}}{925} - 1\right) \quad for\ medium\ sized\ city\ and\ suburban\ centers\ with\ medium\ tree\ density$$

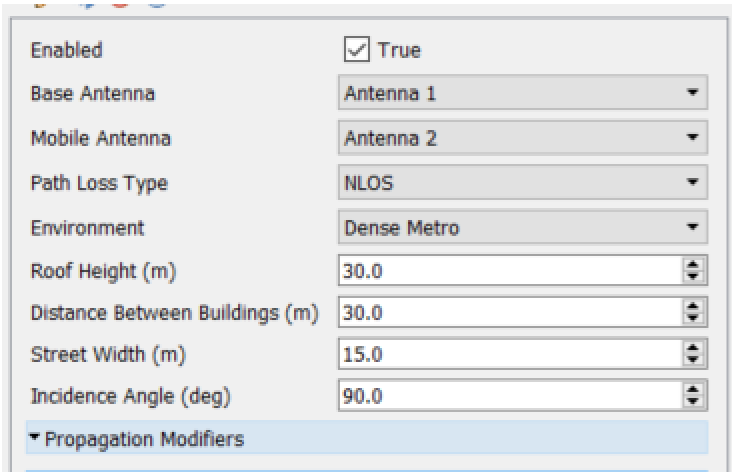
$$k_f = -4 + 1.5 \left(\frac{\frac{f}{MHz}}{925} - 1\right) \quad for\ metropolitan\ centers\ (dense\ urban)$$

Where the term ka represents the increase of the path loss for base station antennas below the roof tops of the adjacent buildings. The terms kd and kf control the dependence of the multi-screen diffraction loss versus distance and radio frequency, respectively.

For the LOS case (i.e., urban canyon), the path loss has a strong LOS component between the base station and mobile station and the median path loss can be computed as:

$$PL_{urban\ canyon} = -31.4 + 26 log_{10} d + 20 log_{10} f$$

The following options are available from the Walfisch-Ikegami Coupling property panel:



Enabled: Toggle on/off the use of the Walfisch-Ikegami Coupling node. If disabled, EMIT will not use this Walfisch-Ikegami Coupling node for any simulations.

Base Antenna: Specify the base station antenna(s) in the Walfisch-Ikegami Coupling node's antenna pair. Note that "(all)" can be used to assign all of the project's antennas to the coupling model and that antenna tags can also be used to specify a group of antennas. If Base Antenna = 'Group A' and Mobile Antenna = 'Group B', then all combinations of antennas from Group A to Group B will use the coupling model.

Mobile Antenna: Specify the mobile antenna(s) in the Walfisch-Ikegami Coupling node's antenna pair. Note that "(all)" can be used to assign all of the project's antennas to the coupling model and that antenna tags can also be used to specify a group of antennas. If Base Antenna = 'Group A' and Mobile Antenna = 'Group B', then all combinations of antennas from Group A to Group B will use the coupling model.

Path Loss Type: Specifies the path loss type for the Walfisch-Ikegami node. The path loss type can be either line-of-sight (i.e., urban canyon) or non-line-of-sight (NLOS).

Environment: Specifies the propagation environment where the antennas are located.

Roof Height: Specifies the typical roof height for the location of the antennas.

Distance Between Buildings: Specifies the typical building to building separation for the location of the antennas.

Street Width: Specifies the typical width of the streets for the location of the antennas.

Incidence Angle: Specifies the angle of incidence of the signal from the base station to the mobile station.

Indoor Propagation Coupling

The Indoor Propagation Coupling model can estimate indoor propagation loss. One use for this ability is in site planning to ensure adequate coverage while minimizing the number of access points required. In Recommendation P.1238-7, the ITU recommends a site-general model which requires minimal path or site specific information. The model assumes that both the base station and the portable terminal are located inside the same building and the indoor radio path loss is then characterized by both an average path loss and its associated fading statistics.

The recommended model accounts for the loss through multiple floors which enables planners to reuse frequencies between floors. The power loss coefficients specified for the model include an allowance for transmission of the signal through walls, obstacles, and other loss mechanisms that are common in buildings. This differs from site-specific models which would require explicitly accounting for the loss due to each wall the signal passes through. The basic indoor path loss model has the form:

$$L_{total} = 20log_{10}f + Nlog_{10}d + L_f(n) - 28 \qquad \text{(dB)}$$

N: distance power loss coefficient
f: frequency (MHz)
d: separation distance (m)
L_f: floor penetration loss factor (dB)
n: number of floors between base station and portable terminal (n>=1)

The tables (from ITU-R P.1238-7) below have some typical parameters for various frequency ranges and building types, which were derived from measurements at various locations. Note that EMIT's implementation of the site general model only includes data for which there is both a power loss coefficient and a floor penetration loss factor. Additionally, if there is data for one building type at a particular frequency, but the other building types are missing that data, then EMIT defaults to the values for the building type that is available.

Power loss coefficients, *N*, for indoor transmission loss calculation

Frequency	Residential	Office	Commercial
900 MHz	–	33	20
1.2-1.3 GHz	–	32	22
1.8-2 GHz	28	30	22
2.4 GHz	28	30	
3.5 GHz		27	
4 GHz	–	28	22
5.2 GHz	30 (apartment) 28 (house)[(2)]	31	–
5.8 GHz		24	
60 GHz[(1)]	–	22	17
70 GHz[(1)]	–	22	–

(1) 60 GHz and 70 GHz values assume propagation within a single room or space, and do not include any allowance for transmission through walls. Gaseous absorption around 60 GHz is also significant for distances greater than about 100 m which may influence frequency reuse distances (see Recommendation ITU-R P.676).

(2) Apartment: Single or double storey dwellings for several households. In general most walls separating rooms are concrete walls.

House: Single or double storey dwellings for a household. In general most walls separating rooms are wooden walls.

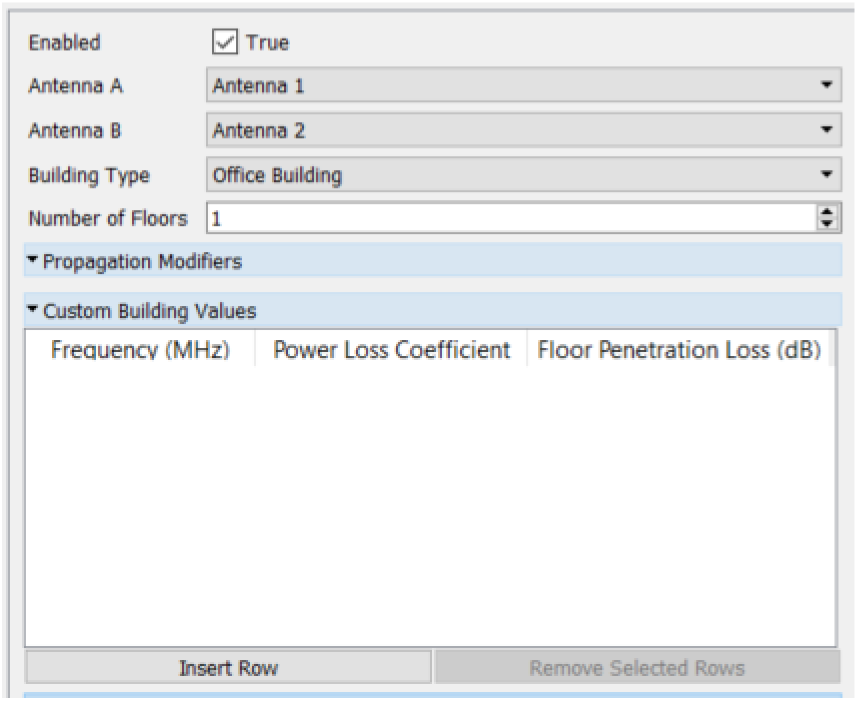
Floor penetration loss factors, L_f (dB) with *n* being the number of floors penetrated, for indoor transmission loss calculation ($n \geq 1$)

Frequency	Residential	Office	Commercial
900 MHz	–	9 (1 floor) 19 (2 floors) 24 (3 floors)	–
1.8-2 GHz	4 n	15 □+ 4 (n – 1)	6 □+ 3 (n – 1)
2.4 GHz	10[(1)] (apartment) 5 (house)	14	
3.5 GHz		18 (1 floor) 26 (2 floors)	
5.2 GHz	13[(1)] (apartment) 7[(2)] (house)	16 (1 floor)	–
5.8 GHz		22 (1 floor) 28 (2 floors)	

(1) Per concrete wall.

(2) Wooden mortar.

The following options are available from the Indoor Propagation Coupling property panel:



Enabled: Toggle on/off the use of the Indoor Propagation Coupling node. If disabled, EMIT will not use this Indoor Propagation Coupling node for any simulations.

Antenna A: Specify the first antenna in the Indoor Propagation Coupling node's antenna pair. Note that "(all)" can be used to assign all of the project's antennas to the coupling model and that antenna tags can also be used to specify a group of antennas. If Antenna A = 'Group A' and Antenna B = 'Group B', then all combinations of antennas from Group A to Group B will use the coupling model.

Antenna B: Specify the second antenna in the Indoor Propagation Coupling node's antenna pair. Note that "(all)" can be used to assign all of the project's antennas to the coupling model and that antenna tags can also be used to specify a group of antennas. If Antenna A = 'Group A' and Antenna B = 'Group B', then all combinations of antennas from Group A to Group B will use the coupling model.

Building Type: Specifies the type of building (e.g. office building, commercial building, residential apartment, or residential house) where the antennas are located.

Number of Floors: Specifies the number of floors between the antennas specified for the Indoor Propagation coupling model. The data specified in ITU-R P.1238-7 limits this to 3 floors or less.

Custom Building Values: Enables a custom indoor environment to be specified in terms of the power loss coefficient and floor penetration loss factors as a function of frequency. Unlike the data in ITU-R P.1238-7, there is no limit on the number of floors that these values can represent.

Log Distance Coupling

The Log Distance Coupling model is a generic extension of the Friis free space path loss model. Unlike the Friis free space model which is limited to estimating the propagation loss for unobstructed, clear paths between the transmitter and receiver, the Log Distance coupling model can be used to estimate the propagation loss for a wide range of environments. The primary difference between the Friis free space model and the Log Distance model is the path loss exponent which can be specified for the latter model based on the environment. The equations for the Log Distance coupling model are shown below:

$$PL_{d>d_0} = PL_{d_0} + 10n\log_{10}\left(\frac{d}{d_0}\right) + \chi$$

$$PL_{d_0} = Path\ loss\ in\ dB\ at\ a\ distance\ d_0$$

$$PL_{d>d_0} = Path\ loss\ in\ dB\ at\ an\ arbitrary\ distance\ d$$

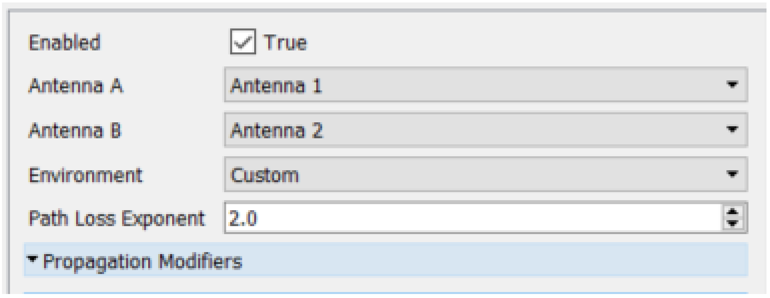
$$n = Path\ loss\ exponent$$

$$\chi = Zero - mean\ Gaussian\ distributed\ random\ variable\ (in\ dB)$$
$$with\ standard\ deviation\ \sigma, used\ to\ model\ a\ shadowing\ effect$$

The table below shows some typical values for the path loss exponent based on the environment type as well as the path loss exponent values used by EMIT for each environment. EMIT also provides a Custom environment type that allows the path loss exponent to be set to any value. By default, the Log Distance coupling model does not include the shadowing effects. However, like most EMIT coupling models, these effects can be modeled using the Shadowing Fading Type specified in the Propagation Modifiers folder.

Environment	Path Loss Exponent (Typical Values)	EMIT Path Loss Exponent
Free Space	2.0	2.0
Urban area cellular radio	2.7-3.5	3.1
Shadowed urban cellular radio	3.0-5.0	4.0
Inside a building with line of sight	1.6-1.8	1.7
Inside a building (obstructed)	4.0-6.0	5.0
Inside a factory (obstructed)	2.0-3.0	2.5

The following options are available from the Log Distance Coupling property panel:



Enabled: Toggle on/off the use of the Log Distance Coupling node. If disabled, EMIT will not use this Log Distance Coupling node for any simulations.

Antenna A: Specify the first antenna in the Log Distance Coupling node's antenna pair. Note that "(all)" can be used to assign all of the project's antennas to the coupling model and that antenna tags can also be used to specify a group of antennas. If Antenna A = 'Group A' and Antenna B = 'Group B', then all combinations of antennas from Group A to Group B will use the coupling model.

Antenna B: Specify the second antenna in the Log Distance Coupling node's antenna pair. Note that "(all)" can be used to assign all of the project's antennas to the coupling model and that antenna tags can also be used to specify a group of antennas. If Antenna A = 'Group A' and Antenna B = 'Group B', then all combinations of antennas from Group A to Group B will use the coupling model.

Environment: Selects an environment type from the pre-defined list in EMIT. The environment type determines the path loss exponent based on the table above. A Custom environment allows the user to specify the path loss exponent used by the Log Distance coupling model.

Path Loss Exponent: Specifies the value used for the Path Loss Exponent in computing the Log Distance coupling.

Erceg Coupling

The Erceg Coupling model is an empirical model developed to estimate the propagation loss in various complex environments. As an empirical model, it is derived from measurements at different frequencies across multiple sites. The Erceg Coupling model is particularly popular for estimating median path loss for WiMAX installations in the 2.5 GHz and 3.5 GHz ranges.

The Erceg Coupling model provides an estimate of the median path loss as a function of carrier frequency, base station and mobile station antenna heights, and the distance between the base station and mobile station. The Erceg Coupling model is applicable over the following range of parameters:

- 800 MHz ≤ frequency ≤ 3700 MHz
- 10 m ≤ h_b ≤ 80 m (base station antenna height)
- ~2 m (mobile station antenna height)
- 0.1 km ≤ distance ≤ 8 km (cell range)

The Erceg coupling model defines the median path loss by:

$$PL = L_0 + 10\gamma \log_{10}\left(\frac{d}{d_0}\right) + s \qquad for\ d \geq d_0$$

$$L_0 = 20 \log_{10}\left(\frac{4\pi d_0}{\lambda}\right)$$

$$d_0 = 100\ m$$

$$\gamma = \left(a - bh_b + \frac{c}{h_b}\right) + x\sigma_\gamma$$

$$s = y\sigma$$

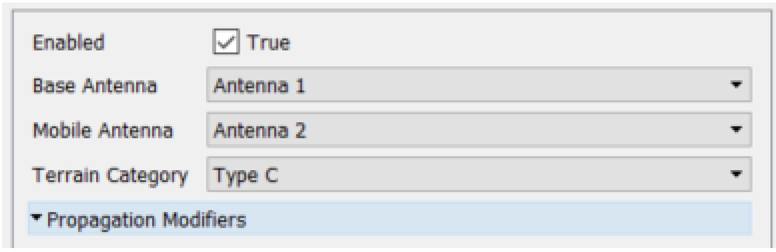
$$\sigma = \mu_\sigma + z\sigma_\sigma$$

$$x, y, z\ are\ Gaussian\ random\ variables\ N(0,1)$$

The parameter values for the equations above depend on the Terrain Category. There are 3 main terrains defined by the Erceg model: A) Hilly with moderate to heavy tree density, B) Hilly with light tree density or flat with moderate to heavy tree density, and C) flat with light tree density. The parameter values used by EMIT for the three categories are shown below.

Parameter	Terrain Category		
	A (Hilly / moderate to heavy tree density)	B (Hilly / light tree density or flat / moderate to heavy tree density)	C (Flat / light tree density)
a	4.6	4.0	3.6
b(m^{-1})	0.0075	0.0065	0.0050
c (m)	12.6	17.1	20.0
σ_γ	0.57	0.75	0.59
μ_σ	10.6	9.6	8.2
σ_σ	2.3	3.0	1.6

The following options are available from the Erceg Coupling property panel:



Enabled: Toggle on/off the use of the Erceg Coupling node. If disabled, EMIT will not use this Erceg Coupling node for any simulations.

Base Antenna: Specify the base station antenna(s) in the Erceg Coupling node's antenna pair. Note that "(all)" can be used to assign all of the project's antennas to the coupling model and that antenna tags can also be used to specify a group of antennas. If Base Antenna = 'Group A' and Mobile Antenna = 'Group B', then all combinations of antennas from Group A to Group B will use the coupling model.

Mobile Antenna: Specify the mobile antenna(s) in the Erceg Coupling node's antenna pair. Note that "(all)" can be used to assign all of the project's antennas to the coupling model and that antenna tags can also be used to specify a group of antennas. If Base Antenna = 'Group A' and Mobile Antenna = 'Group B', then all combinations of antennas from Group A to Group B will use the coupling model.

Terrain Category: Specifies the type of terrain where the base station is located.

5G Channel Model Coupling

The 5G Channel Coupling model is an empirical model developed to estimate the propagation loss for 5G deployments in various complex environments. As an empirical model, it is derived from measurements at different frequencies across multiple sites. The 5G Channel Coupling model is particularly useful for estimating median path loss for deployments utilizing frequencies above 6 GHz. The 5G Channel Coupling model is particularly useful for estimating median path loss for deployments utilizing frequencies above 6 GHz and is derived from NYU's Channel Simulator.

The 5G Channel Coupling model provides an estimate of the median path loss as a function of carrier frequency, base station and mobile station antenna heights, and the distance between the base station and mobile station. The 5G Channel Coupling model is applicable over the following range of parameters:

- 6 GHz ≤ frequency ≤ 100 GHz
- 10 m ≤ h_b ≤ 150 m (base station antenna height)
- Distance ≤ 10 km

The 5G channel coupling model combines a statistical model for 5G wireless systems with NYU's Building Penetration Loss model as well as models for estimating the propagation loss due to atmospheric absorption and rain attenuation. The statistical model defines three potential environments: 1) Urban Macrocell, 2) Urban Microcell, and 3) Rural Macrocell with varying path loss exponents and shadow fading standard deviations for each environment. The equations for the additional path loss in each environment, relative to free space path loss, are given below:

$$RMa\ LOS = 23.1 * \left(1.0 - 0.03 * \frac{(h_b - 35.0)}{35.0}\right) * \log_{10}(distance) + 1.7$$

$$RMa\ NLOS = 30.7 * \left(1.0 - 0.049 * \frac{(h_b - 35.0)}{35.0}\right) * \log_{10}(distance) + 6.7$$

$$Urban\ Models = PLE * 10 * \log_{10}\left(\frac{distance}{ref\_distance}\right) + SF$$

$$h_b = base\ station\ height$$

$$refdistance = 1.0$$

$$SF = Shadow\ Fading$$

$$PLE = Path\ Loss\ Exponent$$

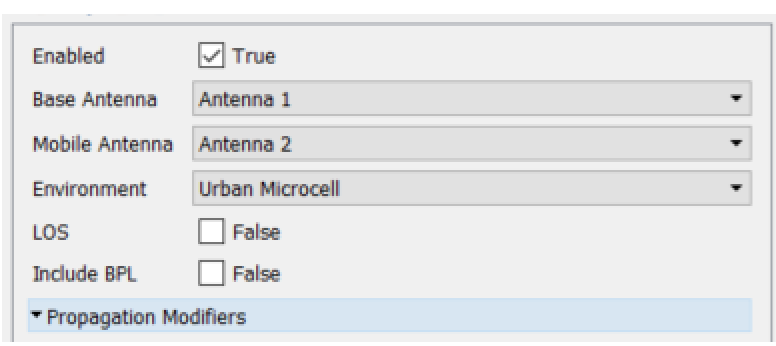
The 5G channel coupling model also provides an estimate of the median path loss when outdoor to indoor communications are of interest. This is by including NYU's Building Penetration Loss (BPL) model in the estimate. The BPL is defined as follows:

$$BPL_{NYU}[dB] = 10 * \log_{10}(A + B * f_c^2)$$

$$A = 5; \quad B = 0.03\ (low\ loss\ buildings)$$

$$A = 10; \quad B = 5\ (high\ loss\ buildings)$$

The following options are available from the 5G Channel Model Coupling property panel:



Enabled: Toggle on/off the use of the 5G Channel Model Coupling node. If disabled, EMIT will not use this 5G Channel Model Coupling node for any simulations.

Base Antenna: Specify the base station antenna(s) in the 5G Channel Model Coupling node's antenna pair. Note that "(all)" can be used to assign all of the project's antennas to the coupling model and that antenna tags can also be used to specify a group of antennas. If Base Antenna = 'Group A' and Mobile Antenna = 'Group B', then all combinations of antennas from Group A to Group B will use the coupling model.

Mobile Antenna: Specify the mobile antenna(s) in the 5G Channel Model Coupling node's antenna pair. Note that "(all)" can be used to assign all of the project's antennas to the coupling model and that antenna tags can also be used to specify a group of antennas. If Base Antenna = 'Group A' and Mobile Antenna = 'Group B', then all combinations of antennas from Group A to Group B will use the coupling model.

Environment: Specifies the propagation environment where the antennas are located.

Line-of-Sight: If true the model assumes the mobile antenna(s) have a clear line-of-sight to the base antenna(s).

Include BPL: If true the NYU Building Pentration Loss (BPL) model is included in the propagation loss calculations.

> **NYU BPL Model:** Specifies whether to use the A and B coefficients for the low-loss or high-loss model.

Coupling Links

Creating a link to an HFSS or HFSS 3D Layout design prompts EMIT to use any S-parameter and/or far field results from an HFSS or HFSS 3D Layout design to define the coupling between antenna and/or emitter ports in the EMIT design. The far field patterns can also be imported into EMIT for use with EMIT's parametric coupling models.

The following requirements must be met in order for EMIT to obtain coupling data from a linked HFSS or HFSS 3D Layout design:

- Only *Terminal* solution types are supported.
- The HFSS or HFSS 3D Layout design must have at least two ports.
- One or more sweeps must be defined over which the S-parameters are computed.
- The EMIT design must be in the same project as the HFSS or HFSS 3D Layout design.
- The HFSS or HFSS 3D Layout design should be a multi-port design with the analysis set up to compute S-parameters that the EMIT design will use for coupling between ports.

The requirements for EMIT to obtain far field patterns directly from a linked HFSS design (HFSS 3D Layout designs are not supported) are less stringent:

- Modal, Terminal, or Hybrid solution types are supported.
- At least one port must be specified.
- The far field pattern must be computed over at least two frequency points.
- The far field pattern must be computed with equally spaced frequency points.
- The EMIT design must be in the same project as the HFSS design.

Use the following procedure to configure an HFSS design to link far field patterns to EMIT:

- Each port should be referenced to a Coordinate System.
- Create an Infinite Sphere to define the angular domain and angular spacing for the far field patterns. For dynamic scenarios, it is recommended to compute the far field patterns over the full sphere. The Infinite Sphere should be linked to each Coordinate System.
- Create a Source Group for each port for which you want to compute a far field pattern.
- Right-click **Excitations** > **Edit Sources**. Ensure that each port is configured with the correct magnitude/phase.
- Run the analysis.

Creating a Coupling Link

Creating a coupling link involves the following steps:

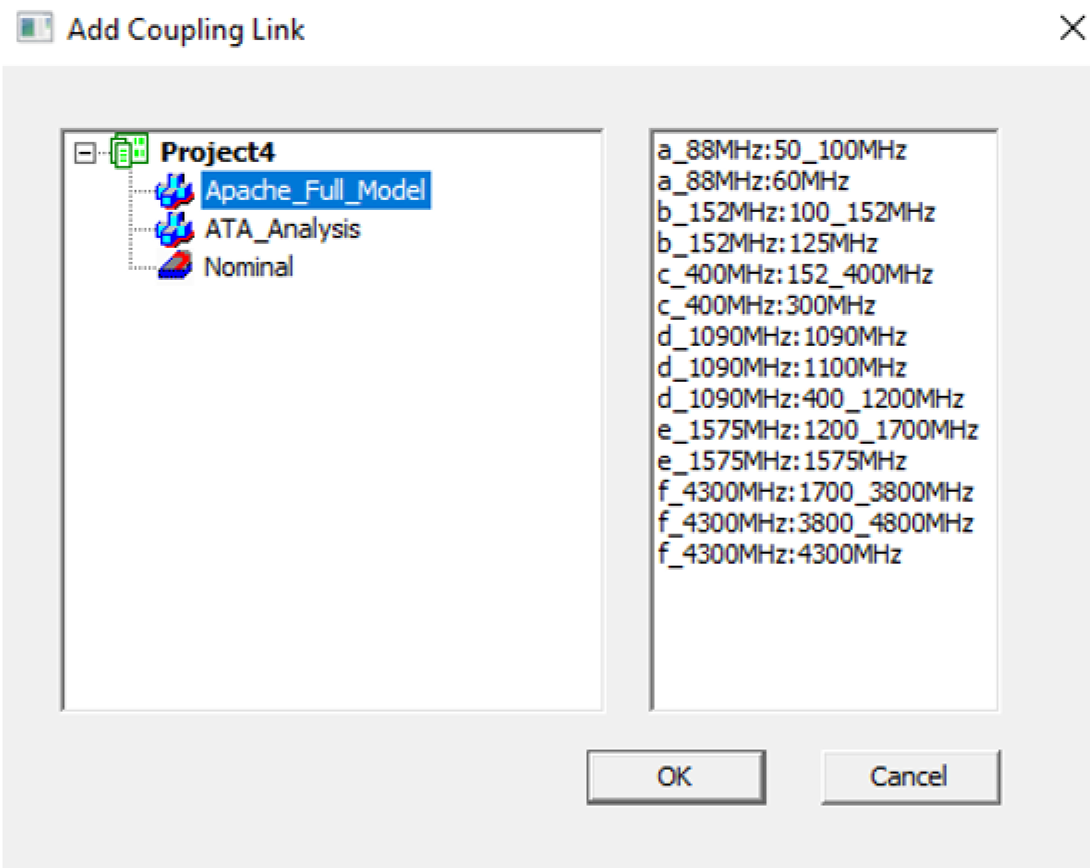
- Adding an HFSS or HFSS 3D Layout link
- Assigning ports
- Activating sweeps

> **Note:**
> **The HFSS or HFSS 3D Layout design must be in the same project as the EMIT design to which it is being added.**

Adding a Coupling link

To add an HFSS or HFSS 3D Layout link:

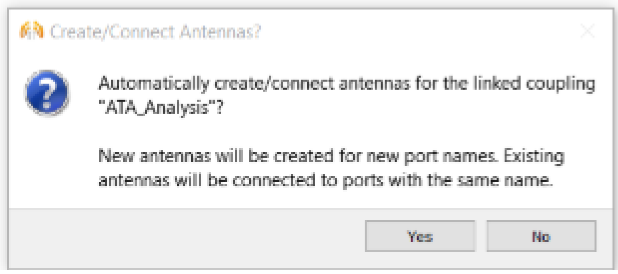
1. Do one of the following:
   - Click **EMIT** > **Add Coupling Link**.
   - In the Project Tree, right-click **Coupling** and select **Add Coupling Link**.
2. The **Add Coupling Link** dialog box displays information about the designs that are available for linking in the project:
   - Select the design that you want to link to in the left pane and click **OK**.



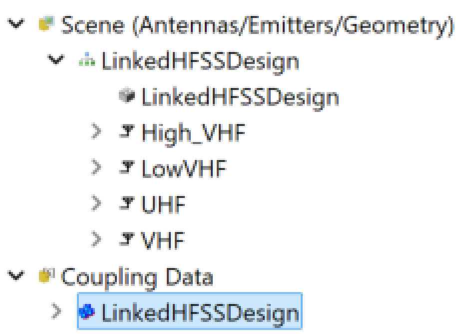
Note:
If you do not want to open the Coupling Editor after each link is added, then you can change this behavior in **Tools > Options > EMIT** by deselecting **Show Coupling Editor for new link**.

The HFSS or HFSS 3D Layout design is then added to Coupling folder in the Project Manager.

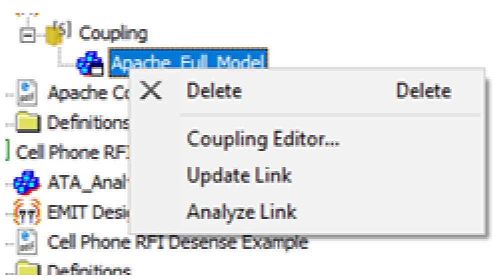
When you click **OK**, the Coupling Editor opens and asks if you'd like to create new antennas for each port in the HFSS or HFSS 3D Layout design. If you choose **Yes**, a new antenna is created in the EMIT design and associated with each of the ports. Choosing **No** leaves the ports unassigned to antennas and you can then associate the ports with EMIT antennas manually in the Coupling Editor.



The linked design will appear below the Coupling Data node in EMIT's Coupling Editor. This contains the sweep information from the HFSS design and the simulated S-parameter data (if applicable). If **Yes** was selected, EMIT's Coupling Editor also shows a group node with a name matching the linked design and all the antennas created as children of this group node. The antenna names match the port names used in the HFSS design. A CAD model of the HFSS design is also imported as an obj model in EMIT (for visualization purposes only).

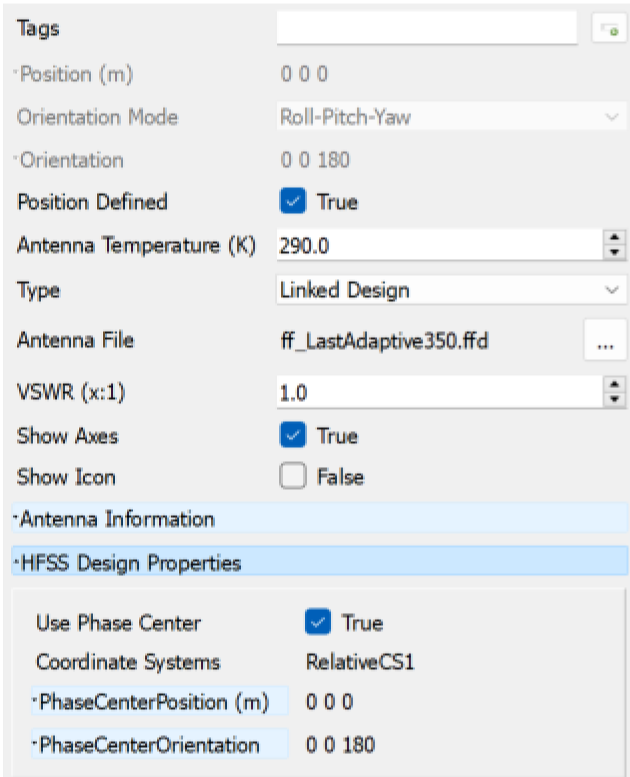


Right-clicking on the HFSS (or HFSS 3D Layout) project node results in the following options:



- **Coupling Editor** – opens the Coupling Editor.
- **Update Link** – updates the linked design in EMIT with any changes that have been made in the HFSS design.
- **Analyze Link** – runs the linked design. Note that the design is analyzed regardless of whether any changes have been made to the design that would require a new analysis.

If the far field patterns were linked with EMIT, the antenna's **Type** will be **Linked Design** and a few additional parameters will be shown:



- **Use Phase Center** – If **True**, the antenna in EMIT will be positioned at the location of the reference coordinate system. If **False**, the EMIT antenna will be located at the linked design's corresponding port position.
- **Coordinate Systems** – The port's reference coordinate system in the linked design. This is the coordinate system that is located closest to the port center position.
- **Phase Center Position** – The location of the reference coordinate system in the linked design, relative to the global origin.
- **Phase Center Orientation** – The orientation of the reference coordinate system in the linked design, relative to the global origin.

4 - EMIT Components

For passive components, EMIT computes the noise added by the component assuming the system is at 290 K. The equation below shows the computation of the added noise generalized to N port components. This noise is typically negligible in cosite interference analyses, but can significantly impact a link analysis. Note that EMIT does not currently compute the noise added due to Terminators.

$$N_{added,j} = kT_{sys} - \sum_{i \neq j} kT_{sys}G_{ji}$$

$$N_{added,j} = Noise\ added\ by\ passive\ comp\ out\ of\ port\ j$$

$$k = Boltmann's\ constant$$

$$T_{sys} = Temperature\ of\ the\ system\ (K)$$

$$G_{ji} = Gain\ of\ the\ component$$

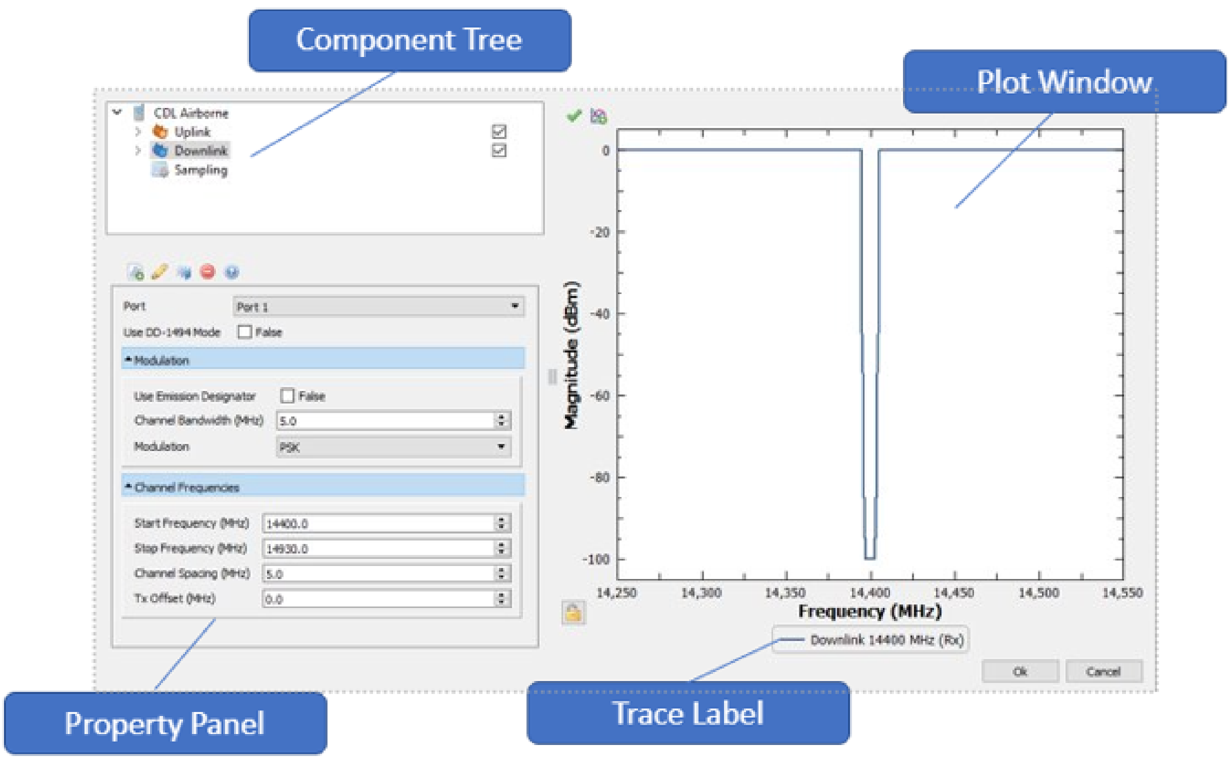
EMIT provides the following components:

- Amplifiers
- Antennas
- Cables
- Circulators
- Dividers
- Emitters
- Filters
- Isolators
- Multiplexers
- Radios
- Switches
- Terminators
- Plot Properties
- Plot Trace Properties
- Plot Marker Properties

EMIT Components Edit Dialog

Settings for EMIT components are specified using the **EMIT Components Edit Dialog**. You can use this dialog to edit parameters associated with EMIT components as well as to provide a built-in plotting panel that displays the pertinent component frequency response as parameters are changed. While each component has different settings that are described in the help sections for the individual components, this section describes the functions of the EMIT Component Edit Dialog that are common to all EMIT components.

The figure below shows the main parts of the EMIT Components Edit Dialog.



Component Tree: The Component Tree displays one or more nodes depending on the component being edited. Each node has a set of properties that define the various aspects of the component. Selecting a node in the Component Tree displays the properties for that node in the Property Panel.

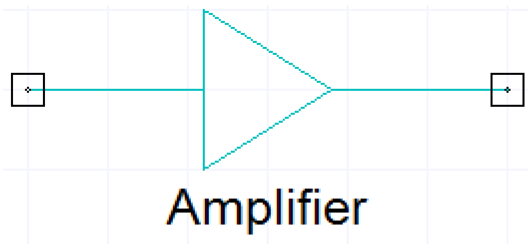
Property Panel: The Property Panel is where you can change the specific settings and options for the component. As settings are altered the component characteristics displayed in the Plot Window are updated automatically.

Plot Window: The plot window displays the frequency response of the component, and updates automatically when any changes are made to the component properties. Selecting a node in the

Component Tree shows the frequency response for that quantity in the Plot Window. Plot properties such as axis labels, and titles can be changed in the plot’s property panel which is accessible by clicking on the **trace label**.

Trace Label: The trace label shows the particular quantity displayed in the plot. Clicking on the trace label changes the property panel to show the plot properties. Subsequent clicks cycle through all the available property panels for the particular plot being displayed.

Amplifiers

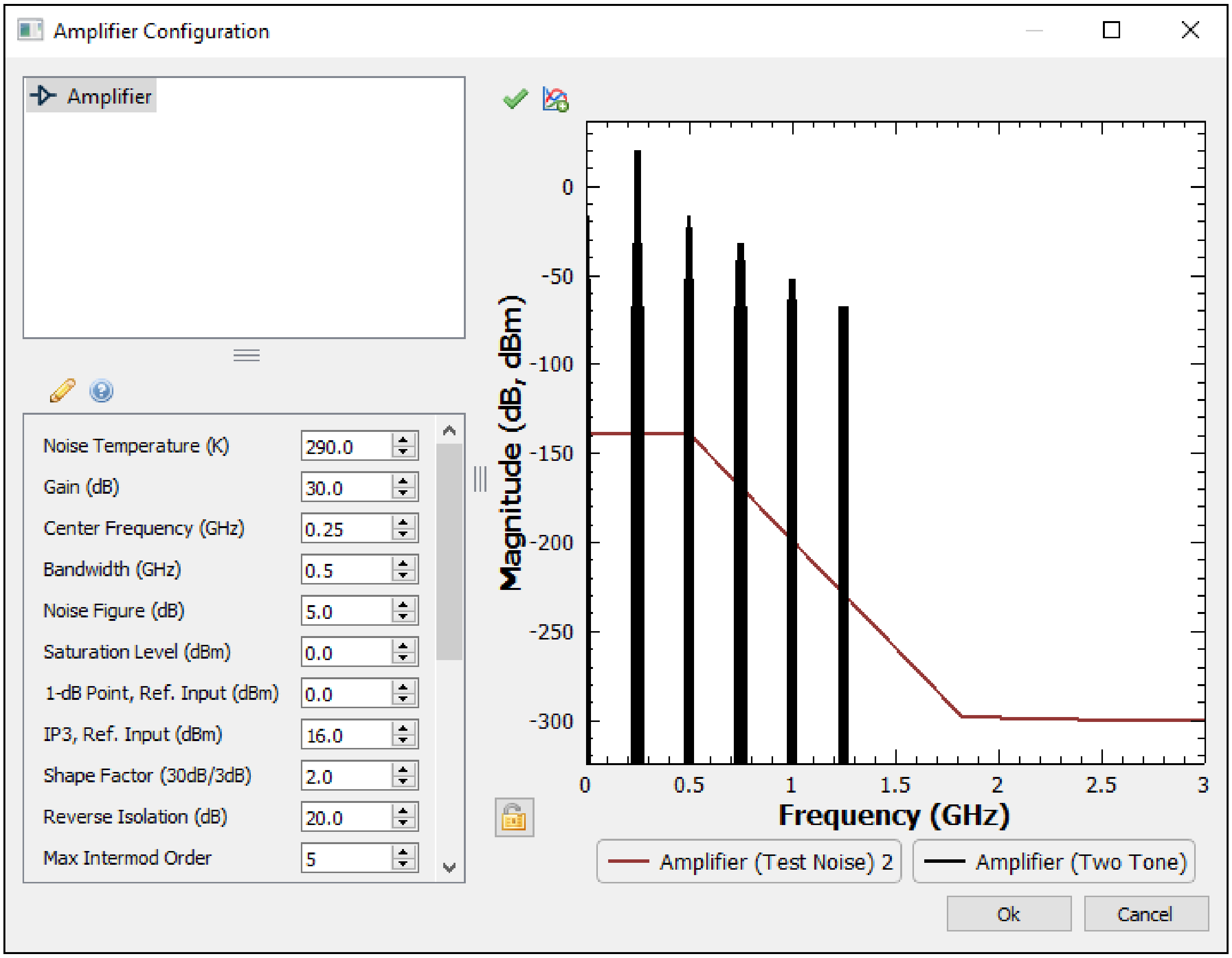


Amplifiers are added to an EMIT design from the EMIT Elements components library.

The gain profile of the amplifier is then defined in the amplifier's configuration window. When computing the EMI margin, EMIT uses the amplifier's transmission characteristics in addition to accounting for both the mismatch loss and insertion loss according to:

$$P_{out}(dB) = P_{in} + 10log_{10}(|S_{21}|^2)$$

Amplifiers in EMIT are modeled as nonlinear components This models the effect of multiple signals on the interference at the receiver, which can result in spectral growth (intermodulation products) within the nonlinear amplifier.



Noise Temperature (K): Noise power added by the amplifier.

Gain: Define the in-band gain of the amplifier in dB.

Center Frequency: The desired signal frequency for which the amplifier is defined.

Bandwidth: The 3-dB bandwidth of the amplifier. Any signals input into the amplifier within the bandwidth will be amplified by the gain of the amplifier.

Noise Figure: The amplifier's noise figure in dB. The noise figure is used to calculate the broadband noise added by the amplifier

Saturation Level: Specifies an input power level for which EMIT will consider the amplifier saturated. When an amplifier is saturated, EMIT will flag it as such in the results and will not trace the input signal any further. The saturation level cannot be set lower than the 1-dB Point.

1-dB Point Ref. Input: The amplifier's 1-dB compression point referred to the input of the amplifier.

IP3, Ref. Input: The amplifier's third order intercept point referred to the input of the amplifier.

Shape Factor: The shape factor determines the frequency roll-off outside the amplifier's bandwidth. The shape factor specifies the slope of the amplifier's gain profile between its 3 dB and 30 dB points. An ideal amplifier that only amplifies signals within its bandwidth would have a shape factor of 1. This slope is then extrapolated from the 30 dB point down to the amplifier's noise floor as shown below. 3 dB Point: The frequency at which the amplifier's gain profile is 3 dB below the amplifier's peak gain. EMIT calculates the 3 dB points from the amplifier's bandwidth and center frequency: 30 dB Point: The frequency at which the amplifier's gain profile is 30 dB below the amplifier's peak gain.

Reverse Isolation: This is the amplifier's reverse isolation. The reverse isolation of an amplifier is a measure of how well a signal applied to the amplifier's output is "isolated" from its input. It is equivalent to the S12 S-parameter.

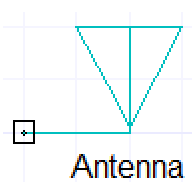
Max Intermod Order: Specifies the highest order of intermodulation product that EMIT will calculate for the amplifier.

Harmonic Intercept Points, Ref. Input: By default, EMIT estimates higher order intercept points based on the value provided for the IP3. The Harmonic Intercept Points table allows you to specify harmonic intercept points that will be used to more accurately compute the intercept points. The intercepts specified are used in place of EMIT's approximation. For further information see the following references:

http://www.microwavejournal.com/articles/3595-a-method-to-predict-the-level-of-intermodulation-products-in-broadband-power-amplifiers

C.A.A. Waas, "A Table of Intermodulation Products", *Journal of the Institution of Radio and Communication Engineers*, Vol. 95, Issue 33, pp. 31-39, Jan. 1948

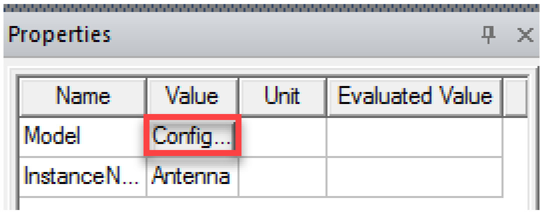
Antennas



Antennas are added to an EMIT design from the EMIT Elements components library. All antennas couple to each other as well as to all emitters in the design.

Antennas can be assigned to ports in dynamically linked HFSS or HFSS 3D Layout designs to provide the necessary coupling data for the EMIT simulation. If an antenna is not assigned to a link port, the value used for coupling to other antennas and emitters is the value defined for the Global Default Coupling (Fixed Coupling).

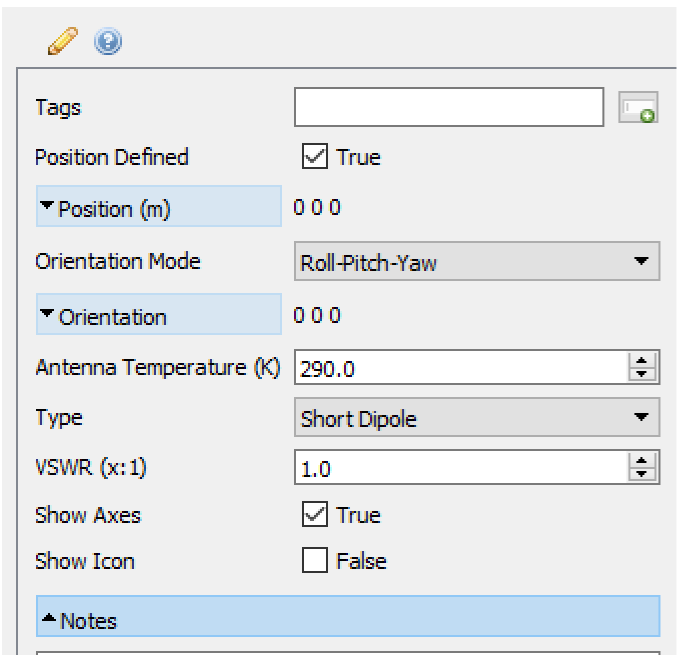
Each antenna has the following properties defined in the Properties window:



Instance Name: Name of the antenna as shown in the schematic editor.

When you click **Configure** or double-click the antenna in the schematic editor, the **Antenna Configuration** window opens

The options available in the **Antenna Configuration** window depend on the type of antenna being used. The properties listed are general properties that apply to all antenna types.



Tags: Tags are used to group multiple antennas together. They are a space delimited list of strings and each antenna can have any number of tags associated with it (or none). After a tag is specified, the tag appears as an option for each of the Antennas that are assigned to Coupling Models. In this way, if multiple antennas share a tag for example, "MobileAntennas"), then the entire group of antennas can be added to a Coupling Model by simply assigning the tag *MobileAntenna* to the Mobile Antenna input of the Hata Coupling model (for one example).

> **Note:**
>
> Tags cannot be the same as the antenna names and the following strings are reserved for internal use by EMIT and cannot by used as tags: " (all)" and "(undefined)".

Position Defined: A valid RF System in EMIT requires an antenna. However, the antenna does not need to be physically placed within the Scene; that is, an antenna in EMIT is not required to have a physical position defined. An antenna without a position is referred to as a positionless antenna in EMIT. By default, antennas added to EMIT are positionless. In order to define an antenna's position the Position Defined should be enabled (set to True). This can be done by clicking the check box, or automatically by interactively placing the antenna.

Antenna Temperature: Noise that the antenna picks up from radiating bodies within its radiation pattern. It is typically a function of the direction in which the antenna is pointing, its radiation pattern and the state of the surrounding environment. The default value is 290 Kelvin, which is the Earth's blackbody radiation temperature.

> **Note:**
>
> The default 290 K noise temperature would result in a Noise Power = -124 dBm for a receiver with a 100 MHz bandwidth. For a simple receiver with a very low sensitivity, this could limit its operational capacity.

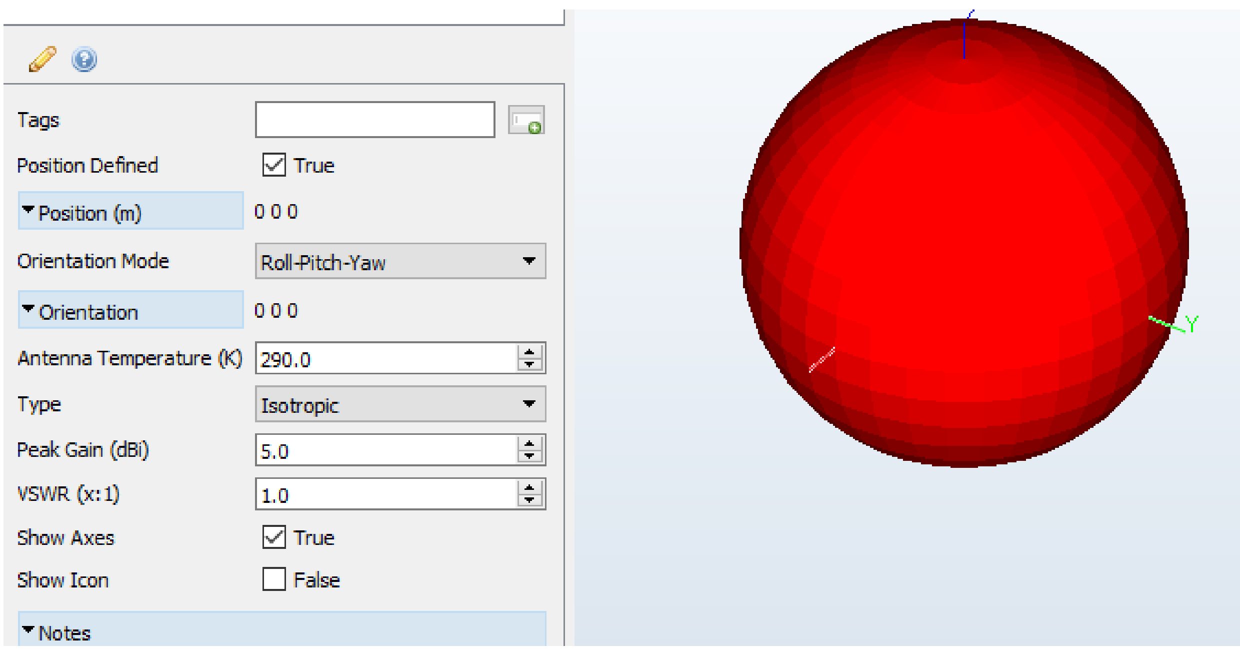
Type: By default in EMIT, new antennas in a project are Isotropic with 0 dBi gain. The directional antenna gain is automatically applied by the specific Coupling Models if applicable (S-Parameter Coupling and Custom Coupling do not apply the antenna gain).

EMIT also provides the option of associating a user-supplied 3D antenna gain pattern with any antenna in the project or using one of several other predefined parametric antennas. The antenna gain pattern selected is then used by EMIT to compute the coupling between antenna pairs. The available antenna types are summarized in the following table.

Isotropic	Creates a 3D antenna gain pattern that is a sphere with constant gain.
By File	User-supplied 3D antenna gain pattern.
Hemitropic	Creates a 3D antenna gain pattern that is a hemisphere with constant gain.
Short Dipole	Creates a toroidal shaped, 3D antenna gain pattern with a peak gain of 1.76 dBi.
Half-wave Dipole	Creates a toroidal shaped, 3D antenna gain pattern with a peak gain of 2.15 dBi.
Quarter-wave Monopole	Creates a 3D antenna gain pattern of a monopole with a peak gain of 5.16 dBi.

Wire Dipole	Creates a half-wave dipole at the user specified frequency. The 3D antenna gain pattern of the Wire Dipole varies with frequency.
Wire Monopole	Creates a quarter wave monopole at the user specified frequency. The 3D antenna gain pattern of the Wire Monopole varies with frequency.
Small Loop	Creates a 3D antenna gain pattern with peak gain of 1.76 dBi. The Small Loop is polarized in the Phi direction.
Directive Beam	Creates a 3D antenna gain pattern with a main beam and up to two sidelobe levels with optional frequency roll-off.
Pyramidal Horn	Creates a pyramidal horn with the user specified dimensions. The 3D antenna gain pattern of the Pyramidal Horn varies with frequency.

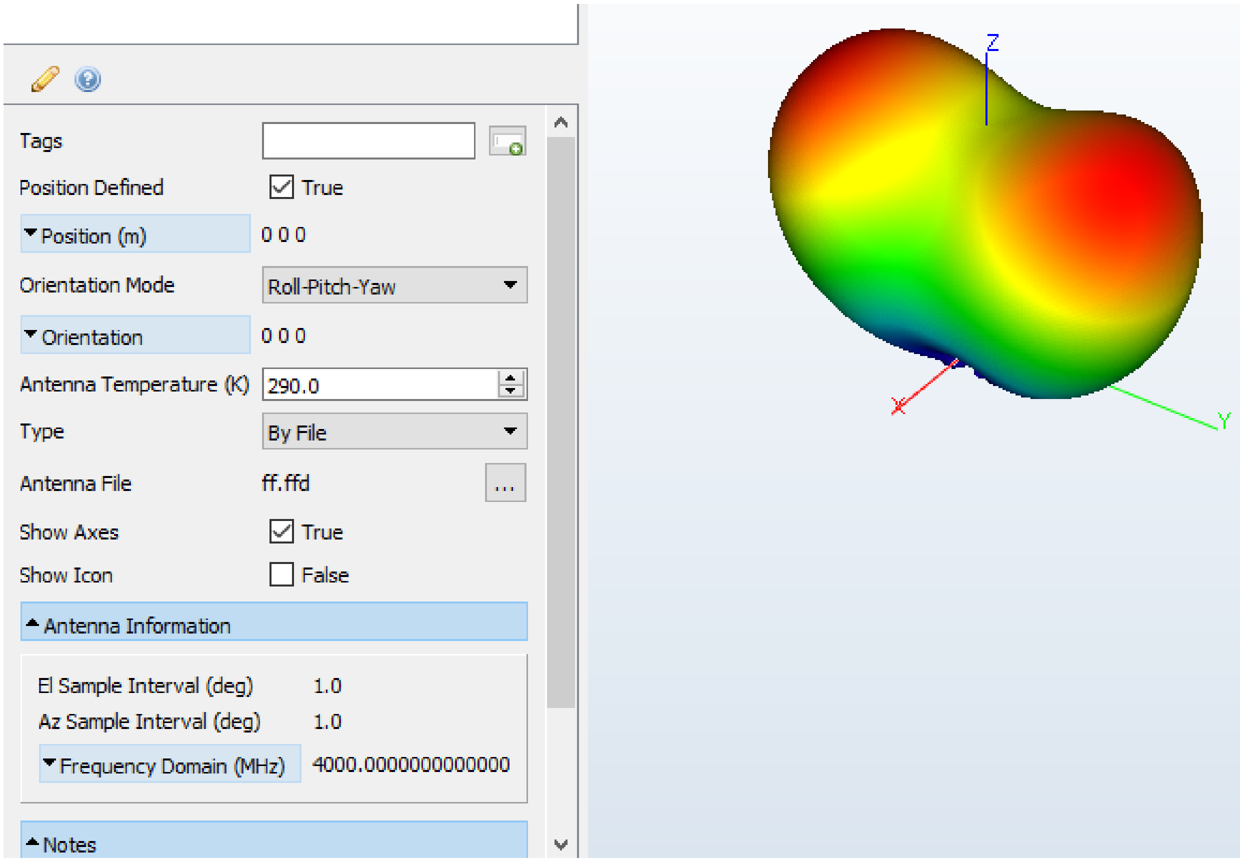
Isotropic: The Isotropic antenna has a constant gain over all frequencies and angles. The pattern is polarized in the theta direction. The pattern does not represent any physical (real) antenna, but is still quite useful for many types of analyses in EMIT. The property panel for an Isotropic antenna and its 3D rendering in EMIT is shown in the following figure.



Peak Gain: The gain of the Isotropic antenna over all frequencies and angles.

For details on the other fields in the property panel, see the General Parameters section.

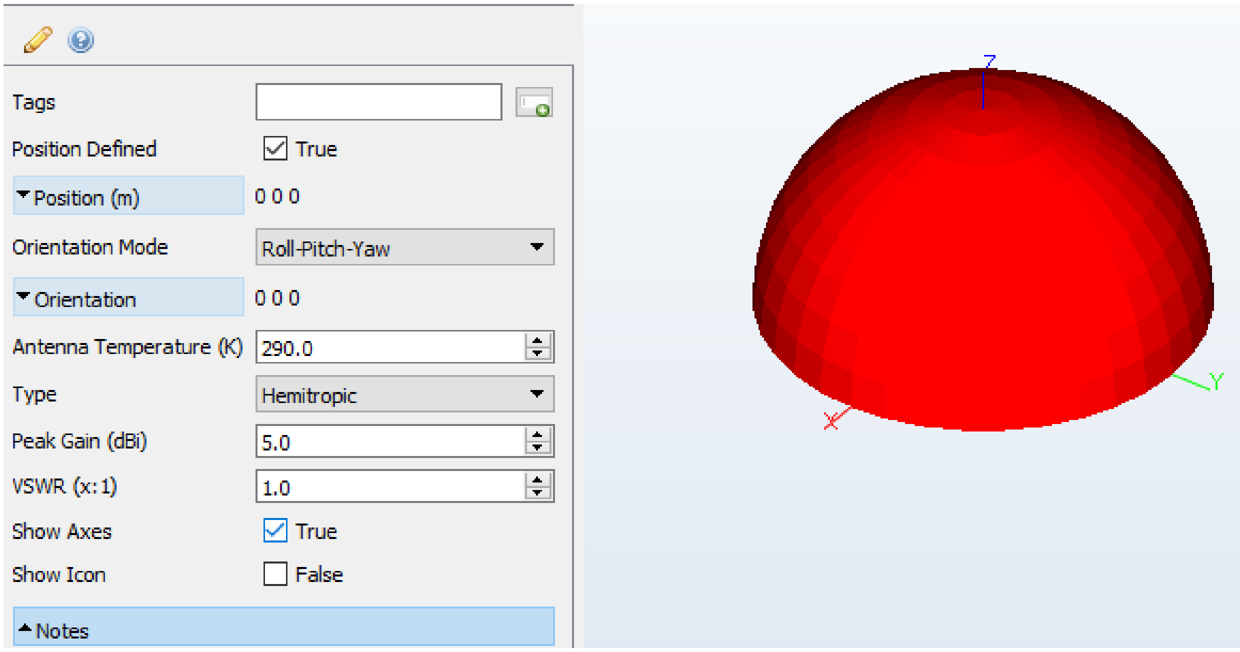
By File: When **By File** is selected, the Antenna File field appears in the property panel. Use the browse button to select a *.patdat, *ffd, *.ffs, or *.ra1 file containing the antenna gain pattern. Note that in order to use the Path Loss and Antenna Gain coupling model, EMIT requires that the spacing between frequencies in the patterns files be equally spaced. If this is not the case, EMIT shows a warning for the antenna and does not use the Path Loss and Antenna Gain coupling model when running the simulation. If you want to remove the warning icon in this case, you can disable the Path Loss and Antenna Gain node in the project tree. Defining an antenna *By File* results in a few additional fields appearing in the property panel as shown in the following example.



Antenna Information: Displays the elevation/azimuth sampling intervals used in the pattern file. The start/stop frequencies, the number of steps and the frequency delta defined in the pattern file are also displayed. The Antenna Information fields cannot be edited.

For details on the other fields in the property panel, see the General Parameters section.

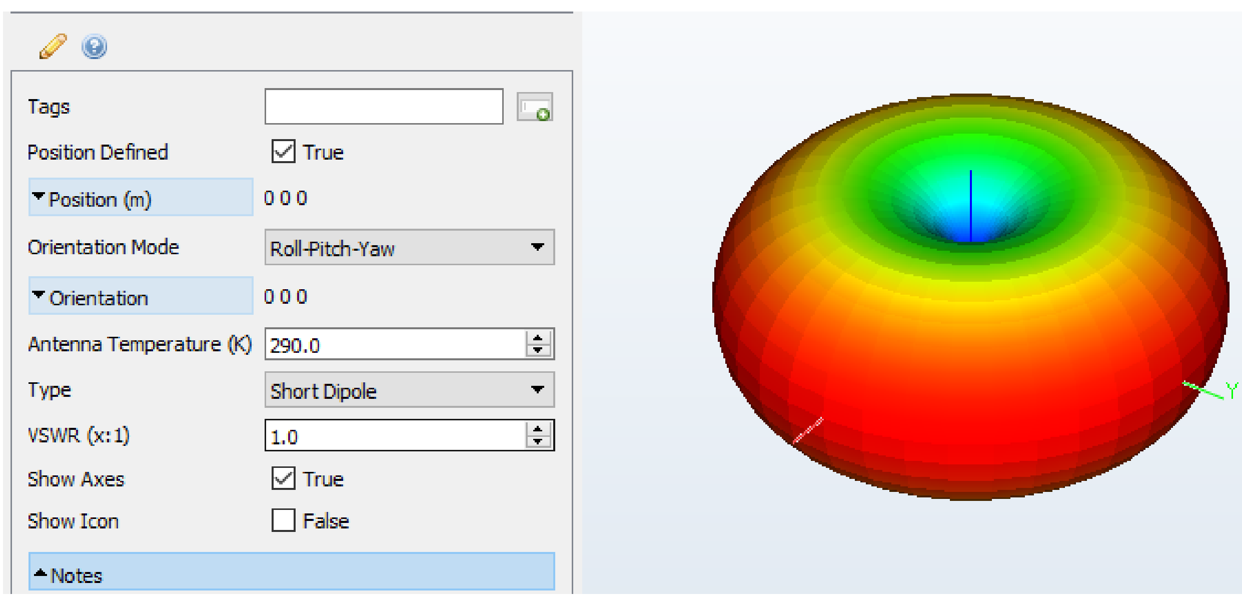
Hemitropic: The Hemitropic antenna has a constant gain over all frequencies and angles of a hemisphere. The pattern is polarized in the theta direction. The pattern does not represent any physical (real) antenna, but is still quite useful for many types of analyses in EMIT. The property panel for an Hemitropic antenna and its 3D rendering in EMIT is shown in the following figure.



Peak Gain: The gain of the Hemitropic antenna over all frequencies and angles. Note that the Hemitropic antenna is only defined for a hemisphere.

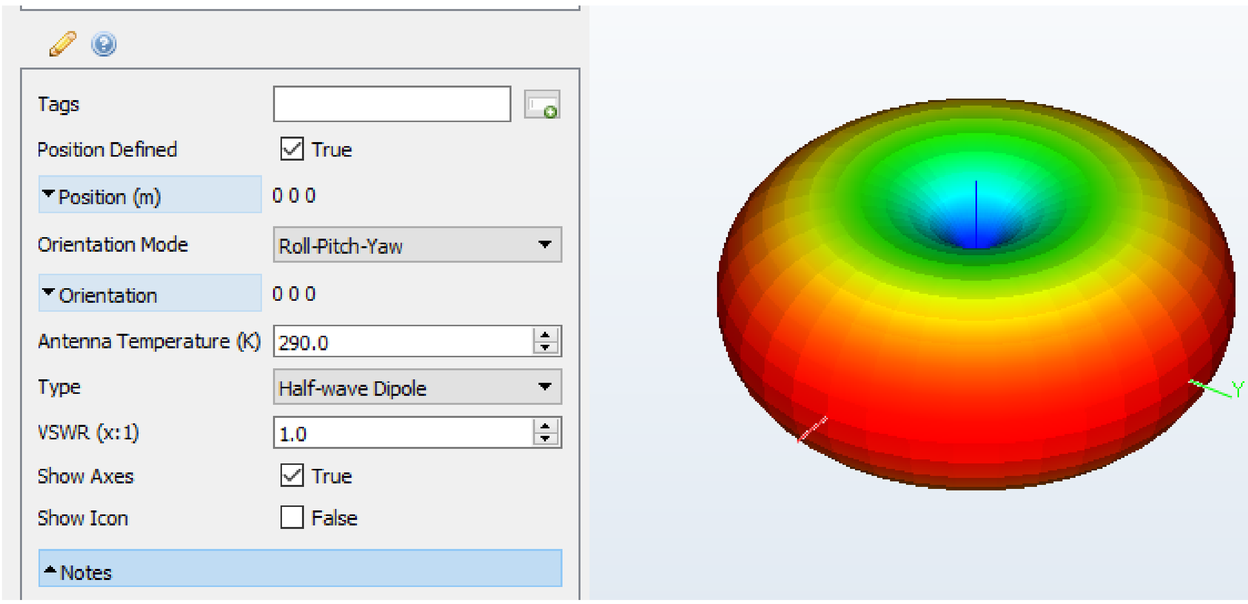
For details on the other fields in the property panel, see the General Parameters section.

Short Dipole: The Short Dipole is an antenna that is constant over all frequencies with a peak gain of 1.76 dBi along the x-y plane. It has the typical toroidal (doughnut-shaped) radiation pattern and is polarized in the theta direction. The property panel for a Short Dipole antenna and its 3D rendering in EMIT is shown in the following figure.



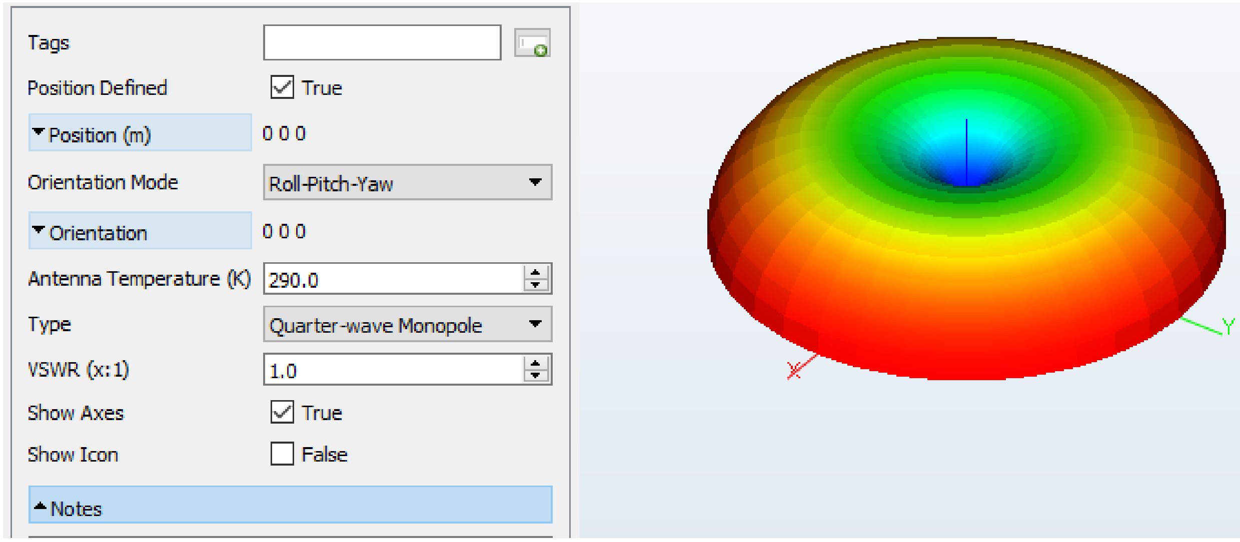
For details on the other fields in the property panel, see the General Parameters section.

Half-wave Dipole: This antenna has the typical toroidal (doughnut-shaped) radiation pattern of a dipole antenna with a peak gain of 2.15 dBi along the x-y plane for all frequencies. The Half Wave Dipole antenna is polarized in the theta direction. The pattern does not represent any physical (real) antenna, but is still quite useful for many types of analyses in EMIT. The property panel for a Half Wave Dipole antenna and its 3D rendering in EMIT is shown in the following figure.



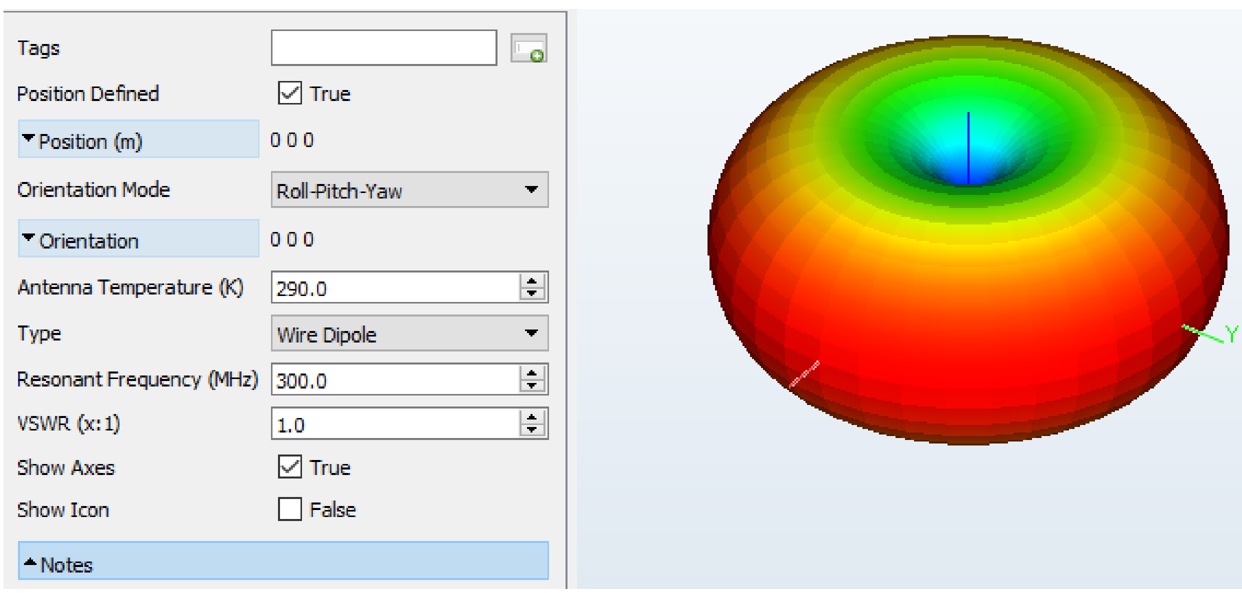
For details on the other fields in the property panel, see the General Parameters section.

Quarter-wave Monopole: This antenna has the typical radiation pattern of a monopole antenna with a peak gain of 5.16 dBi along the x-y plane for all frequencies. The Quarter-wave Monopole antenna is polarized in the theta direction. The pattern does not represent any physical (real) antenna, but is still quite useful for many types of analyses in EMIT. The property panel for a Quarter Wave Monopole antenna and its 3D rendering in EMIT is shown in the following figure.



For details on the other fields in the property panel, see the General Parameters section.

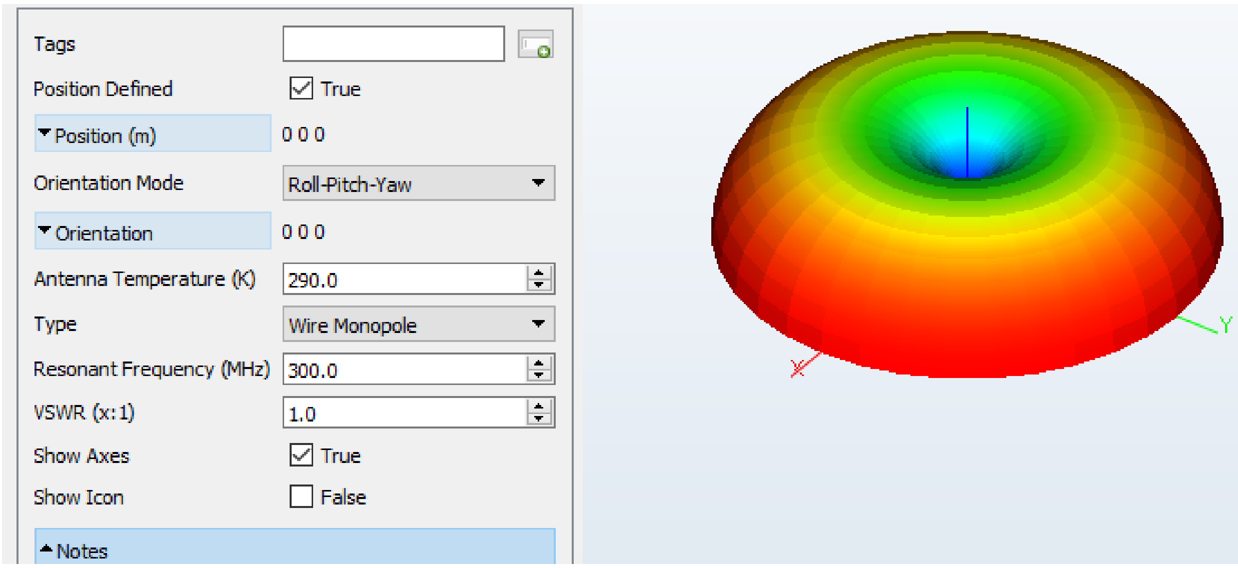
Wire Dipole: The Wire Dipole antenna is a general dipole antenna that is frequency dependent. It represents a physical half wave dipole antenna at the resonant frequency. The property panel for a Wire Dipole antenna and its 3D rendering (at ƛ/10, ƛ/2, and 3ƛ) in EMIT is shown in the following figure.



Resonant Frequency: The resonant frequency is used to define the length of the Wire Dipole. The length of the Wire Dipole is set to be a half wavelength at the resonant frequency.

For details on the other fields in the property panel, see the General Parameters section.

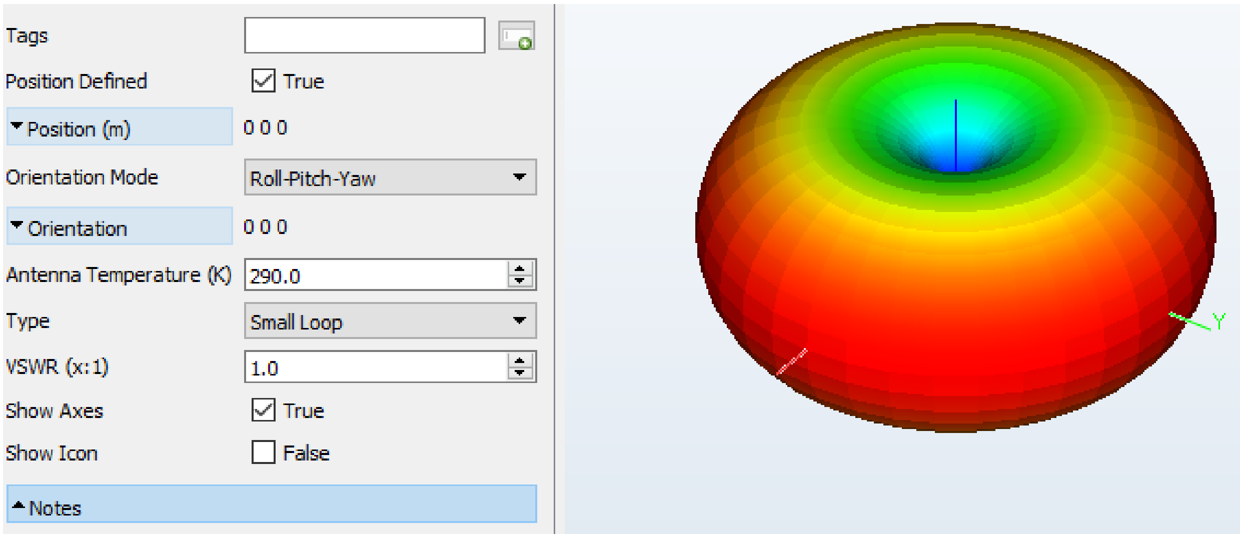
Wire Monopole: The Wire Monopole antenna is a general monopole antenna that is frequency dependent. It represents a physical quarter wave monopole antenna at the resonant frequency. The property panel for a Wire Monopole antenna and its 3D rendering (at ƛ/10, ƛ/4, and 3ƛ) in EMIT is shown in the following figure.



Resonant Frequency: The resonant frequency is used to define the length of the Wire Monopole. The length of the Wire Monopole is set to be a quarter wavelength at the resonant frequency.

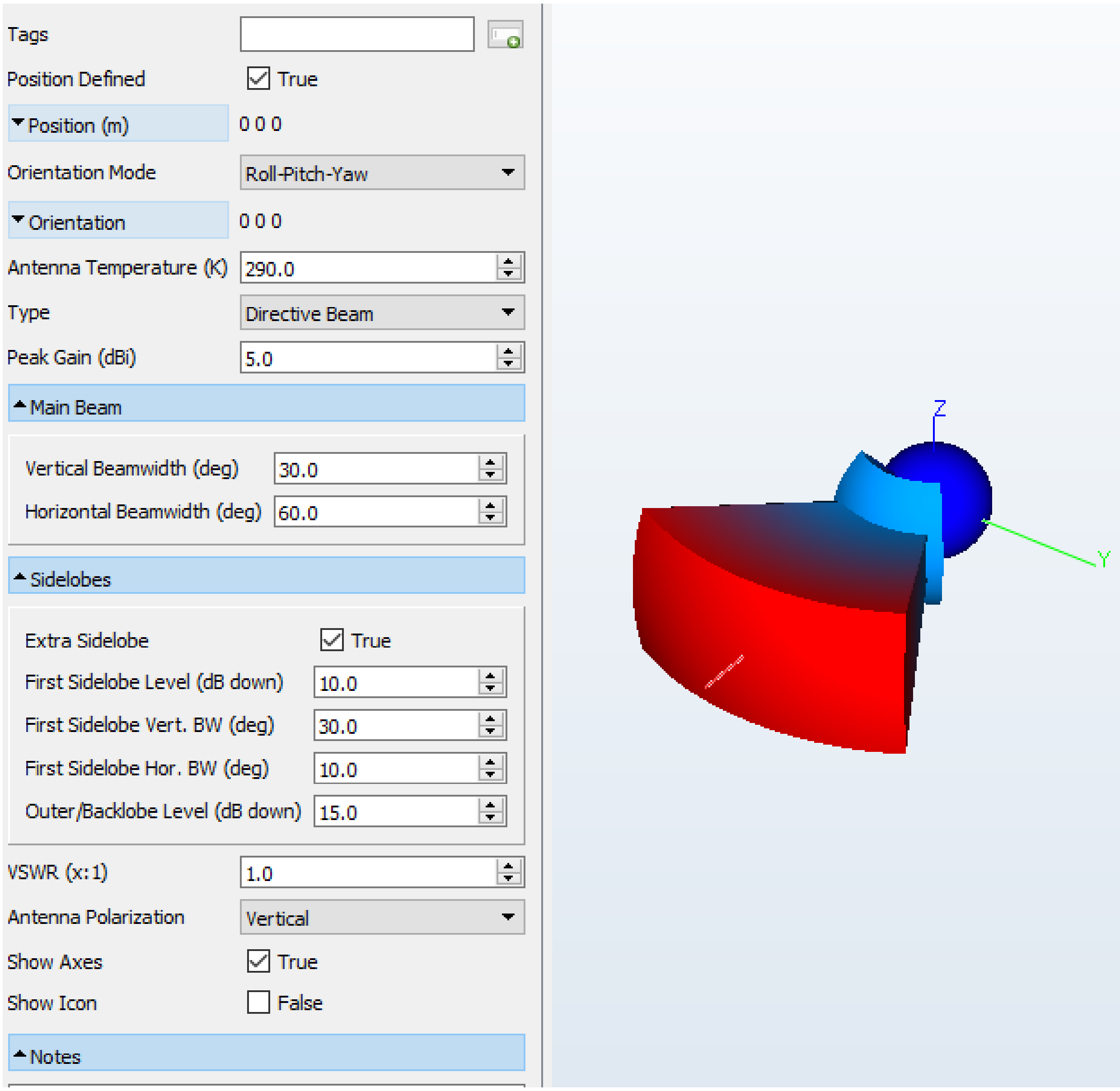
For details on the other fields in the property panel, see the General Parameters section.

Small Loop: The Small Loop antenna has the typical toroidal (doughnut-shaped) radiation pattern of a small loop antenna with a peak gain of 1.76 dBi and is constant for all frequencies. The Small Loop antenna is polarized in the phi direction. The pattern does not represent any physical (real) antenna, but is still quite useful for many types of analyses in EMIT. The property panel for a Small Loop antenna and its 3D rendering in EMIT is shown in the following figure.



For details on the other fields in the property panel, see the General Parameters section.

Directive Beam: The Directive Beam antenna defines a main beam with one or two sidelobe levels and an option for frequency roll-off. The pattern does not represent any physical (real) antenna, but is useful for many types of analyses in EMIT. For example, it provides a rough estimate of the 3D gain pattern of a parabolic dish or horn antenna. The property panel for a Directive Beam antenna and its 3D rendering in EMIT is shown in the following figure.



Peak Gain: The gain of the Directive Beam antenna's main beam.

Vertical Beamwidth (deg): The beamwidth of the main beam in the elevation plane.

Horizontal Beamwidth (deg): The beamwidth of the main beam in the azimuth plane.

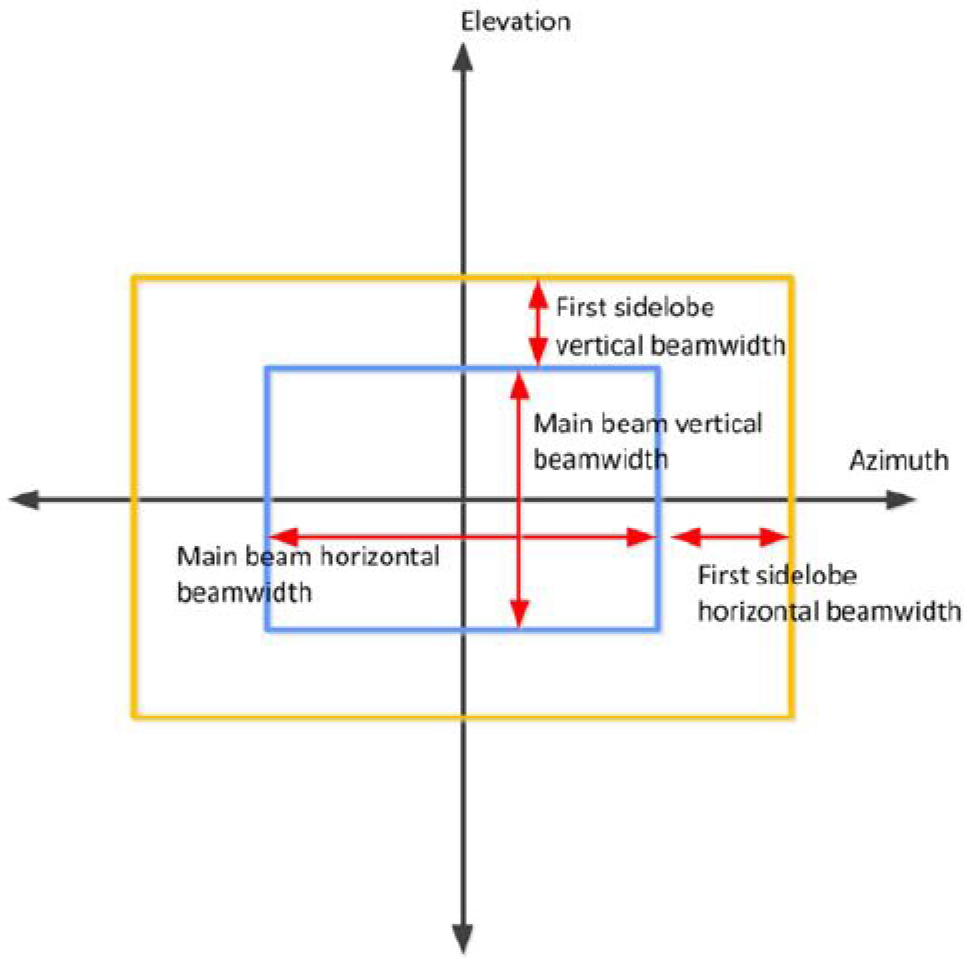
Extra Sidelobe: Toggle (on/off) the option to define two sidelobe levels.

First Sidelobe Level (dB down): The reduction in the gain of the Directive Beam antenna for the first sidelobe level. The value is entered relative to the peak gain.

First Sidelobe Vertical Beamwidth (deg): The beamwidth of the first sidelobe beam in the theta direction. If viewed in the azimuth/elevation (az/el) plane, the first sidelobe vertical beamwidth is defined as the additional beamwidth in the elevation plane.

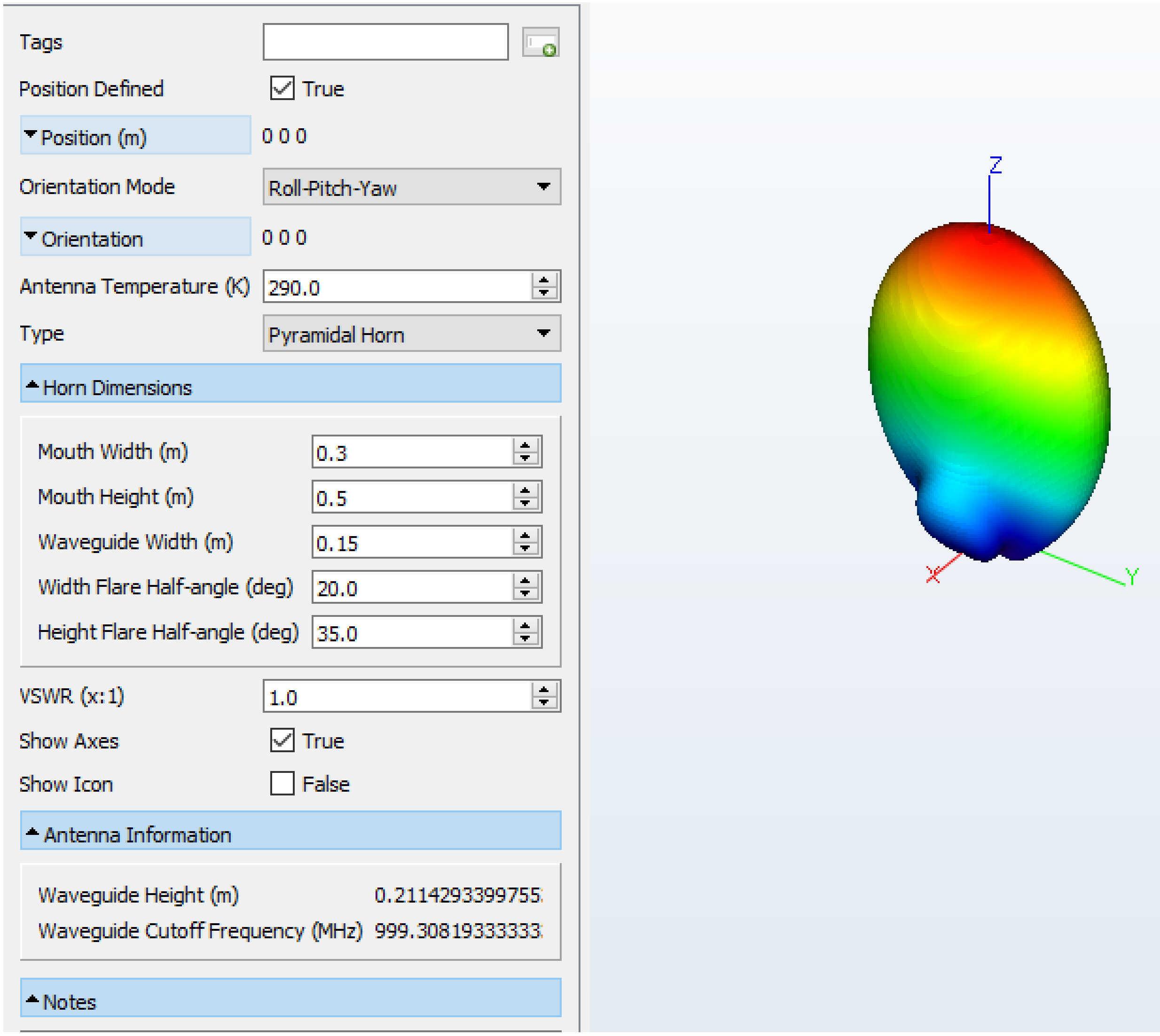
First Sidelobe Horizontal Beamwidth (deg): The beamwidth of the first sidelobe beam in the phi direction. If viewed in the azimuth/elevation (az/el) plane, the first sidelobe horizontal beamw idth is defined as the additional beamwidth in the azimuth plane.

Outer/Backlobe Level (dB down): The reduction in the gain of the Directive Beam antenna for the outer/backlobe level. The value is entered relative to the peak gain. All angles outside the main beam and first sidelobe beam will have a gain value equal to the backlobe level.



For details on the other fields in the property panel, see the General Parameters section.

Pyramidal Horn: The Pyramidal Horn antenna has the typical directional radiation pattern of a horn antenna. The gain of the pyramidal horn antenna varies with frequency. The property panel for a Small Loop antenna and its 3D rendering in EMIT is shown in the following figure.



Mouth Width: The mouth width of the Pyramidal Horn antenna.

Mouth Height: The mouth height of the Pyramidal Horn antenna.

Waveguide Width: The waveguide width (where the flared walls meet the feed) of the Pyramidal Horn antenna.

Width Flare Half-angle: The width flare half-angle (in degrees) of the Pyramidal Horn antenna as measured from the boresight axis to either wall.

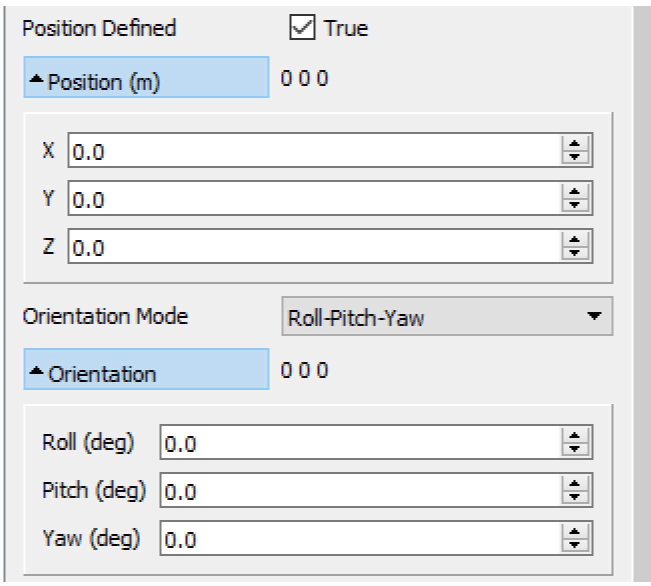
Height Flare Half-angle: The height flare half-angle (in degrees) of the Pyramidal Horn antenna as measured from the boresight axis to either wall.

For details on the other fields in the property panel, see the General Parameters section.

General Parameters:

VSWR: The Voltage Standing Wave Ratio (VSWR) due to the impedance mismatch between the antenna and the RF System (or outboard component). The VSWR field does not appear if the antenna type is By File.

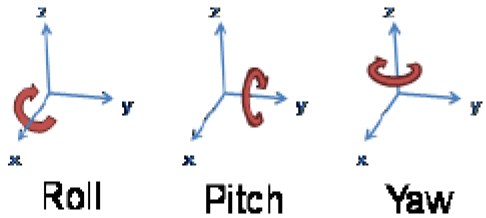
Position Defined: Toggle on/off display of Position and Orientation.



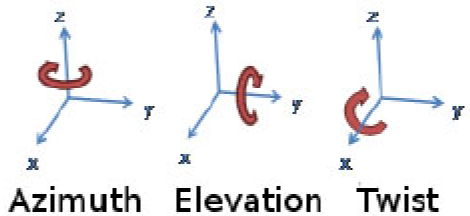
Position: In its collapsed state, the Position panel folder shows the current location (in meters) of the antenna in the scene with respect to the coordinate system of its parent node. Expand the Position panel folder to change the position. Alternatively, you can interactively place the antenna on surfaces of the platform CAD model using point-and-click operations. To do so, select the antenna node in the Project Tree and then press Ctrl+click anywhere on the CAD model. The selected antenna is automatically relocated to the selected point. After point-and-click placement, you should fine tune the position of the antenna by directly editing its position.

Orientation Mode: Select convention for specifying coordinate system orientation (order of rotations) When switching between orientation modes, EMIT converts the angle triplets from one convention to the other. You can switch the Orientation Mode at any time without disrupting the current orientation of the node.

- **Roll-Pitch-Yaw:** The order of rotations is Roll, then Pitch, then Yaw. Roll is always about the reference coordinates' x-axis, while Yaw is always about the current antenna coordinates' z-axis. The arrows represent the rotation of the local coordinate system relative to reference coordinates.



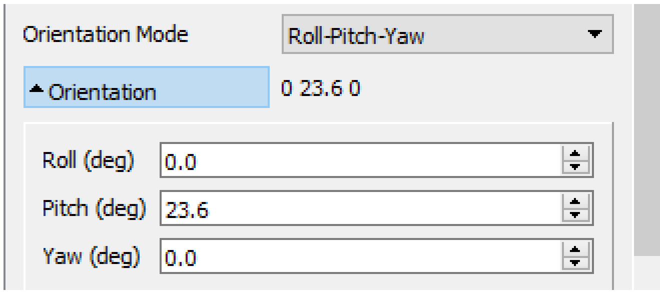
- **Azimuth-Elevation-Twist:** The order of rotations is Azimuth, then Elevation, then Twist. Azimuth is always about the reference coordinates' z-axis, while Twist is always about the current antenna coordinates' x-axis. The arrows represent the rotation of the local coordinate system relative to reference coordinates.



Rotation Definitions

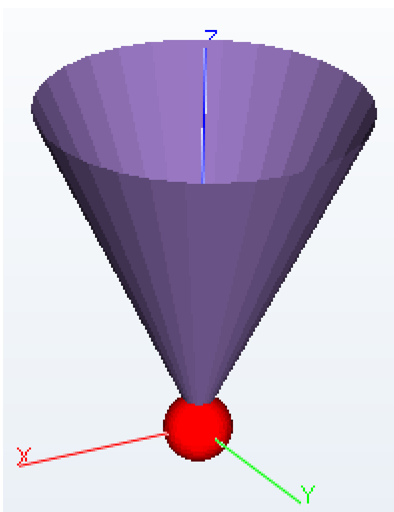
Name	Axis			Sense		Order Applied		
	+X	+Y	+Z	right-handed	left-handed	1st	2nd	3rd
Roll	•				•	•		
Pitch		•			•		•	
Yaw			•		•			•
Azimuth			•		•	•		
Elevation		•			•		•	
Twist	•				•			•

Orientation: Set the orientation of the Antenna node coordinate system .



The orientation of the node's coordinate system is uniquely defined by three rotation angles. These are either in Roll-Pitch-Yaw or Azimuth-Elevation-Twist convention, depending on the current selection of Orientation Mode.

- **Show Axes:** Toggle (on/off) display of the local antenna axes in the 3D window.
- **Show Icon:** Toggle (on/off) display of the antenna marker in the 3D window. Antennas are represented in the 3D window by a cone-shaped object (icon) as shown in the following example.

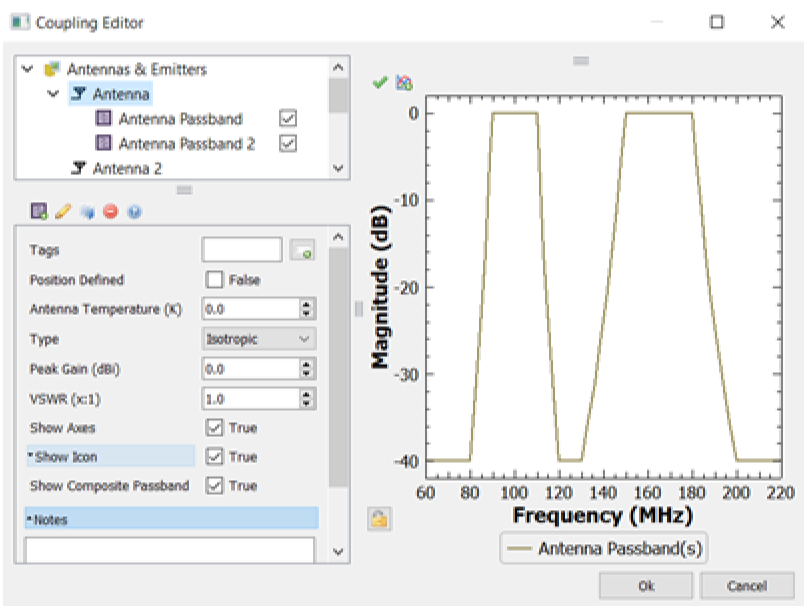


If an antenna pattern is defined for the antenna, it is displayed in place of the icon. **Show Icon** is a toggle panel folder that can be expanded when switched on.

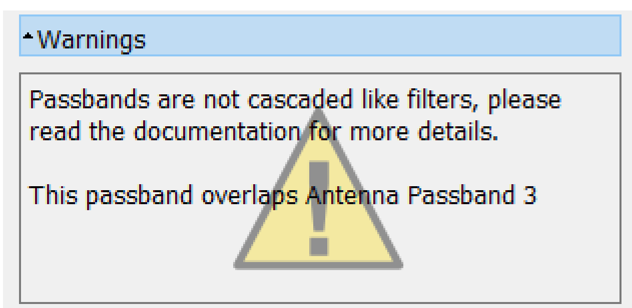
Size: Adjust the size of the cone

Color: Set the color of the cone Antenna Information: Expand this panel folder to see summary data about the antenna. It is available only when EMIT has information to present, as in the case of Type being set to By File.

Show Composite Passband: If an antenna has one or more passbands, the combined spectrum of the passbands can be displayed by selecting **Show Composite Passband**. The passbands must be non-overlapping and are combined by finding the maximum value at each frequency as shown below.



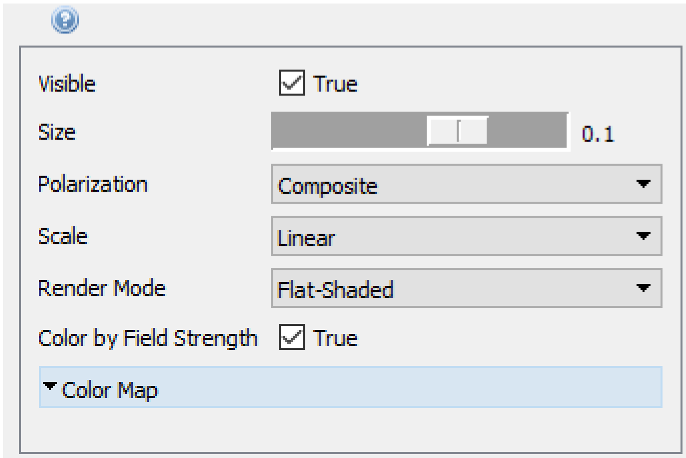
A node warning will be shown in the selected passbands properties as shown below.



Notes: The Notes field provides a text area for you to describe the antenna. The notes are stored with the project.

Pattern

Pattern nodes control the rendering of Antenna nodes as radiation patterns in a 3-D Window view node. In EMIT, only antennas that have a defined pattern have a Pattern node as a child.



Visible: Toggles on/off display of the radiation pattern in the 3D window.

Size: Sets the relative size of the antenna pattern in the 3D window.

The 3-D antenna pattern has no absolute scale. The scale is instead set relative to the overall size of the Scene node contents. Thus, the default size of 0.1 means that the antenna pattern and icon are rendered at roughly 10% of the Scene size.

This setting is controlled by a logarithmic slider. As you increase the slider value, the values increase more rapidly. The Size can be adjusted from 0.1% to 100% of the Scene size.

The size of the associated bounding box is determined by the maximum extension obtained from the Composite polarization. Selecting a different type of polarization does not modify the size of the bounding box. This can result in a bounding box that is much larger than the represented pattern.

Polarization: Sets the displayed polarization of the 3-D radiation pattern

There are several options for the Polarization:

- **Composite** - Root-mean-square (RMS) combination of the horizontal and vertical polarizations (this is the default)
- **Vertical** - θ-component in the antenna's local coordinate system
- **Horizontal** - φ-component in the antenna's local coordinate system
- **RHCP** - Right-hand circular polarization (IEEE convention) In the IEEE convention, RHCP is interpreted as the right thumb pointing in the direction of propagation (away from the transmitting antenna), with the fingers curling (from palm to tips) in the direction of the electric field's temporal advancement.
- **LHCP** - Left-hand circular polarization (IEEE convention), opposite of RHCP (see above)

Scale: Sets the radial scaling of the 3-D radiation pattern

This setting determines how the radius of the rendered surface relates to the selected far-field polarization level. There are several options for Scale:

- **Linear** - Rendered surface radius is proportional to the field magnitude
- **Power** - Rendered surface radius is proportional to the field magnitude square
- **Decibel** - Rendered surface radius is proportional to the field magnitude in decibels [dB]

When this scale option is selected, the dB Cutoff control becomes visible to control the dB level at the origin. This scale option is particularly useful when you need to observe weak sidelobes in the displayed radiation pattern.

dB Cutoff: Sets the dB-level at the origin relative to the peak level of the Composite polarization

This setting is only visible when the Scale is set to Decibel. For Decibel scaling, there is no natural choice for the field-level at the origin of the 3-D polar plot that is the 3-D radiation pattern - it must be picked arbitrarily. The default setting is -30 dB, which means that the field level at the origin is 30 dB down from the peak field level of the Composite polarization, not necessarily the currently selected display polarization.

Decreasing the dB Cutoff increases the range of field levels that are displayed in the 3-D radiation pattern. If you cannot see the sidelobes of your antenna (and you need to), decrease the dB Cutoff.

Render Mode: Controls how the radiation pattern surface is drawn

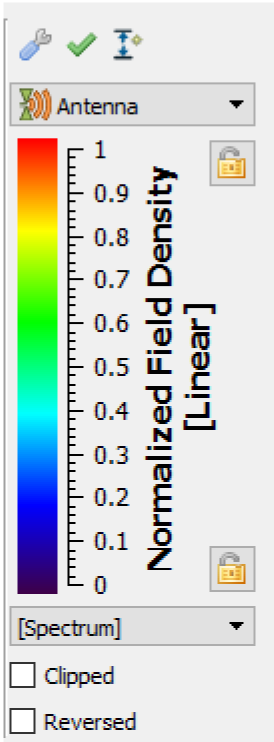
Rendering options are

- **Flat-Shaded** (solid surface, default)
- **Wire-Frame** (mesh)
- **Hidden Wire- Frame** (mesh with hidden-line removal)
- **Shaded Wire-Frame** (combination of Flat- Shaded and Wire-Frame)

Fill Color: Sets surface color used when **Render Mode** is set to Flat-Shaded or Shaded Wire-Frame and Color by Field Strength is switched off

Wire Color: Sets color of wire-frame mesh when **Render Mode i**s set to Wire-Frame or Hidden Wire- Frame.

Color by Field Strength: Switch on to display radiation pattern colored by field strength This setting is only visible when **Render Mode** is set to Flat-Shaded or Shaded Wire-Frame. When active, the color shading is scaled according to the selected Polarization and Scale, with the extents set in the color-bar map. To view or change these extents, right-click the pattern and click the Show Color Map icon . If there are multiple color bars in the project, click the combo box above the color bar, and select the item with the icon followed by the name of the antenna that owns the pattern.

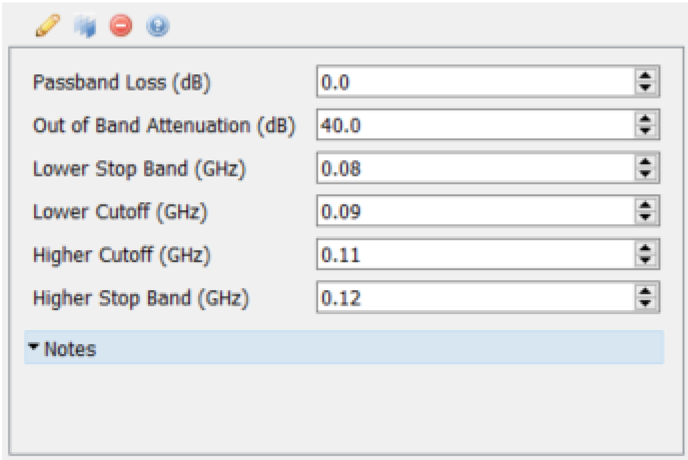


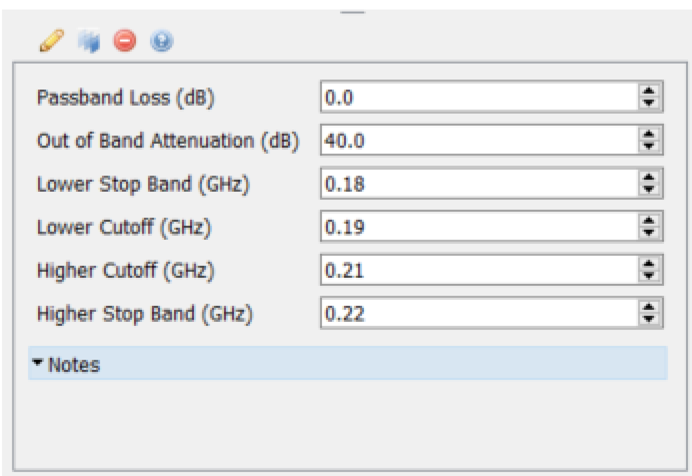
Color Map: Expand the Color Map panel folder to modify the properties of the color map. This folder becomes active if **Color by Field Strength** is True.

> **Minimum:** Set the lower limit of the color scale
>
> This setting can be saved as a node default only if the minimum color map extent is locked ()
>
> **Maximum:** Set the upper limit of the color scale
>
> This setting can be saved as a node default only if the minimum color map extent is locked ()
>
> **Axis Label Font:** Set the font for the axis label of the color bar
>
> **Axis Tick Label Font:** Set the font for the axis tick labels of the color bar

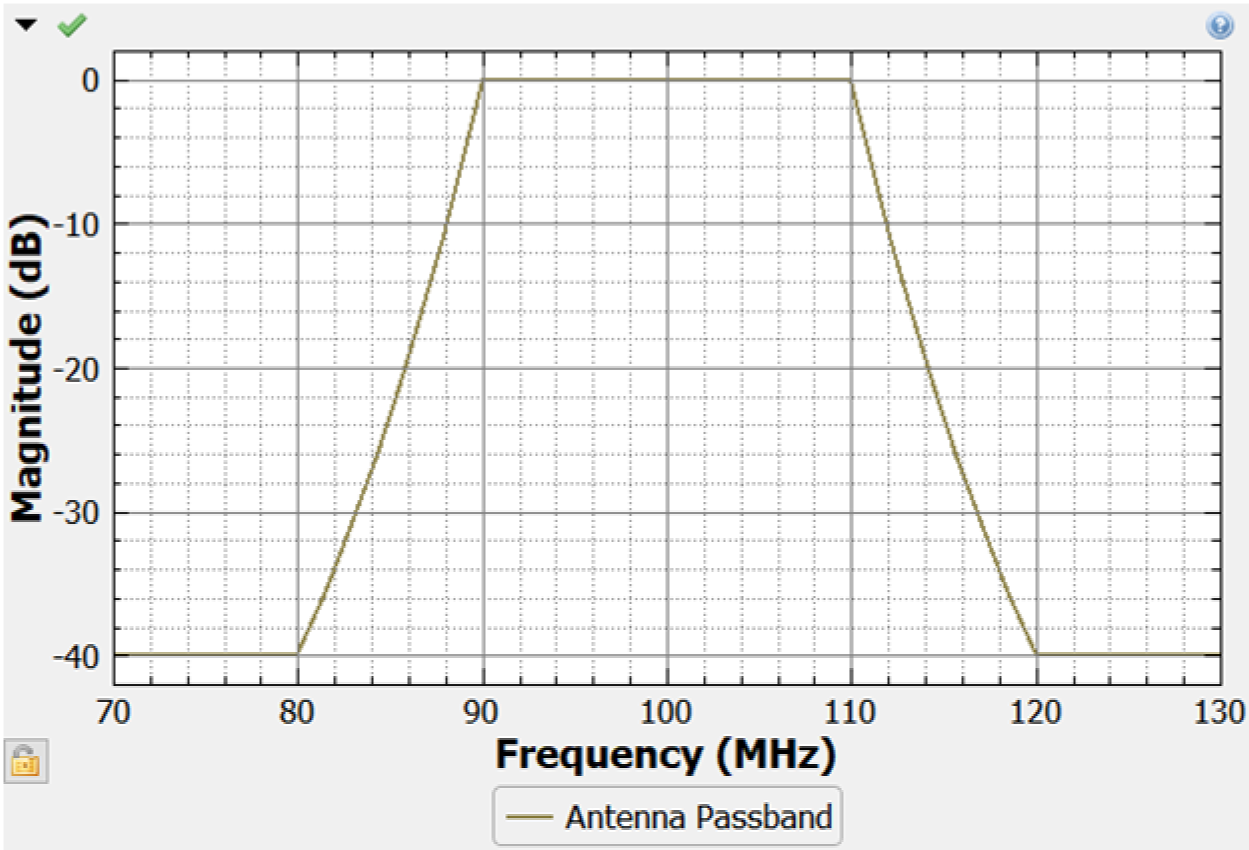
Antenna Passbands

Antenna Passbands are added to EMIT from the Coupling Editor. Antenna passbands are used to add frequency dependence to EMIT's parametric antenna models. They enable users to specify the operational frequency band(s) of an antenna. For example, an isotropic antenna with 0 dBi gain can be created in EMIT. This antenna initially has 0 dBi gain for all frequencies. By creating two passbands with the profiles below, this isotropic antenna can function as a dual band antenna.





The dual band isotropic antenna now has a peak gain of 0 dBi in each of the bands specified (90-110 MHz and 190-210 MHz). At frequencies outside of these two bands, the antenna has a gain of -40 dBi. In general, to determine the gain of an antenna with passbands at each angle for all frequencies, one needs to determine the gain at the desired theta/phi as determined by the antenna radiation pattern and then combine this gain with the magnitude of the passband at the desired frequency. To illustrate this, let's continue with the previous example of a 0 dBi isotropic antenna and determine its gain at 115 MHz and directly at the horizon (theta = 0 deg, phi = 0 deg). An isotropic antenna in EMIT has a constant gain at all angles and so in this specific case the gain at 0 degrees theta, 0 degrees phi is 0 dBi. In the passband plot below, one sees that at 115 MHz the passband has a magnitude of -20 dB. Therefore, this isotropic antenna would have a gain of -20 dBi at 115 MHz when viewed at the horizon.



Each Antenna Passband has the following properties defined in the Properties window:

Passband Loss [dB]: The "in-band" loss of the antenna passband. This is similar to the insertion loss of a filter. For multi-band antennas, the peak gain within each band often varies and this provides one way to model that variation.

Out of Band Attenuation [dB]: The attenuation of the passband at out-of-band frequencies (that is, below the Lower Stop Band and above the Higher Stop Band).

Lower Stop Band Frequency: The maximum frequency of the lower stop band. All frequencies below the lower stop band are attenuated by an amount equal to the out of band attenuation.

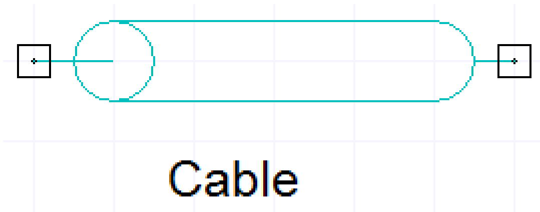
Lower Cutoff Frequency: The lower cutoff defines the minimum frequency of the antenna's passband. All frequencies between the lower cutoff and the higher cutoff are attenuated by an amount equal to the Passband Loss.

Higher Cutoff Frequency: The higher cutoff defines the maximum frequency of the antenna's passband. All frequencies between the lower cutoff and the higher cutoff are attenuated by an amount equal to the Passband Loss.

Higher Stop Band Frequency: The higher stop band defines the minimum frequency of the higher stop band. All frequencies greater than the higher stop band are attenuated by an amount equal to the out of band attenuation.

> **Note:**
> See the Plotting Composite Passbands information on the Antennas page if you want to plot multiple passbands in an overlay.

Cables



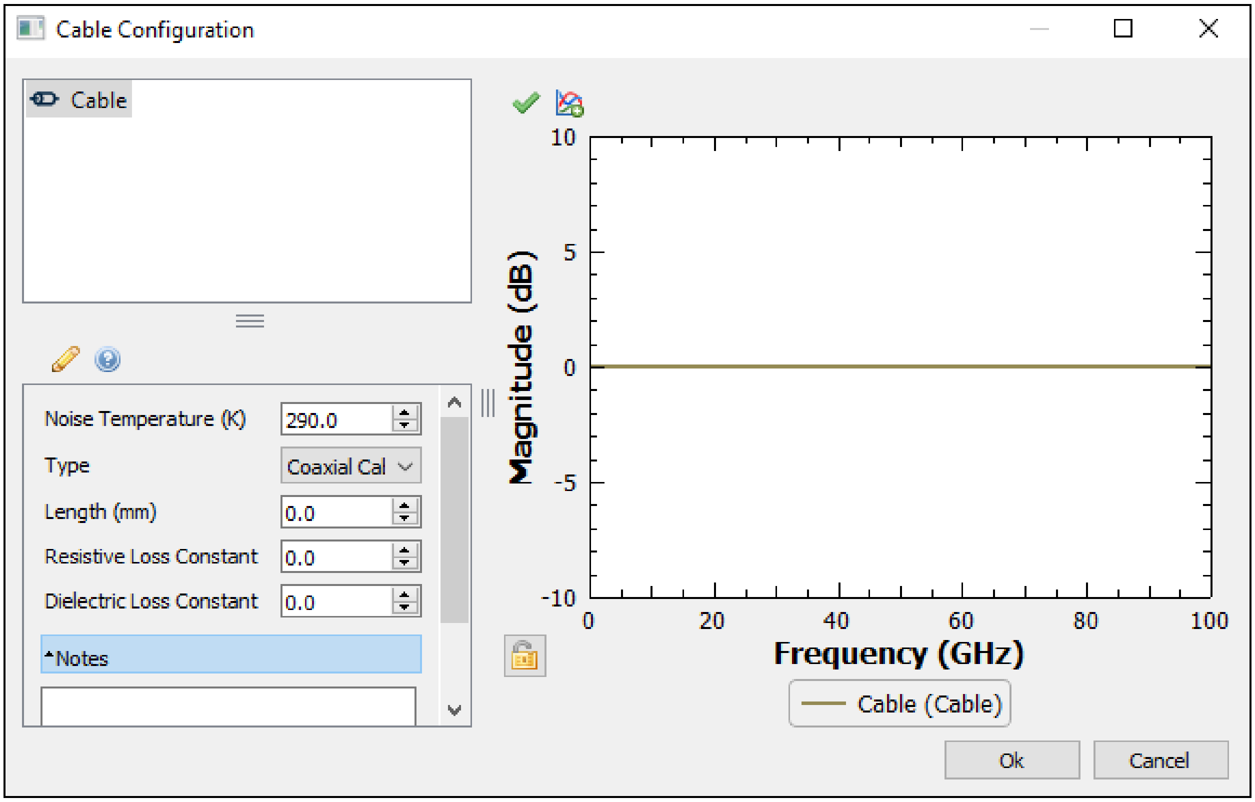
Cables are added to an EMIT design from the EMIT Elements component library.

A cable can be defined in three ways:

- By File.The cable is specified in EMIT by providing a data file with the cable's transmission characteristics. The file can be in the Ansys spectral profile format (.dlxsp) or it can be a Touchstone S-matrix file.
- Constant Loss
- Coaxial Cable, where the Coaxial Cable model calculates the attenuation per foot based on the resistive and dielectric loss constants as described below.

When computing the EMI margin, EMIT uses the cable's transmission characteristics in addition to accounting for both the mismatch loss and insertion loss according to:

$$P_{out}(dB) = P_{in} + 10log_{10}(|S_{21}|^2)$$



Noise Temperature (K): Noise power added by the cable.

In the **Type** drop-down, select one of the following:

- By File
- Constant Loss
- Coaxial Cable

By File

Specify the cable using a *.dlxsp or Touchstone file that contains the cable's transmission characteristics.

- **Filename:** Displays the filename being used for the cable characteristics. Clicking on the browse button permits choosing a new file.
- **Length:** The actual physical length of the cable desired in the design.
- **Measurement Length:** The length of the cable that was measured (or simulated). Note that a Measurement length that differs from the Length will scale the loss specified in the input file by:

$$Total\ Attenuation = File\ Loss\ *\ \left(\frac{Length}{Measurement\ Length}\right)$$

Constant Loss

Specify the cable with a constant attenuation per meter. The loss is constant over all frequencies. The total attenuation due to the cable is calculated by:

$$Total\ Attenuation = Length * (Loss\ per\ meter)$$

Length: The actual physical length of the cable desired in the design.

Loss Per Length: The constant attenuation per meter to use in determining the total attenuation of the cable. Note that the Loss per Length units are always defined as dB/meter.

Coaxial Cable

This cable model allows the user to specify the resistive (K1) and dielectric (K2) loss constants of the coaxial cable. EMIT then computes the attenuation per foot as a function of frequency. The attenuation per 100 feet for a specific frequency is given by:

$$Attenuation\ \left({}^{dB}/_{100\ ft}\right) = K1\sqrt{F} + K2 * F$$

where

$$F = Frequency\ in\ MHz$$
$$K1 = resistive\ loss\ constant$$
$$K2 = dielectric\ loss\ constant$$

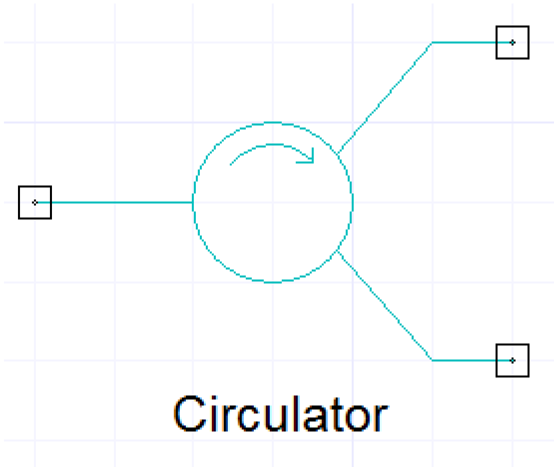
The figure below shows the attenuation as a function of frequency for LMR-400 coaxial cable. The figure is for a 100 ft length of cable.

Length: The actual physical length of the cable desired in the design.

Resistive Loss Constant (K1): The resistive loss constant for the cable as specified in *MIL-C-17 Attenuation and Power Handling Tables*.

Dielectric Loss Constant (K2): The dielectric loss constant for the cable as specified in *MIL-C-17 Attenuation and Power Handling Tables*.

Circulators

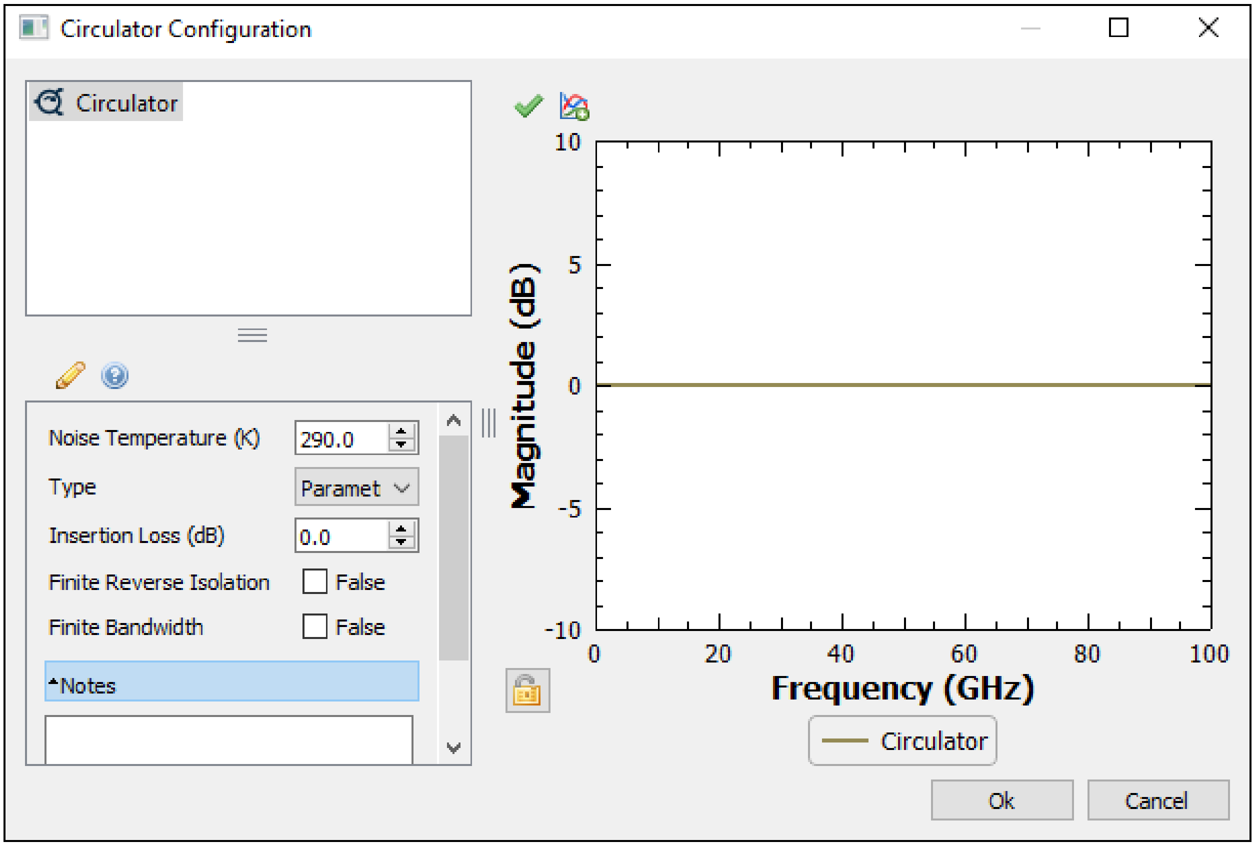


In EMIT a circulator is a passive non-reciprocal three-port device, in which a signal entering any port is transmitted to the next port in rotation (only). A signal applied to port 1 only comes out of port 2; a signal applied to port 2 only comes out of port 3; a signal applied to port 3 only comes out of port 1, so the scattering matrix for an ideal three-port circulator is

$$S = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Unless otherwise specified by the Finite Isolation or Finite Bandwidth settings, EMIT assumes ideal (infinite) isolation and infinite bandwidth for circulators.

(ref: http://en.wikipedia.org/wiki/Circulator)



Noise Temperature (K): Noise power added by the circulator.

In the **Type** drop-down, select one of the following:

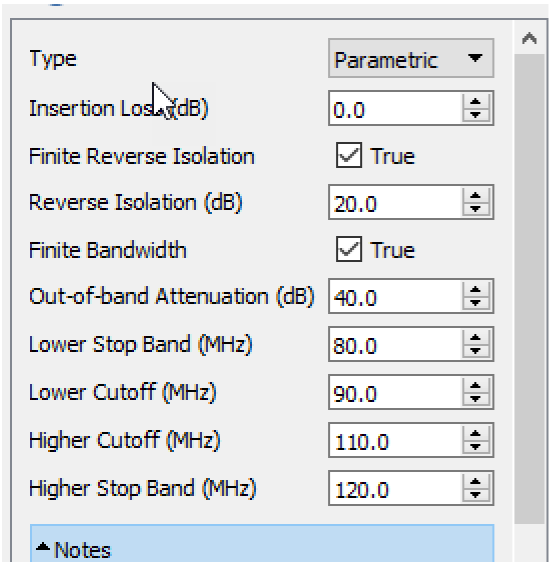
- By File
- Parametric

By File

Define the 3 port frequency characteristics of the Circulator by providing a 3-port Touchstone file containing the S-parameters for the device.

Parametric

Allows definition of the circulator characteristics by entering values for the appropriate parameters.

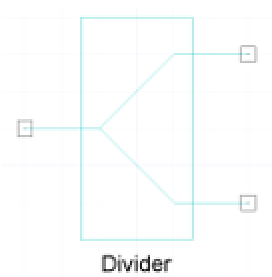


Insertion Loss (dB): Defines the insertion loss between circulating ports of the device. That is, the insertion loss from port 1 to 2, port 2 to 3 and port 3 to 1.

Finite Reverse Isolation: Isolation between the other, non-circulating, ports is assumed to be infinite unless **Finite Reverse Isolation** is enabled (True), in which case the value you provide is used.

Finite Bandwidth: By default, the bandwidth for a Circulator is considered to be infinite. When **Finite Bandwidth** is enabled (True) a finite bandwidth for the Circulator can be defined. The Out-of-Band Attenuation is then applied outside of the defined pass band as shown in the plot below. See Band Pass Filter for definitions of the parameters.

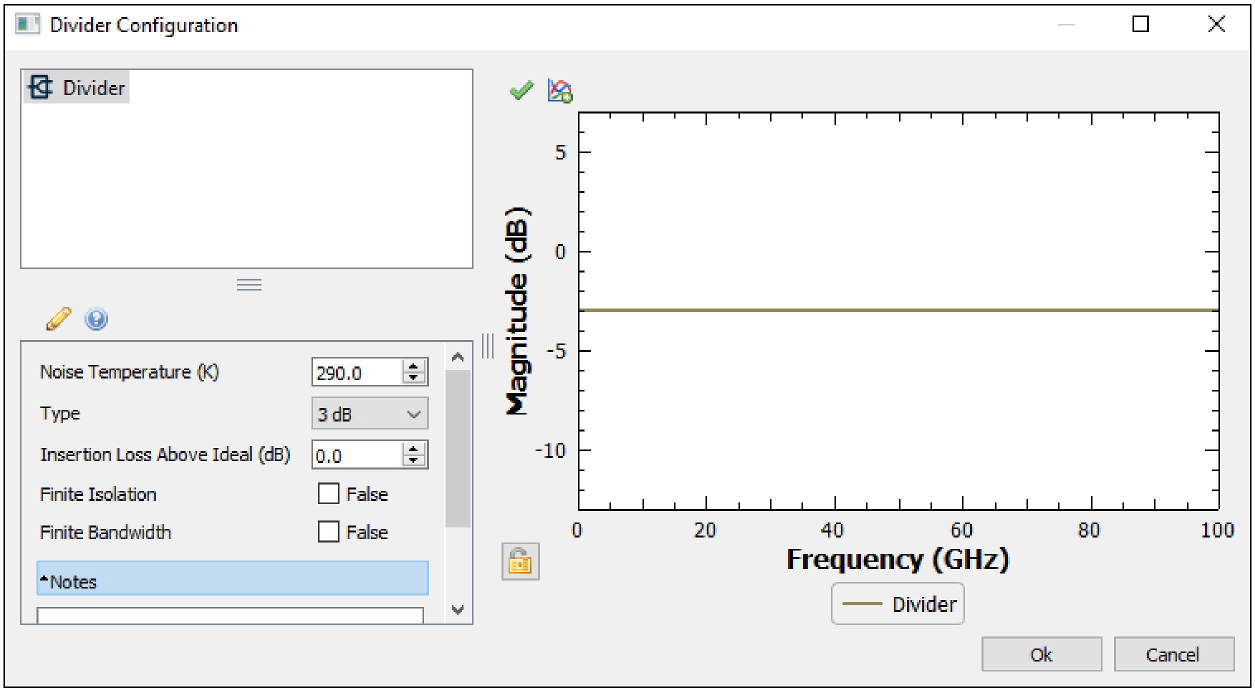
Dividers



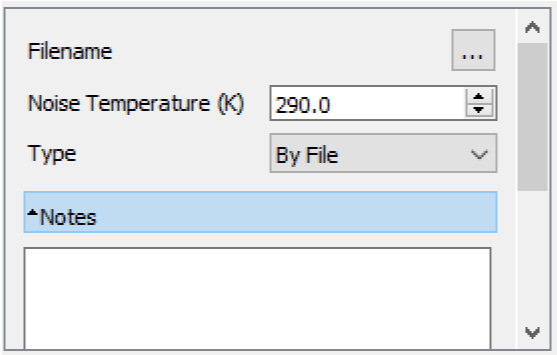
The Power Divider is a 3-port device that divides/combines signals (depending on the direction of signal flow) and is added to an EMIT design from the EMIT Elements components library.

There are three types of power dividers available in EMIT:

- By File
- 3-dB
- Resistive



By File

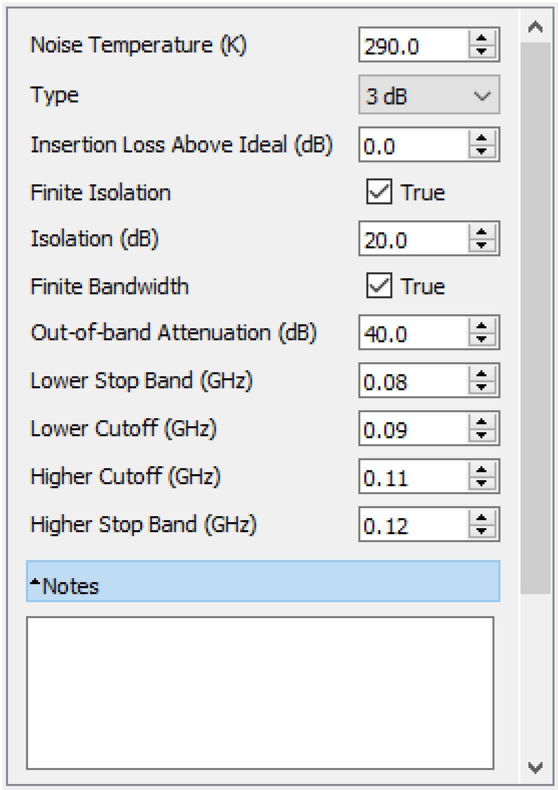


Filename: Define the 3-port frequency characteristics of the power divider by providing a 3-port Touchstone file containing the S-parameters for the device.

Noise Temperature (K): Noise power added by the divider.

3-dB

A 3-dB divider, or splitter, evenly divides the power at the input port between the two output ports.



Noise Temperature (K): Noise power added by the divider.

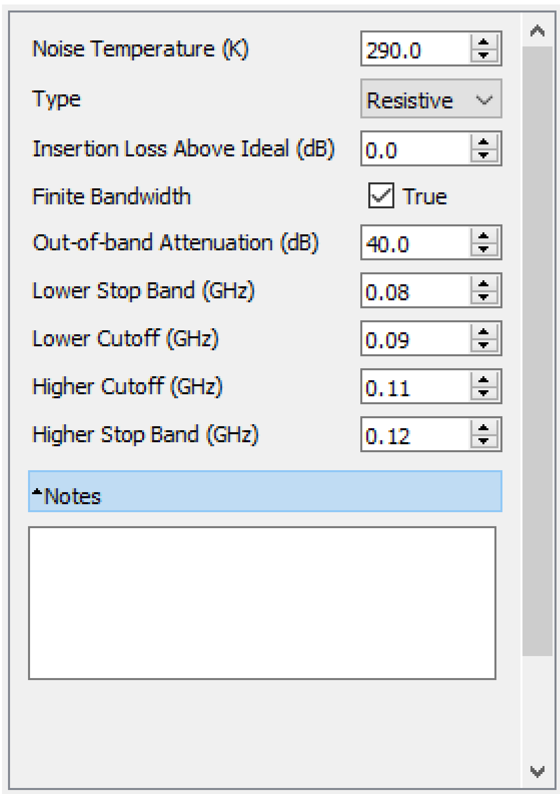
Insertion Loss Above Ideal: Provides additional insertion loss above the ideal for the Type chosen. For example, an ideal Resistive divider has 6 dB of attenuation (at all frequencies). Setting Insertion Loss Above Ideal to, say, 2 dB would lead to a total insertion loss of 8 dB for the divider. This provides a way of introducing additional attenuation in the divider beyond the ideal case.

Finite Isolation: The default model for the power divider is to assume infinite isolation between the 2-port side of the device. So, for example, in the COMBINER shown above, there would be infinite isolation between the two ports on the left-hand side of the device. When the Finite Isolation setting is enabled (True) a finite value can be entered that EMIT will use for the isolation.

Finite Bandwidth: By default, the bandwidth for a power divider is considered to be infinite. When Finite Bandwidth is enabled (True) a finite bandwidth for the Power Divider can be defined. The Out-of-Band Attenuation is then applied outside the defined pass band as shown in the plot below. See: Band Pass Filter for definitions of the parameters.

Resistive

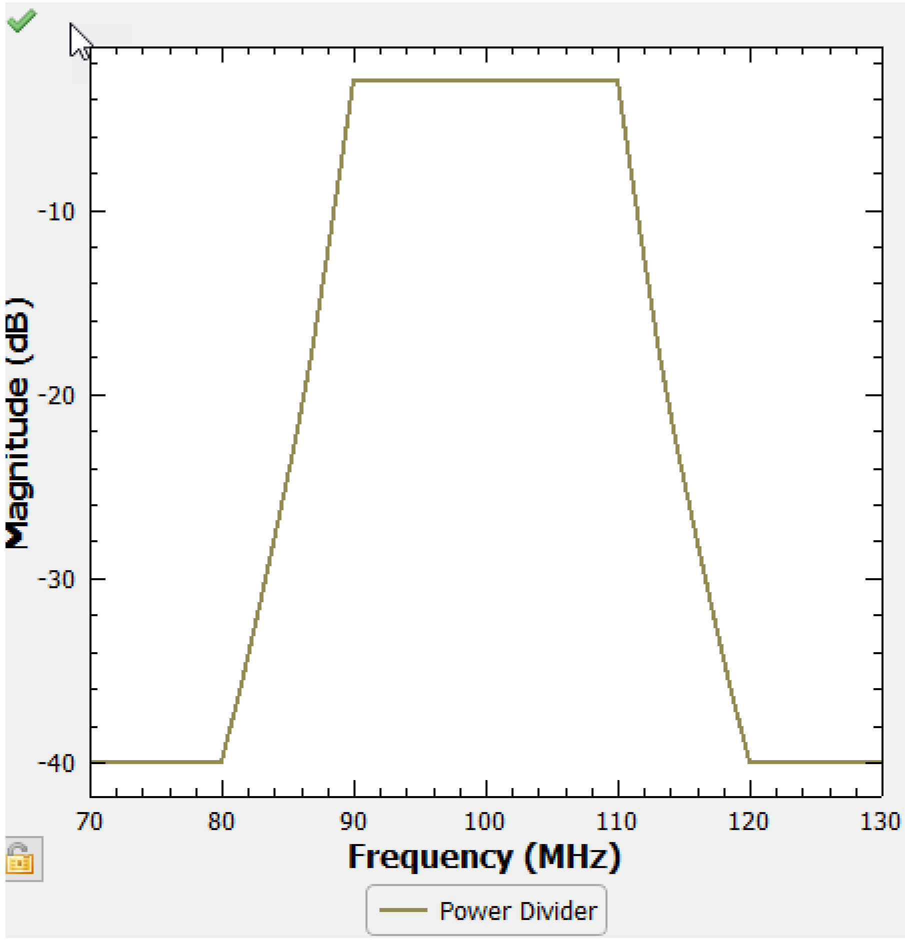
A lossy device, the resistive power divider presents signals at the output ports that are attenuated 6 dB from the input port.



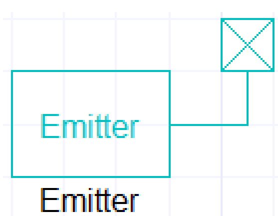
Noise Temperature (K): Noise power added by the divider.

Insertion Loss Above Ideal: Provides additional insertion loss above the ideal for the Type chosen. For example, an ideal Resistive divider has 6 dB of attenuation (at all frequencies). Setting Insertion Loss Above Ideal to, say, 2 dB would lead to a total insertion loss of 8 dB for the divider. This provides a way of introducing additional attenuation in the divider beyond the ideal case.

Finite Bandwidth: By default, the bandwidth for a power divider is considered to be infinite. When Finite Bandwidth is enabled (True) a finite bandwidth for the Power Divider can be defined. The Out-of-Band Attenuation is then applied outside of the defined pass band as shown in the plot below. See Band Pass Filter for definitions of the parameters.



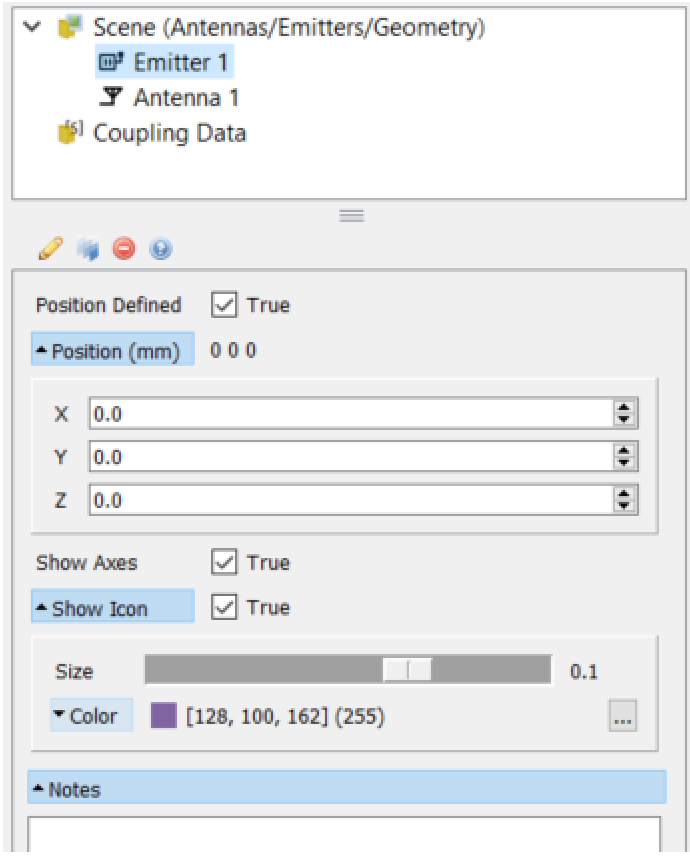
Emitters



An emitter in EMIT represents an unintentional source of radiated power. For example, a clock signal as the source on a PCB trace is modeled as an emitter in EMIT as it can cause interference to an RF system.

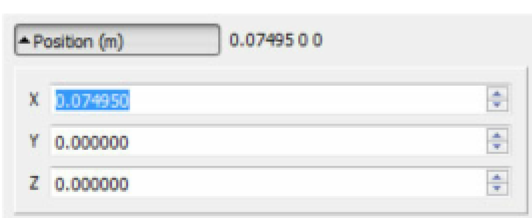
An emitter is specified in a similar way as a radio in EMIT using bands and Tx Spectral profiles. Emitters are added to an EMIT design via the main EMIT design schematic, and an emitter port is automatically included so no antenna can be attached.

Emitters couple to all antennas in the EMIT design, but not to each other. Emitters are assigned to ports in dynamically linked HFSS or HFSS 3D Layout designs to provide the necessary coupling data for the EMIT simulation.



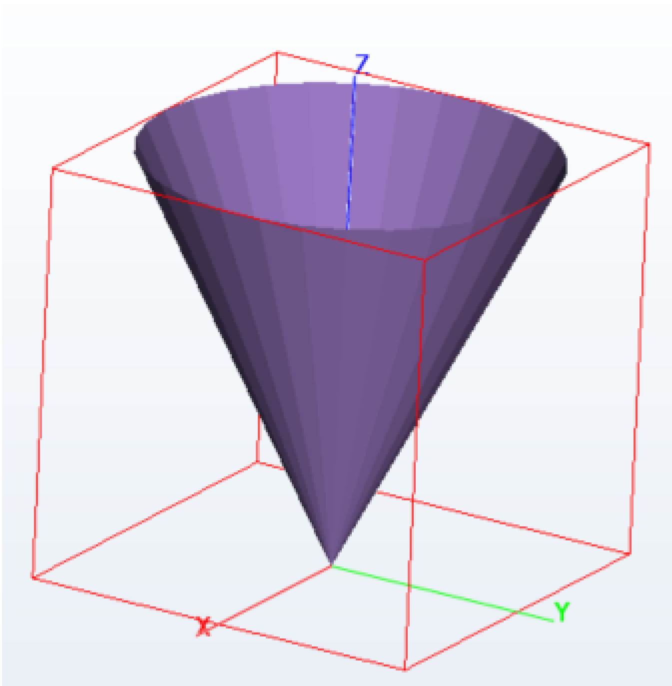
Position Defined: Toggle on/off the ability to define a position for an Emitter.

Position: Set the position (origin) of the Emitter node coordinate system. In its collapsed state, the Position panel folder shows the current location (in meters by default) of the Emitter in the scene with respect to the coordinate system of its parent node. Expand the Position panel folder to change the position. Alternatively, you can interactively place the Emitter on surfaces of the platform CAD model using point-and-click operation. It is recommended after point-and-click placement that you fine tune the position of the Emitter by directly editing its Position.



Show Axes: Toggle (on/off) display of the local emitter axes in the 3D window.

Show Icon: Toggle (on/off) display of the emitter marker in the 3D window. Emitters are represented in the 3D window by a cone-shaped object (icon) as shown in the following example.



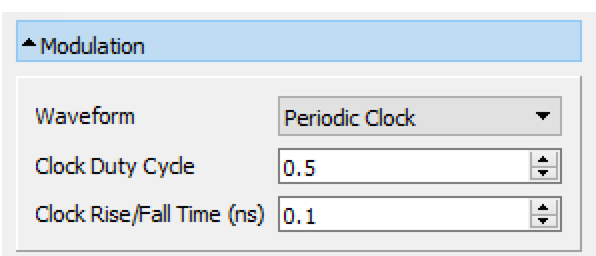
Size: Adjust the size of the cone

Color: Set the color of the cone.

Notes: The Notes field provides a text area for you to describe the emitter. The notes are stored with the project.

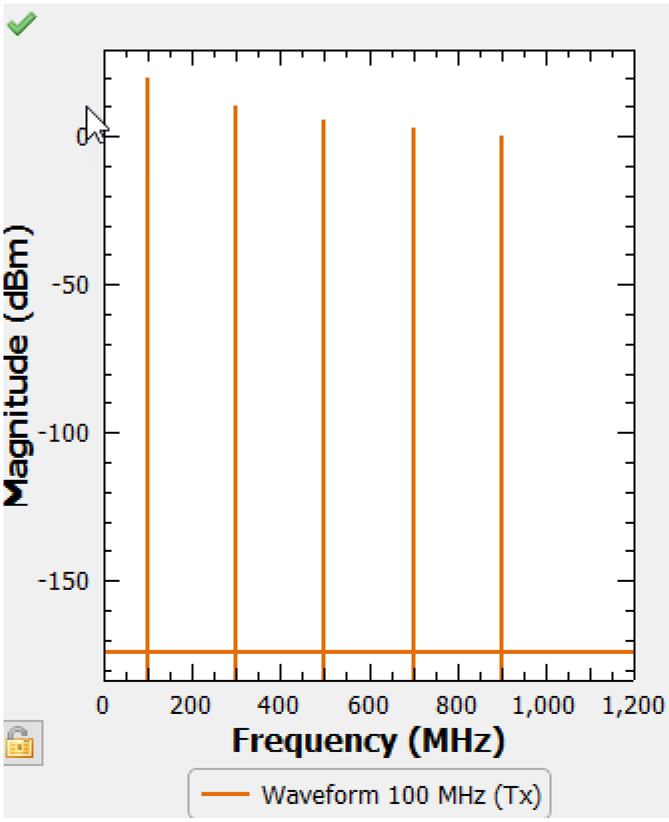
Periodic Clock

The spectrum consists of the fundamental clock frequency and its harmonics. If the Duty Cycle = 0.5, then only the odd harmonics are produced.



Clock Duty Cycle: Defines the clock's duty cycle. A Duty Cycle = 50% (0.5) will suppress the clock's even harmonics.

Clock Rise/Fall Time (ns): Specifies the rise/fall time of the clock pulses. The rise/fall time impacts the amplitudes of the clock harmonics.



Tx Spectral Profile for Periodic Clock

Rx Spectral Profile

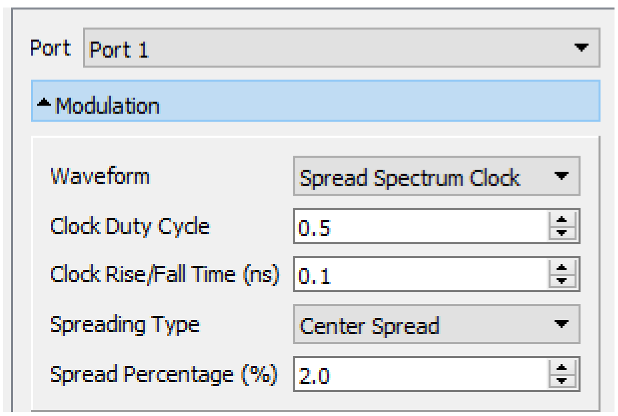
Spread Spectrum Clock

The Spread Spectrum Clock (SSC) differs from the Periodic Clock in a key way. It is represented in EMIT as a broadband power spectral density (PSD) rather than as a narrowband spectrum. This is due to the spreading nature of the clock. As seen below, the envelope of the spectrum is that of a sinc(x) wave.

> **Note:**
>
> Since the SSC signal type is modeled as a broadband PSD, plots of the spectrum are in units of dBm/Hz (despite "dBm" being displayed).

Spread Spectrum Clocking can significantly reduce the peak EMI levels radiated by a system clock. The reduction in the peak levels relative to the peak EMI of a non-spread clock is shown below. EMIT uses a Resolution Bandwidth of 1 Hz for all plots of the SSC spectrum.

$$EMI\ Reduction\ (dB) = 10 \log_{10}\left(\frac{S * f_c}{RBW}\right)$$
$$S = peak\ to\ peak\ spread\ percentage$$
$$f_c = carrier\ frequency$$
$$RBW = resolution\ bandwidth$$



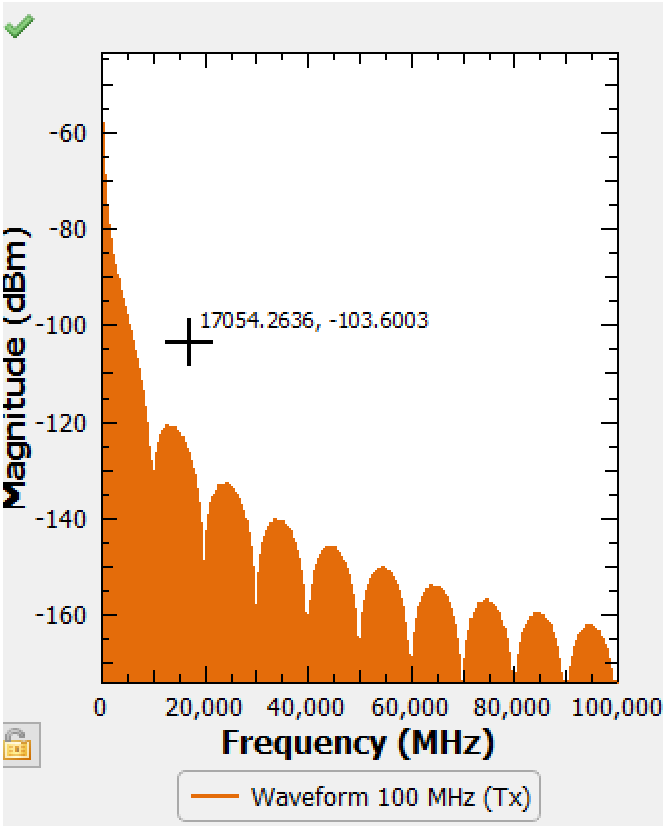
Clock Duty Cycle: Defines the clock's duty cycle. A Duty Cycle = 50% (0.5) will suppress the clock's even harmonics.

Clock Rise/Fall Time (ns): Specifies the rise/fall time of the clock pulses. The rise/fall time impacts the amplitudes of the clock harmonics.

Spreading Type: Specifies the method used to tune the clock frequency over. The tuning ranges, where BW = f_{clock}x(Spreading %), are specified as:

> **Low Spread:** f_{clock}-BW < f < f_{clock}
>
> **Center Spread:** f_{clock}-(BW)/2 < f < f_{clock} +(BW)/2
>
> **High Spread:** f_{clock}< f < f_{clock} +BW

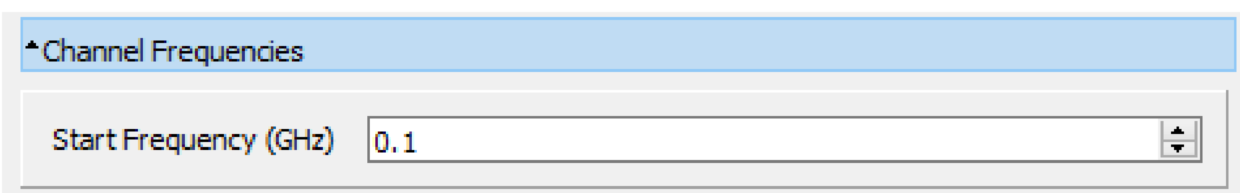
Spread Percentage (%): Determines the bandwidth over which the clock frequency is tuned.



[Tx Spectral Profile for Spread Spectrum Clock]

[Rx Spectral Profile]

Channel frequencies

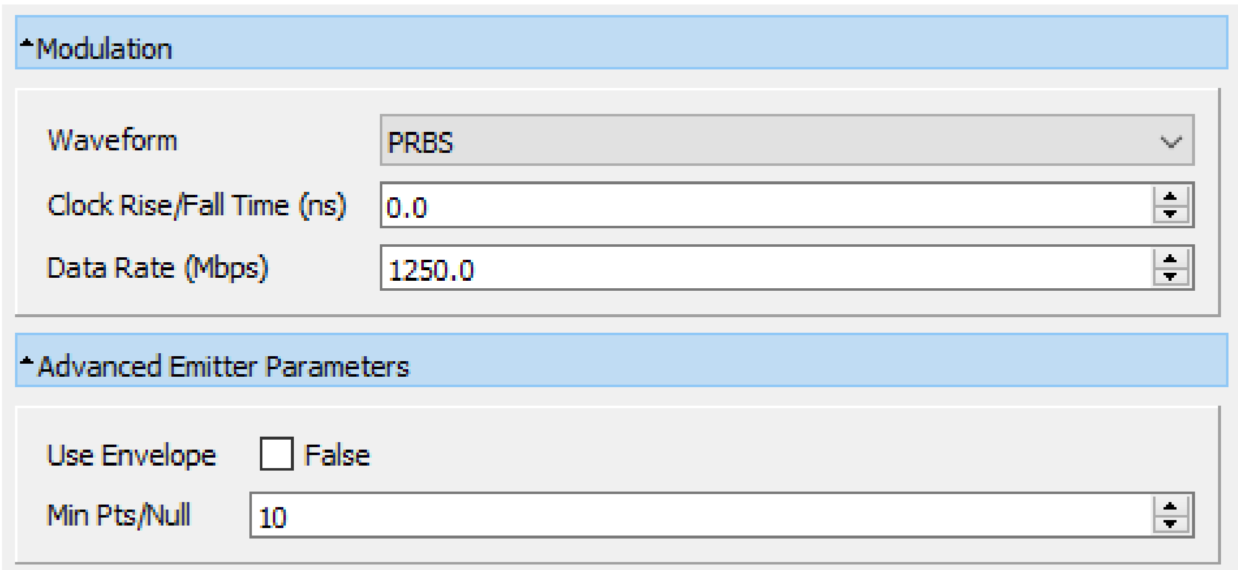


Start Frequency (GHz): The tuned channel (frequency) of the clock/oscillator.

PRBS

The Pseudo-Random Bit Sequence (PRBS) spectrum is represented in EMIT as a broadband power spectral density rather than as a narrowband spectrum and it defines the spectrum of a random bit stream. As seen below, the envelope of the spectrum is that of a sinc(x) wave. As with the SSC signal type, since the PRBS signal type is modeled as a broadband PSD, plots of the spectrum are in units of dBm/Hz (despite "dBm" being displayed).

> **Note:**
>
> Due to the specialized nature of the PRBS signal type, care must be taken when specifying the Channel Frequencies in the Configuration. For a PRBS, the Frequency Range should be set to 1 Hz in order to result in a valid Tx channel.



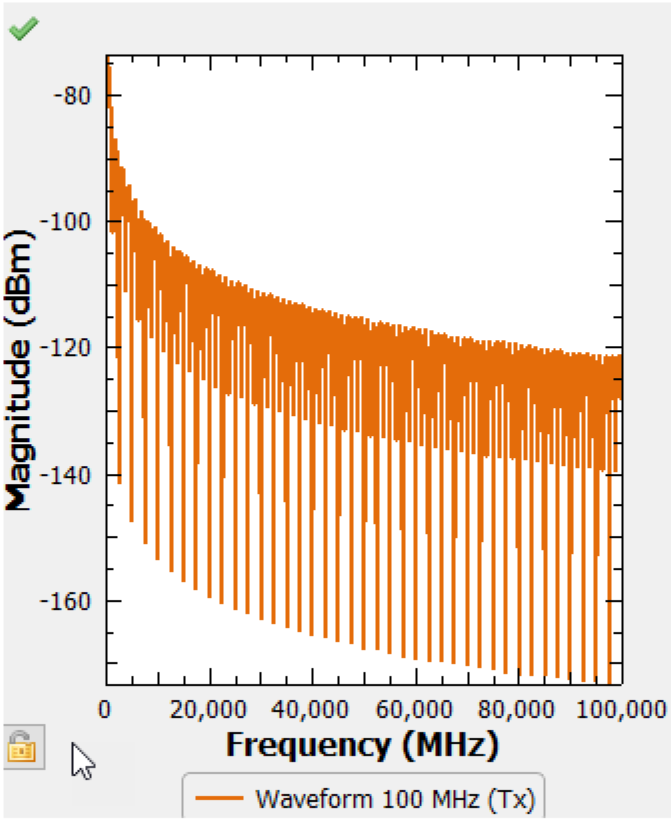
Clock Rise/Fall Time (ns): Specifies the rise/fall time of the clock pulses. The rise/fall time impacts the amplitudes of the spectral components.

Data Rate (Mbps): Number of bits transmitted per second. The data rate determines the peaks and nulls of the spectrum.

Use Envelope: If true, then the PRBS spectrum is modeled as a worst-case envelope which substantially reduces the number of points required and reduces simulation complexity. If false, the full spectrum is modeled with all peaks and nulls.

Min Pts/Null: Specifies a minimum number of points to use between each null of the computed spectrum. A greater number of points will provide a more accurate spectrum, but can also lead to increased simulation time. Note that EMIT uses three frequency ranges with different step sizes to reduce the number of points required for a smooth curve. This could result in some plot artifacts when zooming in.

Note that the peak voltage that specifies the clock drive voltage is entered under the Tx Spectral Profile settings.



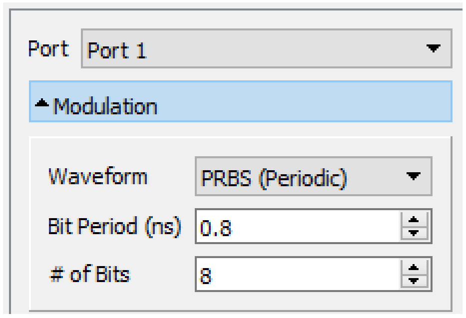
[Tx Spectral Profile for PRBS]

[Rx Spectral Profile]

PRBS (Periodic)

A narrowband representation of a repeating (periodic) Pseudo-Random Bit Stream. As seen below, the envelope of the spectrum is that of a sinc(x) wave.

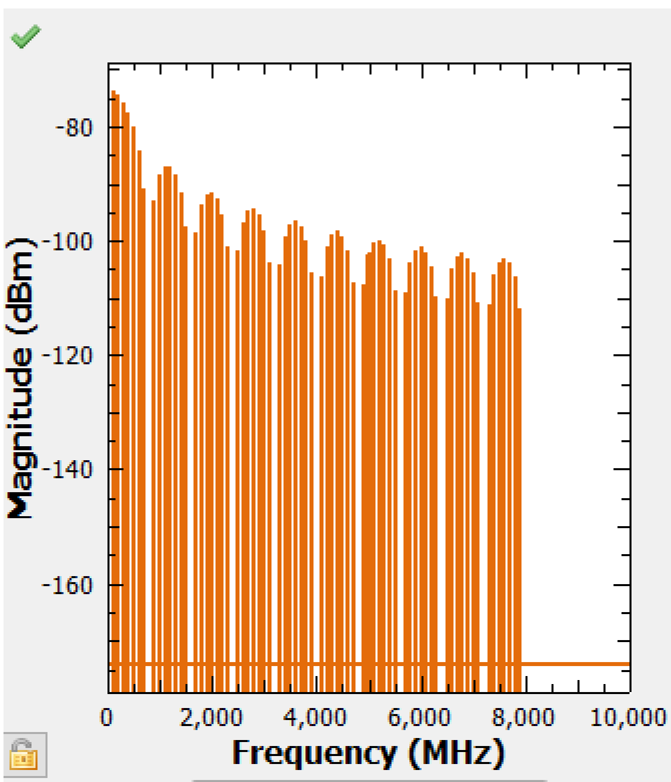
> **Note:**
>
> Due to the specialized nature of the PRBS (Periodic) signal type, care must be taken when specifying the Channel Frequencies in the Configuration. For a PRBS (Periodic) Channel Set, the Frequency Range must be set to (1/Bit Period)/(# of Bits) Hz in order to result in a valid Tx channel.



Bit Period (ns): Binary stream's bit period in nanoseconds.

# of Bits: Length of the binary sequence that is repeated.

Note that the peak voltage that specifies the clock drive voltage is entered under the Tx Spectral Profile settings

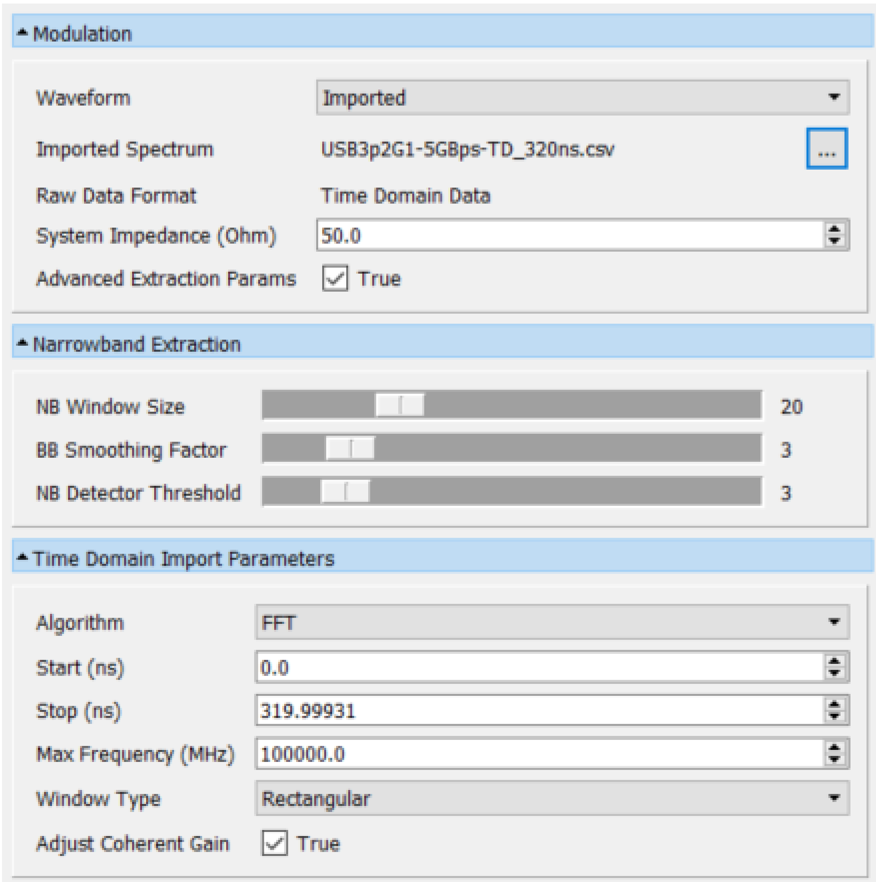


Tx Spectral Profile for PRBS (Periodic)

Rx Spectral Profile

Imported

Models an imported spectrum.



Imported Spectrum: Filename for the CSV file to import.

Raw Data Format: Read-only parameter specifying the format of the raw imported data (time or spectral domain).

System Impedance (Ohm): Measured (simulated) impedance of the system modeled by the imported file.

Advanced Extraction Params: Toggles on/off the Narrowband Extraction parameters. The default parameters are used when toggled off.

Narrowband Extraction

Narrowband Window Size: Number of frequency points to use when computing the local mean and standard deviation used for the narrowband component extraction.

Broadband Smoothing Factor: Factor by which the number of broadband frequency points is reduced. The local peaks are used to create an envelope of the raw imported data.

Narrowband Detector Threshold: Required number of standard deviations that a peak needs to be above the local mean to be characterized as a narrowband component.

Time Domain Import Parameters

Algorithm: Use either an FFT or Fourier Transform to convert the imported time domain data to the spectral domain.

The FFT provides Fourier Series (or Fourier Coefficients) of the transient signal with DFT, where periodicity is implicitly assumed and the period is defined as T0 = Stop - Start. The FFT algorithm re-samples the input waveform by interpolating to evenly spaced time points.

> **Note:** Based on the Nyquist-Shannon Theorem or Sampling Theorem, the sampling rate should be at least twice the bandwidth of the signal to avoid aliasing.
> Spectral plots based on the FFT are single-sided with magnitudes of non-zero harmonics doubled because the negative frequency components are folded to the positive side.

The Fourier Transform algorithm provides the Fourier transform of a signal defined in [Start, Stop] by leveraging the FFT algorithm. The Fourier transform of a time limited signal f(t) in [0,T0] is:

$$\mathrm{F}(f) = \int_{-\infty}^{\infty} \mathrm{f}(t) e^{-i2\pi f t} dt = \int_{0}^{T_0} f(t) e^{-i2\pi f t} dt$$

The Fourier series of a periodic signal that is equal to f(t) in [0,T0] and with period T0 is:

$$c_n = \frac{1}{T_0} \int_{0}^{T_0} f(t) e^{-i \frac{2\pi}{T_0} n t} dt$$

Hence,

$$c_n = \frac{1}{T_0} \mathrm{F}\left(\frac{n}{T_0}\right)$$

In other words, the Fourier series of f(t) based on FFT can be multiplied by T0 in order to get the Fourier Transform of f(t) at sampled frequencies n/T0.

Start (ns): Initial time of the data to perform the FFT over.

Stop (ns): Final time of the data to perform the FFT over.

Max Frequency (MHz): Highest frequency for which to compute the FFT. By default Max Frequency = (number of time pts)*(frequency resolution).

Window Type: Windowing function to apply to the FFT.

Kaiser Parameter: Determines the shape of the window via a trade off between the main-lobe width and side lobe level.

Adjust Coherent Gain: Adjusts the windowed signal by the coherent power gain.

File Format

The file can contain data in either the frequency or time domain. For frequency domain data, the CSV should contain 2 columns: frequency, amplitude. For time domain data, the CSV should contain 2 columns: time, voltage. The units for each column should be defined in a comment line at the top of the file with each column's information in quotation marks and the units in square brackets []: "Time [ns]", "V(signal_name) [mV]".

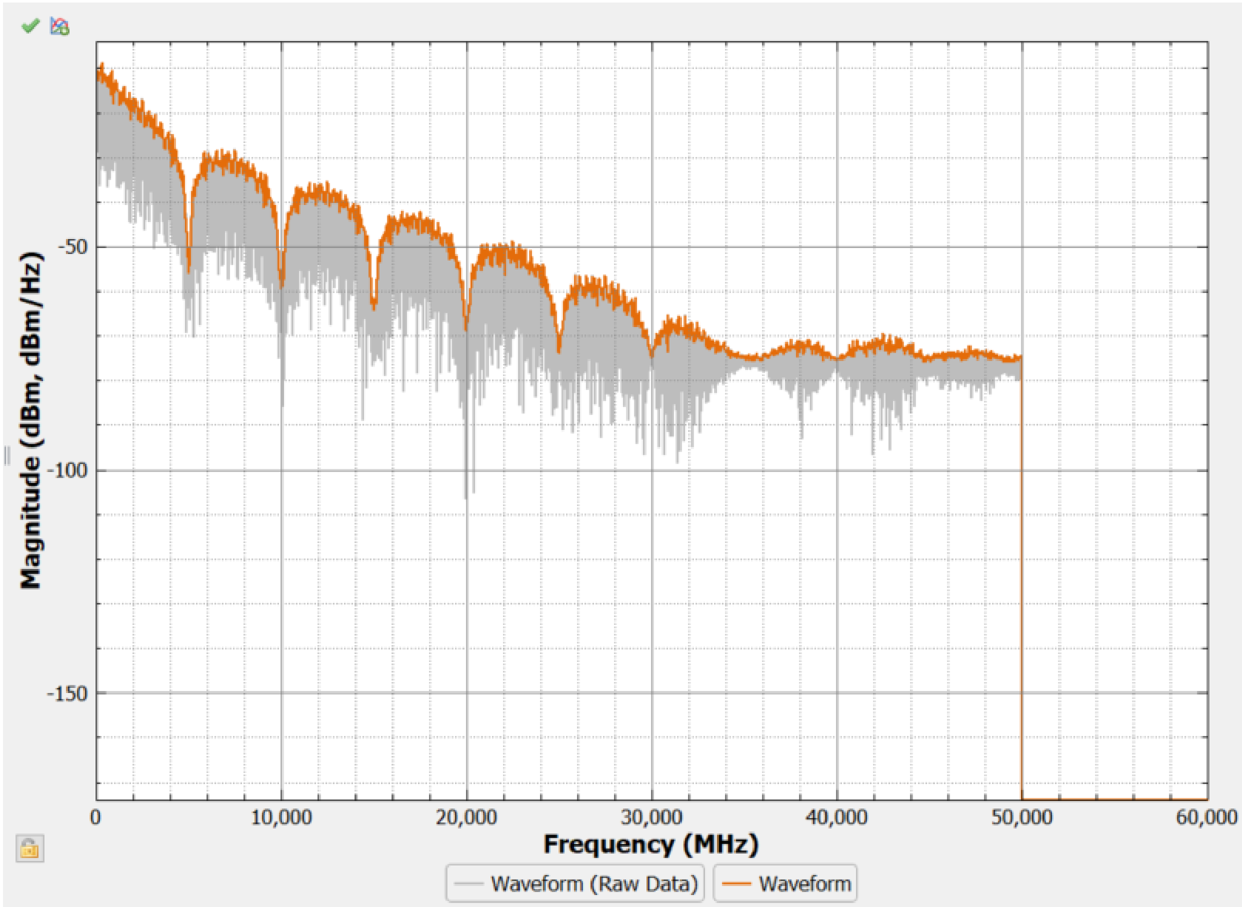
Time Domain files must have two columns. The first row of the file must specify the units with column one specifying the time units and column two specifying the voltage units.

- Time Units: ps, ns, us, ms, s
- Voltage Units: nv, uv, mv, v, kv, megv

Spectral Domain files must have two columns. The first row of the file must specify the units with column one specifying the frequency units and column two specifying the power units.

- Frequency Units: Hz, kHz, MHz, GHz, THz
- Spectral files must have units of dBV for the power

Comment lines: Any lines encapsulated in "" will be skipped by the importer.

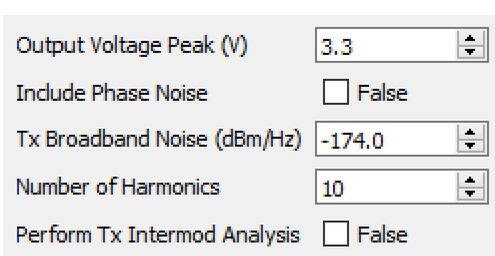


Tx Spectral Profile

The following are the Tx Spectral Profiles for waveform modulations for Emitter:

- Periodic Clock
- Spread Spectrum Clock
- PRBS
- PRBS (Periodic)

Periodic Clock



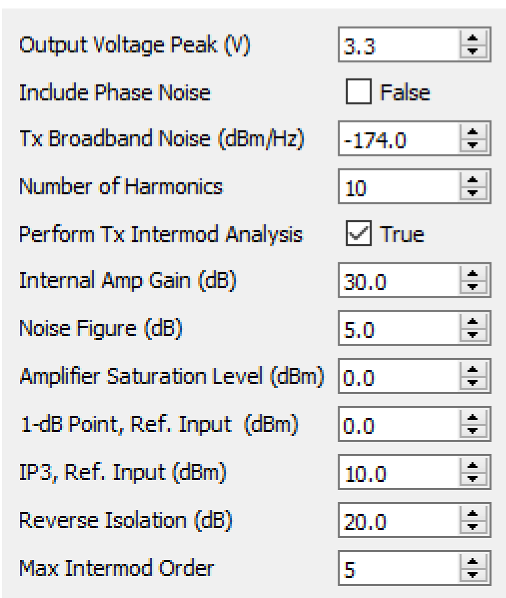
Output Voltage Peak (V): The peak power.

Include Phase Noise: Enable/disable phase noise in the broadband noise model of the transmitter spectrum. The phase noise decreases by 20 dB/decade.

Tx Broadband Noise (dBm/Hz): The amplitude of the radio's broadband power density spectrum. Tx broadband noise levels are specified in terms of dBm/Hertz. The average power level of the broadband noise should be input here. EMIT then computes the peak broadband noise level within a receiver's tuned channel when calculating the EMI Margin. The minimum broadband noise in EMIT is limited to the thermal noise floor of a 50-ohm system at room temperature which is -174dBm/Hz. Values smaller than that cannot be entered here.

Number of Harmonics: The number of harmonics (including the fundamental) included in the Tx Spectral Profile computed by EMIT. Since this includes the fundamental, the number of harmonics in the spectral profile is one less than this setting. Note that by this convention, the Nth harmonic has harmonic order (N+1).

The following shows the additional options for Periodic Clock modulation when **Perform Tx Intermod Analysis** is set to **True**.



Internal Amp Gain (db): Define the in-band gain of the amplifier in dB.

Noise Figure (dB): The amplifier's noise figure in dB. The noise figure is used to calculate the broadband noise added by the amplifier

Amplifier Saturation Level (dBm): Specifies an input power level for which EMIT will consider the amplifier saturated. When an amplifier is saturated, EMIT flags it as such in the results and does not trace the input signal any further. The saturation level cannot be set lower than the 1-dB Point.

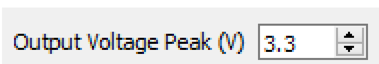
1-dB Point Ref. Input (dBm): The amplifier's 1-dB compression point referred to the input of the amplifier.

IP3, Ref. Input (dBm): The amplifier's third order intercept point referred to the input of the amplifier.

Reverse Isolation (dB): This is the amplifier's reverse isolation. The reverse isolation of an amplifier is a measure of how well a signal applied to the amplifier's output is "isolated" from its input. It is equivalent to the S12 S-parameter.

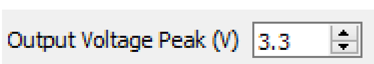
Max Intermod Order: Specifies the highest order of intermodulation product that EMIT will calculate for the amplifier.

Spread Spectrum Clock



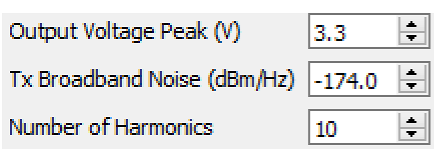
Output Voltage Peak (V): The peak power.

PRBS



Output Voltage Peak (V): The peak power.

PRBS (Periodic)



Output Voltage Peak (V): The peak power.

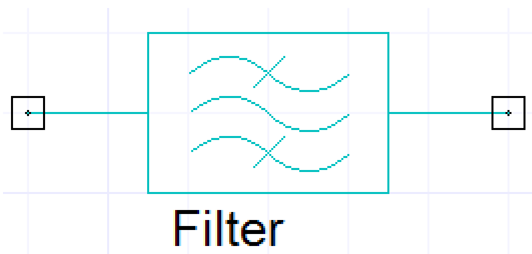
Tx Broadband Noise (dBm/Hz): The amplitude of the radio's broadband power density spectrum. Tx broadband noise levels are specified in terms of dBm/Hertz. The average power level of the broadband noise should be input here. EMIT then computes the peak broadband noise level within a receiver's tuned channel when calculating the EMI Margin. The minimum broadband noise in EMIT is limited to the thermal noise floor of a 50-ohm system at room temperature which is -174dBm/Hz. Values smaller than that cannot be entered here.

Number of Harmonics: The number of harmonics (including the fundamental) included in the Tx Spectral Profile computed by EMIT. Since this includes the fundamental, the number of harmonics in the spectral profile is one less than this setting. Note that by this convention, the Nth harmonic has harmonic order (N+1).

Rx Spectral Profile

Emitters are sources of interference and do no receive any signals. Therefore, no Rx Spectral Profile is required and is disabled in the Emitter configuration window.

Filters



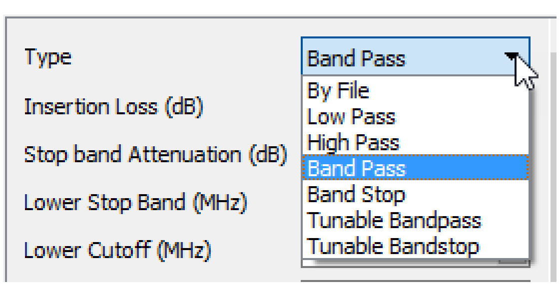
Filters are added to an EMIT design from the EMIT Elements component library. EMIT provides two methods of specifying the filter:

- Using one of the built-in parametric models
- Providing a data file

EMIT's simulation uses the filter transmission characteristics when computing the EMI margin. For the *By File* filter type, the filter is specified in EMIT by providing a data file with the filter's transmission characteristics. The file can be in the Ansys spectral profile format (.dlxsp) or it can be a Touchstone S-matrix file. When computing the EMI margin, EMIT uses the filter's transmission characteristics in addition to accounting for both the mismatch loss and insertion loss according to:

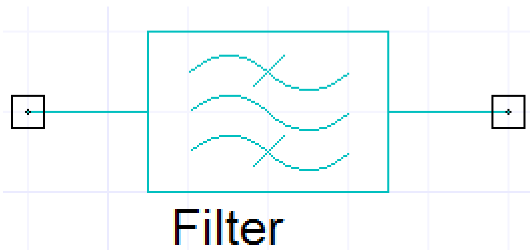
$$P_{out}(dB) = P_{in} + 10log_{10}(|S_{21}|^2)$$

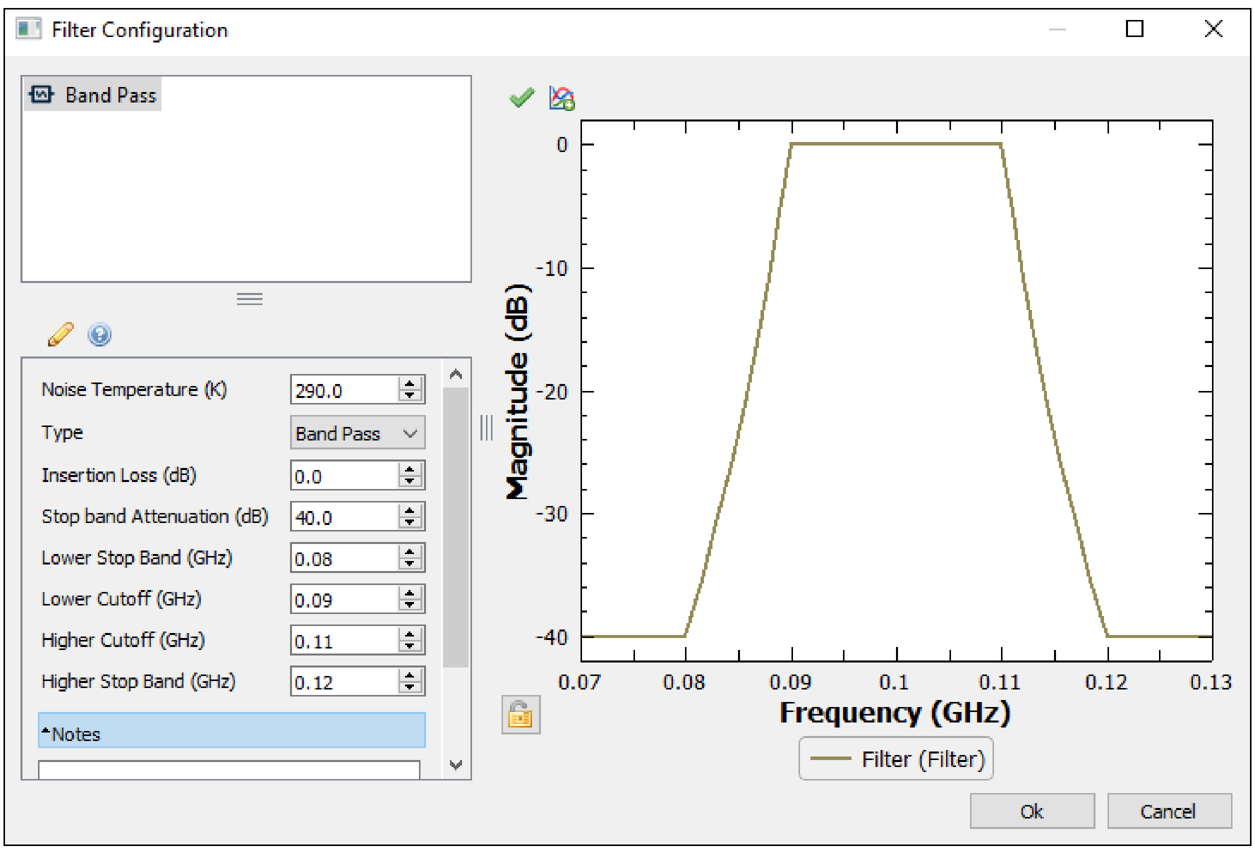
You select the filter from the **Type** drop-down:



- By File
- Low Pass
- High Pass
- Band Pass
- Band Stop
- Tunable Band Pass
- Tunable Band Stop

Band Pass





Defines a band pass filter with the parameters shown in the configuration window above. A band pass filter passes frequencies in the pass band with minimal attenuation and rejects (attenuates) signals outside of the pass band. A band pass filter will attenuate both low and high frequencies as seen below in the plot of a band pass filter in EMIT.

Noise Temperature (K): Noise power added by the filter.

Insertion Loss: Insertion loss defines the magnitude by which pass band signals are attenuated and is specified in dB. The pass band of the band pass filter is defined as all frequencies that are above the lower cutoff frequency and below the higher cutoff frequency.

Stop Band Attenuation: Stop band attenuation defines the magnitude of the attenuation for signals with frequencies below the lower stop band frequency and above the higher stop band frequency. The attenuation of signals with frequencies between the lower stop band frequency and the lower cutoff frequency or between the higher cutoff frequency and the higher stop band frequency is determined by calculating the slope between these points and assuming a linear roll-off.

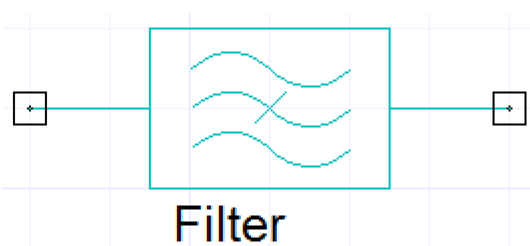
Lower Stop Band: The lower stop band defines the maximum frequency of the lower stop band. All frequencies below the lower stop band value are attenuated by an amount equal to the stop band attenuation.

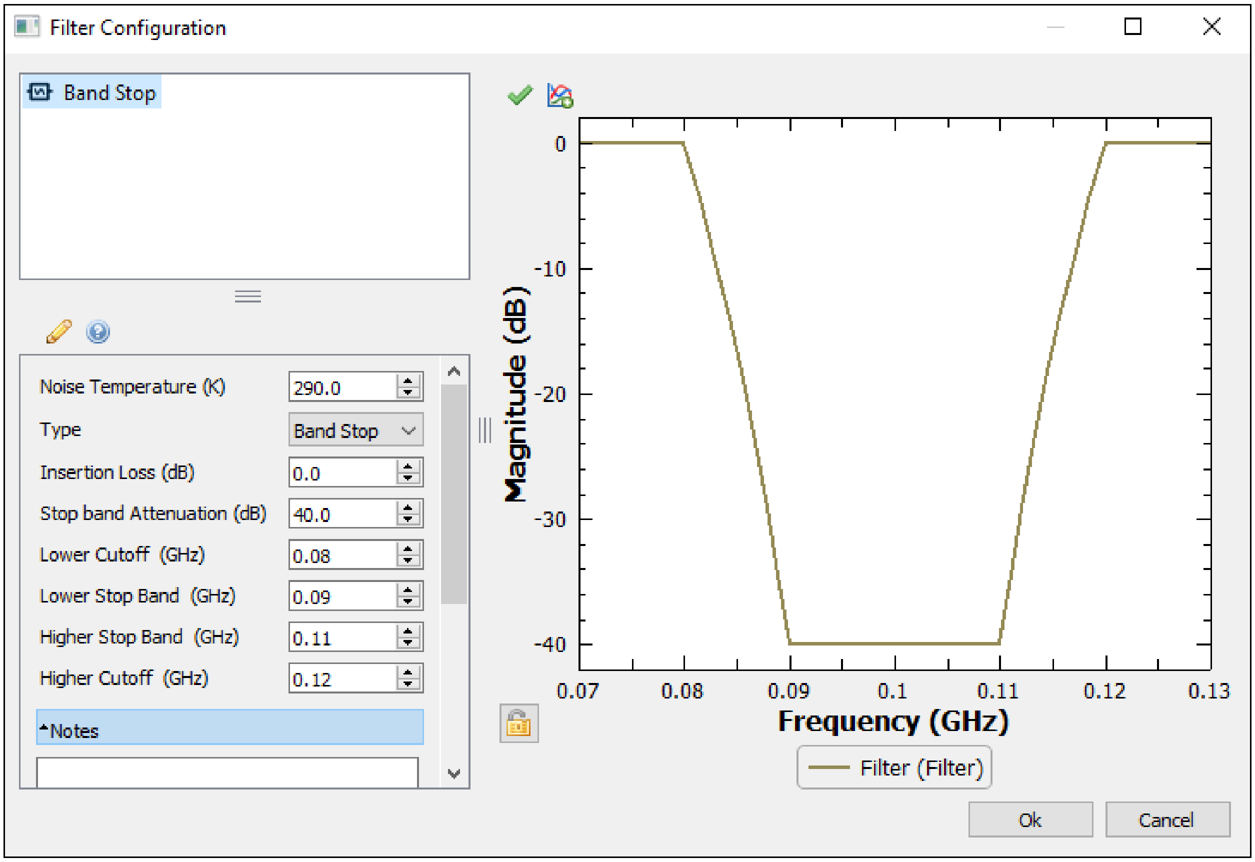
Lower Cutoff: The lower cutoff defines the minimum frequency of the pass band. All frequencies between the lower cutoff and the higher cutoff are attenuated by an amount equal to the filter's insertion loss.

Higher Cutoff: The higher cutoff defines the maximum frequency of the pass band. All frequencies between the lower cutoff and the higher cutoff are attenuated by an amount equal to the filter's insertion loss.

Higher Stop Band: The higher stop band defines the minimum frequency of the higher stop band. All frequencies greater than the higher stop band value are attenuated by an amount equal to the stop band attenuation.

Band Stop





Defines a band stop filter with the parameters shown in the configuration window above. A band stop filter passes frequencies in the pass bands with minimal attenuation and rejects (attenuates) signals outside of the pass bands. A band stop filter will pass both low and high frequencies and attenuate a specific band of frequencies as seen below in the plot of a band pass filter in EMIT. A band stop filter can be thought of as a low pass filter and a high pass filter cascaded together with a portion of their stop bands overlapping. In essence, a band stop filter has two pass bands. A band stop filter is sometimes referred to as a notch filter.

Noise Temperature (K): Noise power added by the filter.

Insertion Loss: Insertion loss defines the magnitude by which pass band signals are attenuated and is specified in dB. The pass bands of the band stop filter are defined as all frequencies that are below the lower cutoff frequency and above the higher cutoff frequency.

Stop Band Attenuation: Stop band attenuation defines the magnitude of the attenuation for signals with frequencies above the lower stop band frequency and below the higher stop band frequency. The attenuation of signals with frequencies between the lower cutoff frequency and the lower stop band frequency or between the higher stop band frequency and the higher cutoff frequency is determined by calculating the slope between these points and assuming a linear roll-off.

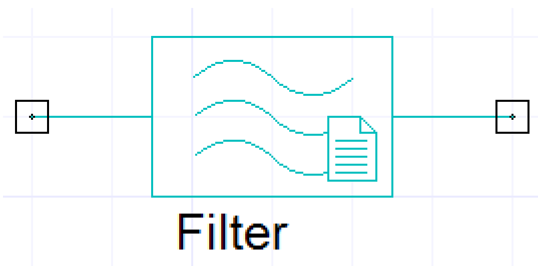
Lower Cutoff: The lower cutoff defines the maximum frequency of the lower pass band. All frequencies between DC and the lower cutoff frequency are attenuated by an amount equal to the filter's insertion loss.

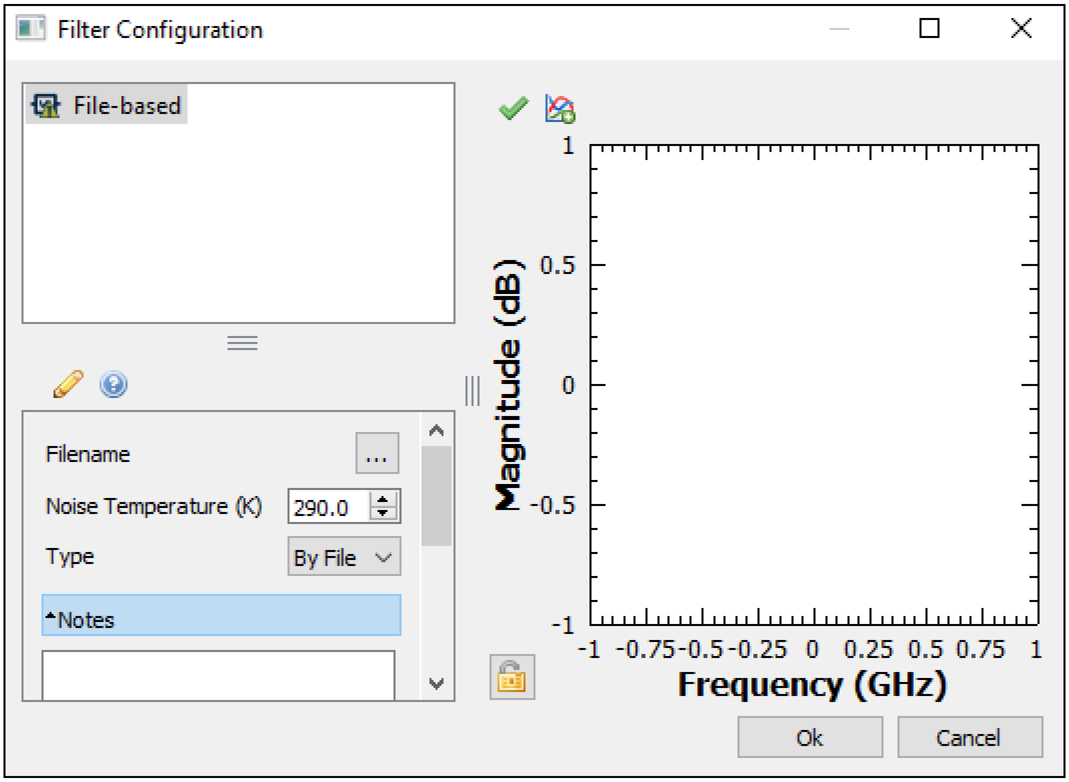
Lower Stop Band: The lower stop band defines the minimum frequency of the stop band. All frequencies between the lower stop band value and the higher stop band value are attenuated by an amount equal to the stop band attenuation.

Higher Stop Band: The higher stop band defines the maximum frequency of the stop band. All frequencies between the lower stop band value and the higher stop band value are attenuated by an amount equal to the stop band attenuation.

Higher Cutoff: The higher cutoff defines the minimum frequency of the higher pass band. All frequencies above the higher cutoff are attenuated by an amount equal to the filter's insertion loss.

By File



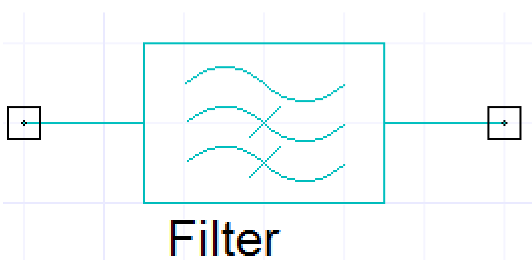


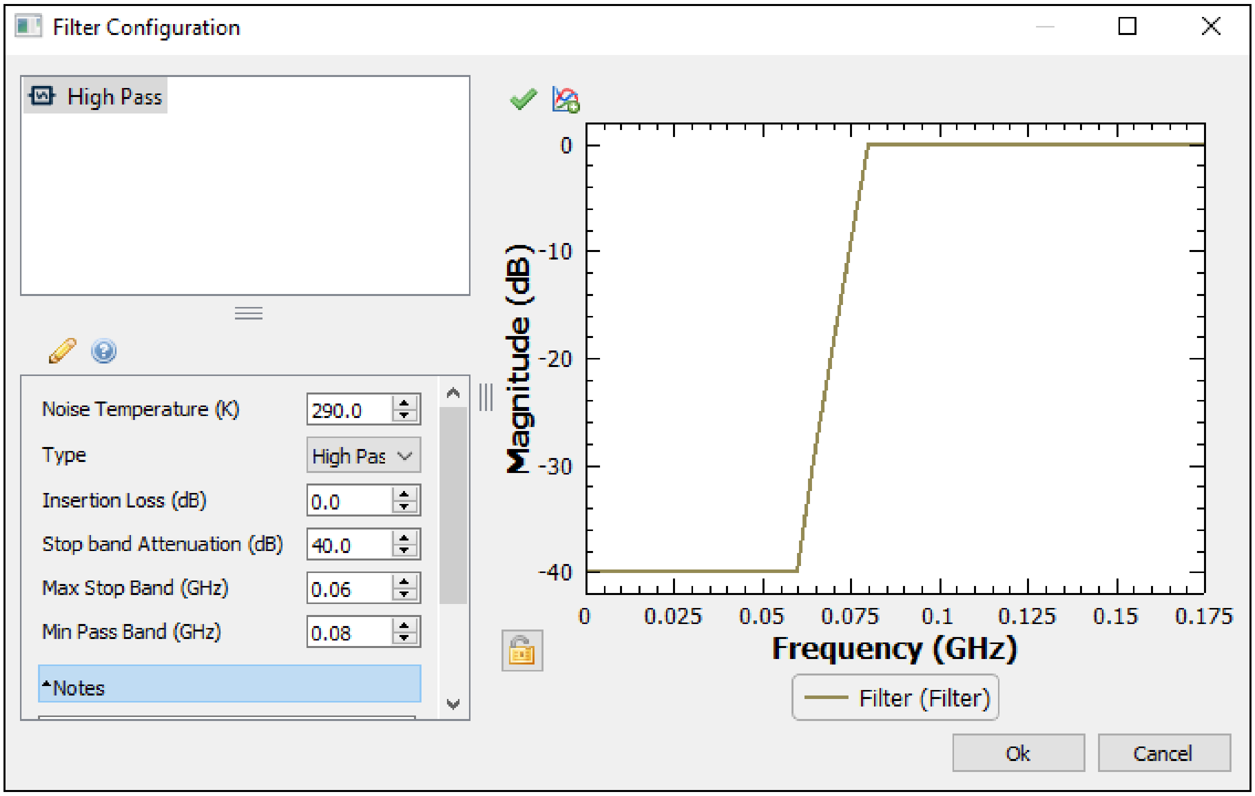
Specify the filter using a *.dlxsp or Touchstone file which contains the filter's transmission characteristics.

Filename: Displays the file name being used for the filter characteristics. Click the browse button to choose a new file.

Noise Temperature (K): Noise power added by the filter.

High Pass





Defines a high pass filter with the parameters shown in the configuration window above. A high pass filter passes high frequency signals with minimal attenuation and attenuates (reduces the amplitude of) signals below the cutoff frequency. Also shown is a plot of a high pass filter in EMIT.

Noise Temperature (K): Noise power added by the filter.

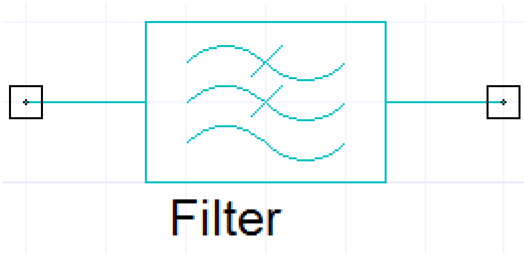
Insertion Loss: Insertion loss defines the magnitude of the attenuation of signals in the pass band and is specified in dB. The pass band of the high pass filter is defined as all frequencies above the minimum pass band frequency.

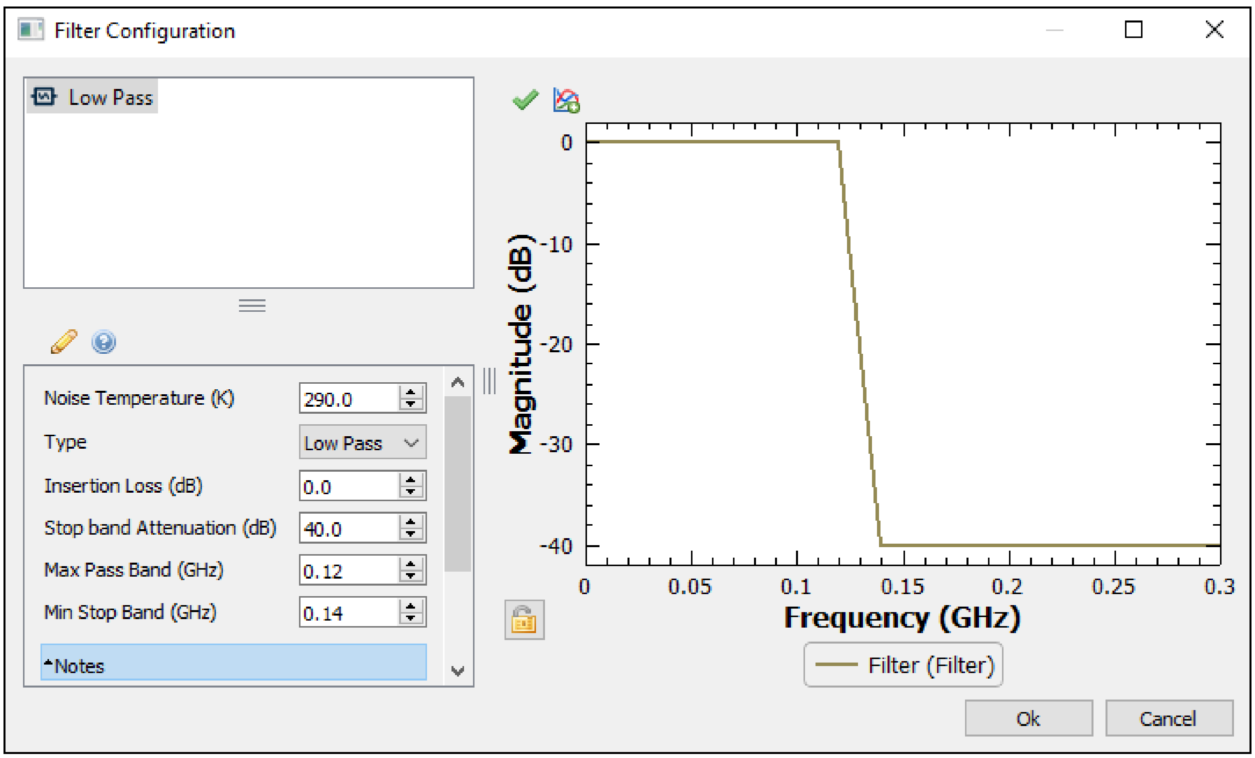
Stop Band Attenuation: Stop band attenuation defines the magnitude of the attenuation for signals with frequencies lower than the maximum stop band frequency. The attenuation of signals with frequencies between the maximum stop band frequency and the minimum pass band frequency is determined by calculating the slope between these two points and assuming a linear roll-off.

Max Stop Band: The max stop band defines the maximum frequency of the stop band. All frequencies below the max stop band value are attenuated by an amount equal to the stop band attenuation.

Min Pass Band: The min pass band defines the minimum frequency of the pass band which is also commonly referred to as the filter's cutoff frequency.

Low Pass





Defines a low pass filter with the parameters shown in the configuration window above. A low pass filter passes low frequency signals with minimal attenuation and attenuates (reduces the amplitude of) signals above the cutoff frequency. Also shown is a plot of a low pass filter in EMIT.

Noise Temperature (K): Noise power added by the filter.

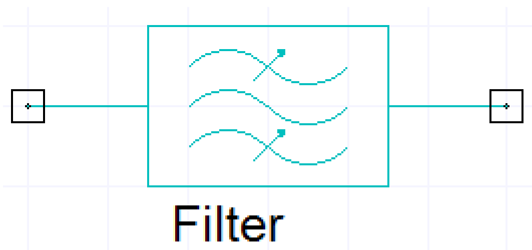
Insertion Loss: Insertion loss defines the magnitude of the attenuation of signals in the pass band and is specified in dB. The pass band of the low pass filter is defined as all frequencies from DC up to the maximum pass band frequency.

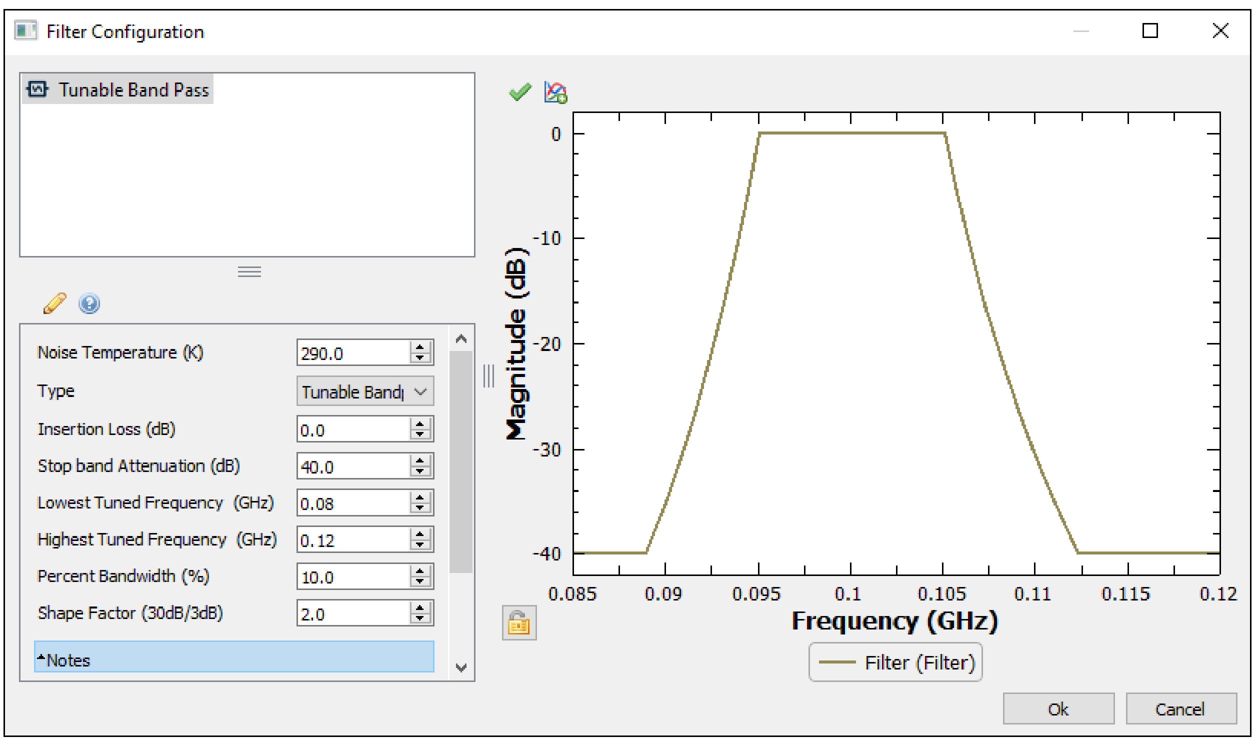
Stop Band Attenuation: Stop band attenuation defines the magnitude of the attenuation for signals with frequencies greater than the minimum stop band frequency. The attenuation of signals with frequencies between the maximum pass band frequency and the minimum stop band frequency is determined by calculating the slope between these two points and assuming a linear roll-off.

Max Pass Band: The max pass band defines the maximum frequency of the pass band which is also commonly referred to as the filter's cutoff frequency.

Min Stop Band: The min stop band defines the lowest frequency of the stop band. All frequencies above the min stop band value are attenuated by an amount equal to the stop band attenuation.

Tunable Band Pass





Defines a tunable band pass filter with the parameters shown in the configuration window above. A tunable band pass filter passes frequencies in the pass bands with minimal attenuation and rejects (attenuates) signals outside of the pass bands. A tunable band pass filter will attenuate both low and high frequencies and pass a specific band of frequencies as seen below in the plot of a band pass filter in EMIT.

Tunable band pass filters are defined in terms of a tuning range and a percent bandwidth. For each channel within a transceiver, the spectral profile of the tunable filter will be recreated. For example, assume a transmitter is defined that operates from 100-110 MHz in 2 MHz channels and the filter has a 10% bandwidth. When EMIT is simulating the 100 MHz channel, the filter is centered at 100 MHz with a bandwidth of 10 MHz (red). For the 102 MHz channel, the filter is centered at 102 MHz with a 10.2 MHz bandwidth (blue). For the 110 MHz channel, the filter is centered at 110 MHz with an 11 MHz bandwidth (green). This is illustrated in the plot below.

> **Note:**
>
> When a transceiver is operating at a different frequency for transmit than for receive (e.g. for self interaction with a Tx Offset), the tunable filter always tunes based on the transmit channel frequency.

Noise Temperature (K): Noise power added by the filter.

Insertion Loss: Insertion loss defines the magnitude by which pass band signals are attenuated and is specified in dB.

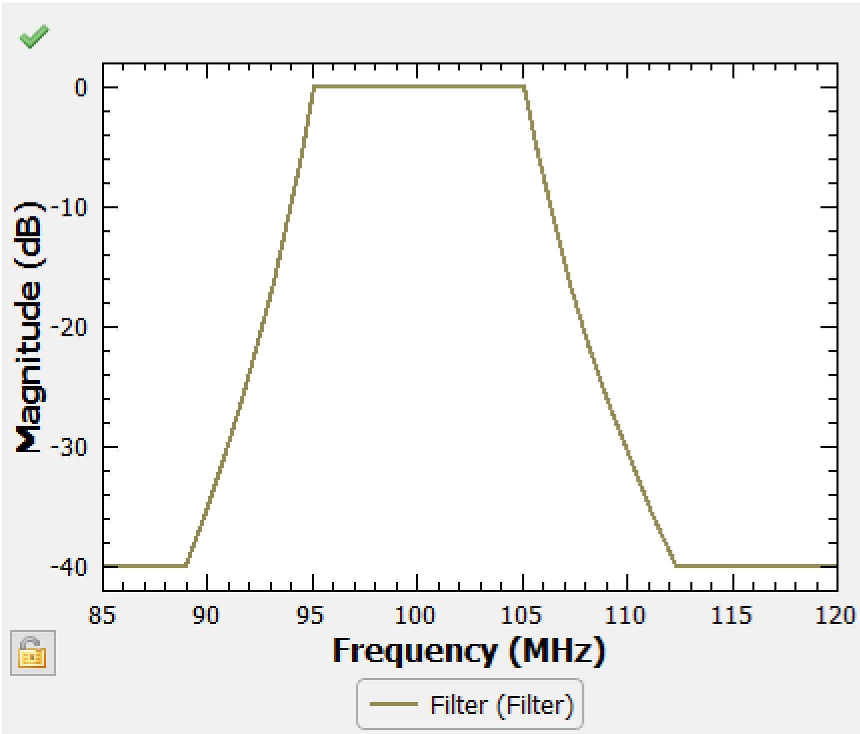
Stop Band Attenuation: Stop band attenuation defines the magnitude of the attenuation for signals with frequencies outside the pass band.

Lowest Tuned Frequency: The lowest tuned frequency defines the lower end of the tuning range for the filter. If any Tx and/or Rx channels are defined below this frequency, then the filter will be centered at the lowest tuned frequency, rather than at the channel frequency.

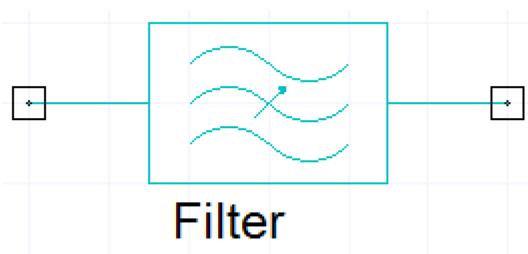
Highest Tuned Frequency: The highest tuned frequency defines the higher end of the tuning range for the filter. If any Tx and/or Rx channels are defined above this frequency, then the filter will be centered at the highest tuned frequency, rather than at the channel frequency.

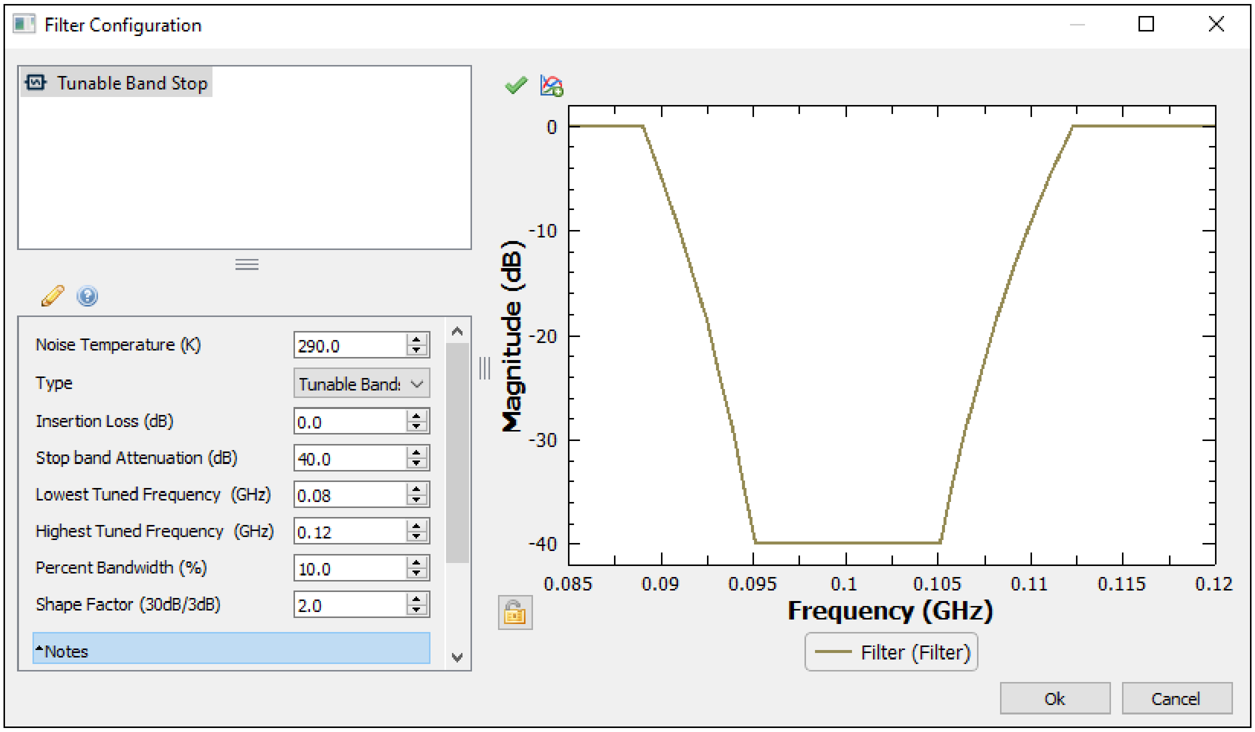
Percent Bandwidth (%): The percent bandwidth defines the the bandwidth of the filter stop band (notch) with respect to the center frequency of the filter.

Shape Factor: The shape factor describes the roll-off of the filter between the 3 dB and 30 dB bandwidth points.



Tunable Band Stop





Defines a tunable band stop filter with the parameters shown in the configuration window above. A tunable band stop filter passes frequencies in the pass bands with minimal attenuation and rejects (attenuates) signals outside of the pass bands. A tunable band stop filter will pass both low and high frequencies and attenuate a specific band of frequencies as seen below in the plot of a band stop filter in EMIT. A band stop filter can be thought of as a low pass filter and a high pass filter cascaded together with a portion of their stop bands overlapping. In essence, a band stop filter has two pass bands. A band stop filter is sometimes referred to as a notch filter.

Tunable band stop filters are defined in terms of a tuning range and a percent bandwidth. For each channel within a transceiver, the spectral profile of the tunable filter will be recreated. For example, assume a transmitter is defined that operates from 100-110 MHz in 2 MHz channels and the filter has a 10% bandwidth. When EMIT is simulating the 100 MHz channel, the filter is centered at 100 MHz with a bandwidth of 10 MHz (red). For the 102 MHz channel, the filter is centered at 102 MHz with a 10.2 MHz bandwidth (blue). For the 110 MHz channel, the filter is centered at 110 MHz with an 11 MHz bandwidth (green). This is illustrated in the plot below.

> **Note:**
>
> When a transceiver is operating at a different frequency for transmit than for receive (e.g. for self interaction with a Tx Offset), the tunable filter always tunes based on the transmit channel frequency.

Noise Temperature (K): Noise power added by the filter.

Insertion Loss: Insertion loss defines the magnitude by which pass band signals are attenuated and is specified in dB. The pass bands of the band stop filter are defined as all frequencies that are below the lower cutoff frequency and above the higher cutoff frequency.

Stop Band Attenuation: Stop band attenuation defines the magnitude of the attenuation for signals with frequencies above the lower stop band frequency and below the higher stop band frequency.

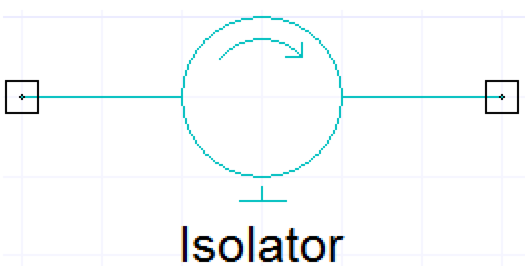
Lowest Tuned Frequency: The lowest tuned frequency defines the lower end of the tuning range for the filter. If any Tx and/or Rx channels are defined below this frequency, then the filter will be centered at the lowest tuned frequency, rather than at the channel frequency.

Highest Tuned Frequency: The highest tuned frequency defines the higher end of the tuning range for the filter. If any Tx and/or Rx channels are defined above this frequency, then the filter will be centered at the highest tuned frequency, rather than at the channel frequency.

Percent Bandwidth (%): The percent bandwidth defines the the bandwidth of the filter stop band (notch) with respect to the center frequency of the filter.

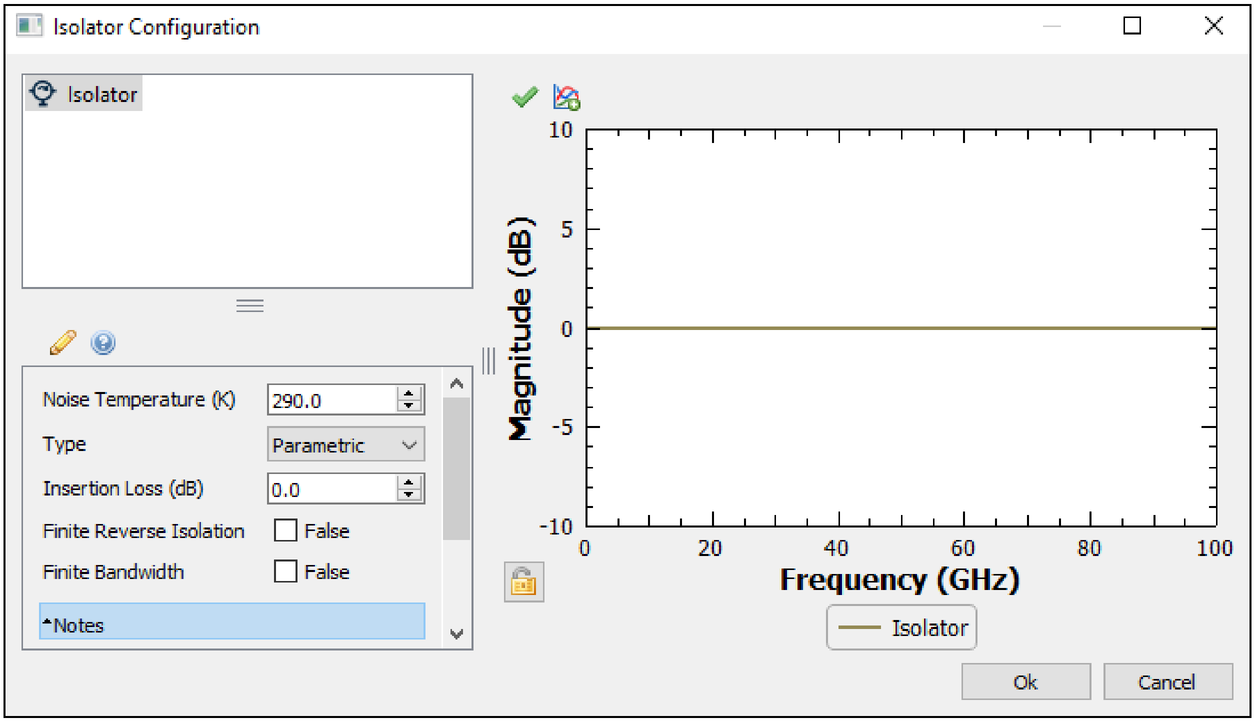
Shape Factor: The shape factor describes the roll-off of the filter between the 3 dB and 30 dB bandwidth points.

Isolators



In EMIT an isolator is a passive non-reciprocal two-port device, in which a signal entering port 1 is transmitted to port 2 but not vice versa. In practice this is usually achieved by terminating one port of a Circulator. A signal applied to port 1 only comes out of port 2. Unless otherwise specified by the Finite Reverse Isolation or Finite Bandwidth settings, EMIT assumes ideal (infinite) reverse isolation and infinite bandwidth for isolators.

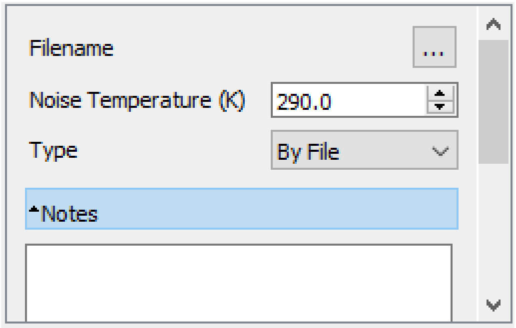
Isolator



In the **Type** drop-down, select one of the following:

- By File
- Parametric

By File

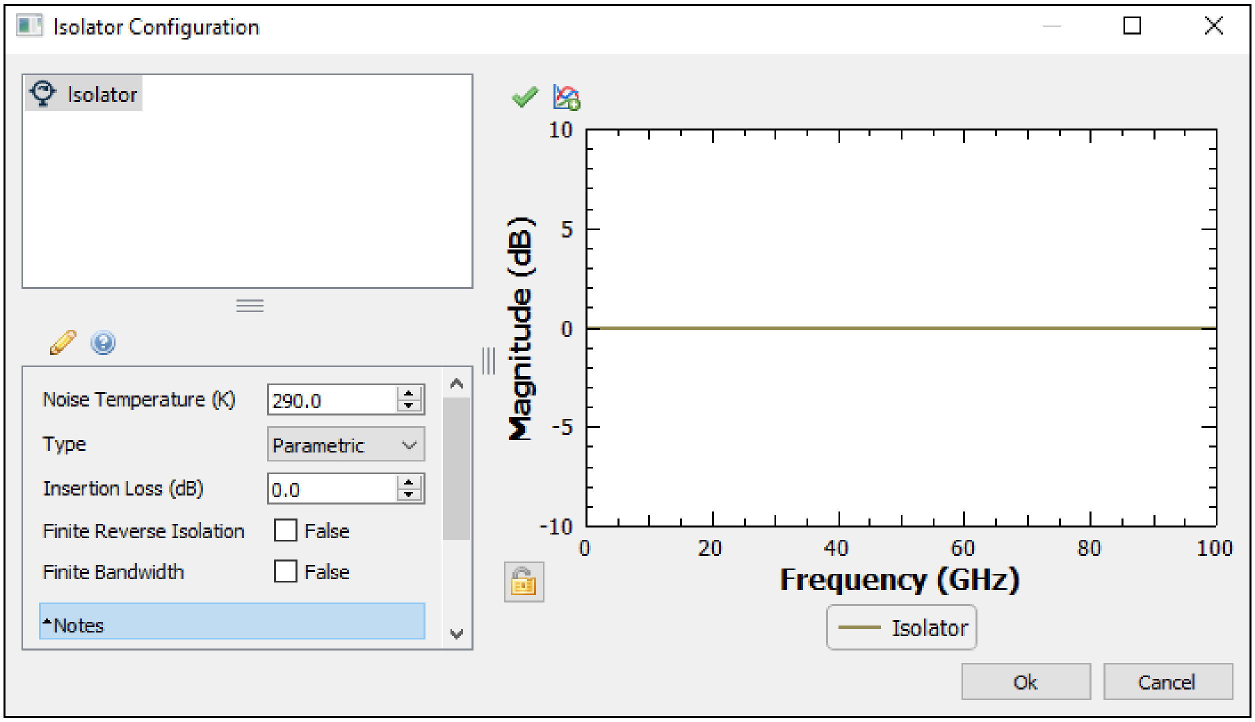


Filetype: Defines the 2 port frequency characteristics of the Isolator by providing a 2-port Touchstone file containing the S-parameters for the device.

Noise Temperature (K): Noise power added by the isolator.

Parametric

Allows definition of the Isolator characteristics by entering values for the appropriate parameters.

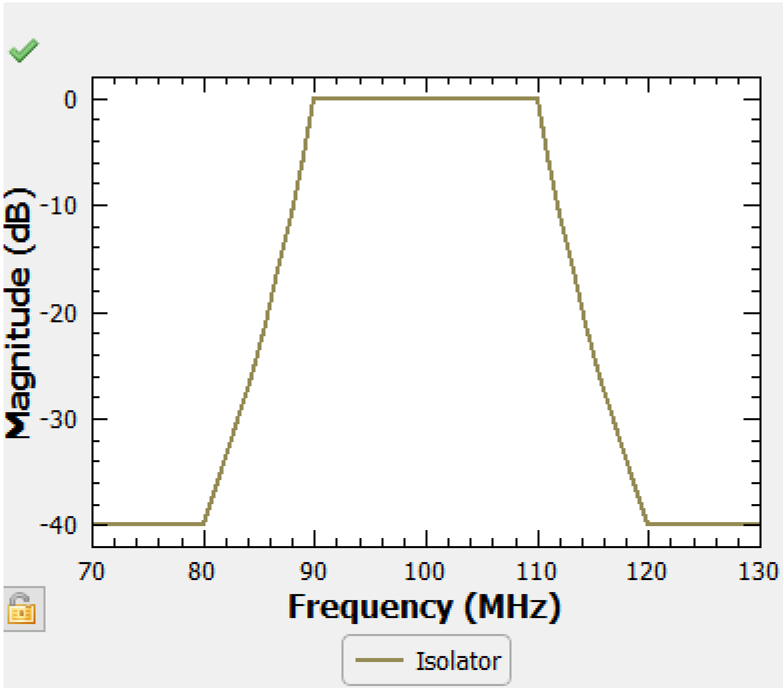


Noise Temperature (K): Noise power added by the isolator.

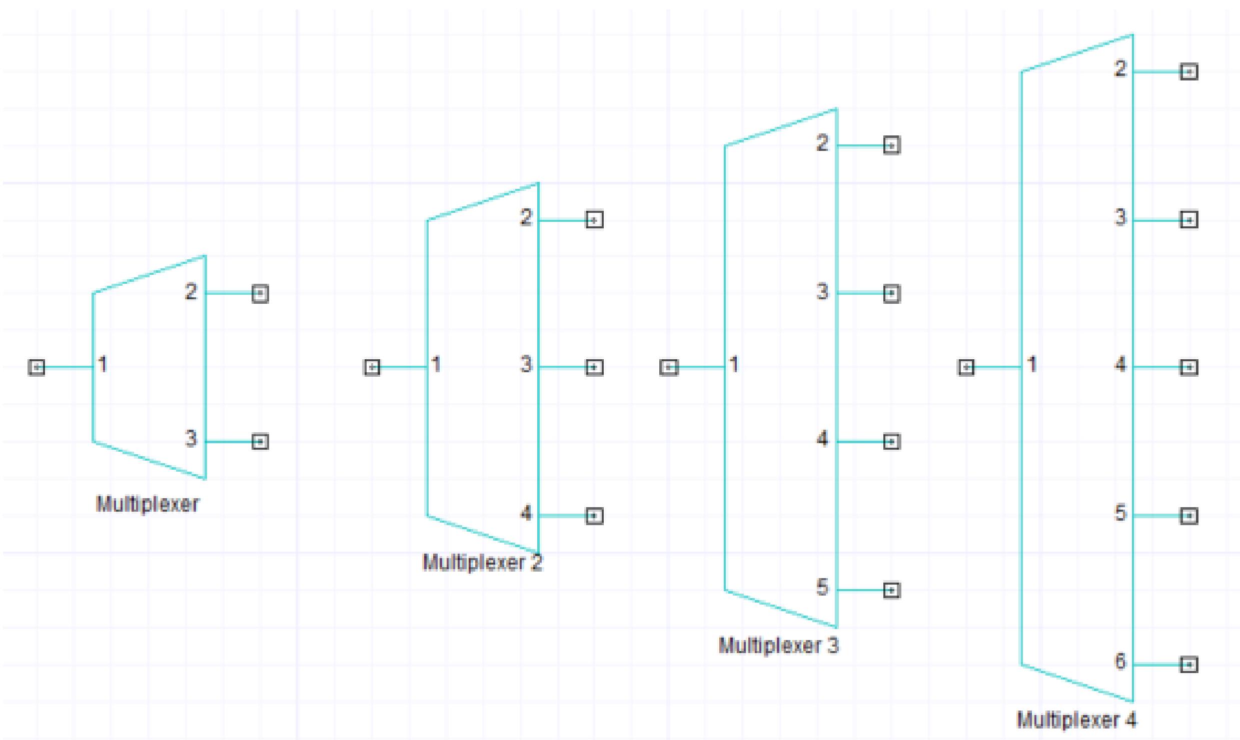
Insertion Loss (dB): Defines the insertion loss between ports of the device.

Finite Reverse Isolation: The reverse isolation between the output (port 2) and input (port 1) is assumed to be infinite unless Finite Reverse Isolation is enabled (True), in which case the user-provided value is used.

Finite Bandwidth: By default, the bandwidth for a Isolator is considered to be infinite. When Finite Bandwidth is enabled (True) a finite bandwidth for the Isolator can be defined. The Out-of-Band Attenuation is then applied outside of the defined pass band as shown in the plot below. See Band Pass Filter for definitions of the parameters.



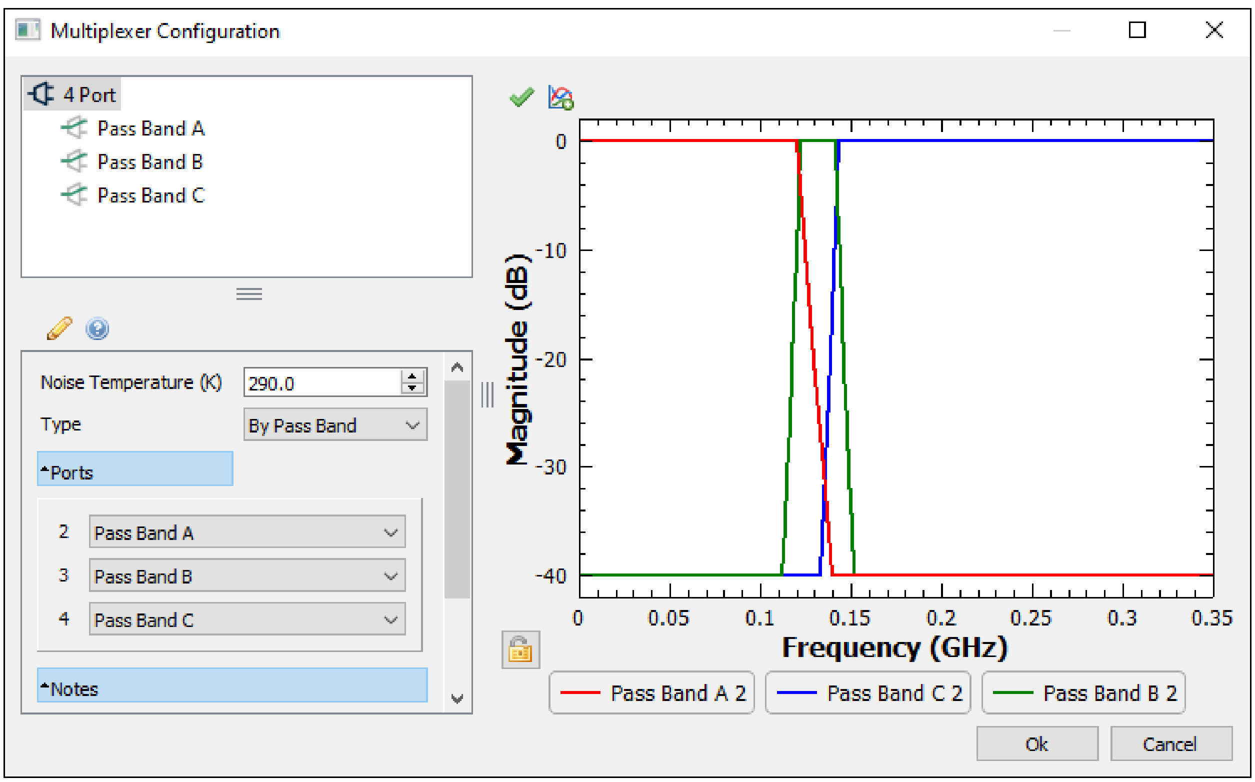
Multiplexers



A multiplexer is a N-port device that routes signals from port 1 to the other N ports based on the frequency filtering characteristics defined between the input and output ports. A multiplexer is essentially a multi-port filter and, as described below, the port-to-port couplings are defined in the same way as a filter in EMIT. EMIT supports 3-, 4-, 5-, and 6-port multiplexers, as shown in the figure above.

A multiplexer's port-to-port filtering characteristics can be defined parametrically using pass bands or by reading in an N-port Touchstone file. When defining the multiplexer using pass bands, (N-1) pass bands must be added to the multiplexer and defined in each pass band's property panel. Each pass band defines the 2-port insertion loss between pairs of ports on the multiplexer.

Note that pass bands define the coupling between the input port and each of the output ports. It is not possible to include coupling between other port pairs (for example, isolation between output ports) when defining a multiplexer using pass bands. When the pass bands are defined, the isolation between the output ports is assumed to be infinite. The pass bands can be defined parametrically or using a 2-port Touchstone file. In order to specify a finite isolation between the output ports, the entire multiplexer must be defined by importing an appropriate Touchstone file (that is, a 3-port, 4-port, 5-port or 6-port Touchstone file).



In the **Type** drop-down, select one of the following:

- By Pass Band
- By File

Selecting the multiplexer node in the component tree shows the composite pass bands for the Multiplexer. This provides a high-level view of each of the defined pass bands.

By Pass Band

Defines the N-port frequency characteristics of the multiplexer by adding and defining (N-1) pass bands which appear in the configuration tree as children of the multiplexer node. The pass bands define the frequency port-to-port filtering characteristics using parametric models or by file import.

Noise Temperature (K): Noise power added by the multiplexer.

Ports: Define which pass bands are associated with which ports of the multiplexer.

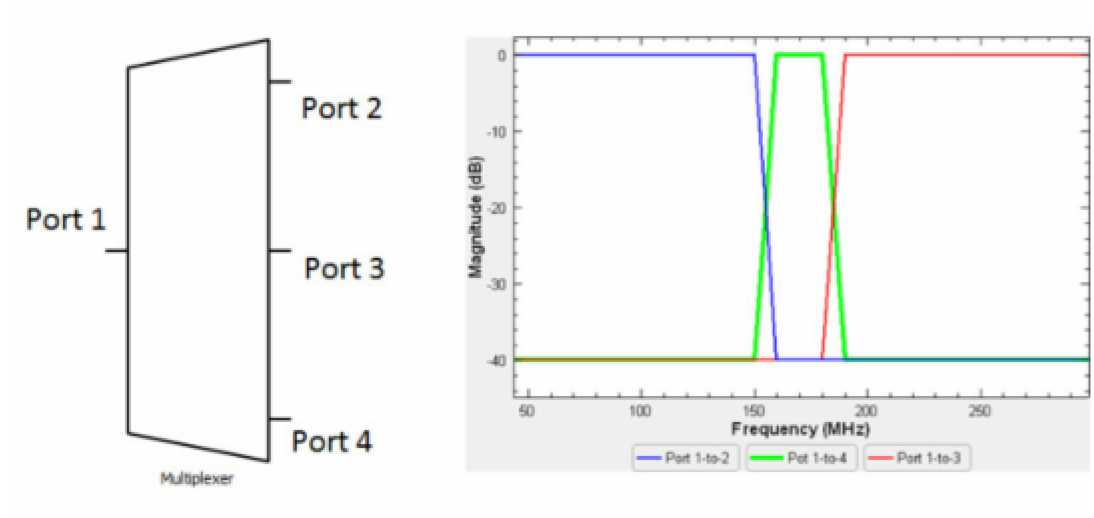
By File

Filename: Defines the N-port frequency characteristics of the multiplexer by providing an N-port (3-port, 4-port, 5-port, or 6-port) Touchstone file containing the S-Parameters for the device. The number of ports in the Touchstone file must match the number of ports for the multiplexer.

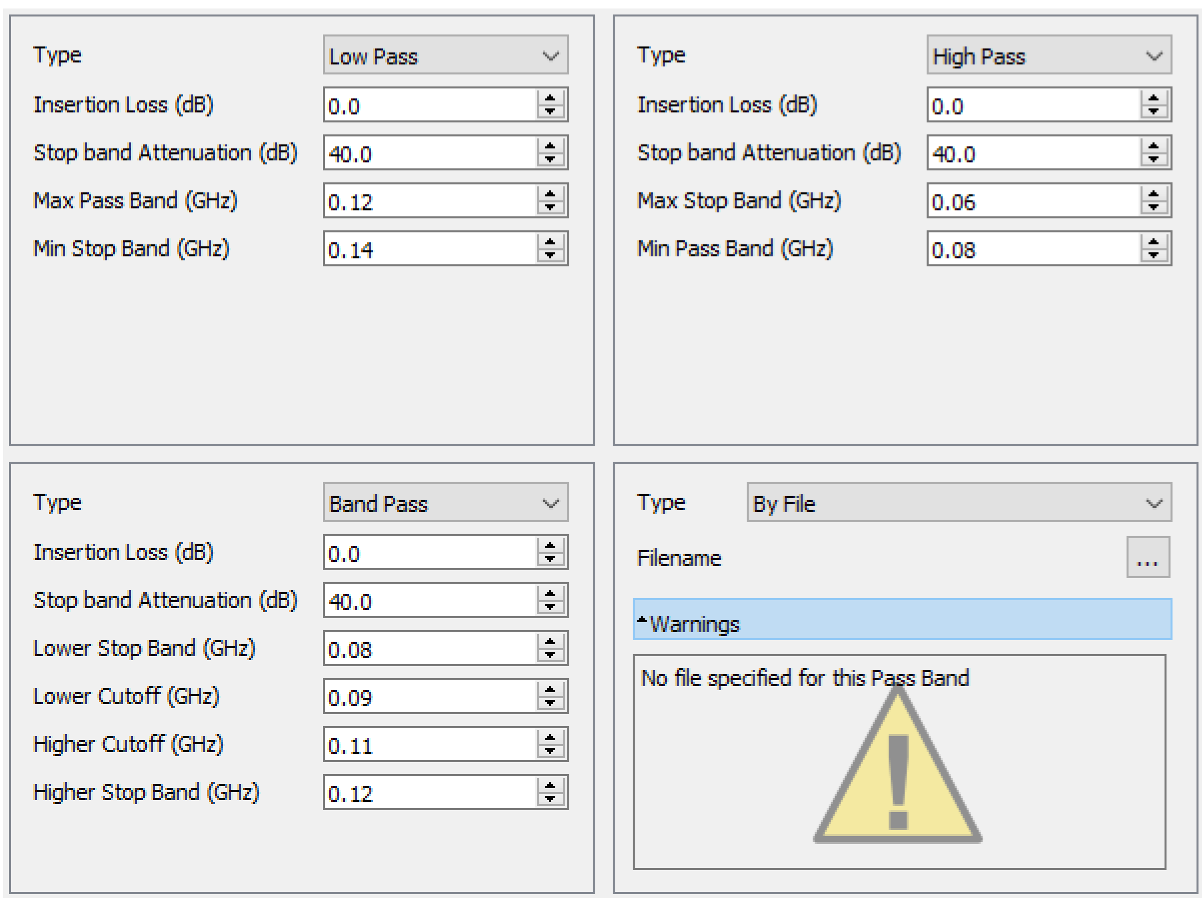
Noise Temperature (K): Noise power added by the multiplexer.

Multiplexer Passbands

A Multiplexer's port-to-port filering characteristics can be defined parametrically using passbands that can be defined by reading in a 2-port Touchstone file or by defining its filter characteristics parametrically. Each passband defines the coupling between the common port and one of the output ports. It is not possible to include coupling between other port pairs (for example, isolation between output ports) when defining a Multiplexer using passbands. In order to do so, the Multiplexer must be defined by importing an appropriate N-Port Touchstone file. When defining the Multiplexer using passbands, (N-1) passbands must be defined. Each passband defines the 2-port insertion loss between the common port and an output port of the Multiplexer. In the example shown in the figure below, three passbands are defined corresponding to each of the colored traces in the plot.



EMIT provides four different ways of defining a passband. The property panels for the four methods are slightly different as shown in the figure below.



Type: Select the method to use for defining the passband. If the type is not changed (that is, left as By File), then a file containing the frequency response must be specified.

By File: The user defines the 2-port frequency characteristics of the passband by providing a 2-port Touchstone file containing the S-Parameters for the device. Port 1 in the Touchstone file is associated with the common port of the Multiplexer.

Low Pass: Defines a low pass filter characteristic for the passband with the parameters shown in the property panel above. A low pass passband passes low frequency signals with minimal attenuation and attenuates (reduces the amplitude of) signals above the cutoff frequency.

High Pass: Defines a high pass filter characteristic for the passband with the parameters shown in the property panel above. A high pass passband passes high frequency signals with minimal attenuation and attenuates signals below the cutoff frequency.

Band Pass: Defines a band pass filter characteristic for the passband with the parameters shown in the property panel above. A band pass passband passes frequencies in the pass band with minimal attenuation and rejects (attenuates) signals outside of the passband. A band pass passband will attenuate both low and high frequencies.

Insertion Loss: Defines the magnitude of the attenuation of signals in the passband and is specified in dB. For a Low Pass Passband, the passband is defined from DC up to the maximum passband frequency. For a High Pass Passband, the passband is defined as all frequencies above the minimum passband frequency and for Band Pass Passbands, the passband is defined as all frequencies above the lower cutoff frequency and below the higher cutoff frequency.

Stop Band Attenuation: Defines the magnitude of the attenuation for signals with frequencies outside the passband.

Max Pass Band: Defines the maximum frequency of the passband (for Low Pass Passbands) which is also commonly referred to as the filter's cutoff frequency.

Min Stop Band: Defines the lowest frequency of the stop band (for Low Pass Passbands). All frequencies above the min stop band value are attenuated by an amount equal to the stop band attenuation. The attenuation at frequencies between the max pass band and min stop band values are linearly interpolated.

Max Pass Band: Defines the maximum frequency of the stop band (for High Pass Passbands). All frequencies below the max stop band value are attenuated by an amount equal to the stop band attenuation.

Min Pass Band: Defines the minimum frequency of the pass band (for High Pass Passbands) which is also commonly referred to as the filter's cutoff frequency. The attenuation at frequencies between the min pass band and max pass band values are linearly interpolated.

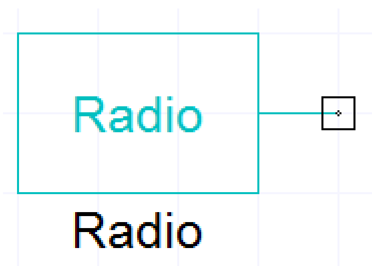
Lower Stop Band: Defines the maximum frequency of the lower stop band (for Band Pass Passbands). All frequencies below the lower stop band value are attenuated by an amount equal to the stop band attenuation.

Lower Cutoff: Defines the minimum frequency of the pass band (for Band Pass Passbands). All frequencies between the lower cutoff and the higher cutoff are attenuated by an amount equal to the filter's insertion loss.

Higher Cutoff: Defines the maximum frequency of the pass band (for Band Pass Passbands). All frequencies between the lower cutoff and the higher cutoff are attenuated by an amount equal to the filter's insertion loss.

Higher Stop Band: Defines the minimum frequency of the higher stop band (for Band Pass Passbands). All frequencies greater than the higher stop band value are attenuated by an amount equal to the stop band attenuation.

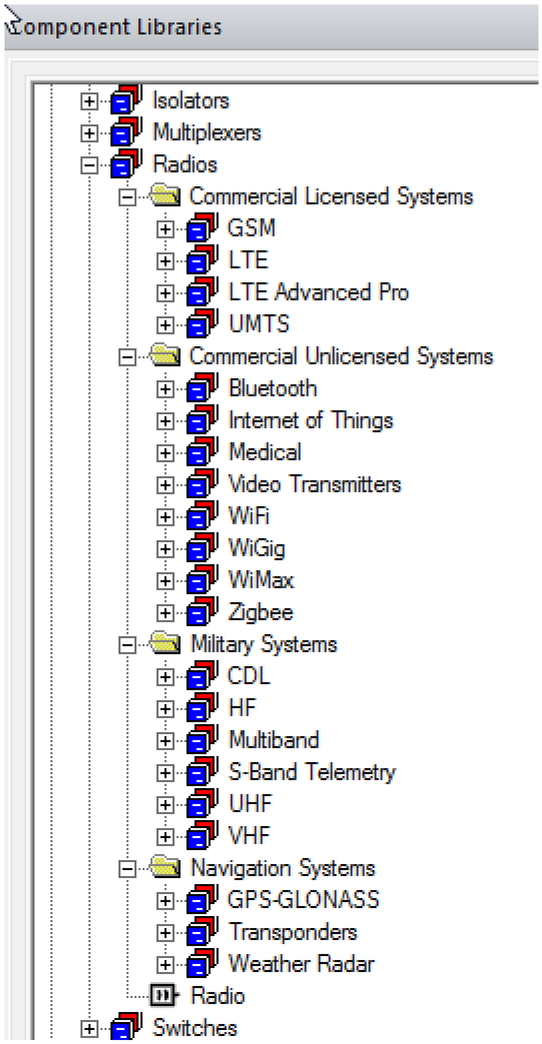
Radios

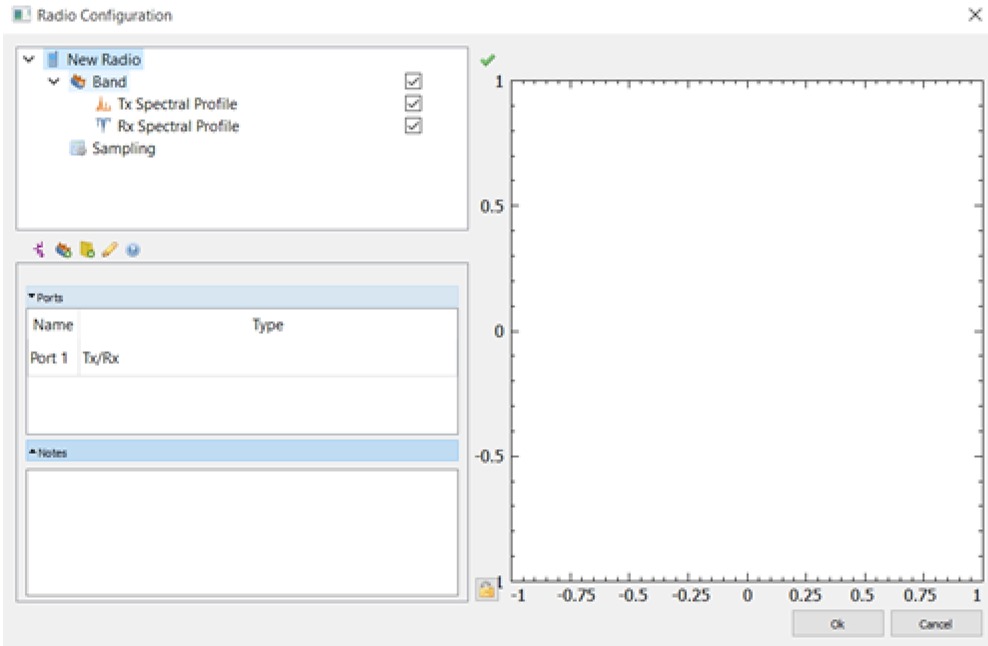


Ansys EMIT has a radio component that contains Tx spectral and Rx spectral profiles.

Ansys EMIT also provides four libraries that contain common radio models that are available when creating EMIT projects. These libraries are divided by application area:

- **Commercial Licensed Systems:** Common commercial radios that operate primarily in licensed radio bands. Cellular radios are the most popular of these. LTE Advanced Pro - Extended Models extend the "base" LTE Advanced Pro model with models for additional channel bandwidths beyond the 20 MHz channel bandwidth.
- **Commercial Unlicensed Systems:** Common commercial radios that primarily operate in unlicensed spectrum, for example WiFi, Bluetooth, and ZigBee.
- **Military Systems:** Radios primarily used by the military with limited commercial uses. Some common examples are the ARC-210 and PRC-117.
- **Navigation Systems:** Radios whose primary function is navigation related. For example, GPS and aircraft transponders.


Component Libraries
Isolators
Multiplexers
Radios
Commercial Licensed Systems
GSM
LTE
LTE Advanced Pro
UMTS
Commercial Unlicensed Systems
Bluetooth
Internet of Things
Medical
Video Transmitters
WiFi
WiGig
WiMax
Zigbee
Military Systems
CDL
HF
Multiband
S-Band Telemetry
UHF
VHF
Navigation Systems
GPS-GLONASS
Transponders
Weather Radar
Radio
Switches



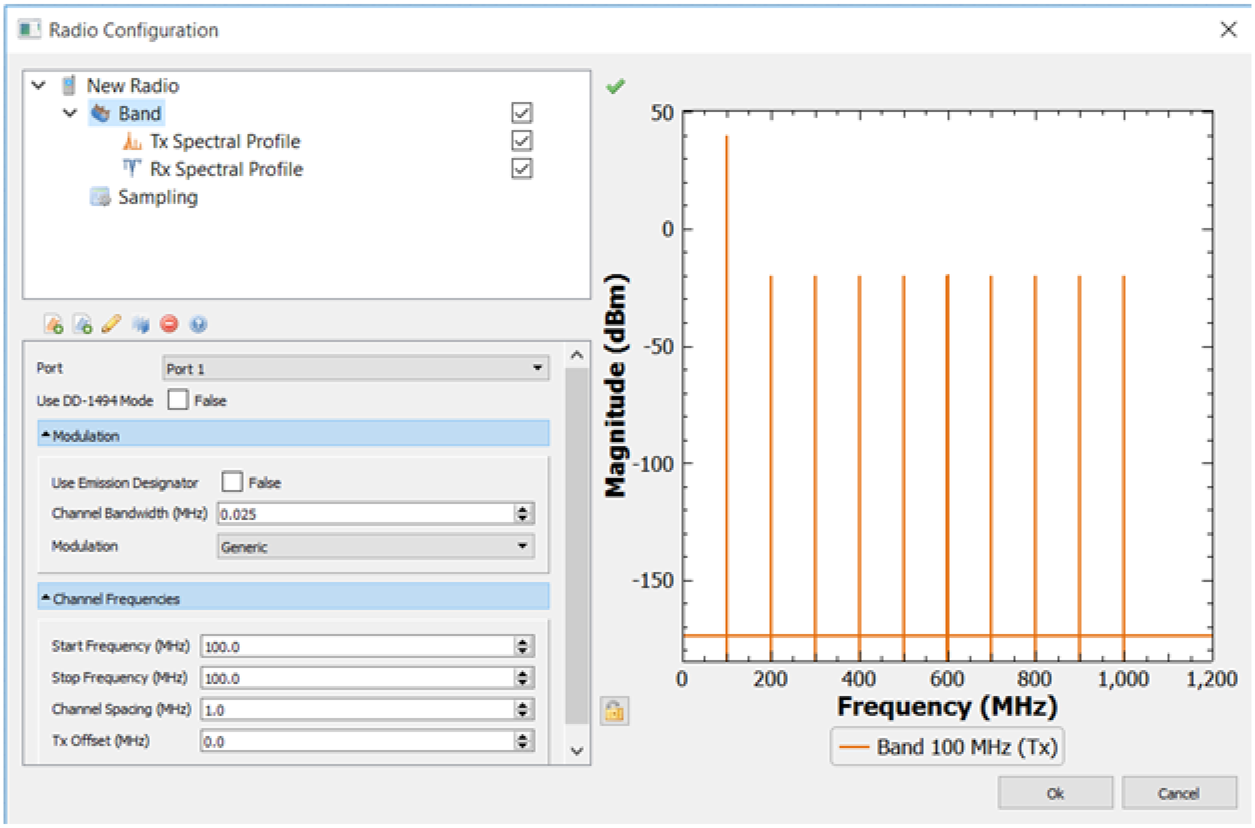
A Radio in EMIT can represent a transceiver, transmitter and/or receiver. A radio has a single "port". A radio gets connected to antennas and other components via this port. The connections are defined in the RF system schematic editor. Essentially, you can think of a radio's port as an antenna connector on the radio.

The port on a radio can be designated as transmit (Tx Only), receive (Rx Only) or both (Tx/Rx), and this is done in the Radio configuration window shown above

The specifications (channel frequencies, modulation, etc.) for each radio are defined by one or more bands that appear as children to the radio node as shown in the project tree snippet below, where three bands are shown: VHF Band, UHF Band and GPS Band. Children of the band nodes contain the specifications for Tx (Tx Spectral Profile) and Rx (Rx Spectral Profile) characteristics for the parent band.

The sampling node provides settings to control which of the available channels are used in the simulation.

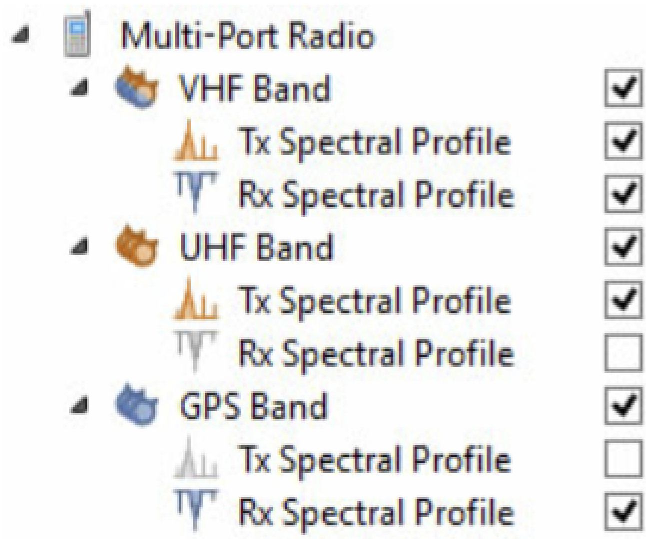
Bands



Bands in EMIT define the operating parameters (channel frequencies, power levels, modulation, etc.) for their parent radio. The bands appear as children of the radio. A radio can have any number of bands, and folders can be used to organize the bands. Children of the band node contain the specifications for Tx (Tx Spectral Profile) and Rx (Rx Spectral Profile) characteristics for the parent band. Multiple bands can be assigned to the radio's port.

All bands contain a Tx Spectral Profile and Rx Spectral Profile node as children. When both of these are enabled the band can be both Tx and Rx (i.e., a transceiver). When only one or the other is enabled the Band can only Tx or Rx.

The Band icon changes to indicate the Tx and/or Rx functionality of the Band as shown below where the VHF Band is a transceiver band, the UHF Band is a transmitter band, and the GPS band is a receiver band.



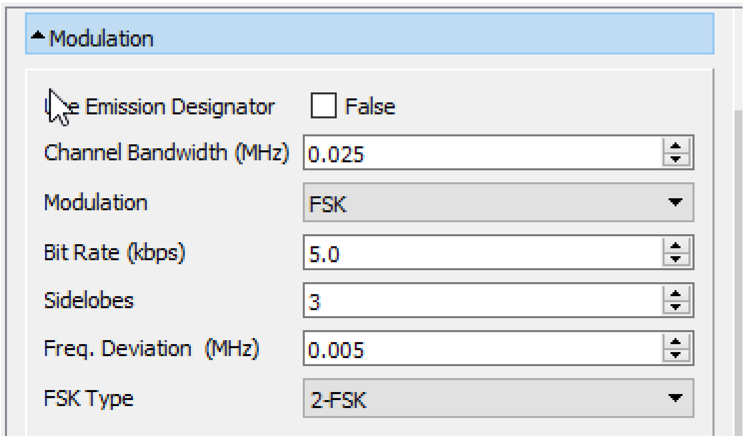
Bands can be plotted to show their Tx and/or Rx spectral characteristics by selecting the Band node in the configuration tree. When plotting a band, you can switch between displaying the Tx or Rx characteristic (for a transceiver band) and the channel frequency by clicking on the trace label at the bottom of the plot.

Port: Defines which radio port is to be associated with the band. Multiple bands can be assigned to the radio's port. Note that at the current time radios are limited to a single port.

Use DD-1494 Mode: When installing radio systems on a military base or other Department of Defense sites, it is necessary to obtain equipment certification from the National Telecommunications and Information Administration (NTIA) by submitting DD form 1494, Application for Equipment Frequency Allocation. The DD form 1494 includes the proposed technical characteristics of the overall system, transmitter, receiver, and antenna (ref. DD 1494 Preparation Guide). Many of the properties that EMIT requires to define Tx and/or Rx characteristics can be found in the DD-1494 for a particular radio, and these DD-1494's are frequently available and provide a useful source of data for creating radio models in EMIT. In these cases, enabling the Use DD-1494 Mode option in EMIT causes EMIT to change some of the labels on its data fields to correspond to the nomenclature used in the DD-1494. This simplifies the process of copying data from the DD-1494 into EMIT.

In cases where there is a difference in the Properties window due to enabling the DD-1494 mode, both versions of the Properties panel will be shown and explained in this manual.

Modulation: Contains the parameters that define the particular signal modulation associated with the band.



With the Emission Designator unselected (False), the channel bandwidth and modulation is defined by the user.

Channel Bandwidth: The bandwidth of the tuned channel.

Modulation: The modulation type is selected via the drop-down Modulation menu. The particular parameters shown depend on the modulation type selected.

For Rx channels, the selected modulation type affects the Rx Spectral Profile in only two ways:

- For digital modulation types, it provides a processing gain parameter to be entered in the Rx Spectral Profile properties panel.
- For AM modulation, the provided modulation index is used to compute the Rx susceptibility as follows:

$$P_{sb} = P_{rcvd}\left(1 - \frac{1}{1 + 0.5m^2}\right)\ [W]$$

$$S_{rx} = P_{sb} - SNR + 3\ [dB]$$

where

$P_{rcvd} = Received\ signal\ power\ level\ as\ specified\ by\ the\ user$

$m = modulation\ index$

$P_{sb} = Power\ in\ a\ single\ sideband$

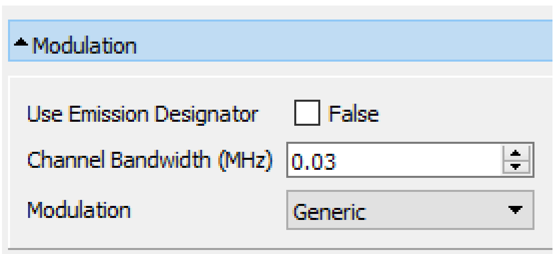
$SNR = Required\ signal - to - noise\ ratio\ as\ specified\ by\ the\ user$

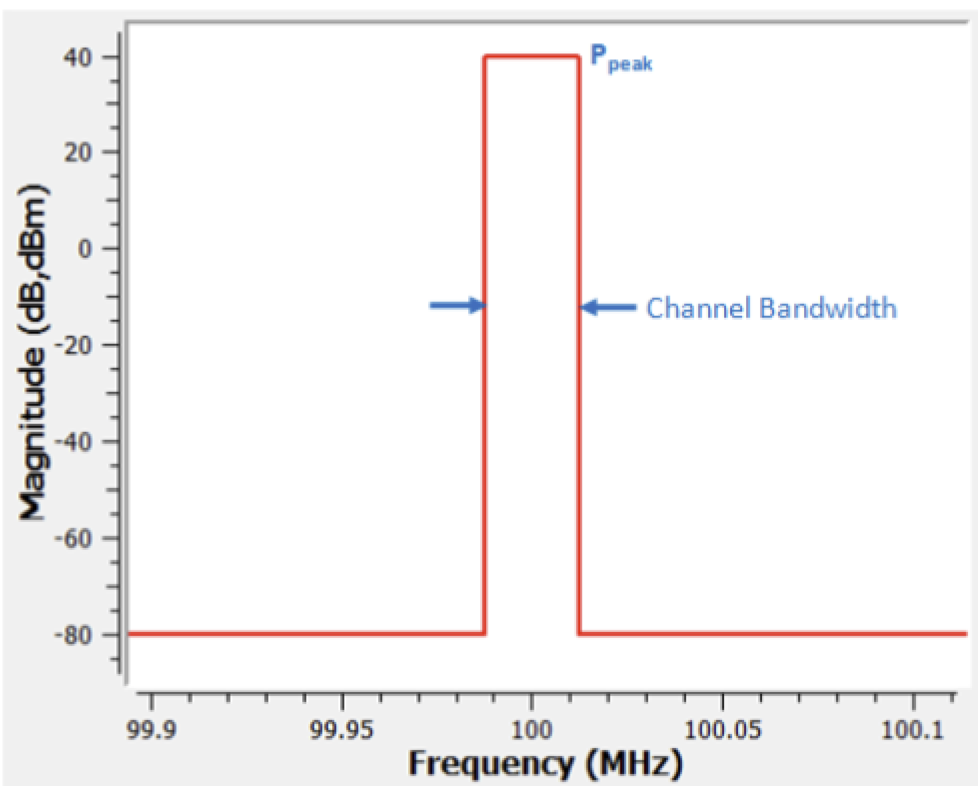
$S_{rx} = Receiver\ in - band\ susceptibility$

For Tx channels, the shape of the fundamental emission (as well as the harmonics) is determined by the specifics of the particular modulation being used. EMIT offers a menu of modulations and the parameters associated with the modulation scheme change when different modulations are selected. EMIT approximates the spectral shape for each modulation type using a worst-case step-wise representation. The details on the specific modulation types currently supported in EMIT and their associated properties are provided in the following. Further details regarding Tx emissions can be found in the Tx Spectral Profile section of this documentation.

Generic

Ideal, constant level spectrum across the specified channel bandwidth.

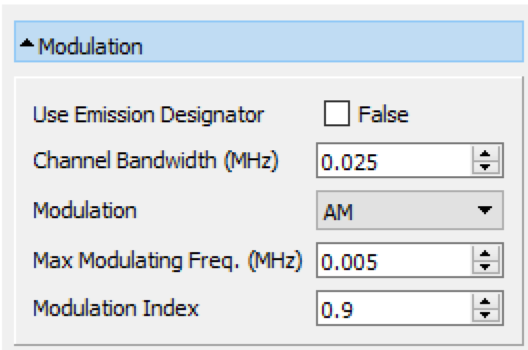


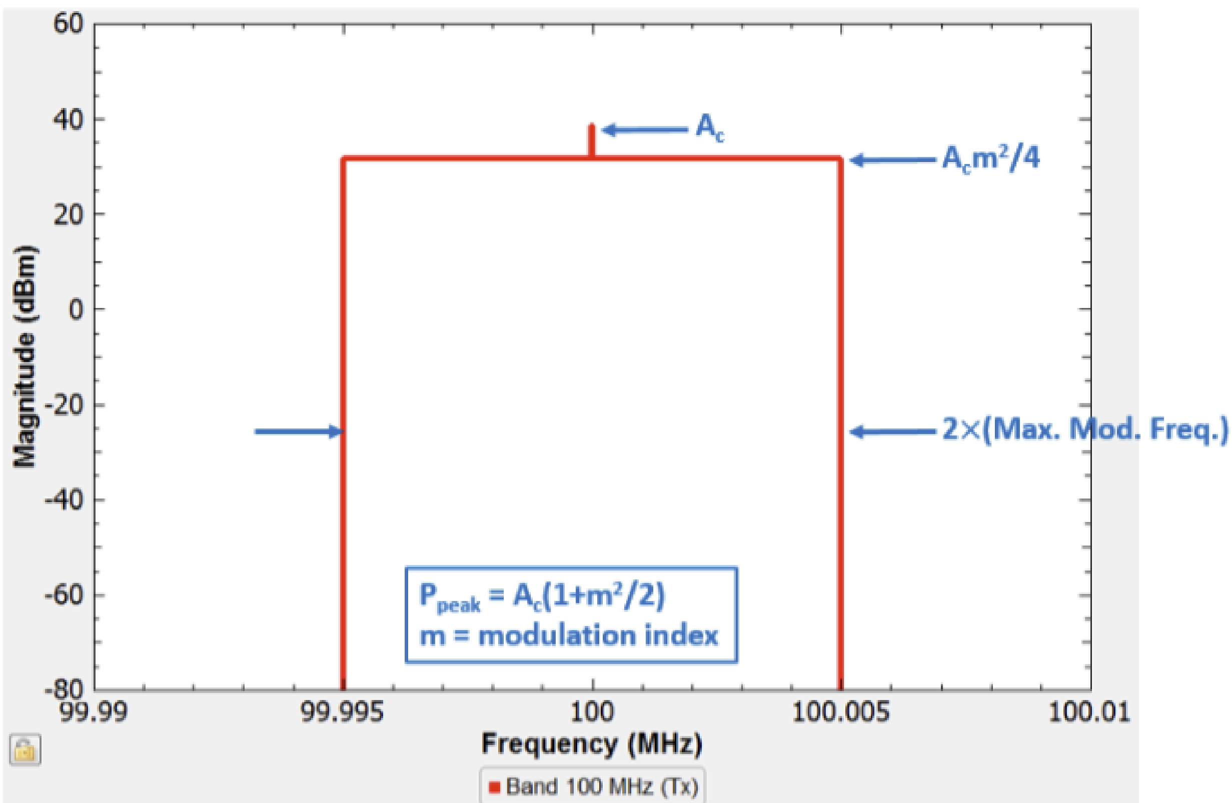


Channel Bandwidth: The bandwidth of the tuned channel.

AM

Amplitude Modulation. In the figure below, A_c represents the carrier power level.





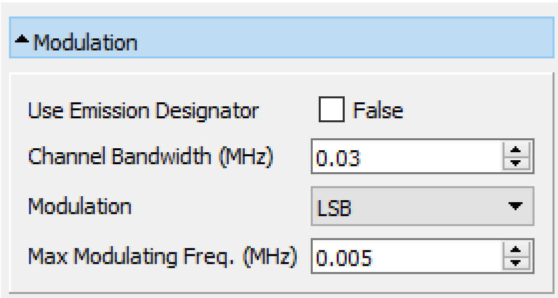
Channel Bandwidth: The bandwidth of the tuned channel.

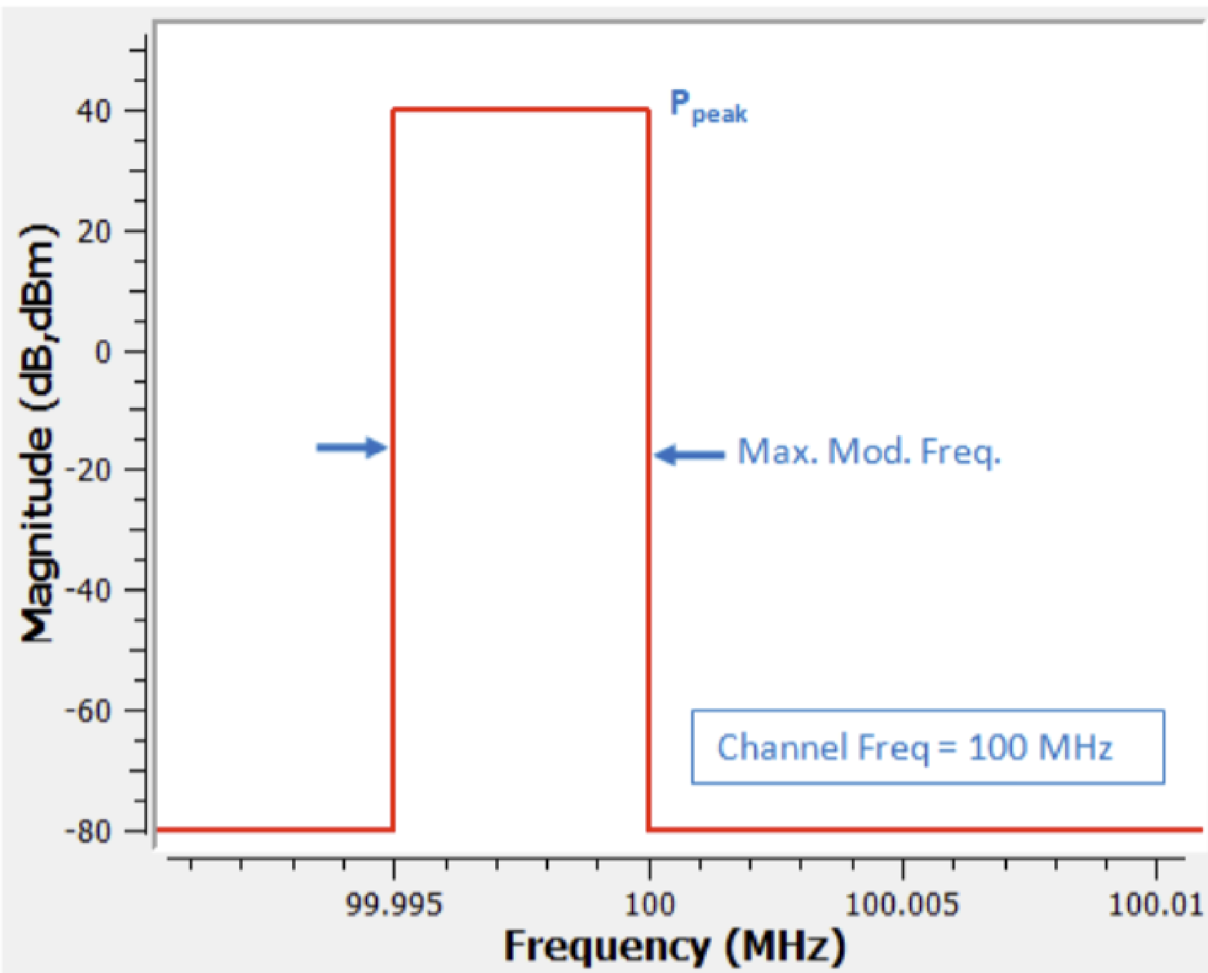
Max Modulating Freq. (MHz): The highest frequency of the modulating waveform.

Modulation Index: The AM modulation index.

LSB

Lower Side Band. Spectrum is below the suppressed carrier (channel) frequency.



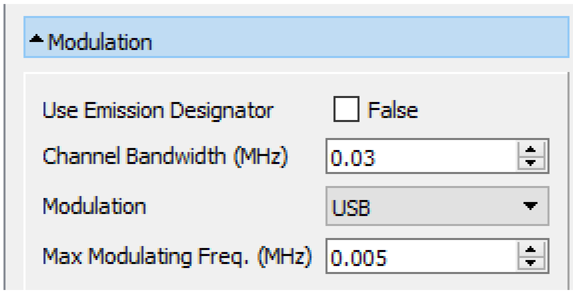


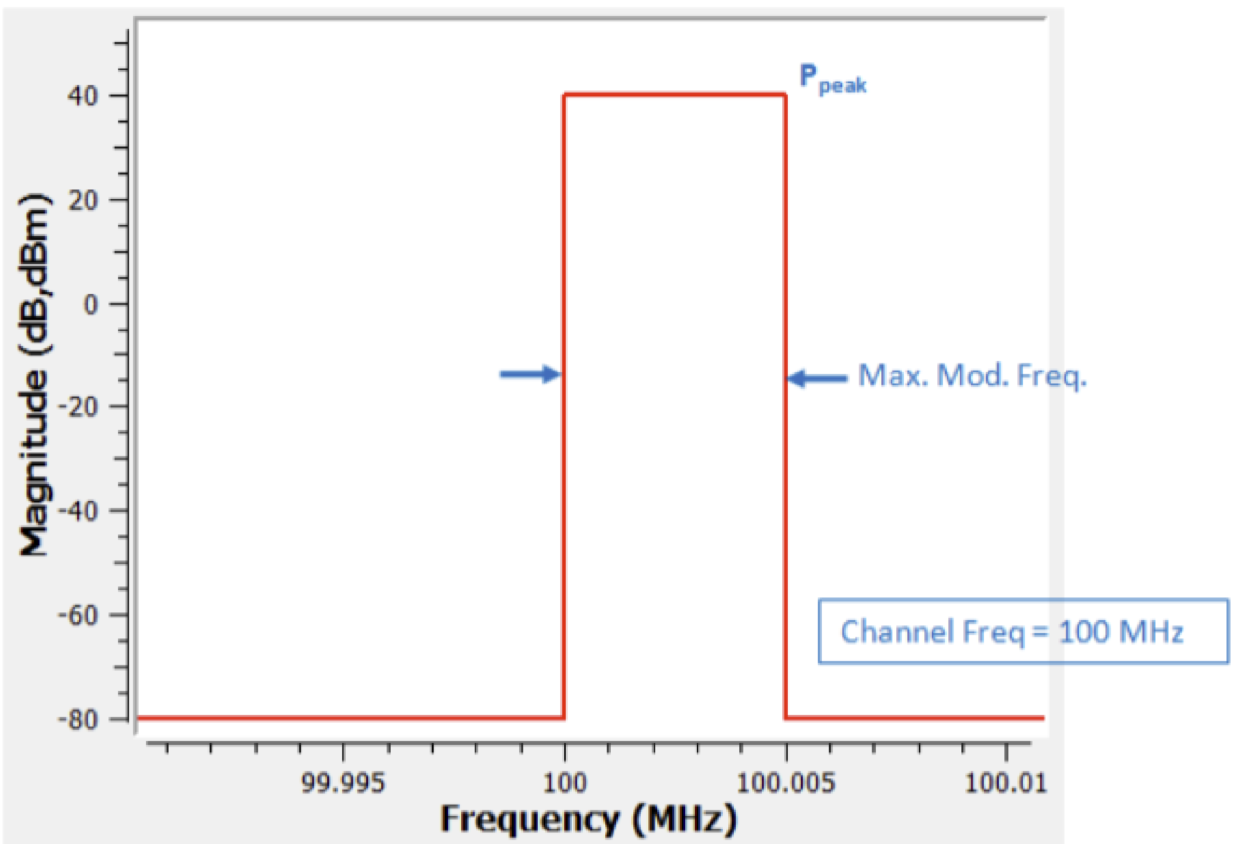
Channel Bandwidth: The bandwidth of the tuned channel.

Max Modulating Freq. (MHz): The highest frequency of the modulating waveform.

USB (Upper Side Band)

Spectrum is above the suppressed carrier (channel) frequency.



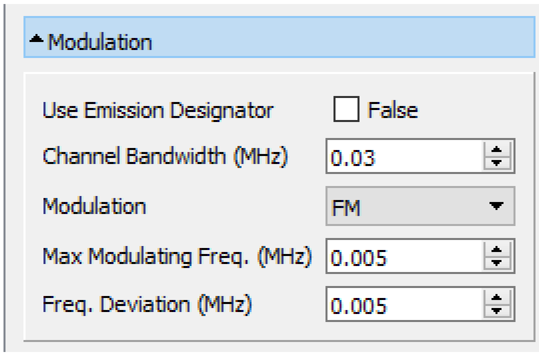


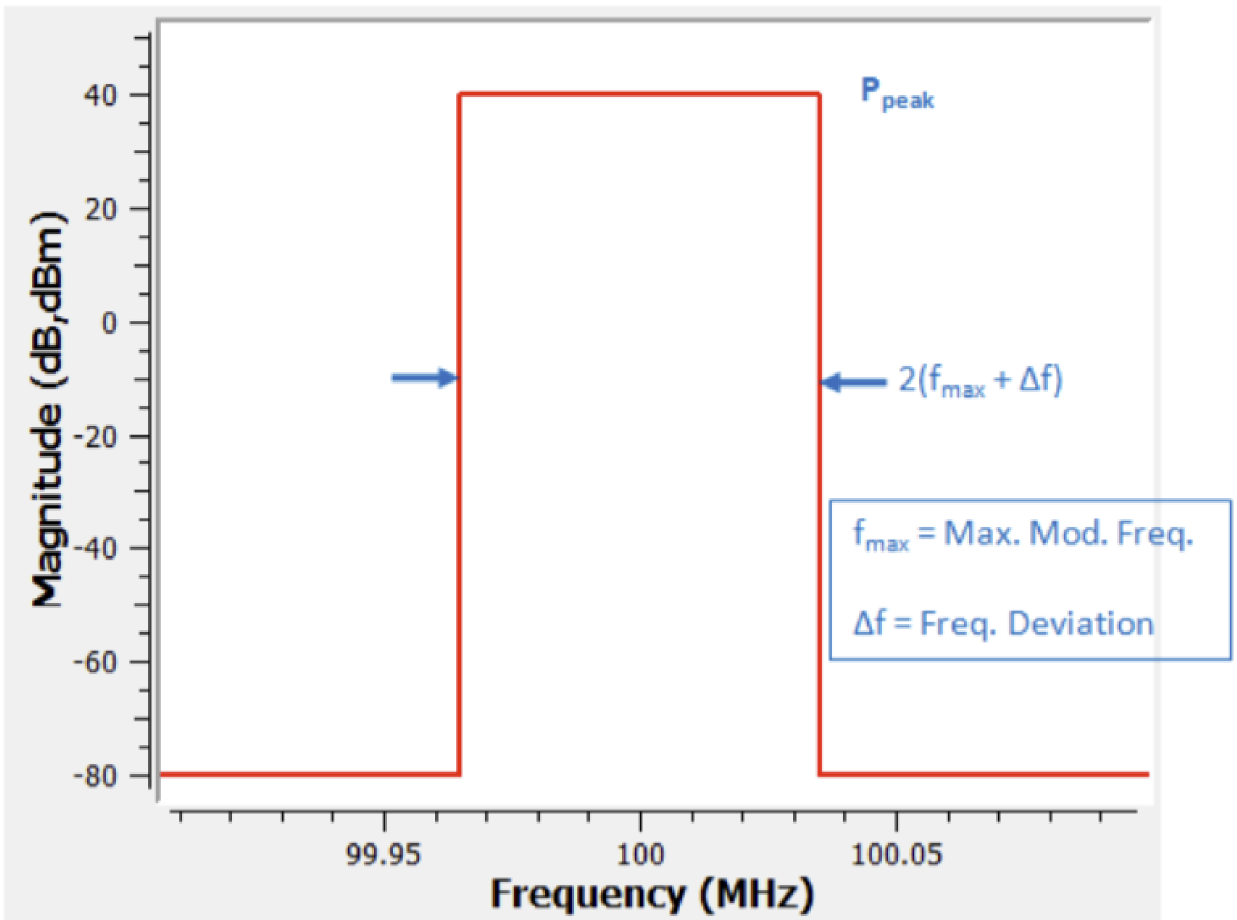
Channel Bandwidth: The bandwidth of the tuned channel.

Max Modulating Freq. (MHz): The highest frequency of the modulating waveform.

FM

Frequency Modulation. Bandwidth based on Carson's rule.





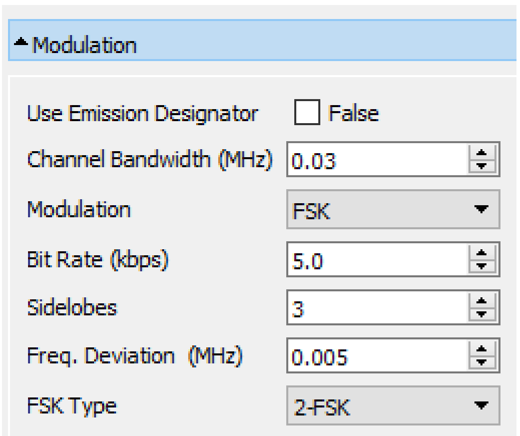
Channel Bandwidth: The bandwidth of the tuned channel.

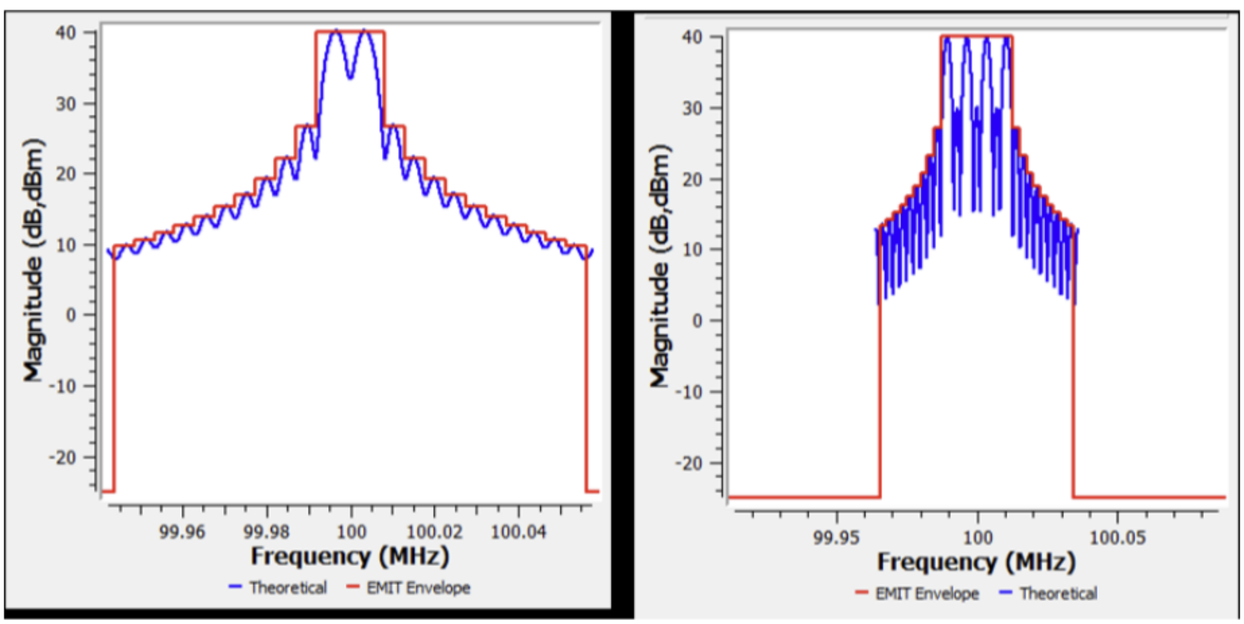
Max Modulating Freq. (MHz): The highest frequency of the modulating waveform.

Freq. Deviation (MHz): FM frequency deviation.

FSK

Frequency Shift Keying. Worst-case envelope computed by EMIT as shown below for 2-FSK (left) and 4-FSK (right). After the specified number of sidelobes the spectrum drops to the noise floor, in this example defined as -25 dBm.





Channel Bandwidth: The bandwidth of the tuned channel.

Bit Rate (kbps): Data rate of the baseband digital signal.

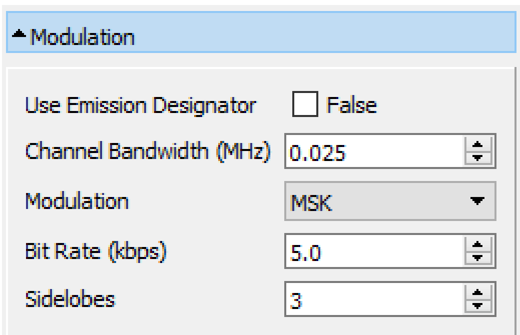
Sidelobes: Number of sidelobes to include in the Tx spectrum before dropping to the noise floor.

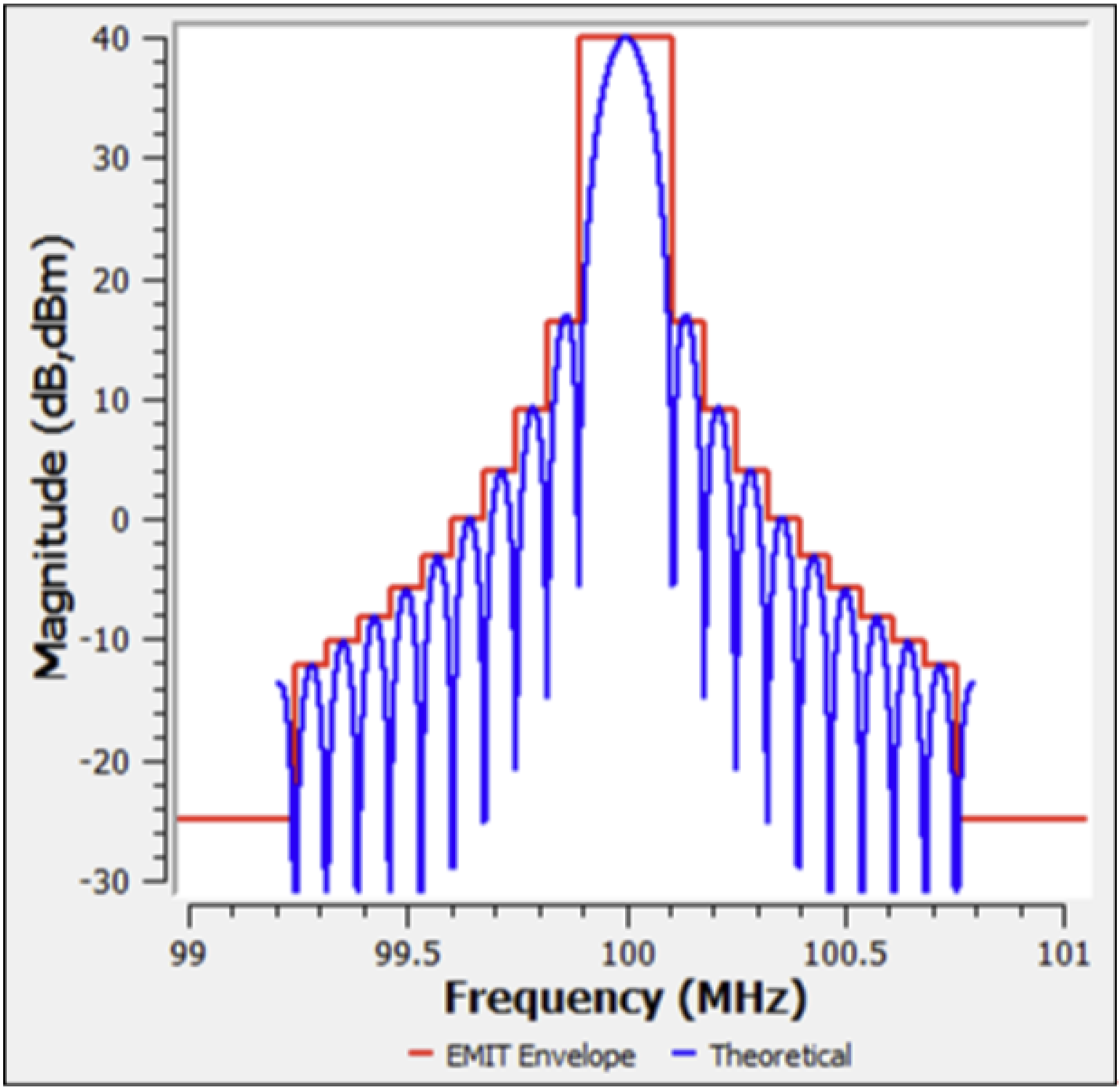
Freq. Deviation (MHz): Frequency deviation.

FSK Type: Specific type of FSK modulation desired.

MSK

Minimum Shift Keying. Worst-case envelope computed by EMIT as shown below. After the specified number of sidelobes, the spectrum drops to the noise floor, in this example defined as -25 dBm.





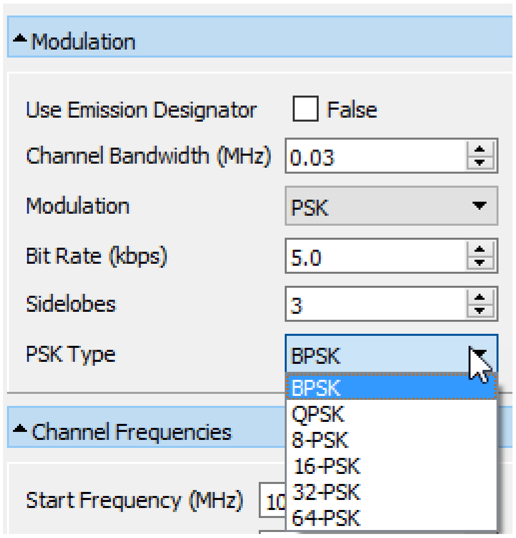
Channel Bandwidth: The bandwidth of the tuned channel.

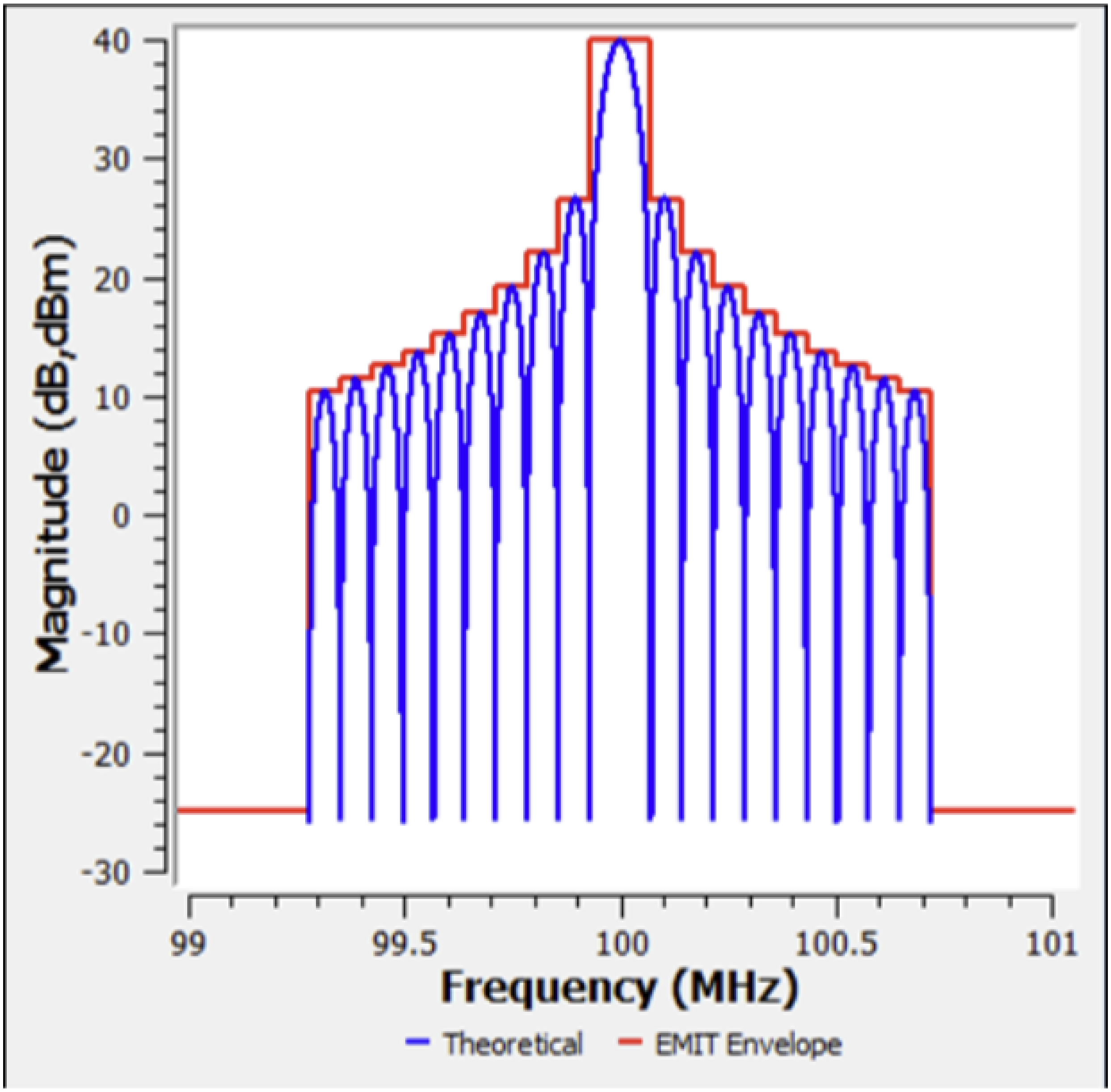
Bit Rate (kbps): Data rate of the baseband digital signal.

Sidelobes: Number of sidelobes to include in the Tx spectrum before dropping to the noise floor.

PSK

Phase Shift Keying. Worst-case envelope computed by EMIT as shown below. After the specified number of sidelobes, the spectrum drops to the noise floor, in this example defined as -25 dBm.


Modulation
Use Emission Designator
False
Channel Bandwidth (MHz)
0.03
Modulation
PSK
Bit Rate (kbps)
5.0
Sidelobes
3
PSK Type
BPSK
BPSK
QPSK
8-PSK
16-PSK
32-PSK
64-PSK
Channel Frequencies
Start Frequency (MHz)



Channel Bandwidth: The bandwidth of the tuned channel.

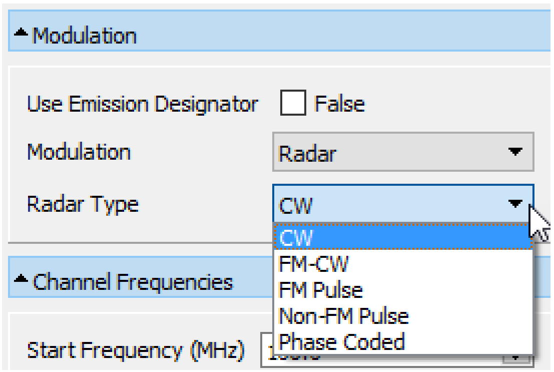
Bit Rate (kbps): Data rate of the baseband digital signal.

Sidelobes: Number of sidelobes to include in the Tx spectrum before dropping to the noise floor.

PSK Type: Specific type of PSK modulation desired.

Radar

The spectral masks are calculated using the Radar Spectrum Engineering Criteria (RSEC) defined by the National Telecommunications and Information Administration (NTIA) in the "Manual of Regulations and Procedures for Federal Radio Frequency Management." A copy of the manual can be obtained from the NTIA's website (NTIA Manual of Regulations).



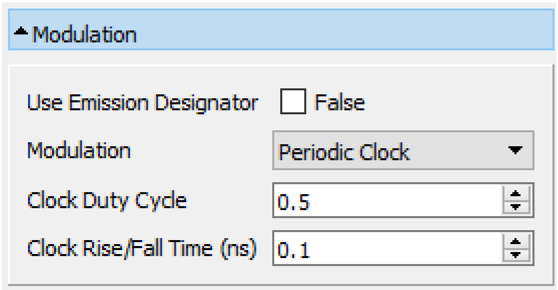
The RSEC specifies various types of radars (CW, FM-CW, FM Pulse, Non-FM Pulse, and Phase Coded) as well as multiple "Groups" (or Criteria). The specific group that a radar belongs to is primarily determined by its function, operating frequencies, and peak or average power levels. Based on the type of radar and the group that it belongs to, the RSEC then specifies the emission mask limits of the transmitted spectrum. EMIT computes the in-channel region of the radar spectrum based off the 40-dB bandwidths specified by the RSEC. The broadband noise level is then defined based off the specified slope from the 40-dB bandwidth points down to the X-dB (ultimate suppression level) bandwidth points.

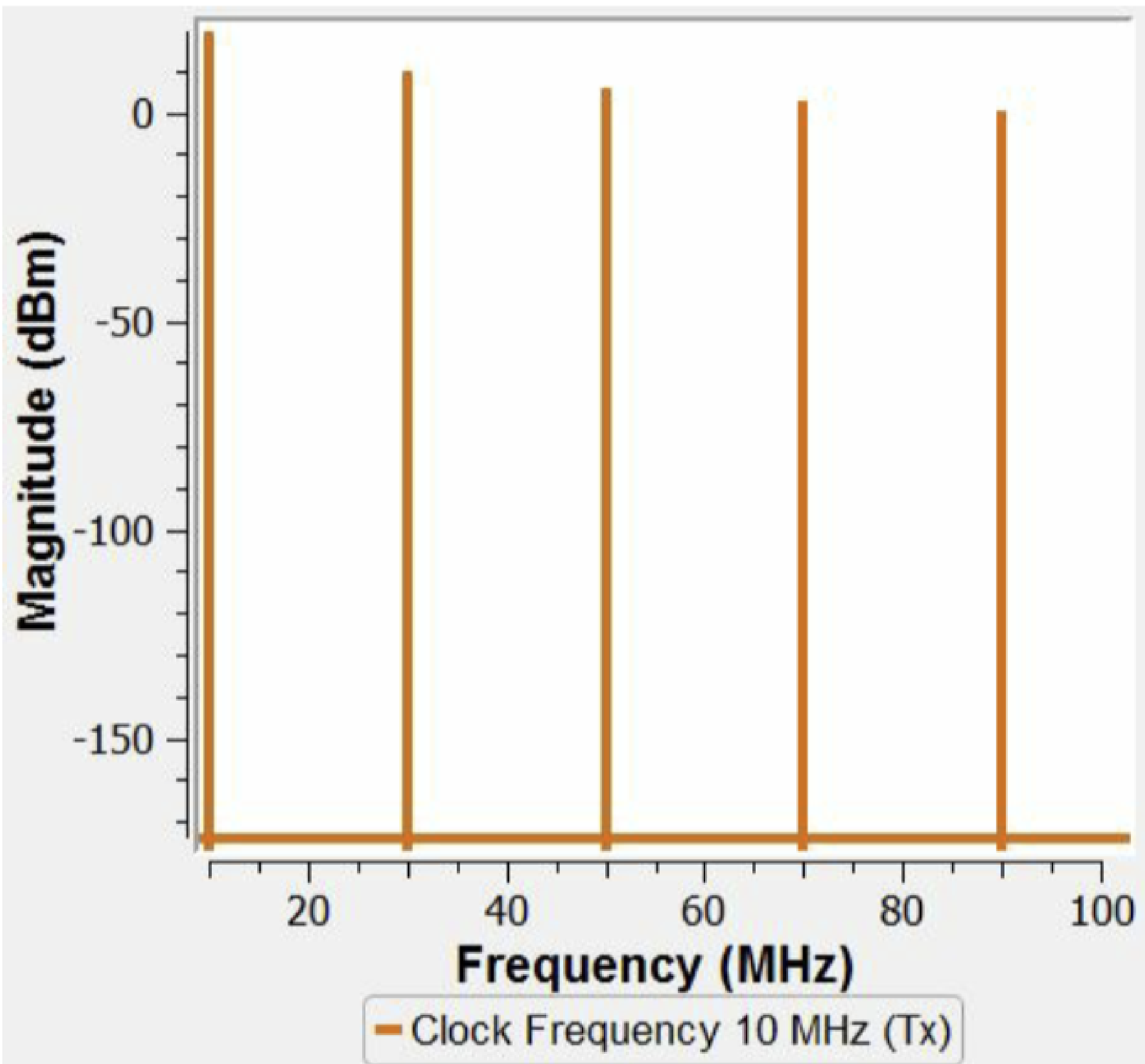
Note:

In some cases the RSEC does not fully define the emission mask limits for a specific radar type and group. For these cases, EMIT defaults to the limits specified for Group C radars, which in general are 'All radars not included in Group A, B, D, or E.

Periodic Clock

The spectrum consists of the fundamental clock frequency and its harmonics. If the Duty Cycle = 0.5, then only the odd harmonics are produced.





Clock Duty Cycle: Defines the clock's duty cycle. A Duty Cycle = 50% (0.5) will suppress the clock's even harmonics.

Clock Rise/Fall (ns): Specifies the rise/fall time of the clock pulses. The rise/fall time impacts the amplitudes of the clock harmonics.

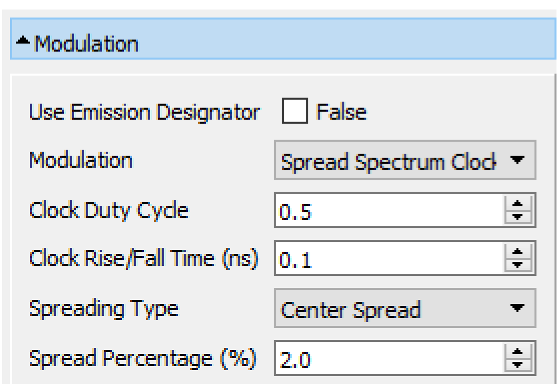
The peak voltage that specifies the clock drive voltage is entered in the Tx Spectral Profile settings.

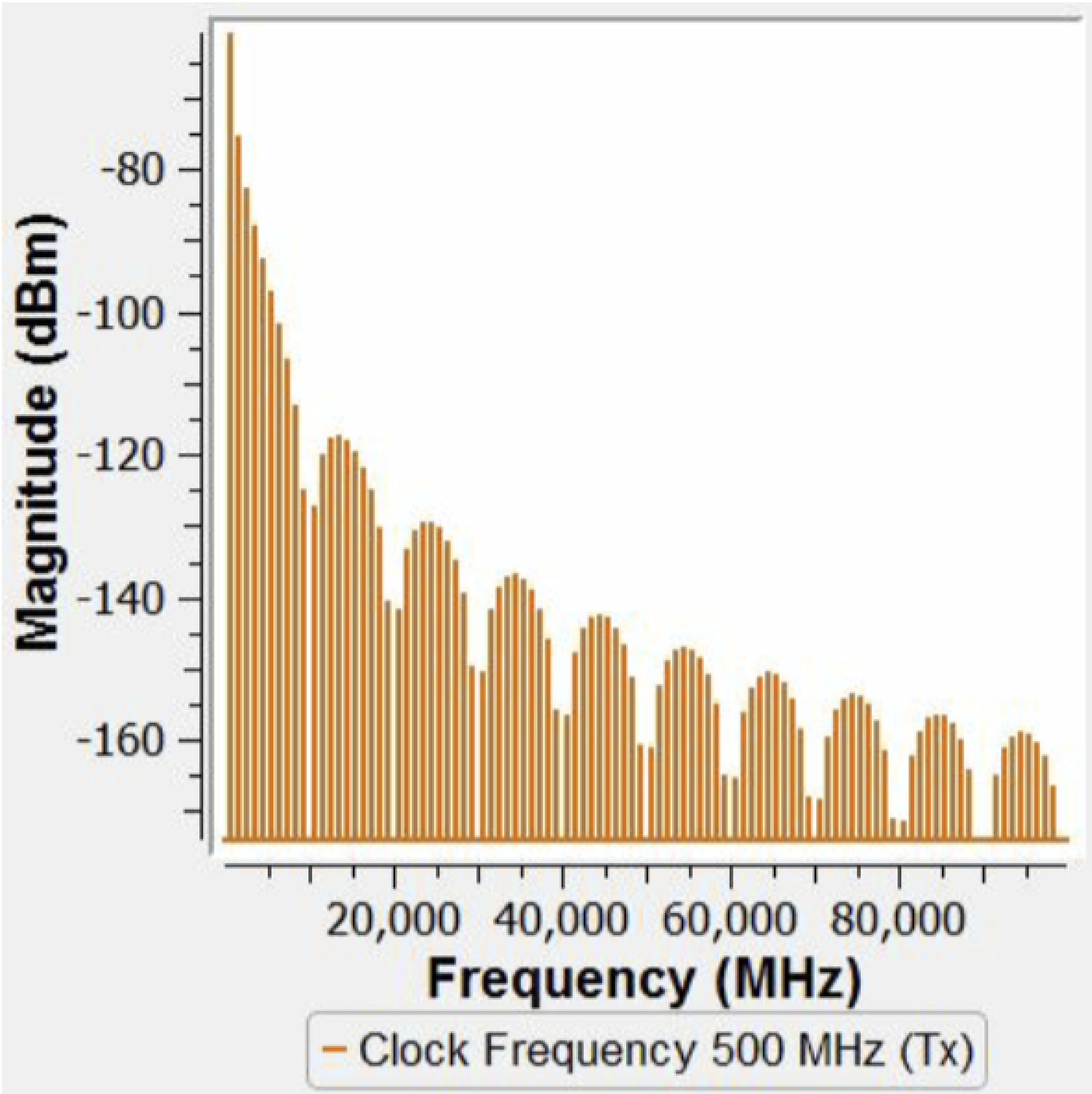
Spread Spectrum Clock

The Spread Spectrum Clock (SSC) differs from the Periodic Clock in a key way. It is represented in EMIT as a broadband power spectral density (PSD) rather than as a narrowband spectrum. This is due to the spreading nature of the clock. As seen below, the envelope of the spectrum is that of a sinc(x) wave.

> **Note:**
>
> Since the SSC signal type is modeled as a broadband PSD, plots of the spectrum are in units of dBm/Hz (despite "dBm" being displayed).

Spread Spectrum Clocking can significantly reduce the peak EMI levels radiated by a system clock. The reduction in the peak levels relative to the peak EMI of a non-spread clock is shown below. EMIT uses a Resolution Bandwidth of 1 Hz for all plots of the SSC spectrum.





Clock Duty Cycle: Defines the clock's duty cycle. A Duty Cycle = 50% (0.5) will suppress the clock's even harmonics.

Clock Rise/Fall Time (ns): Specifies the rise/fall time of the clock pulses. The rise/fall time impacts the amplitudes of the clock harmonics.

Spreading Type: Specifies the method used to tune the clock frequency over. The tuning ranges, where BW = f_{clock}x(Spreading %), are specified as:

Low Spread: f_{clock}-BW < f < f_{clock}

Center Spread: f_{clock}-(BW)/2 < f < f_{clock} +(BW)/2

High Spread: f_{clock}< f < f_{clock} +BW

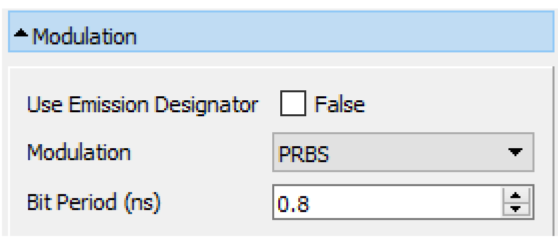
Spread Percentage (%): Determines the bandwidth over which the clock frequency is tuned.

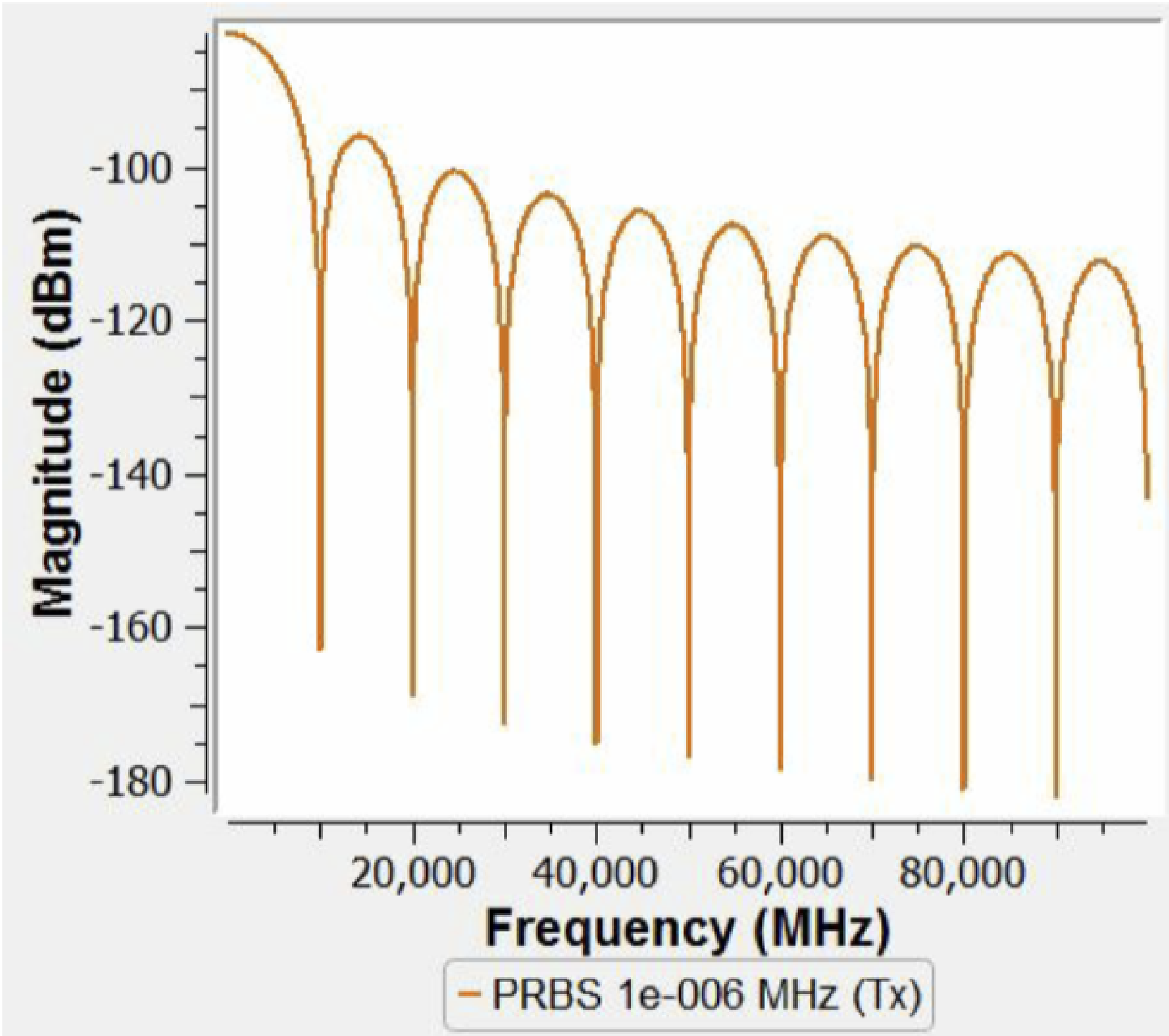
The peak voltage that specifies the clock drive voltage is entered in the Tx Spectral Profile settings.

PRBS

The Pseudo-Random Bit Sequence (PRBS) spectrum is represented in EMIT as a broadband power spectral density rather than as a narrowband spectrum and it defines the spectrum of a random bit stream. As seen below, the envelope of the spectrum is that of a sinc(x) wave. As with the SSC signal type, since the PRBS signal type is modeled as a broadband PSD, plots of the spectrum are in units of dBm/Hz (despite "dBm" being displayed).

> **Note:**
>
> Due to the specialized nature of the PRBS signal type, care must be taken when specifying the Channel Frequencies in the Configuration. For a PRBS, the Frequency Range should be set to 1 Hz in order to result in a valid Tx channel.





Bit Period (ns): Binary stream's bit period in nanoseconds. The bit period is equal to the width of a single bit.

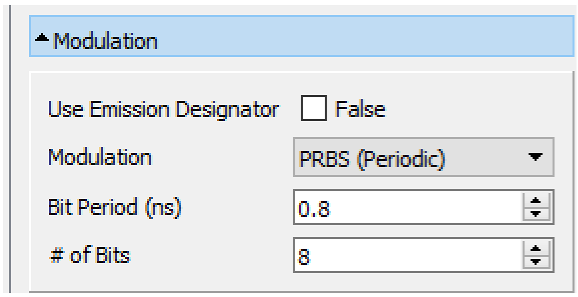
The peak voltage that specifies the clock drive voltage is entered in the Tx Spectral Profile settings.

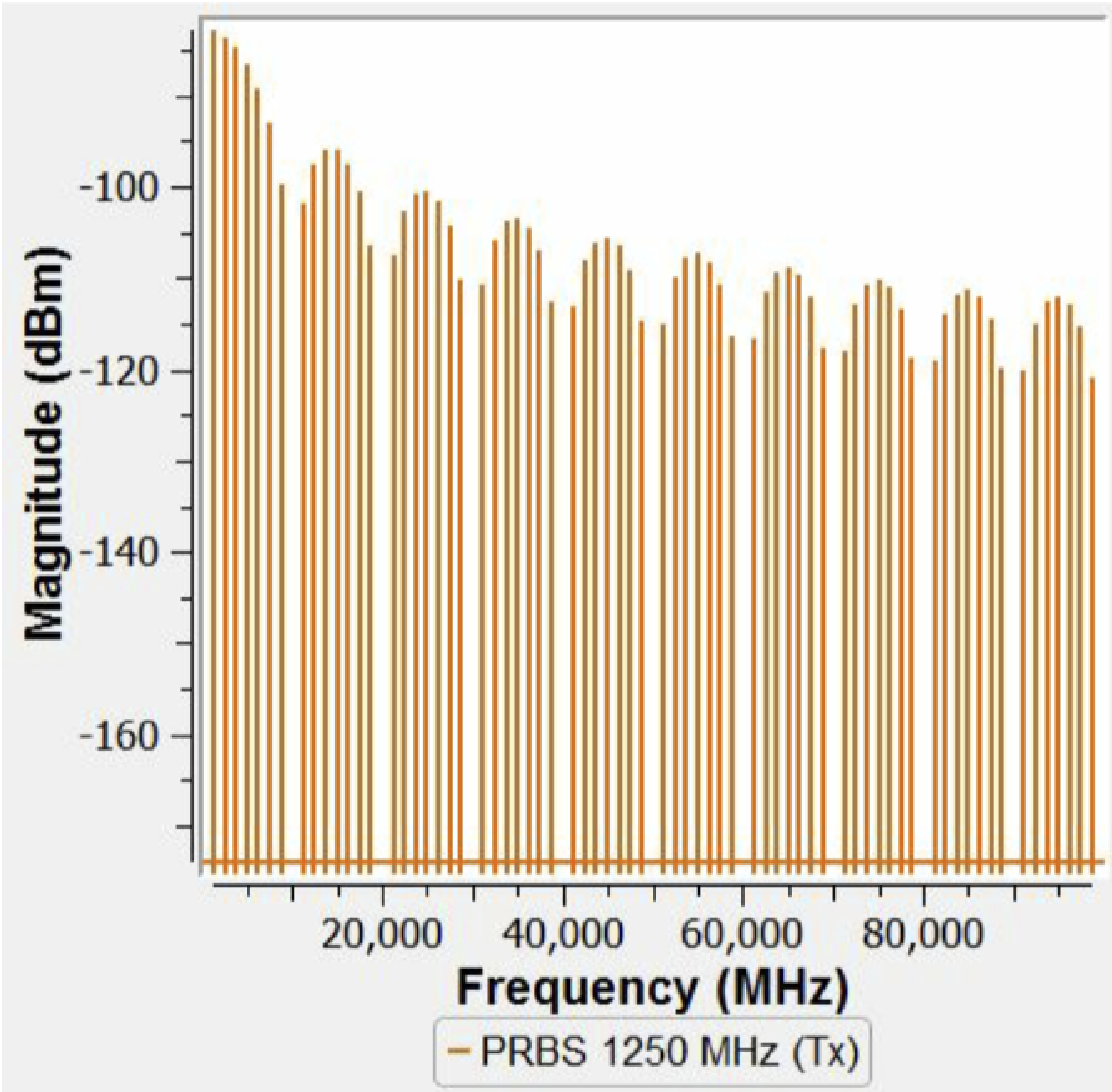
PRBS (Periodic)

A narrowband representation of a repeating (periodic) Pseudo-Random Bit Stream. As seen below, the envelope of the spectrum is that of a sinc(x) wave.

Note:

Due to the specialized nature of the PRBS (Periodic) signal type, care must be taken when specifying the Channel Frequencies in the Configuration. For a PRBS (Periodic) Channel Set, the Frequency Range must be set to (1/Bit Period)/(# of Bits) Hz in order to result in a valid Tx channel.





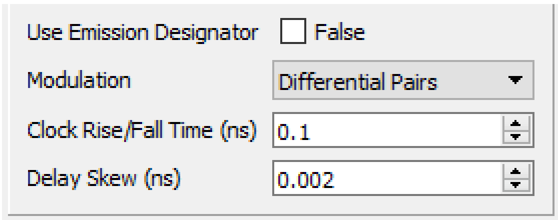
Bit Period (ns): Binary stream's bit period in nanoseconds.

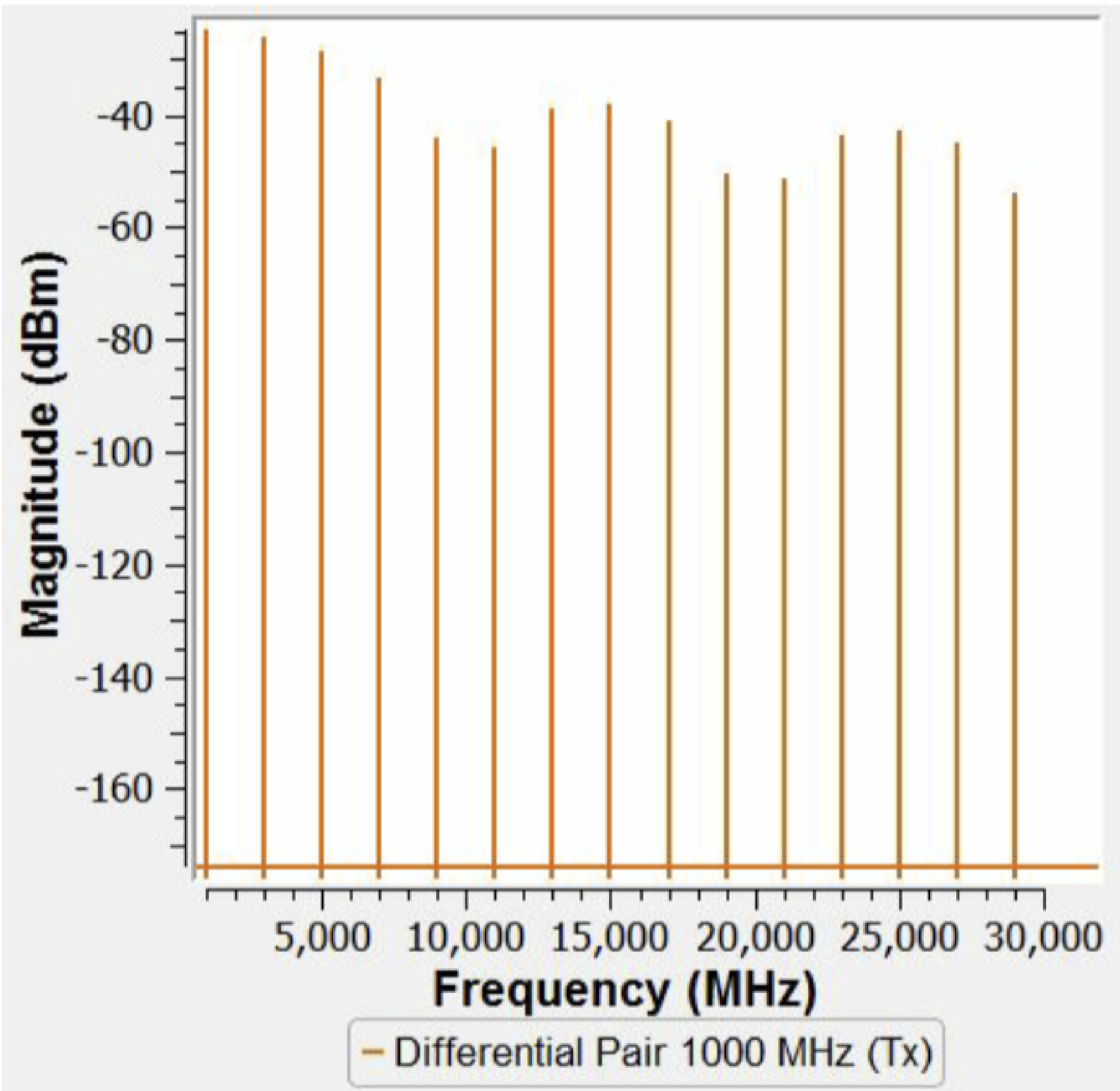
# of Bits: Length of the binary sequence that is repeated.

The peak voltage that specifies the clock drive voltage is entered in the Tx Spectral Profile settings.

Differential Pairs

Models the common mode narrowband spectrum resulting from the delay skew between a differential pair.





Clock Rise/Fall (ns): Specifies the rise/fall time of the clock pulses. The rise/fall time impacts the amplitude's of the clock harmonics.

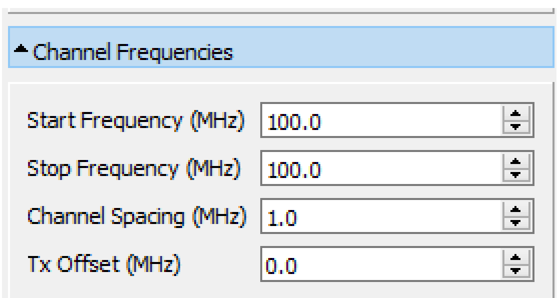
Delay Skew (ps): The time delta when one half of a differential signal does not transition at exactly the same time as the other half which can be caused by one of the signal source switching at a slightly different time or different trace or wire lengths between the pair.

The peak voltage that specifies the clock drive voltage is entered in the Tx Spectral Profile settings.

Channel Frequencies

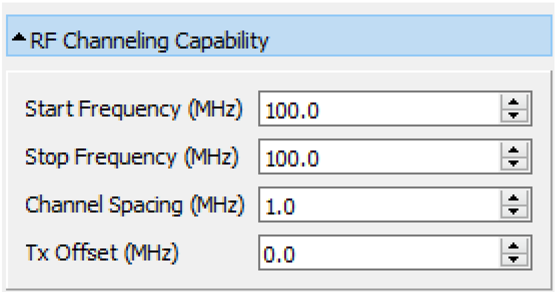
Defines the discrete channel frequencies for the band. The channels begin at the **Start Frequency** and increase by the **Channel Spacing** up to the **Stop Frequency**. Each band in EMIT is limited to 100,000 total channels. If more than 100,000 channels are required, then the band should be duplicated with the frequency range divided between the two bands. Note that both bands can still be assigned to the same radio port in this situation.

A Tx Offset can be specified in which case the Tx channels will be offset by this amount from the Rx channel frequencies as specified by the Start and Stop frequencies.

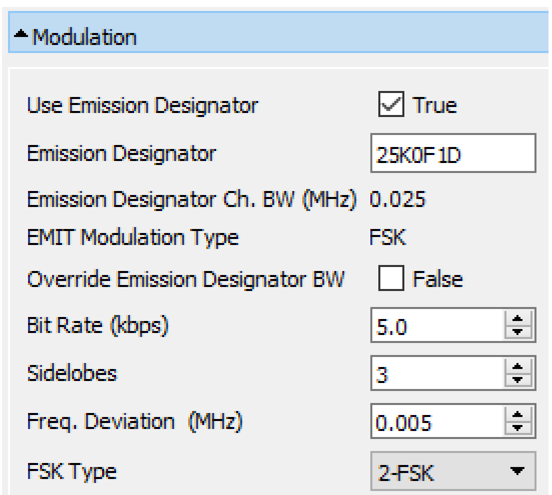


RF Channeling Capability (in DD-1494 Mode): Defines the discrete channel frequencies for the band. The channels begin at the **Start Frequency** and increase by the **Channel Spacing** up to the **Stop Frequency**. Each band in EMIT is limited to 100,000 total channels. If more than 100,000 channels are required, then the band should be duplicated with the frequency range divided between the two bands. Note that both bands can still be assigned to the same radio port in this situation.

A Tx Offset can be specified in which case the Tx channels will be offset by this amount from the Rx channel frequencies as specified by the Start and Stop frequencies.



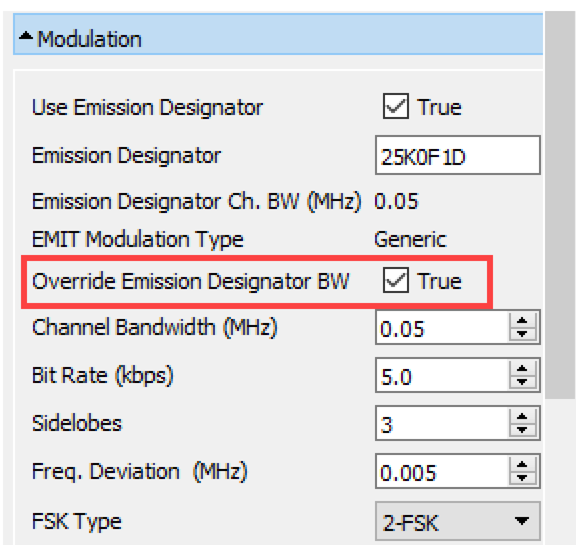
Use Emission Designator: An emission designator utilizes a seven-character "word" to represent the bandwidth, modulation, nature of signal, and type of information transmitted by a particular radio. When this is available, EMIT can determine the modulation parameters that it needs automatically from the emission designator. To do so, **Use Emission Designator** is selected and the designator is entered into the **Emission Designator** field. For example, for the designator given by 25K0F1D, EMIT's Modulation dialog is displayed as:



The table below lists all the supported modulations and the "EMIT modulation" that they are automatically converted to.

Description	Symbol	EMIT Modulation
Emission of an unmodulated carrier	N	Generic
Emission in which the main carrier is amplitude-modulated (including cases where sub-carriers are angle-modulated):		
Double-sideband	A	AM
Single-sideband, full carrier	H	AM
Single-sideband, reduced or variable level carrier	R	AM
Single-sideband, suppressed carrier	J	AM
Independent sidebands	B	AM
Vestigial sideband	C	AM
Emission in which the main carrier is angle-modulated:		
Frequency modulation	F	FM/FSK
Phase modulation	G	PSK
Emission in which the main carrier is amplitude and angle-modulated either simultaneously or in a pre-established sequence	D	Not supported
Emission of pulses:	P, K, L, M, Q, V	Not supported
Cases not covered above, in which an emission consists of the main carrier modulated, either simultaneously or in pre-established sequence, in a combination of two or more of the following modes: amplitude, angle, pulse	W	Not supported
Cases not otherwise covered	X	Not supported

Where the emission designator has been used to determine the associated channel bandwidth and modulation type. In some cases, the user may wish to change the channel bandwidth as determined from the emission designator and this can be done by selecting the **Override Emission Designator BW** option, in which case the bandwidth entered will override that defined by the emission designator. In the example shown below, a channel bandwidth of 50KHz will be used by EMIT instead of the 25KHz defined by the emission designator (see above).

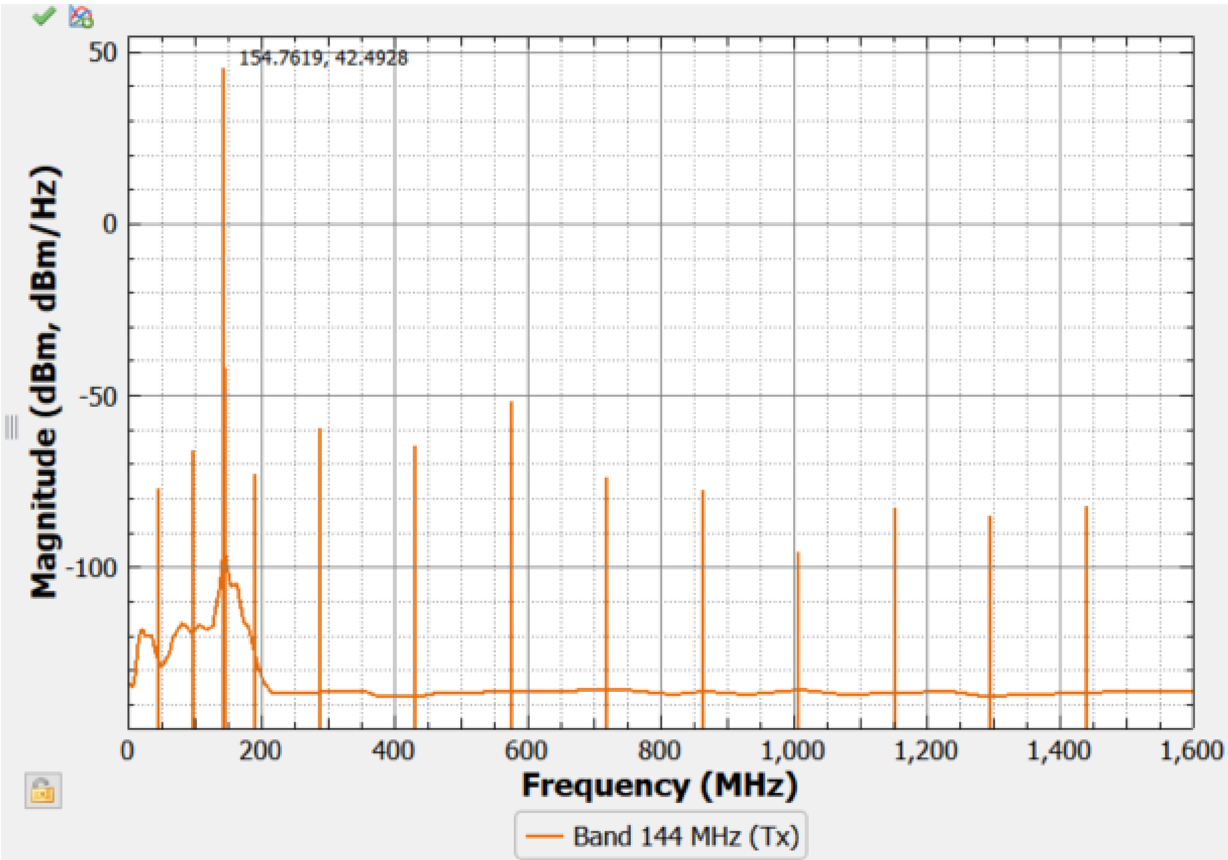


Measurements

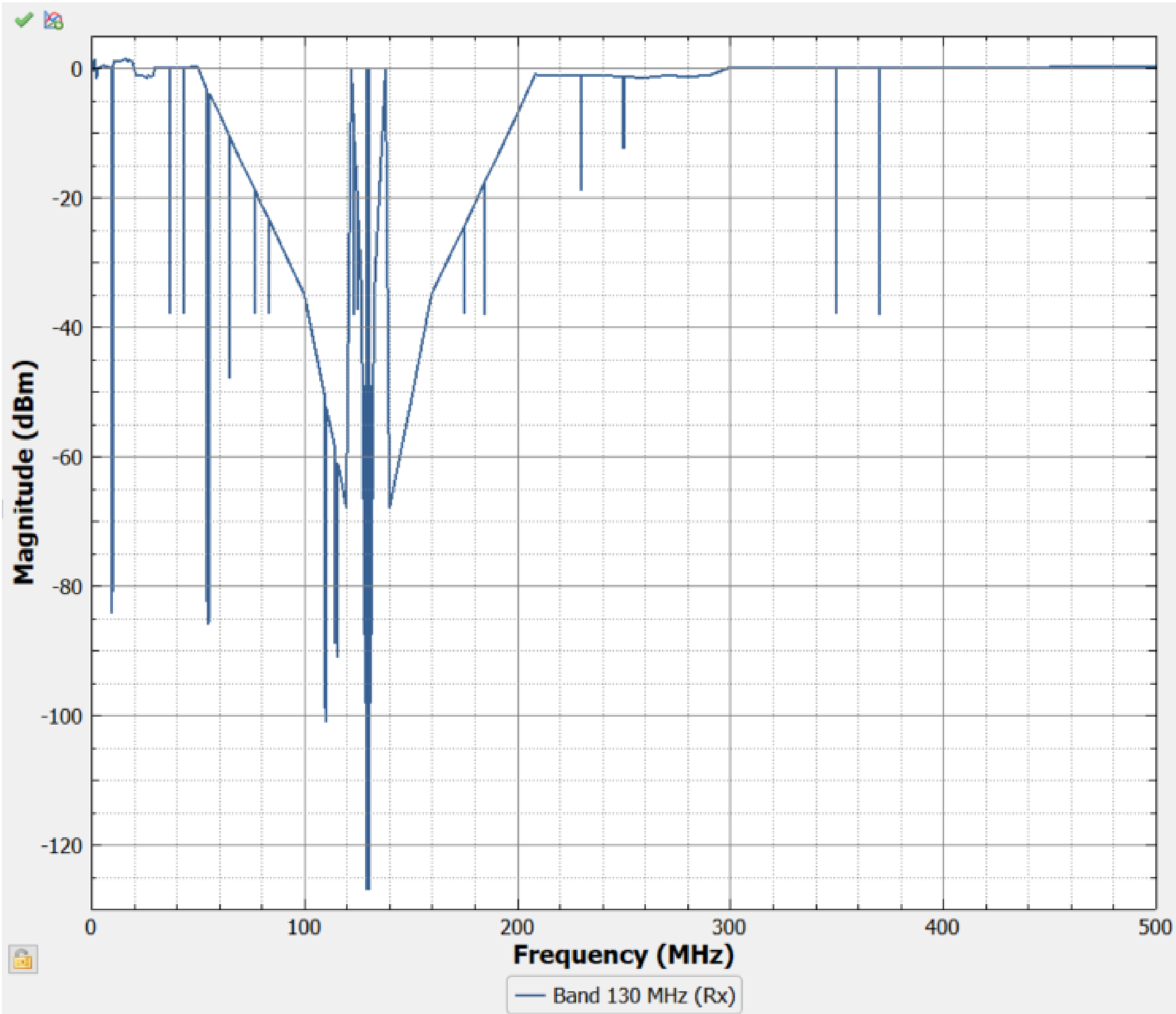
Measurements in EMIT define channels for the RF System and radio that they are associated with. The measurements appear as children of the Band and can supplement or replace the parametric channel definitions specified by the band. A Band can have a Measurement for each channel in the band. If a measurement is present for at least one channel, EMIT will extrapolate the measured characteristics to other channels in the Band. For non-measured channels that fall between two measurements, the channel characteristics (that is, fundamental, harmonics, and broadband noise) are determined by blending the measured data together. The blended data is a weighted combination of the two measurements with the closer, in frequency, measurement channel getting a larger weight. Note that spurs are displayed only for the measured channels. Additionally, if one measurement file defines 10 harmonics (or mixer products) and the other file defines only 5 harmonics (or mixer products), then the amplitude and shape from the first file are used for the harmonics (or mixer products) not defined in the second file. Measurements are added to a Band by importing a measurement file.

A Tx Measurement defines one or more parts of the emission spectrum for a single channel within the Tx Band. Likewise, a Rx Measurement defines one or more aspects of the susceptibility spectrum for a single receive channel within the Rx Band. In EMIT, computed spectra which incorporate measurement data are referred to as measured channels.

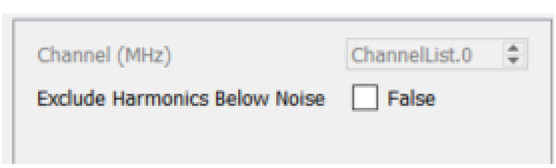
An example measured Tx channel spectrum is shown below. The spectrum consists of the fundamental (carrier) and its harmonics, as well as other non-harmonic spurious emissions (spurs). It can be seen that the measurement provides modeling fidelity beyond that of parametric channels with customized harmonic levels, spurs, and noise profile. The bandwidth and shape of the fundamental and harmonics as well as the noise profile depend on the measurement data.



An example measured Rx channel spectrum is shown below. The spectrum consists of the tuned RF channel, its intermediate frequency, its mixer products and other spurious responses (spurs). It is clearly seen that the measurement provides modeling fidelity significantly beyond that of parametric models.



Measurement data can be imported using one of the RF System Characteristic Files or as Measurement Files. Note that when *.dlxch files are imported, the data is automatically converted and stored as a *.meas file in the project directory. The property panel displays key information that has been imported from the file.



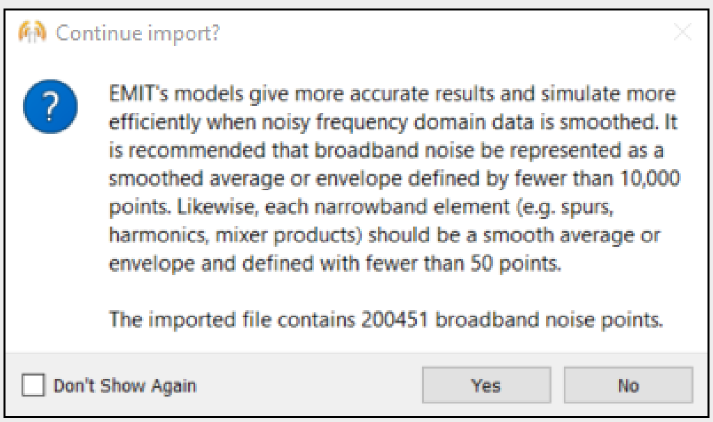
Channel (MHz): This is the center frequency of the channel represented by the measurement.

Exclude Harmonics Below Noise: When a harmonic level is indistinguishable from the measurement noise, the conservative assumption must be that the harmonic level is equivalent to the measurement noise level. Advanced measurement systems will detect this condition and flag the harmonic that was below the noise floor. Setting the **Exclude Harmonics Below Noise** checkbox causes EMIT to ignore harmonics which the measurement system indicated were below the noise. The harmonic level then defaults to the parametric harmonic model specified in the Band. When this checkbox is unchecked, EMIT uses the measured harmonic level (the noise level at the harmonic frequency).

Note:

Ansys recommends that broadband noise be represented as a smoothed average or envelope defined by fewer than 10,000 points. Similarly, each narrowband component (e.g., fundamental, spurs, harmonics and mixer products) should be a smooth average or envelope defined by fewer than 50 points. It is possible to use more points than this, but it can significantly impact the efficiency of EMIT simulations.

If these recommendations are exceeded, EMIT will display the following warning:



Sampling

The *Sampling* node defines the operational channels (frequencies) to be considered in the EMIT simulation.

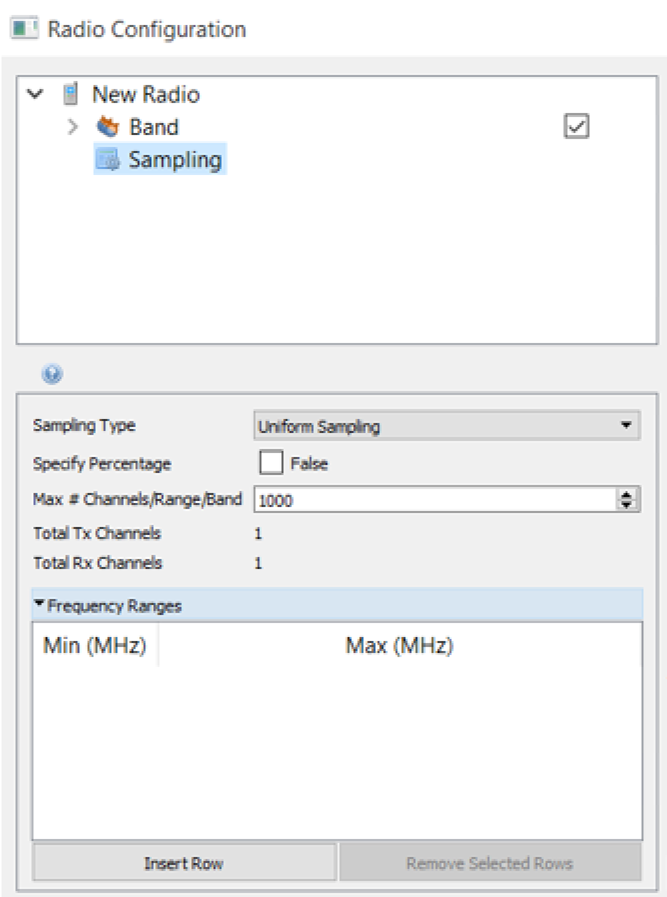
Each Radio in EMIT may be capable of operating on a very large number of bands and channels. In most cases, it is not desired to run the EMIT simulation for all possible available channels that a radio is capable of operating on. There are two primary reasons for this.

- First, in a given situation, a radio may not be used operationally on all available channels. For example, a multi-band radio installed on a vehicle may only be used on one of the bands that it is capable of. In other words, in a given situation it operates only on a subset of the available channels. In this case one would want to analyze only the channels that are to be used, not all possible channels available in the Radio.

- Secondly, many radios are capable of being tuned in very small increments, which can result in a very large number of available channels that the radio can be operated on, and analyzing every single channel can lead to long simulation times in EMIT and an enormous amount of data to pore over after the simulation is complete.

For these reasons it is usually not desirable or necessary to simulate all available channels. The Sampling settings provide a way to define the particular frequency ranges to include in the simulation as well as how to sample the available channels within those ranges.

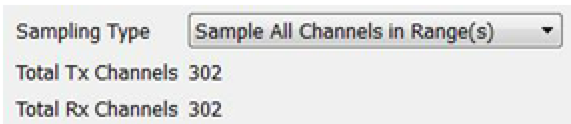
The property settings for Sampling are found in the Radio configuration dialog by selecting the Sampling node as shown below:



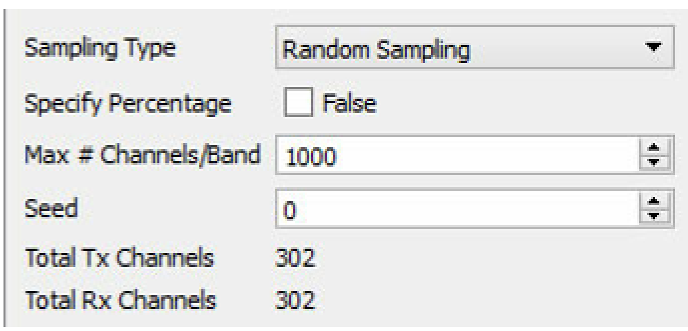
These settings define which channels (frequencies) available in the *Sampling* to include in the EMIT simulation. The way this works is that the *Frequency Ranges* settings define ranges of frequencies to be included. EMIT automatically finds all channels within these ranges that the radio has defined. These channels can all be included in the analysis, or they can be sampled based on the sampling settings defined.

Sampling Type: Defines how the channels available in the defined *Frequency Ranges* are sampled for the EMIT simulation.

Sample All Channels in Range(s): This setting includes all channels within the defined Frequency Ranges. If no Frequency Ranges are defined, EMIT uses all channels available in the Radio's Bands. The total number of Tx and Rx channels that are included in the analysis are shown in the non-editable **Total Tx Channels** and **Total Rx Channels** fields shown below.



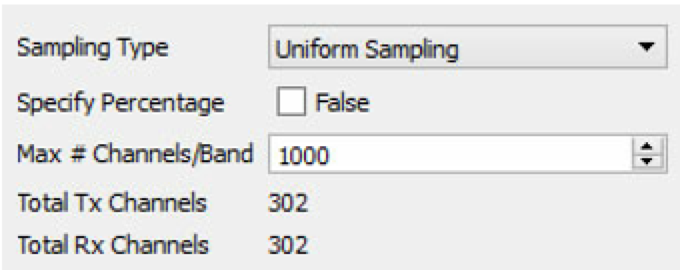
Random Sampling: This setting uses a uniform distribution to generate a random sampling of all channels within the defined *Frequency Ranges*. If no *Frequency Ranges* are defined, EMIT samples from all channels available in the Radio's Bands. The number of channels to be included can be limited by specifying a **Max # Channels/Band** or by defining a percentage of the available channels to include in the random sampling. The **Seed** setting provides an integer random number seed for EMIT to generate the random sampling. This allows the "random" analysis that is performed to be reproduced.



Consider a case where a *Configuration* has a single Band with 100,000 available channels. If we set the **Max # Channels/Band** to 1,000, EMIT includes a random sampling of 1,000 of the available 100,000 channels in the simulation. If we choose **Specify Percentage** and enter 50%, EMIT includes a random sampling of 50,000 of the available 100,000 channels in the simulation.

If the number of available channels is less than the **Max # Channels/Band** specified, all channels are included in the simulation.

Uniform Sampling: This setting includes a uniform (equally spaced in frequency) sampling of all channels within the defined *Frequency Ranges*. If no *Frequency Ranges* are defined, EMIT samples from all channels available in the Radio's Bands. The number of channels to be included can be limited by specifying a **Max # Channels/Band** or by defining a percentage of the available channels to include in the uniform sampling.



Consider a case where a *Sampling* has a single Band with 100,000 available channels. If we set the **Max # Channels/Band** to 1,000, EMIT includes a uniform sampling of 1,000 of the available 100,000 channels in the simulation. If we choose **Specify Percentage** and enter 50%, EMIT includes a uniform sampling of 50,0000 of the available 100,000 channels in the simulation.

If the number of available channels is less than the **Max # Channels/Band** specified, all channels are included in the simulation.

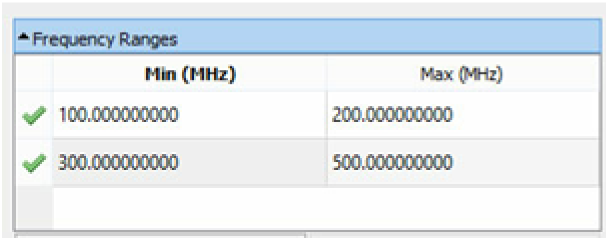
Specify Percentage: If enabled (True), EMIT samples this percentage of the total available channels.

Max # Channels/Band: Maximum number of channels to be included within the defined *Frequency Ranges* for each Band that is enabled for this Radio.

Total Tx Channels: Non-editable value showing the total number of Tx channels to be included in the simulation based on the defined sampling.

Total Rx Channels: Non-editable value showing the total number of Rx channels to be included in the simulation based on the defined sampling.

Frequency Ranges Folder: When expanded, a table is available for defining frequency ranges to be considered for inclusion in the EMIT simulation. EMIT finds and uses all available channels within the defined *Frequency Range*s. If no *Frequency Ranges* are defined, EMIT uses all channels available in the Radio's Bands. In the example shown below, EMIT includes all channels from those available that fall within the 100-200 MHz and 300-500 MHz ranges.



Frequency Ranges

	Min (MHz)	Max (MHz)
✓	100.000000000	200.000000000
✓	300.000000000	500.000000000

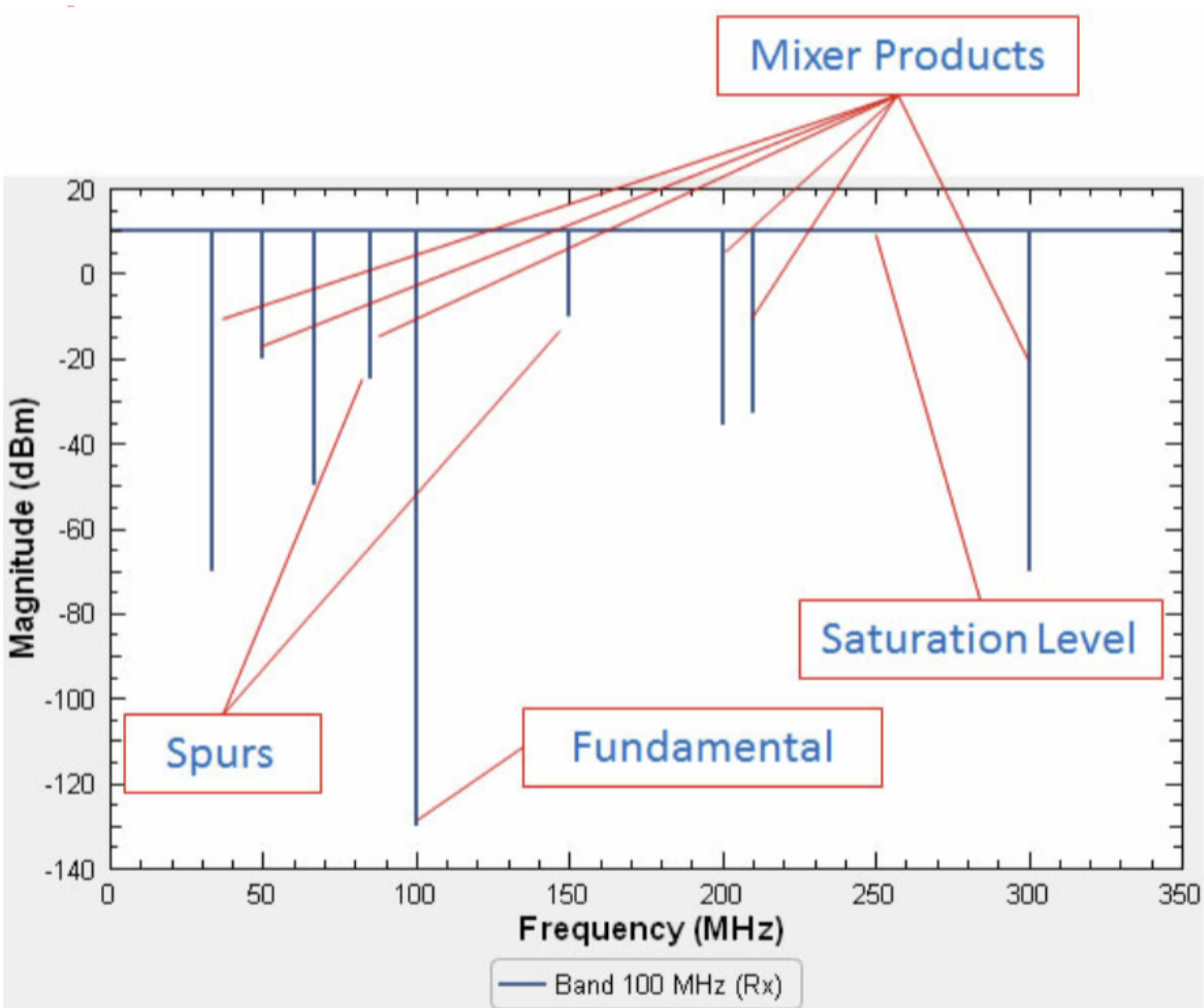
Insert Row: Add a row to the *Frequency Ranges* table.

Remove Selected Rows: Delete selected rows from the *Frequency Ranges* table.

Rx Spectral Profile

A Rx Spectral Profile completely defines the broadband susceptibility spectrum for each channel within the band. Each channel in a band shares the same operating parameters. However, different bands for a single radio may have different operating parameters. The spectra that EMIT computes are based on the parameters supplied for the bands and are referred to as parametric channels.

An example Rx channel spectrum is shown below. The spectrum consists of the fundamental (tuned channel), mixer products and other spurious responses (spurs). The shape of the spurs is taken to be rectangular and the shape of the fundamental is defined by the selectivity (defined below). The specifics on how to specify all of the Rx Band parameters is covered in the sections below.



Sensitivity Units: Power units used for specifying the radio's sensitivity.

Min. Receive Signal Pwr: The amplitude of the desired signal that this Band is attempting to receive. This value is often determined by performing a link budget analysis for a desired communication link. The units are those specified by the Sensitivity Units. It is important to note that this is the power at the Rx antenna port. This Min. Receive Signal Pwr does not get affected by any attenuation present in the Rx signal path between the Rx antenna and Rx input.

Noise Figure (DD-1494 Mode Only): The noise figure of the receiver.

SNR at Rx Signal Pwr (dB): The SNR required for the receiver to decode the desired signal for a given bit error rate (BER). The basic susceptibility model of the Rx Band to in-channel interference is calculated by:

$$S_{rx} = P_{rcvd} - SNR\ [dB]$$

where

P_{rcvd} = Received signal power level as specified by the user
SNR = Required signal-to-noise ratio as specified by the user
S_{rx} = Receiver in-band susceptibility

Processing Gain: The processing gain of a system, often defined as the ratio of the spread bandwidth to the despread (or baseband) bandwidth. The in-channel susceptibility model for an Rx Band with processing gain specified is calculated by:

$$S_{rx} = P_{rcvd} - SNR + PG\ [dB]$$

where

P_{rcvd} = Received signal power level as specified by the user
SNR = Required signal-to-noise ratio as specified by the user
PG = Required processing gain as specified by the user
S_{rx} = Receiver in-band susceptibility

Apply PG to Narrowband Only: By default, EMIT applies the Processing Gain to both the narrowband and broadband noise EMI Margins. If the Apply Processing Gain to Narrowband Only parameter is toggled to true though, then the processing gain will only apply to the narrowband EMI Margins.

SNR/SINAD at Sensitivity: This is equivalent to the SNR at Rx Signal Pwr specified above, with the terminology set to match that found on common DD-1494 forms.

Saturation Level: The maximum value of the Rx susceptibility envelope (see plot above where the saturation level is at 10 dBm).

Perform Rx Intermod Analysis: When enabled (True) this feature includes a nonlinear model internal to the radio to enable Rx generated intermodulation products to be computed. This analysis is included only for the current Band and different nonlinearities can be specified for different Bands of a single Radio. This feature permits a radio's nonlinear characteristics to be included in the radio model instead of requiring it to be modeled as external component as in previous versions of EMIT.

The Rx Internal Amplifier is designed so the nonlinear nature of the receiver's front end can be modeled in a manner similar to EMIT's general amplifiers. However, there are some key restrictions with the Rx Internal Amplifier. The Rx Internal Amplifier will not provide any gain to the signals nor will it add any noise to the spectra incident at the receiver's port. Additionally, the internal amplifiers are assumed to have infinite bandwidth and thus there is no shape factor and no "filtering" of any of the input spectra. Finally, all power incident on the receiver port is assumed to be transmitted through the port resulting in no reflected power, that is, infinite reverse isolation.

Amplifier Saturation Level: Specifies an input power level for which EMIT will consider the amplifier saturated. When an amplifier is saturated, EMIT flags it as such in the results and does not trace the input signal any further. The saturation level cannot be set lower than the 1-dB Point.

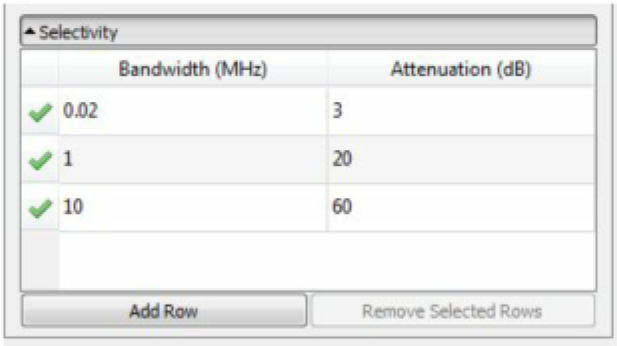
1-dB Point Ref. Input: The amplifier's 1-dB compression point referred to the input of the amplifier.

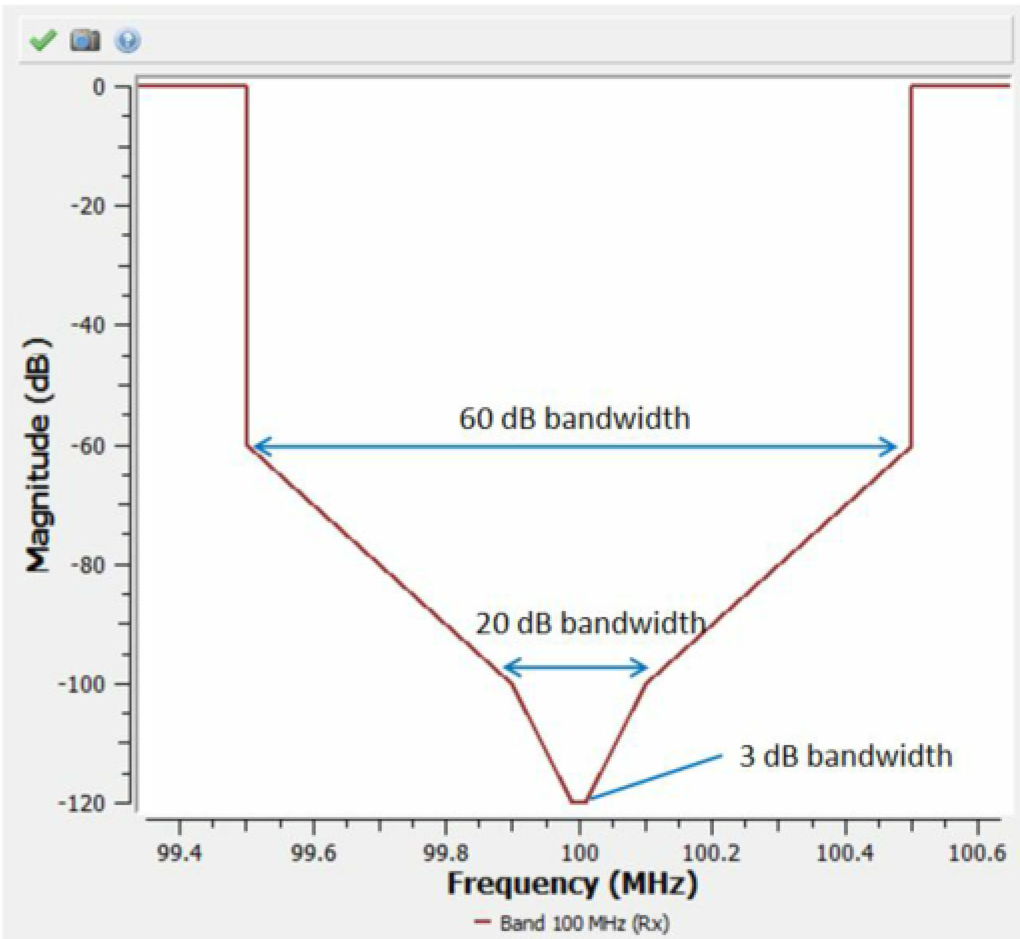
IP3, Ref. Input: The amplifier's third order intercept point referred to the input of the amplifier.

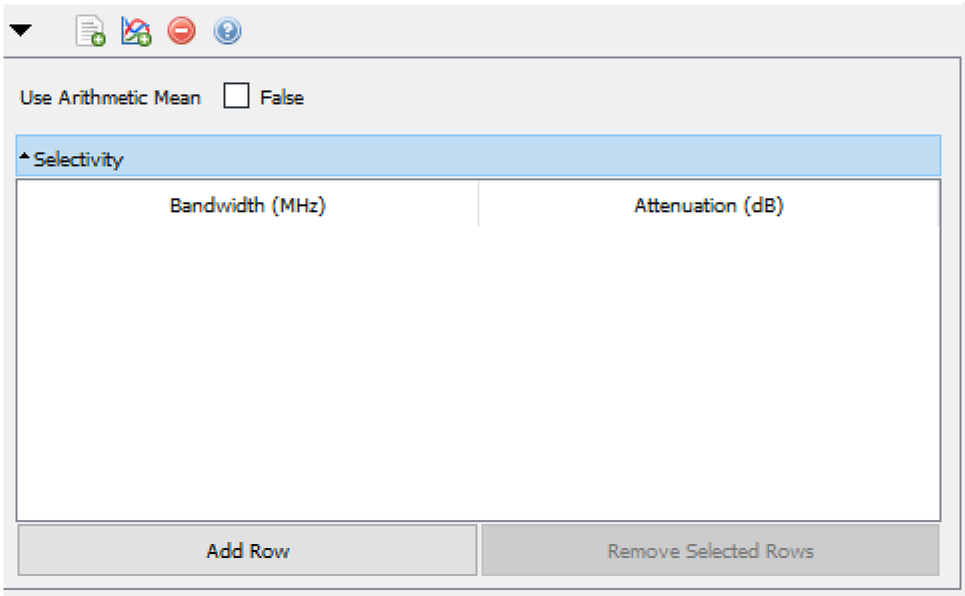
Max Intermod Order: Specifies the highest order of intermodulation product that EMIT will calculate for the amplifier.

Selectivity Profile

The Selectivity Profile defines the spectral profile of the receiver around the tuned channel. The selectivity is specified as a bandwidth and an attenuation as shown below. By default, the bandwidths are centered about the tuned channel using the geometric mean while the attenuations are relative to the "peak" susceptibility value. If no bandwidth-attenuation pairs are specified, the Rx Spectral Profile will be rectangular with the bandwidth defined by the Band's channel bandwidth. There is no limit on the number of bandwidth-attenuation pairs that can be specified, enabling the Rx Spectral Profile to be modeled with a high degree of fidelity. The example below shows a Rx Band tuned to 100 MHz. The three bandwidth-attenuation values specified in the top figure produce the Rx Susceptibility envelope shown in the lower plot.

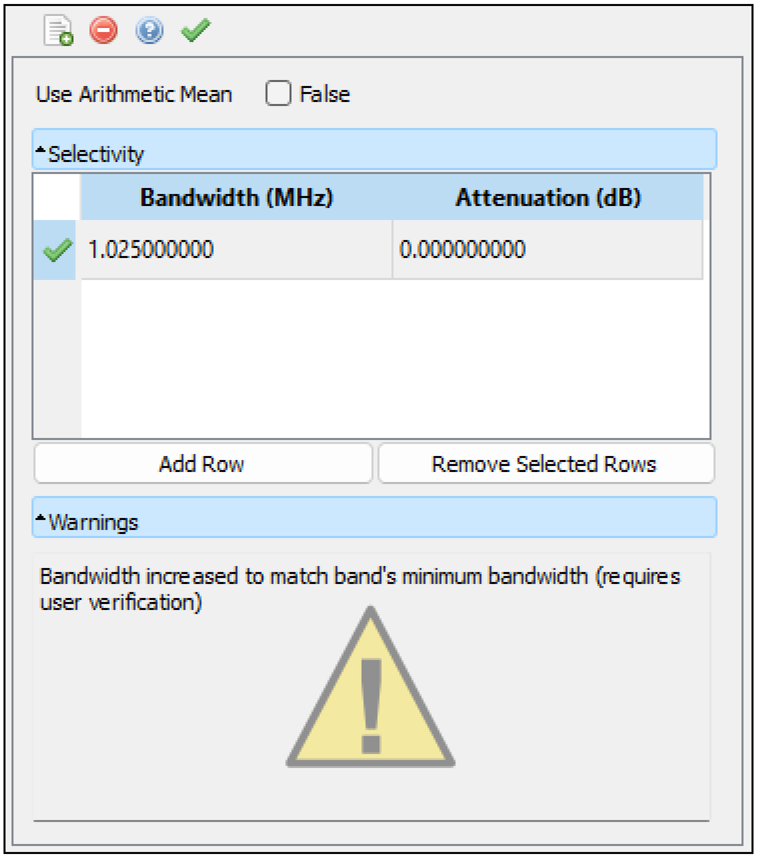

Selectivity
Bandwidth (MHz)
Attenuation (dB)
0.02
3
1
20
10
60
Add Row
Remove Selected Rows


Magnitude (dB)
0
-20
-40
-60
-80
-100
-120
60 dB bandwidth
20 dB bandwidth
3 dB bandwidth
99.4
99.6
99.8
100
100.2
100.4
100.6
Frequency (MHz)
Band 100 MHz (Rx)



- **Use Arithmetic Mean:** If true, the arithmetic mean is used to center the bandwidths about the tuned channel. If false, the geometric mean is used to center the bandwidths about the tuned channel.
- **Add Row:** Add a row to the table.
- **Remove Selected Rows:** Delete the selected rows from the table.

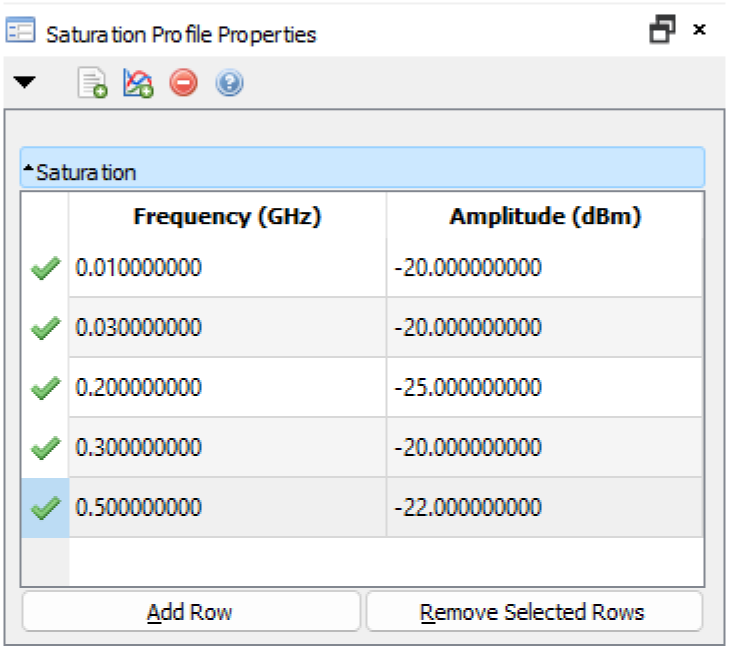
Verify warnings are used to highlight changes automatically made by EMIT due to a user change in a different node. They are purely informational and can be cleared by clicking the green check mark at the top of the property panel.

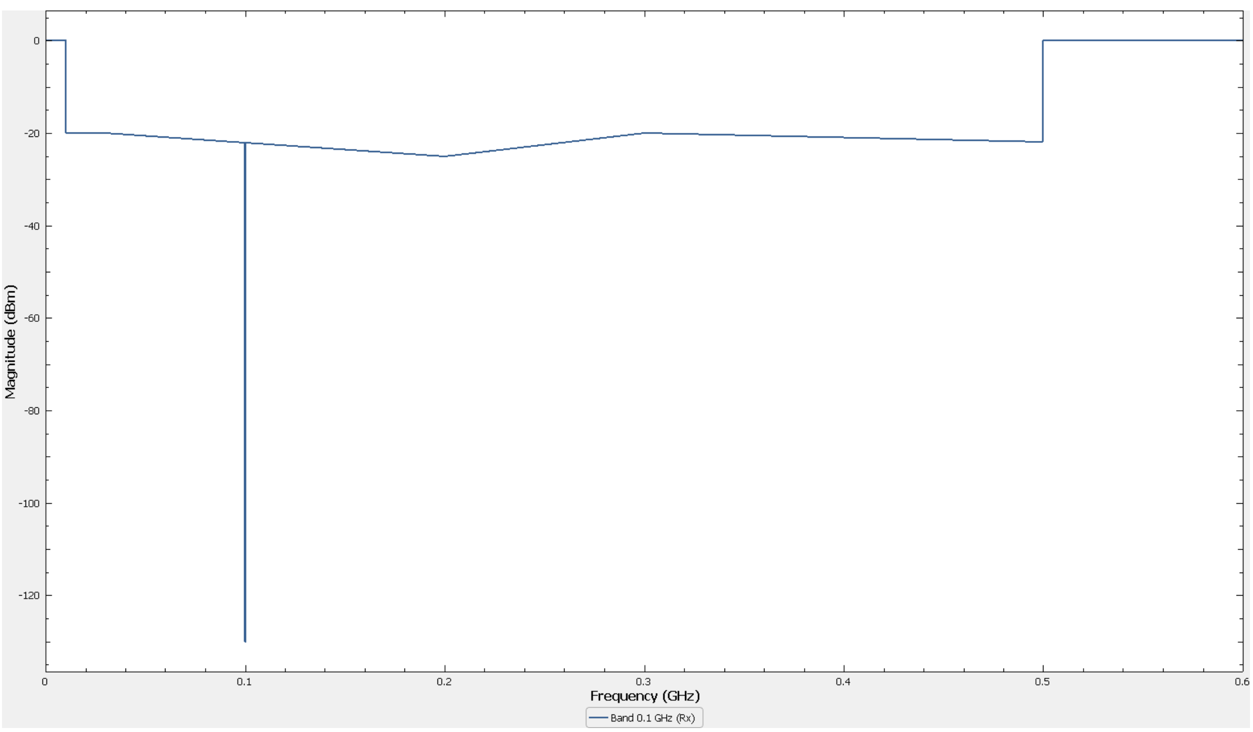


Saturation Profile

The Saturation Profile defines the out-of-band characteristics of the receiver's spectral profile. The saturation is specified as frequency/amplitude pairs. If the saturation profile is not defined, the receiver's saturation level will be defined by the Saturation Level specified in the Rx Spectral Profile node. There is no limit in the number of frequency/amplitude pairs that can be specified, however, too many pairs can result in longer simulations.

The example below shows a receiver's spectral profile, including the tuned channel, with 5 frequency/amplitude pairs defined in the Saturation Profile and a global Saturation Level = 0 dBm defined in the Rx Spectral Profile.





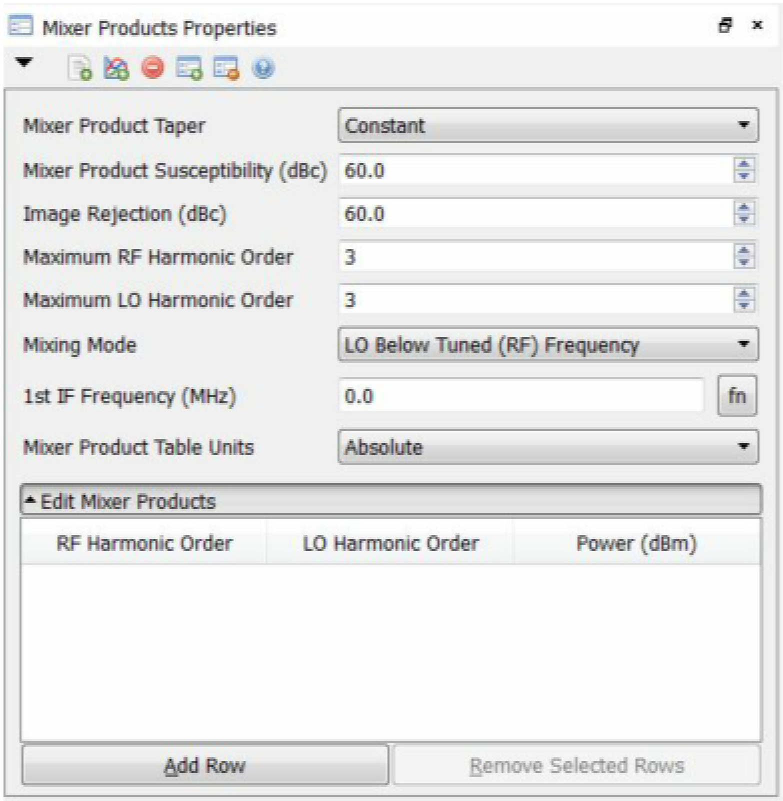
The data can also be imported from a comma-separated value (csv) file. The file should have 2 columns: frequency (in Hz), power (in dBm). It can also contain comment lines prefixed by "#"

which will be ignored by the import.

There are no parameters associated with the Saturation Profile node.

Mixer Products

Mixer Products are a class of Rx spurious responses that result from the nonlinear mixers present in the frequency conversion stages of a receiver. If a receiver's first intermediate frequency (IF) and mixing mode is known, then EMIT can compute the location of these mixer products and include them in the Rx susceptibility profile.

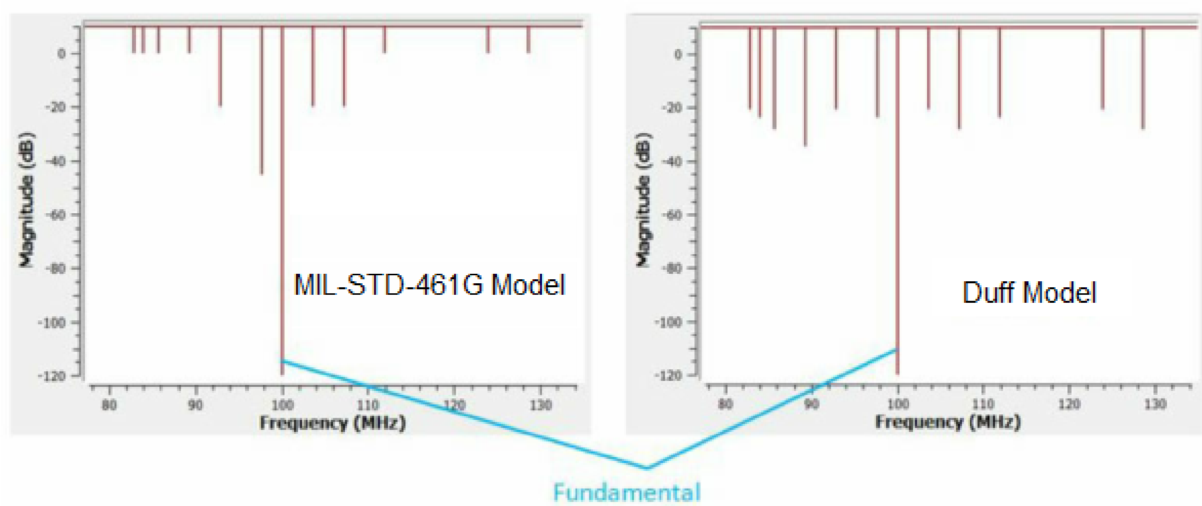


Mixer Product Taper: EMIT offers several tapers that can be optionally applied to the Rx Spectral Profile mixer products. The tapers determine the amplitude of each of the mixer products. If Constant is selected, the amplitude of the mixer products is set at a constant amplitude relative to the fundamental. The MIL-STD-461G model applies the limits defined in MIL-STD-461G to the mixer product amplitude levels. The Duff Model determines the amplitude level of the Nth mixer product based on:

$$P_r(f_{sr}) = P_r(f_r) + I * log(p) + J$$

where

$$P_r(f_{sr}) = Rx\ susceptibility\ at\ spurious\ frequency$$
$$P_r(f_r) = Rx\ susceptibility\ at\ tuned\ frequency$$
$$p = integer\ associated\ with\ the\ local\ oscillator\ harmonic$$
$$I = slope\ of\ spurious\ response\ susceptibility\ in\ dB/decade$$
$$J = intercept\ in\ dB\ relative\ to\ fundamental\ susceptibility$$



> **Note:** Which of the following settings are available in the Mixer Products property panel depends on which Mixer Product Taper is selected as indicated in each item's description.

Mixer Product Susceptibility (dBc): If Constant is selected for the mixer product taper, the mixer product susceptibility determines the amplitude (above the fundamental) of the mixer products.

Mixer Product Slope (dB/decade): Used by the Duff Model only, the mixer product slope is I in the equation above. It determines the rate at which the amplitude of the local oscillator harmonics decrease and is defined in dB/decade.

Mixer Product Intercept (dB/decade): Used by the Duff Model only, the mixer product intercept is J in the equation above. J is determined from a statistical analysis of the receiver and is specified in dB above the fundamental sensitivity.

Minimum Tuning Frequency: Only visible if MIL-STD-461G is selected as the mixer product taper. This is the minimum tuning frequency for the receiver band.

Maximum Tuning Frequency: Only visible if MIL-STD-461G is selected as the mixer product taper. The bandwidth at which the Rx Band's susceptibility envelope is 80 dB above the in-channel susceptibility level. The equation for determining the level of the mixer products when using the MIL-STD-461G taper is:

80 dB Bandwidth: Only visible if MIL-STD-461G is selected as the mixer product taper. This is the minimum tuning frequency for the receiver band.

$$P_r(f_{sr}) = P_r(f_r) + \frac{160}{BW} * (f_{sr} - f_r) \qquad for: \left(f_r - \frac{BW}{2}\right) < f_{sr} < \left(f_r + \frac{BW}{2}\right)$$

where

$$\begin{aligned} P_r(f_{sr}) &= Rx\ susceptibility\ at\ spurious\ frequency \\ P_r(f_r) &= Rx\ susceptibility\ at\ tuned\ frequency \\ f_{sr} &= spurious\ signal\ frequency \\ f_r &= desired\ signal\ frequency \\ BW &= 80\ dB\ bandwidth\ of\ the\ receiver \end{aligned}$$

Image Rejection (dBc): Defines the Rx susceptibility level at the Image Frequency. The image frequency is given by:

$$f_{\mathrm{img}} = \begin{cases} f + 2f_{\mathrm{IF}}, & \text{if } f_{\mathrm{LO}} > f \text{ (high side injection)} \\ f - 2f_{\mathrm{IF}}, & \text{if } f_{\mathrm{LO}} < f \text{ (low side injection)} \end{cases}$$

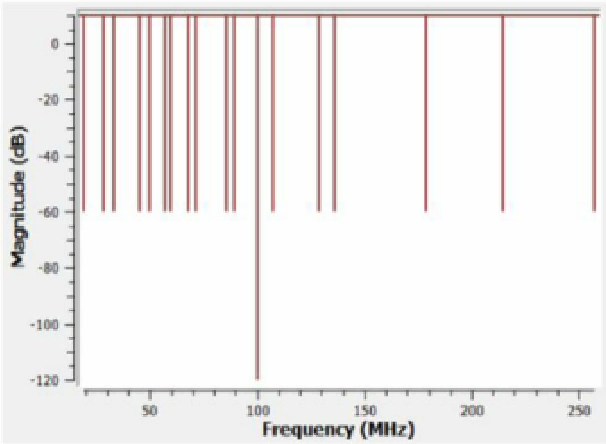
where f is the tuned channel frequency.

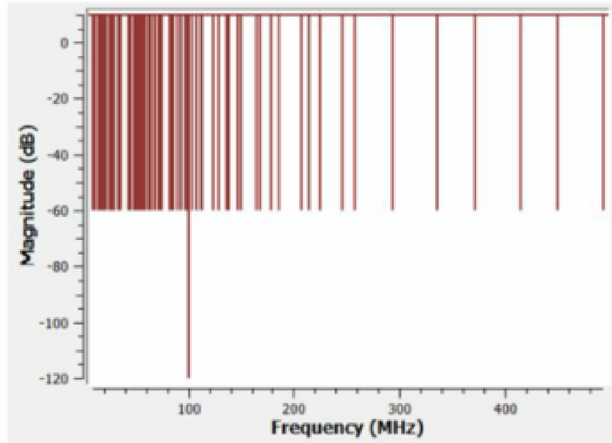
Spurious Rejection (dBc): Used for DD-1494 mode only, the spurious rejection defines the Rx susceptibility level for all mixer products except the Image Frequency.

Maximum RF Harmonic Order: Maximum value to use for m when computing mixer products, where:

$$f_{IF} = |mf_{RF} \pm nf_{LO}|$$

Maximum LO Harmonic Order: Maximum value to use for n when computing mixer products. The effect of increasing the RF and LO maximum harmonic orders is shown below for m=n=3 (left) and m=n=6 (right):





Mixing Mode: Defines the relationship between the RF (tuned channel), IF and LO frequencies. There are three options here:

LO Above Tuned (RF) Frequency: Use when the LO frequency is above the tuned channel frequency:

$$f_{LO} > f$$

LO Below Tuned (RF) Frequency: Use when the LO frequency is below the tuned channel frequency:

$$f_{LO} < f$$

LO Above/Below Tuned (RF) Frequency: This setting is used when the LO frequency is above the tuned channel frequency over part of the Band and below it for the other part. In this case two additional parameters will appear as shown below to define where the transition from high LO to low LO occurs. The Use High LO setting tells EMIT whether the LO frequency is higher than the tuned channel frequency (LO Above Tuned (RF) Frequency) or below the RF Transition Frequency.

1st IF Frequency (MHz): The intermediate frequency for the radio's first conversion stage. For receivers that have a non-constant IF that depends on the tuned channel (RF) frequency, an expression can be used to define the IF using the Expression Editor.

Mixer Product Table Units: Specifies the units used for the power level defined in the table. The power can be entered in either absolute power or relative to the tuned channel's "peak" susceptibility level.

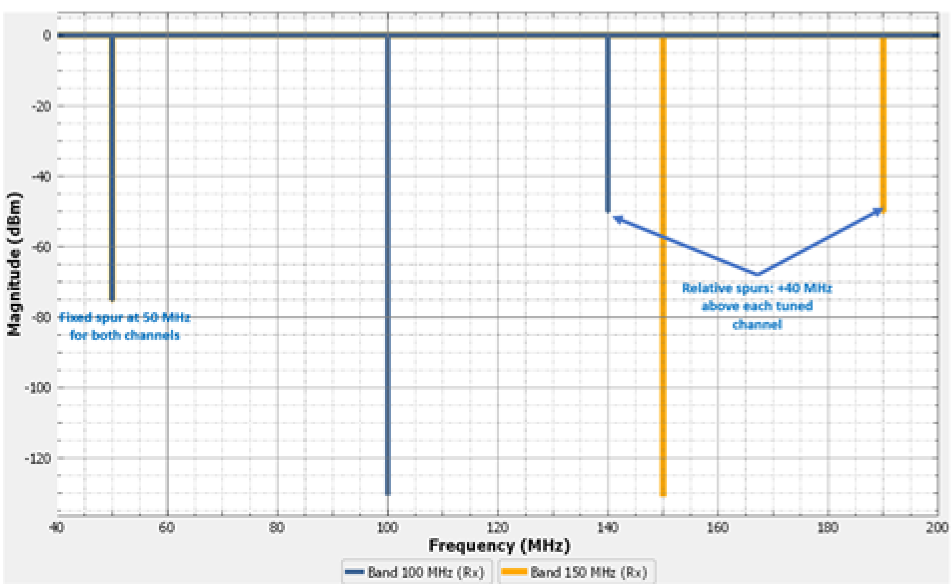
Add Row: Add a row to the table. The tuned channel and image frequency (RF Order = +/-1, LO Order = 1) are not configured in the table. The image frequency spur is configured by the Image Rejection parameter, and the tuned channel susceptibility is configured by the Rx Spectral Profile parameters or the Selectivity table.

Remove Selected Rows: Delete the selected rows from the table.

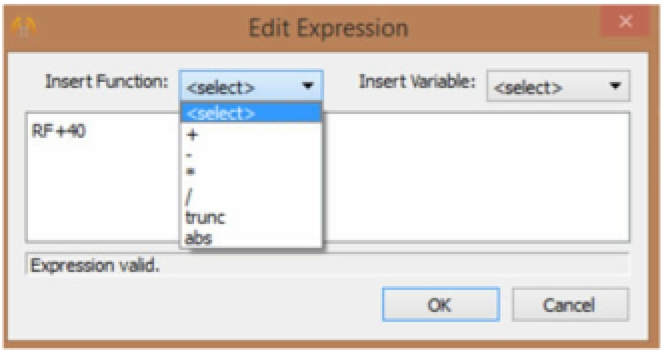
Spurious Responses

Non-harmonic spurious responses (spurs) can be added to a Rx Spectral Profile by adding a Spurious Responses node. Any number of spurs can be added in the table appearing in the Spurious Responses property panel. There are two types of spurs available: Fixed and Relative. Fixed spurs are at the same frequency for all channels in the band. Relative spurs are specified as a function of the channel frequency. Spurious Responses are defined by their frequency, bandwidth, and power (susceptibility).

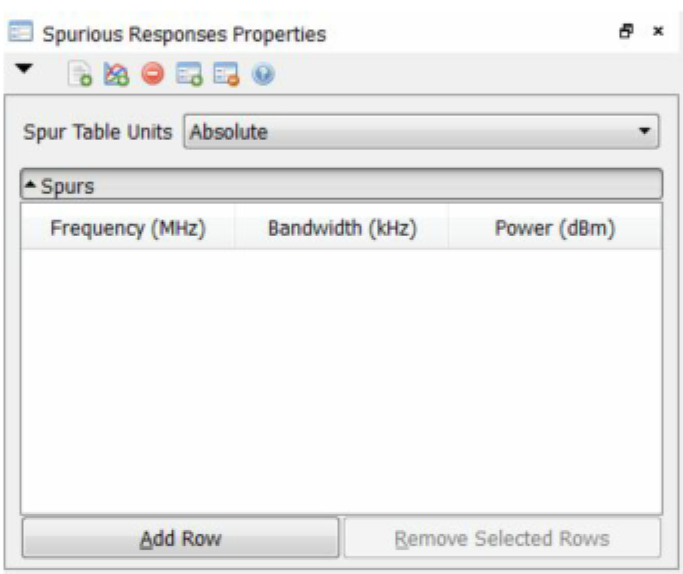
When entering data for the frequency of a receiver spur in the Spurious Responses table, the frequency can be a fixed number or it can be an expression defined with respect to the channel frequency. To enter a fixed frequency, simply enter a number in the Frequency field. For a spur relative to the channel (RF) frequency, an expression can be entered into the Frequency field. In the example shown below, there are two spurs defined. The first is at a fixed frequency of 50 MHz with a bandwidth of 20 kHz and amplitude of -75 dBm. The second is a relative spur that is located 40 MHz above the channel frequency. This 40 MHz offset is maintained for all channels in the Band, while the fixed spur remains at 50 MHz for all channels.



When you click on the Frequency field in the Spurious Response table, you can enter a number (for a fixed spur) or a mathematical expression involving numbers and the variable "RF" (representing the tuned channel frequency). By pressing the  button, you can access the **Expression Editor** for creating expressions.



The tables can be populated manually by adding rows and typing in the desired information or the data can be imported via an external CSV file. The power can be entered in either absolute or relative units (relative to the tuned channel's peak power) as defined by the Spur Table Units parameter.



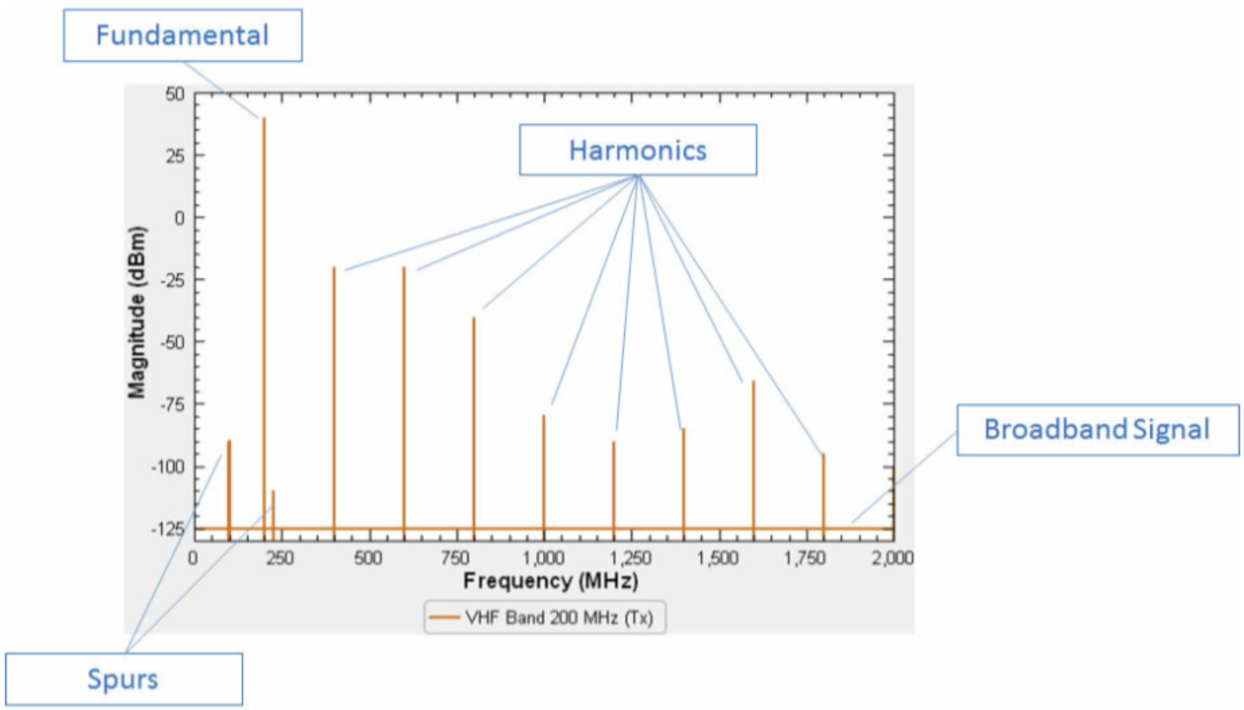
Spur Table Units: Specifies the units used for the power level defined in the table. The power can be entered in either absolute power or as a power level relative to the tuned channel's "peak" susceptibility.

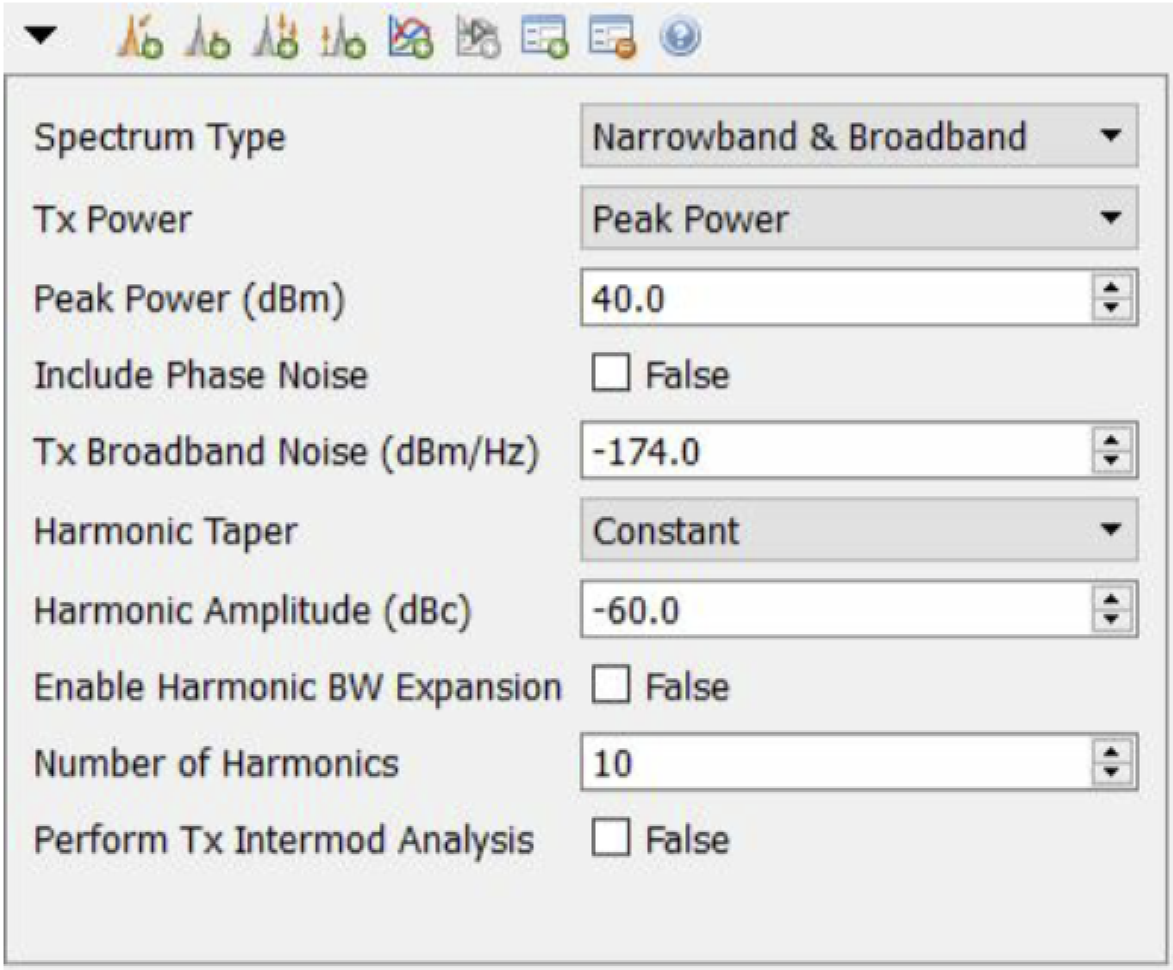
Add Row: Add a row to the table.

Remove Selected Rows: Delete the selected rows from the table.

Tx Spectral Profile

The Tx Spectral Profile completely defines the broadband emission spectrum for each channel within the Band. Each channel in a Band shares the same operating parameters. However, different Bands for a single radio may have different operating parameters. The spectra that EMIT computes are based on the parameters supplied for the Bands and are referred to as parametric channels.





Spectrum Type: As shown in the plot above, a Tx Spectral Profile generally consists of both narrowband signal components (fundamental, spurs & harmonics) and a broadband signal component. In some cases, however, it may be desired tomodel a Tx Spectral Profile with only a broadband signal component. For example,an emitter that is characterized by a wideband noise density profile. This setting selects whether to include narrowband signal components in the Tx Spectral Profile.

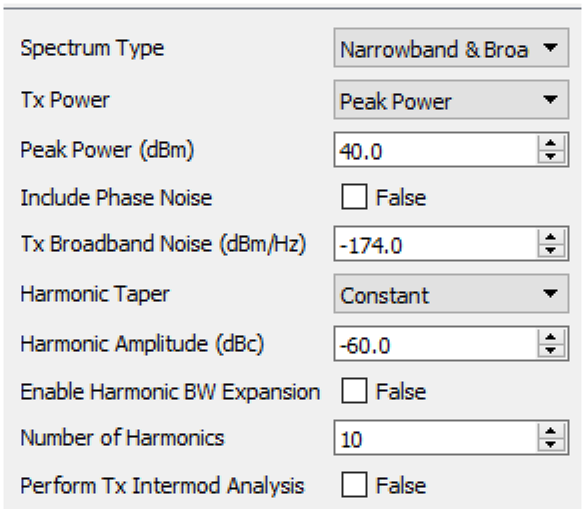
Narrowband & Broadband

Tx Spectral Profile includes both narrowband and broadband components. The narrowband signal characteristics are defined by the Tx Spectral Profile properties and optionally by adding a Narrowband Emissions Mask, Custom Tx Harmonics and/or Spurious Emissions to the Tx Spectral Profile. Each of these appear in the configuration tree as children of the Tx Spectral Profile node. The frequency dependence of the broadband signal can be defined by adding a Broadband Noise Profile to the Tx Spectral Profile.

Broadband Only

Tx Spectral Profile includes a broadband only signal component. The frequency dependence of the Broadband signal can be defined by adding a Tx Broadband Noise Profile to the Tx Spectral Profile.

Tx Broadband Noise: The amplitude of the radio's broadband power density spectrum. Tx broadband noise levels are specified in terms of dBm/Hertz. The average power level of the broadband noise should be input here. EMIT will then compute the peak broadband noise level within a receiver's tuned channel when calculating the EMI Margin. The minimum broadband noise in EMIT is limited to the thermal noise floor of a 50-ohm system at room temperature which is -174dBm/Hz. Values smaller than that cannot be entered here.



Tx Power: This setting determines whether the Tx power will be specified as Peak Power or Average Power for the current Tx Spectral Profile for modulation types that are specified in terms of power. For some modulation types, the amplitude is specified instead as a peak voltage as shown below, in which case this setting will not be present in the Tx Spectral Profile property panel. Note: The Tx power specified here is the power at the output of the Tx antenna port and includes any gain specified when Perform Tx Intermod Analysis is enabled.

Peak Power: The peak power of the fundamental (carrier) in dBm.

Average Power: The average power of the fundamental (carrier) in dBm.

Include Phase Noise: Enable/disable phase noise in the broadband noise model of the transmitter spectrum. The phase noise decreases by 20 dB/decade.

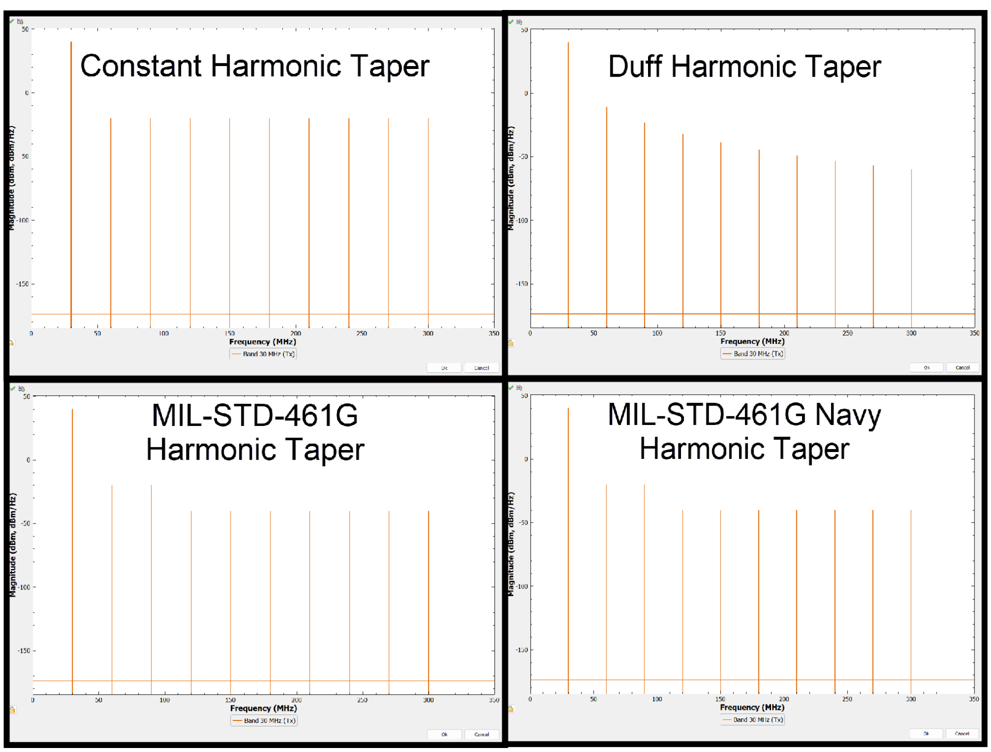
Tx Broadband Noise: The amplitude of the radio's broadband power density spectrum. Tx broadband noise levels are specified in terms of dBm/Hertz. The average power level of the broadband noise should be input here. EMIT then computes the peak broadband noise level within a receiver's tuned channel when calculating the EMI Margin. The minimum broadband noise in EMIT is limited to the thermal noise floor of a 50-ohm system at room temperature which is -174dBm/Hz. Values smaller than that cannot be entered here.

Harmonic Taper: EMIT offers several tapers that can be optionally applied to the Tx Spectral Profile's harmonics. The tapers determine the amplitude of each of the harmonics. The MIL-STD-461G model applies the limits defined in MIL-STD-461G to the harmonic amplitude levels. The MIL-STD-461G Navy model applies the limits defined in the standard for Navy shipboard applications. The Duff Model determines the amplitude level of the Nth harmonic based on:

$$P_{n-th} = P_{fo} + AlogN + B$$

where:

- P_{n-th} = average power in dBm at the N^{th} harmonic
- P_{fo} = fundamental power in dBm
- N = harmonic number
- A,B = constants derived from a transmitter statistical analysis where:
  - A corresponds to the slope in dB per decade, and
  - B corresponds to the amplitude intercept at the fundamental in dB above fundamental



Harmonic Amplitude: The amplitude of harmonics located at integer multiples of the channel frequency. All harmonics are initially set at this level and can be changed individually by adding Custom Tx Harmonics from the Tx Spectral Profile menu. Harmonic levels are specified in terms of dB relative to the carrier (dBc). Note that Harmonic Amplitude is only visible when the Harmonic Taper is set to Constant.

Enable Harmonic BW Expansion: By default, EMIT sets the bandwidth of each harmonic to be the same as the fundamental bandwidth. When Enable Harmonic Bandwidth Expansion is enabled, EMIT will instead set the harmonic bandwidths equal to the harmonic number times the fundamental bandwidth. That is, the second harmonic bandwidth will be twice the fundamental bandwidth, the third harmonic bandwidth will be three times the fundamental bandwidth, etc.

Number of Harmonics: The number of harmonics (including the fundamental) included in the Tx Spectral Profile computed by EMIT. Since this includes the fundamental, the number of harmonics in the spectral profile is one less than this setting. Note that by this convention, the Nth harmonic has harmonic order (N+1).

2nd Harmonic Level (DD-1494 Mode Only): The amplitude of the 2nd harmonic of the channel frequency specified in terms of dB relative to the carrier (dBc).

3rd Harmonic Level (DD-1494 Mode Only): The amplitude of the 3rd harmonic of the channel frequency specified in terms of dB relative to the carrier (dBc).

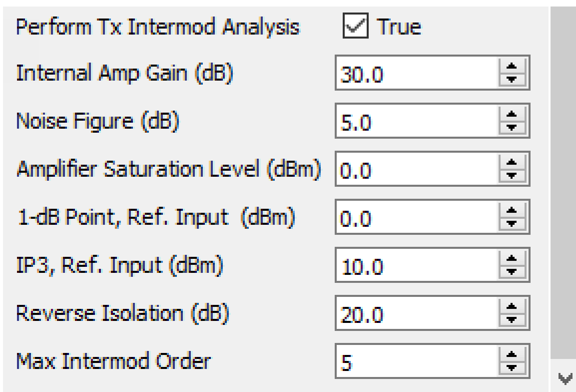
Other Harmonic Level (DD-1494 Mode Only): The amplitude of the higher order harmonics of the channel frequency specified in terms of dB relative to the carrier (dBc).

Perform Tx Intermod Analysis: When enabled (True), this feature includes a nonlinear amplifier internal to the radio. The amplifier is included only for the current Band and different amplifiers can be specified for different Bands of a single Radio. This feature permits a radio's nonlinear amplifier characteristics to be included in the radio mode.

> **Note:** The output power specified as the Tx Power includes any gain specified here and the gain value here does not impact the output power of the Tx as specified by the Tx Power. These parameters are used to perform a nonlinear analysis on spectra from other transmitters that are incident on this port and then reflected away from this transmitter.

The Tx Internal Amplifier is designed to model the nonlinear effects of the transmitter's front end on external signals that are incident on the Tx port. The characteristics of this Band's transmitted spectrum are entirely defined by the above parameters (e.g. Peak Power, Harmonic Amplitude, Broadband Noise, etc.) and are not impacted by the specifications of the internal amplifier. However, the narrowband components comprising the transmitted spectrum will mix with the external signals that are incident on the Tx port. These external signals will also have the internal amplifier's specifications applied including the reverse isolation and gain of the amplifier. Note that unlike the outboard amplifier, the Tx internal amplifier is modeled with an infinite bandwidth and thus there is no "filtering" applied to the mixed and reflected spectra.

It is important to understand that when modeling internal Tx amplifiers when using this option, the amplifier is treated as being active only for channel combinations that include the Tx using the internal amplifier. What this means is that if a single RF System contains multiple Tx ports with internal amplifiers, then only one amplifier within that RF System will be active at any one time with the active amplifier corresponding to the Tx channel included in the channel combination being simulated. Another way to state this is if you pick a channel combination in an N-to-1 simulation, there will only be N internal amplifiers treated as active. Those are the N amplifiers with the ports associated with the N Tx channels.



Internal Amp Gain: Define the in-band gain of the amplifier in dB.

Noise Figure: The amplifier's noise figure in dB. The noise figure is used to calculate the broadband noise added by the amplifier

Amplifier Saturation Level: Specifies an input power level for which EMIT will consider the amplifier saturated. When an amplifier is saturated, EMIT flags it as such in the results and does not trace the input signal any further. The saturation level cannot be set lower than the 1-dB Point.

1-dB Point Ref. Input: The amplifier's 1-dB compression point referred to the input of the amplifier.

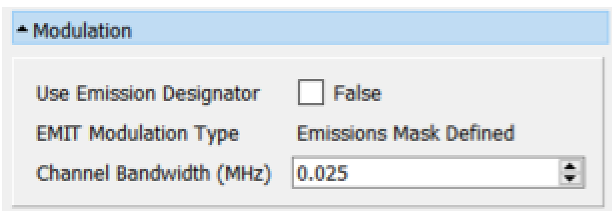
IP3, Ref. Input: The amplifier's third order intercept point referred to the input of the amplifier.

Reverse Isolation: This is the amplifier's reverse isolation. The reverse isolation of an amplifier is a measure of how well a signal applied to the amplifier's output is "isolated" from its input. It is equivalent to the S12 S-parameter.

Max Intermod Order: Specifies the highest order of intermodulation product that EMIT will calculate for the amplifier.

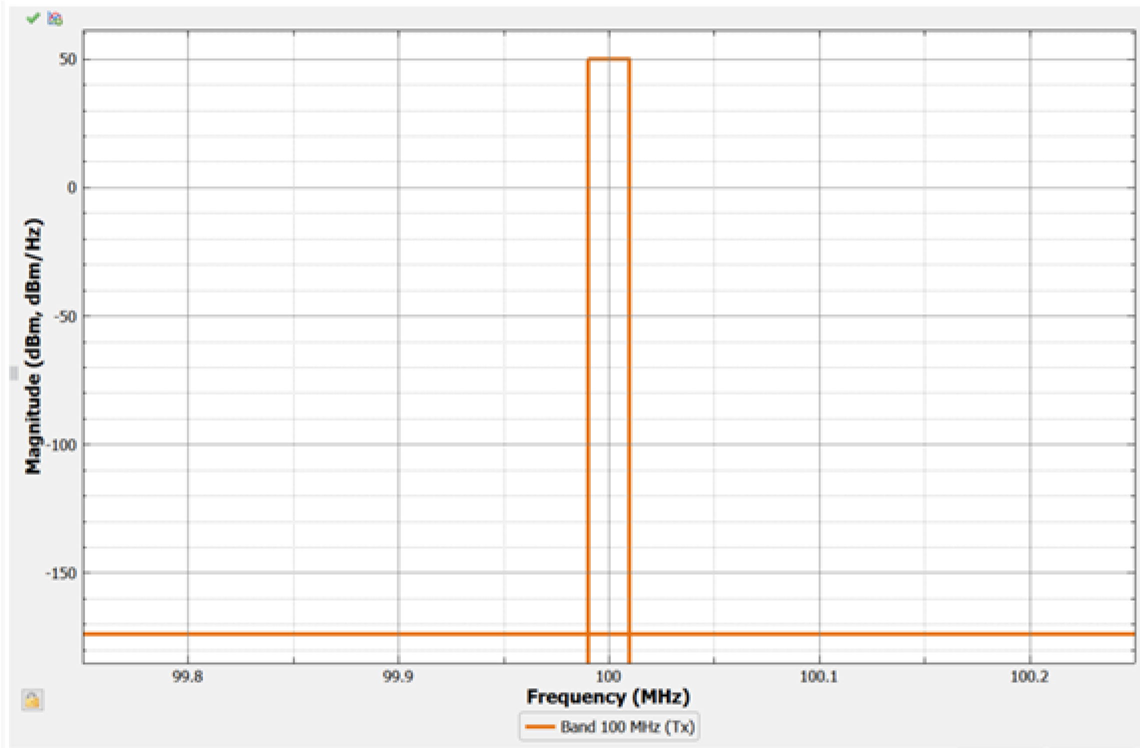
Narrowband Emissions Mask

The Narrowband Emissions Mask allows a user-defined shape (mask) to be used to define the spectral shape of the fundamental and harmonic emissions. This is accomplished by entering bandwidth and attenuation (or frequency and amplitude) data in the Narrowband Emissions Mask property panel or by importing a CSV file containing the mask definition. This feature provides an alternative to specifying the modulation type in a Band for determining the shape of the narrowband emissions spectrum. When a Narrowband Emissions Mask is added to a Tx Spectral Profile, the modulation setting in the Band is no longer available and is replaced with the text "Emissions Mask Defined" as shown below.

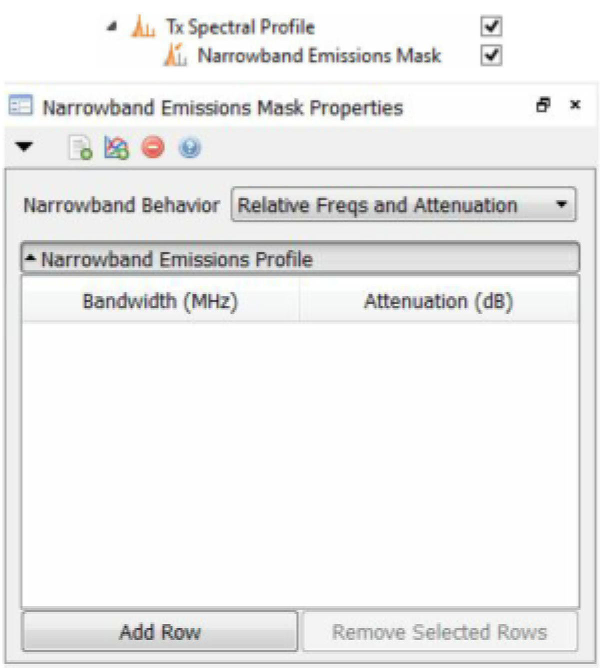


The Narrowband Emissions Mask will be used to define the shape of the Tx Spectral Profile fundamental and harmonics instead of the modulation type. By default, the Narrowband Emissions Mask is defined by a number of bandwidth/attenuation pairs specifying the shape of the mask outside of the Channel Bandwidth as defined in the Band. The Bandwidth values must be greater than the defined Channel Bandwidth and the Attenuation defines the attenuation relative to the Tx fundamental power. The Narrowband Emissions Mask can also be specified using frequency-amplitude pairs. In this case, absolute frequencies are specified for the mask and an absolute power level is specified for each frequency defined.

As an example, consider the Tx Spectral Profile shown below that is defined by a Generic modulation with a Channel Bandwidth = 20 kHz and Peak Tx Power = 50 dBm. The plot below shows the fundamental of this Tx Spectral Profile for a 100 MHz channel frequency.

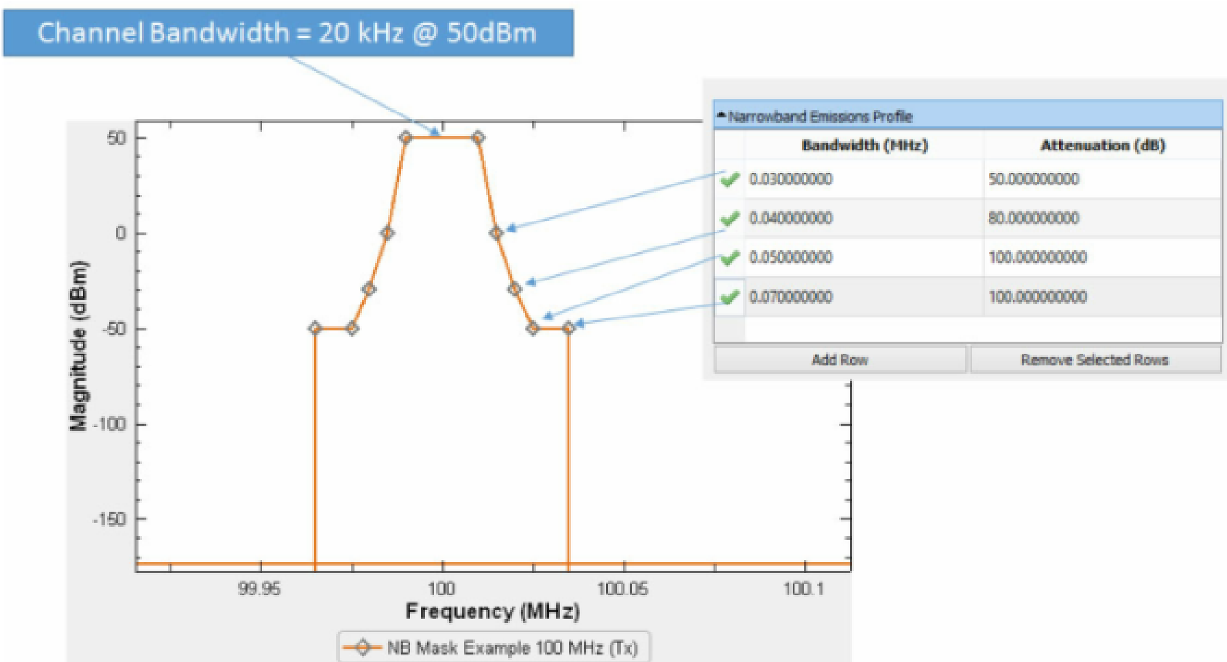


Now, you add a Narrowband Emissions Mask to the Tx Spectral Profile to modify this. The tree nodes and the new Narrowband Emissions Mask property panel are shown below.



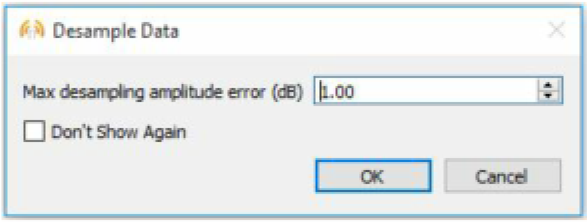
You can populate the table manually by adding rows and typing in the desired bandwidth and attenuation information, or you can import this data from an external CSV file. The Narrowband Emissions Mask defines the attenuation of the emissions outside of the channel bandwidth. Supposed that you want to create a Narrowband Emissions Mask that falls off outside of the 20 kHz channel bandwidth by 50 dB at 30 kHz bandwidth, 80 dB at 40 kHz bandwidth, 100 dB at 50 kHz bandwidth and then remains at that level to a bandwidth of 70 kHz. Note that the bandwidths are defined centered on the channel frequency. This situation is illustrated below where the Narrowband Emissions Mask of the fundamental for a 100 MHz channel frequency is shown.

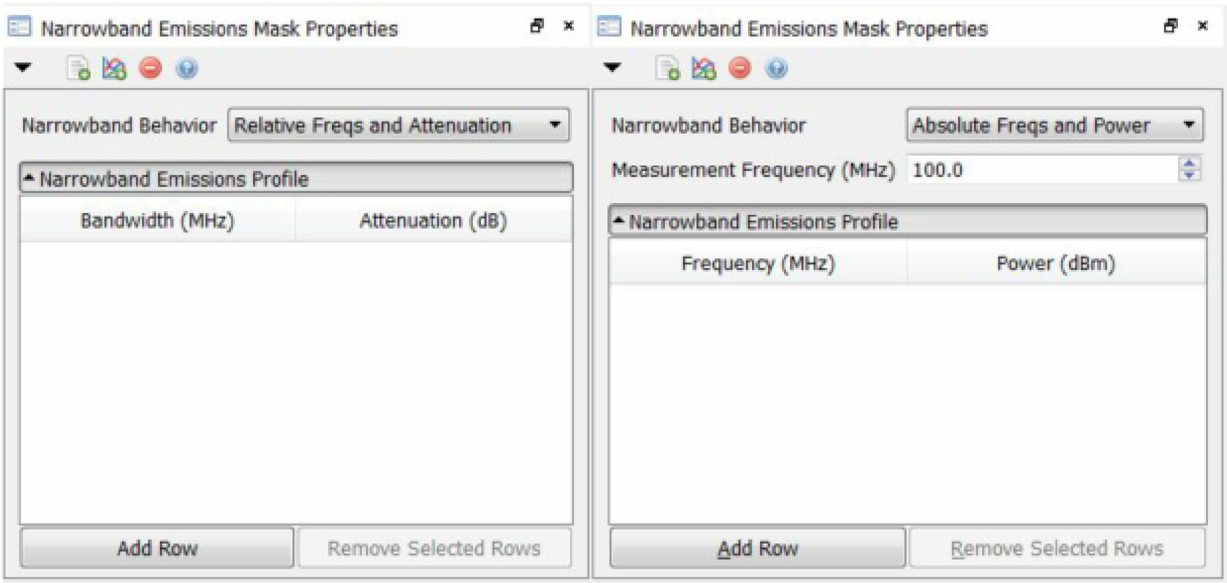
> **Note:** EMIT automatically applies the shape of the fundamental to all harmonics defined for the Band.



Import CSV: Narrowband Emissions Masks can be imported via CSV files. This data is often generated either from measurements or a detailed circuit model of the transmitter, such as one generated by Ansys' Circuit Design. Thus, the data may contain hundreds or thousands of closely spaced frequency samples, which can adversely affect the performance of EMIT's cosite simulation engine. To minimize the impact of dense frequency sampling, EMIT enables users to downsample the spectral data as it is imported.

On importing a narrowband spectral mask, users must specify the **Maximum Desampling Amplitude Error (dB)**. EMIT's desampling algorithm reduces the number of frequency points, if and only if, the error introduced is less than this specified value. Note that EMIT will linearly interpolate between the points.





Narrowband Behavior: If set to Relative Frequencies and Attenuation, then the Narrowband Emissions Mask is defined by bandwidth-attenuation pairs with the bandwidth's centered around the channel frequency and the attenuation defined relative to the fundamental's peak power and specified in dBc. If Absolute Freqs and Power is selected, the the mask is defined by frequency-amplitude pairs. Both the frequency and amplitude values are defined as absolute values with the units specified in the column headers.

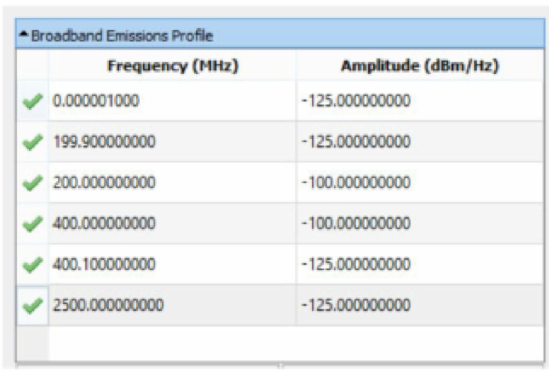
Measurement Frequency: This is only visible when Absolute Freqs and Power is selected and defines the reference frequency about which the mask is specified. If a Band contains multiple channels, then the frequency-amplitude pairs will automatically be shifted when applied to channels other than the Measurement Frequency.

Add Row: Add a row to the Narrowband Emissions Mask. The columns in each row can then be edited. If Relative Frequencies and Attenuation is selected, then the bandwidth specified must be greater than the Channel Bandwidth which is defined in the Band node. The attenuation values are defined relative to the Peak Tx Power defined in the parent Tx Spectral Profile.

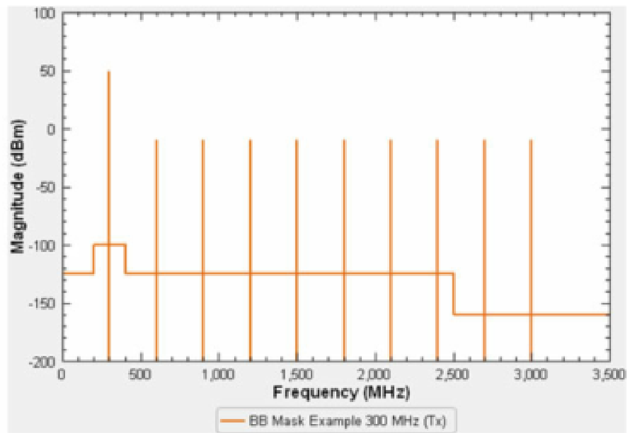
Remove Selected Rows: Delete selected rows from the table.

Tx Broadband Noise Profile

The Tx Broadband Noise Profile allows a user-defined shape (profile) to be used to define the spectral shape of the broadband component of the Tx Spectral Profile. This is accomplished by entering frequency-amplitude pairs in the Tx Broadband Noise Profile property panel or by importing a CSV file containing the profile definition. The Tx Broadband Noise Profile overrides the Tx Broadband Noise and Phase Noise settings in the Tx Spectral Profile node for any defined frequency ranges. An example is shown below.

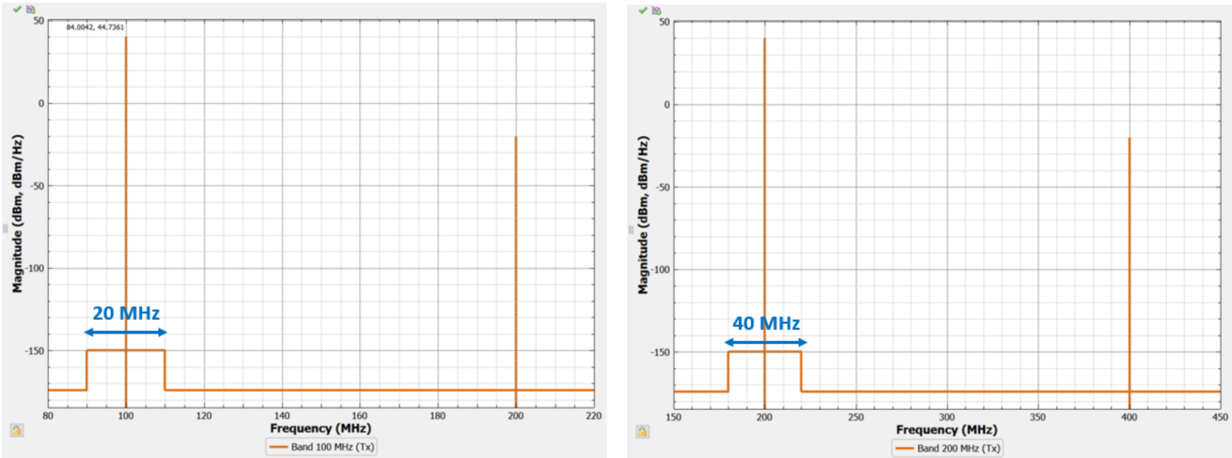

Broadband Emissions Profile

Frequency (MHz)	Amplitude (dBm/Hz)
0.000001000	-125.000000000
199.900000000	-125.000000000
200.000000000	-100.000000000
400.000000000	-100.000000000
400.100000000	-125.000000000
2500.000000000	-125.000000000

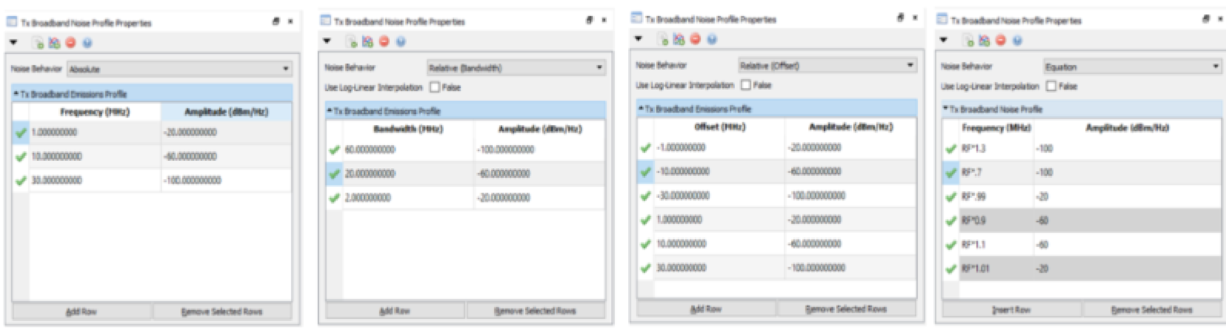


> **Note:** Above the highest frequency defined in the Tx Broadband Noise Profile table, the level returns to -160 dBm/Hz which is the level specified in the Tx Spectral Profile node for Tx Broadband Noise for this example.

Tx Broadband Noise Profiles can be defined as Absolute, Relative (Bandwidth), Relative (Offset) or Equation. An Absolute profile maintains the same noise profile at all channel frequencies of the transmitter as defined in the Band. Relative (Bandwidth) and Relative (Offset) profiles will shift in frequency as the transmit channel frequency changes, maintaining the same relative frequency behavior with respect to the transmit channel. An Equation based profile will set the broadband frequencies based on the specified equations, which can be constants or functions of the channel frequency. Thus, a Band that operates from 100-200 MHz with a broadband noise profile specified as 0.9*RF and 1.1*RF will have a 20 MHz bandwidth when tuned to the 100 MHz channel and a 40 MHz bandwidth when tuned to the 200 MHz channel.



Linear interpolation is used between points for the Absolute definition. The Relative (Bandwidth), Relative (Offset) and Equation based profiles can be configured to linearly interpolate between the values specified in the tables or to use a log-linear interpolation (that is, linear interpolation performed in the log domain).



Noise Behavior: Defines whether the Tx Broadband Noise Profile is Absolute (fixed with respect to the transmit channel frequency), Relative (Bandwidth or Offset) (moves relative to the transmit channel frequency) or Equation (varies depending on the equation). Note that for the former, the table supports frequency-amplitude pairs with the frequencies defined according to the configured units. The Relative (Bandwidth) profile is defined in terms of bandwidth-amplitude pairs with the bandwidth specified relative to the channel frequency. The Relative (Offset) profile is defined in terms of offset/amplitude pairs where the offset is added to the transmit tuned channel frequency. The offset values may be positive or negative. The Equation profile is defined in terms of frequency-amplitude pairs where the frequencies are determined based on equations which can evaluate to a constant frequency or vary based on the channel frequency. Note that the frequency units are always the users preferred units and the amplitude's are always in dBm/Hz.

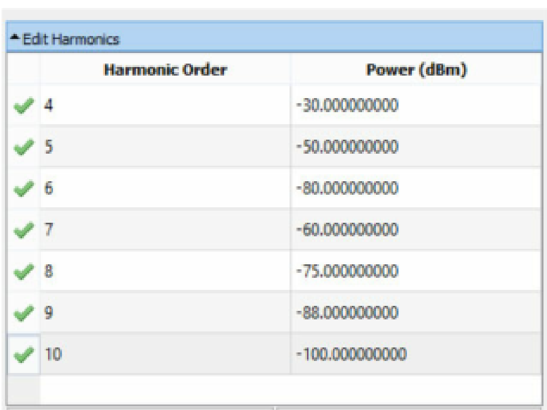
Use Log-Linear Interpolation: If true, EMIT performs a linear interpolation in the log domain to compute the broadband noise amplitudes between the specified frequencies. If false, EMIT uses standard linear interpolation to compute the broadband noise amplitudes.

Add Row: Add a row to the table.

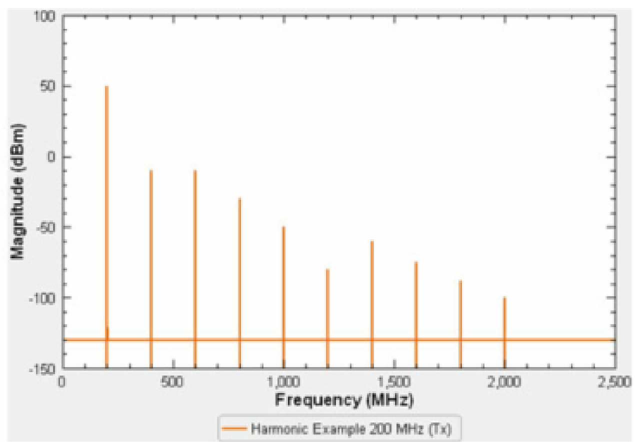
Remove Selected Rows: Delete the selected rows from the table.

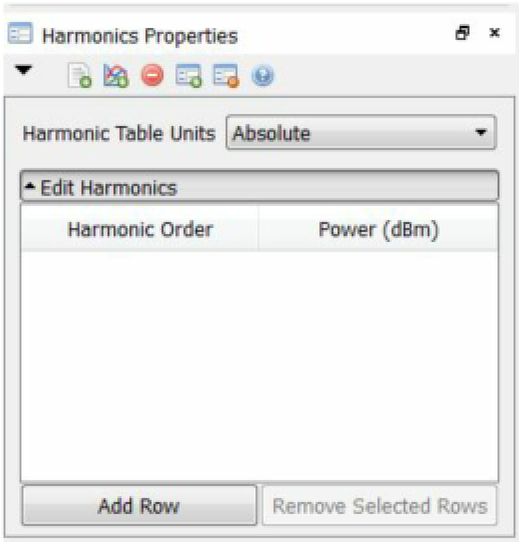
Harmonics

The Harmonics node allows the amplitude of individual harmonics in the Tx Spectral Profile to be set to any desired power level. This is accomplished by entering the harmonic number and associated power level in the Harmonics property panel or by importing a CSV file containing the harmonic definitions. The Harmonic Order defines the integer multiplier of the harmonic. For example, the harmonic located at three times the fundamental frequency has Harmonic Order = 3. The amplitudes of harmonics defined in the Harmonics Table overrides those determined by the settings in the parent Tx Spectral Profile node. An example is shown below for a 200 MHz channel frequency. In this case, the amplitude of the 2nd and 3rd harmonics is set by the Tx Spectral Profile node settings.


Edit Harmonics

Harmonic Order	Power (dBm)
4	-30.000000000
5	-50.000000000
6	-80.000000000
7	-60.000000000
8	-75.000000000
9	-88.000000000
10	-100.000000000





Harmonic Table Units: Specifies the units used for the power level defined in the table. The power can be entered in either absolute or relative units (relative to the peak power of the tuned channel).

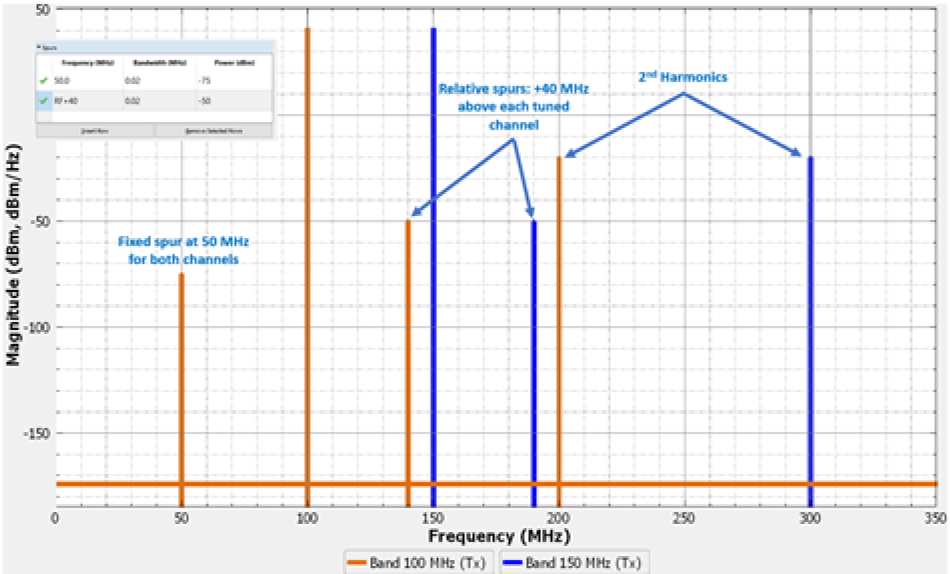
Add Row: Add a row to the table.

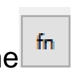
Remove Selected Rows: Delete the selected rows from the table.

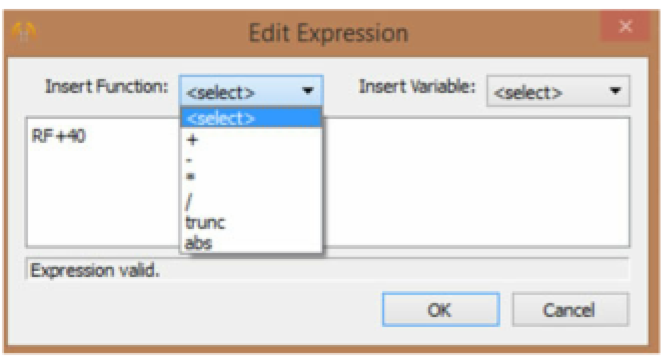
Spurious Emissions

Non-harmonic spurious emissions (spurs) can be added to a Tx Spectral Profile by adding a Spurious Emissions node. Any number of spurs can be added in the table appearing in the Spurious Emissions property panel. There are two types of spurs available: Fixed and Relative. Fixed spurs remain at the same frequency for all channels in the band. Relative spurs are specified as a function of the channel frequency. Spurious Emissions are defined by their frequency, bandwidth, and power. All spurs defined in the table are assumed to have a rectangular shape.

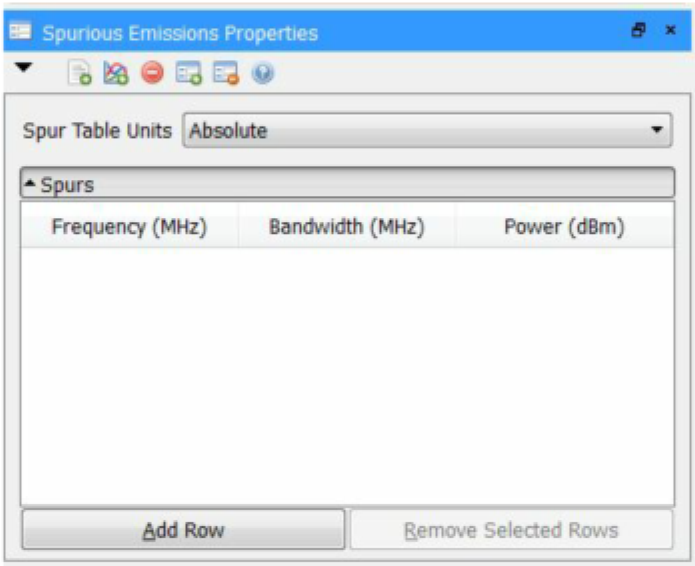
When entering data for the frequency of a Tx spur in the Spurious Emissions property panel, the frequency can be a fixed number or it can be an expression defined with respect to the channel frequency. To enter a fixed frequency, simply enter a number in the Frequency field. For a spur relative to the channel (RF) frequency, an expression can be entered into the Frequency column. In the example below, there are two spurs defined. The first is at a fixed frequency of 50 MHz with a bandwidth of 20 kHz and amplitude = -75 dBm. The second is a relative spur that is located 40 MHz above the channel frequency. This 40 MHz offset is maintained for all channels in the Band while the fixed spur remains at 50 MHz for all channels.



When you click on the Frequency field in the Spurious Emissions table, you can enter a number (for a fixed spur) or a mathematical expression involving numbers and the variable "RF" (representing the tuned channel frequency). By pressing the  button, you can access the **Expression Editor** for creating expressions.



The tables can be populated manually by adding rows and typing in the desired information or the data can be imported via an external CSV file. The power can be entered in either absolute or relative units (relative to the tuned channel's peak power) as defined by the Spur Table Units parameter.

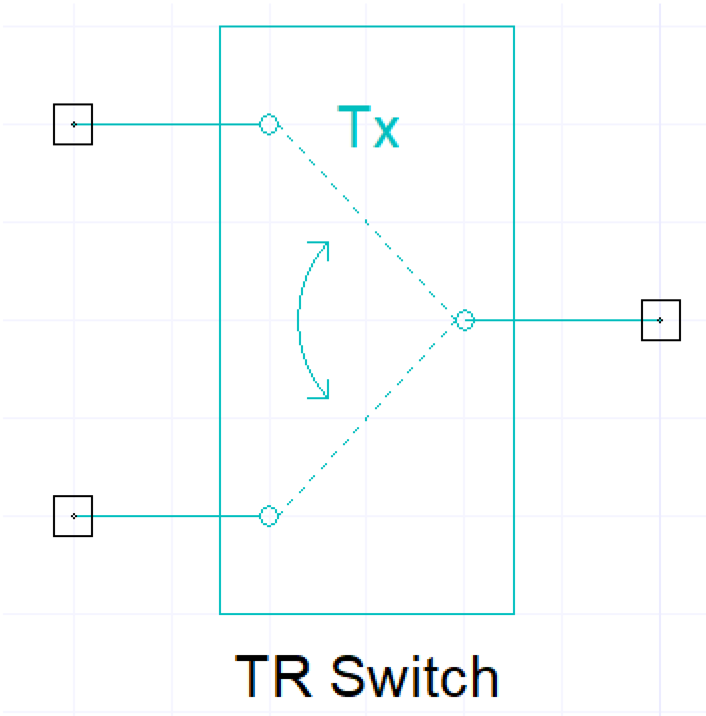


Spur Table Units: Specifies the units used for the power level defined in the table. The power can be entered in either absolute power or as a power level relative to the tuned channel's peak amplitude (dBc).

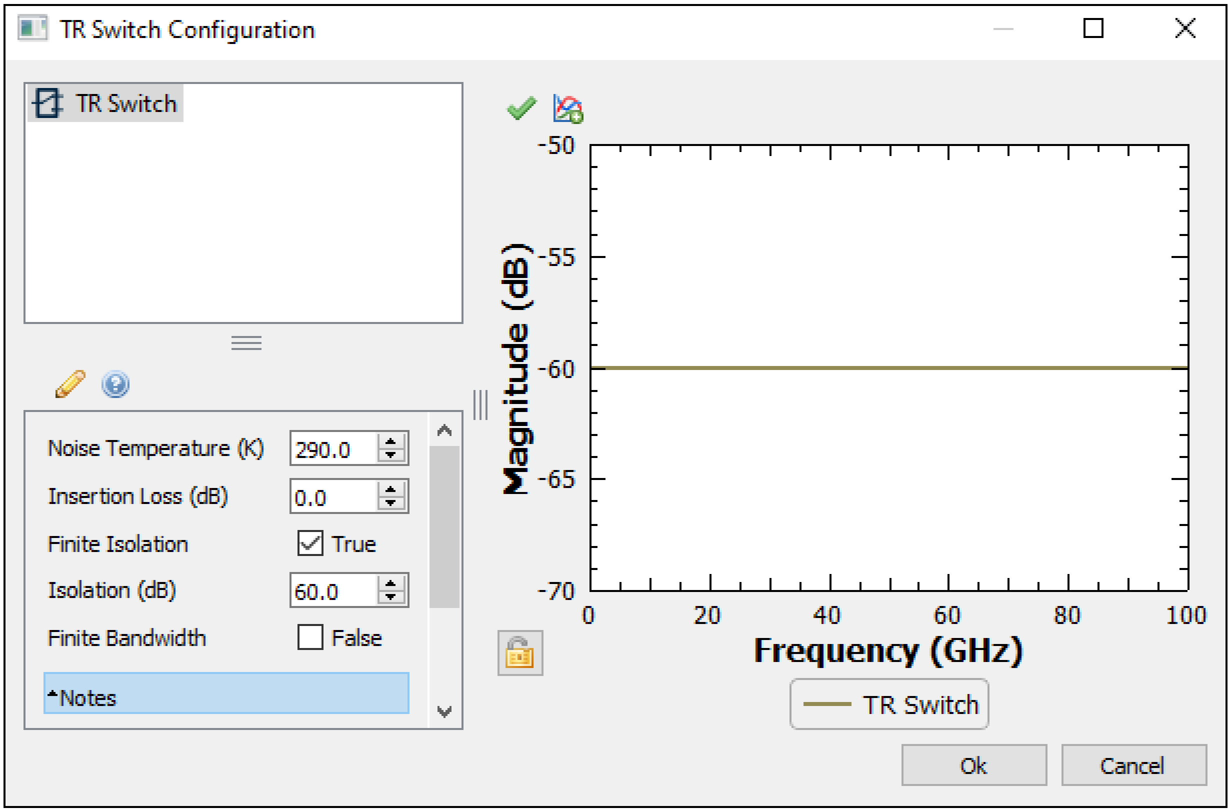
Add Row: Add a row to the table.

Remove Selected Rows: Delete the selected rows from the table.

Switches



Switches are added to an EMIT design from the EMIT Elements component library. There is currently only one type of switch (T/R) available in EMIT.



The TR Switch component in EMIT is a 3-port device that controls the direction of signal flow. The orientation of the switch can be changed via the “flip” options on the component's right-click menu in the schematic editor.

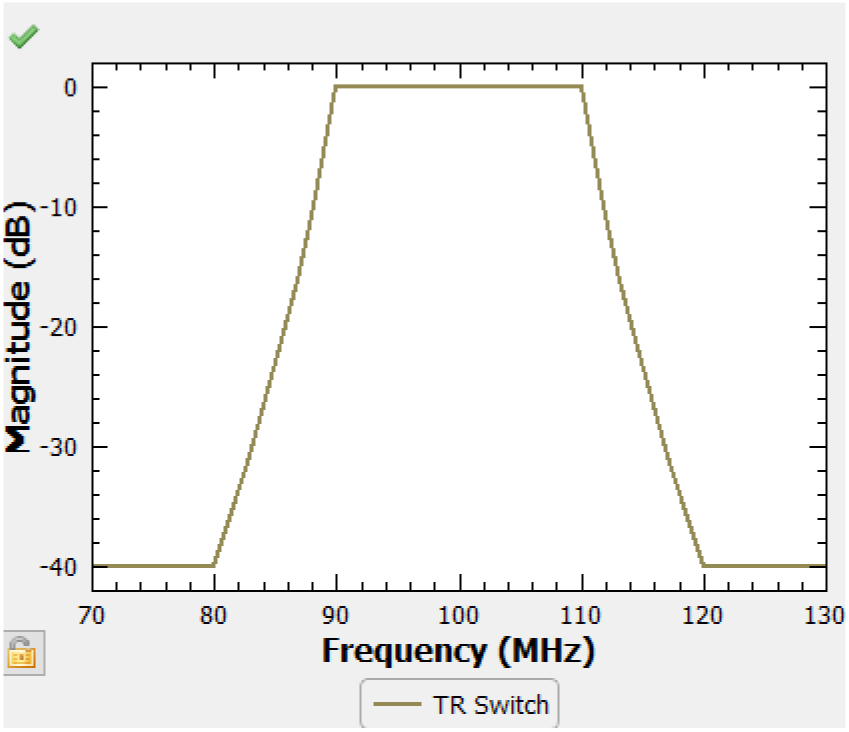
When Finite Isolation is disabled (False), the TR Switch is an ideal switch and power can flow only across the connected port. EMIT automatically determines whether the TR Switch is in the Tx configuration (that is, the Tx Port is connected to the Common port) or the Rx configuration that is, the Rx Port is connected to the Common port) based on the simulated mode of operation of the configuration.

Noise Temperature (K): Noise power added by the switch.

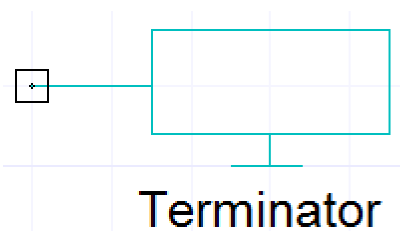
Insertion Loss (dB): Defines the insertion loss between connected ports of the device. That is, the insertion loss from the Tx Port to the Common Port and the Common Port to the Rx Port.

Finite Isolation: Isolation between the disconnected ports of the device is assumed to be infinite unless **Finite Isolation** is enabled (True), in which case the value you provided is used.

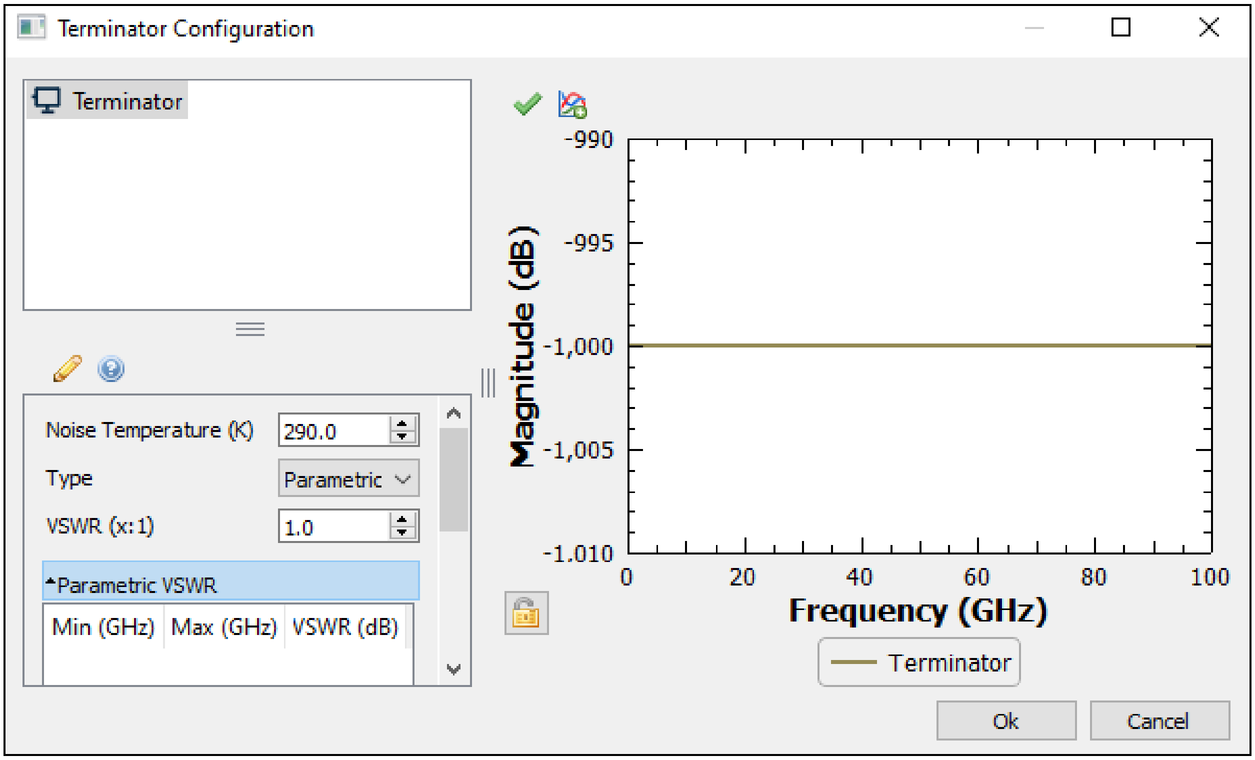
Finite Bandwidth: By default, the bandwidth for an TR Switch is considered to be infinite. When **Finite Bandwidth** is enabled (True) a finite bandwidth for the TR Switch can be defined. The Out-of-Band Attenuation is then applied outside the defined pass band as shown in the plot below. See Band Pass Filter for definitions of the parameters.



Terminators



The Terminator component in EMIT is a 1-port device that attenuates and reflects signals. The return loss of the input signal can be defined by file or parametrically by specifying the Voltage Standing Wave Ratio (VSWR). A VSWR=1.0 implies that all of the input power is absorbed by the terminator and that there is no power reflected.



In the **Type** drop-down, select one of the following:

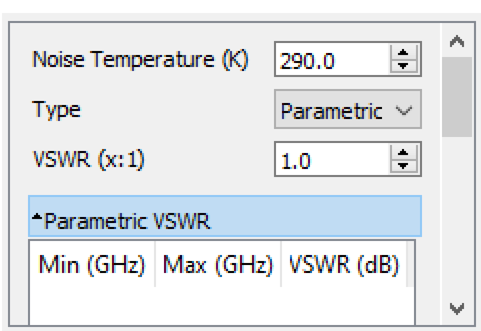
- By File
- Parametric

By File

Noise Temperature (K): Noise power added by the terminator.

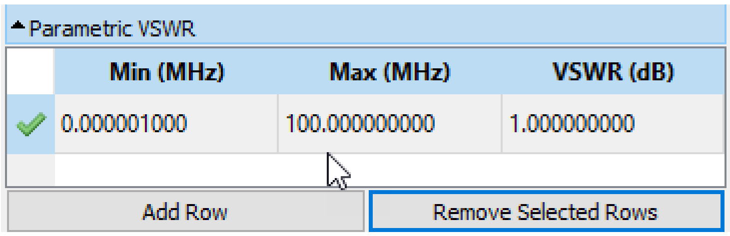
Filename: Defines the 1 port frequency characteristics of the Terminator by providing a 1-port Touchstone file containing the S-parameters for the device.

Parametric



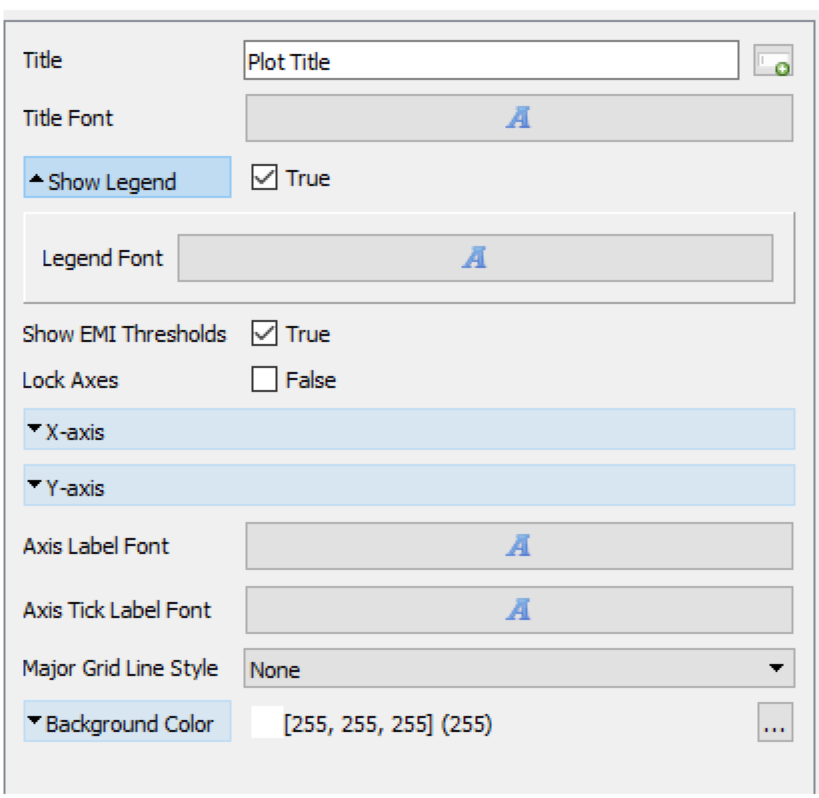
Noise Temperature (K): Noise power added by the terminator.

VSWR: Panel for defining the terminator's characteristics parametrically. You can specify the Voltage Standing Wave Ratio (VSWR) of the terminator over multiple frequency ranges using the Parametric VSWR table. If no values are entered in the table, EMIT uses a constant VSWR for all frequencies.



Plot Properties

The Properties window allows you to control the properties and appearance of the plot as described below.



Title: Set the title text displayed at the top of the plot. While editing the Title, you can click the button to the right of the entry field to insert a variety of symbols (encoded as rich text) at the current cursor position.

> **Note:**
>
> If you clear the title text and the title displayed at the top of the plot area does not completely clear, enter a single space to make it explicitly blank.

> **Note:**
>
> HTML tags may be added to this text. Supported tags include <h1> through <h5> header tags, for bold, <i> for italics, <sup> and <sub> for superscript and subscript, and many others. The Title Font property is often easier to use, but HTML tags allow more flexibility, such as mixed font text, adding newlines via <br>, and changing colors as in . Most HTML tags require an ending tag with the slash '/' character so that the modified text is encapsulated. For example,
>
> Bold <i>bold-italic</i>^{superscript}
>
> generates **Bold *bold-italic* superscript**

Title Font: Brings up a dialog to change the font for the title text displayed at the top of the plot.

Show Legend Toggle Panel Folder

Toggle on/off trace legend at the bottom of the polar plot. When switched on, **Show Legend** becomes a panel folder that can be expanded to show additional settings.

Legend Font: Brings up a dialog to change the font for the legend text

Show EMI Thresholds:

Lock Axes: Toggle on/off locking of the axes extents. EMIT determines default axes based on the data extents of all traces currently being plotted. When toggled on, the Plot window axes will not update in response to indirect user actions that would potentially influence the default choice of axes, such as creating a new plot trace by dragging a node to the plot window. Switch the lock on when you are happy with the current axes extents and don't want them disrupted. Axes will still automatically update in response to more direct actions, such as explicitly changing X-axis Min, clicking the Default Axes button, or performing a dragged-rectangle plot zoom. Also, changing the Scale setting in a Plot-Trace node can force the axes to update.

When toggled off, axes extents are more liable to automatically update in response to changes in plot trace contents.

The **Lock Axes** toggle can also be controlled from the plot window by clicking the button.

X-Axis Panel Folder

Expand the X-Axis panel folder to modify the horizontal axis extents and tick marks of the plot.

X-Axis Min, Max: Set the extents of the horizontal axis

Max Major Ticks (X): Control the number of major tick-mark intervals along the X-axis

The plot has both major and minor tick-marks. The major tick-marks are longer and numerically labeled. Max Major Ticks (X) sets the maximum number of major intervals along the horizontal axis, including partial intervals at the beginning or end of the axis extents. The plot axis management logic may choose a fewer number of intervals, but it will not exceed this limit.

Max Minor Ticks (X): Control the number of minor tick-mark intervals along the X-axis

The plot has both major and minor tick-marks. The minor tick-marks are shorter and not numerically labeled. Max Minor Ticks (X) sets the maximum number of minor tick-mark intervals between major ticks along the horizontal axis. The plot axis management logic may choose a fewer number of intervals, but it will not exceed this limit.

Y-Axis Panel Folder

Expand the Y-Axis panel folder to modify the vertical axis extents and tick marks of the plot.

Y-Axis Min, Max: Set the extents of the vertical axis

Y-Axis Range (dB): Set the span of the Y-axis (visible when the Plot-Trace Scale is set to Decibel)

Max Major Ticks (Y): Control the number of major tick-mark intervals along the Y-axis

The plot has both major and minor tick-marks. The major tick-marks are longer and numerically labeled. Max Major Ticks (Y) sets the maximum number of major intervals along the vertical axis, including partial intervals at the beginning or end of the axis extents. The plot axis management logic may choose a fewer number of intervals, but it will not exceed this limit.

Max Minor Ticks (Y): Control the number of minor tick-mark intervals along the Y-axis

The plot has both major and minor tick-marks. The minor tick-marks are shorter and not numerically labeled. Max Minor Ticks (Y) sets the maximum number of minor tick-mark intervals between major ticks along the vertical axis. The plot axis management logic may choose a fewer number of intervals, but it will not exceed this limit.

Axis Label Font: Brings up a dialog to change the font for the x and y axis labels

Axis Tick Label Font: Brings up a dialog to change the font for the numbers next to the axis ticks

Major Grid Line Style: Set the line style for plot area grid lines extending from the major tick-marks

The default is None, meaning no major tick-mark grid lines.

Major Grid Color: Set the color used for major tick-mark grid lines

This setting becomes available when Major Grid Line Style is set to any value other than None.

Minor Grid Line Style: Set the line style for plot area grid lines extending from the minor tick-marks

Minor tick-mark grid lines are only available when major tick-mark grid lines are switched on (i.e. set to a line style other than None).

Minor Grid Color: Set the color used for minor tick-mark grid lines

This setting becomes available when both major and minor grid lines are switched on.

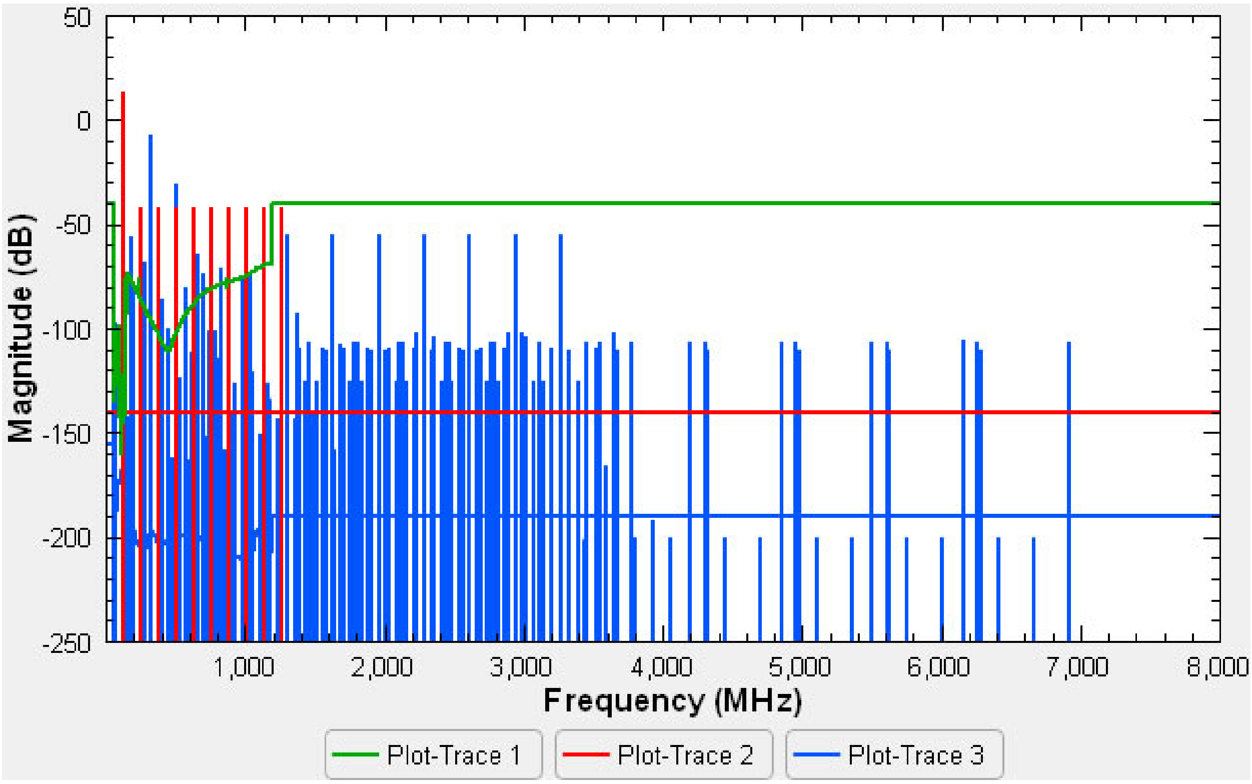
Background Color: Set the background color of the Plot window

Expand this panel folder to set the red-green-blue (RGB) levels for the background color. Click the ellipsis button ... to choose the color from a palette.

Plot Trace Properties

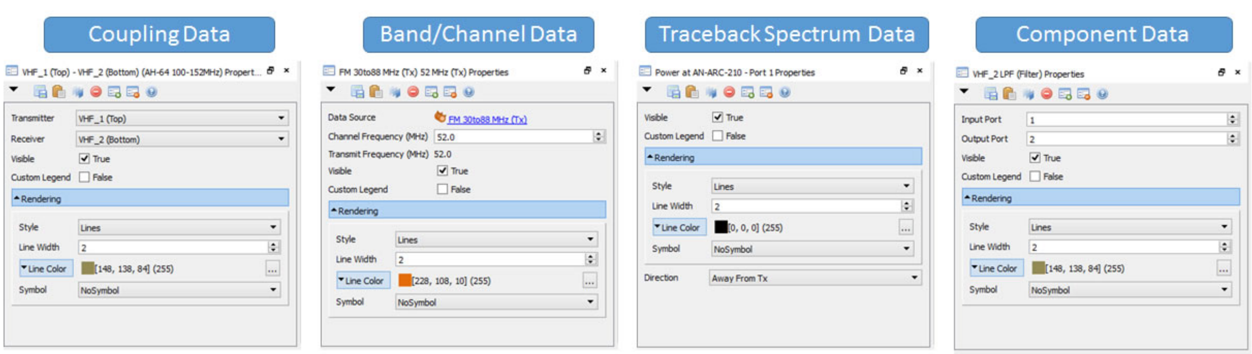
Each Plot node contains one or more Plot Trace nodes. Each Plot Trace node corresponds to a data trace (drawn line) in the plot window. The purpose of the Plot Trace node is to configure the rendering properties of the trace line in the plot, its legend, and to select from the available options of the data that it represents.

Each trace in a plot appears in a legend at the bottom of the plot window as seen in the following example. Selecting one of the legend entries in the plot window will highlight the corresponding trace and bring up the properties for the Plot Trace.



The Plot Trace property panel contains settings to control the rendering properties of the trace drawn in the Plot window and its legend (at the bottom of the plot). The settings available in the Plot Trace properties varies somewhat depending on the particular type of data being plotted. There are four Plot Trace types in EMIT corresponding to different types of data being plotted and the properties for each of these is shown below.

> **Note:**
> Plots of the response of an Amplifier contain some special features. Please refer to the Amplifiers section for more details.



Transmitter / Receiver: (Available when plotting Coupling Data) Plots the coupling data associated with the specified Transmitter and Receiver antennas.

Data Source: (Available when plotting Band data) Indicates the name of the Band node associated with the Plot Trace node. The node name is shown as a link and clicking it takes you to that data node in the project tree. This field is not editable.

Direction: (Available when plotting power spectrum data from the Interaction Diagram) Select whether to plot the power spectrum for the signal moving Away from Tx (to the right) or Toward Tx (to the left).

Input / Output Port: (Available when plotting Components data) Selects the input and output ports for plotting the desired S-parameters.

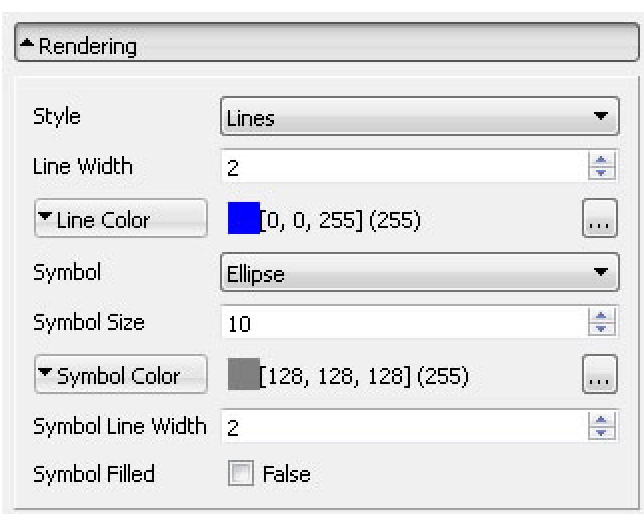
Visible: Toggles whether the trace is displayed in the Plot.

Custom Legend: This is a toggle panel folder. When switched on, Custom Legend becomes a panel folder. Expand this panel folder to specify the custom text of the trace in the plot legend. You can click the button to the right of the entry field to insert a variety of symbols (encoded as rich text). The plot legend appears at the bottom of the plot window.

> **Tip:** If you are having issues seeing the full legend in the plot window after changing a custom legend label, try adjusting the size of the plot window.

> **Tip:** HTML tags may be added to the legend text to control the font and formatting.

Rendering Panel Folder: The rendering panel folder contains settings for configuring how the plot trace is drawn: line style, line width, line color, symbol type, symbol size, symbol color, symbol line width, and symbol fill. For plots with many traces there are many options for differentiating them.



Style: Specify the trace line style: None, Lines, Dotted, and so on.

Line Width: Specify the line thickness.

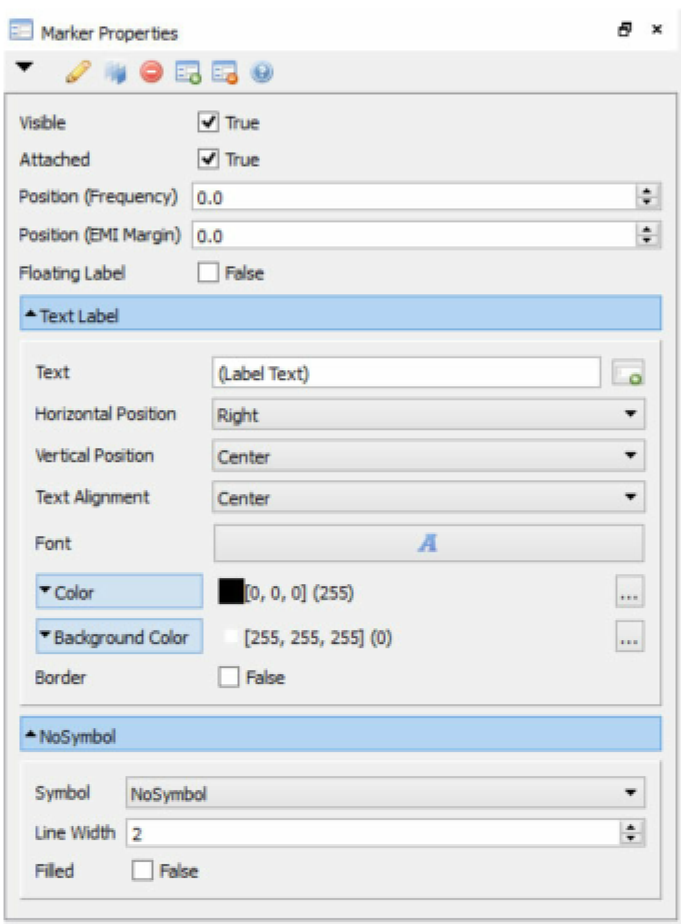
Line Color: Set the line color. Expand this panel folder to set the red-green-blue (RGB) levels for the line color. Click the ... button to choose the color from a palette.

Symbol: Select a marker symbol to be placed at every sample point: No Symbol (default), Ellipse, Rect, and so on.

Plot Marker Properties

By default, the Result Plot has a marker on the highest EMI margin. The associated marker label will specify the EMI Margin and the origin of the interfering signal by default, but can be customized if desired.

The Marker property panel contains settings to control the text, symbol, and rendering properties of the marker in the window. It also allows you to move the marker and attach or detach it from the plotted data.



Visible: Toggles whether the marker is displayed in the Plot.

Attached: Toggle on/off whether the marker is positioned in axes coordinates or plot window coordinates. Determines whether the marker is positioned relative to the axes (and therefore moves with the scale) or relative to the plot window. Markers can be moved within the plot window by clicking on them and then dragging. Attached markers will snap to the plot data point nearest to the mouse cursor when dragging.

Position (Frequency): Set the X-axis (frequency) position of the attached marker. This settings is only visible when Attached is enabled (true).

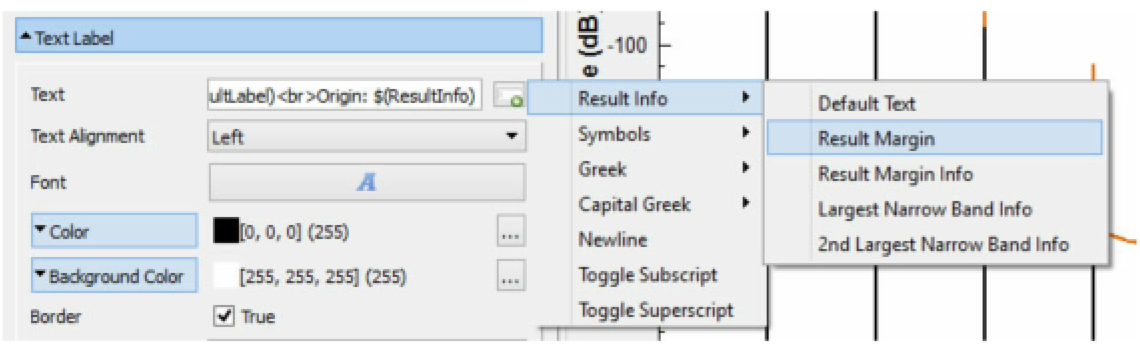
Position (EMI Margin): Set the Y-axis (EMI Margin) position of the attached marker. This settings is only visible when Attached is enabled (true).

Floating Label: Toggle on/off whether the label text behaves as a floating label or is positioned adjacent to attached marker symbol. When switched off (default), the text label is kept next to the marker, which is positioned relative to the axes. When switched on, the text label behaves like a floating marker and is positioned relative to the window frame. This setting is only visible when Attached is switched on (true).

Position from Left, Top (%): Set the position of the unattached marker (or floating label) as a fraction (%) of the plot window size. This pair of settings is only visible when Attached is switched off (false) or when Floating Label is switched on (true). Unattached markers (and floating labels) can also be interactively positioned in the plot or polar plot window by clicking on them and then dragging.

Text Label Panel Folder: The Text Label panel folder contains settings for configuring the text, symbol, and rendering of the marker.

Text: Specify the marker text. The label supports rich-text formatting, so you can insert HTML-style text formatting and special characters. While editing the text, you can click the button to the right of the entry field to insert a variety of symbols (encoded as rich text) at the current cursor position. Additional options for EMI Info are available in Result Plots as shown below.



Horizontal Position: Set the horizontal position of the label relative to the position of the marker. This setting is only visible when *Attached* is switched on and *Floating Label* is switched off.

Vertical Position: Set the vertical position of the label relative to the position of the marker. This setting is only visible when *Attached* is switched on and *Floating Label* is switched off.

Text Alignment: Set the alignment (that is, justification) of the text. This setting only effects multi-line text.

Font: Set the font of the marker text. Certain types of rich-text formatting in the text itself may override this.

Color: Set the color of the marker text. Expand this panel folder to set the red-green-blue (RGB) levels for the text color. Click the icon button to choose the color from a palette.

Background Color: Set the background color of the marker text. Expand this panel folder to set the red-green-blue (RGB) levels for the text background color. Click the icon button to choose the color from a palette.

Border: Toggle on/off whether a border is displayed around the text label.

Border Width: Set the thickness of the text label border. This setting is only visible when *Border* is switched on.

Border Color: Set the color of the text label border. Expand this panel folder to set the red-green-blue (RGB) levels for the text label border color. Click the icon button to choose the color from a palette. This setting is only visible when *Border* is switched on.

Symbol Panel Folder: The Symbol panel folder contains settings for configuring an optional symbol displayed at the position of an attached marker. It is only available when *Attached* is switched on (true).

Symbol: Select a symbol to be placed at the marker position. Choose **NoSymbol** to remove the symbol.

Symbol Size: Set the size (in points) of the symbol.

Symbol Color: Set the color of the symbol.

Symbol Line Width: Set the line thickness symbol.

Symbol Filled: Fill the interior of the symbol with the Symbol Color.

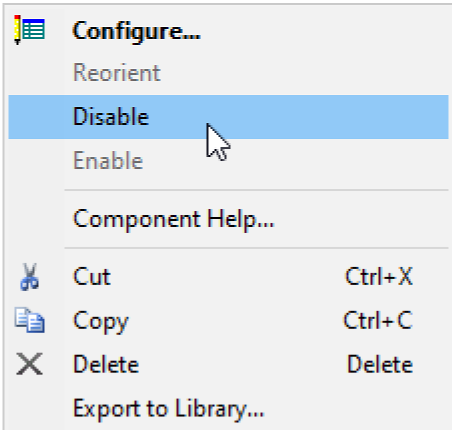
Enabling and Disabling EMIT Components

Emitters and Radios can be enabled or disabled from the Schematic editor. Disabled components are omitted from the analysis. Any RF components connected to a disabled radio are also omitted.

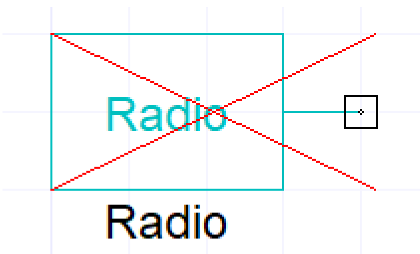
By default, placed components are enabled.

To disable:

1. Select one or more Radios/Emitters in the Schematic Editor.
2. Right-click and select **Disable**.

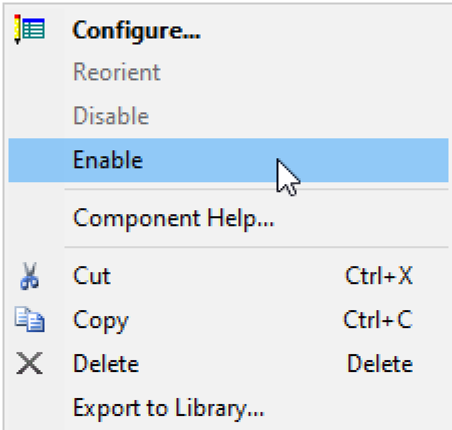


A red X appears over the component(s):

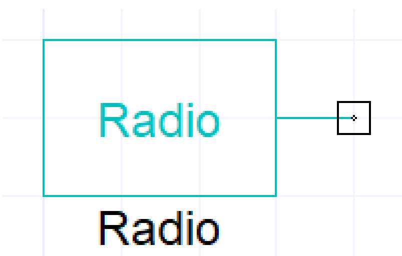


To enable:

1. Select one or more Radios/Emitters in the Schematic Editor.
2. Right-click and select **Enable**.



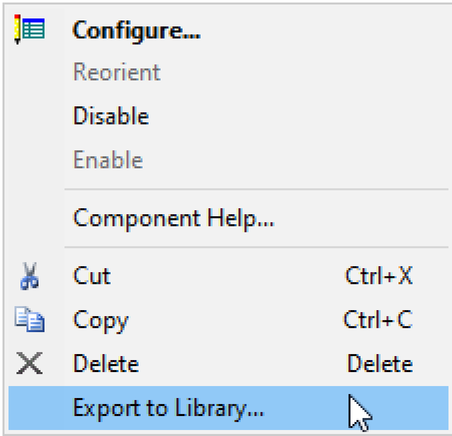
The red X disappears from over the component(s).



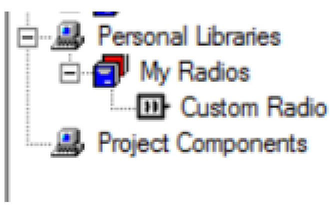
Saving EMIT Components to the Library

You can save EMIT components you've created to your personal library so that they are available for use in other EMIT designs.

1. To do this, right-click an EMIT component and select **Export to Library**.



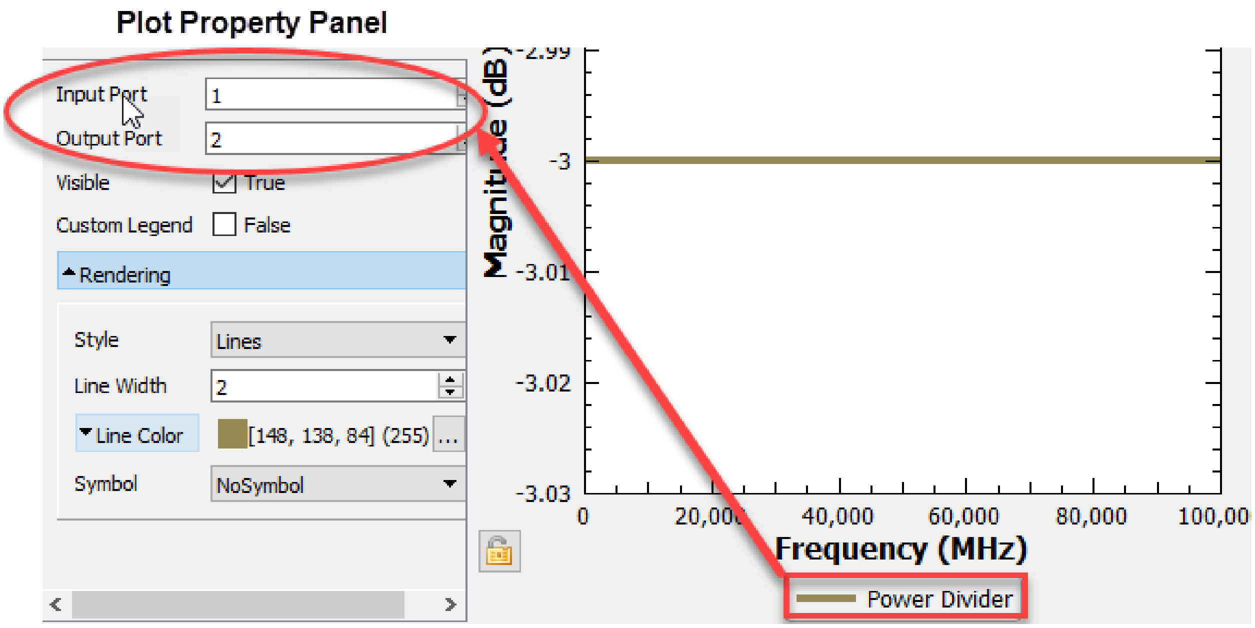
2. In the resulting dialog box, choose which library to place the component into or select a current library. In the dialog box shown below, the *Custom Radio* component is placed into PersonalLib in a library folder called *My Radios*. The resulting library component uses the instance name of the component in the schematic, in this case *Custom Radio*.



The models in the library are now available for use in all EMIT designs across all projects.

Plotting Multiport Spectral Profiles

When plotting the spectral profile of components with two or more ports, you can control the particular port pair being plotted from the Plots property panel. To access this, click the legend in the plot area. This automatically displays the plot trace property panel in the Properties window where the port data being displayed can be changed as shown below.



Plot Property Panel
Input Port
1
Output Port
2
Visible
True
Custom Legend
False
Rendering
Style
Lines
Line Width
2
Line Color
[148, 138, 84] (255)
Symbol
NoSymbol
Magnitude (dB)
-2.99
-3
-3.01
-3.02
-3.03
0
20,000
40,000
60,000
80,000
100,000
Frequency (MHz)
Power Divider

5 - EMIT Theory

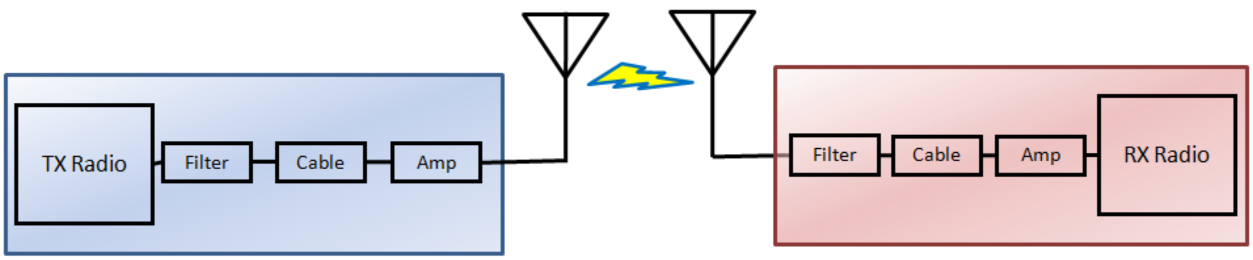
EMIT's powerful solver engine performs a power flow analysis for wideband signal spectra originating at one or more Transmitter (Tx) and traveling, sometimes via very complex routes, to each victim Receiver (Rx).

Upon arrival at each Rx, EMIT compares the received signal spectrum with the Rx's susceptibility to compute several interference metrics (EMI Margins, Sensitivity, Desense or Availability) and quantify the interference at the Rx. Each time a signal spectrum encounters a component, it is modified by the component's wideband characteristics, including potential nonlinear effects.

Each signal spectrum in EMIT can contain both narrowband and broadband elements. This section provides a brief overview of the methodology used In EMIT's simulation engine.

(1-1) Simulation Theory

EMIT's (1-1) simulation looks at each individual Tx/Rx pair in the design, along with any components (filters, cables, amplifiers, multiplexers, and so on) that may be present.
EMIT simulates each Tx/Rx pair separately to compute each of the interference metrics. Each component in the Tx/Rx chain is modeled in EMIT by its frequency-dependent characteristics (called a Spectral Profile) that are either supplied by the user or computed by EMIT.



For predicting cosite interference, EMIT calculates an EMI margin, which is a metric that compares the received interference power level to the susceptibility of the Rx. EMIT calculates three different EMI margins to help identify the cause of any interference:

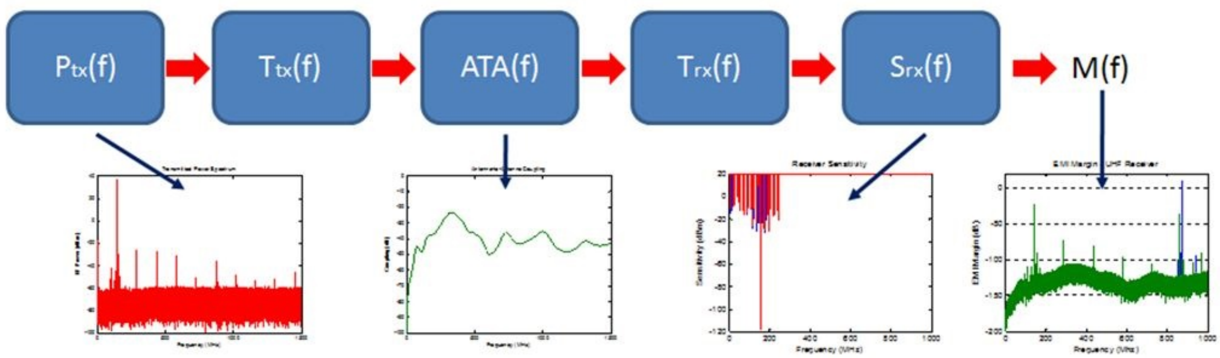
- Point EMI Margin
- Peak in-channel EMI Margin
- Noise in-channel EMI Margin

Other interference metrics available are Sensitivity, Desense and Availability . Each of these is derived from the computed EMI Margin. All calculations are described in further detail below.

Point EMI Margin

Point EMI Margin calculations proceed by cascading all spectral profiles In the Tx/Rx chain to arrive at the Point EMI Margin (M) for the Tx/Rx pair. This calculation is performed for each Tx/Rx pair In the scene. This is shown conceptually in the figure below.

> **Note:**
>
> The cascaded outboard component spectral profiles (T(*f*)) are not shown for clarity.



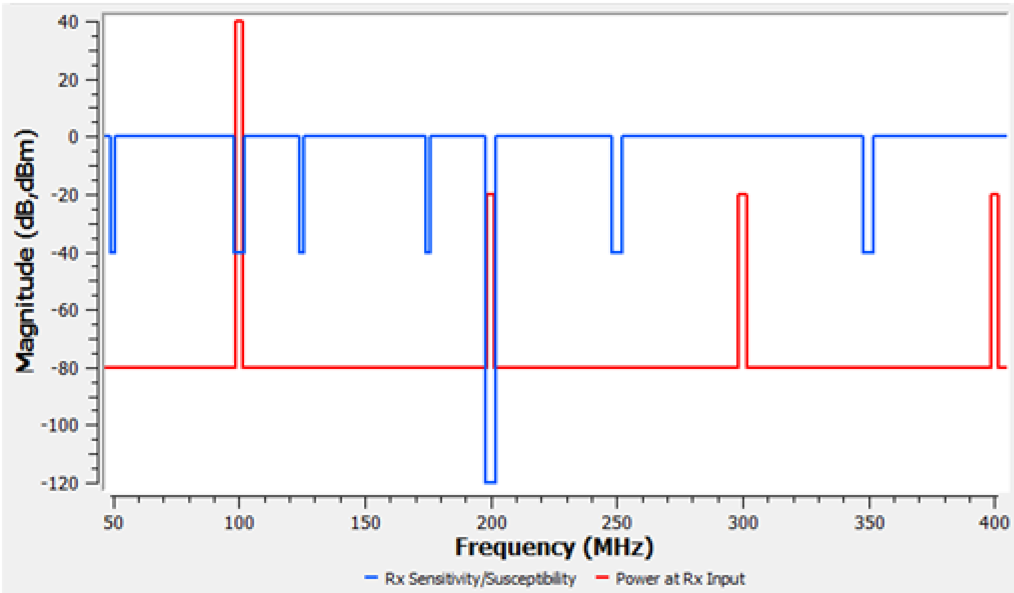
$$\mathrm{M}\left(f\right)=\mathrm{P_{tx}}\left(f\right)+\mathrm{T_{tx}}\left(f\right)+\mathrm{ATA}\left(f\right)+\mathrm{T_{rx}}\left(f\right)-\mathrm{S_{rx}}\left(f\right)$$

Where:

- M = Point EMI Margin (dB)
- P_{tx} = Transmitted Power (dBm)
- T_{tx} = Sum of all Tx Outboard Component Transfer Functions (dB)
- ATA = Antenna-to-Antenna Coupling (dB)
- T_{rx} = Sum of all Rx Outboard Component Transfer Functions (dB)
- S_{rx} = Receiver Susceptibility (dBm)

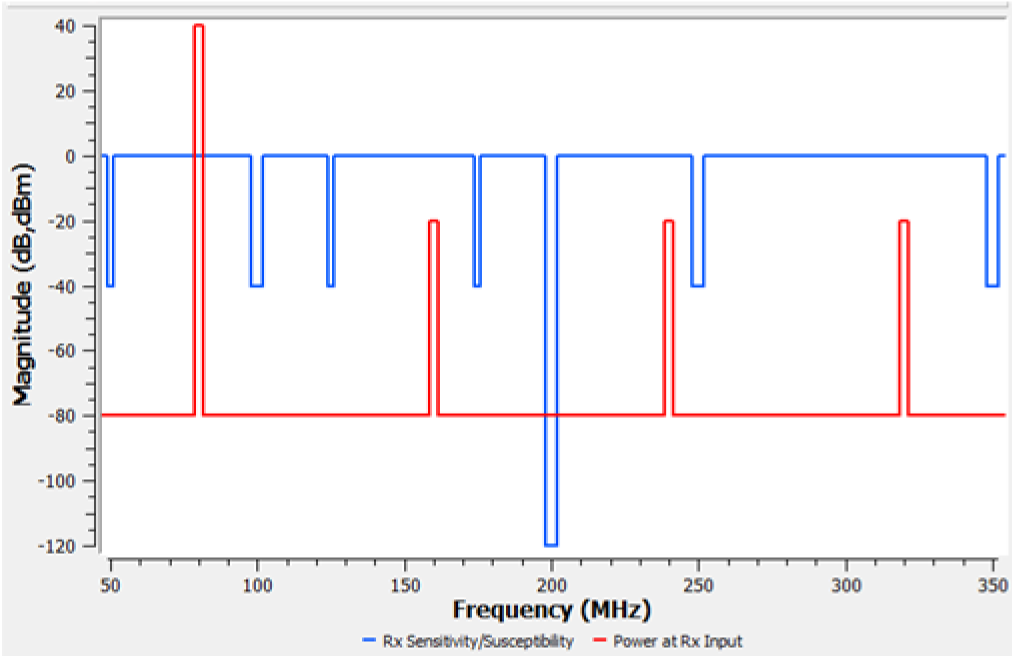
The Point EMI margin can only be computed over the frequency range for which all spectral profiles in the Tx/Rx chain are defined. EMIT automatically determines the frequency range common to all spectral profiles during simulation. The Point EMI margin will thus be computed from a frequency equal to the largest minimum frequency for all of the spectral profiles up to a frequency equal to the largest maximum frequency for all of the spectral profiles.

An EMI margin that exceeds a set threshold (0 dB by default) indicates interference at the Rx. This occurs when the power at the Rx input exceeds the susceptibility of the Rx. The plot below shows the spectral profiles representing the power at the Rx input (red) and the Rx susceptibility (blue). Assuming a fixed value coupling of 0 dB is used, there will be two frequencies for this case where the Point EMI margin is positive, representing Interference. At 100 MHz and 200 MHz the Rx Input power (red) exceeds the Rx susceptibility (blue).



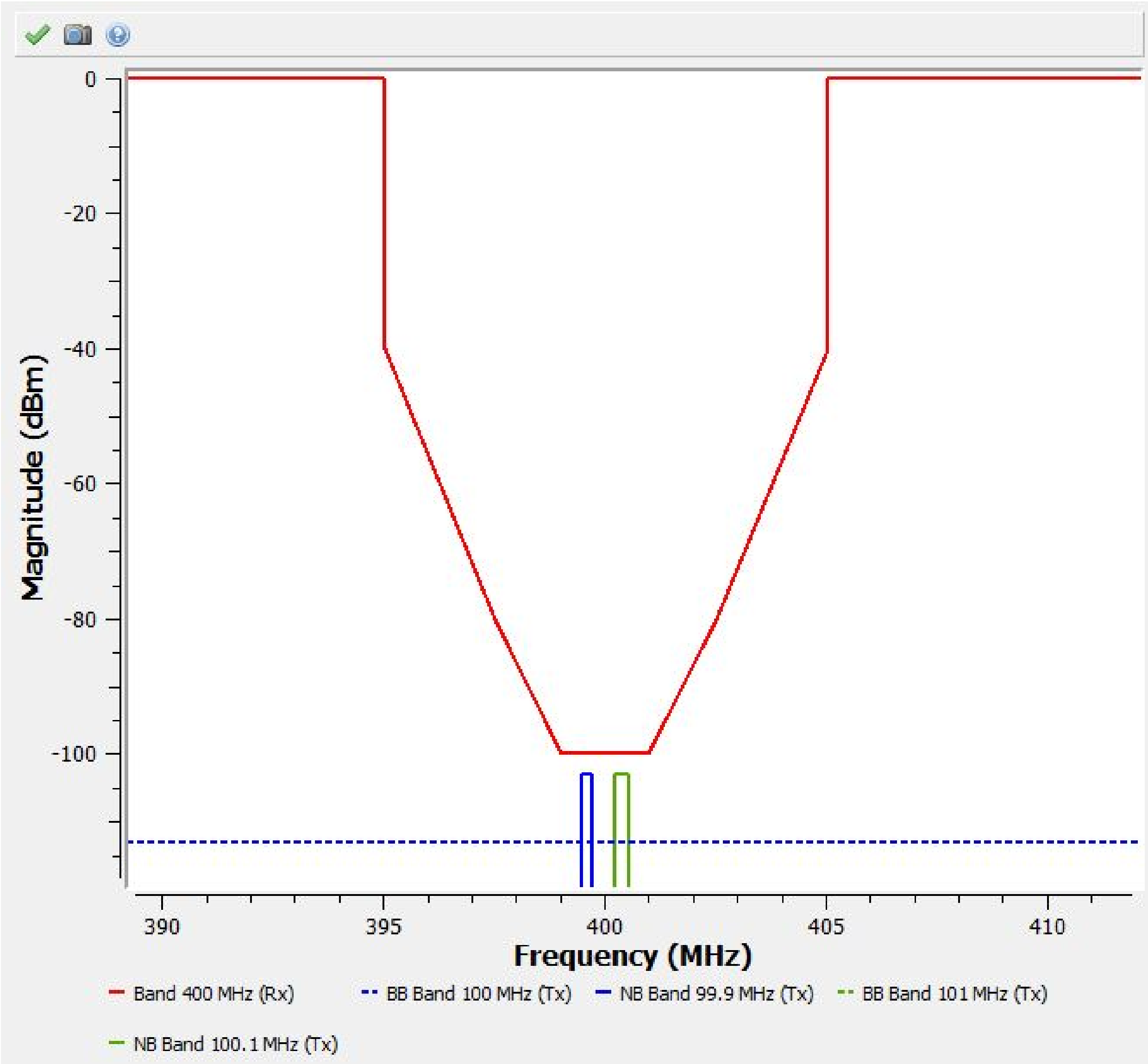
The Rx Saturation Level for this example is at 0 dBm and the Tx Broadband Noise is at -80 dBm. These extreme values may or may not represent the actual system performance, depending on how the Tx and Rx characteristics were produced. For example, if the susceptibility of the Rx has been measured, the saturation level may represent the maximum power level that could confidently be put Into the Rx front end without damaging it. This does not necessarily indicate that interference will occur at that input power level, but that tests were not performed at higher power levels to determine the interference threshold. On the Tx side, the noise floor could represent the dynamic range of the instruments used for the measurements and not the actual broadband noise emissions of the Tx. In cases like this, the EMIT simulation may produce positive EMI margins when in reality they would not be present.

Consider the plot below, which is similar to the case above except for a different Tx frequency. Computing the Point EMI Margin as outlined above will result In a positive EMI margin at 80 MHz where the Rx Input power (red) exceeds the Rx susceptibility (blue), and at 200 MHz where the Tx noise floor at -80 dBm exceeds the susceptibility of the Rx. If the 0 dB Rx saturation level truly represents a power level that causes Interference then this is a real Interference event. On the other hand, if the 0 dB saturation level is a limitation of the Rx characterization and not a true indicator of interference as described above, then this positive EMI margin could be a false alarm.



Peak In-Channel EMI Margin

The second EMI Margin calculated by EMIT is the Peak In-Channel EMI Margin. The Peak In-Channel EMI Margin calculates the total power in the Rx channel due to narrowband (NB) signal components. As shown In the figure below, multiple NB signals can lie within the Rx channel bandwidth. If the amplitude of these signals is below the susceptibility envelope (red) then the Point EMI Margin will be negative. However, the total power at the Rx's detector may be above the susceptibility threshold.



The Peak EMI Margin is calculated by summing the peak power of each NB signal component that falls within the Rx channel's 3-dB bandwidth and comparing it with the in-channel susceptibility:

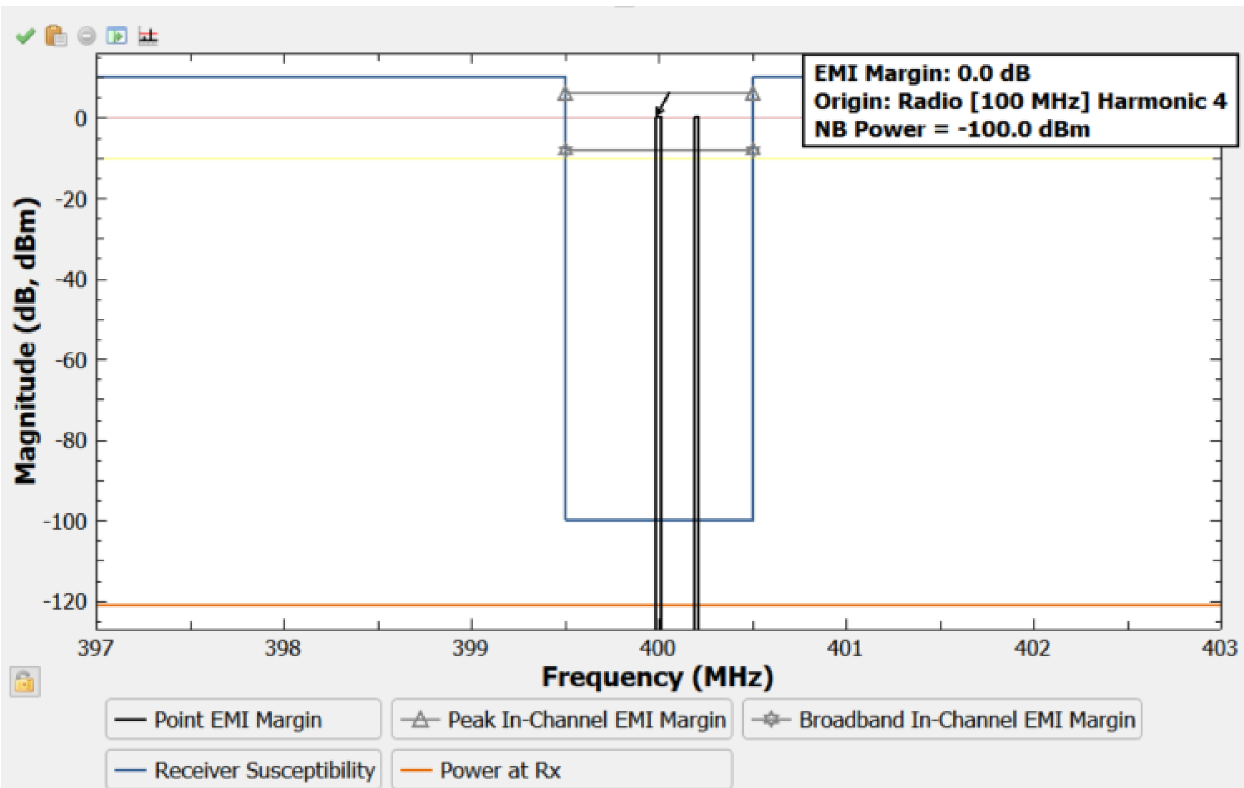
$$\mathrm{P_{peak}} = 20 * \log_{10}\left(\sum_{\mathrm{i=1}}^{N}\sqrt{P_i}\right)\ [dB]$$

$$\mathrm{M_{peak}} = \mathrm{P_{peak}} - \mathrm{S_{rx}}\ [dB]$$

Where:

- P_i = Peak power (in Watts) of the i^{th} narrowband signal component
- P_{peak} = Total power of all *N* narrowband signal components (dB)
- S_{rx} = Receiver in-channel susceptibility (dB)
- M_{peak} = Peak EMI Margin (dB)

As with the Point EMI Margin, a positive Peak EMI Margin indicates interference at the Rx. This occurs when the total power due to NB signal components within the Rx channel exceeds the susceptibility of the Rx. The Peak EMI Margin is plotted as gray triangles in the figure below. Since the Peak EMI Margin is approximately 6 dB, there is marginal interference at the Rx. In this example, a fixed coupling of 0 dB was used so the amplitude of both these narrowband signals at the input to the Rx is -100 dB which is below the Rx's susceptibility envelope, as expected.



Broadband In-Channel EMI Margin

The Broadband In-Channel EMI Margin calculates the potential for interference due to broadband noise In the Rx channel bandwidth. To compute this, the total noise power within the channel bandwidth is calculated and compared to the Rx in-channel susceptibility:

$$\mathrm{P_{Noise}} = \int_{f1}^{f2} \mathrm{P_n}\,(f)df \qquad [W]$$

$$\mathrm{M_{Noise}} = \mathrm{P_{Noise}} + 3 - \mathrm{S_{rx}} \qquad [dB]$$

Where the +3 dB in the equation for M_{noise} converts the average power level to peak power level of the in-channel broadband noise, and:

- P_{Noise} = Total in-channel average noise power (Watts)
- $P_n(f)$ = Noisepower spectral density (W/Hz)
- f1 = $f_{\mathrm{rx}} - \frac{\mathrm{BW_{rx}}}{2}$
- f2 = $f_{\mathrm{rx}} - \frac{\mathrm{BW_{rx}}}{2}$
- BW_{rx} = 3 dB IF Filter bandwidth
- f_{rx} = Rx channel center frequency
- S_{rx} = Rx in-channel susceptibility (dB)
- M_{Noise} = noise in-channel EMI Margin (db)

A positive Broadband In-Channel EMI Margin indicates interference at the Rx. This occurs when the total power due to broadband noise within the Rx channel exceeds the susceptibility of the Rx. The Broadband In-Channel EMI Margin is plotted as gray stars in the previous figure. Since the Point EMI Margin is approximately 0 dB and the Noise in-channel EMI Margin is approximately -8 dB, only the Point EMI Margin makes a significant contribution to the Peak In-Channel EMI Margin, which is not always the case.

Sensitivity

For many applications, it is desirable to compute the sensitivity of a Rx in the presence of "noise" signals. In this context, "noise" is intended to denote an interference spectrum and generally consists of both narrowband and broadband components. The sensitivity is defined as the minimum power required for the Signal of Interest (SoI) to be able to receive and decode the signal properly. The sensitivity is given by the following equation:

$$\text{(Equation 1)} \qquad \mathrm{P_{sens}} = \left(\frac{\mathrm{S}}{\mathrm{N}}\right)\left(\mathrm{P_{Rx\text{-}Noise}} + \mathrm{P_1}\right)$$

Where:

- S/N = Minimum signal to noise ratio required by the receiver
    - The required S/N ratio (where N is taken to be Noise + Interference) is defined in the Rx Spectral Profile
- $P_{Rx\text{ - }Noise}$ = Receiver noise floor = kTBF
- kT = Boltzmann's constant * absolute temperature
- B = Receiver bandwidth
- F = Receiver noise factor
- P_1 = Total in-channel interference power

The total interference power, P_1,is the sum of all broadband and narrowband Interference within the Rx bandwidth and is given by:

$$(\text{Equation 2}) \quad P_1 = B \cdot \sum P_{BB} + \sum P_{NB}$$

Where:

- P_{BB} = Broadband noise power density
- P_{NB} = Narrowband noise power components

EMIT computes these interfering powers as part of the calculation tor the Noise in-channel EMI Margin and Peak in-channel EMI Margin.

Consider the relationship between the EMI Margin (M) that EMIT traditionally computes and the sensitivity, as defined above. EMIT defines the Rx susceptibility (S_{Rx}) as the minimum interference power that will cause interference to the Rx. Although S_{Rx} is a wideband quantity, here we consider only the in-channel susceptibility, which is given by:

$$(\text{Equation 3}) \quad S_{Rx} = \frac{P_{min}}{(S/N)}$$

From (Equation 1), see that:

$$(\text{Equation 4}) \quad (S/N) = \frac{P_{sens}}{(P_{Rx\text{-}Noise} + P_1)}$$

So (Equation 3) can be written as:

$$(\text{Equation 5}) \quad S_{Rx} = \frac{P_{min}(P_{Rx\text{-}Noise} + P_1)}{P_{sens}}$$

Solving for sensitivity yields:

$$(\text{Equation 6}) \quad P_{sens} = \frac{P_{min}(P_{Rx\text{-}Noise} + P_1)}{S_{Rx}}$$

EMIT's EMI Margin (M) is by definition given by:

$$(\text{Equation 7}) \quad M = \frac{(P_{Rx\text{-}Noise} + P_1)}{S_{Rx}}$$

From (Equation 3), this can be rewritten as:

$$(\text{Equation 8}) \quad M = \frac{(P_{Rx\text{-}Noise} + P_1)}{(P_{min}/(S/N))} = \frac{(S/N)(P_{Rx\text{-}Noise} + P_1)}{P_{min}} = \frac{P_{sens}}{P_{min}}$$

Thus, the simple relationship between EMIT's EMI Margin and Sensitivity:

$$(\text{Equation 9}) \quad P_{sens} = M \cdot P_{min}$$

Once all of the EMI Margins have been computed in EMIT, the Sensitivity is computed using (Equation 9) with the P_{min} value known from the Rx Spectral Profile definition. However, the sensitivity value will never be permitted to be less than the Rx's P_{min} since that value is taken to be the absolute smallest signal that the receiver can decode property.

> **Note:**
>
> Sensitivity is defined in the Rx channel bandwidth and only appears on plots as a single symbol at the Rx channel frequency.

Desense

The Desense of an Rx channel quantifies how much the sensitivity of an Rx channel is degraded in the presence of an in-channel interference. That is, how much the total in-channel interfering power exceeds the Rx's in-channel susceptibility. This is computed simply as the sum of the Peak in-channel EMI Margin and the Noise in-channel EMI Margin.

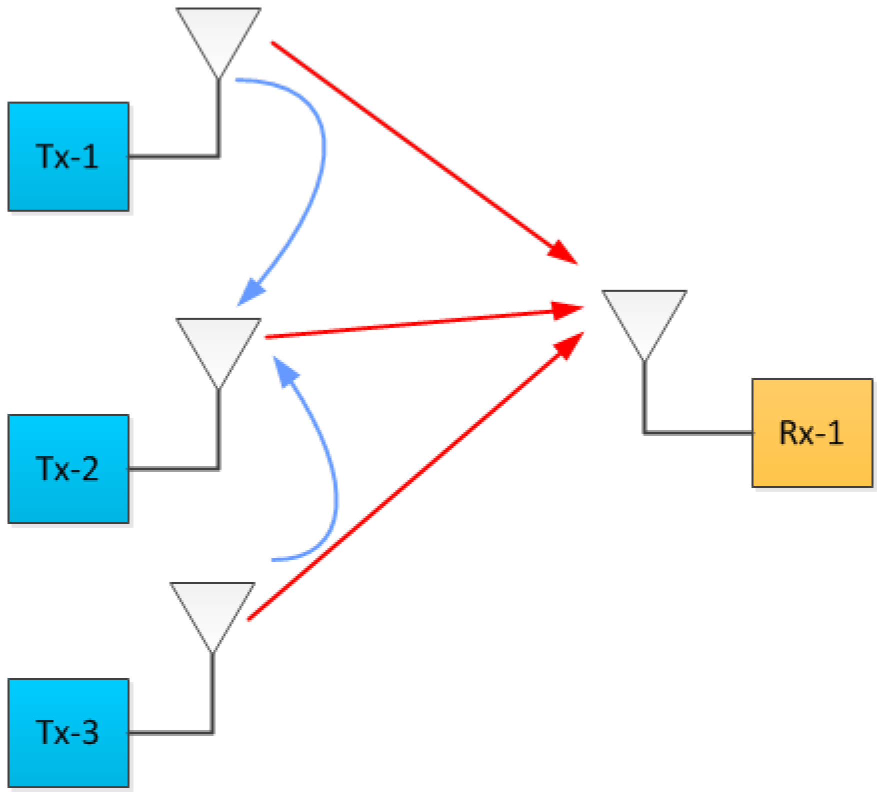
This value also reflects how much desensitization ("desense") a Rx experiences in the presence of interference as compared to P_{min}. Positive (in dB) values of Desense indicate that the Rx's sensitivity has been reduced as compared to P_{min},while negative (in dB) values of Desense indicate that the in-channel interference is below the Rx's interference threshold.

Availability

The availability is given as a percentage (%) and quantifies what percentage of Rx channels in a Rx Band are experiencing interference (as defined by the Availability EMI Margin Threshold). The Tx and Rx band or channels for displaying availability are selected in the EMIT analysis window. Availability provides a useful way to mimic frequency-hopping systems.

(N-1) Simulation Theory

EMIT's (N-1) simulation capability computes EMI Margins for Rx channel frequencies that are potential victims of multiple Txs operating simultaneously, including transmitter-to-transmitter intermodulation (Tx-Tx intermod). This analysis looks at the effect of multiple transmitters on the interference at each Rx. It computes the interference at a Rx due to the superposition of all Tx signals and the Intermodulation Products (IMPs) created in one Tx due to radiation of another Tx coupling into its front end. The IMP are then re-radiated and can interfere with a Rx. Alternatively, the spectra of multiple Txs can couple into the Rx's front end, where IMPs are created due to nonlinearities in the low noise amplifier (LNA). The figure below conceptually shows these two scenarios. Signals from all three transmitters are coupling into the receiver, where they can generate MPs in the Rx's front end.



Another potential cause of IMPs are those that arise due to the nonlinear mixing of the two signals that occurs at junctions of dissimilar metals near the Txs (the so-called "rusty bolt effect"). EMIT is currently unable to model the rusty bolt effect or other Passive Intermodulation (PIM) phenomena.

In EMIT, Tx-Tx intermods can be generated for Txs (or RTs) that have outboard amplifiers or that have an internal amplifier defined in the Tx Spectral Profile properties. It should be noted that the generation of IMPs in EMIT is not limited to (N-1) simulations or to computing Tx-Tx intermod. Any time more than one narrowband signal component enters an amplifier that is located anywhere in the signal path (Tx and/or Rx), there is the potential for IMPs and EMIT fully accounts for this.

For the inter-TX intermod described above, the power at the input of an amplifier in a "victim" Tx due to another co-located Tx is calculated according to:

$$\mathrm{P_{in,amp}}\left(f\right) = \mathrm{P_{tx,i}}\left(f\right) + \mathrm{T_{tx,i}}\left(f\right) + \mathrm{ATA}\left(f\right) + \mathrm{T_{tx,v}}\left(f\right)$$

Where:

- $P_{in,amp}$ = Power at input terminal of the amplifier in the victim Tx (dBm)
- $P_{tx,i}$ = Transmit power of the interfering Tx (dBm)
- $T_{tx,i}$ = Sum of all interfering Tx outboard component transfer functions (dB)

- $T_{tx,v}$ = Sum of all victim Tx outboard component transfer functions, including amplifier reverse isolation (dB)

The output power spectrum of the victim Tx then couples to a Rx in the same manner that a typical Tx output spectrum couples to a victim Rx. The Tx-Tx intermods are also generated in the same manner as MPs in the Rx front end LNA.

In general, the following procedure is used to generate the output of an amplifier for all amplifiers, regardless of their location in the Tx or Rx signal paths:

1. Combine the spectra of all incoming signals. For Tx-Tx intermods this requires combining the spectra of the "victim" Tx with the interfering Tx (or Txs). For amplifiers in the Rx front end, this requires combining the spectra of all Txs that couple into the Rx including any Tx-Tx spectra.
   - Broadband (BB) emissions are converted from dBm/Hz to Volts/Hz (assuming R = 1Ohm) and added together.
   - Narrowband (NB) components are added on a frequency basis. If two or more frequencies overlap, the power levels are combined.
2. The instantaneous voltage level at the input to the amplifier is calculated to check for amplifier saturation. The power levels at all frequencies are converted to voltages. These voltages are summed to compute the instantaneous voltage (assuming all spectra arrive at the Rx simultaneously). If the total voltage is greater than the 1-dB compression point for the amplifier, then the amplifier is saturated and the simulation ends.
3. The center frequency of each NB signal component is determined. Each NB signal component is treated as a tone with an amplitude determined by the peak power of the NB signal component. A "two-tone" nonlinear analysis is then performed for every pair of NB tones at the input to the amplifier.
4. The Input Intercept Points (IIP) are estimated using the following approximations:

   $$\mathrm{IIP}_2 = \mathrm{P}_{1dB} + 27$$

   $$\mathrm{IIP}_3 = \text{As specified in amplifier model}$$

   $$\mathrm{IIP}_4 = \mathrm{IIP}_3 - 2$$

   $$\mathrm{IIP}_5 = \mathrm{IIP}_4 - 2$$

   ...

   $$\mathrm{IIP}_N = \mathrm{IIP}_{N\text{-}1} - 2$$

   Where:

   - IIP_N = N^{th} order intercept points referred to the input
   - P_{1dB} = 1dB compression point referred to the input
5. The IMPs are then calculated from the IIPs according to:

$$\mathrm{IMP}_{\mathrm{n}}^{\mathrm{L}} = \mathrm{P}_2 - (n-1) * (\mathrm{IIP}_n - \mathrm{P}_1)$$

$$\mathrm{IMP}_{\mathrm{n}}^{\mathrm{H}} = \mathrm{P}_1 - (n-1) * (\mathrm{IIP}_n - \mathrm{P}_2)$$

Where:

- $\mathrm{IMP}_{\mathrm{n}}^{\mathrm{L}}$ = The lower frequency n^{th} order intermod product (dBm)
- $\mathrm{IMP}_{\mathrm{n}}^{\mathrm{H}}$ = The higher frequency n^{th} order intermod product (dBm)
- P_1, P_2 = Power level of the two tones (dBm)

6. The bandwidth of each IMP Is computed by:

$$\mathrm{BW}_{\mathrm{total}} = \mathrm{p} * \mathrm{BW}_1 + \mathrm{q} * \mathrm{BW}_2$$

Where:

- BW_1 and BW_2 = The bandwidths of the two individual NB components
- p and q = The order of the two tones

7. This process is repeated for all two-tone pairs at the input to the amplifier.

The approximation used for computing higher order IIPs shown above will eventually result in IIPs in the linear portion of the gain curve, which is nonphysical. In other words, the assumption becomes invalid. For this reason, EMIT computes successive higher order IIPs and stops at the last one that is greater than the 1dB compression point. EMIT then proceeds to compute the amplitude of the IMPs and does so for all up to the highest order IIP available. If the amplitude of any of these falls below the noise level, EMIT will discard it.

Alternatively, the nonlinear performance of an amplifier can be specified using the Harmonic Input Intercept Points table. When measuring amplifier performance, it is usually easier to measure the amplitude of harmonics produced by an amplifier being driven by a single-frequency input signal. This allows the harmonics to be characterized by a Harmonic Input Intercept Point (HIIP). Measurement of the output harmonic levels yields an $HIIP_n$ for each of the *n* harmonics. The HIIP has the same meaning as an IIP but is specific to the harmonics.

Following the formulation presented by R.L.Smith in a paper at Microwave Journal, we can estimate the amplitude of IMPs from the HIIPs using the following equation:

$$\mathrm{IMP}_n = \alpha \mathrm{P}_A + \beta \mathrm{P}_B - (n-1)\mathrm{HIIP}_n + 20\log\left(\frac{\mathrm{n}!}{(\alpha!)(\beta!)}\right)$$

Refer to the referenced paper for full definitions of the variables.

EMIT provides the option of specifying HIIPs for an amplifier as an alternative to using the approximate IIP recursion relations shown previously. If an HIIP is defined in EMIT for an amplifier, then the MPs are computed according to the above equation where the HIIPs provided are used directly and others will be approximated according to:

$$\text{HIIP}_n = \text{HIIP}_{\text{n-1}} - 2\,[\text{dB}]$$

EMIT then calculates the various interference metrics using this combined spectrum (Tx-Tx intermods, Rx generated MPs, etc.) and the simulation proceeds in a similar manner to the (1-1) simulation. Note that the order of the components matters for the nonlinear analysis. For example, if a transmitter has an amplifier and filter, the amplitude of any MPs created In the filter will be highly dependent on whether the filter is between the Radio and the Amplifier or if the filter is between the Amplifier and Antenna.

In general, the number of Tx/Rx channel combinations that need to be simulated can grow very quickly even for moderately sized projects. For example, a simple 5 Tx and 5 Rx scenario with each Tx or Rx operating on 20 channels will contain 10,000 Tx/Rx channel pairs (including self-interactions). Add modeling the (N-1) interactions (that is, Tx-to-Tx intermodulation and multiple Tx spectra arriving at the Rx simultaneously) and the number of channel combinations quickly balloons to almost 20 million.

Individual Tx-to-Rx simulations can be enabled/disabled by Ctrl+clicking on the associated square in the Scenario Matrix. This allows a subset of the project to be simulated.

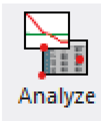
Memory requirements for extremely large simulations should be considered as well. Users can set the maximum amount of memory that EMIT can use for a simulation in the Preferences panel. This sets the maximum amount of memory that EMIT will use when running a simulation. If during a run the memory used by EMIT reaches this limit, a warning is issued and the simulation is stopped. The default value is 75% of the available memory.

With appropriate add-on Ansys Electronic HPC Packs, EMIT's computational engine can take advantage of computers with multi-core processors and graphical processing units (GPUs) to run parts of the simulation in parallel. Users can specify the number of cores (also called threads) to use for the simulation by selecting the Preferences node in the Project Manager. When Automatic Multithreading is enabled, EMIT uses all available system resources for the simulation. Deselecting this option provides a field to allow the user to select the specific number of threads to be used. For example, on a computer with a four-core CPU, a user may wish to reduce the number of cores used by EMIT to two in order to leave system resources free for other tasks while EMIT is running.

Simulations can be stopped during a run and results for all Band pairs that have completed will be available for analysis (that is, all channel combinations for a given band pair must have completed for EMIT to have the results available). If changes are made to a project after a simulation is completed, only the results for systems affected by the change are purged.

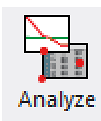
6 - Analysis and Results

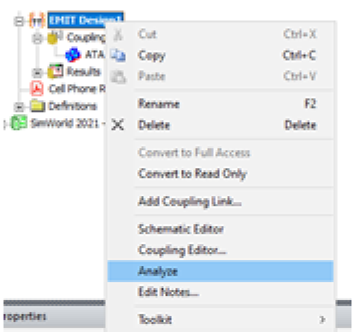
Ansys EMIT results are accessed via the **Analyze** icon on the **Simulation** ribbon menu of the EMIT design.



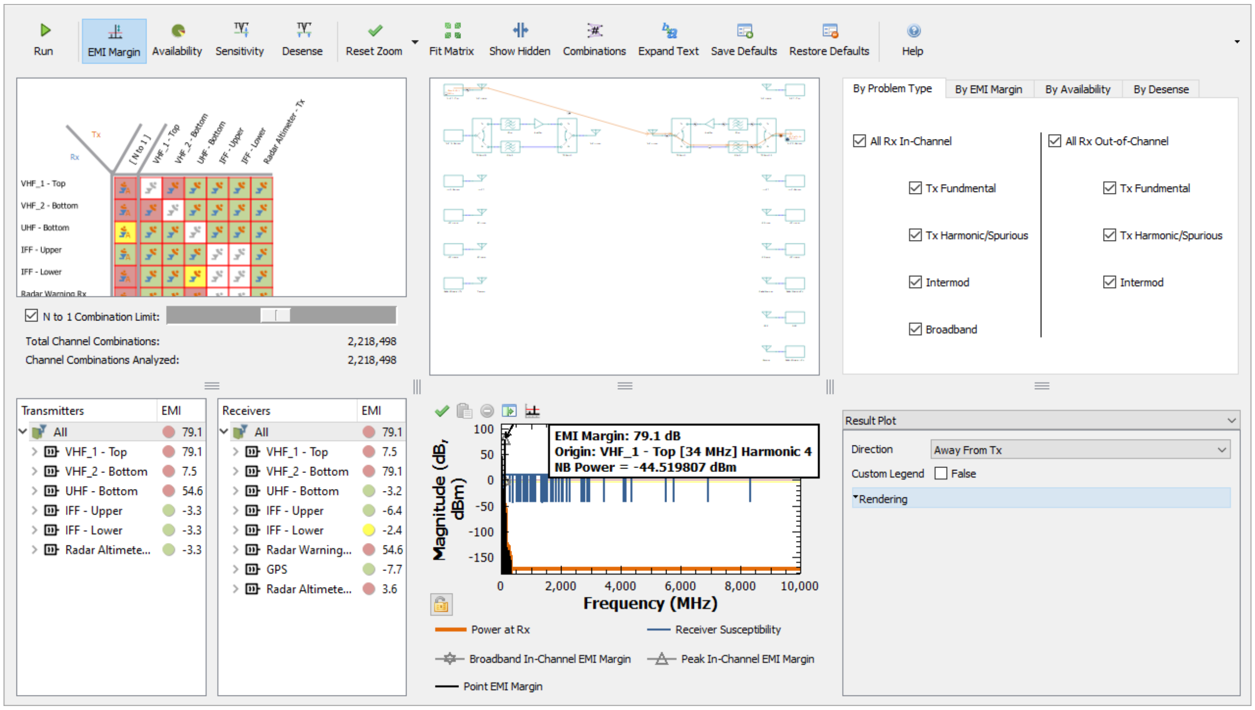
Results

An Ansys EMIT simulation is performed via the **Analyze** icon on the **Simulation** ribbon of the EMIT design or by right-clicking on the EMIT design in the Project Manager and selecting **Analyze**.

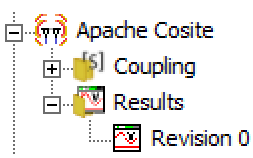




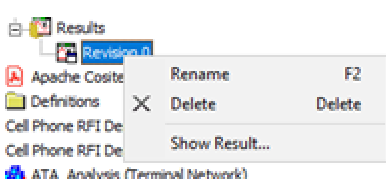
The **Analyze** action opens the EMIT Integrated Results Window, which provides a single interface for running EMIT simulations and visualizing and exploring the results



All the results and information available in the Integrated Results Window are saved and remain available within the project, even if changes are made to the initial design. In this way, the full results for multiple revisions of an EMIT design are always available via the Integrated Results Window. The results for each simulation are available in the Results folder in the Project Manager.



Each result node has options shown below via a right-click menu:

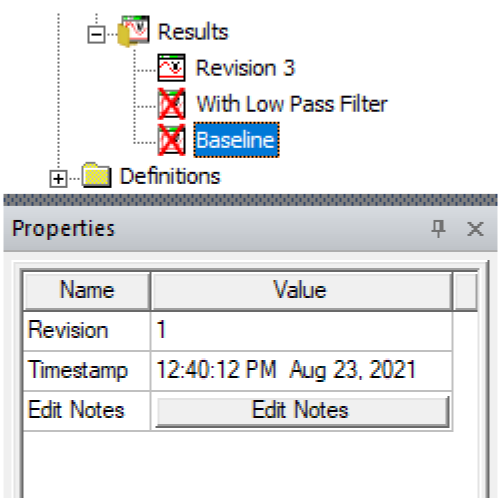


Show Resultopens the **Integrated Results Window** (or brings it to the foreground if already open) for the selected results revision. Double-clicking the results node also opens the Integrated Results Window. Multiple Integrated Results Windows can be opened simultaneously, allowing comparisons of the results for different revisions of the design.

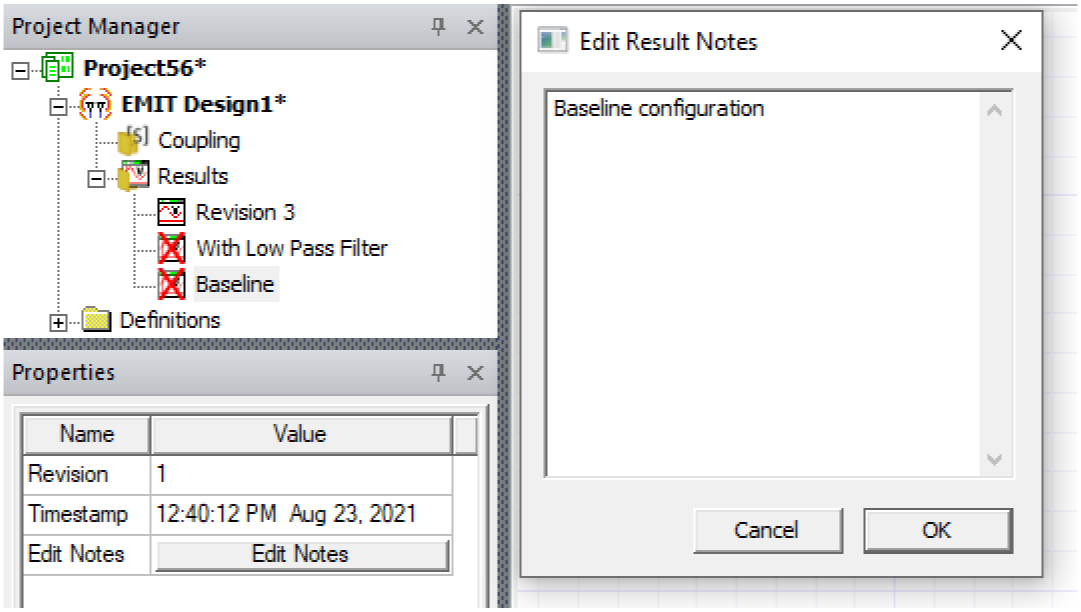
When the EMIT design is changed, the results for the previously computed design revision are no longer in sync with the current revision of the design, but are still retained within the project for future reference. To indicate that these results are no longer associated with the current revision of the EMIT design, the icon for that results node has a red “X” added to it, as shown below. (This icon also appears on the icon displayed in the title bar of the Integrated Results Window.)

When you analyze the latest revision of the design, a new results node for the current revision appears in the **Results** folder, with an automatically incremented revision number.

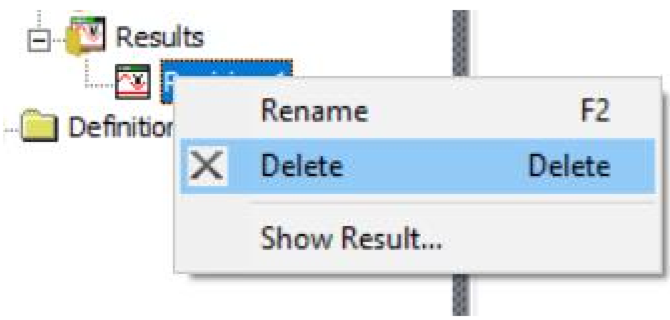
Selecting a results revision in the Project Manager displays the revision number, a timestamp showing the date and time created, and an **Edit Notes** button in the **Properties** window.



Click the **Edit Notes** button to launch the **Edit Notes** dialog box. Document the revision in the text box, and click **OK** to save the notes.

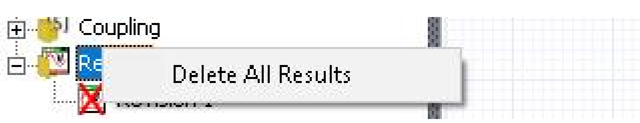


Right-click a result and choose **Delete** to delete the result. This action deletes the result data and cannot be undone. An alert will prompt to confirm the delete action.



A result can also be deleted by selecting the result and pressing the **Delete** key on the keyboard.

Delete all results for the current design using the **Delete All Results** option in the right-click menu of the **Results** folder.

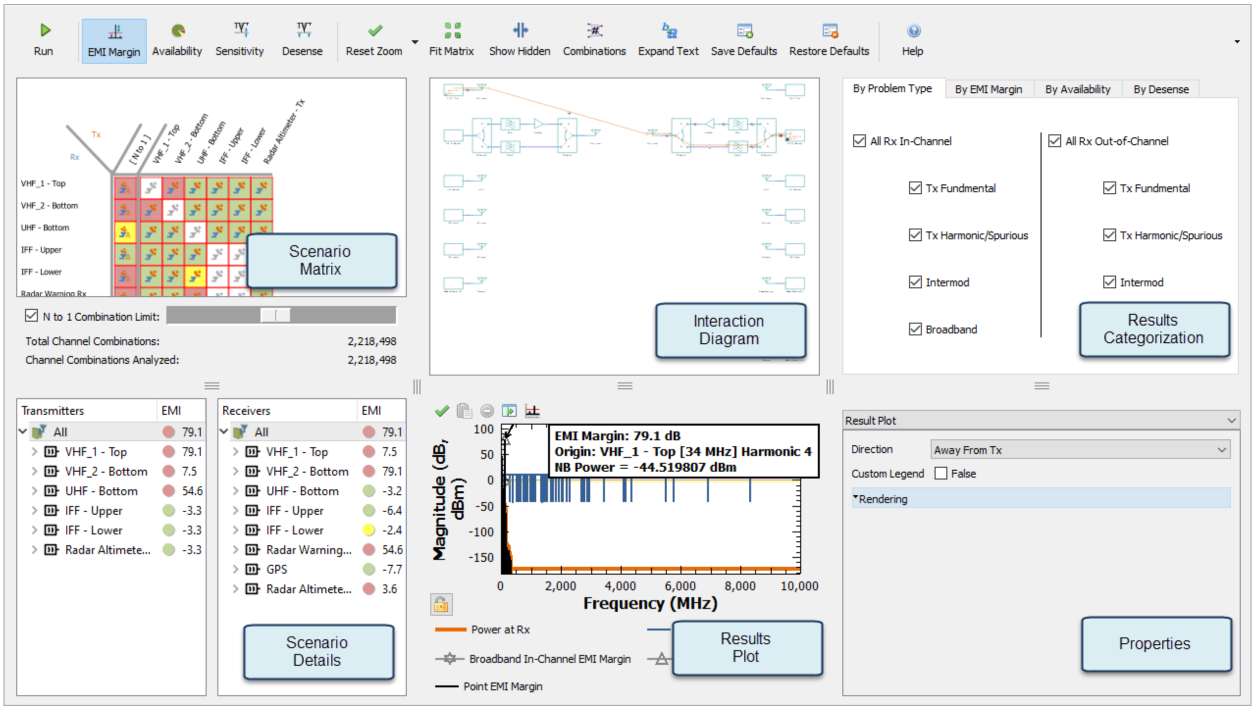


Analysis and Results

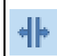
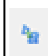
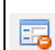
Integrated Results Window

Integrated Results Window

The EMIT Integrated Results window provides a single interface for running EMIT simulations and visualizing and exploring the results. The window is completely interactive and the displayed results are automatically updated as changes are made within the window's different sections.



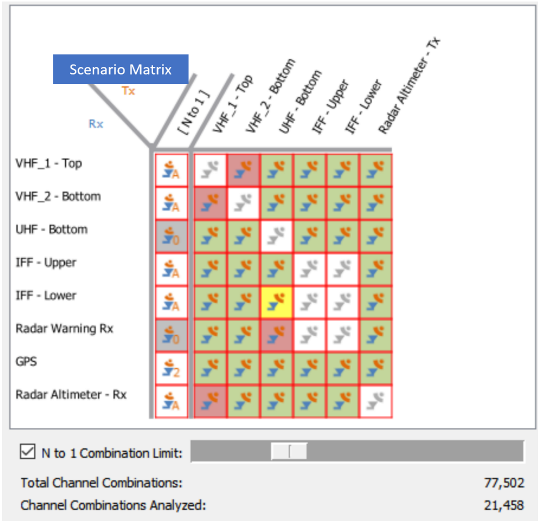
Run		Run the simulation for all enabled interactions.
Purge Results		Purge (delete) all simulation results (this cannot be undone).
EMI Margin		Select EMI Margin results mode. Displays EMI margins in all result views.
Availability		Select Availability results mode. Displays Availability in Scenario Details. Other results views display EMI Margin.
Sensitivity		Select Sensitivity results mode. Displays sensitivity in all result views. Scenario Matrix colors set by Desense thresholds.
Desense		Select Desense results mode. Displays Desense margins in all result views.

Reset		Reset Matrix View, Interaction Diagram zoom, or Plot Axes.
Fit Matrix to Panel		Resizes the Matrix within the Scenario Matrix.
Show Hidden Rows and Columns in Matrix		Toggle on/off display of all hidden rows and columns. When disabled, any rows/columns that were hidden are shown faded in the matrix.
Channel Combinations		Show/hide the combos.
Expand Text		Show untruncated component names in Interaction Diagram.
Save Defaults		Saves certain current settings as a user default for new designs. This includes the result type and all the values from the By EMI Margin, By Availability, and By Desense tabs of the Categorization panel.
Restore Defaults		Restores the original factory default values altered by Save Defaults for the user. Applies only to new designs, not the current one.

Scenario Matrix

EMIT's Scenario Matrix provides a compact way for viewing and manipulating scenarios in an EMIT project. The Scenario Matrix presents the complete scene in a matrix format with all Tx radios listed horizontally and all Rx radios listed vertically. The order of the rows and columns is determined by the position of the radios in the EMIT schematic. The leftmost column represents N-to-1 situations that consider interference to the Rx's due to the simultaneous operation of multiple Tx's. Grayed out entries in the Scenario Matrix represent Tx/Rx interactions that have been disabled and are not included in the EMIT simulation. Interactions can be enabled and disabled by using Ctrl+click on the desired square in the matrix.

Right-clicking on an item in the **Scenario Matrix** displays that item's action menu:



Simulations are run from the Integration Results Window toolbar. After the simulation has been run, the entries in the Scenario Matrix are color-coded to allow for rapid visual identification of problematic system interactions. The colors and associated threshold levels are defined in Result Categorization. Partially colored squares indicate the simulation has not been completed for that interaction square. Hovering the cursor over matrix entries displays a pop-up that shows the number of channel combinations that are included in the analysis and, if the square has been run, the maximum EMI margin, sensitivity or desense value (depending on the active result mode) for the associated Tx/Rx combination.

You can configure the number of N-to-1 simulations to include below the Scenario Matrix graphical display area. While EMIT can analyze the effects on any Rx when all active Tx's are operating simultaneously, it is often advantageous for both reducing simulation time as well as yielding insight into the interference mechanisms to analyze only a subset of active Tx's against a given Rx. By adjusting the slider **N to 1 Combination Limit**, you can control the limit to the number of multi-interferer combinations simulated for each receiver band. As the slider is adjusted, the **[N to 1]** column in the Scenario Matrix will automatically adjust. In the above Scenario Matrix, the UHF - Bottom and Radar Warning Rx receivers will not have any multi-interferer combinations analyzed as shown by the grayed out column entry with a "0" overlaid. The GPS receiver shows a "2" in the **[N to 1]** column and thus all 2-to-1 combinations will be analyzed and the remaining receivers all have an "A" in the **[N to 1]** column, thus all multi-interferer combinations possible will be analyzed. Deselecting the **N to 1 Combination Limit** will simulate all multi-interferer combinations in the scenario.

You can view the project statistics for **Total Channel Combinations** and **Channel Combinations Analyzed** under the Scenario Matrix. The statistics summarize the total number of channel combinations to be included in the simulation (which grows rapidly when modeling N-to-1 interactions) and the number of channel combinations that have been analyzed thus far.

Total Channel Combinations:	77,502
Channel Combinations Analyzed:	21,458

Note:

It may seem counter-intuitive, but sometimes higher order **N** requires fewer channel combinations than lower order **N**. Consider a simple case of 1 receiver against 5 transmitters (Tx_A, Tx_B, Tx_C, Tx_D, and Tx_E) where each transmitter has a single channel. Since each transmitter only has 1 channel, setting the **N to 1 Combination Limit = 1** would allow the *5 to 1* case to be analyzed:

$$Total\ Combos = Tx_{A_{combos}} * Tx_{B_{combos}} * Tx_{C_{combos}} * Tx_{D_{combos}} * Tx_{E_{combos}}$$

$$Total\ Combos = 1 * 1 * 1 * 1 * 1 = 1$$

By comparison, simulating all of the 3 to 1 cases would require significantly more combos:

$$Total\ Combos = \begin{array}{l} Tx_{A_{combos}} * Tx_{B_{combos}} * Tx_{C_{combos}} + \\ Tx_{A_{combos}} * Tx_{B_{combos}} * Tx_{D_{combos}} + \\ Tx_{A_{combos}} * Tx_{B_{combos}} * Tx_{E_{combos}} + \\ Tx_{A_{combos}} * Tx_{C_{combos}} * Tx_{D_{combos}} + \\ Tx_{A_{combos}} * Tx_{C_{combos}} * Tx_{E_{combos}} + \\ Tx_{A_{combos}} * Tx_{D_{combos}} * Tx_{E_{combos}} + \\ Tx_{B_{combos}} * Tx_{C_{combos}} * Tx_{D_{combos}} + \\ Tx_{B_{combos}} * Tx_{D_{combos}} * Tx_{E_{combos}} + \\ Tx_{C_{combos}} * Tx_{D_{combos}} * Tx_{E_{combos}} \end{array}$$

$$Total\ Combos = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 9$$

In this case, EMIT automatically simulates the *5 to 1* case since it includes all non-linear combinations that would be analyzed by the *3 to 1* cases and a worst-case analysis of the total noise at the input of the receiver.

Right-clicking in various locations within the **Scenario Matrix** provides options for settings within the Scenario Matrix. The following right-click options are available:

Right-Click on Matrix Square:

- **Run:** Run the EMIT simulation for the selected square.
- **Enabled:** Toggle the selected square between enabled/disabled.

Right-Click on Row:

- **Enable:** Enable all items in the selected row to be included in the simulation.
- **Disable:** Disable all items in the selected row from being included in the simulation.
- **Hidden:** Hide all items in the selected row.
- **Run:** Run EMIT simulation for all enabled items in the selected row.

Right-Click on Column:

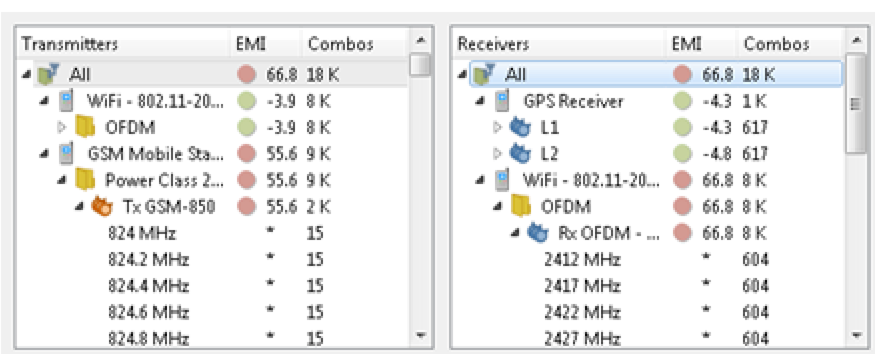
- **Enable:** Enable all items in the selected column to be included in the simulation.
- **Disable:** Disable all items in the selected columns from being included in the simulation.
- **Hidden:** Hide all items in the selected column.
- **Run:** Run EMIT simulation for all enabled items in the selected column.

Right-Click on Tx/Rx Corner:

- **Unhide All:** Show all rows and columns that were previously hidden.

Scenario Details

EMIT's Scenario Details provides a powerful way to view and navigate the results of an EMIT simulation using an intuitive hierarchical tree structure. A key feature in EMIT is the link between all of the results views. These results views include the Scenario Matrix, Interaction Diagram, Scenario Details, Result Plot and Result Categorization. Selections and/or settings made in any of these windows are automatically reflected in all of the others. For example, selecting a Tx and Rx pair in the Scenario Details will show the corresponding results in all of the other result windows as well. These linked views provide an extremely powerful diagnostic approach for the rapid identification of the origin of interference.



> **Note:**
>
> The Scenario Details displays only 1-to-1 simulation results and not N-to-1 results. If an N-to-1 case is selected in the Scenario Matrix, all results views will reflect that selection except the Scenario Details which will display no results and an appropriate message.

The Scenario Details window contains two trees. The one on the left is the Tx tree and the one on the right the Rx tree. At the top level (shown below) the Scenario Details shows all of the Tx and Rx radios as nodes in the trees. Expanding the radio nodes provides access to results at the band and channel levels.

Interaction Diagram

EMIT's Interaction Diagram provides a powerful way to visualize the signal paths causing interference in complex cosite scenarios. A key feature in EMIT is the link between all of the results views. These results views include the Scenario Matrix, Interaction Diagram, Scenario Details, Result Plot and Result Categorization. Selections and/or settings made in any of these windows are automatically reflected in all of the others. These linked views provide an extremely powerful diagnostic approach for the rapid identification of the origin of interference.

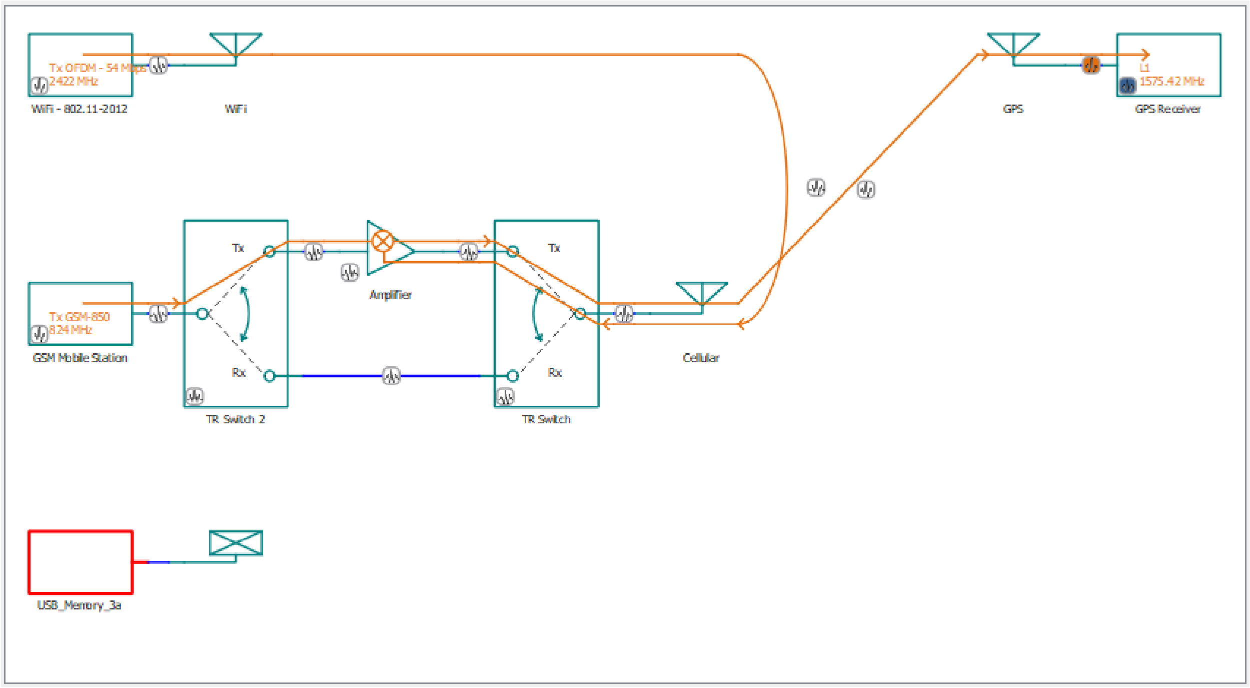
The Interaction Diagram shows a block diagram view of the entire (below) or selected parts of the scenario. Selected systems are shown in the Interaction Diagram correspond to those selected in either the Scenario Matrix or Scenario Details. All of the Tx's are in a column on the left-hand side of the Interaction Diagram and all of the Rx's on the right.

In order to display the signal paths in a palatable fashion, the arrangement of components in the RF systems may be slightly different than shown in the EMIT schematic editor, although all connections of components will remain as defined.

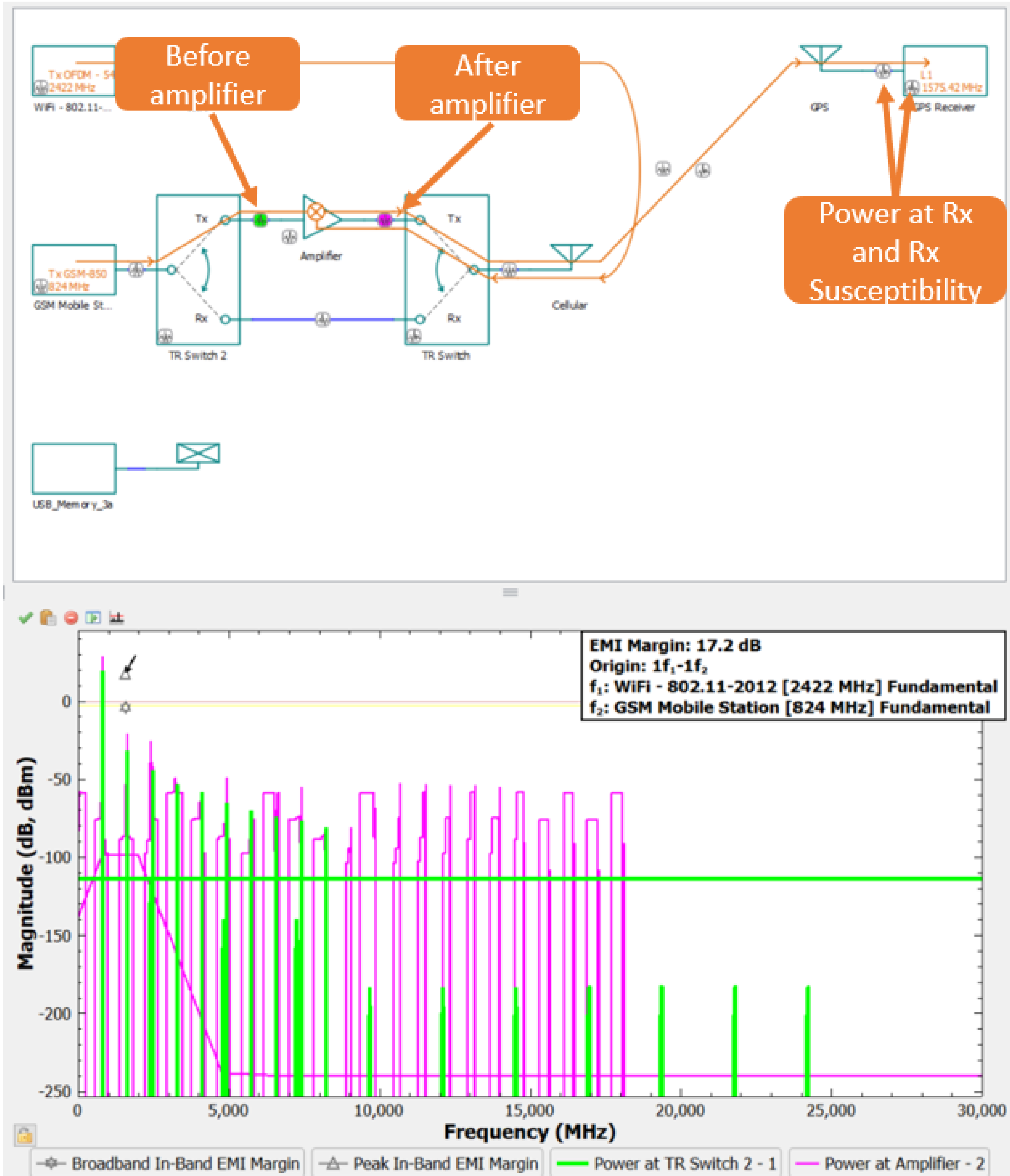
Selecting a component in the interaction diagram causes its values to be shown in the property panel.

There are button toggles available around the interaction diagram. These turn on and off corresponding traces in the result plot. The following types of traces are available:

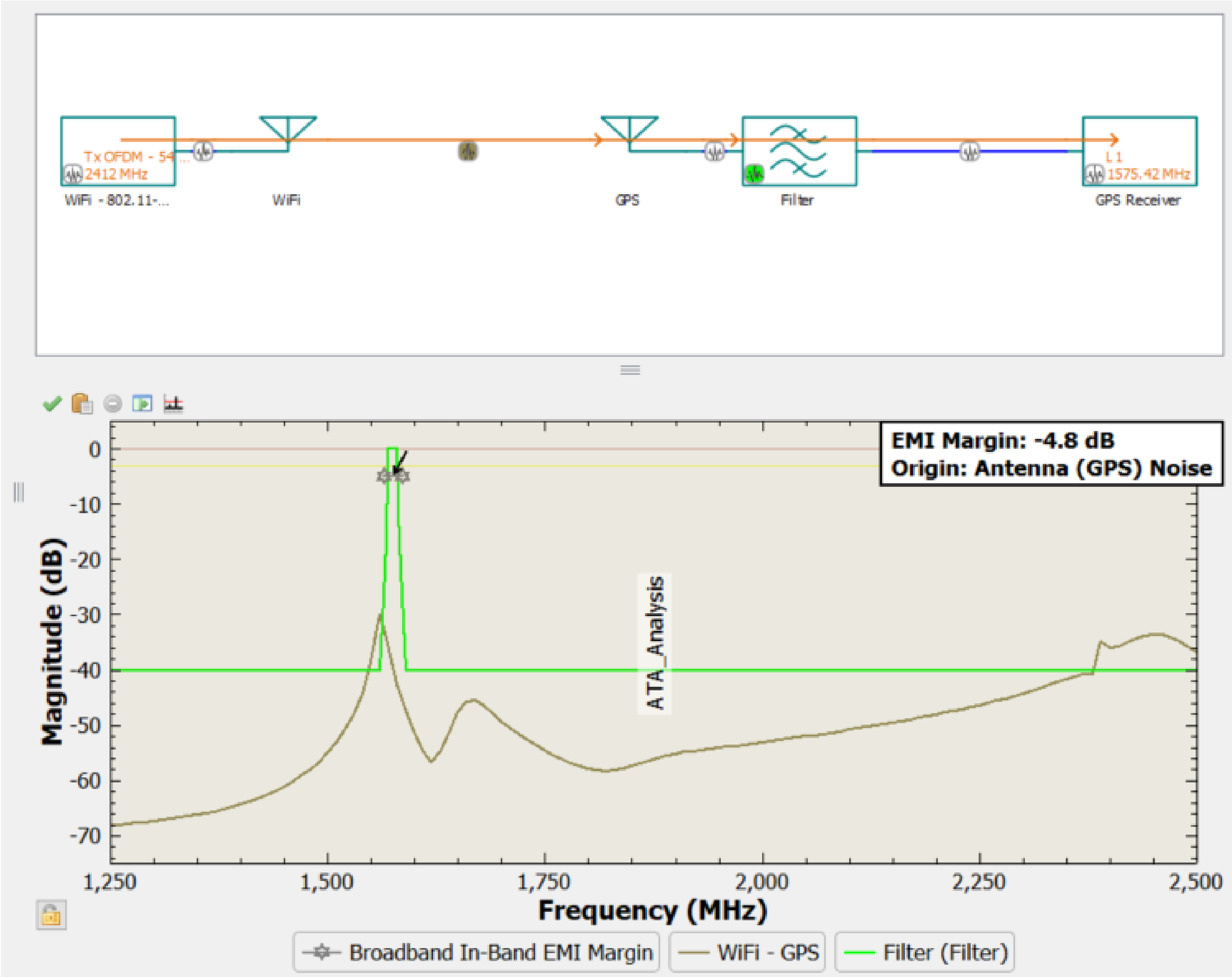
- outboard component
- traceback spectrum
- coupling
- tx power
- rx susceptability



These power spectrum toggles allow users to quickly control the traces shown in the Result Plot, thereby greatly simplifying the analysis. In the figure below, the Power at Rx and Receiver Susceptibility traces were disabled via the Interaction Diagram toggles while the power spectrum at the input and output of the amplifier were enabled. This highlights how most of the narrowband components at the input to the receiver were generated by intermodulation due to the coupling of the WiFi transmitter's spectrum to the GSM radio's external amplifier.

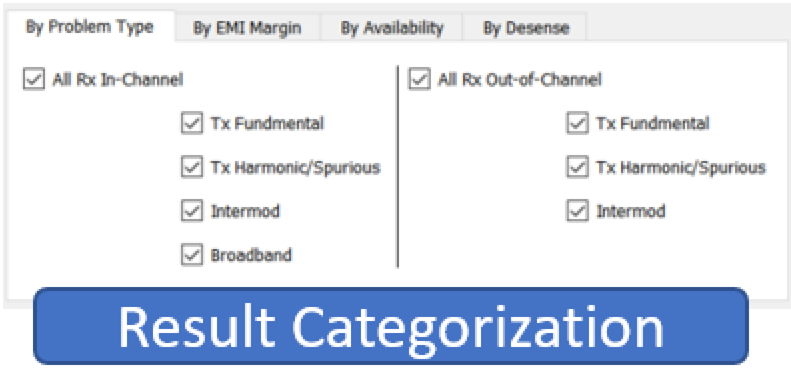


The Interaction Diagram toggles also enable the ability to view the coupling along the interference path and the characteristics of each component in the **Results Dialog**. In the figure below, you can see the band pass filter for the GPS Receiver plotted alongside the coupling between the WiFi and GPS antennas.



Result Categorization

EMIT's Result Categorization tabs provide a way to filter the display of results by the category of interference and to define interference threshold levels and associated colors via settings on the tabbed views in the window as shown below.

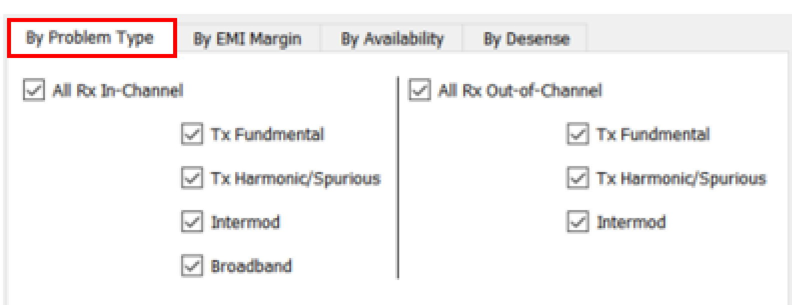


A key feature in EMIT is the link between all of the results views. These results views include the Scenario Matrix, Interaction Diagram, Scenario Details, Result Plot and Result Categorization. Selections and/or settings made in any of these windows are automatically reflected when appropriate in all of the others. For example, changing a setting in the Result Categorization will make the corresponding change in all of the other result windows as well. These linked views provide an extremely powerful diagnostic approach for the rapid identification of the origin of interference. Result Categorization Actions: Changes to settings in the Result Categorization window result in changes to all affected results views. There are four ways to categorize the results:

- By Problem Type
- By EMI Margin
- By Availability
- By Desense

These settings are applied to the results views as changes are made (that is, this results filtering is a post-processing step).

By Problem Type

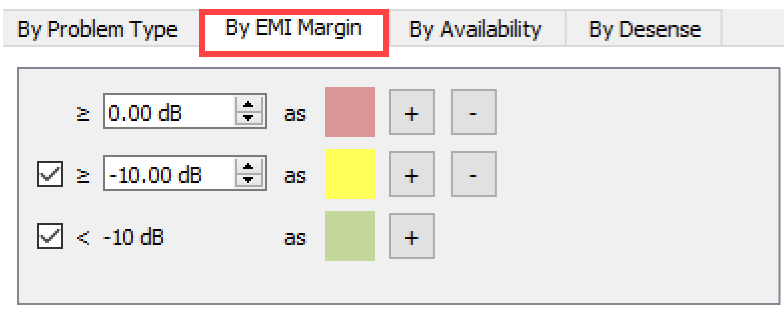


From the Rx perspective, there are two main categories of interference:

- **In-Channel** - Interference results from interfering signals that are within the 3-dB bandwidth of the Rx's tuned channel
- **Out-of-Channel** - Interference is due to interfering signals outside of the 3-dB bandwidth of the Rx's tuned channel

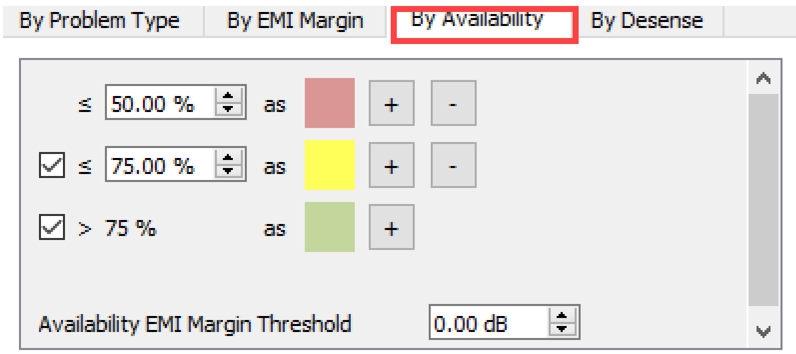
EMIT computes both In-Channel and Out-of-Channel interference for each Rx channel. The interference signals at a Rx can originate as a Tx fundamental, harmonic, spurious emission, broadband power or be due to an intermodulation product being generated by an amplifier. When computing the EMI margin at each Rx, EMIT keeps track of the origin of all interfering signals. This allows the results to be filtered based on the type of interference (In-Channel or Out-of-Channel) and/or the origin of the interference (Tx Fundamental, Tx Harmonic/Spurious, Intermod and/or Broadband). The filtering to be performed (and reflected in all results views) is controlled by the boxes that are enabled/disabled in the By Problem Type in the Result Categorization window. When a category is disabled, the results associated with that category are not displayed in any of the results views. The other results views are automatically updated as selections are made in the Result Category window and a 'note' is added to these views when a results filter is applied. (Note that EMIT does not currently compute out-of-channel interference caused by broadband power).

By EMI Margin



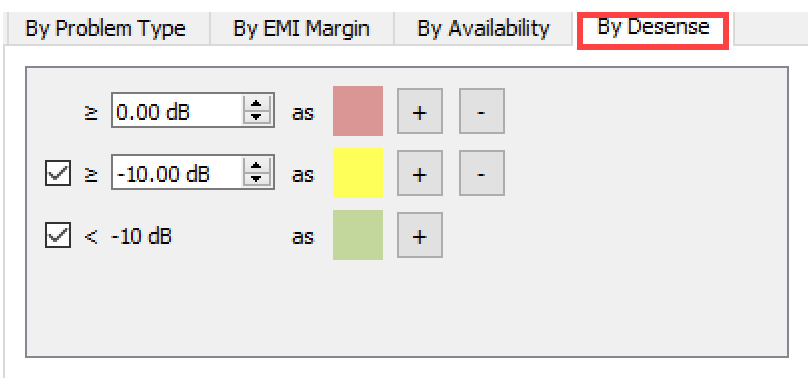
Both the Scenario Matrix and Scenario Details windows color results based on EMI margin thresholds that are defined by the user in the By EMI Margin tab of the Result Categorization window. Threshold levels are set by entering the threshold value and choosing an associated color (by clicking on the colored square). Threshold levels can be added or removed using the "+" and "-" buttons beside each one. Threshold levels will be ignored if the check box to the left of each level definition is unchecked. Horizontal lines at each threshold level are drawn using the specified color in the Result Plot as well. An example Scenario Matrix showing multiple threshold levels is shown in the figure below.

By Availability



Only the Scenario Details windows color results based on availability thresholds that are defined by the user in the By Availability tab of the Result Categorization window. Other results views will display the EMI Margin. Threshold levels are set by entering the threshold value and choosing an associated color (by clicking on the colored square). Threshold levels can be added or removed using the "+" and "-" buttons beside each one. Threshold levels will be ignored if the check box to the left of each level definition is unchecked. When calculating availability, EMIT counts each EMI Margin that exceeds the value defined n **Availability EMI Margin Threshold**as a problem that counts against availability. That is, if this threshold value is exceeded, that channel is deemed unavailable.

By Desense



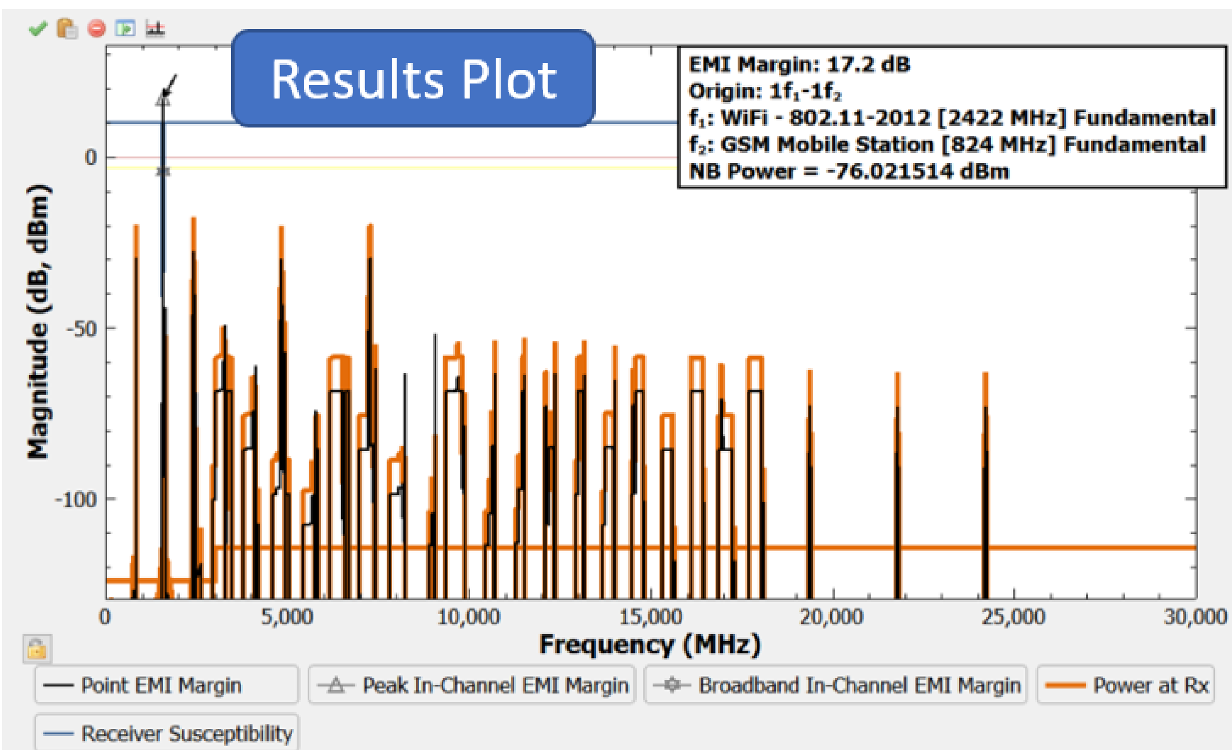
Both the Scenario Matrix and Scenario Details windows color results based on desense thresholds you define. Threshold levels are set by entering the threshold value and choosing an associated color (by clicking on the colored square). Threshold levels can be added or removed using the "+" and "-" buttons beside each one. Threshold levels are ignored if the check box to the left of each level definition is unchecked. Horizontal lines at each threshold level are drawn using the specified color in the Result Plot as well.

Note:

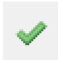
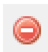
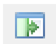
Specifying results thresholds/colors for results by Sensitivity is not available. This is because sensitivity is a unique Rx characteristic, and so it is not useful to categorize all Rx's by the same sensitivity threshold values. For this reason, when the sensitivity is the selected result mode, the coloring of the Scenario Matrix will be done based on the thresholds defined for Desense.

Results Plot

EMIT's Results Plot provides a powerful approach to visualizing the wideband interference for individual Tx/Rx channel combinations in complex cosite scenarios. The Results Plot also provides powerful automated diagnostics to identify the specific cause of multiple interference contributors. A key feature in EMIT is the link between all of the results views. These results views include the Scenario Matrix, Interaction Diagram, Scenario Details, Results Plot and Result Categorization. Selections and/or settings made in any of these windows are automatically reflected in all of the others. These linked views provide an extremely powerful diagnostic approach for the rapid identification of the origin of interference.



The toolbar in the Results Plot window provides the following actions and are active depending on the current selection in the plot window:

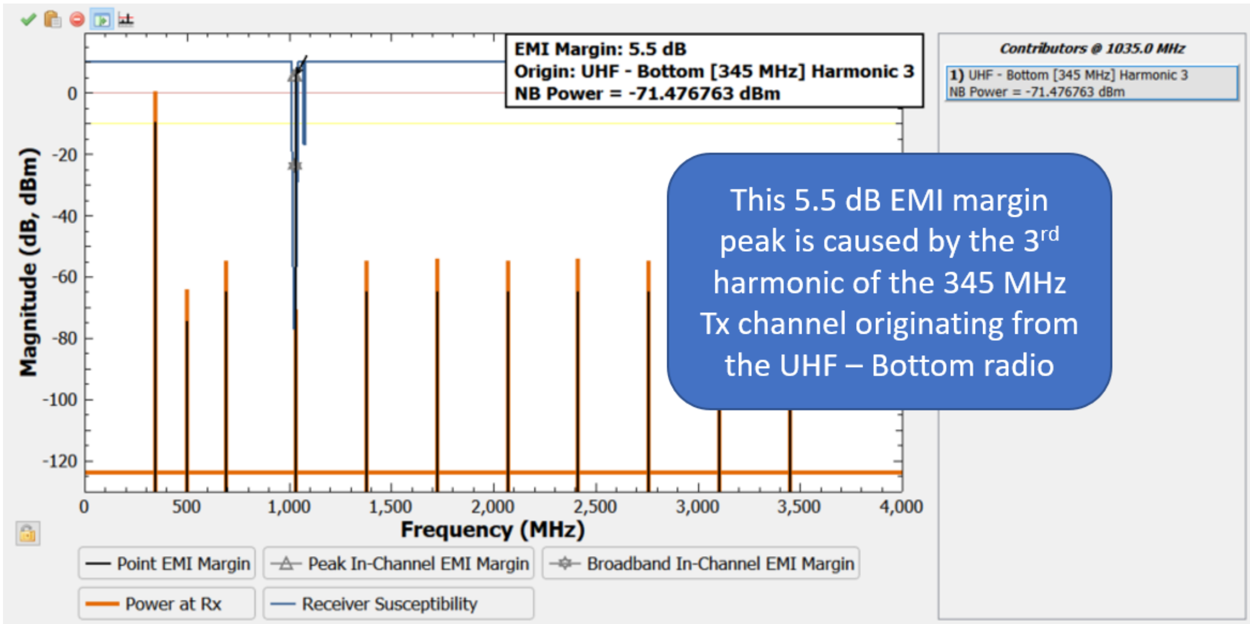
	Reset the plot axes to the default values.
	Active when a plot trace is selected, copies the trace data to the clipboard for pasting into other applications.
	Delete the selected trace from the plot. Note that the default Result Plot Traces cannot be deleted (but they can be hidden).
	Show the Contributors Panel in the Results Plot window.
	Show the plot properties in the Properties area of the Integrated Results window.

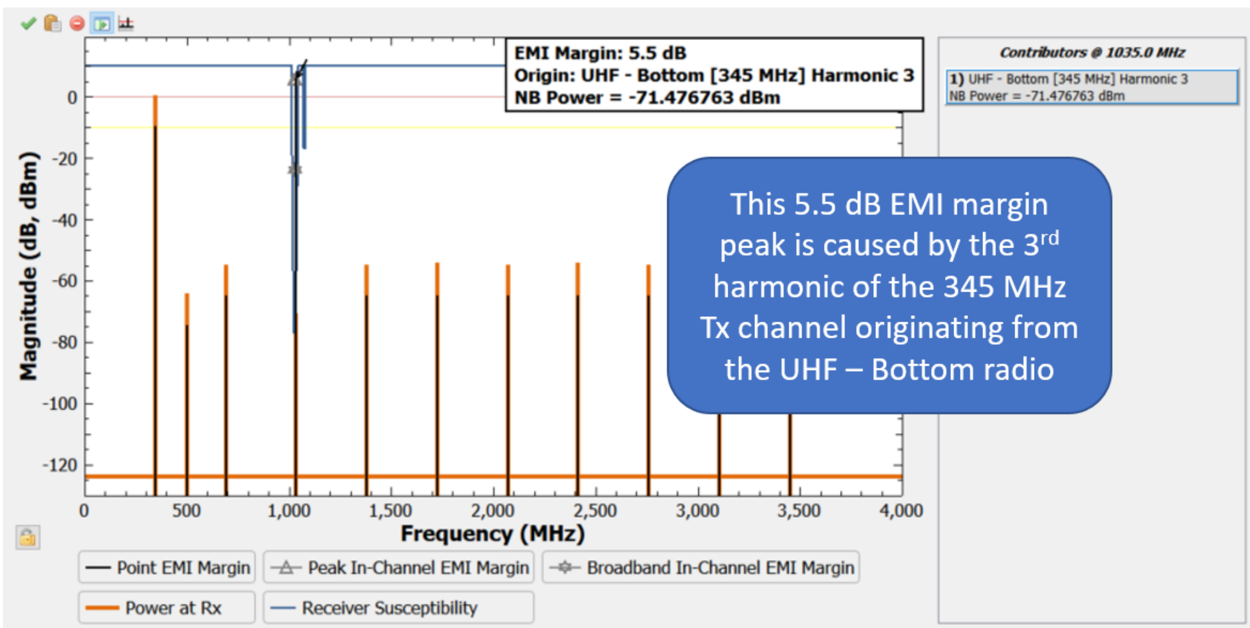
The Results Plot shows the data associated with the worst-case Tx/Rx channel combination for the Tx/Rx pair selected in either the Scenario Matrix or Scenario Details. By default, each Result Plot shows the following plot traces:

- Receiver Susceptibility for the Rx channel
- Interference power at the Rx
- Point In-Channel EMI Margin (When EMI Margin or Availability result mode is selected)
- Broadband In-Channel EMI Margin (When EMI Margin or Availability result mode is selected)
- Peak In-Channel EMI Margin (When EMI Margin or Availability result mode is selected)
- Sensitivity (When Sensitivity result mode is selected)
- Desense (When Desense result mode is selected)

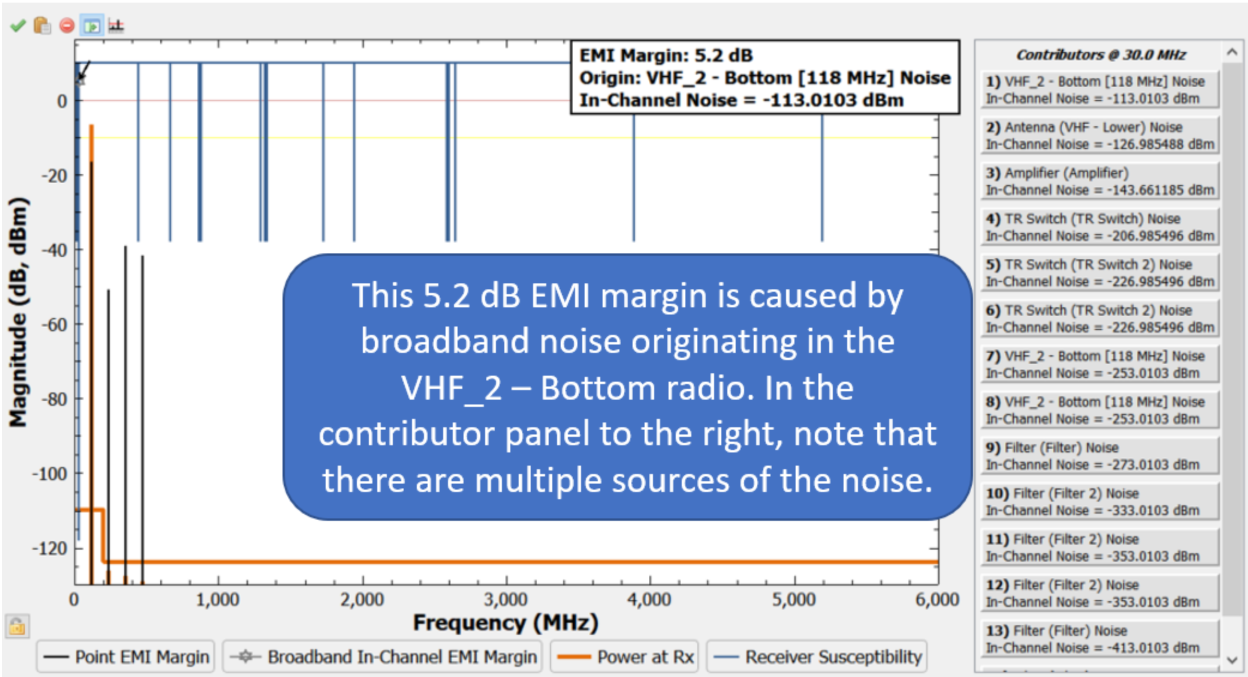
In addition to plotting the above data, the Result Plot also has a Result Marker that identifies the root-cause of specific interference. There are two elements to this Result Marker. The first is an arrow in the plot area that points to the desired interference feature. By default, this arrow points to the largest interference problem, but it can be positioned to point at any peak in the interference results data by dragging it with the mouse. The second element of Result Marker is the text box appearing in the plot area. This box provides diagnostics that report the cause of the interference identified by the arrow pointer. The information in this box changes when the location of the arrow changes to reflect the currently highlighted feature. The location of the text box in the plot area can be changed by dragging it with the mouse to the desired position.

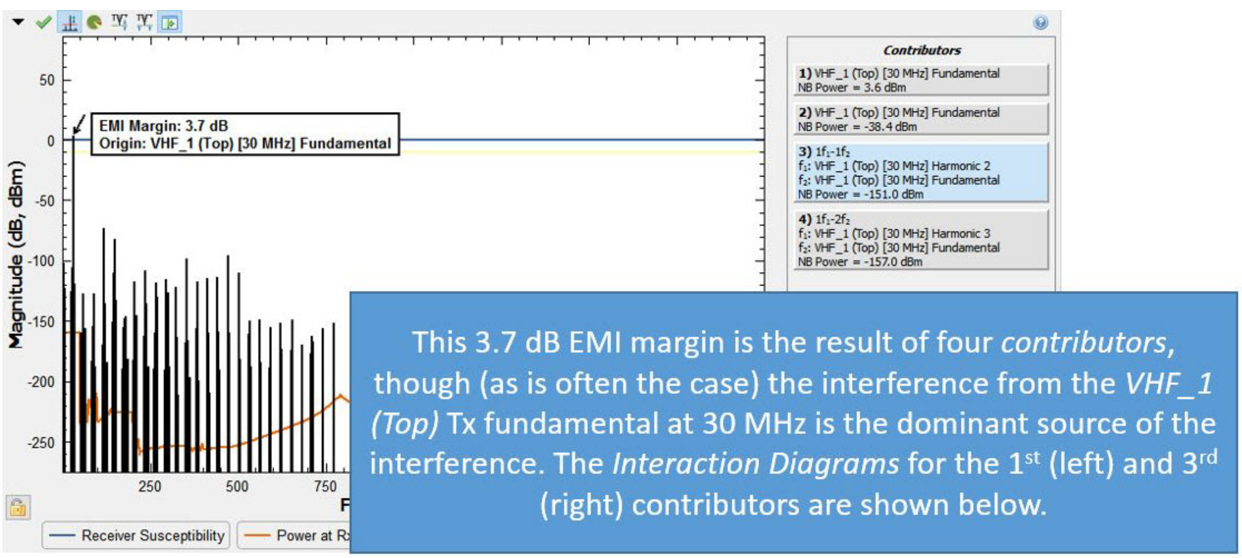
The information provided in the text box identifies the interference value and the origin of the interference. The origin of interference is identified by the Tx radio names, the offending Tx channels and the component (fundamental, harmonic, spur or noise) of the Tx contributing to the highlighted EMI margin. This is best demonstrated via an example, and below are several figures demonstrating this powerful feature.

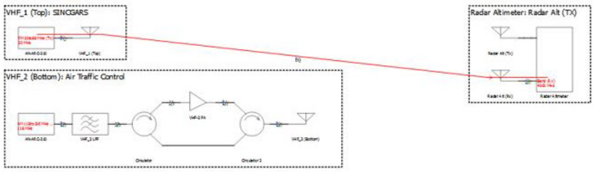


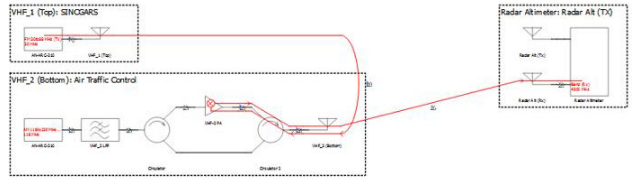


The Results Plot also includes a **Contributors** list on the right side. This panel can be hidden by selecting the icon at the top of the Results Plot. As shown in the example below, the Contributors list specifies the spectral components that "contribute" to the interference selected by the marker. The contributors are ordered by amplitude, with the highest-amplitude contributor at the top of the list. If there are more than 30 contributors at the selected frequency, only the top 30 are displayed. In the below example, the interference is dominated by the VHF_2 (Bottom) radio's broadband noise level, but the noise generated by the external amplifier also contributes to the Noise EMI margin. Each contributor listed can be selected (using the up/down arrows or via left-click), which will update the Interaction Diagram so that it shows the path of the current contributor. The currently selected contributor is highlighted in blue.

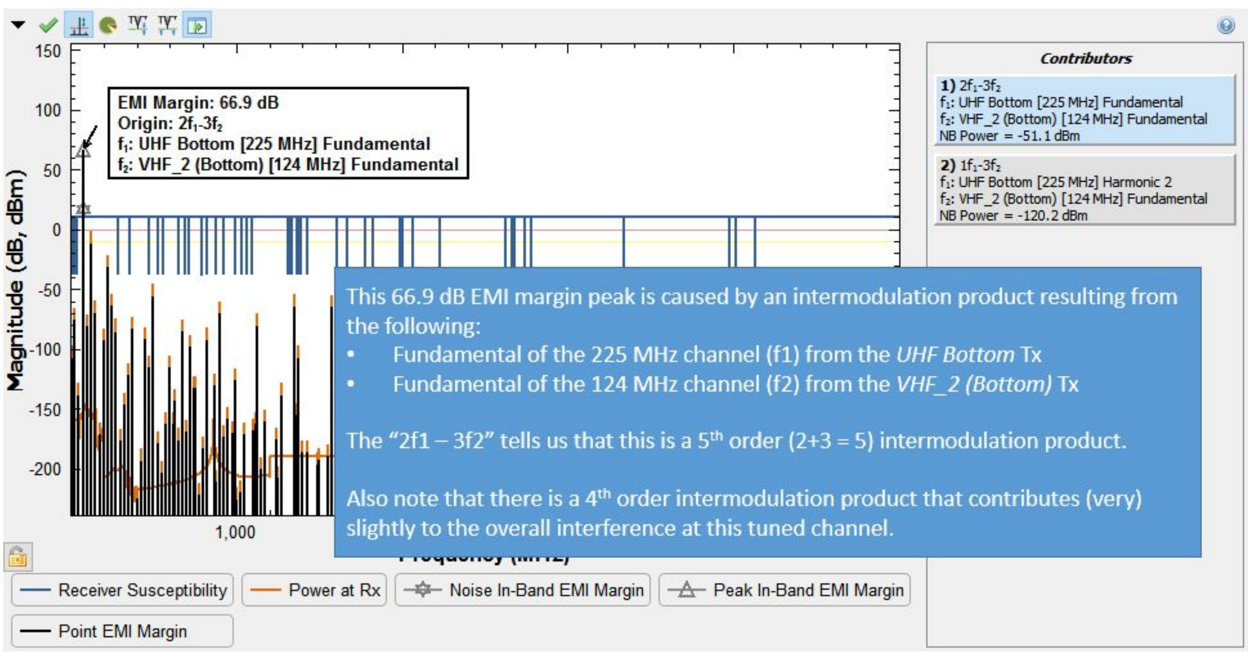






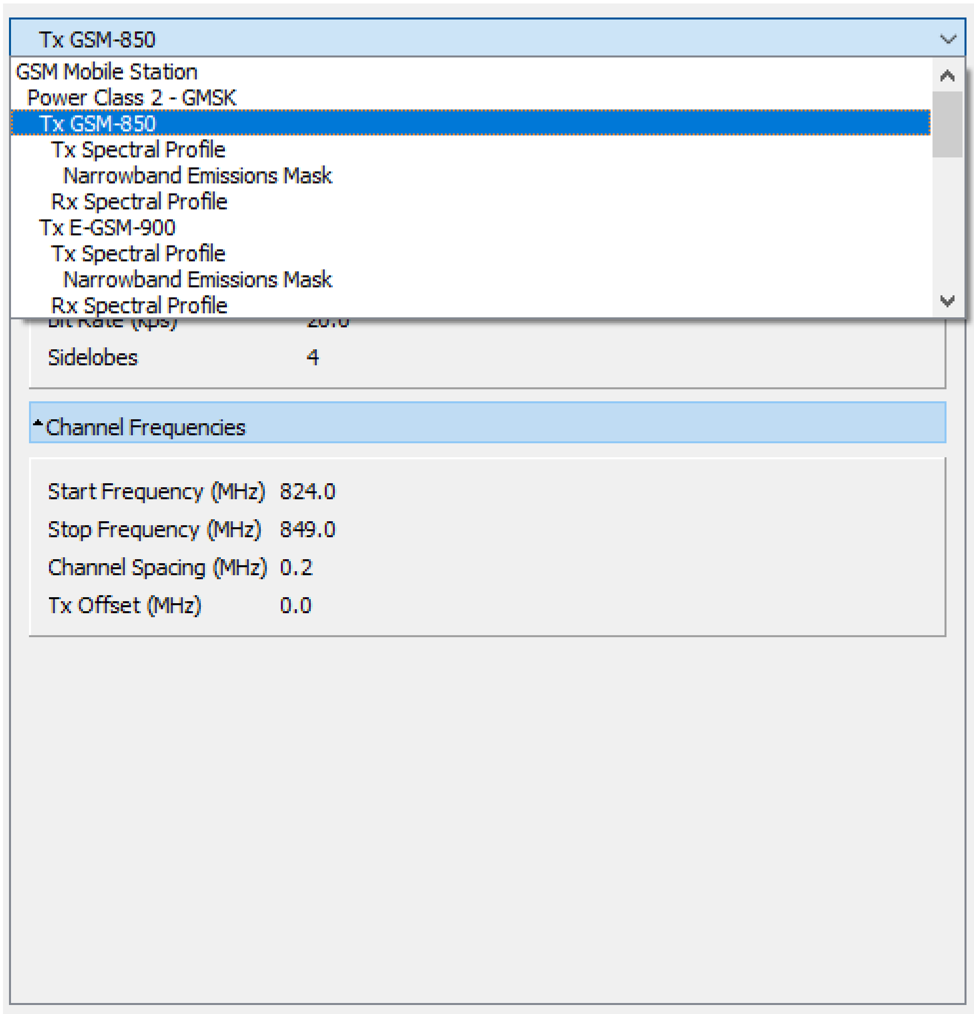


For interference caused by intermodulation products, the Result Marker provides all the information necessary to completely identify the order and origin of the intermodulation product as demonstrated in the example shown below.



Results Property Panel

This panel shows various plot and component properties. Plot properties are editable, but component properties are read-only and are just there for reference. If a component is selected in the interaction diagram, the combo box at the top allows access to groups of properties associated with that component. Otherwise, it allows you to select access to properties for the plot itself or various traces currently shown in the plot.



Plot Properties

Plot Trace Properties

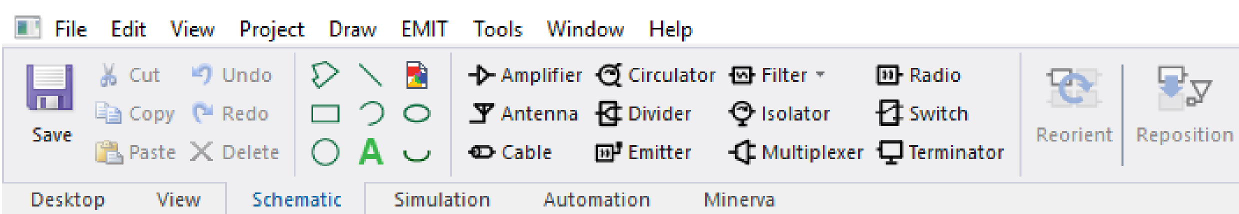
7 - EMIT Schematic Editor

Use the schematic editor to create and edit EMIT designs. You create a design by starting the schematic editor and placing EMIT components into a default empty schematic. You can also use primitive drawing elements.

When you edit an EMIT design, the schematic editor automatically enforces EMIT-specific connection rules to ensure that a valid EMIT RF system is created. The EMIT solver is based on a power spectral flow methodology and so some connections that may be permissible when wiring a nodal-based circuit design are not valid for an EMIT design. For example, any circuit node can only connect two wires or ports together. More than two connections are not permitted at a circuit node in EMIT systems.

EMIT components are placed into the schematic either via the EMIT schematic ribbon, or the EMIT component library. Both offer the same selection of components to choose from.

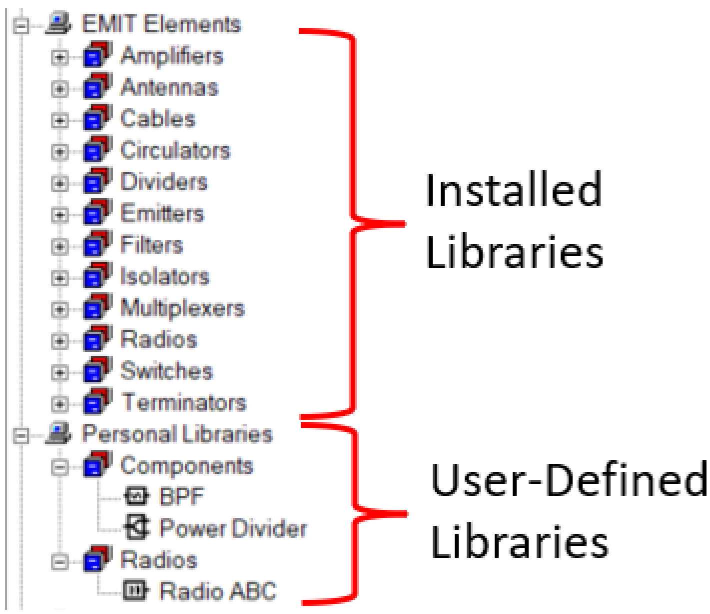
Schematic Ribbon: To place components in the schematic via the schematic ribbon, simply select the desired component from the ribbon and the component will be placed in the schematic. Repeated selections add additional components and place them via the auto-placement feature (see below). Some components (e.g., Filters) offer a further selection of specific type when selected.



Component Library: To place components in the schematic via the components library, simply drag the desired component from the components library tree to the desired location in the schematic. Once placed in the schematic, the component can be placed using the auto-placement feature (see below) if desired.

After making changes to a component, the components can be exported to a user library. Right-click on any component and select **Export to Library** to save a component to the library. This makes the component easily accessible across multiple designs and/or projects. Once stored to a library, users can rename a component by right-clicking on it in the library tree and selecting **Rename Library Component**. Similarly, library components can be deleted by right-clicking on them in the library tree and selecting **Delete Library Component**.

> **Note:**
> External files (for example *.dlxsp and *.snp) that are referenced by components are automatically copied and stored in the *EmitExternalFiles* subdirectory within a user's *PersonalLib* directory. This prevents them from being accidentally deleted if the original component is deleted. However, if a library component is deleted, these copied files do NOT get deleted because multiple components can reference the same file.



Auto-Placement: The EMIT schematic editor has an auto-placement feature to assist in placing and connecting components in the EMIT design.

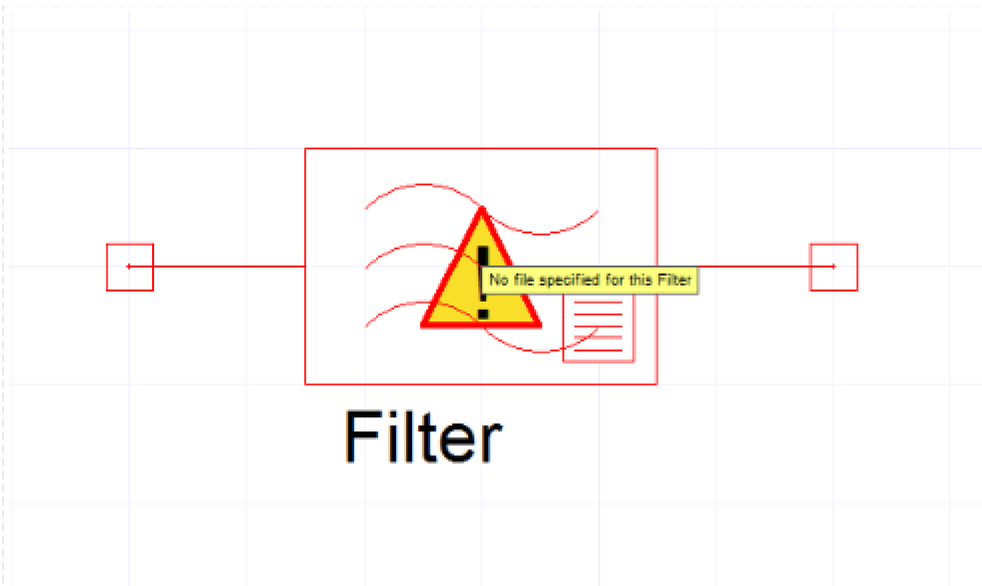
When components are added from the Schematic ribbon, the placement (and orientation for some components) is automatically determined. Components are automatically connected to the first valid port (starting from the top) that is visible in the working area of the schematic. Alternatively, if a particular component is selected in the schematic when a new component is added from the ribbon, the new component automatically connects to the first valid port of the selected component. If this is not the desired position, use **Reposition** from the Schematic ribbon to move the selected component to the next available port. In this way, components can be easily moved and connected to the desired port in the schematic.

The **Reorient** action from the Schematic ribbon cycles through all of the possible orientations of the selected component.

As noted, only the visible portion of the Schematic is used when determining the auto-placement, and so it is very useful to zoom in on the region of the schematic that you want to place the components.

By using the Schematic ribbon to place components along with the auto-placement, Reorient and Reposition features, complex schematics can be configured more rapidly than by using manual placement and wiring, although the manual approach is still available for use and can be helpful in many situations.

A warning symbol will show up on any components in the schematic editor that require further information to completely define a valid component. Hovering the mouse cursor shows a pop-up indicating what the problem is as shown below.



Primitive Drawing Elements

You can add primitive drawing elements such as lines, arcs, circles, rectangles, polygons, and text. You can also import images in different formats. You can use the included drawing elements or imported images to create symbols for schematic pages. These drawn elements do not affect the simulation, but can help to document or clarify your schematic.

Drawing a Polygon

To add a closed polygon to a schematic:

1. Click **Draw>Primitive>Polygon** or click the **Polygon** icon on the **Schematic** ribbon.

   This enables the polygon drawing mode.

2. Move the cursor to the desired location in the schematic and click to draw the first vertex.
3. Move the cursor to the desired location for each subsequent vertex and click.

   This draws a straight line between the designated vertices, and a straight line between the last vertex specified and the first.

4. Right-click and click **Finish** to complete the polygon.

   The polygon remains selected, with the drag handles visible at the vertices. If necessary, you can select a handle to drag any vertex.

   The polygon includes a **Properties** window in which you can edit the polygon’s properties, including border color, border width, fill style, fill color, and coordinates of the vertices.

5. Click anywhere on the schematic off the polygon to deselect it.

Drawing a Rectangle

To add a rectangle to a schematic:

1. Click **Draw>Primitive>Rectangle** or click the **Rectangle** icon on the **Schematic** ribbon.

   This enables the rectangle drawing mode.

2. Move the cursor to the desired location in the schematic and click to place the first corner of the rectangle.
3. Move the cursor to the desired location for the opposite corner of the rectangle and click.

   This shows the rectangle selected, with the drag handles on the corners visible. If necessary, you can select a handle to resize the rectangle.

   The rectangle includes a **Properties** window in which you can edit the rectangle’s properties, including border color, border width, fill style, fill color, center coordinates, width, height, and angle of rotation.

4. Click anywhere on the schematic off the rectangle to deselect it.

Drawing a Circle

To add a circle to a schematic.

1. Click **Draw>Primitive>Circle** or click the circle icon on the **Schematic** ribbon.

   This enables the circle drawing mode.

2. Move the cursor to the desired location in the schematic and click to specify the center point of the circle.
3. Move the cursor to the desired location for the diameter of the circle and click.

This dynamically draws the circle and shows the circle selected, with the drag handles on the end points. If necessary, you can select a handle on the circle to change the diameter. Notice that the element includes a Properties window.

The circle includes a **Properties** window in which you can edit the circle's properties, including fill color, border width, border color, fill style, center coordinates, and radius.

4. Click anywhere on the schematic off the circle to deselect it.

Drawing a Line

To add a line to a schematic:

1. Click **Draw>Primitive>Line** or click the **Line** icon on the **Schematic** ribbon.

   This enables the line drawing mode.

2. Move the cursor to the desired location in the schematic and click to draw the first end point of the line.
3. Move the cursor to the desired location for the second endpoint of the line and right-click and click Finish.

   This draws a straight line between the two points and shows the line selected, with the drag handles on the end points highlighted. If necessary, you can select a handle on the line to change the location of the endpoints.

   The line includes a **Properties** window in which you can edit the line's properties, including color, line width, line style, coordinates of the endpoints, and objects such as arrowheads to be placed at line ends.

4. Click anywhere on the schematic off the line to deselect it.

Drawing an Arc

To add an arc to a schematic:

1. Click **Draw>Primitive>Arc** or click the Arc icon on the **Schematic** ribbon.

   This enables the arc drawing mode.

2. Move the cursor to the desired location in the schematic and click to draw the first end point of the arc.
3. Move the cursor to the desired location for the second endpoint of the arc. and click.

   This draws an arc between the two points and gives the cursor a drag-handle at the mid-point of the arc.

4. Drag the mid-point to establish the arc, and click to release the mode.

This shows the arc selected, with the drag handles on the end points and midpoint of the arc highlighted. If necessary, you can select a handle on the arc to change the location of the endpoints or the arc.

The arc includes a **Properties** window in which you can edit the arc’s properties, including color (applies to both border and fill), line width, fill style, center coordinates, radius, and starting and ending angles.

5. Click anywhere on the schematic off the arc to deselect it.

Adding an Image

To add an image, such as a bitmap or PNG file, to a schematic:

1. Click **Draw>Primitive>Image** or click the image icon on the **Schematic** ribbon.

   This displays a file browser.

2. Browse to the location of the file you want to insert, and specify the file format, such as bitmap (.bmp) or PNG (.png).
3. Click the **Open** button to import the image to the schematic.
4. Position your cursor at the desired location and click.

   The image appears at the cursor location, remaining selected so that you can resize it as needed by dragging the cursor. Resizing maintains the original proportions.

5. Click again to anchor the image and release it from the cursor.

   The image remains selected, with the drag handles on the corners visible. If necessary, you can select a handle to resize the image.

   The image includes a **Properties** window in which you can edit the image’s properties, including: center coordinates, angle of rotation, width, and height. A check box toggles display of a border and enables additional border properties including: border color and width. Additional check boxes allow the image to be mirrored left-to-right, and to establish a link to the image file.

Adding Text to a Schematic

To add text to a schematic:

1. Click **Draw>Primitive>Text** or click the  icon on the toolbar.

   This enables text mode.

2. Move the cursor to the desired location in the schematic and click.

   This places a highlighted text box with “Default Text” written and highlighted within it.

3. Type in the text box to replace the default text.

   The text box includes a **Properties** window in which you can edit the text's properties, including: text color, location (coordinates), text angle of rotation, text size, font, and justification. A check box toggles display of a bounding rectangle. When displayed, additional rectangle properties can be displayed including: border color and width, fill color and style, the rectangle's center coordinates, width, height, and angle of rotation. Note that the rectangle center and angle can differ from the text center and angle.

4. Click anywhere on the schematic to deselect the text.

Drawing an Ellipse

To add an ellipse to a schematic:

1. Click **Draw>Primitive>Ellipse** or click the ellipse icon on the **Schematic** ribbon.

   This enables the ellipse drawing mode.

2. Move the cursor to the desired location in the schematic and click to specify the first end point of the ellipse.
3. Move the cursor to the desired location for the second endpoint of the ellipse, and click.

   This dynamically draws the ellipse with the drag handles on the end points and midpoint of the arc highlighted. If necessary, you can select a handle on the ellipse to change the diameter.

   The ellipse includes a **Properties** window in which you can edit the ellipse's properties, including fill color, border width, border color, fill style, center coordinates, width, height, and angle.

4. Click anywhere on the schematic off the ellipse to deselect it.

Drawing a Curve

To add a Bezier curve to a schematic:

1. Click **Draw>Primitive>Curve** or click the Curve icon on the **Schematic** ribbon.

   This enables the curve drawing mode.

2. Move the cursor to the desired location in the schematic and click to draw the first end point of the curve.
3. Move the cursor to the desired location for the second endpoint of the curve. and click.
4. Right-click to select **Place and Finish**.
5. Drag the mid-point to establish the curve, and click to release the mode.

This shows the curve selected, with the drag handles on the end points and midpoint of the curve highlighted. If necessary, you can select a handle on the curve to change the location of the endpoints or the curve.

The curve includes a **Properties** window in which you can edit the curve's properties, including color (applies to both border and fill), line width, line style, and vertexes.

6. Click anywhere on the schematic off the curve to deselect it.

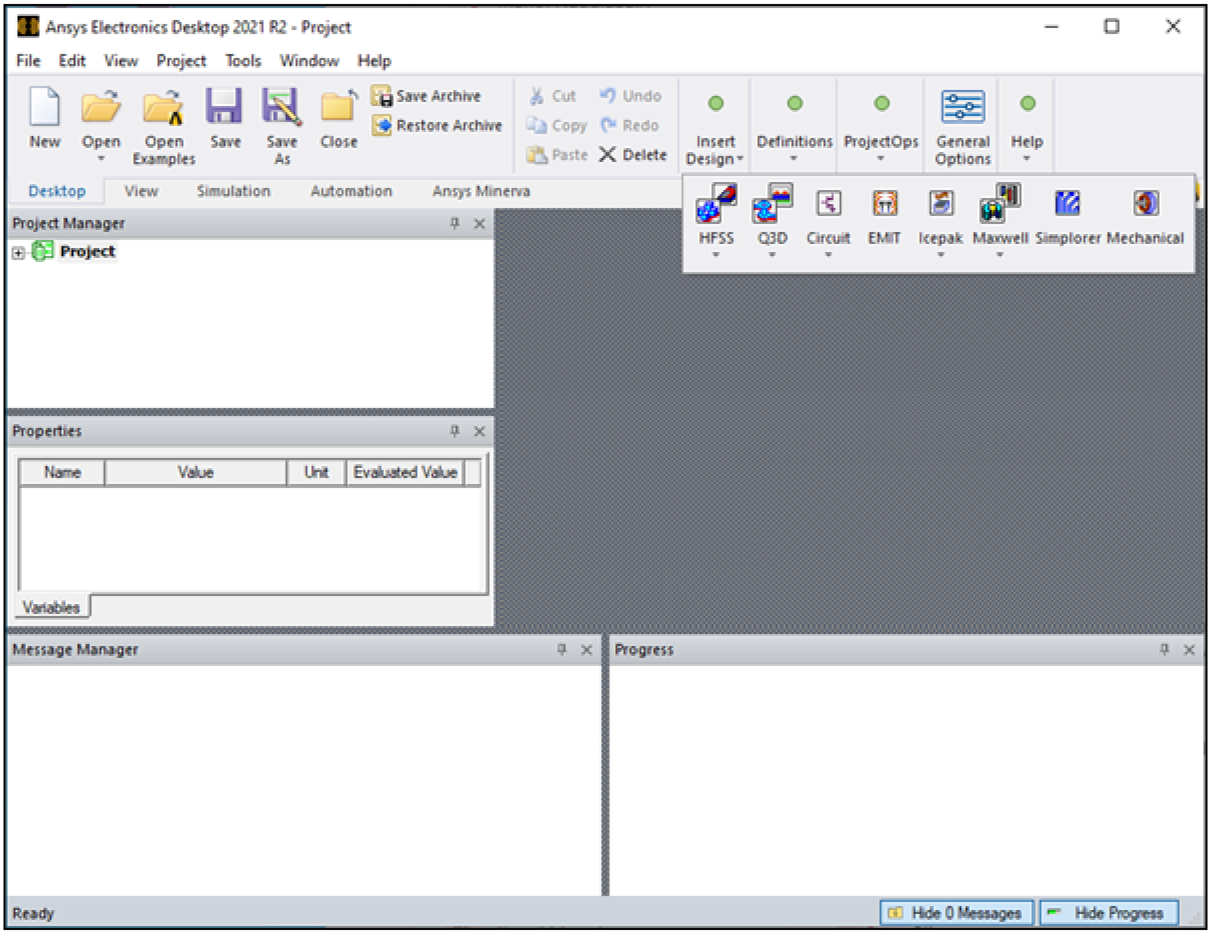
8 - EMIT in the Ansys Electronics Desktop Context

The following sections provide information about using EMIT in the context of Ansys Electronics Desktop:

- Getting Started with Ansys Electronics Desktop
- Working with Ansys Electronics Desktop Projects

Getting Started with Ansys Electronics Desktop

The Ansys Electronics Desktop, shown in the following figure, provides a comprehensive environment for designing and simulating various electronic components and devices. The Electronics Desktop consists of a unified user interface where electromagnetic designs, thermal designs, and circuits can be created. Typically, you can create or import a design, set up the simulation, validate your design, run the analysis, and post process the results.



The desktop includes the following design types:

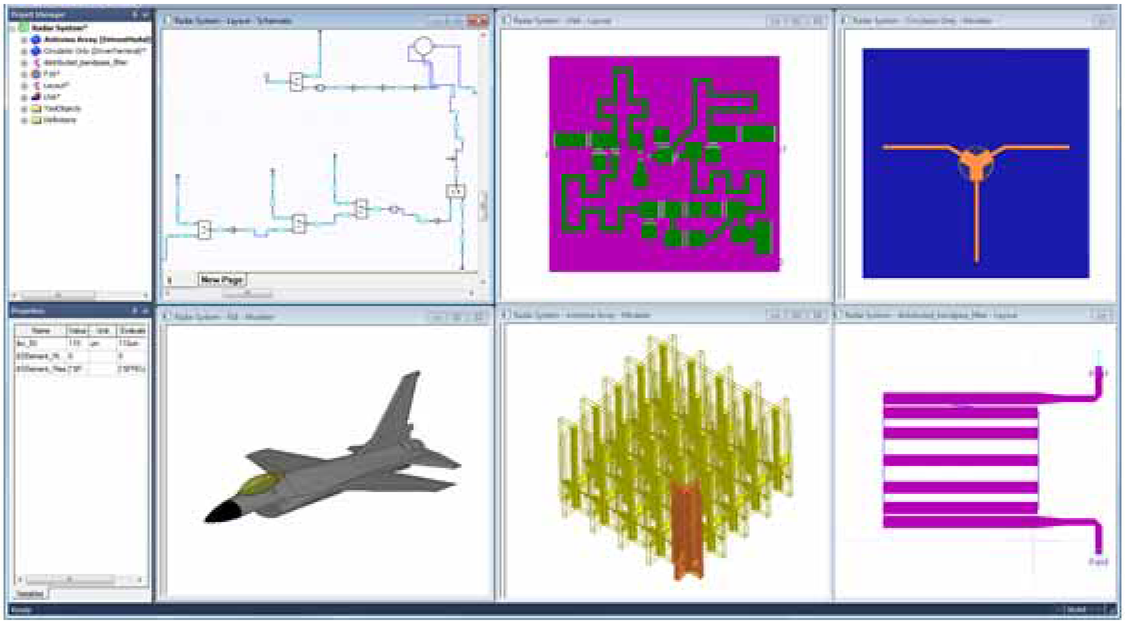
- **HFSS** – a general purpose 3D interface for the design, analysis, and simulation of electromagnetic components.
- **HFSS 3D Layout** – a full-wave, layout-based electromagnetic simulator with a specialized interface for geometries created in layout.
- **Q3D Extractor** – a quasi-static 3D solver for extracting lumped RLGC parameters and Spice models.
- **2D Extractor** – a 2D solver for extracting per-unit-length RLGC parameters of transmission lines.
- **Circuit** – a schematic-based interface to the Nexxim circuit simulator.
- **Circuit Netlist** – a netlist (text-based) interface to the Nexxim circuit simulator.
- **EMIT** – system simulation for predicting and mitigating radio frequency interference (RFI) in electronic devices.
- **Maxwell 3D** – uses finite element analysis (FEA) to solve three-dimensional (3D) electrostatic, magnetostatic, AC magnetic (eddy current), and transient problems.

- **Maxwell 2D** – uses finite element analysis (FEA) to solve two-dimensional (2D) electrostatic, magnetostatic, AC magnetic (eddy current), and transient problems.
- **RMxprt** – a template-based electrical machine design tool that provides fast, analytical calculations of machine performance and 2-D and 3-D geometry creation for detailed finite element calculations in Maxwell.
- **Maxwell Circuit** – sets up external circuit designs to supply excitations to coil terminals for Maxwell 2D and 3D AC Magnetic and Transient designs.
- **Simplorer** – an integrated, multi-domain, mixed-signal simulator for complex technical systems. Simplorer is a subset of the Twin Builder standalone product. Please see the Twin Builder help for more information.
- **Icepak** – a fluid and thermodynamic simulator for electronic systems and components.
- **Mechanical** – perform modal analyses to determine natural vibration frequencies and thermal analyses to determine temperatures and heat flux.

You can access all of these design types and features from the **Project** menu, and any combination of design types can be inserted into a single project file. The schematics can be used to wire up the different field solver models and create a model of a high-level system. Ansys Electronics Desktop provides an efficient way to manage complicated projects that require several different analysis tools to model all of their pieces. Designs can also be parameterized. With the help of the Optimetrics feature, the best design variations can be made available to other modules when the designs are linked into a higher-level simulation. This lets you study the effect of varying a design parameter on the behavior of the entire system.

You can access these design types and features from the Windows launcher. You can use the ACT Toolkit for HFSS-EMA3D Datalink to launch and use this tool.

The following illustration shows how the Ansys Electronics Desktop can be used to model different components for radar system analysis. An antenna array created in HFSS is linked to an IE design of an F16 aircraft. The low noise amplifier and bandpass filter are two important components in the receiver part of the radar module circuit design. The low noise amplifier and the filter can be modeled in HFSS 3D Layout and linked together in a circuit simulation, along with other components of the radar module connected to the antenna array. The outputs of the radar module can be used to drive the antennas using the push excitation feature, whereby the voltages on the antenna array ports can be automatically set to correspond to those of the driving circuit. The push excitation feature enables the user to view electromagnetic fields when the array is driven by the radar module circuit.



Ansys Electronics Desktop Student

> **Important:**
>
> Do not install the same release version of Electronics Desktop and Electronics Desktop Student on the same machine.

Ansys Electronics Desktop Student is a free Windows version of the Electronics Desktop that allows you to model, mesh (if applicable), solve, and post-process in HFSS, Q3D Extractor, 2D Extractor, Circuit, Maxwell 3D and Maxwell 2D, RMxprt, and Icepak. The following list of limitations applies to all supported design types in Ansys Electronics Desktop Student.

> **Note:**
>
> Unsupported design types (e.g., Mechanical) do not open in Electronics Desktop Student.

- Geometry export not supported
- Import of DXF and STEP files only
- Local solve only (remote configuration not supported)
- High Performance Computing limited to 4 cores
- optiSLang and LSDSO not supported

- Integration with Ansys Workbench not supported
- Beta features not supported
- Linux version not supported

Refer to the following sections to review limitations specific to the supported design types in Ansys Electronics Desktop Student.

HFSS Student Limitations

- SBR+ not supported
- Mesh assemblies not supported
- Mesh element count limit (analysis and post-processing)
  - 3D volume: 64,000 elements
  - 3D surface: 8,000 elements
  - 2D: 2,000 triangles
- Circuit model generation from S-parameters not supported

Q3D Extractor and 2D Extractor Student Limitations

- Mesh element count limit (analysis and post-processing)
  - 3D volume: 64,000 elements
  - 3D surface: 8,000 elements
  - 2D: 2,000 triangles
- Circuit model generation from S-parameters not supported

Circuit Limitations

- Netlist design type not supported
- Component limit: 50 components
- Command line analysis initialization not supported
- Device data generation not supported
- Access to layout functionality not supported
- Circuit model generation from S-parameters not supported

Maxwell and RMxprt Student Limitations

- Mesh element count limit (analysis and post-processing)
  - 3D volume: 64,000 elements
  - 3D surface: 8,000 elements

    - 2D: 2,000 triangles
- 3D and 2D Transient simulations

Icepak Student Limitations

- Mesh element count limit (analysis and post-processing): 512,000 elements
- Classic Icepak files (.tzr) with ECAD not supported

System Requirements

Ansys Electronics Desktop supports certain versions of Windows and Linux. For supported platforms and system requirements, go to the Platform Support website and select the following document:

Ansys 2025 R2 **- Platform Support by Application / Product (PDF)**

This document covers all Ansys products. Refer to the **Electronics Applications** section.

Limitations of Linux Installations:

While the majority of the Ansys Electromagnetics Suite applications and features are supported on both Windows and Linux platforms, some are not supported on Linux. For up-to-date platform support by product:

1. Go to the Ansys Platform Support website.
2. From the list of documents, click the link, Ansys 2025 R2 - Platform Support by Application / Product (PDF).
3. This document includes platform support information for all Ansys products. Refer to the section labeled Electronics Applications.

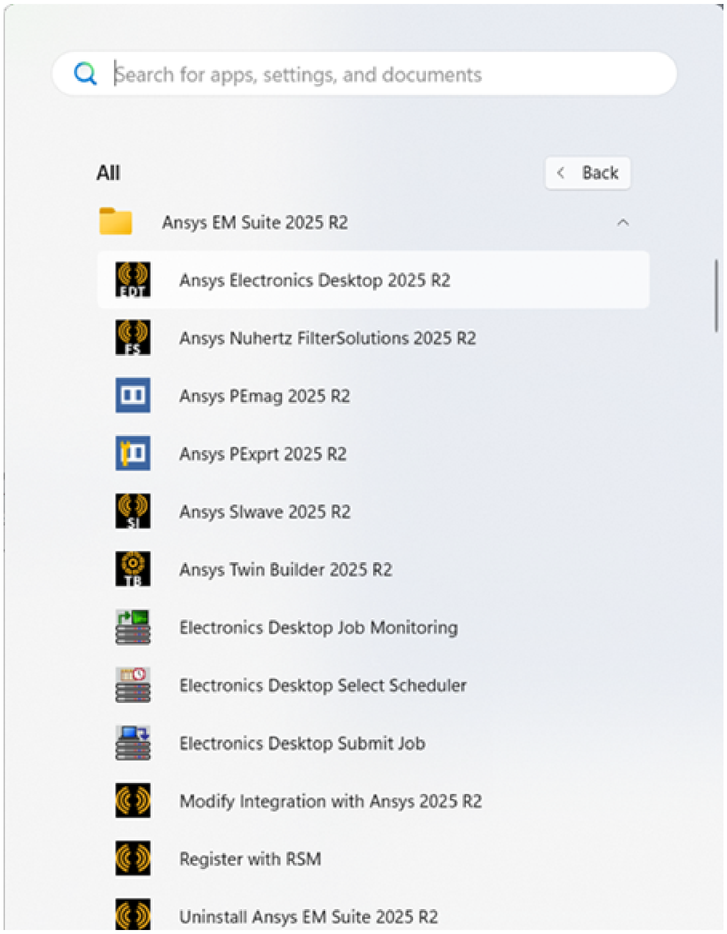
Launching Ansys Electronics Desktop

Follow the instructions below to launch Ansys Electronics Desktop in either Windows or Linux.

Windows:

Once you have installed Ansys Electronics Desktop, start the program using one of the following methods:

- From the Start menu, select **Ansys EM Suite 2025 R2** > **Ansys Electronics Desktop 2025 R2**.



If the program fails to start, ensure that:

- You have installed the licensing option provided to you. For detailed information on installing the software and licenses, see: Ansys EM Suite Installation. This topic provides links to the Ansys installation guides (Windows and Linux) and product-specific platform support information.
- Ansys QA Services are not enabled. To disable, change the value of the ENABLE_ANSYS_QA_SERVICES environment variable to 0.

Linux:

On Linux systems, an interactive Ansys Electronics Desktop session can be launched either from the command line, or from a GNOME-compliant desktop environment. The graphical session must be running directly on the machine's console or via a remote connection like VNC.

Launching AEDT from the command line:

When AEDT is installed, a sub-directory is created under the installation directory: /v252/AnsysEM/. This directory contains the software executable files and wrapper scripts.

To launch AEDT from the command line, invoke it via its wrapper script. If, for example, AEDT has been installed in the /opt/ansys_inc/ directory, the wrapper would be:

- /opt/ansys_inc/v252/AnsysEM/ansysedt

When this command is run for the first time on a machine, messages will be printed to the terminal as the software automatically configures itself. AEDT will then launch.

> **Note:**
>
> The installation directory contains a file named ansysedt.exe. Do not attempt to run this directly. The corresponding wrapper scripts perform essential environment setup tasks that the software needs to function properly.

Starting from GNOME desktop:

- Click the Activities button on the GNOME panel to show installed applications:

- Click the icon for Ansys EM:

Obtaining Information about the Software and Release

To obtain information about the software and release:

1. Click **Help** > **About Ansys Electronics Desktop**.
   The **About Ansys® Electromagnetics Suite [release number]** dialog box appears, listing information about the product.
2. Click the **Installed Components** tab to view a list of software installed.

3. Click the **Client License Settings** tab to view information about the following:
   - Provider Name
   - Ansys License Version
   - FlexNet Publisher Servers
   - Admin Directory
   - Customer Number
   - FLEXlm Version
   - Redirect Info
4. To export the software information:
   a. Click **Export**.
   The **Save As** dialog box appears.
   b. Browse to the location where you want to save the information as a text file.
   c. Type a name for the file in the **File name** text box. The **Save as type** drop-down menu is already specified as **Export (*.txt)**.
   d. Click **Save**.
5. Click **OK** to close the **About Ansys® Electromagnetics Suite [release number]** dialog box.

Ansys Product Improvement Program

This product is covered by the Ansys Product Improvement Program, which enables ANSYS, Inc., to collect and analyze anonymous usage data reported by our software without affecting your work or product performance. Analyzing product usage data helps us to understand customer usage trends and patterns, interests, and quality or performance issues. The data enable us to develop or enhance product features that better address your needs.

How to Participate

The program is voluntary. To participate, select **Yes** when the Product Improvement Program dialog box appears. Only then will collection of data for this product begin.

How the Program Works

After you agree to participate, the product collects anonymous usage data during each session. When you end the session, the collected data is sent to a secure server accessible only to authorized Ansys employees. After Ansys receives the data, various statistical measures such as distributions, counts, means, medians, modes, etc., are used to understand and analyze the data.

Data We Collect

For all products that offer the Ansys Product Improvement Program, we only collect anonymous data such as session statistics, hardware information, types of loading, solution types, solution statistics, and similar data. The specific data collected varies from product to product.

For Ansys Electronics, we collect the following information:

- Application
    - Build information
    - System information
        - Country
        - Country code
        - CPU architecture
        - CPU brand
        - CPU identifier
        - Graphics card
        - Operating system
        - Operating system version
        - Processor count
        - Time zone
        - Total RAM value
- Session
    - Workbench session
    - Total CPU time
    - Execution mode
    - Start method
    - Number of processes
    - Number of compute nodes (HPC)
    - Session begin
    - Session end
- Mesh
    - Number of nodes
    - Number of elements
    - Number of zones
    - Number of faces

Data We Do Not Collect

The Product Improvement Program does not collect any information that can identify you personally, your company, or your intellectual property. This includes but is not limited to names, addresses, file names, part names, geometry- or design-specific inputs, material property values, etc. We make no record of where we collect data from.

Opting Out of the Program

You may stop your participation in the program any time you wish. To do so, select menu item **Help > Ansys Product Improvement Program** or ribbon item **Help > Ansys > Ansys Product Improvement Program**. This activates the **Ansys Product Improvement Program** dialog box. On this dialog select the **No I am not willing to participate** radio button to opt out of the program. Select the **Yes, I am willing to participate in the Ansys Product Improvement Program** radio button to opt in to the program. Click the **OK** button to accept the radio button selection. Data is no longer collected or sent if you opt out.

> **Note:**
>
> You can disable the Ansys Product Improvement Program for all users so that each user is not prompted to enable the Program when they first start Electronics Desktop. After installing the software, run the following command as a user with permissions to modify the installed file set:
>
> ```
> UpdateRegistry.exe -set -ProductName ElectronicsDesktop2025.2 -
> RegistryKey
> Desktop/Settings/ProjectOptions/ProductImprovementOptStatus -
> RegistryLevel install -RegistryValue 1
> ```

The ANSYS, Inc., Privacy Policy

All Ansys products are covered by the ANSYS, Inc., Privacy Policy, which you can read here.

Frequently Asked Questions

1. *Am I required to participate in this program?*

   No, your participation is voluntary. We encourage you to participate, however, as it helps us create products that will better meet your future needs.

2. *Am I automatically enrolled in this program?*

   No. You are not enrolled unless you explicitly agree to participate.

3. *Does participating in this program put my intellectual property at risk of being collected or discovered by ANSYS?*

   No. We do not collect any project-specific, company-specific, or model-specific information.

4. *Can I stop participating even after I agree to participate?*

   a. Select the menu item **Help** > **Ansys Product Improvement Program** or ribbon item **Help** > **Ansys** > **Ansys Product Improvement Program**. This will activate the **Ansys Product Improvement Program** dialog box.

   b. Select the **No I am not willing to participate radio** button to opt out of the program.

   c. Click the **OK** button to accept the radio button selection.

   d. Data will no longer be collected or sent.

5. *Will participation in the program slow the performance of the product?*

   No, the data collection does not affect the product performance in any significant way. The amount of data collected is very small.

6. *How frequently is data collected and sent to Ansys servers?*

   The data is collected during each use session of the product. The collected data is sent to a secure server once per session, when you exit the product.

7. *Is this program available in all Ansys products?*

   Not at this time, although we are adding it to more of our products at each release. The program is available in a product only if this Ansys Product Improvement Program description appears in the product documentation, as it does here for this product.

8. *If I enroll in the program for this product, am I automatically enrolled in the program for the other Ansys products I use on the same machine?*

   Yes. Your enrollment choice applies to all Ansys products you use on the same machine. Similarly, if you end your enrollment in the program for one product, you end your enrollment for all Ansys products on that machine.

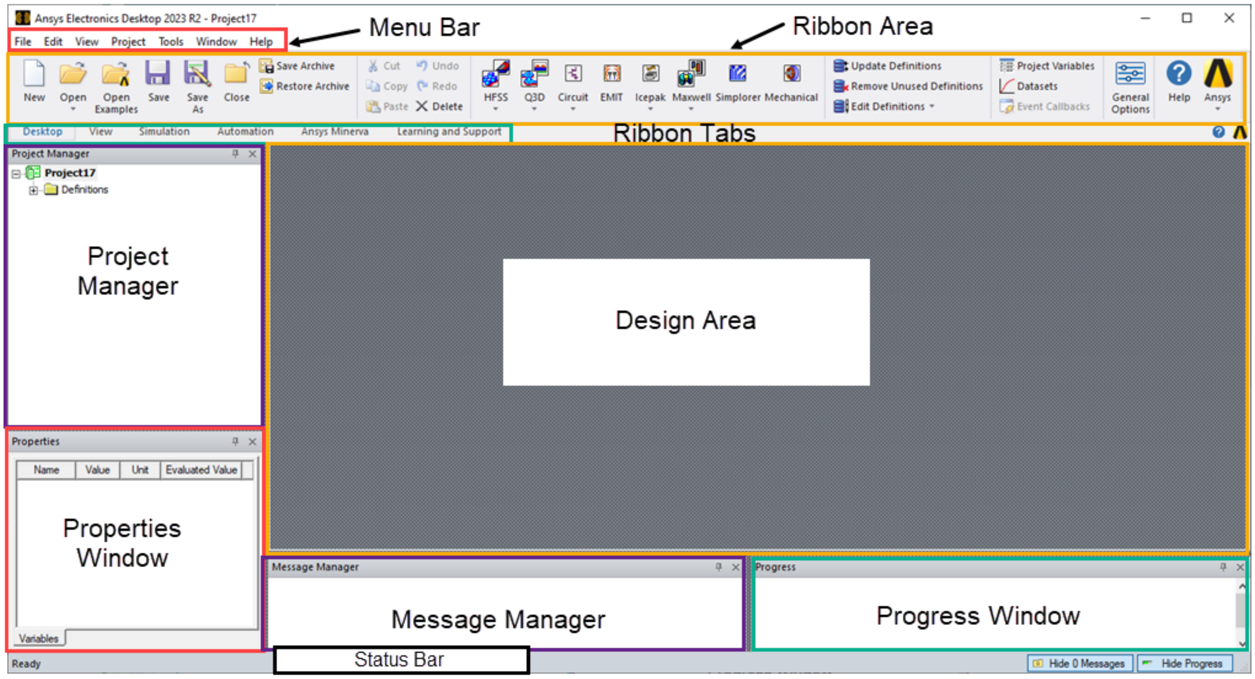
9. *How is enrollment in the Product Improvement Program determined if I use Ansys products in a cluster?*

   In a cluster configuration, the Product Improvement Program enrollment is determined by the host machine setting.

Ansys Electronics Desktop Overview

The Ansys Electronics Desktop consists of several windows, a menu bar, multitab ribbon, and a status bar. You can customize the appearance of the desktop by moving, hiding, or showing windows. Some menus, and other features, change depending on the type of project that is loaded and the editor that is active in the Design Area.

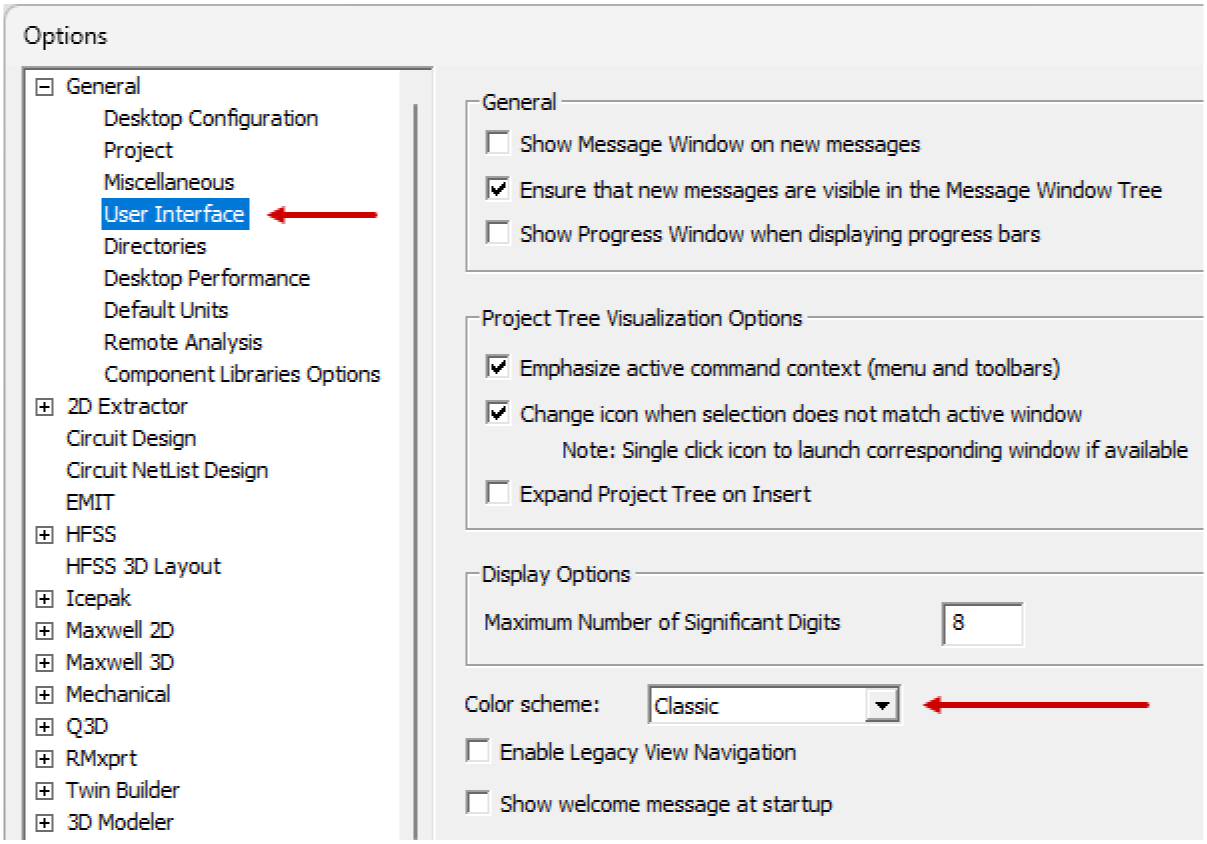
> **Note:**
>
> To ensure that all Electronics Desktop dialog boxes are visible on your monitor, it is recommended to not scale the display resolution above 100%.



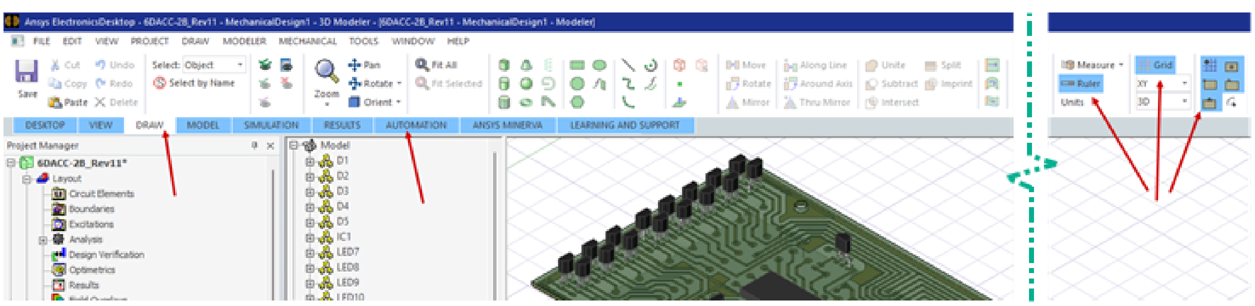
Choosing a Color Scheme [Beta]

In addition to the classic Ansys Electronics Desktop color scheme, two additional color schemes are now available as beta features. Choose the desired color scheme as follows:

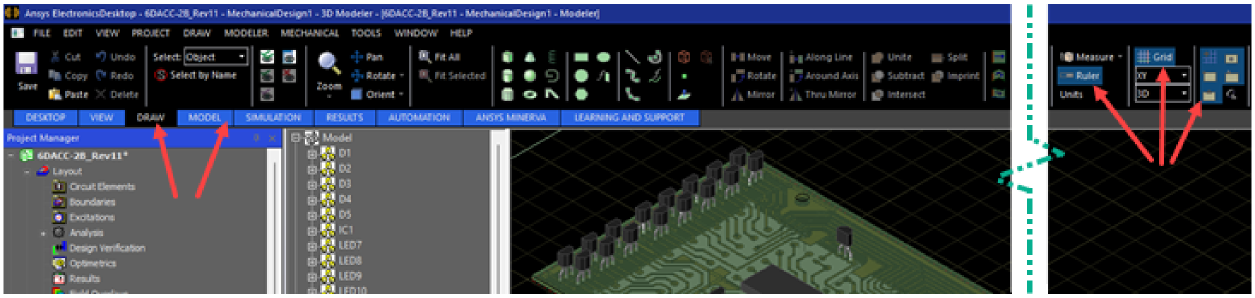
1. Access the *Options* dialog box using one of the following two methods:
   - On the **Desktop** ribbon tab, click  **General Options**.
   - From the menu bar, click **Tools** > **Options** > **General Options**.
2. In the tree at the left side of the dialog box, expand **Desktop Configuration** and select **User Interface**:



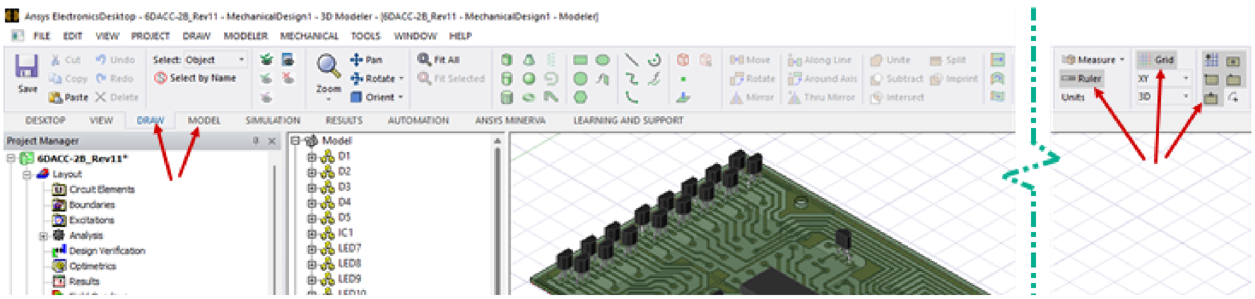
3. From the **Color scheme** drop-down menu, choose one of the following three options:
   - **Light (beta):** A light gray and white scheme with light blue highlighting for selected ribbon icons and for inactive ribbon tabs. The active ribbon has a white background (including the tab):



   - **Dark (beta):** A dark gray and black scheme with medium blue highlighting for selected ribbon icons and for inactive ribbon tabs. The active ribbon has a black background (including the tab):



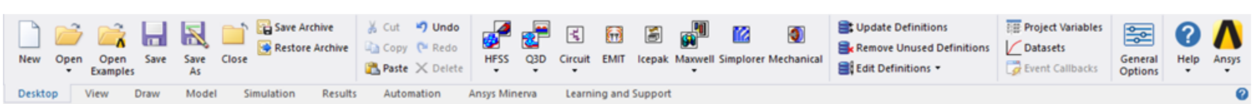
- **Classic:** A light gray and white scheme with medium gray highlighting for selected ribbon icons. The ribbon background is light gray. The name of the active ribbon tab has a blue font. A dark gray font is used for the inactive ribbon tab names:



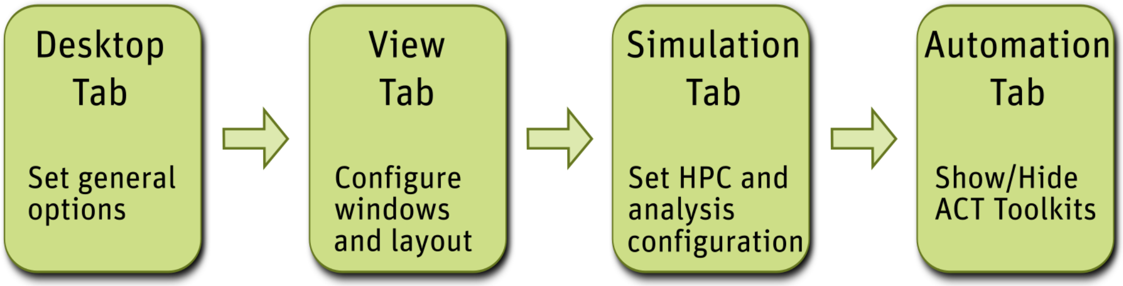
4. Click **OK** to close the *Options* dialog box.

Working with Ribbons

The ribbon is the rectangular area across the top of the application. It comprises various tabs, each one representing a subset of commands available from the menus. The initial set of tabs (Desktop, View, Simulation, Automation) offers commands for adding and opening projects, selecting solvers, configuring the Desktop display (window choice, size, and position), configuring the simulation environment, setting scripted Event Callbacks or General Options, and showing Automation features for recording and using scripts.



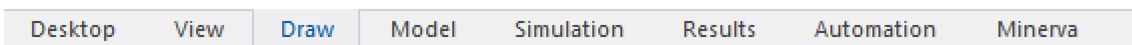
The initial workflow for local configuration and personal customization is:



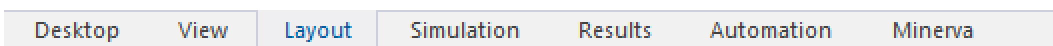
Available ribbon tabs depend on the design type. After you open or add a project and insert a design, you see additional ribbon tabs appear. The visible tabs and features are those that are appropriate for the design type and solver.

The **Desktop, View, Simulation, Automation, Minerva** and **Learning and Support** tabs appear for all design types. Additional tabs appear as follows:

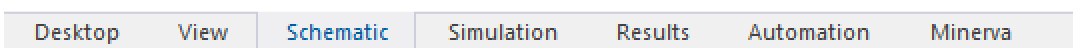
- For HFSS, Icepak, Maxwell, Mechanical, and Q3D designs, the **Draw**, **Model**, and **Results** tabs also appear:



- For an HFSS 3D Layout design, the **Layout** and **Results** tabs also appear:



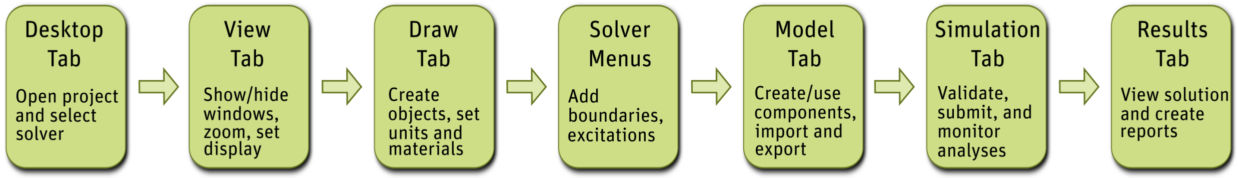
- For Circuit and Simplorer designs, the **Schematic** and **Results** tabs also appear:



In addition to **Desktop**, **View**, **Minerva** and **Learning and Support** the ribbon tabs that are applicable to EMIT designs are described as follows:

- **Simulation Tab** – contains tools that allow you to run simulations, perform validation checks, and schedule and monitor tasks.
- **Results Tab** – contains tools for creating reports and viewing solution data.
- **Automation Tab** – contains tools that allow you to run and record scripts, show or hide ACT extensions (Windows only), install PyAEDT (Beta), and access toolkits.

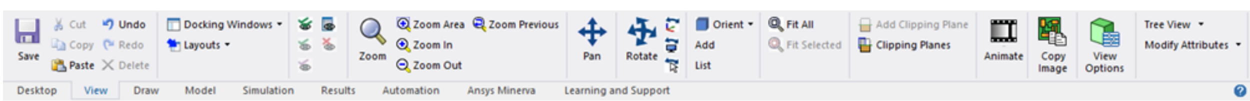
Each tab contains features specific to the design type, and the general workflow is from left to right.



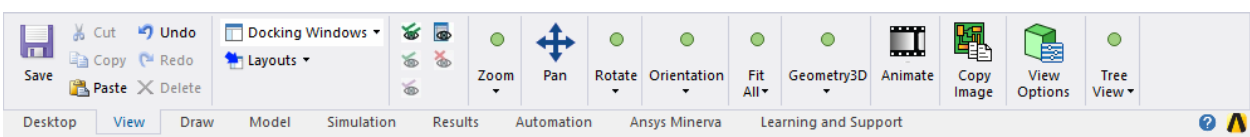
Note that workflow can vary depending on tasks at hand. For example, if you open a completed model in HFSS or use 3D components or the ACT Antenna Design toolkit (Windows only), you may not need the features of the **Draw** tab. If Validation on the **Simulation** tab identifies problems in a model, you may need to use the **Modeler** menu commands for Analysis and Heal, or use the Discovery link feature. For some tasks, such as assigning excitations or boundaries in HFSS, you must use the command menus, rather than the tabs.

Sizing the Ansys Electronics Desktop window affects the icons displayed on each tab, with priority given to the most used features.

For example, If you have inserted an HFSS project, the **View** tab displays commands appropriate for the active editor:

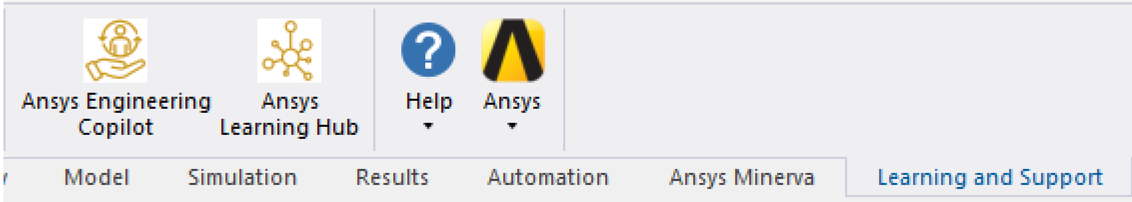


If you reduce the size of the Desktop window, the ribbon tabs become compressed. Fewer icons are shown, and available features are moved into drop-down menus rather than shown separately. The following example shows the **View** tab in a compressed state:



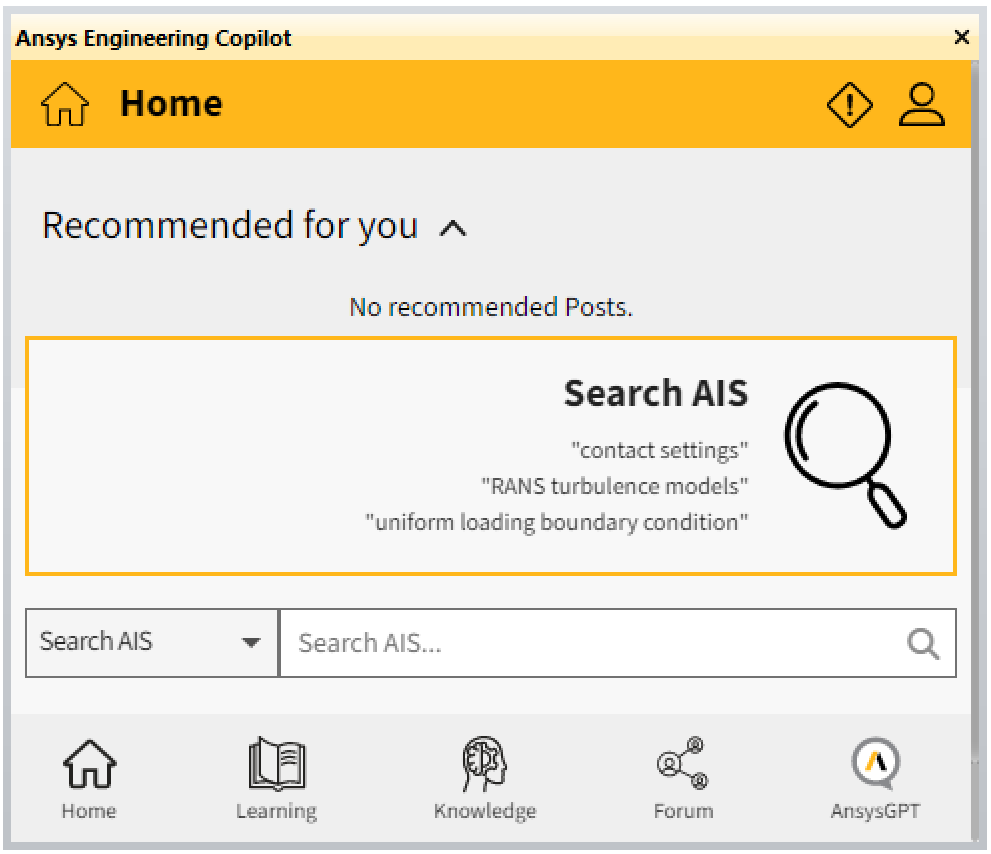
Other ribbon tabs are similarly compressed when the window size does not support the fully expanded arrangement of features.

The **Learning and Support** tab provides easy access to Technical advice, instruction, and examples on Ansys websites.



Ansys Engineering Copilot – Single point access to Ansys Innovation Space resources, such as AnsysGPT, Support Cases, Courses, Knowledge, Forum, and more.

You first login to your account. Then you are provided with a new window.



- **Ansys Innovation Courses** – Provide a wide range of Ansys Electronics Engineering courses using on-demand, self-paced video training and quizzes.
- **Ansys Knowledge** – Provides expert curated knowledge materials from FAQs to tutorials on simulation topics.

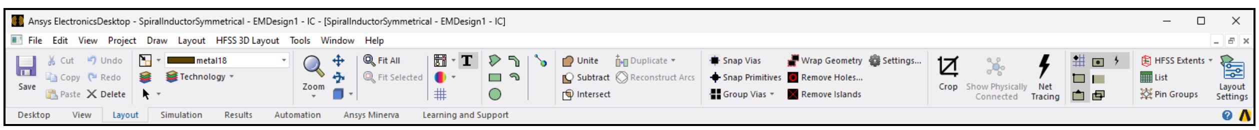
- **Ansys Learning Forum** – Provides Ansys blog containing discussion and presentations from Ansys experts, partners and customers.
- **AnsysGPT** – AnsysGPT is a Machine Learning (ML) Large Language Model (LLM) which has been trained in Ansys company information, including online help contents.

Ansys Learning Hub – opens a web page with subscription based access to, virtual and self-paced learning across the Ansys Software portfolio.

About Electronics Desktop - opens a dialog with version and release information, Ansys Electronics Desktop installed components, and licensing information.

IC Mode Layout Ribbon

In addition to exclusive tabs, HFSS 3D Layout contains two potential environments (i.e., General mode and IC mode). General mode typically opens by default. Different commands and options are available in each mode (e.g., after opening a GDSII file or selecting the **IC** layout from the **Design Settings** window (i.e., **HFSS 3D Layout** > **Design Settings** > **Design Mode** tab), the **Layout** tab will reconfigure to match the following screenshot).



The general workflow remains the same in both environments. Refer to Switching to IC Mode.

Ansys Electronics Desktop Windows

Ansys Electronics Desktop contains the following windows:

- Project Manager
- Message Manager
- Progress
- Properties

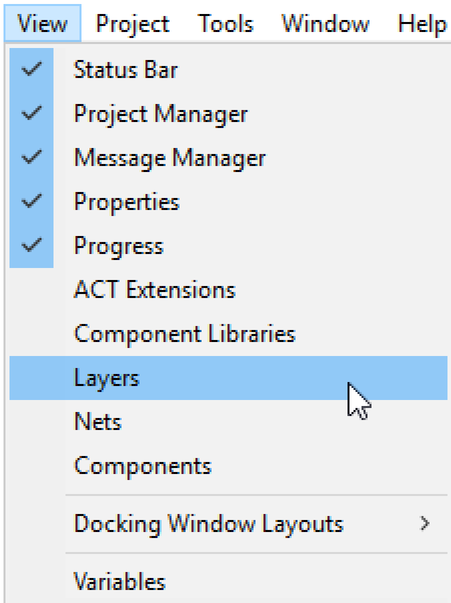
You can toggle the display of each window, auto-hide windows, or move and resize windows. Additionally, you can toggle the status bar or add ACT Extensions (Windows only).

Showing and Hiding Windows

To toggle the windows that comprise Ansys Electronics Desktop, use the **View** menu or the **View** tab on the ribbon.

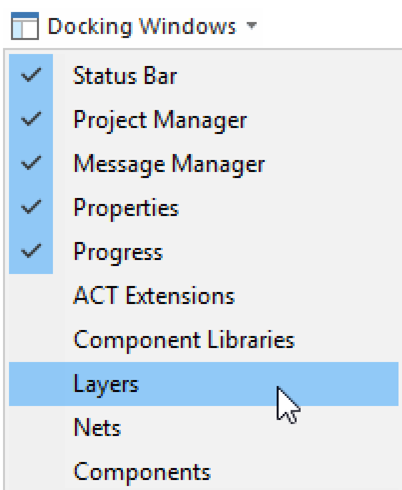
From the **View** menu:

- Click a window to show or hide it.



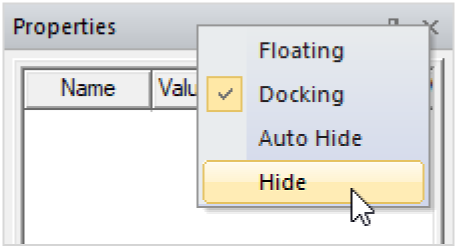
From the **View** tab:

- Use the **Docking Windows** drop-down menu to show and hide windows.



You can also hide a window directly from the window:

- Click the X in the window title bar, or
- Right-click in the window title bar and select **Hide**.



The visibility setting of a window is retained from one desktop session to the next.

Auto Hiding Windows

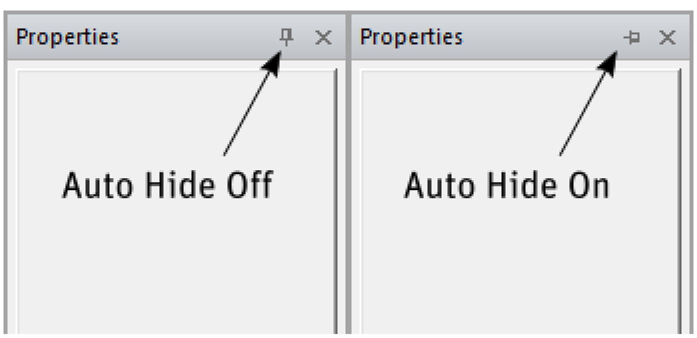
Windows can be moved out of the way so that only their title bars show. When you hover over this title bar, the window appears. This mode is called Auto Hide.

To automatically hide a window until the cursor is hovered over it:

- Click the pin icon in the window title bar, or
- Right-click in the window title bar and select **Auto Hide**.

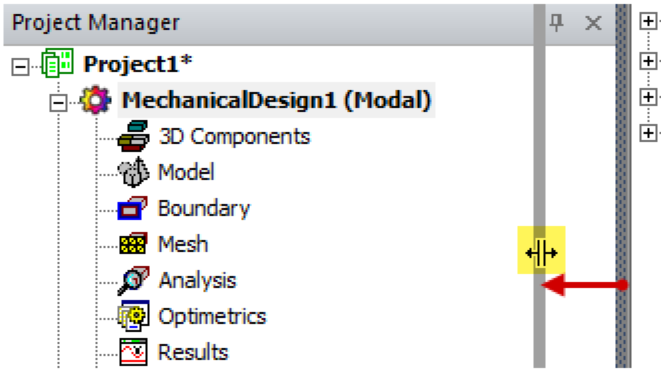
> **Note:** If a window shares vertical or horizontal space with another window, the first click will expand it to take up the entire area while hiding all other windows. Click the pin again to hide the window.

When a window is in Auto Hide mode, the pin icon flips on its side. Click the icon again to disable Auto Hide.

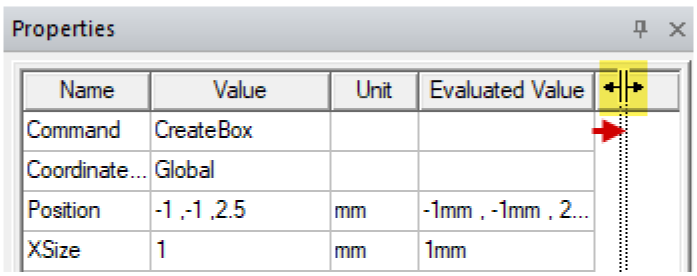


Moving and Resizing Windows

You can resize a window by clicking and dragging a vertical or horizontal window border. The cursor appearance changes to indicate when you're over a movable window border (as highlighted in yellow below):

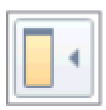


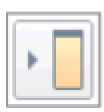
You can also click and drag the borders of columns within certain windows containing tabular data, such as the **Properties** window:

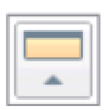


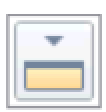
Major Ansys Electronics Desktop windows can be docked in various locations or can float over other windows. To move a window, click and drag the window's title bar. While dragging a window's location, several docking position icons appear on the screen.

By releasing the mouse button over the following icons, you can dock a window in the following four locations along the perimeter of the user interface:

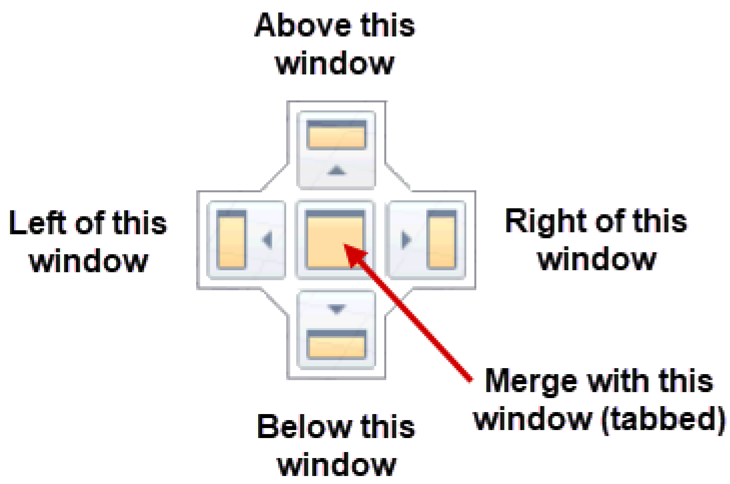
Left edge of user interface: 

Right edge of user interface: 

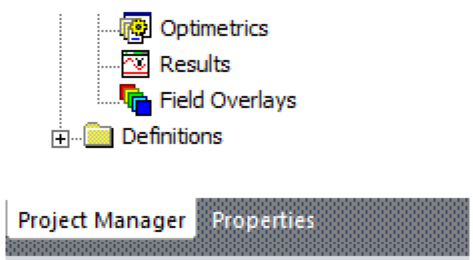
Top of user interface: 

Bottom of user interface: 

Additionally, you can use the following group of five docking locations to position the window relative to another window:



When you release the mouse button over the middle icon, the two windows are merged, with tabs at the bottom to select which one to view:



If the middle icon does not appear, it's because tabbed windows are not supported by one of the windows. For example, you cannot merge any window with the Modeler window.

If you release the mouse button while not over any of the above nine icons, the window stays in its current dragged location as a floating window.

Project Manager Window

The **Project Manager** window displays the open project's structure, which is referred to as the project tree.

The **Project Manager** window displays details about all open Ansys Electronics Desktop projects. The tree display is specific to the design type. For example:

To show or hide the **Project Manager** window, do one of the following:

- Click **View** > **Project Manager**.

A check box appears next to this command if the **Project Manager** window is visible.

- Right-click in the toolbars area on the desktop, and then click **Project Manager** on the shortcut menu.

A check box appears next to this command if the **Project Manager** window is visible.

Working with the Project Tree

The project tree is located in the **Project Manager** window and contains details about all open Ansys Electronics Desktop projects, as shown below:

The top node listed in the project tree is the project name. It is named Project*n* by default, where *n* represents the order in which the project was added to the current session of Ansys Electronics Desktop. Expand the project icon to view all the project's design information, material definitions, and 3D Components (if any).

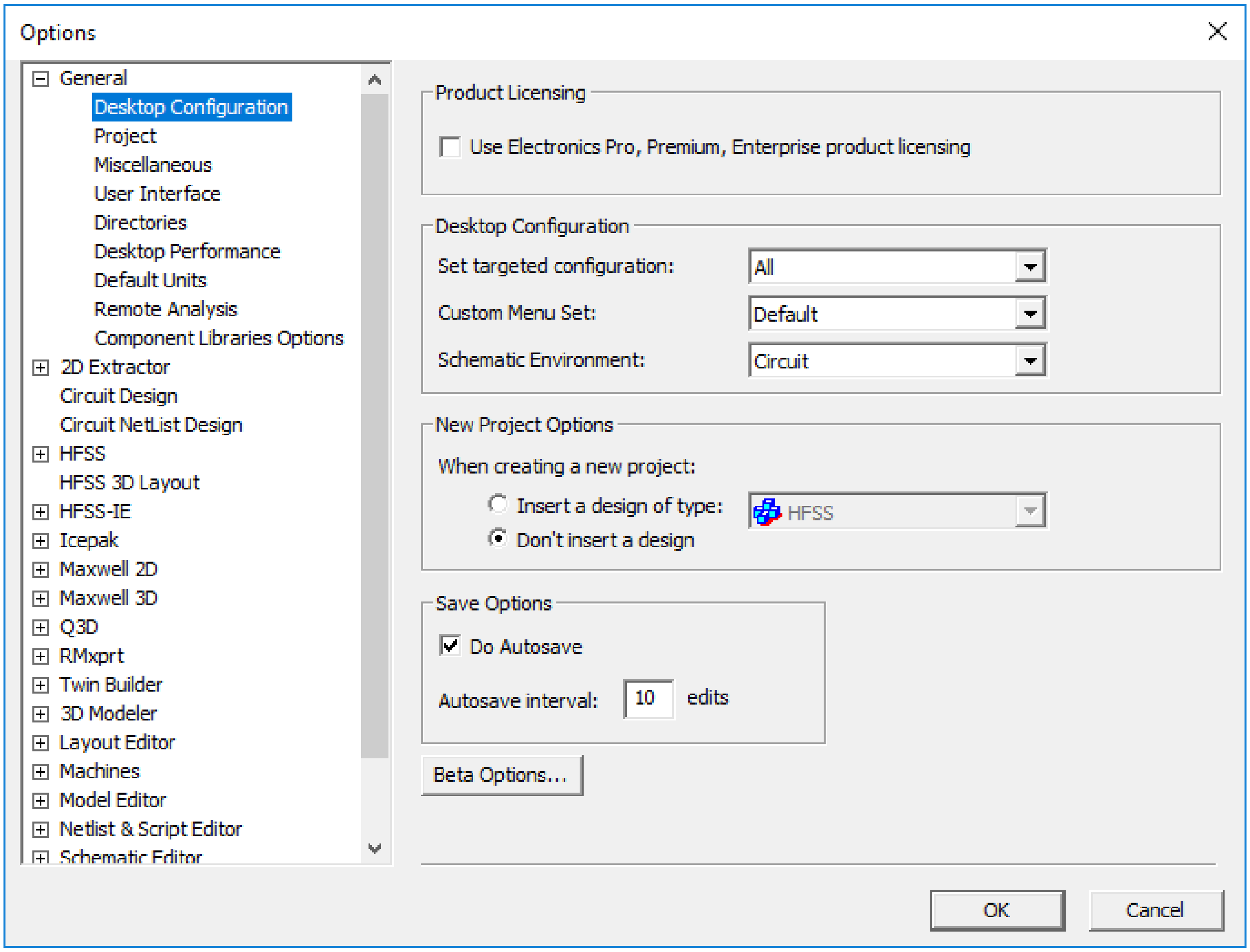
By default, the Project tree icon for the active window is highlighted. See General Options: Miscellaneous for options.

Setting the Project Tree to Expand Automatically

You can set the project tree to automatically expand when an item is added to a project.

1. Click **Tools** > **Options** > **General Options**.

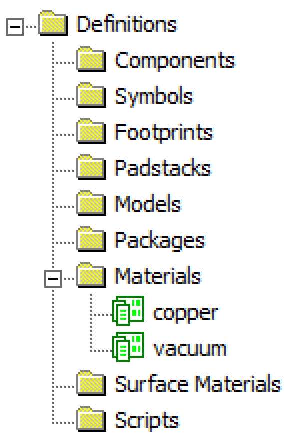
   The **Options** window appears.



2. Click **User Interface**.
3. In the **Project Tree Visualization Options** area, check the **Expand Project Tree on Insert** check box.
4. Click **OK**.

Viewing Material Definitions

The definitions node is listed at the bottom of the project tree and displays all of the material definitions that are assigned to the objects in the active model.



Viewing Ansys Electronics Desktop Design Details

When you insert an Ansys Electronics Desktop design into a project, it is listed as the second node in the project tree. It is named Ansys Electronics DesktopModel*n* by default, where *n* represents the order in which the design was added to the project. Expand the design icon in the project tree to view all of the specific data about the model, including its boundary conditions and material assignments, and field solution and post-processing information.

The Ansys Electronics DesktopModel*n* node contains the following project details:

3D Components	Displays any 3D Components added to the design.
Model	Displays the model geometries in the design.
Analysis	Displays the for an Ansys Electronics Desktop design. A solution setup specifies how Ansys Electronics Desktop will compute the solution.
Optimetrics	Displays any Optimetrics setups added to an Ansys Electronics Desktop design.
Results	Displays any that have been generated.
Field Overlays	Displays fields, which are representations of basic or derived field quantities on surfaces or objects. Plot folders are listed under **Field Overlays.** These folders store the project's plots and can be customized. See Setting Field Plot Defaults for information on how to customize the plot folders.
Documentation	Displays files you have added as documentation.
Definitions	Displays definitions for EMIT, including Components, Materials, and more (as applicable).

> **Note:**
>
> To edit a project's design details:
>
> - In the project tree, double-click the design setup icon that you want to edit.
>
> A window appears with that setup's parameters, which you can then edit.

Message Manager Window

The **Message Manager** window displays informational messages about various processes in Ansys Electronics Desktop, including messages related to simulations and any errors they may have produced.

Display or hide the **Message Manager** in one of two ways:

- Click **View** > **Message Manager**
  A check box appears next to this command if the **Message Manager** is visible.
- On the status bar, click **Show Messages** or **Hide Messages**.

Messages in the **Message Manager** window are organized first by project, then by circuit. Because a design can contain multiple circuits and subcircuits, sometimes with multiple analyses for each, this organization helps you to quickly determine where errors have occurred. The following icons appear next to a message to indicate information, warnings, errors, or actions:

 Indicates an informative message.

 Indicates a warning message that may require your attention.

 Indicates an error message that may require your attention.

 Indicates the existence of an action that is associated with the message. Click on the message to invoke the action (the cursor will change to a hand icon when it is placed over the action message).

Right-click in the **Message Manager** window to open a pop-up menu with the following options:

- **Clear messages** for the current model.
- **Copy messages** to the clipboard. This can be helpful for sending messages to application engineers.
- **Details** – opens an information dialog that contains the project and design data for the specified message.
- **Go to Reference** – allows you to right-click on an intersection error message after running a validation check. This selects intersecting objects in the current design being validated.

Setting the Message Manager to Open Automatically

You can set the **Message Manager** to open automatically to show new messages and errors and warnings.

Showing New Messages

You can set the **Message Manager** to automatically open when a new message appears.

1. Click **Tools** > **Options** > **General Options**.

   The **Options** window appears.
2. Expand **General** and click **User Interface**.
3. In the **General** area, check the **Show Message Window on new messages** check box.
4. Click **OK**.

Automatically Expanding the Message Manager Tree

You can set the **Message Manager** Tree to automatically expand when a new message is added.

1. Click **Tools** > **Options** > **General Options**.

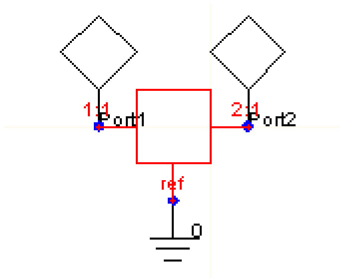
   The **Options** window appears.
2. Expand **General** and click **User Interface**.
3. In the **General** area, check the **Ensure that new messages are visible in the Message Window Tree** check box.
4. Click **OK**.

Action Messages

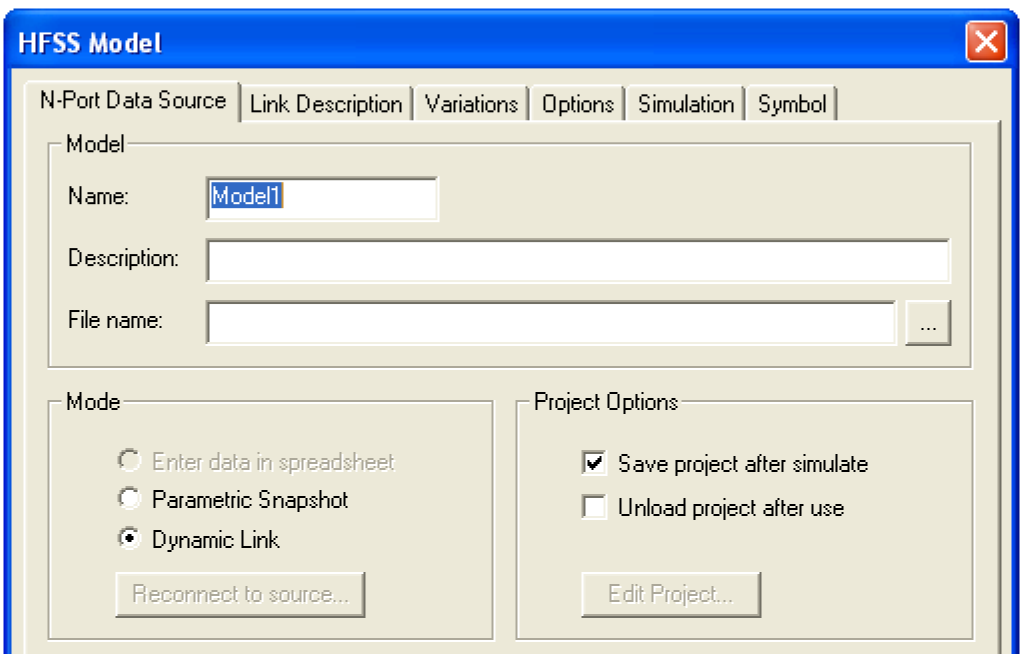
Messages displayed in the **Message Manager** can be associated with actions that can be invoked in order to address a condition you are alerted to by the message. If a message has an associated action, a magnifying-glass icon will be present to the left of the message icon. When the cursor moves over a message with an action, the cursor changes to a hand.

You can invoke a message action by clicking the magnifying glass icon, by double-clicking the message, or by right-clicking the message and selecting **Go To Reference** from the pop-up menu.

One example of a message-associated action is the selection of an associated object:



Another example is the opening of a related window:



Clearing Messages

The **Message Manager** clears at the start of each analysis. To manually clear messages for a project, right-click the message tree and select **Clear Messages for *<ProjectName>***.

Hiding the Message Manager Window until Messages Appear

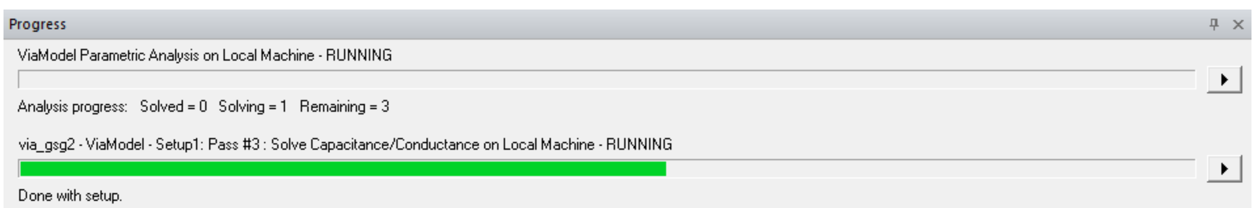
If you prefer to hide the Message Manager until a message is added:

1. Select **View** > **Progress Window** to hide the window. For more information see the View drop-down menu.
2. Click **Tools** > **Options** > **General Options**.
3. Check the **Show message window on new messages** check box.

The **Message Manager** window will re-open when Ansys Electronics Desktop reports errors, warnings, or successful completion of any simulation.

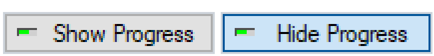
Progress Window

The **Progress** window monitors a simulation while it is running. Each simulation has its own progress bar. Right-clicking the bar allows you to abort the simulation or view simulation details.



To display or hide the **Progress** window, do one of the following :

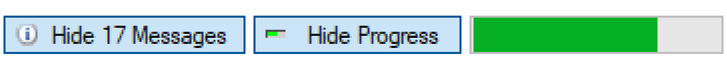
- Click **Show Progress** or **Hide Progress** on the status bar:



- Click **View** > **Progress Window**.

A check box appears next to this command if the **Progress** window is visible.

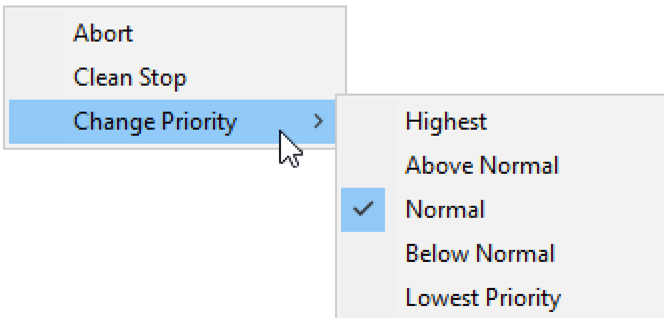
When more than one progress bar is active, the top progress bar is represented on the status bar with a progress indicator.



The progress window is also dockable, so you can reposition it.

Stopping or Aborting Simulation Progress

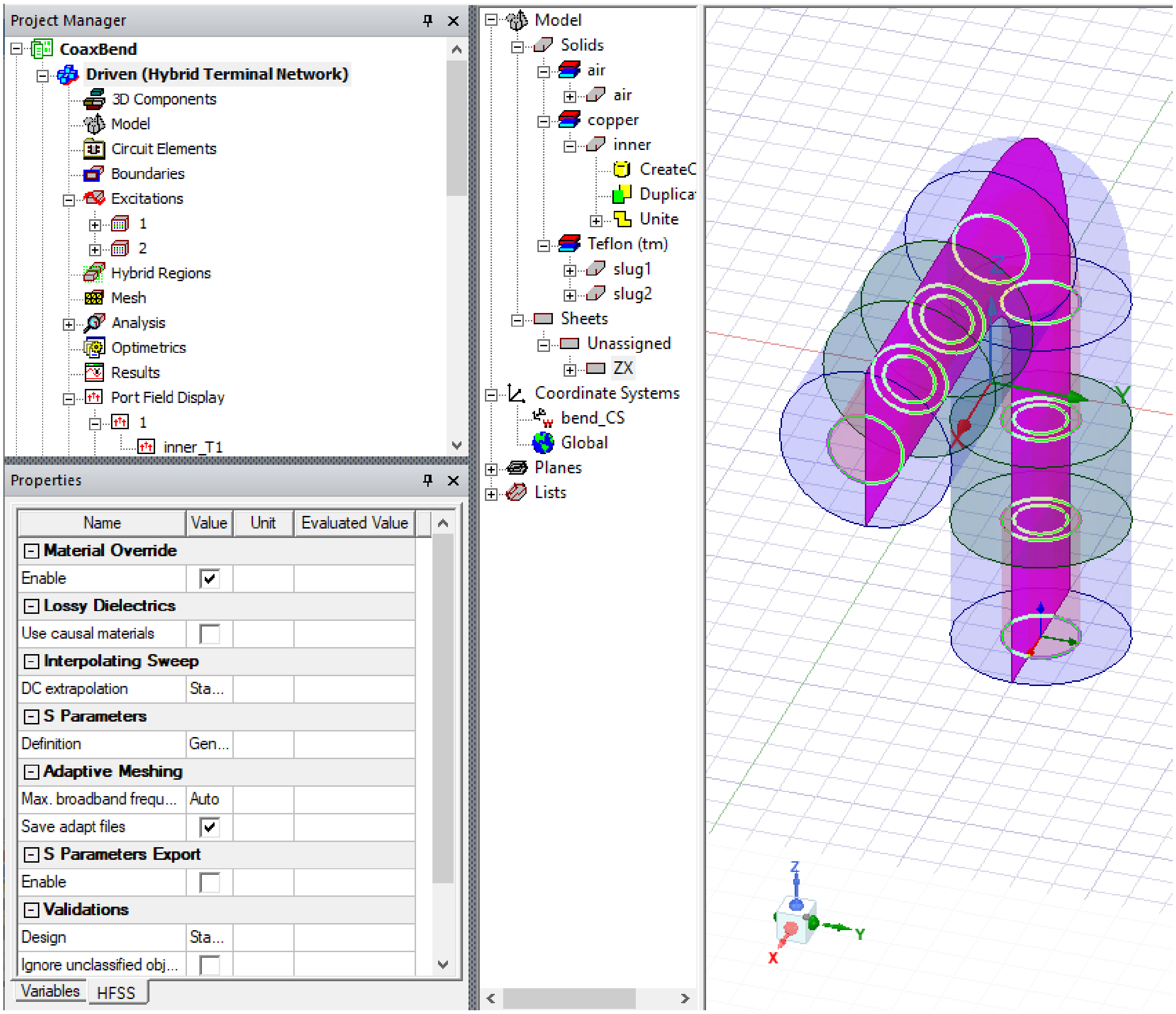
- To abort progress, right-click in the **Progress** window and select **Abort**.
- To stop the simulation cleanly between time steps, right-click in the **Progress** window and select **Clean Stop**.
- To change priority, right-click. From the **Change Priority** submenu, select from Highest, Above Normal, Normal, Below Normal, and Lowest.



Properties Window

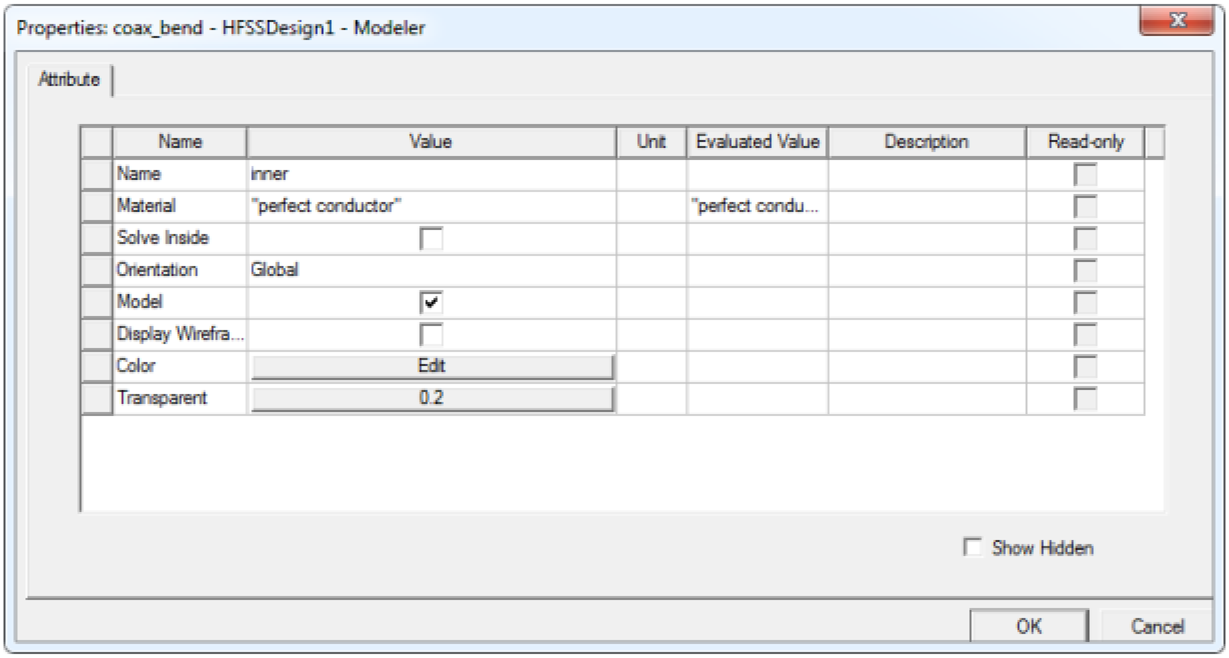
The **Properties** window displays and allows you to edit the properties of an item selected in the project tree, the history tree, or the **3D Modeler** window. The properties shown and their editability vary depending on the type of item selected.

You can choose to show or hide a docked **Properties** window as part of the desktop.



You can move and resize the docked **Properties** window within the desktop to suit your work style. When the **Properties** window is docked, it displays the properties of any item you select in the Project tree, the History Tree, or the 3D Modeler window. Select **View** > **Properties** to remove the docked properties window.

Regardless of whether or not you display a docked **Properties** window on the desktop, you can open an undocked **Properties** window for any item in the project tree or history tree by double-clicking the item.



Opening the Properties Window

1. Select the object whose properties you want to view.
2. Click **Edit** > **Properties**.

   The **Properties** window for that object appears.

3. When you are finished making changes, click **OK**.

Rather than opening a separate window, you can have the **Properties** window displayed within the desktop.

Showing and Hiding the Properties Window

To show or hide the **Properties** window on the desktop, do one of the following:

- Click **View** > **Properties**.

A check box appears next to this command if the **Properties** window is visible.

- Right-click in the toolbars area at the top of the desktop, and then click **Properties** on the shortcut menu.

A check box appears next to this command if the **Properties** window is visible.

Setting the Properties Window to Open Automatically

To set the **Properties** window to open after an object is drawn, do the following:

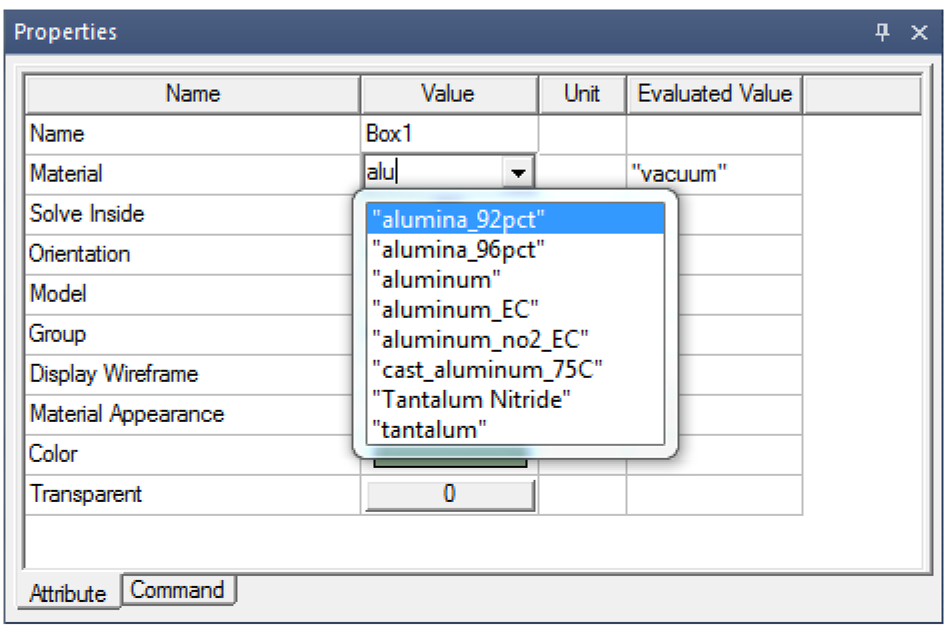
1. Click **Tools** > **Options** > **General Options**.

   The **Options** window appears.
2. Expand **3D Modeler** and select **Drawing**.
3. Check the **Edit property of new primitives** check box.

Hereafter, when you draw an object in point mode the **Properties** window opens. However, if you draw an object in Dialog mode, this setting is ignored.

Auto-Complete for Variables and Properties in Electronics Desktop

When you edit a properties or variable text field, Electronics Desktop can display possible matches for what you type. This can help if a variable or material name is long. You can save time by selecting a pre-determined match rather than typing out the entire name. The following figure shows an example of auto-complete for material names:

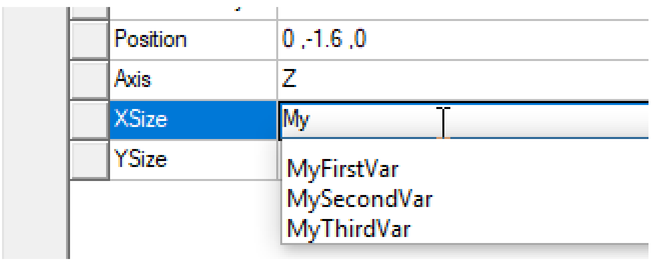


Using Auto-Complete

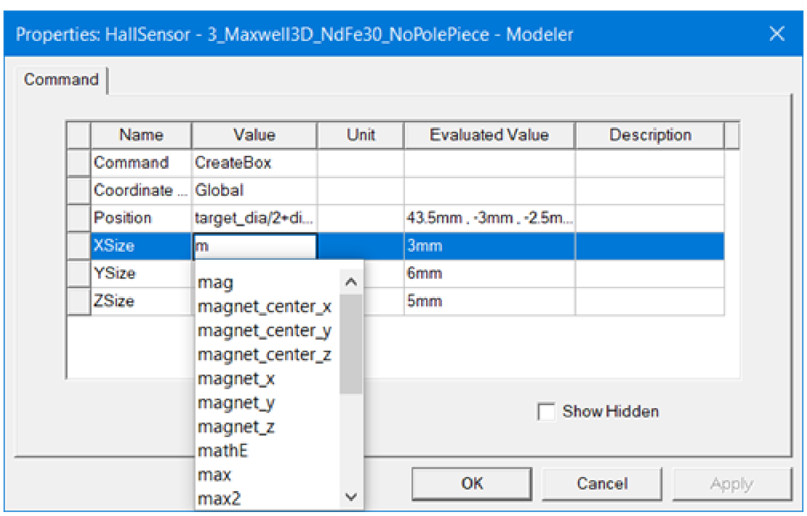
Certain commonly used text fields have auto-complete configured. When you start typing in these cases, matches display in a list below the text field. If there is no matching text, then no list displays. The list is automatically sized, but you can resize it. Electronics Desktop remembers the new size when doing additional auto-complete matching in the same text field. Switching to a different text field resets the sizing.

Hitting **Tab** or **Enter** accepts the current selection for auto-complete, which replaces the text typed with the full auto-complete match and hides the list. Otherwise, hitting the escape key **Esc** hides the auto-complete list. Typing more letters causes auto-complete matching to resume.

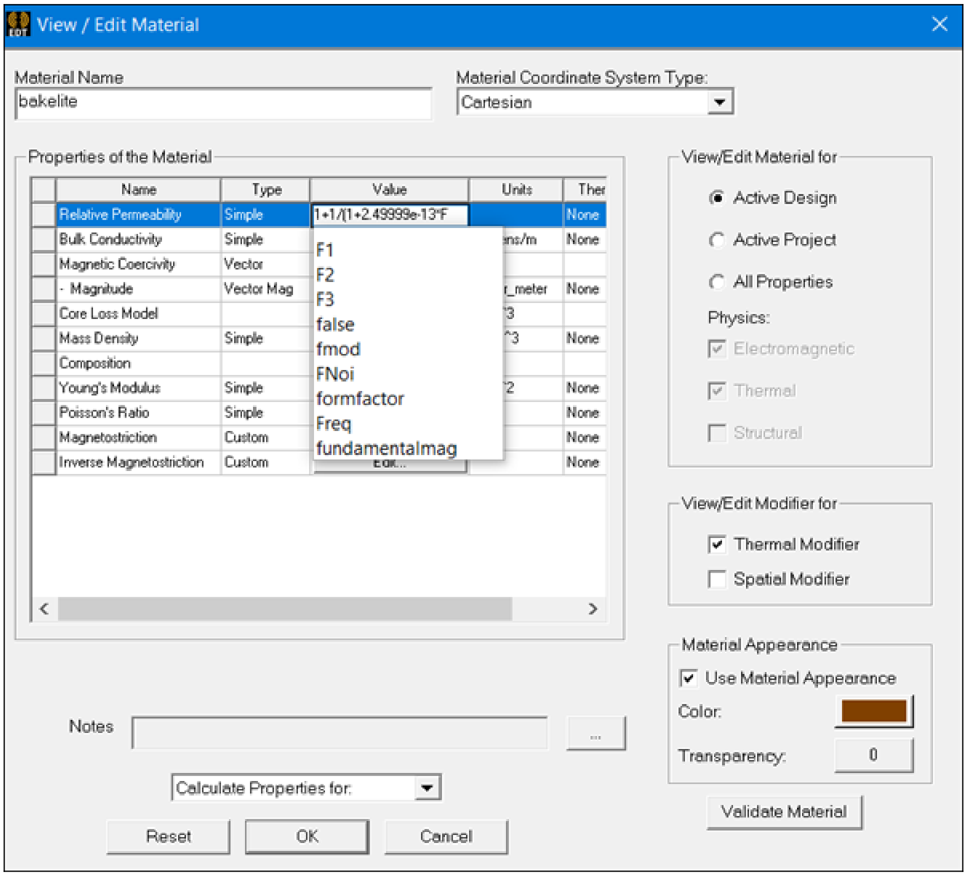
Where you select design properties, auto-complete works with expressions for the values, providing matches for the names of constants, intrinsic variables, functions, project variables (prefixed with "$"), and design properties. When project variables are displayed (that is, you select the project in the Project tree), auto-complete for value expressions also works, except without matching design properties. The variable auto-complete list does not include Separator Variables or Hidden variables since these could cause invalid formulas.



Properties of other items may also have auto-complete configured to work with value expressions (for example, properties of a CreateBox command or of a circuit component). The following shows a design property with auto-complete matching:

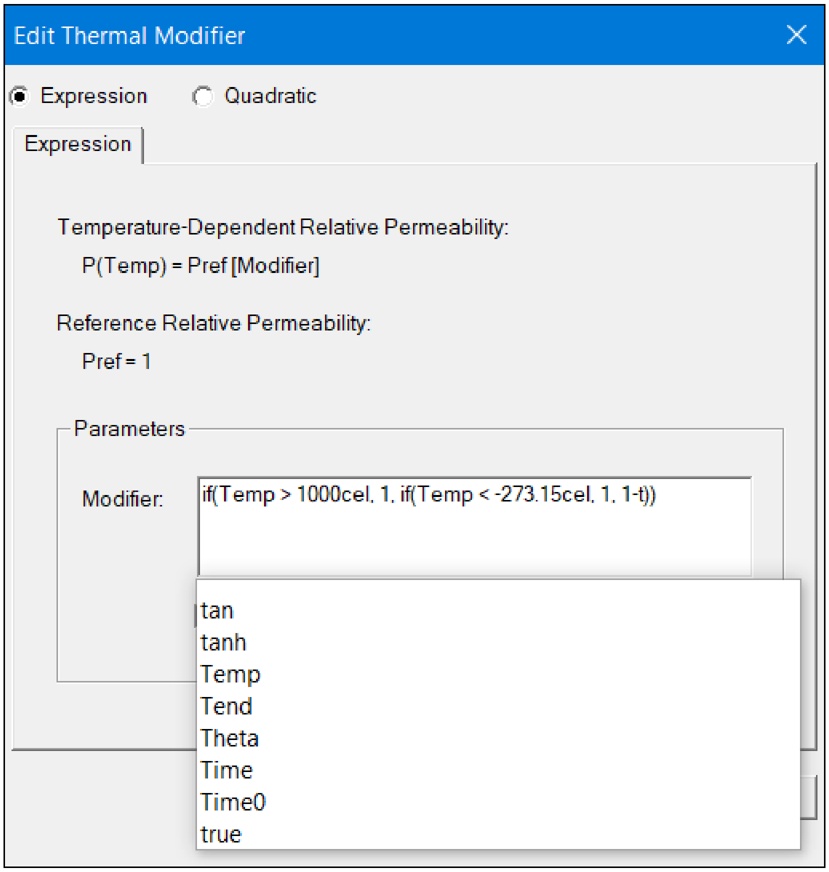


Auto-complete also works with value expressions for material properties:



Note the scroll bar, indicating that there are more matches than those currently displayed.

Auto complete also works with thermal modifier expressions for Materials:



Modifying Object Command Properties Using the Properties Window

The **Command** tab in the **Properties** window displays information about an action selected in the history tree that was performed either to create an object (such as the **Draw** > **Box** command) or to modify an object (such as the **Edit** > **Duplicate** > **Mirror** command).

Not all command properties can be modified. In general, command properties you can typically modify are numeric values, such as position values, size values, and various other coordinate values. You can also modify many of the unit settings for a command property. You can modify names, but names must include only letters, numbers and underscores. Illegal names are not accepted and generate a message in the Message Manager window.

1. In the history tree, select the command for which you want to edit properties.

> **Tip:**
>
> Press and hold **Ctrl** to select multiple commands. If you select multiple commands, only the common (shared) properties will display under the **Command** tab.

2. Under the **Command** tab in the **Properties** window, edit the command's properties.

Depending on the property type, you can edit it by doing one of the following:

- Select the check box to apply the property; clear the check box to disable the property.
- Click in the field and edit the numeric values or text, and then press **Enter**.
- Click the button and edit the current settings in the window that appears.
- Click the attribute and select a new setting from the menu that appears.

Properties Dialog Box

Like the **Property** window, the **Properties** dialog box shows the properties or parameters of selected objects, and, where appropriate, allows editing the values of these properties. It extends **Property** window functions with additional editing commands, and with settings for tuning, optimization, sensitivity, and statistical analysis, that are not available through the **Property** window. It is also used to define project variables. The **Properties** dialog box has the following tabs:

- General

Opening the Properties Dialog Box

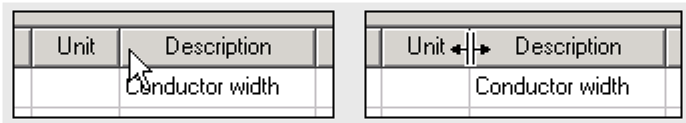
You can open the **Properties** dialog box in two ways:

- To open the **Properties** dialog for a component, double-click the component.
- Right-click the component and select **Properties**.

You can manipulate the **Properties** dialog box in the following ways:

- Close the **Properties** dialog box by clicking the **Esc** key.
- Resize the **Properties** dialog box by dragging its edges.
- Change the relative widths of adjacent columns by dragging the header separators

between them.



General and Symbol Tabs

The **General** tab in the Properties Dialog Box lists the selection's name, symbol name, reference designator, and so on. These are generally not editable. The information is identical to that on the General tab of the Property Window.

The **Symbol** tab provides information on the location of the component symbol in the schematic. The information is identical to that on the Symbol tab of the Properties Window.

When the **Properties** dialog box is opened for a design (**Design Properties**), both the General and Symbol tabs are initially empty.

Property Window for Schematic and Layout Editors

The **Property** window shows the properties or parameters of objects selected in the schematic and layout editors, and objects in the **Project Manager** window. It allows you to modify property values. The **Property** window has three tabs:

- The Param Values Tab

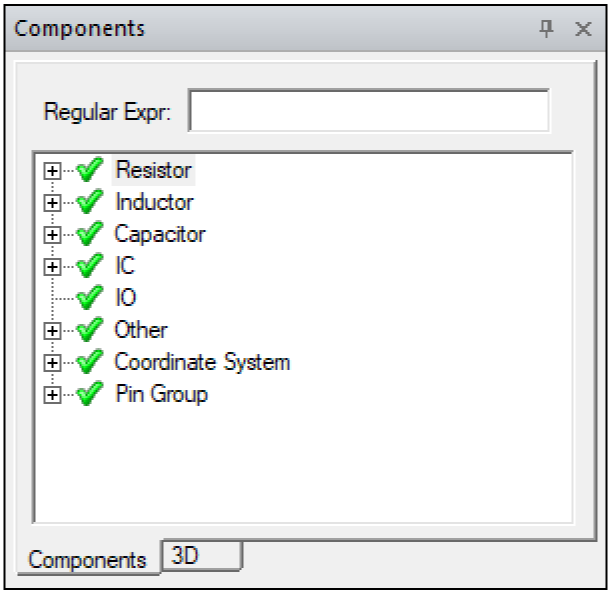
Closing the Property Window

To close the **Property** window, click **View** > **Properties** to deselect it. You can repeat this step to reopen the window.

For more information, see the View Pulldown Menu.

Working with the Components Window

The **Components** window is a dockable window that is used to view and configure various component settings for Layout components. Use the **Components** window to select a component or a component class, add ports to a component, and configure multiple ports across components. You can also select multiple components in the Layout editor to configure pins and ports for each simultaneously. The window can be resized and relocated.



To show or hide the **Components** window on the desktop, click **View** > **Components**. A check box appears next to the command if the **Components** window is visible.

With a component selected, the following configuration controls are available from the right-click menu:

- **Type** – reclassifies the component type to **Resistor**, **Inductor**, **Capacitor**, **IC**, **IO**, or **Other**.
- **Enable**/**Disable** - enables or disables the component.
- **Model** – opens the **Component Model** window, where you can change the component model definition, including die, solder ball, and port properties. Changing the model scope will update every reference designator that belongs to the model definition.
- **Create Ports on Component** – adds ports for the component
- **Remove Ports on Component** – removes ports for the component.
- **Fit Selected** – fits the selected component in the active view window.

> **Note:** When a component classification is selected, any changes are applied to all components within the class.

For more information, see Layout-based Component Encapsulation.

Component Libraries Window

The Component Libraries Window is a dockable window that is used to select and insert various pre-configured components. The window can be resized and relocated.

To show or hide the Component Libraries Window on the desktop:

- Click **View>Component Libraries**

A check box appears next to the command if the Component Libraries Window is visible.

Ansys Electronics Desktop Menus

The menu bar enables you to perform all Ansys Electronics Desktop tasks, such as managing project files, customizing the desktop, drawing objects, and setting and modifying project parameters.

Ansys Electronics Desktop contains the following menus, which appear at the top of the desktop:

File	Contains commands to manage Ansys Electronics Desktop project files and printing options.
Edit	Contains commands to modify the objects in the active model and undo and redo actions.
View	Contains commands to display or hide desktop components and model objects, modify 3D Modeler window visual settings, and modify the model view.
Project	Contains commands to add a specific design type to the active project; view and define datasets, project variables, and event callbacks.
Tools	Contains commands to modify the active project's material library, arrange the material libraries, run and record scripts, update project definitions from libraries, customize the desktop's toolbars, and modify many of the software's default settings.
Window	Contains commands to rearrange the 3D Modeler windows and toolbar icons.
Help	Contains commands to access the help system and view the current Ansys Electronics Desktop version information.

Once you have inserted a design type, the menu bar also includes menus specific to that type. These may include:

Draw	Contains commands to draw one-, two-, or three-dimensional objects, and sweep one- and two-dimensional objects.
Modeler	Contains commands to import, export, and copy 2D Modeler files and 3D Modeler files; assign materials to objects; manage the 3D Modeler window's grid settings; define a list of objects or faces of objects; control surface settings; perform boolean operations on objects; and set the units for the active design.
Ansys	Contains commands to set up and manage all the parameters for the active

Electronics Desktop	project. Most of these project properties also appear in the project tree.

Menu Bar

The **Menu Bar** appears at the top of the screen immediately beneath the Ansys Electronics Desktop title bar. It contains menus used for controlling the application and the various editors and viewers used to construct the models, set up the analyses, and postprocess the simulation results. The menu bar and the command ribbon contain many of the same commands and options. However, the menu bar provides the more comprehensive set of commands and options. The ribbon is more visual and generally requires fewer mouse clicks to navigate, but you can navigate the menu bar without using a mouse.

Click on a **Menu Bar** entry to open its menu. Alternatively, the **Alt** key makes it easy to access the menus without using a mouse, as follows:

- You can select a menu in the **Menu Bar** by pressing the **Alt** key and then pressing the underlined letter of the menu you wish to activate (the underlines appear when the **Alt** key is pressed). Background highlighting indicates which menu is selected. Finally, press **Enter** to open the menu.
- After pressing the **Alt** key, you can change the selected menu by pressing the left (⇐) and right (⇒) arrow keys. Again, press **Enter** to open the desired menu.

Once a menu is open, you can press additional letters to drill down within the menu. Alternatively, use the down (⇓) and up (⇑) arrows to change the selection. Use the right (⇒) arrow key to open a subordinate menu for the selected command. As always, press **Enter** to execute the selected command.

Order of Menus

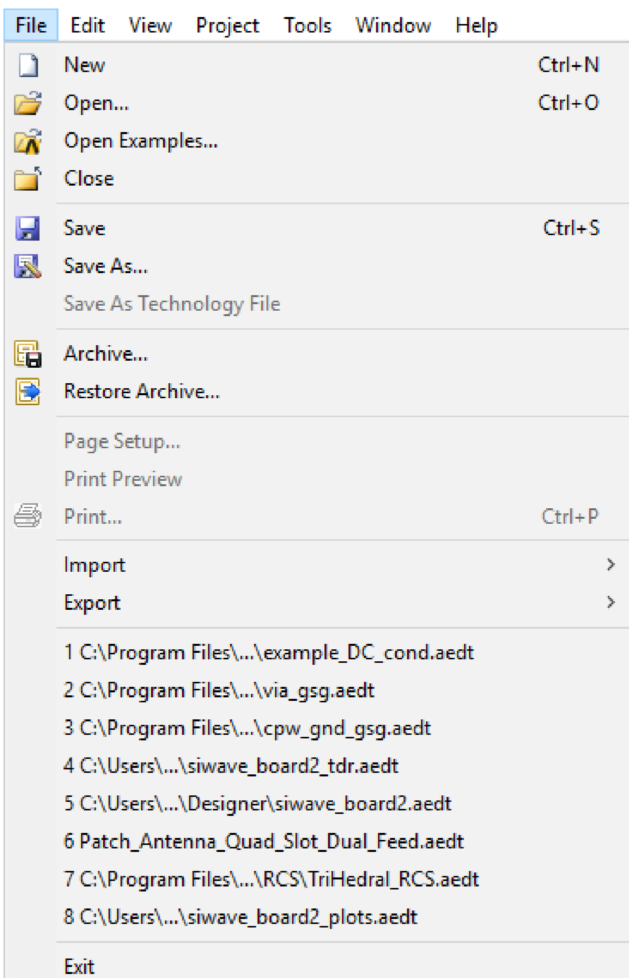
When a project is loaded, the Ansys Electronics Desktop drop-down menus (from left to right) are:

File Edit View Project Draw [Modeler | Layout | Schematic](1) *<Design-Type>*(2) Tools Window Help

> **Note:**
> 1. These three menu bar entries depend on the active design type:
>   - **Modeler** appears for all designs that use the *Modeler* window (including all 3D types, 2D Extractor, and Maxwell 2D).
>   - **Layout** appears only for HFSS 3D Layout designs.
>   - **Schematic** appears for Circuit and Maxwell Circuit designs.
> 2. The name of the active *Design-Type* appears in this position, and the menu contains the design-type-specific commands.

These menus are described in the following sections.

File Menu



File menu items are common operations on files and projects. You can search the Help to find information on any of the commands that appear on Ansys Electronics Desktop menus.

- Click **New** (or type **n**) to set up a new project.
- Click **Open** (or type **o**) to open an existing project.
- Click **Open Examples** to open an example project.
- Click **Close** (or type **c**) to close the selected project. If the project has changed since the last save, you will be prompted to save the project before closing.
- Click **Save** (or type **s**) to save the selected project.
- Click **Save As** (or type **a**) to save the selected project under a different name or in a different directory.
- Click **Save As Technology File** (or type **t**) to save the selected top-level design as a technology file (*.asty). You must have a design selected to activate this menu item. You cannot export a technology file from a project or from a subcircuit.
- Click **Archive** to archive the selected project and any selected additional files. If the project has changed since the last save, you will be prompted to save the project before closing.
- Click **Restore Archive** (or type **r**) to restore a previously archived project.
- Click **Page Setup** (or type **u**) to set up formatting to print the active window in the Design Area.
- Click **Print Preview** (or type **v**) to display a preview of the print job.
- Click **Print** (or type **p**) to print the active window.
- Click **Import** to open a submenu and select a file for import.
- Click **Export** to open a submenu and select a file for export.
- Click the name of a project (or type its number) to open a project from the listing of recently opened projects.
- Click **Exit** (or type **x**) to exit Ansys Electronics Desktop. If any project has unsaved changes, you will be prompted to save the project before closing.

Edit Menu

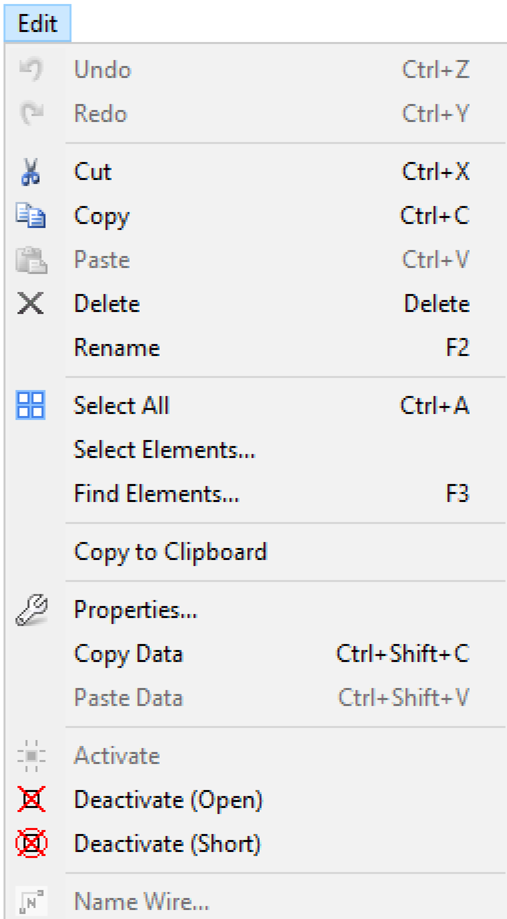
The **Edit** menu changes, depending on the active window in the Design Area.

These menus include:

- 3D Modeler

Edit Menu for Schematic Editor

When the Schematic Editor is active in the Design Area, the **Edit** menu appears similar to the following:



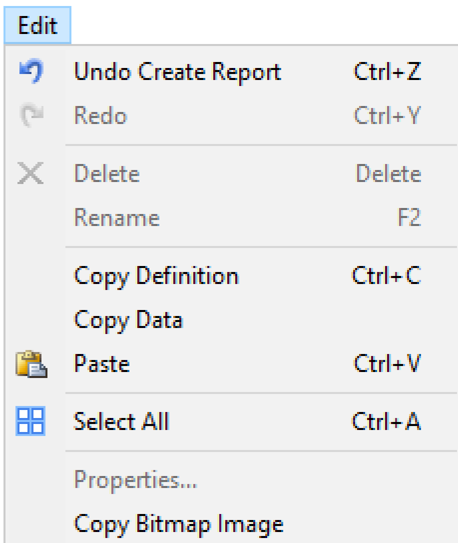
The Schematic Editor Edit menu contains the following options:

- **Undo** – allows you to undo the last action, which is specified after "Undo."
- **Redo** – allows you to redo an undone action, which is specified after "Redo."
- **Cut / Copy / Paste** – allows you to copy and paste schematic elements.
- **Delete** – deletes a selected element.
- **Rename** – renames a selected element.
- **Select All** – selects everything in the design area.
- **Select Elements** – allows you to select certain nets or components.
- **Find Elements** – opens the **Find Elements** window, where you can search for elements by property name and value.
- **Copy to Clipboard** – copies the visible design area to the clipboard in bitmap format, for pasting into another program or document.
- **Properties** – displays the relevant properties window for the selected object.
- **Copy Data** – copies properties data for the selected object.

- **Paste Data** – pastes properties data into a selected object.
- **Activate** – causes the activation of one or more selected components.
- **Deactivate (Open)** – temporarily converts the component into an open circuit.
- **Deactivate (Short)** – temporarily converts the component into a short circuit.
- **Name Wire** – allows you to rename a selected wire.

Edit Menu for Report Window

When the Report Window is the active window, the **Edit** menu appears similar to the following:



The Report Window Edit menu contains the following options:

- **Undo** – allows you to undo the last action, which is specified after "Undo."
- **Redo** – allows you to redo an undone action, which is specified after "Redo."
- **Delete** – deletes a selected report element.
- **Rename** – renames a selected report element.
- **Copy Definition** – copies the report definition.
- **Copy Data** – copies report data.
- **Paste** – pastes the copied information at your selected location.
- **Select All** – selects everything in the report window.
- **Properties** – opens the relevant Properties window for the selected report element.
- **Copy Bitmap Image** – copies the entire report window as a bitmap image that can be pasted into another program or document.

View Menu

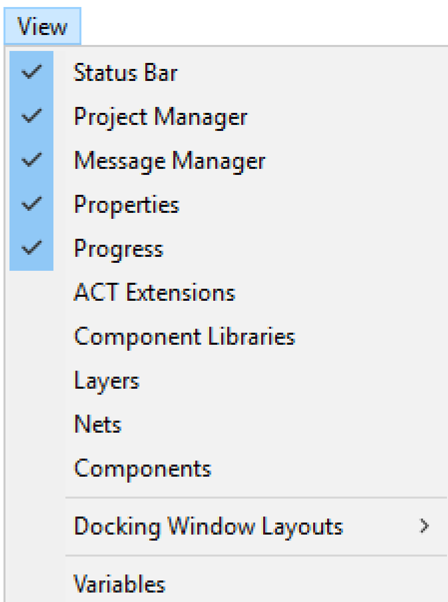
The **View** menu changes, depending on the active window in the Design Area.

These menus include:

- Basic View Menu

Basic View Menu

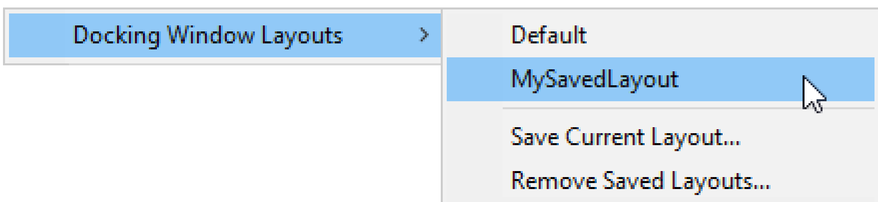
When no editor is open in the design area, or when the Netlist Editor or Report Window is active, the **View** menu appears similar to the following:



- The check boxes toggle the display of the **Status Bar**, **Message Manager Window**, **Project Manager**, **Properties Window**, and **Progress Window**. You can enable the display of **Component Libraries**, **Layers**, **Nets**, **Components** and **Variables** windows.
- The **ACT Extensions** command opens the *ACT Extensions* window, which lets you work with ACT integration tools. For more information, see the ACT Extensions Window section.

> **Note:**
>
> The **ACT Extensions** window and the design wizards it contains (*5G Wizard, HFSS Antenna Design Toolkit, HFSS-EMA3D Link,* and *Maxwell Eccentricities*) are only available for the Windows version of the Ansys Electronics Desktop software. These items are not available when the software is installed on a Linux platform.

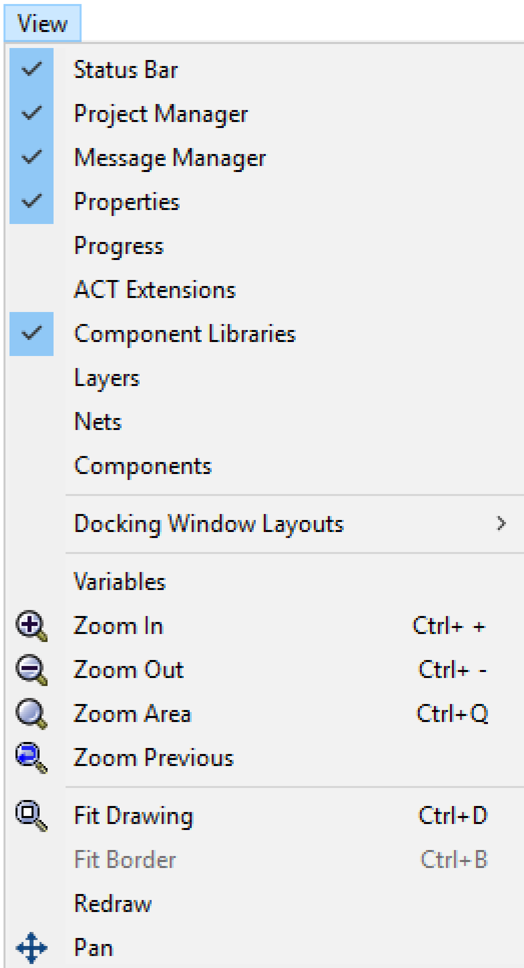
- The **Docking Window Layouts** submenu lets you select from the **Default** and any saved window layouts, **Save Current Layout**, or **Remove Saved Layouts** by selecting them from a list.



- The **Variables** option opens the **Project and Design Variables** window, allowing you to set variables.

Schematic Editor View Menu

When the Schematic Editor is active, the **View** menu appears similar to the following:

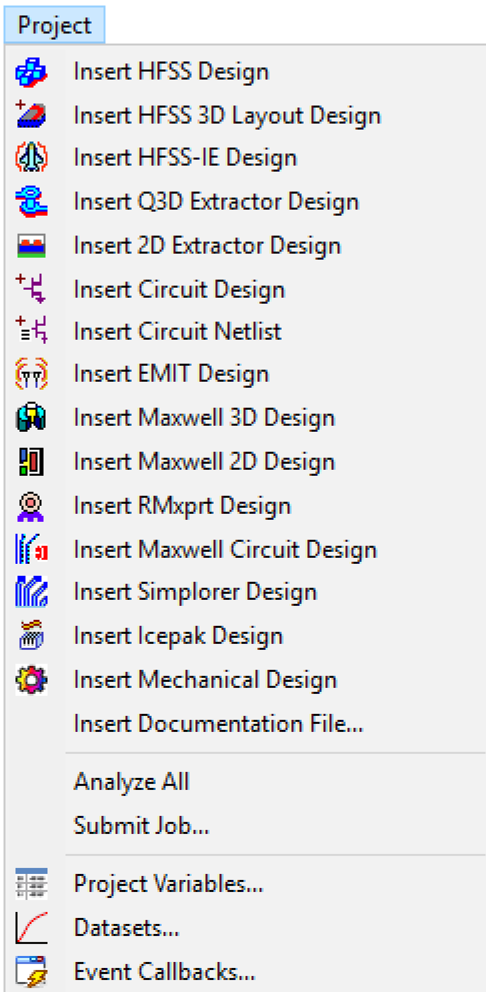


In addition to the options available on the Basic View Menu, the Schematic Editor View Menu contains the following:

- **Zoom In** / **Zoom Out** / **Zoom Area** / **Zoom Previous** – control the portion of the design displayed in the design area.
- **Fit Drawing** – changes the display so that the entire drawing fits to the edges of the design area.
- **Fit Border** – when a border has been added to a drawing, scales the bordered contents to fit within the bounds of the window.
- **Redraw** – redraws the current design.
- **Pan** – allows you to pan across the drawing.

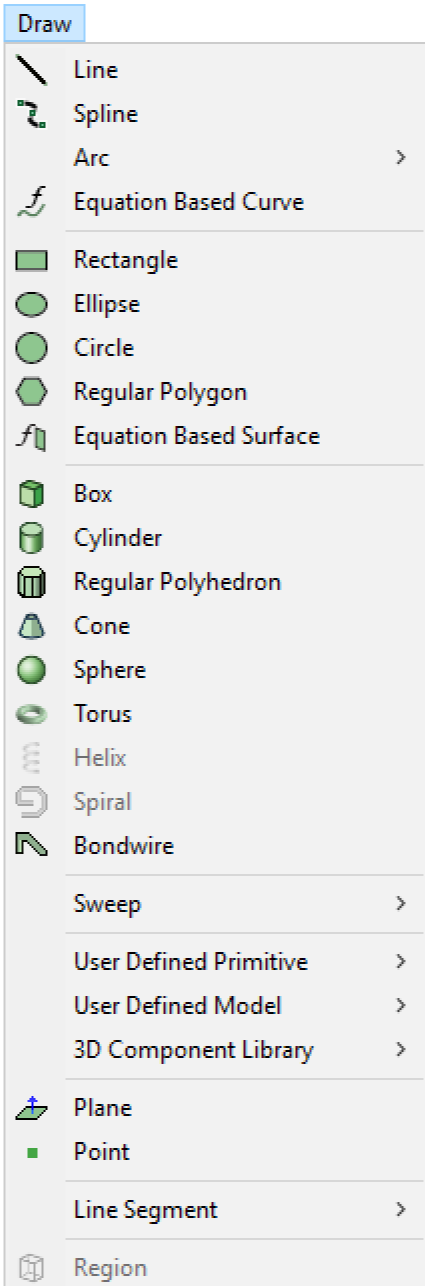
Project Menu

To create (insert) a new design, you must first open a new project folder by selecting **New** from the **File** menu. Then select a project to insert from the Project menu. Or you can click the icon corresponding to the type of design you wish to create.



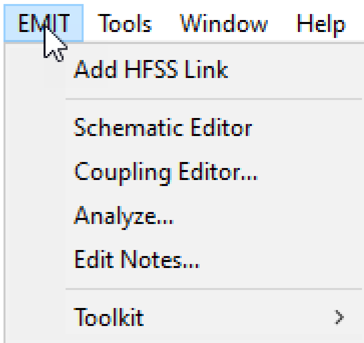
Layout Editor Draw Menu

The **Draw** menu for the **Layout Editor** is context sensitive and appears slightly different depending on the type of design that is loaded.



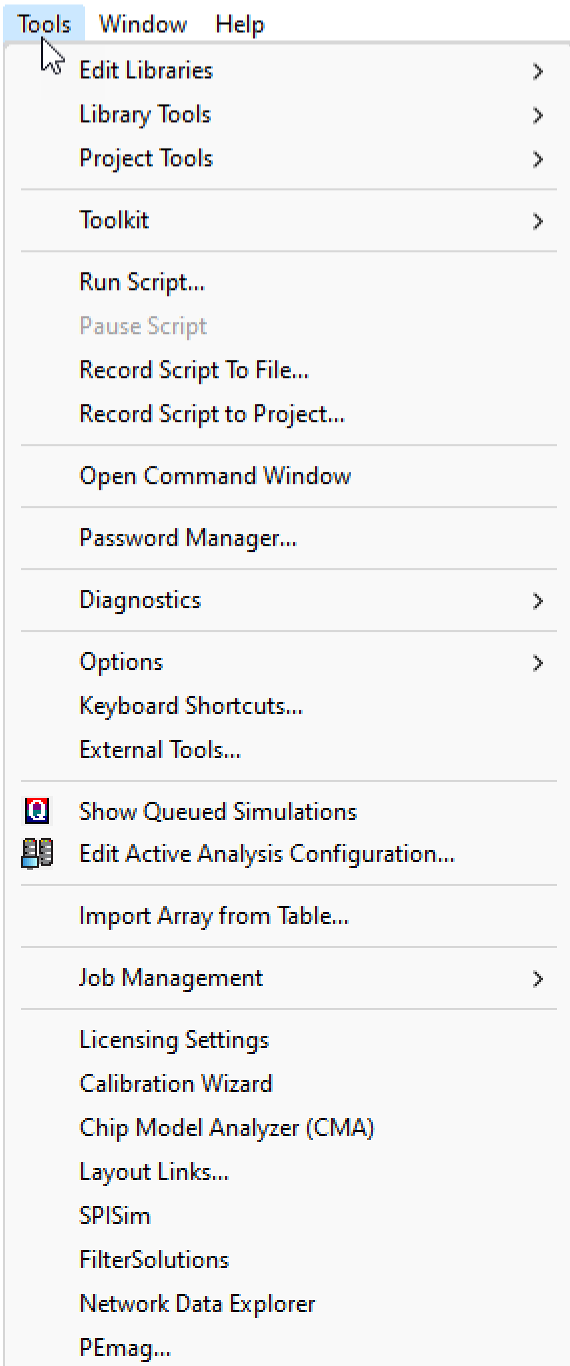
Editor and Design Specific Menus

The Top Menu bar contains editor and design menus that are specific to the editor or viewer that is active in the Design area. The specific menus correspond to the type of design that is inserted using the Project menu.

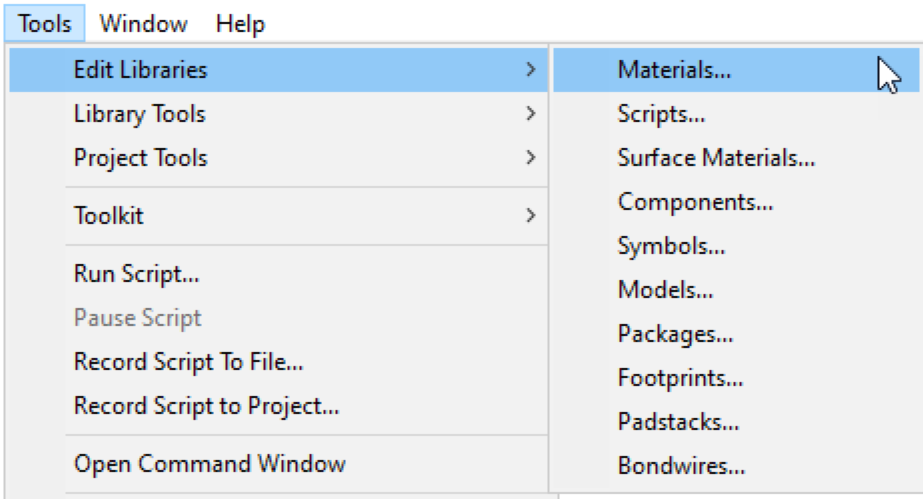


Tools Menu

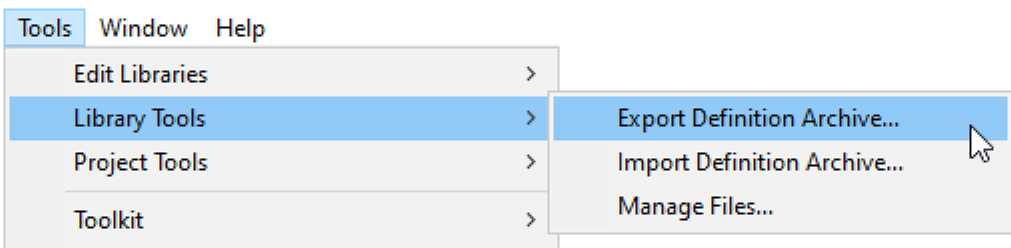
The **Tools** menu contains operations that are common to all design types:



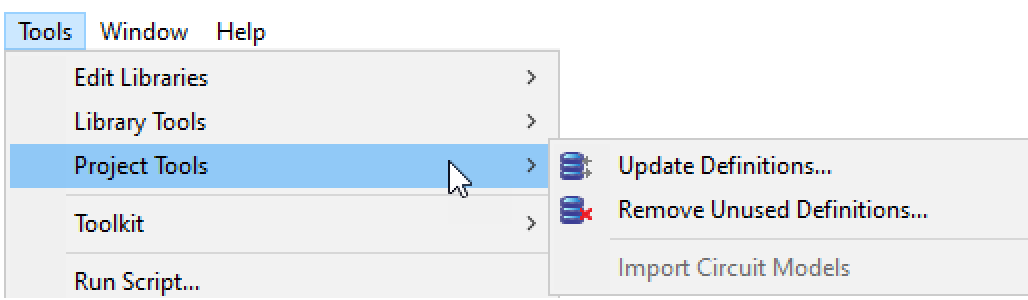
- Click **Tools** > **Edit Libraries** to view commands allowing access to the libraries available for the various Ansys Electronic Desktop solves, such as Materials, Scripts, and Components, Symbols and Models for Simplorer, Icepak, and Circuit.



- Click **Tools** > **Library Tools** to view tools for importing or exporting definition archives for UserLib and Syslib and to manage files relevant for various solver libraries.

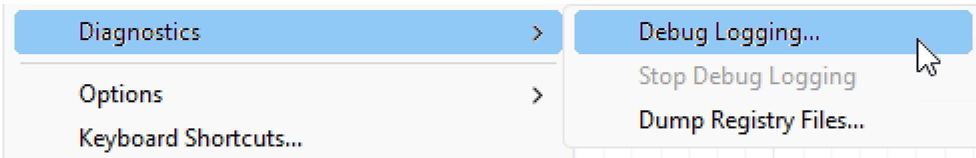


- Click **Tools** > **Project Tools** to manage definition libraries relevant to various solvers.

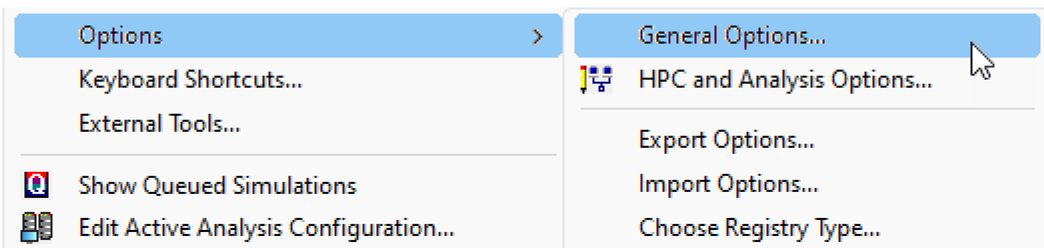


- Click **Tools** > **Toolkit** to install PyAEDT or update product menus to show any added Python-based toolkits.
- Click **Tools** > **Runscript...** to run scripts that you have recorded. A Pause command is available. You can also **Record Script to File...** or **Record Script to Project...**. See the Scripting Help for more information.
- Click **Tools** > **Open Command Window...** to open an IronPython command window. See the Scripting Help for more information.

- Click **Tools** > **Password Manager...** to open the *Password Manager* window to control access to resources.
- Click **Tools** > **Diagnostics** to access the following diagnostics commands:



  - Click **Debug Logging** to enable debug logging via an environment variable.
  - Click **Stop Debug Logging** to end debug logging. This option is only available while debug logging is active.
  - Click **Dump Registry Files** to make a copy of all user, administrator, and default-settings configuration files, placing them in the folder you select.
- Click **Tools** > **Options** to set General Options and HPC options.

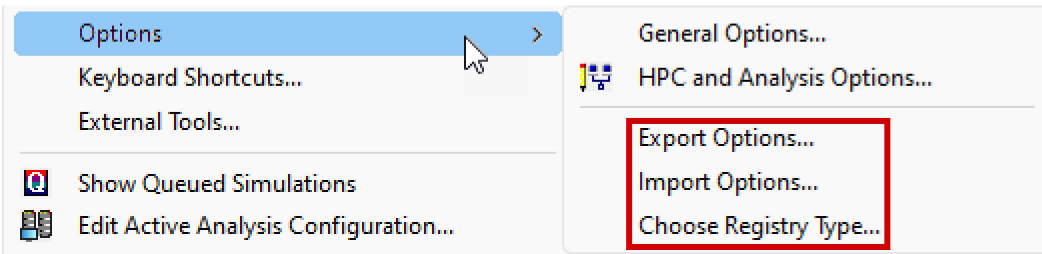


  Other available commands in this fly-out menu are **Export Options**, **Import Options**, and **Choose Registry Type**. See Understanding Registry Tools for more information on these three commands.

- Click **Tools** > **Keboard Shortcuts...** to manage your keyboard shortcut behavior.
- Click **Tools** > **External Tools...** to manage menu access to external tools.
- Click **Tools** > **Show Queued Simulations** to view a dialog listing simulations in a queue and providing tools to manage the queue.
- Click **Tools** > **Edit Active Analysis Configuration** to view a dialog that lets you view and manage the current configuration.
- Click **Tools** > **Import Array from Table** to import an array definition from a .csv file.
- Click **Tools** > **Job Management...** to select a scheduler, and to submit and monitor remote simulations.
- Click **Tools** > **License Settings** to launch the Licensing Settings tool.
- Additional Tools listed include Calibration Wizard, and Network Analyzer, and other tools used to help you with Ansys Electronics Desktop projects and solvers.

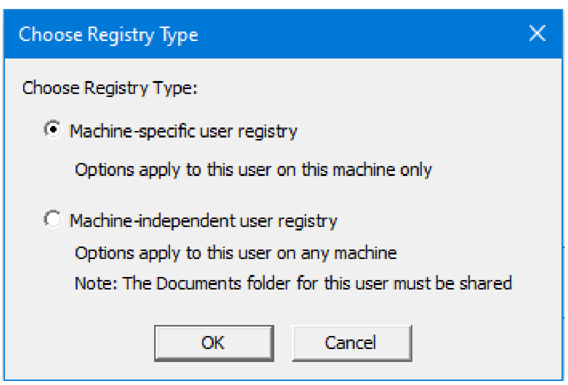
Understanding Registry Tools

In the context of this topic, when we say *"Registry"* we are referring to the files where the various Ansys Electronics Desktop application options are stored. We are not referring to the Windows registry. Specifically, this topic discusses the **Export Options**, **Import Options**, and **Choose Registry Type** commands that are listed in the fly-out menu that appears when you click **Tools** > **Options** from the menu bar.



Choose Registry Type:

Click **Choose Registry Type** to switch between two available options. The following dialog box appears:



The choices are summarized as follows:

- **Machine-specific user registry:** This is the default type. You must set up the configuration manually for each user account on each machine where the Ansys Electronics Desktop software is run. You can do so by exporting and importing options (see below), setting the options in the user interface on each machine, or copying configuration files between machines and renaming them. The hostname of the computer is part of the filename for machine-specific registries.
- **Machine-independent user registry:** Allows you to share a single user configuration file among multiple machines. This choice has the advantage of keeping the options synchronized between all the user's machines.

> **Note:**
>
> The term "host" is frequently used to refer to any computer used to run a particular application. In the Ansys Electronics Desktop Help, the terms *"host-specific"* and *"machine-specific"* are used interchangeably, as are *"host-independent"* and *"machine-independent."*

When you switch registry types, the settings from the currently active registry are copied into the new registry, which then becomes the active one. The filenames for user registries are as follows:

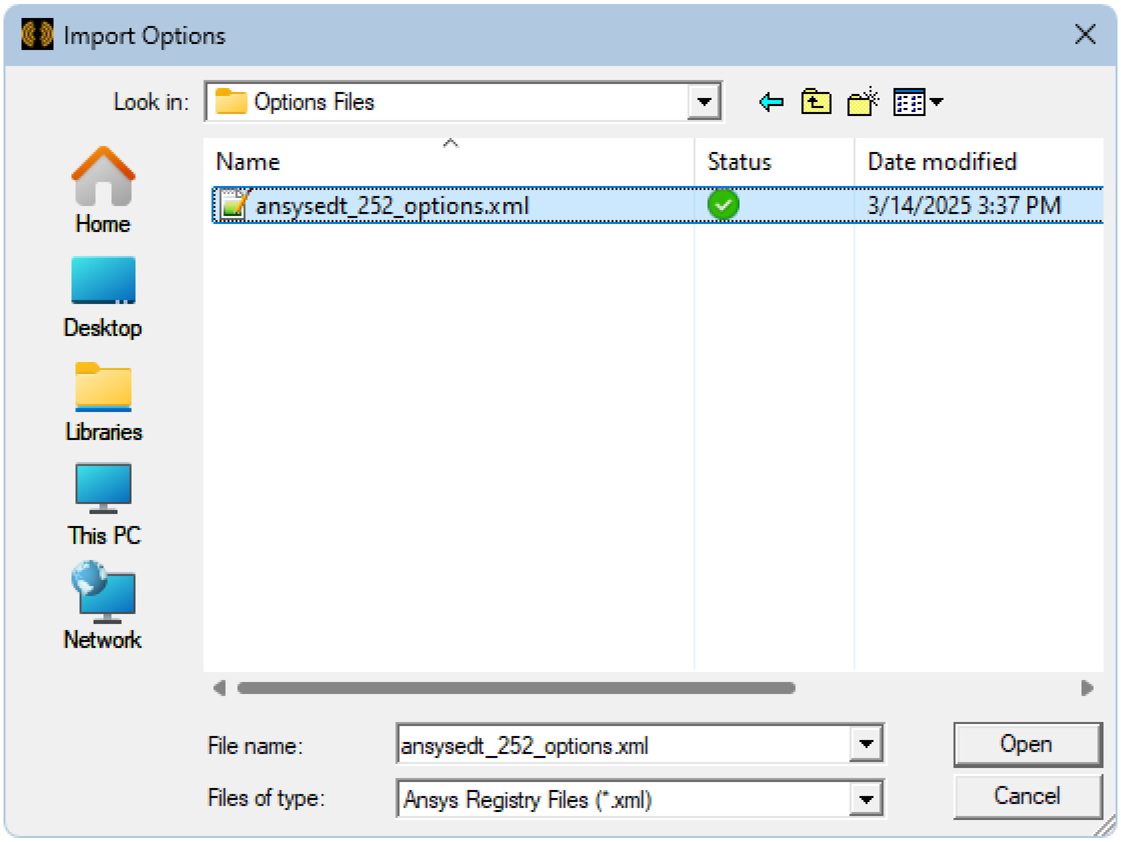
- Machine-specific: **<*hostname*>_user.xml**
- Machine-independent: **user.xml**

> **Note:**
>
> For a machine-independent user registry to work, the *Documents* folder must be shared by all machines on which you run the software, and you must log into the same account on each machine. The user registry file is stored in the path, ...Documents\Ansoft\ElectronicsDesktop20*yy.m*, where *yy* is the last two digits of the version year, and *m* is the minor release number. The Documents folder must be mapped to a network folder accessible to all machines or to an online folder (such as OneDrive, for example). Otherwise, you would have to copy the registry file to each machine, and they would *not* remain synchronized, defeating the purpose of the machine-independent registry.

Exporting and Importing Options:

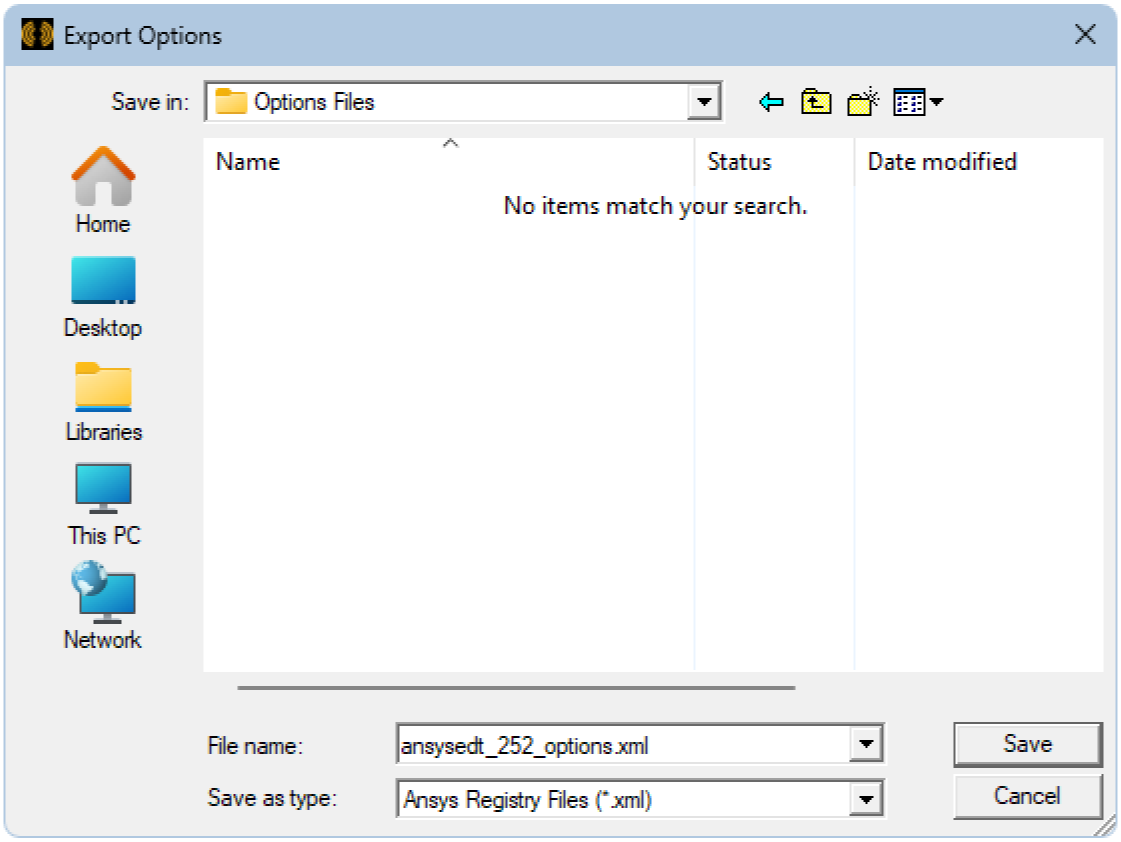
You can export options to an XML file, which can then be imported into the registry. This capability is useful if you want to use one set of options for a given situation and a different set for another. Simply export the options for each different configuration you use. Then, import the file that is applicable to your current work. You can also use this capability to transfer options to a different user or machine.

When you click **Import Options**, the following dialog box appears, in which you can navigate to the option file folder and select the desired file to import:



When you import an options file, the currently active registry type (specified using the **Choose Registry Type** command) determines to which registry file the options are written. After importing an options file, you will be prompted to restart the application, which is recommended.

When you click **Export Options**, the following dialog box appears, in which you navigate to the folder where you want to place the file and specify the filename:



Options are always exported from the currently active registry. The usual precedence rules apply when getting the value to export. For example, if the current registry being exported is machine-specific, and a particular value is missing, the value in the machine-independent registry will be used (if available). See the **Note** at the bottom of Setting Options via Configuration Files for more information on the order of precedence.

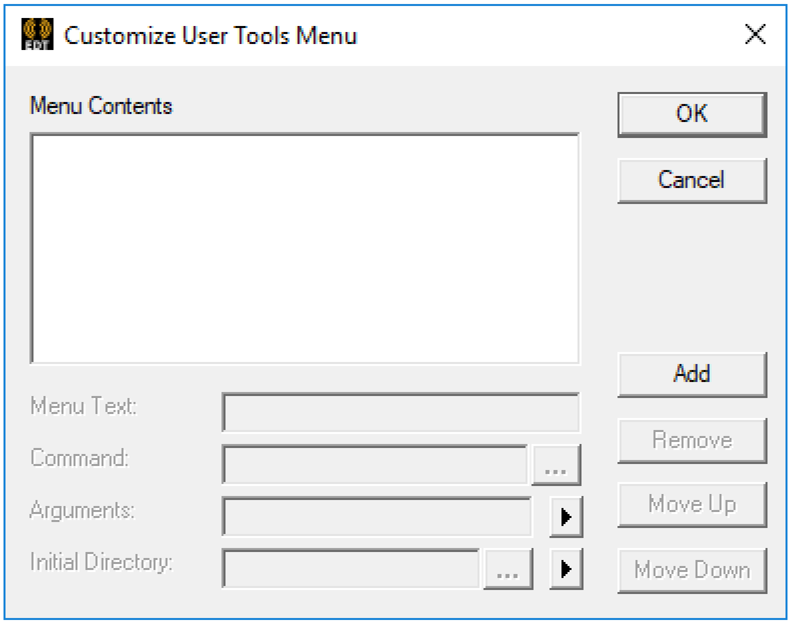
> **Note:**
>
> An exported user options file contains only the settings that are exposed in the general options (that is, the *Options* dialog box) and view options (*3D UI Options* dialog box). So the exported file contains a subset of the configuration settings. HPC and other settings (such as window locations and window size data) are *not* included in the exported options.

Adding External Tools to the Tools Menu

To add an executable to the Tools menu:

1. Click **Tools** > **External Tools**.
2. This displays the **Customize User Tools Menu** dialog box.

   If a **User Tools** menu item has been defined, its contents are displayed. Command buttons let you Add new commands and Delete selected commands, and Move Up and Move Down commands. You can specify the command line arguments to the program and the directory from which it will run.



3. To add a custom Tools menu entry, click **Add**.

   This enables the following fields:

   Menu Text field – displays [new tool] as text you will replace with the text you want to appear in the User Tools menu.

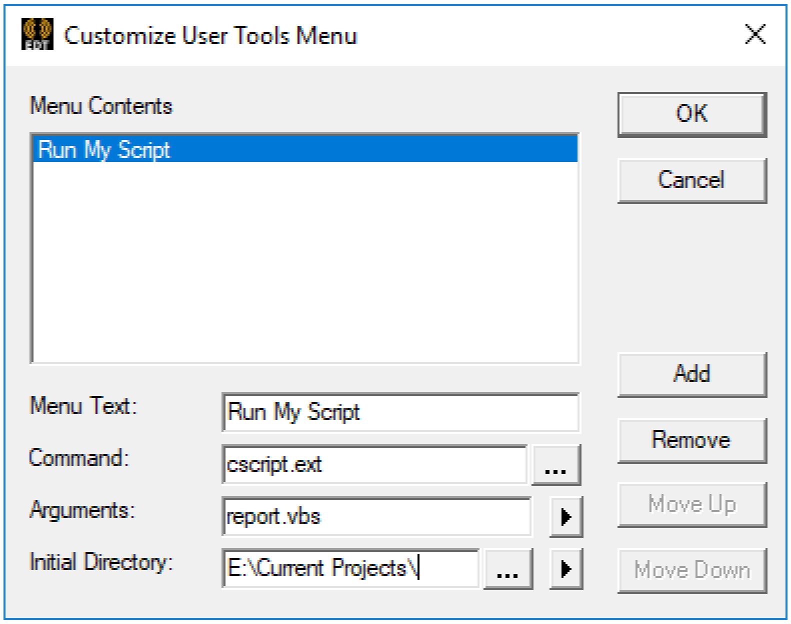
   Command field – displays the external executable. An ellipsis button [...] lets you navigate to the file location.

   Arguments field – accepts command arguments from the **>** button menu selections for File Path, File Directory, File Name, File Extension, Project Directory, or Temp Directory.

   Initial Directory – specifies the initial directory for the command to operate. The ellipsis button {...] displays a dialog that lets you navigate folders in your desktop, or across the network.

4. Click **OK** to add the External Tools menu to EMIT or Cancel to close the dialog without changes.

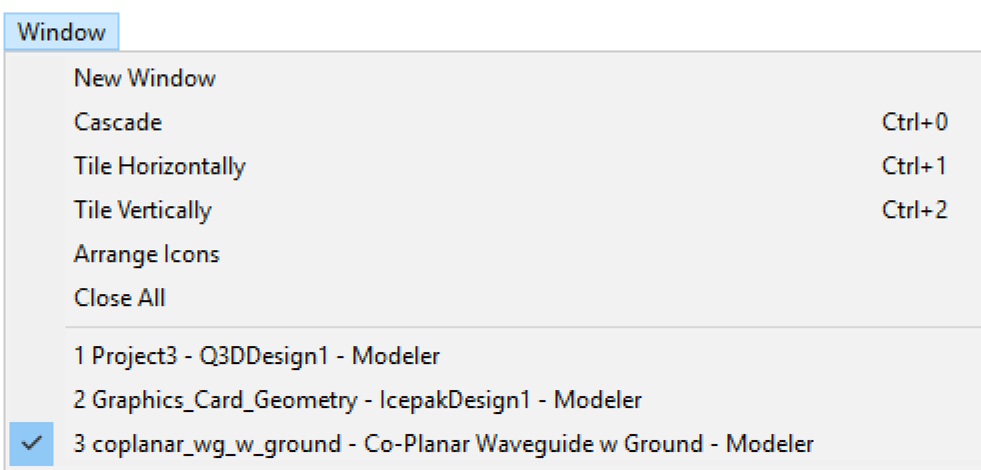
   You can also add scripts to the **Tools** menu. Assuming you have a script to generate custom reports called report.vbs, use the cscript.exe program to execute your script.



This example shows the cscript.exe program added to the **Tools** menu as **Run My Script**. The command line argument to the csscript.exe program is report.vbs. You can also name the directory in which it will be run.

Window Menu

The **Window** menu contains common window control operations, as well as a list of projects that are currently open.



- Click **New Window** (or type **n**) to open a new window in the Design Area. The new window will show the active design.

- Click **Cascade** (or type **c**) to arrange the open design windows in overlapping sequence:

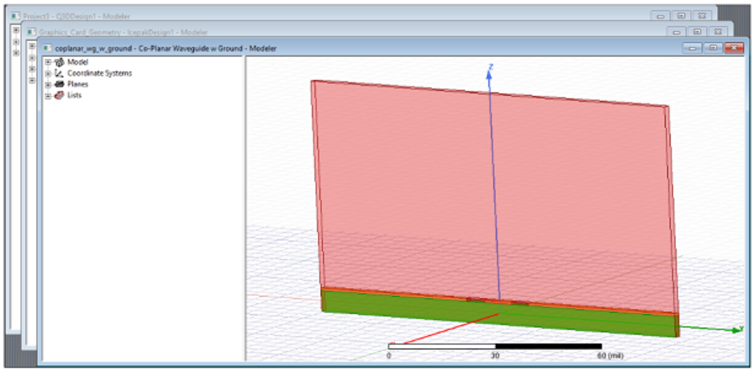
  You can use the **Cascade** operation to restore all windows to their default sizes after one or more of them have been maximized (enlarged to fill the entire Design Area).

- Click **Tile Horizontally** (or type **h**) to arrange the open design windows in a top-to-bottom sequence:
- Click **Tile Vertically** (or type **v**) to arrange the open design windows in a side-to-side sequence:

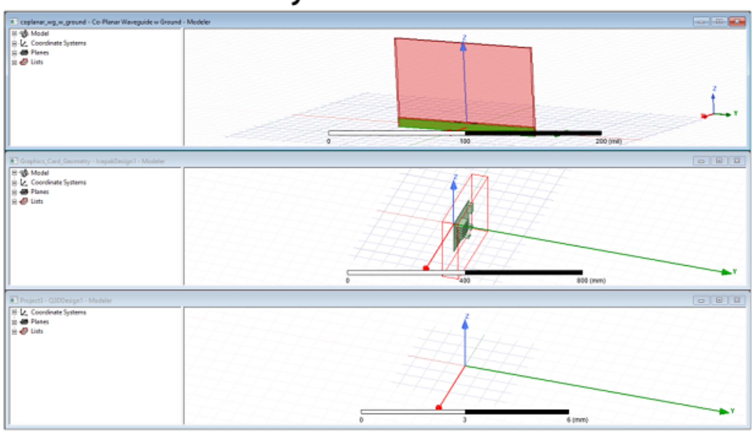
> **Tip:**
>
> You can use the shortcut key sequences Ctrl+0, Ctrl+1, and Ctrl+2 to execute the Cascade, Tile Horizontal, and Tile Vertical operations, respectively. The shortcuts can be used at any time, bypassing the Window menu.

- Click **Arrange Icons** (or type **a**) to restore iconized windows to a neat row at the bottom of the Design Area, after the icons have been repositioned manually.
- Click **Close All** (or type **l**) to close all the editor windows in the Design Area.

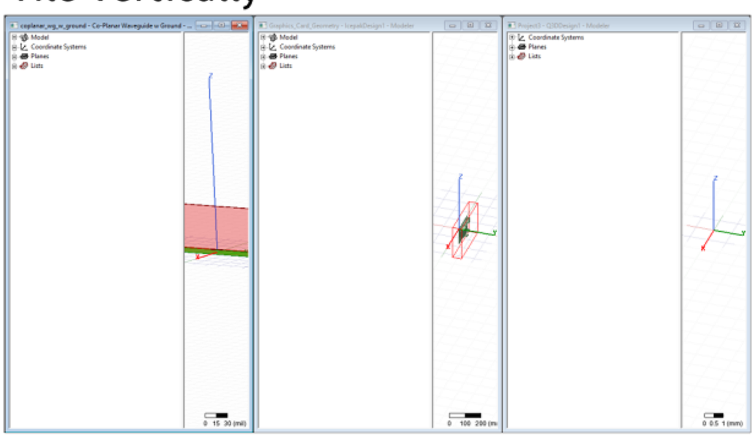
Cascade



Tile Horizontally



Tile Vertically



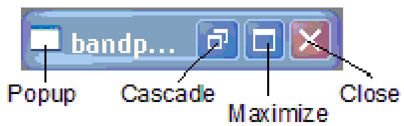
Working with Editor Windows

Each editor window in the Design Area has size controls in its top bar:

Clicking the white rectangle on the left opens a pop-up:

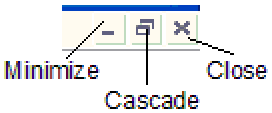
The **Move** and **Size** operations are not needed. To move an editor window, drag it by the top bar into the desired location. To resize an editor window, position the cursor at an edge or corner, and drag the border in or out.

- Click **Minimize** (or use the Minimize icon) to collapse the design window into an icon:



- Clicking the white rectangle on an iconized window opens the same pop-up as the one shown above. To restore an iconized window to its previous size (cascaded or maximized), click the white rectangle, then select **Restore** on the pop-up. Alternatively, use the Cascade or Maximize buttons on the icon to restore the window to the desired configuration. Use the Close button to close the iconized editor window. Reposition the icons anywhere in the Design area by dragging them with the left mouse button. To restore the icons to a neat row at the bottom of the Design Area, use the **Arrange Icons** command on the Window pulldown.
- To relocate an iconized window in the Design Area, drag it with the left mouse button.

- Click **Maximize** (or use the Maximize icon) to enlarge the editor window to fill the entire Design Area. In this configuration, Minimize, Cascade, and Close icons appear at the upper right of the Ansys Electronics Desktop, on the same level as the Top Menu bar:



- To restore a maximized window to its initial size and top bar, use the **Cascade** operation.
- Click **Close** (or use the Close icon) to close an editor window.
- Click **Next** to move the active focus to another editor window.
- See Design Area for a summary of the available editor windows.

Help Menu

Ansys Electronics Desktop features extensive help documentation, including text search and context-sensitive help items. To access the main help system, click **Help** on the top menu bar.

The Help Menu displays different selections depending upon the type of design inserted to the active project. The basic selections for EMIT are:

> **Note:** Ansys Electronics Desktop Student includes access to PDF documentation only.

- **EMIT Help** – opens the EMIT help within the Electronics help system. You can also access PDF versions from within the help system.
- **EMIT Scripting Help** – opens the EMIT scripting help.
- **EMIT PDFs** – provides access to PDFs for EMIT, including the main help, scripting guide, and Getting Started Guides.
- **Ansys Customer Support** – opens a browser page to the Ansys Customer Portal. At the website you can learn more about Ansys products and services and log on to contact Ansys technical support staff.
- **What's New in this Release** – opens a PDF that describes *What's New in Ansys Electronics Desktop* for 2025 R2.
- **Ansys Product Improvement Program** – opens a window describing the Product Improvement Program option.
- **License Settings** – opens help for the License Settings tool.
- **Ansys Engineering Copilot** – opens a feature for learning and support within select Ansys products. This feature displays in an independent window within supported applications. It retrieves content from various sources, including the Ansys Innovation

Space (AIS), Ansys documentation and learning resources, and the AnsysGPT™ virtual AI assistant.

- **Home** – includes a search function that will search the courses, knowledge base, and learning forum for information relevant to the search term you have entered (searches the entire Ansys Innovation Space (AIS)). The Search AIS field also provides filters to narrow your search in a particular information category; however, this feature does not automatically filter based on physics type or the product you are using.
- **Ansys Innovation Courses** – the Search function can be used to access a wide range of Ansys Electronics Engineering courses that include on-demand, self-paced video training and quizzes.
- **Ansys Knowledge** – provides access to Ansys expert knowledge articles. The search filter is preset based on the application you are using, but you can change the filter as desired.
- **Ansys Learning Forum** – provides access to discussions and presentations from Ansys experts, partners and customers. Users can also post questions to the forum, which is moderated by Ansys employees.
- **AnsysGPT** – is a multilingual Ansys Artificial Intelligence (AI) Virtual Assistant. The AnsysGPT™ assistant takes your query and produces answers based on internal and external expert knowledge of the product. It draws upon information from Ansys Knowledge articles, Ansys Innovation Courses, Ansys Learning Hub courses, the Ansys Learning Forum, Ansys documentation, Ansys.com, and the official Ansys YouTube channels, as well as other Ansys information sources.

To use the Copilot tool:

- You must have a current AnsysGPT™ AI+ license. Ansys Engineering Copilot is a licensed product at no additional cost.
- You must be registered on Ansys Customer Support Space (ACSS).

For more information on Ansys Engineering Copilot, see the Ansys Engineering Copilot User Guide.

- **Ansys Learning Hub** – opens a web page with subscription-based access to virtual and self-paced learning across the Ansys Software portfolio.
- **About Ansys Electronics Desktop** – opens a dialog box that displays the Ansys Electromagnetics Suite release number and contains tabs that show information about the **Installed Components** and **Client License Settings**.

Context-Sensitive Help

To access context-sensitive help from the Ansys Electronics Desktop user interface, press **F1** while your cursor is on an item. The help system specific to the product opens.

Shortcut Menus

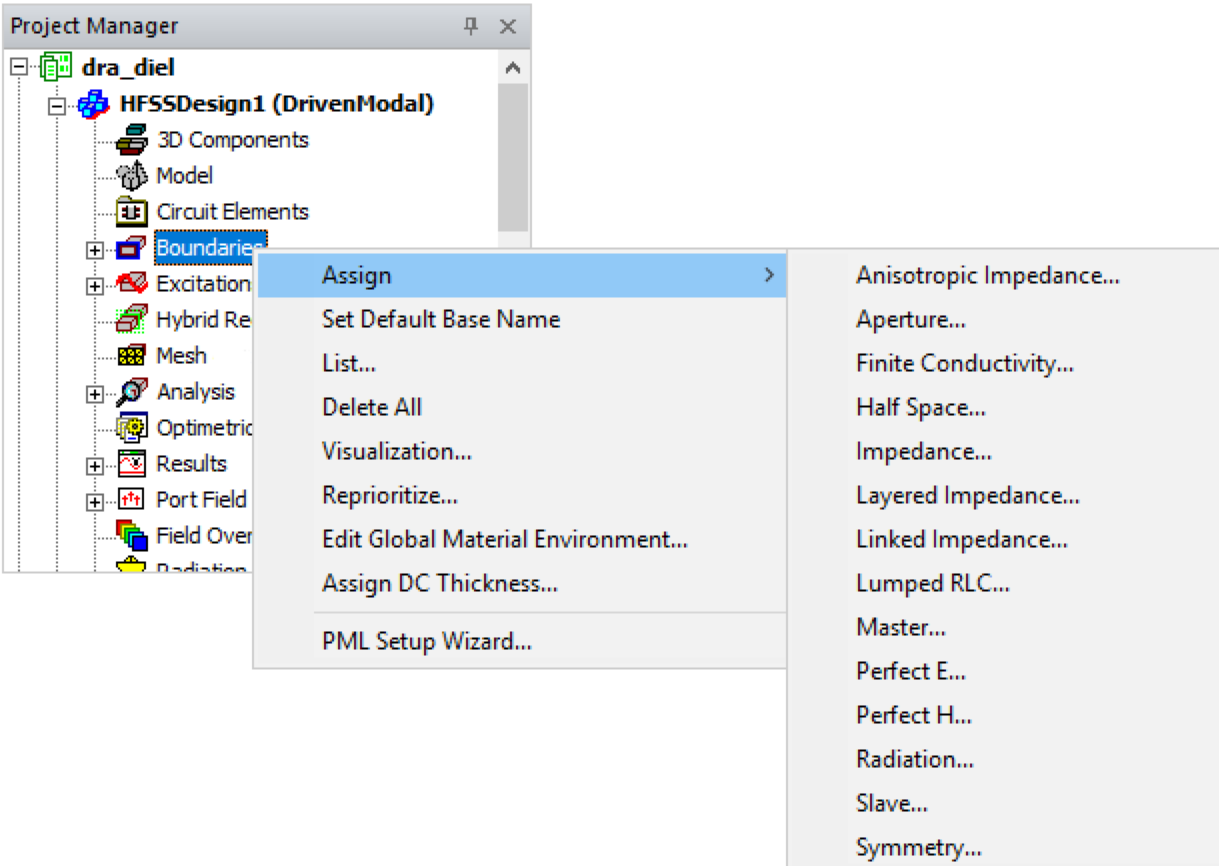
A variety of shortcut menus — menus that appear when you right-click a selection or in a window — are available in the **3D Modeler** window, in the **Project Manager** window, in the **History Tree**, and in the **Progress** window.

Shortcut menu in the 3D Modeler window	Use the shortcut menu in the **3D Modeler** window to select, magnify, and move objects (zoom, rotate, etc.); change the view; perform boolean operations; assign materials, boundaries, excitations, or mesh operations to objects; and work with field overlays.
Shortcut menus in the Project Manager window	Use the shortcut menus in the **Project Manager** window to manage Ansys Electronics Desktop project and design files and design properties; assign and edit boundaries, excitations, and mesh operations; add, analyze, and manage solution setups; add Optimetrics analyses; create post-processing reports; insert far- and near-field radiation setups; edit project definitions; and run Maxwell SPICE.
Shortcut menus in the History Tree	Use the shortcut menus in the **History** tree to expand or collapse groupings. If you select particular objects in the history tree, the shortcut menu lists the commands that you can apply to the selected object (s).
Shortcut menus in the Progress window	Use the shortcut menus in the **Progress** window during a simulation to **Abort** or **Clean Stop**.

> **Note:**
>
> All commands available on shortcut menus are also available from the menu bar.

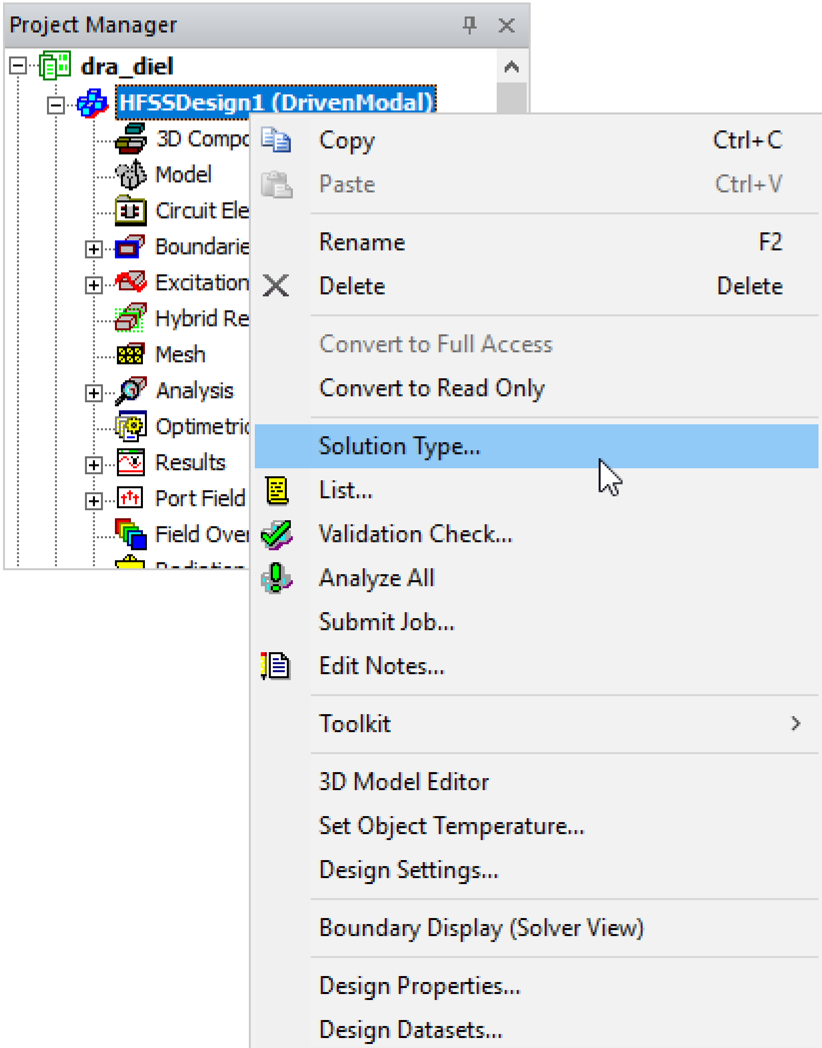
Shortcut Menus in the Project Manager Window

Each node, or item, in the project tree has a shortcut menu. For example, from the shortcut menu for the **Boundaries** icon, you can assign boundaries to selected objects; review information for all the boundary assignments for the active design; remove all boundary assignments; show or hide a boundary's geometry, name, or vectors; change the priority of a previously assigned boundary; and use the PML Setup wizard to create a perfectly matched layer (PML) boundary.



Other nodes have shortcut menus appropriate for the context.

For example, the following figure shows the shortcut menu when you right-click the design.



Customizing Ansys Electronics Desktop Menus

Ansys Electronics Desktop comes with a Menu Bar. You can modify the menu bar by using customized User Interface (UI) setups to add, remove, rename, and relocate commands. You can also add commands that execute external scripts or configure right-click shortcut menus for folders and items in the Project Tree, which allows for extensive customization of the Ansys Electronics Desktop Menus.

Customized setups are implemented using named subfolders that contain XML files used to configure customized menu settings. The subfolders included with the Ansys Electronics Desktop application are:

- config/UI/ElectronicsDesktop/EM
- config/UI/ElectronicsDesktop/RF

- config/UI/ElectronicsDesktop/RF.0
- config/UI/ElectronicsDesktop/SI
- config/UI/ElectronicsDesktop/SI1.0
- config/UI/ElectronicsDesktop/SI2.0
- config/UI/ElectronicsDesktop/Twin Builder

Each customized UI type has its own subfolder. Within these subfolders, XML files hold the menu configurations as well as the right-click menus for that UI type.

Select Different UI Types

To change the menu display from the default UI to a different UI type:

- Click **Tools** > **Options** > **General Options** > **General** > **Desktop Configuration**.
- Choose the UI type that you wish to use from the **Custom Menu Set** drop-down menu and click **OK**.

To switch from a customized UI type to the default UI:

- Click **Tools** > **Revert To Default UI**.

Add a New Customized UI Type

A new folder needs to be added to the config/UI/ElectronicsDesktop folder for any new customized UI type. All XML files for this UI type must go in this folder. The new UI type will appear in the **Custom Menu Set** drop-down menu (**Tools** > **Options** > **General Options** > **General** > **Desktop Configuration**).

For any UI type, Ansys Electronics Desktop displays the default UI menus for any products/contexts that are not in the xml files. If there is error processing any XML file, the default UI menu is displayed for that product. Check the message window for the names of problematic XML files and suggestions on how to fix them.

XML files are only processed once when you first switch to that UI type. After you make changes to any XML file, in order for it to be reprocessed, navigate to **Tools** > **Options** > **General Options** > **General** > **Desktop Configuration** and reselect the UI type from the **Custom Menu Set** drop-down menu. Then click **OK**.

Names of the XML Files

Below is the list of XML files that can be placed in a folder for a new UI Type. If any of these XML files does not exist in this folder, the default menu setting displays for that product.

- 2D Extractor.xml – used for 2D Extractor projects.
- Circuit Design.xml – used for Circuit Design projects.
- Circuit Netlist.xml – used for Circuit Netlist projects.
- EMIT.xml – used for EMIT projects.
- HFSS 3D Layout Design.xml – used for HFSS 3D Layout Design projects.

- HFSS.xml – used for HFSS projects.
- HFSS-IE.xml – used for HFSS-IE projects.
- Icepak.xml – used for Icepak projects.
- Maxwell 2D.xml – used for Maxwell 2D projects.
- Maxwell 3D.xml – used for Maxwell 3D projects.
- Maxwell Circuit.xml – used for Maxwell Circuit projects.
- NoDesignUI.xml – used for menu settings when no project is selected.
- Q3D Extractor.xml – used for Q3D projects.
- RightClickMenu.xml – used for all right-click menu settings.
- Twin Builder.xml – used for Twin Builder projects.

Valid XML Elements and Attributes

1. Root Element is **DesignerMenu**:
    - **xmlns** – required attribute which needs to be set to the following:

      <DesignerMenu xmlns="http://www.ansys.com/uiConfigMenu">
    - **UseProjectWindowSelectionContext** – child of DesignerMenu, appears zero or one time.

      > **Note:**
      >
      > This is only meaningful if it is used in NoDesignUI.xml. If this is set to true, then clicking in the Project Window on the Project icon or the Definitions icon (or a subitem) will show "Project" context in NoDesignUI.xml instead of active design context. No setting or setting this to false makes this UI type behaves the same as the default behavior of Ansys Electronics Desktop. The menus and toolbars are always shown for the active design unless there's no design at all.
    - **Context** – child of DesignerMenu. It appears at least one time and has a required "name" attribute. For details on setting the context name, see the "Context Name" section below.
2. Child elements of **Context**:
    - **TopMenu** – For RightClickMenu.xml, do not use this element. You can specify child elements of TopMenu listed below under Context. For any other xml files, you need at least one TopMenu child element for any Context.

      Child elements of TopMenu:

- **MenuName** – required, appears only one time.
- **popupMenu** – may appear one or multiple times; child elements are the same as those of Topmenu.
- **LeafMenu** – may appear one or multiple times.

  Child elements of **LeafMenu**:

  - **MenuName** – optional string.

    Pay attention to the character reference "&" in XML. In order for the name of the menu to be displayed as “Tools”, the XML syntax must be:

    <MenuName>&Tool</MenuName>

    “&“ is the character reference for ”&” in XML, while “&Tool” tells Ansys Electronics Desktop that Alt+T is the shortcut key for this menu.

  - **MenuID** – optional number.

    For a list of valid MenuIDs, please see ***Command IDs for Customizing AEDT Menus.xlsx*** under <*install_dir*>/v<*release_ number*>/[Win64 or Linux64]/Help.

  - **ShowBitMap** – optional, “Yes” or “No” (default)
  - **Accelerator** – optional string (Example: Ctrl+N)

  > **Note:**
  >
  > To add a new LeafMenu, both “MenuName” and “MenuID” are required.

- **CustomMenu** – may appear one or multiple times; used to add a customized menu to run an external script (vbs or python).

  Child elements of **CustomMenu**:

  - **MenuName** – see LeafMenu for Usage
  - **ShowBitmap** – see LeafMenu for Usage
  - **Accelerator** – see LeafMenu for Usage
  - **ScriptPath** – required string; used to supply path to the script (use of $PROJECTDIR, $PERSONALLIB, $USERLIB, $SYSLIB variables is also allowed).

    Example:

    <ScriptPath>C:/Users/jwei/Python/HelloWorld.vbs</ScriptPath>

or

<ScriptPath>$PERSONALLIB/HelloWorld.py</ScriptPath>

> **Note:**
>
> To add a new CustomMenu, both MenuName and ScriptPath are required.

- **Separator** – may appear one or multiple times; has no child element.

3. All elements have an optional attribute of **action**.

   Valid values are:

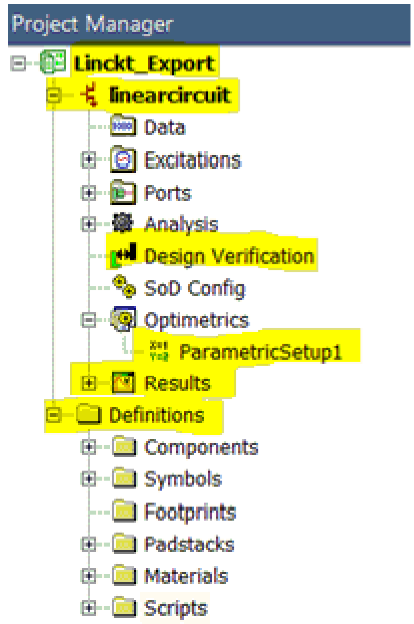
   - **add** – can be applied to all elements other than “DesignerMenu” or “Context” . This element and all its child elements will be added to its parent. An optional attribute of “position” can be followed here to specify the position of this newly added menu. Position starts from 1. If no position is specified, this menu will be appended to its parent menu.
   - **useDefault** – default menus will be used for this menu(matched by MenuName or MenuId). Any of it’s child menu will be processed according to its action setting.
   - **delete** – can be applied to all elements other than “DesignerMenu” or “Context” . This menu (matched by MenuName or MenuID) will be deleted from its parent.

> **Note:**
>
> - For “DesignerMenu” and “Context”, only no action setting or “useDefault” is valid.
> - The “DesignerMenu” action setting is significant because it is how you specify whether you want to construct your own menus or modify existing default menus. If this is set to “useDefault”, default menu settings are used for any menus not listed in your XML files. If no action is set, only the “File”, “Window” and “Help” menus display; no other default menus display. Menus listed in the XML file display after the “File” menu and before the “Window” menu. In this case, no action is needed for any other elements in the XML file. If any action is specified for any element, you will receive an error message and a default menu will be displayed for that context. A “Revert To Default” menu will be appended to the “Tools” menu. If no “Tools” menu is specified as a TopMenu, a “Tools” menu is created before the “Window” menu with one “Revert To Default” menu item.
> - For any other elements, a missing action setting means that the parent action will be used.
> - If any element’s action is set to “add”, then no action is needed for any of its child elements. If any action is specified, it will be ignored. This menu and all of its child menus will be added to the default menu.
> - If any TopMenu or pop-upMenu’s action is set to “delete”, you don’t need to list any of its child menus. The menu will be deleted with all child menus.

Context Name

1. The “name” attribute specifies the name of the context for this product. You can have an "All" context if you set the action of DesignerMenu to UseDefault to specify a menu setting that you want to apply to all contexts for that product.
2. Valid context name for different xml files:
    - 2D Extractor.xml — “All”, “3d modeler”, “report2d”
    - CircuitDesign.xml — “All”, "SchematicEditor", “Layout”, “Netlist”, “report2d”
    - Circuit Netlist.xml —“All”, “Design”, “Netlist”
    - HFSS.xml — “All”, “3d modeler”, “report2d”
    - HFSS 3D Layout Design.xml — “All”, “Design”, “Layout”, “Layout3D Editor”, “report2d”
    - HFSS-IE.xml — “All”, “3d modeler”, “report2d”
    - NoDesignUI.xml — “All”, “No Context”, “project”, “FilterDesign”
    - Q3D Extractor.xml — “All”, “3d modeler”, “report2d”

- RightClickMenu.xml — the following figure illustrates valid context names and how they display in Ansys Electronics Desktop.



- **Project Folder** – "Project"; in the above figure, “Project” represents the “Linckt_Export” project folder.
- **Design Instance Folder** – “Circuit Design”, “HFSS 3D Layout Design”, “HFSS”, “Q3D Extractor”, “Circuit Netlist”, “HFSS-IE”, “2D Extractor”
- **Definitions Folder** – “Definitions”; in the above figure, the Definitions folder is highlighted.
- For any Folder or Item under the Design Instance or Definitions folder, use the “/” notation to specify the path to the Folder or Item for context name. In the above figure: Use “Circuit Design/Results” for the Results folder. Use “Circuit Design/Design Verification” for the Design Verification folder.
  - For any folder or item under this, use the context name for the folder appended with “/Item”. In the above figure, “Circuit Design/Optimetrics/Item” represents ParametricSetup1.

> **Note:**
>
> For the Report folder under Results, please use the context name for the Results folder appended with "Report"; for example: "Circuit Design/Results/Report".
>
> For the Trace folder under Report, please use the context name for the Report appended with "Trace"; for example: "Circuit Design/Results/Report/Trace"

Sample XML Files

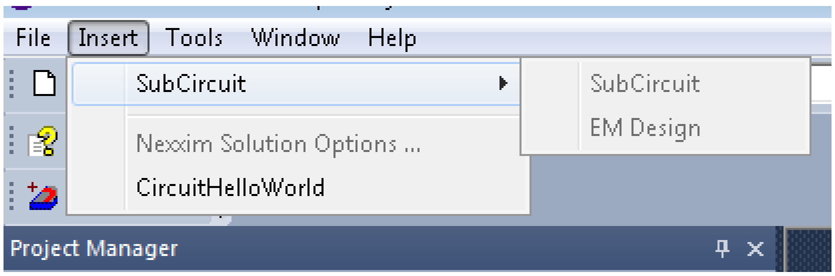
1. **Circuit Netlist.xml Sample**

```
<DesignerMenu xmlns="http://www.ansys.com/uiConfigMenu">
   <Context name="Design">
      <TopMenu>
         <MenuName>Insert</MenuName>
         <pop-upMenu>
            <MenuName>S&amp;ubCircuit</MenuName>
            <LeafMenu>
               <MenuName>&amp;SubCircuit</MenuName>
               <MenuID>55500</MenuID>
               <ShowBitMap>No</ShowBitMap>
               <Accelerator></Accelerator>
            </LeafMenu>
            <LeafMenu>
               <MenuName>&amp;EM Design</MenuName>
               <MenuID>55502</MenuID>
               <ShowBitMap>No</ShowBitMap>
               <Accelerator></Accelerator>
            </LeafMenu>
         </pop-upMenu>
         <Separator></Separator>
         <LeafMenu>
            <MenuName>Nexxim Solution &amp;Options ...</MenuName>

            <MenuID>38460</MenuID>
            <ShowBitMap>No</ShowBitMap>
            <Accelerator></Accelerator>
         </LeafMenu>
         <CustomMenu>
            <MenuName>CircuitHelloWorld</MenuName>
            <ScriptPath>$USERLIB/HelloWorld.vbs</ScriptPath>
            <ShowBitMap>No</ShowBitMap>
         </CustomMenu>
      </TopMenu>
```

```
    </Context>
</DesignerMenu>
```

Menus created by processing the above sample, Circuit Netlist.xml

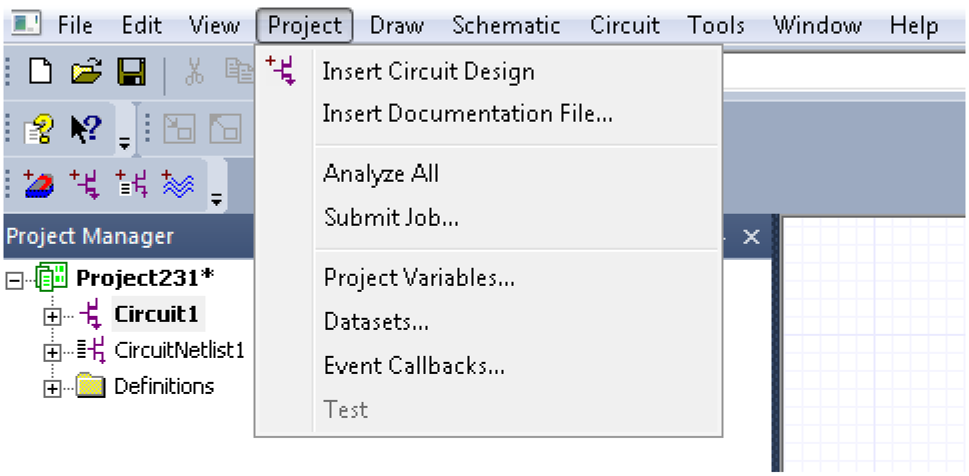


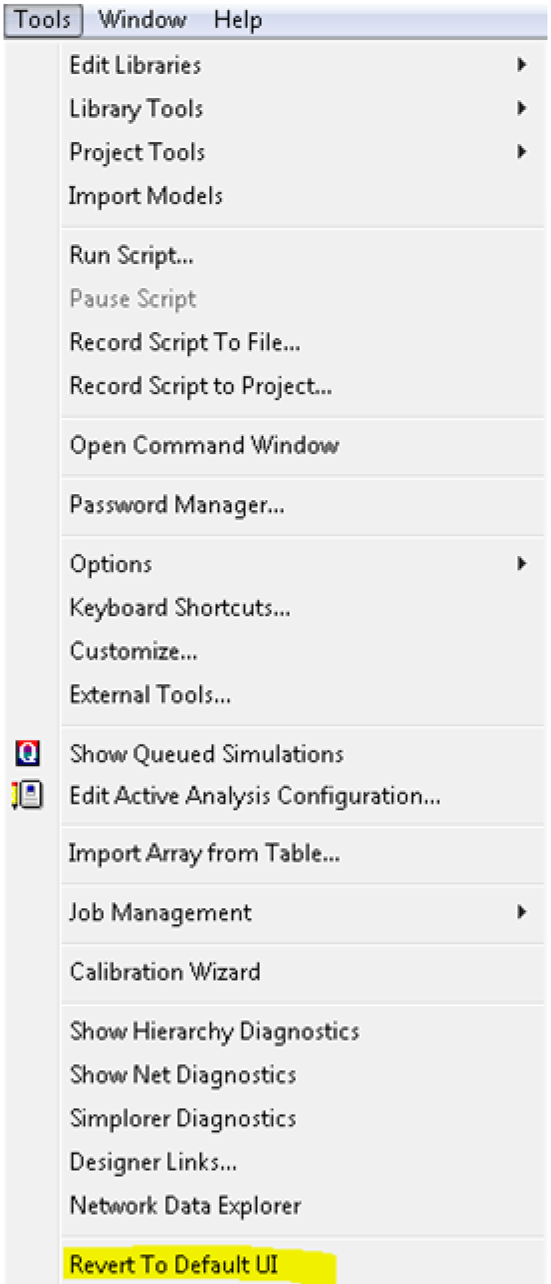
2. **Circuit Design.xml Sample**

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<DesignerMenu xmlns="http://www.ansys.com/uiConfigMenu"
action="useDefault">
  <UseProjectWindowSelectionContext>false</UseProjectWindowSelectionC
  ontext>
  <Context name="All">
    <TopMenu>
          <MenuName>&amp;Project</MenuName>
      <LeafMenu action="delete">
                 <MenuName>Insert HFSS Design</MenuName>
      </LeafMenu>
      <LeafMenu action="delete">
                 <MenuName>Insert HFSS 3D &amp;Layout
        Design</MenuName>
      </LeafMenu>
      <LeafMenu action="delete">
                 <MenuName>Insert HFSS-IE Design</MenuName>
      </LeafMenu>
      <LeafMenu action="delete">
                 <MenuName>Insert Q3D Extractor Design</MenuName>
      </LeafMenu>
      <LeafMenu action="delete">
```

```
                    <MenuName>Insert 2D Extractor Design</MenuName>
            </LeafMenu>
            <LeafMenu action="delete">
                    <MenuName>Insert Circuit &amp;Netlist</MenuName>
            </LeafMenu>
            <LeafMenu action="add">
                    <MenuName>Test</MenuName>
                    <MenuID>3333</MenuID>
            </LeafMenu>
        </TopMenu>
    </Context>
</DesignerMenu>
```

Menus created by processing the above XML sample, Circuit Design.xml



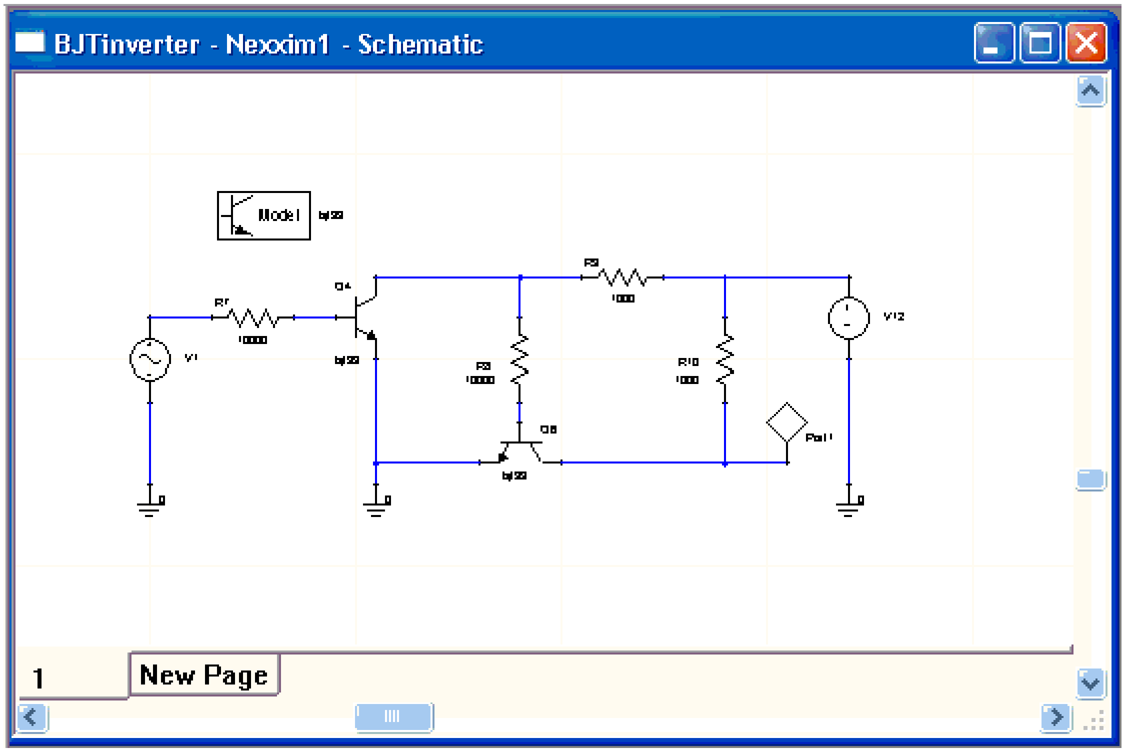


Ansys Electronics Desktop Design Area

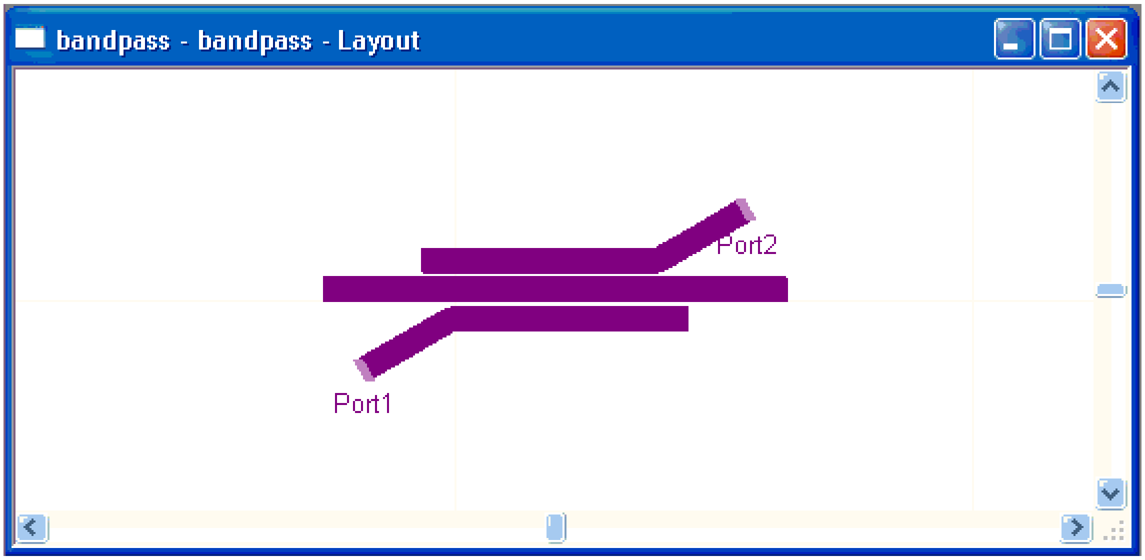
The Design Area of the desktop can display one or more editor windows and report windows, depending on the type of designs you create or load. See the Desktop Windows topic for ways to manipulate the windows in the Design Area.

Schematic Editor Window

The **Schematic Editor** window allows you to place components and wire them together.



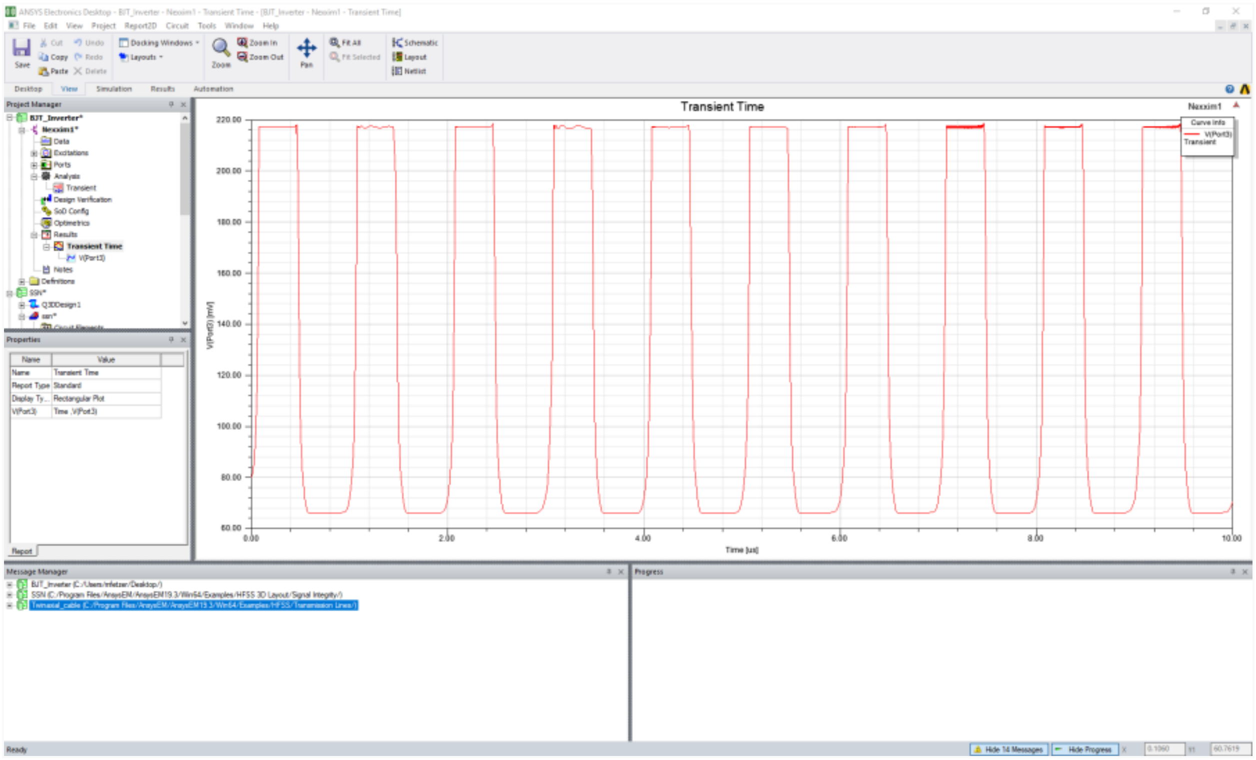
You can move components by simply selecting and dragging them. Copy and paste can be used on components and their wires within the schematic editor. You can also copy and paste to other schematics. For more information, see The Schematic Editor Window.



The Layout Editor shows the physical realization of the circuit. For more information, see Layout Editor.

Report Window

When a design has been successfully simulated, you can generate a report of results in a wide variety of forms, including XY graphs, polar graphs, 3D graphs, Smith charts, and data tables. Various attributes of each can be customized to your liking. The following shows a 2D report window in Electronics Desktop:



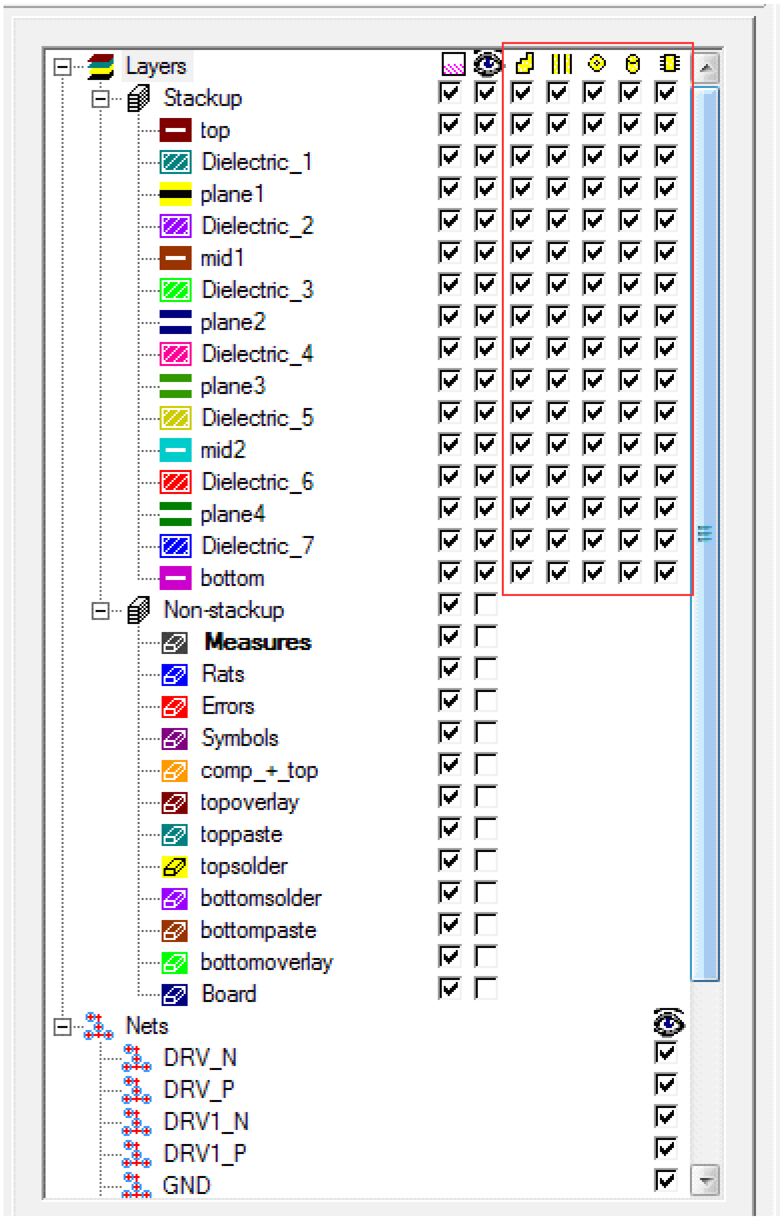
For more information, see Postprocessing and Generating Reports.

Layout Window

The Layout Window is a dockable Ansys Electronics Desktop window that can be resized and relocated, and can be used to view and configure various layout settings. Use the right-click menu of the Layout Window to configure the following:

- **Set Active** – makes the current layer the active layer.
- **Show This Layer Only** – makes the current layer the only visible layer.
- **Show All Dielectrics** – makes all Dielectric layers visible.
- **Hide All Dielectrics** – makes all Dielectric layers invisible.
- **Show All Signals** – makes all Signal layers visible.
- **Hide All Signals** – makes all Signal layers invisible.

Use the **Layout** window to alter the following controls for setting visibility by layout-object type.



The following controls are available:

 Controls the visibility of shapes

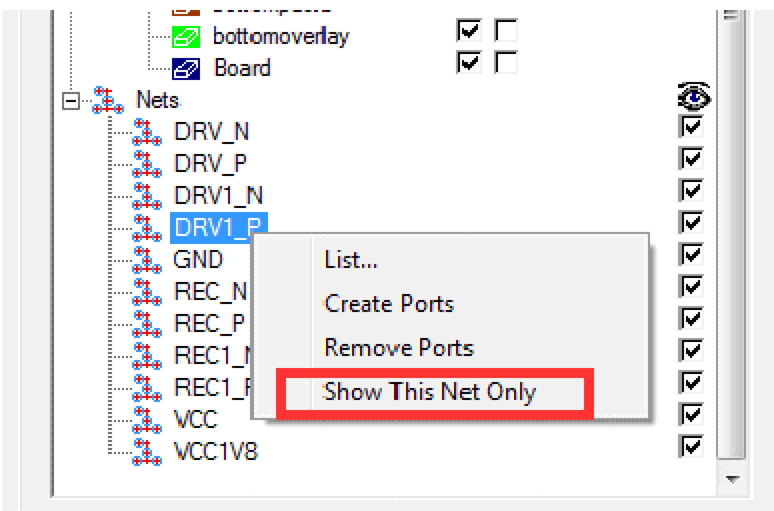
 Controls the visibility of lines (paths)

Controls the visibility of pads

Controls the visibility of holes

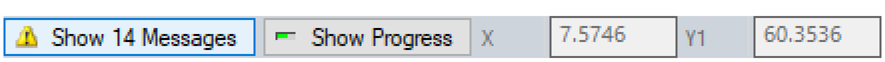
Controls the visibility of components

You can also turn visibility off for all nets but the selected net by right-clicking and selecting **Show This Net Only**.

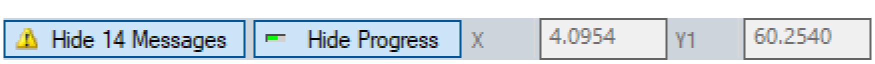


Ansys Electronics Desktop Status Bar

The status bar is located at the bottom of the application window. By default, it contains buttons to Show or Hide the Message window and Progress window.



Note that the Show Messages button indicates the number of messages.



When more than one progress bar is active, the top progress bar is represented on the status bar with a progress indicator:

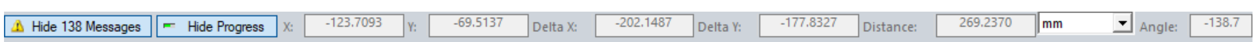
The status bar also displays helpful information about the current selection or command.

Directions for inputs appear on the left side of the status bar:

Depending on the command being performed, the status bar can display the following:

- **X**, **Y**, and **Z** coordinate boxes
- A drop-down menu for entering absolute, relative, cartesian, cylindrical, or spherical coordinates
- The model's units of measurement

These coordinate boxes and drop-down menus appear on the right side of the status bar:



To toggle status bar display:

- Click **View** > **Status Bar**.

A check mark next to this command indicates that the status bar is visible.

Keyboard Shortcuts

The following keyboard shortcuts apply to EMIT in general:

F1	Help (context-sensitive)
Ctrl + F4	Close window
Alt + F4	Close program
Ctrl + C	Copy
Ctrl + N	New Project
Ctrl + O	Open
Ctrl + P	Print
Ctrl + V	Paste

Ctrl + X	Cut
Ctrl + Y	Redo
Ctrl + Z	Undo
Ctrl + 0	Cascade windows
Ctrl + 1	Tile windows horizontally
Ctrl + 2	Tile windows vertically

See the subtopics in this Help branch for additional information concerning shortcuts and view navigation.

To customize the shortcut assignments, use **Tools > Keyboard Shortcuts**. Not all shortcuts are customizable.

Desktop Shortcuts

Modifier + key	Hold down the modifier, such as Shift or Ctrl, and press the key.

The following Ansys Electronics Desktop shortcut key combinations are available at any time:

Ctrl + N	New
Ctrl + O	Open
Ctrl + P	Print
Ctrl + S	Save
Ctrl + 0	Cascade windows
Ctrl + 1	Tile windows horizontally
Ctrl + 2	Tile windows vertically
Delete	Delete
F1	Open help

Schematic Shortcuts

Modifier + key	Hold down the modifier (such as Shift or Ctrl) and press the key.

The following shortcut key combinations are available when the **Schematic** window is active:

Ctrl + Z	Undo
Ctrl + Y	Redo

Ctrl + X	Cut
Ctrl + C	Copy
Ctrl + V	Paste
Ctrl + A	Select all
Ctrl + "+"	Zoom in
Ctrl + "-"	Zoom out
Ctrl + Q	Zoom area
Ctrl + D	Fit drawing
Ctrl + E	Reposition component
Ctrl + MMB + drag	Pan (see note below)
Shift + MMB + drag	Zoom (see note below)

Note:

For view navigation shortcuts (last two rows of the table above):

- **MMB** = Middle Mouse Button.
- These are the default view navigation assignments for version 2025 R2. Optionally, you can choose to use the legacy shortcuts that were appliable to 2023 R2 and earlier versions. See Choosing the View Navigation Options for more information.

Choosing the View Navigation Options

New mouse button and modifier key assignments (hotkeys) have been added for navigating the model view (rotate, pan, and zoom functions). Simultaneously, legacy hotkeys for these functions continue to be supported by default. Optionally, you can choose to disable the legacy view navigation shortcuts that were applicable to version 2023 R2 and earlier, as follows:

1. Access the *Options* dialog box using one of the following two methods:
   - From the menu bar, click **Tools** > **Options** > **General Options**.
   - From the **Desktop** ribbon tab, click **General Options**.
2. In the tree at the left side of the dialog box, select **General** > **User Interface**. Then:
   - To use either the legacy or current view navigation buttons and hotkey combinations, select **Enable Legacy View Navigation**. This option is selected by default.
   - To use only the current view navigation behavior and disable legacy button/hotkey support, ensure that **Enable Legacy View Navigation** is cleared.
3. Click **OK**.

The navigation option change becomes effective immediately (no program restart required).

Current vs. Legacy View Navigation

The following table summarizes changes in mouse-button and hotkey assignments between the current scheme (version 2024 R1 and newer) and the legacy scheme. "*Legacy*" assignments are applicable to version 2023 R2 or earlier and when the *Enable Legacy View Navigation* option is selected in newer versions):

> **Note:**
>
> The *"Current"* assignments are available whether or not the *Enable Legacy View Navigation* option is selected:

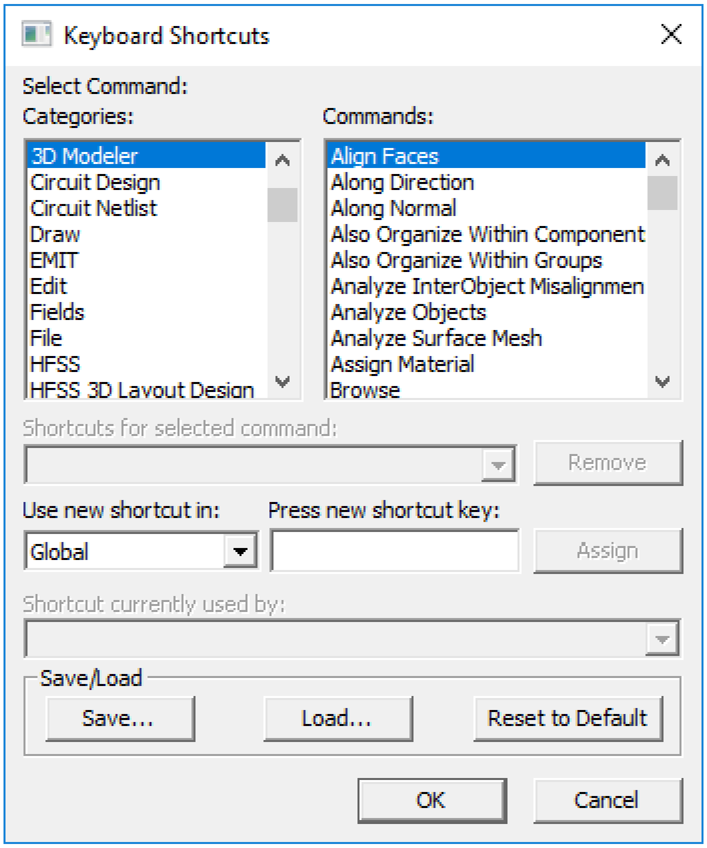
Function	**Key / Mouse Button Assignment**	
	Current	**Legacy**
Pan (all contexts)	Ctrl + MMB + drag or Ctrl + *arrow keys*	Shift + LMB + drag or Ctrl + *arrow keys*
Rotate (3D contexts)	MMB + drag or Alt + *arrow keys*	MMB + drag or Alt + LMB + drag or Alt + *arrow keys*

Function	Key / Mouse Button Assignment	
	Current	Legacy
Zoom (all contexts)	Wheel or Shift + MMB + drag or Ctrl + "+", Ctrl + "-"	Wheel or Alt + Shift + LMB + drag or Ctrl + "+", Ctrl + "-"

- "**LMB**" = Left Mouse Button
- "**MMB**" = Middle Mouse Button
- "**Wheel**" refers to rotation of the mouse wheel.
- In some 2D contexts, where rotation is not applicable, you can use **MMB + drag** to pan the view for either scheme.
- For **Alt + *arrow keys***, rotation is about a vertical or horizontal axis, regardless of the model view orientation.
- For zooming, if using the "=\+" key, instead of the "+" key on the numeric keypad, do **not** press Shift.

Custom Keyboard Shortcuts

Click **Tools** > **Keyboard Shortcuts** to open the **Keyboard Shortcuts** window. Here, you can view existing assignments, create new shortcuts, and save or load assignment files.



Selecting a **Command Category** populates the **Commands** list with the available commands for that category. If the command has an assigned shortcut, it is displayed in the **Shortcuts for the selected command** field. You can use the **Remove** button to disable the shortcut for the selected command. If the selected command does not have an assigned shortcut, the **Shortcuts for selected command** field and the **Remove** button are unavailable.

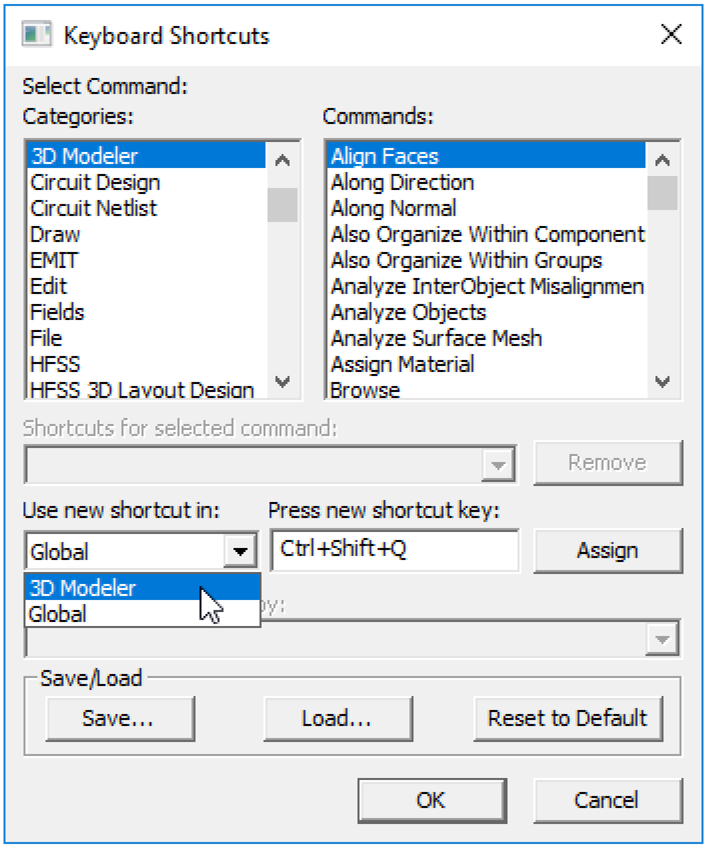
To create a new shortcut key:

1. Select the **Category** and **Command**.
2. If you want to disable a current assignment for the selected command, click **Remove**.
3. To assign a keyboard shortcut, place the cursor in the **Press new shortcut key** field.

   The field displays the keystrokes you make. When you have made keystrokes, the **Assign** button becomes available. If you combine keystrokes, these are displayed with a plus sign (+) between them. For example, Ctrl+P or Alt+O.

   If the shortcut you select is currently in use, that information displays in the **Shortcut currently used by** field.

4. The **Use new shortcut in** drop-down list displays **Global** by default, which means that the shortcut will apply to all applicable contexts. If a limited context exists, the menu offers a selection.



5. When you have made your desired assignments, you can save them to a named file.

   Click **Save...** to save the assignments to your desired location in Ansoft Keyboard Shortcut (*.aks) format.

   If you have an existing *.aks file, use the **Load...** button to locate and select the file.

6. Click **OK** to save the current settings, or **Reset to Default**.

Using the Password Manager to Control Access to Resources

Ansys Electronics Desktop lets you specify library resources that require password access, and encryption of those resources. For convenience, the same password can apply to multiple resources. To access the **Password Manager,** click **Tools** > **Password Manager**. This displays the **Password Manager** window.

To Specify a New Password Protected Resource

1. Click **Tools** > **Password Manager**.
2. Click **New**.

   This opens the **New Encrypted Resource** dialog box.
3. Specify the name of the resource that you want to protect and click **OK**.

   This displays the **Enter Passwords** window, which contains radio buttons to let you:
   - Enter Password and confirm for **Full Access** or for **Execute Only Access**.
   - **Use Ansys Password** (for execute only). This option does not require you to enter a password, but still encrypts the library.
4. Once you have selected a radio button and specified any necessary passwords correctly, click **OK**.

   The **Password Manager** updates to show the resource.

To Encrypt a Resource

1. Click **Tools** > **Password Manager**.
2. The **Password Manager** window appears.
3. In the Password Manager, select an existing resource to highlight it.
4. Click **Encrypt File**.

   A file browser window appears.
5. Use the drop-down menu to select the appropriate **Files of Type** filter.

   Any existing resources in the selected directory will appear.
6. When you have selected the appropriate resource, click **OK**.

   This encrypts the resource.

> **Note:**
>
> The **Expire** resource on option lets you select the date when the password expires for the relevant resource.

Running Ansys Electronics Desktop from a Command Line

Ansys Electronics Desktop includes line arguments that can be included when launching from a command line or terminal prompt. All command-line arguments are case-insensitive. The commands associated batch options can also be used with a Job Management Interface for submitting jobs to Ansys or RSM and other supported schedulers.

Command-line Syntax

`ansysedt <options> <run command(s)> <project file/script file>`

It is good practice to put quotation marks around the path to the solver executable, and around the full path to the project. This ensures that spaces in the path or project will not be an issue. The same is true of the design name, if there are spaces.

Run Commands

The following command line run commands are available for Ansys Electronics Desktop. Of the commands (BatchSave, BatchSolve, BatchExtract, RunScript, RunScriptandExit), one or none must be specified as arguments after the solver executable. When none is specified, you may specify a project or archive to open when Electronics Desktop launches, and can only use the **-Iconic** and **-Help** options. The commands are further described below:

- **-BatchSave** [options] **<***file or folder specifier***>**

  Saves a named project or folder containing one or more project files or folders to the current version. You can run this command with the **-Iconic**, **-Logfile**, and **-ng** options.

  If no migration is necessary (*e.g.* a project containing a circuit design): The original file will be overwritten with the new version, and no backup will be made.

  Special behavior when opening backed up files from the user interface: Files located in the AnsysEM_Backup directory would be the original non-migrated files that were backed up during an earlier –batchsave or open. Opening a file located in the AnsysEM_Backup directory via the open file dialog box will invoke the Parasolid migration, and the migrated file will be saved in the parent folder of the AnsysEM_Backup folder with a unique name so as not to overwrite any existing files.

  If a ***file*** is given, **-batchsave** will run on the specified file.

  If a ***folder*** is given, **-batchsave** will run on all files underneath the specified folder.

  .aedt, .aedtz, and .a3dcomp files will be processed.

  Allowed options:

  norecurse: only process files in top level of input folder

  forcearchive: project files are converted, then saved as archive files.

  previewonly: don't perform actual conversion, instead write preview of filenames to batchlog.

  outputfolder=<outputfolder>: save converted files to <outputfolder>, preserving subfolder structure.

- **-BatchSolve <***project file name***>**

  By default, this run command solves all adaptive setups, Optimetrics setups, and sweeps found in the project file. You can run this command with the **-Iconic**, **-Logfile**, **-ng**, and **-WaitForLicense** options. If parallel solve is possible, you can use the **-Distribute** option.

  If you wish to specify which setups **-BatchSolve** completes, you can use additional parameters:

  - **[*designName*]** – batch solve all setups for the specified design in the project file.
  - **[*designName*]:Nominal** – batch solve all nominal setups for the specified design in the project file.
  - **[*designName*]:Optimetrics** – batch solve all Optimetrics setups for the specified design in the project file.
  - **[*designName*]:[*Nominal/Optimetrics*]:[*SetupName*]** – batch solve the specified Nominal or Optimetrics setup in the specified design.

  If you wish to specify whether **-BatchSolve** setups are completed locally or remotely, you can use the following options:

  - **-Local** – performs the **-batchsolve** on the local machine.
  - **-Remote -machineList** – performs the **-batchsolve** on a remote machine. The <*machineList*> should provide a single hostname.
  - **-Distributed -machineList** – performs a distributed **-batchsolve** using a specified machine list.

    The **-machineList** parameter for a **-Distributed** setup can be formatted three ways:

    - **-MachineList list= "<***machine1***>, <***machine2***>, ..."** – machine names (either by IP address or hostname) are separated by commas. If the list contains any whitespace, it must be enclosed in quotation marks. The number of distributed COM engines run on each host is equal to the number of times the hostname appears in the list. That is, if host1 appears in the list once, and host2 appears twice, then one COM engine will run on host1 and two COM engines will run on host2.

      list= accepts the following additional modifiers:

      <*MachineName***>:<***TasksOnMachine***>:<***CoresOnMachine***>:<** *GPUsOnMachine***>**

      Duplicate machine names are not permitted. The integer for <*CoresOnMachine*> must be greater than the integer for <*TasksOnMachine*>. If -auto is specified with a machine list, the number of tasks for each machine must be -1.

Example:

```
list="Orion:4:8:90%:1, Aries:3:12, Pluto:6:12"
```

> **Note:**
>
> Duplicate machines are not allowed when specifying these additional modifiers. The number of cores must be greater than the number of tasks.

- **-MachineList file= "<***machineListFilepath***>"** – machine names (either by IP address or hostname) are listed in a file (one per line), and you specify the filepath. The number of distributed COM engines run on each host is equal to the number of times the hostname appears in the file. That is, if host1 appears in the file once, and host2 appears twice, then one COM engine will run on host1 and two COM engines will run on host2.

  file= accepts the following additional modifiers, in the file itself:

  <*MachineName***>:<***TasksOnMachine***>:<***CoresOnMachine***>:<** *GPUsOnMachine***>**

  Duplicate machine names are not permitted. The integer for <*CoresOnMachine*> must be greater than the integer for <*TasksOnMachine*>. If -auto is specified with a machine list, the number of tasks for each machine must be -1.

  Example:

  ```
  "Orion:4:8:90%:1",
  "Aries:3:12",
  "Pluto:6:12",
  ```

  > **Note:**
  >
  > Duplicate machines are not allowed when specifying these additional modifiers. The number of cores must be greater than the number of tasks.

- **-MachineList num= "<***numberofDistributedEngines***>"** – This format is used when a scheduler (such as LSF, PBS, SGE or HPC) is used to manage the jobs sent to a cluster of hosts. In a scheduler environment, you can specify the number tasks for distributed processing. In this case, you do not specify the machine names after the flag because the names are provided by the scheduler. For example, in the Windows HPC environment, you can write the number of tasks as follows:

```
-MachineList num=4
```

Distributed setups can also take the following optional arguments. When these are not present, the behavior defaults to single-level distributed solutions with no change in order of precedence among possible distribution types.

The arguments are:

- **includeTypes= <***default***>|<***distribution type 1, distribution type 2*, **...>** – If included distribution types are specified, only the listed distribution types are enabled. If default is specified, the default set of enabled distribution types is used. To see valid distribution types for your design, click **Simulation** > **Analysis Config** to open the **Analysis Configuration** window and view the types on the **Job Distribution** tab.

  If the list contains any whitespace, it must be enclosed in quotation marks. For example:
  ```
  "includeTypes=Frequencies,Mesh Assembly"
  ```
- **excludeTypes= <***default***>|<***distribution type 1, distribution type 2*, **...>** – If excluded distributed types are specified, all distribution types except those listed will be enabled. If default is specified, the default set of enabled distribution types is used. To see a valid distribution types for your design, click **Simulation** > **Analysis Config** to open the **Analysis Configuration** window and view the types on the **Job Distribution** tab.

  If the list contains any whitespace, it must be enclosed in quotation marks. For example:
  ```
  "excludeTypes=Frequencies,Mesh Assembly"
  ```
- **maxLevels= <**1 | 2**>** – the maximum number of levels of job distribution (the current maximum is 2). See: Selecting Optimal Configurations for Distributed Analysis.
- **numLevel1=** – when two-level distribution is selected (**maxLevels=2**), this specifies the number of level 1 tasks.

-Auto [**NumDistributedVariations=<***num***>**]

This flag enables automatic HPC settings and must be used with one of the following options:

- **-machinelist list=**<*machine list*>, with tasks for each machine set to -1
- **-machinelist numcores=**<*num*>, under a scheduler

All design types being solved must support **-auto** or the solve will be aborted.

The NumDistributedVariations option can be used to specify the number of optimetrics variations to solve simultaneously. The default is to solve optimetrics variations sequentially.

Arguments with **-auto** in a scheduler environment:

- **numcores=**<*total number of cores*>

  Total number of cores, and can be used only with -auto and in a scheduler environment.

- **file=**"<*tmachine list file path*>"

  The specified file can contain line delimited machine specifiers as described above.

- **num=**<*tnum distributed tasks*>

  This is the total number of tasks and used only in a scheduler environment.

- **numgpus=**<*tnumber of GPUs to use*>

  This is the total number of GPUs and used only in a scheduler environment. numgpus must be combined with either **num=** or **numcores=**.

  You can also specify how a **-BatchSolve** distributes Optimetrics variations:

- **-auto** – Without additional parameters, the batch log file will specify that Optimetrics variations be solved sequentially. If -auto is specified with a machine list, the number of tasks for each machine must be -1.
- **-auto NumDistributedVariations=<***num***>** – You can specify an integer value greater than 1. This is the number of variations that will be solved in parallel.

- **-BatchExtract <***BatchExtract script file name***> <***project file name***>** – allows the following commands to be executed non-graphically via script and without checking out any GUI licenses:
  - ExportProfile
  - ExportConvergence
  - ExportMeshStats
  - ExportNetworkData
  - ExportNMFData
  - ExportEigenmodes
  - ExportTransientData
  - Update Reports
  - ExportToFile

  A project file *must* be specified when **-BatchExtract** is used. Commands in the script file will only be executed in the specified project.

> **Important:**
>
> - **-ng** must be used with **-BatchExtract**, or it will fail.
> - Only the scripts listed above are supported for **-BatchExtract**. Including unsupported script commands will terminate script execution.

- **-RunScript <***script file name***>** – runs the specified script. You can use the **-ScriptArgs** option to add one or more arguments to this command, and the **-Iconic** option.
- **-RunScriptAndExit <***script file name***>** – runs the specified script and exits Electronics Desktop. You can use the **-ScriptArgs** option to add one or more arguments to this command. You can also use **-Iconic**, **-Logfile**, and **-WaitForLicense**.

> **Note:**
>
> **-BatchSolve <***DesignName***>** is mutually exclusive with **-RunScriptAndExit <***ScriptName***>**.

- **-monitor** – during non-graphical analysis, you can monitor progress and messages. Progress, warning and info messages are logged to the standard output stream. Error and fatal messages are logged to the standard error stream. Schedulers intercept these streams and provide commands for display of this output. See individual scheduler documentation for specifics.

- **-grpcsrv** – For external cpython scripting support. If not specified with **ansysedt.exe**, the Start GRPV server listens on a port in the default range 50051:51051. If specified, Start GRPV server listening on the the specified port number. An error is issued if the port is already used. For example:

```
ansysedt.exe // Start GRPV server listening on a port in the default range 50051:51051.

ansysedt.exe -grpcsrv portnumber // Start GRPV server listening on the the port number. error if the port already used.

ansysedt.exe -grpcsrv 50051:50150 // Start GRPV server listening on a port range 50051:50150.

ansysedt.exe -grpcsrv 50051:100 // Start GRPV server listening on a port range 50051:50151.
```

Job Management from the Command Line

- **-showmonitorjob** – Launch the monitor job dialog.
- **-showsubmitjob** – Launch the submit job dialog.

- **-showselectscheduler** – Launch the select scheduler dialog.

Run Command Examples

A distributed **-BatchSolve** of a specified design's Optimetrics setups, with a specified machine list:

```
C:\Program Files\ANSYS Inc\v252\AnsysEM\ansysedt -distributed -
machinelist list="255.255.1.1,255.255.1.2" -batchsolve
myDesign:Optimetrics "C:\myProject.aedt"
```

A **-BatchExtract** operation using paths to a script file and a project file:

> **Note:** The example is for an HFSS project. The path, project and design names will vary depending on what the user specifies.

```
ansysedt -ng -batchextract exportToFile.py "C:\Program Files\ANSYS
Inc\v252\AnsysEM\Examples\ElectronicsDesktop\HFSS\RF
Microwave\OptimTee.aedt"
```

Where exportToFile.py contains:

```
oDesktop.RestoreWindow()
oProject = oDesktop.SetActiveProject("OptimTee")
oDesign = oProject.SetActiveDesign("HFSSDesign1")
oModule = oDesign.GetModule("ReportSetup")
oModule.UpdateReports(["XY Plot 1"])
oModule.ExportToFile("XY Plot 1", "exportToFilePy.csv")
```

A **-BatchSolve** of a specified design's nominal setups, run in a minimized window, with a specified log file:

```
ansysedt -Iconic -LogFile "H:\Logs\mylog.log" -BatchSolve
myDesign:Nominal "H:\Projects\MyProject.aedt"
```

Specifying Project Files

Specifying a project file opens that project when Electronics Desktop launches. If **-BatchSolve** is set, the project will also be solved.

You can specify an archive file instead of a project file. If **-Batchsolve** is set, the project will be automatically restored and solved. Otherwise, you are prompted for a restore location, and the project will be restored and opened.

When a **-Batchsolve** is being performed on an archive file, you may also specify **-archiveoptions**:

- **overwritefiles** – allows non-project/results-extracted files to overwrite existing files.
- **path= <*projectFilepath*>** – extracts the project file and associated files to the specified path. If not specified, the archive will be extracted into the same directory as the archive file.
- **repackageresults** – Add batchsolve results back to archive file.
- **winpath= <*windowsProjectFilepath*>** – specifies the Windows-specific path to the extracted project file. This is used when a batch job is to be run on a Linux system, but monitored on Windows.

Options

The following options can be associated with one or more of the run commands:

- **-autoextract** – exports profile (as text), convergence (as text), and report data (as CSV) for the requested project/design/setup in a batch job. Once the solve is complete, an export directory is created (for example, "Project1.aedtexport" for a project named "Project1.aedt") that contains a subrectory for each design name. You can also specify –`autoextract “reports, fieldplots”` to also generate *.aedtplt files for each field plot and possible *.avz file (for import and display in Ensight) for all valid field plots. Export files reside within each design-name directory, and include setup name, design variation, job ID, and problem type, as applicable.

  > **Note:**
  >
  > - The -autoextract option is only valid when used with -BatchSolve
  > - The -autoextract option is automatically added for all Ansys Cloud Burst jobs submitted from Electronics Desktop. There is also an additional "reports" and/or "fieldplots" option that immediately follow "-autoextract". This causes all reports to be exported as CSV files at the end of the batch solve, after the profile and convergence have been exported.
  > - For example, you can specify `-autoextract “reports, fieldplots”` to also generate *.aedtplt files for each field plot and possible *.avz file for all valid field plots.

- **-batchoptions** – for batch jobs, specifies any of the options in **Tools** > **Options**. See additional information.
- **-batchoptionhelp** – opens a window showing -batchoptions help. The paths shown in this window can be used with batchoptions and the Update Registry Get and Set commands. See: Setting or Removing Option Values in Configuration Files: UpdateRegistry Command.

- **-distribute** – distributes a batch solve to multiple machines. This option must be combined with the **-BatchSolve** run command. See: Distributed Analysis.
- **-help** – opens a window displaying command line options. This can only be used without a run command.
- **-iconic** – runs Electronics Desktop with the window iconified (minimized).
- **-logfile <*filePath*>** – specifies a log file. If none is specified, <project_name>.log will be written to the <project_name>.batchinfo directory.
- **-monitor** – enables batch job output to standard output and standard error streams.
- **-ng** – runs Electronics Desktop in non-graphical mode. This *must* be used with the **-BatchExtract** command. For the ansysedtng beta option, see: Running Ansys Electronics Desktop from a Command Line (Nongraphical) Beta.
- **-waitforlicense** – directs Electronics Desktop to wait for unavailable licenses.
- **-scriptargs** – used in conjunction with **-RunScript** or **-RunScriptAndExit**, adds arguments to the specified script. You can pass multiple arguments to **-scriptargs** by surrounding the arguments in quotation marks. For example:

  ```
  ansysedt.exe -scriptargs "Design1 Setup1" -RunScriptAndExit
  C:\temp\test.py
  ```

  In Python, the command line parameter following **-scriptargs** is passed without modification as a single string in the ScriptArgument python variable.

  In either case, **Design1** is taken into Electronics Desktop as the first argument, and **Setup1** as the second argument. If you failed to use quotation marks, **Design1** would be taken as the first argument and **Setup1** would not be understood by Electronics Desktop.

-Batchoptions

All options that are specified through **Tools** > **Options** go into the user-level registry.

> **Note:**
>
> - Options are arranged as keys and values (in a structure similar to the *Windows Registry*). However, these options are not a part of the *Windows Registry* but are separately stored and maintained by the Ansys Electronics Desktop software.
> - For access to options and functionality beyond what is directly accessible via the user interface or batch options, refer to the documentation of the **UpdateRegistry** tool. This tool is discussed in the following help topic and in the topics that follow it in the same branch of the product help:
>
>   Setting or Removing Option Values in Configuration Files: UpdateRegistry Command

You can override the option registry entries via the **-batchoptions** command line. These overrides apply only to the current Desktop session. The registry setting overrides may be specified on the command line, or may be in a file with the file pathname specified on the command line. Batch jobs can be submitted from the command line or through Electronics Desktop's job submission window.

Large Scale DSO offers two new batchoptions related to the redistribution ability.

- LargeScaleDSO/VarRedistribution, where 0 disables redistribution (default), and 1 enables it.
- LargeScaleDSO/RedistributionLimit, is a positive integer specifying the minimum estimated remaining time (in minutes) for variations to redistribute to another task. The default is 3.

> **Note:**
>
> **-batchoptions** is only valid for batch jobs. It is ignored if you have not specified **-BatchSolve**, **-BatchSave**, or **-BatchExtract**.

-Batchoptions Examples

The batchoption **CreateStartingMesh** is available for the following products:

HFSS, HFSS-3DLayout, Icepak, Q3D, Q2D, Maxwell3D, Maxwell2D, and Mechanical.

When this option is set, only the initial mesh and manual mesh operations are completed for the batch solution.

> **Note:**
>
> No adaptive meshing occurs for any Icepak or Mechanical solutions. Therefore, after solving, the final mesh will be identical to the starting mesh for these two design types.

This example enables CreateStartingMesh for HFSS, and runs a batch solution of the specified project:

```
ansysedt -batchoptions "'HFSS/CreateStartingMesh'=1" -batchsolve
"D:\projects\MyProject.aedt"
```

See: Additional Examples of -Batchoptions Use.

Export Options Files

The **Tools** > **Options** > **Export Options Files** command writes XML files containing the options settings at all levels to the specified directory. This feature is intended to make it easier for different users to use Ansys Electromagnetics Suite 2025 R2 installed on shared directories or network drives. See: Example Uses for Export Options Features.

Examples and Further Explanations of -batchoptions Use

This section provides further examples and explanations of -batchoptions.

- Example with registry settings specified on the command line
- Example with registry settings specified in a file
- When to Use the -batchoptions Desktop Command Line Option

The following examples use general Desktop and HFSS-specific settings. This feature is available for all desktop products.

- The registry path separator is the slash (/) character.
- Each complete registry key (that is, a registry path and option name) is enclosed in single quotes.
- Registry string values are enclosed in single quotes.
- After the -batchoptions switch, the set of registry keys and values that follows it must be enclosed in double quotes. However, if a batchoptions file is referenced (instead of listing the options directly on the command line), the double quotes are not used around the filename.
- Backslashes in registry key string values must be escaped with another backslash (\\ ), since a backslash by itself is an escape code within strings.

Example with registry settings specified on the command line

```
ansysedt.exe -batchsolve -batchoptions
"'Desktop/Settings/ProjectOptions/NumberOfProcessors'=4
'HFSS/NumCoresPerDistributedTask'=2   (* This option is not
currently available from the Add Batchoption dialog box.)
'Desktop/ProjectDirectory'='C:\\projects\test'" projectname.aedt
```

This command line overrides registry values of the NumberOfProcessors (Desktop/Settings/ProjectOptions key) and ProjectDirectory (Desktop key) options.

> **Note:**
>
> - Multiple registry settings may appear in a single -batchoptions value, separated by whitespace.
> - The -batchoptions value must be enclosed in double quotes if it contains any whitespace.

Example with registry settings specified in a file

```
ansysedt.exe -batchsolve -batchoptions <filename>
projectname.aedt
```

where the referenced file, <*filename*>, contains:

```
$begin 'Config'
  'Desktop/Settings/ProjectOptions/NumberOfProcessors'=4
  'HFSS/NumCoresPerDistributedTask'=2
  'Desktop/ProjectDirectory'='C:/projects/test'
$end 'Config'
```

This command overrides the registry values of the NumberOfProcessors (Desktop/Settings/ProjectOptions key), NumCoresPerDistributedTask (HFSS key), and ProjectDirectory (Desktop key) options. These overrides apply only to the current Electronics Desktop session.

> **Note:**
>
> - The -batchoptions **<*filename*>** value must be enclosed in double quotes if it contains whitespace
> - The **$begin 'Config'** and **$end 'Config'** lines are required

For additional options you can override from the command line with **-batchoptions**, see:

- For -batchoptions Use: Project Directory and Lib Paths
- For -batchoptions Use: TempDirectory
- For -batchoptions Use: Various Desktop Settings

When to Use the -batchoptions Desktop Command Line Option

You can set analysis parameters for batch mode jobs using the graphical user interface (GUI). For example, you can set all options for a batch job using the **Add Batch Option** dialog box, which is accessed through the **Submit Job To** dialog box. These parameter settings include the following solver options (several examples, not a complete list):

- HPCLicenseType
- tempdirectory
- Desktop/AutoExtractReports
- Desktop/Settings/ProjectOptions/MumberOfProcessors
- <*design_type*>/DefaultProcessPriority
- <*design_type*>/EnableGPU and/or EnableGPUForEye – Applicable to Circuit Design, EMIT, HFSS, HFSS 3DLayout, and Maxwell 3D (also EnableGPUForSBR for HFSS only)
- <*design_type*>/MPIVendor (either “Intel” or “Microsoft”) – Applicable to HFSS, HFSS 3DLayout, Icepak, Maxwell 2D, Maxwell 3D), and Q3D

For graphical analyses that do **not** use batch mode, you specify the analysis parameters using the GUI (**Tools** > **Options** > **General Options** or **HPC and Analysis Options**). These settings are written to the registry when you exit the Electronics Desktop program. The settings are read from the registry when the application is started. Therefore, when you start the Electronics Desktop application, all settings retain the values from the previous session of the same user on the same machine. If there was not a previous session of the same user on the same machine, then the values are obtained from other registry configuration files or from a default value.

When running a batch analysis, any setting that is not specified using the **-batchoptions** command line option is taken from the registry. This value is typically the setting from the last session of the same user on the same machine. However, the **-batchoptions** command line option allows you to override the parameter with values specified on the command line or in a batchoptions file. The values specified using the **-batchoptions** command line option only apply to the batch job, and do not affect the parameter values stored in the registry.

If important **-batchoptions** values are not specified when running a batch job, the paramaters could be affected by an interactive session running on the same host by the same user. Parameter changes can occur if the user sets an option in the GUI and then exits the program, or if another process that accesses the registry exits. To be sure of the desired batch job outcome, avoid changing options in concurrent interactive sessions or include the desired -batchoptions in the command line.

For -batchoptions Use: Project Directory and Lib Paths

The PersonalLib, syslib and userlib settings are a little different from other settings. If the final directory name is different from what is expected, then PersonalLib, syslib or userlib is appended as a final directory. In addition, these settings may come from a different registry value if the registry values shown above are not set.

Registry Key	Default Value	Units or Values	Description
Desktop/ProjectDirectory	subdirectory of user's HOME directory or “Documents” directory	Directory pathname	Directory where new projects are created
Desktop/PersonalLib	PersonalLib subdirectory of user's HOME directory or “Documents” directory	Directory pathname	Directory PersonalLib is appended if final directory is not PersonalLib
Desktop/syslib	syslib subdirectory of installation directory	Directory pathname	Directory syslib is appended if final directory is not syslib
Desktop/userlib	userlib subdirectory of installation directory	Directory pathname	Directory userlib is appended if final

Registry Key	Default Value	Units or Values	Description
			directory is not userlib

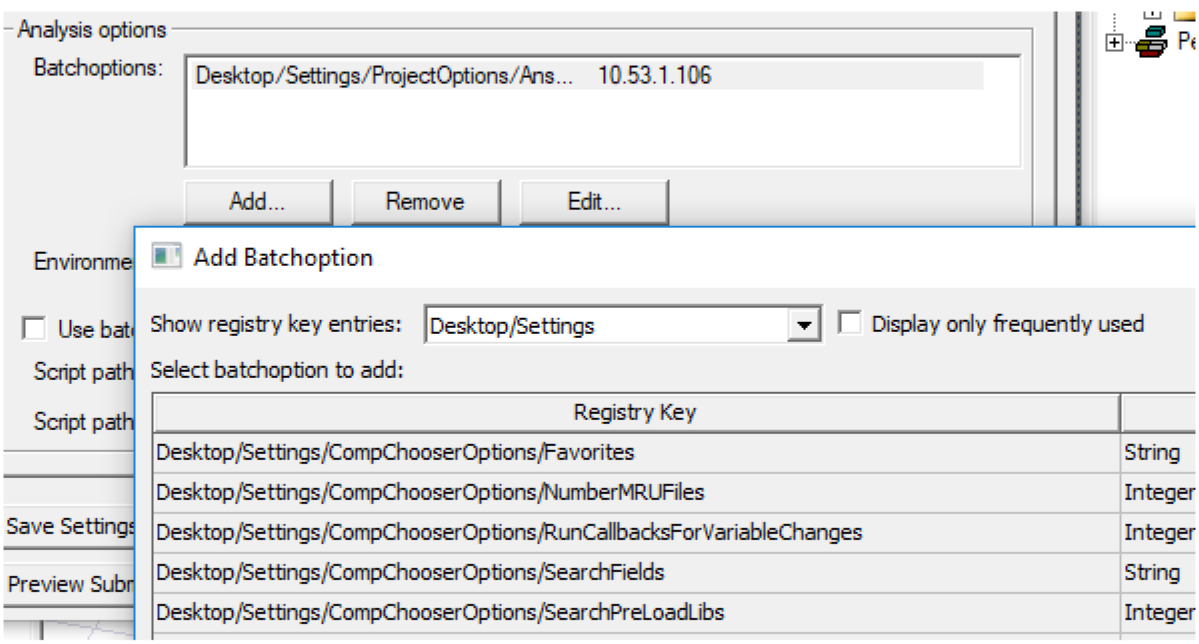
For -batchoptions Use: TempDirectory

Registry Key	Default Value	Units or Values	Description
TempDirectory	Set by installer	-	Directory for temporary files

For -batchoptions Use: Various Desktop Settings

Note that most of these options only affect the GUI. To view these options from the *Submit Job To* window:

1. In the *Analysis options* area of the *Submit Job To* window, click **Add...**.
2. In the *Add Batchoption* window that appears, select **Desktop/Settings** from the **Show registry key entries** drop-down menu.



Registry Key	Default Value	Units or Values	Description
Desktop/Settings/ProjectOptions/AnimationMemory	200	Megabytes (MB)	Stop animations when available memory falls below this value
Desktop/Settings/ProjectOptions/AnsoftCOMPreferredIPAddress (see note below table)	“” (empty string)	IP address (as a string)	IP address used to connect from COM engines to ansysedt.exe
Desktop/Settings/ProjectOptions/AnsysEMPreferredSubnetAddress (see note below table)	“” (empty string)	IP address (as a string)	Subnet used to connect COM engines to ansysedt.exe – allowed formats are: • IPv4 network prefix in CIDR notation Example: 123.123.123.0/24 • IPv4 network prefix with /subnet mask appended Example: 123.123.123.0/255.255.255.0
Desktop/Settings/ProjectOptions/AutoSaveInterval	10	edits	Number of edits to allow between autosaves
Desktop/Settings/ProjectOptions/AutoShowMessageWindow	1 (true)	0 (false) or 1 (true)	Show message window on new messages
Desktop/Settings/ProjectOptions/AutoShowProgressWindow	0 (false)	0 (false) or 1 (true)	Show progress window when starting a simulation
Desktop/Settings/ProjectOptions/DiskLimitForAbort	0	Megabytes (MB)	A warning is issued when available disk space falls below this value

Registry Key	Default Value	Units or Values	Description
Desktop/Settings/ProjectOptions/DoAutoSave	1 (true)	0 (false) or 1 (true)	Enables autosaves if true
Desktop/Settings/ProjectOptions/DrawStateIconsInProjectTree3	1 (true)	0 (false) or 1 (true)	Change icon when selection does not match active window
Desktop/Settings/ProjectOptions/ExpandMessageTreeOnInsert	1 (true)	0 (false) or 1 (true)	Ensure that new messages are visible in the message window tree
Desktop/Settings/ProjectOptions/ExpandOnInsert	0 (false)	0 (false) or 1 (true)	Expand project tree on insert
Desktop/Settings/ProjectOptions/HighlightActiveContextInProjectTree2	1 (true)	0 (false) or 1 (true)	Emphasize active command context (menu and toolbars)
Desktop/Settings/ProjectOptions/SavePreviewImagesInProject File	1 (true)	0 (false) or 1 (true)	Save preview images in project file
Desktop/Settings/ProjectOptions/UpdateReportOnFileOpen	0 (false)	0 (false) or 1 (true)	Update reports on file open

Note:

The preferredIP address and preferred subnet address settings are mutually exclusive. If both are specified to be non-empty strings, then the preferred IP address takes precedence, and the preferred subnet address is ignored. This feature is typically used for cluster environments using batch solves. The setting can be made via batchoptions but can also be done via UpdateRegistry.

Running Ansys Electronics Desktop from a Command Line (Nongraphical) Beta

> **Note:**
>
> Running Electronics Desktop using the ansysedtng command is currently a beta feature. For information on enabling beta options, see General Options: Desktop Configuration.

Ansys Electronics Desktop includes line arguments that can be included when launching from a command line or terminal prompt. All command-line arguments are case-insensitive. The commands associated batch options can also be used with a Job Management Interface for submitting jobs to Ansys or RSM and other supported schedulers.

> **Note:**
>
> Linux users whose installations lack a MainWin home directory receive an error when running the ansysedt command. To run Elecronics Desktop from a command line successfully on such Linux machines, use the ansysedtng command and options described in this section.

Command-line Syntax

```
ansysedtng <options> <run command(s)> <project file/script file>
```

It is good practice to put quotation marks around the path to the solver executable, and around the full path to the project. This ensures that spaces in the path or project will not be an issue. The same is true of the design name, if there are spaces.

Run Commands

The following command line run commands are available for Ansys Electronics Desktop. BatchSolve must be specified as arguments after the solver executable. When none is specified, you may specify a project or archive to open when Electronics Desktop launches, and can only use the **-Help** option. The commands are further described below:

- **-BatchExtract <***BatchExtract script file name***> <***project file name***>** – allows the following commands to be executed non-graphically via script and without checking out any GUI licenses:
- ExportProfile
- ExportConvergence
- ExportMeshStats

- ExportNetworkData
- ExportNMFData
- ExportEigenmodes
- ExportTransientData
- Update Reports
- ExportToFile

- A project file *must* be specified when **-BatchExtract** is used. Commands in the script file will only be executed in the specified project.

- > **Important:**
  >
  > - Only the scripts listed above are supported for **-BatchExtract**. Including unsupported script commands will terminate script execution.

- **-BatchSolve <***project file name***>**

  By default, this run command solves all adaptive setups, Optimetrics setups, and sweeps found in the project file. You can run this command with the **-Logfile** and **-WaitForLicense** options. If parallel solve is possible, you can use the **-Distribute** option.

  If you wish to specify which setups **-BatchSolve** completes, you can use additional parameters:

  - **[*designName*]** – batch solve all setups for the specified design in the project file.
  - **[*designName*]:Nominal** – batch solve all nominal setups for the specified design in the project file.
  - **[*designName*]:Optimetrics** – batch solve all Optimetrics setups for the specified design in the project file.
  - **[*designName*]:[*Nominal/Optimetrics*]:[*SetupName*]** – batch solve the specified Nominal or Optimetrics setup in the specified design.

  If you wish to specify whether **-BatchSolve** setups are completed locally or remotely, you can use the following options:

  - **-Local** – performs the **-batchsolve** on the local machine.
  - **-Remote -machineList** – performs the **-batchsolve** on a remote machine. The <*machineList*> should provide a single hostname.
  - **-Distributed -machineList** – performs a distributed **-batchsolve** using a specified machine list.

    The **-machineList** parameter for a **-Distributed** setup can be formatted three ways:

- **-MachineList list= "<*machine1*>, <*machine2*>, ..."** – machine names (either by IP address or hostname) are separated by commas. If the list contains any whitespace, it must be enclosed in quotation marks. The number of distributed COM engines run on each host is equal to the number of times the hostname appears in the list. That is, if host1 appears in the list once, and host2 appears twice, then one COM engine will run on host1 and two COM engines will run on host2.

  list= accepts the following additional modifiers:

  <*MachineName***>:<***TasksOnMachine***>:<***CoresOnMachine***>:<** *GPUsOnMachine***>**

  Duplicate machine names are not permitted. The integer for <*CoresOnMachine*> must be greater than the integer for <*TasksOnMachine*>. If -auto is specified with a machine list, the number of tasks for each machine must be -1.

  Example:

  ```
  list="Orion:4:8:90%:1, Aries:3:12, Pluto:6:12"
  ```

  > **Note:**
  >
  > Duplicate machines are not allowed when specifying these additional modifiers. The number of cores must be greater than the number of tasks.

- **-MachineList file= "<*machineListFilepath*>"** – machine names (either by IP address or hostname) are listed in a file (one per line), and you specify the filepath. The number of distributed COM engines run on each host is equal to the number of times the hostname appears in the file. That is, if host1 appears in the file once, and host2 appears twice, then one COM engine will run on host1 and two COM engines will run on host2.

  file= accepts the following additional modifiers, in the file itself:

  <*MachineName***>:<***TasksOnMachine***>:<***CoresOnMachine***>:<** *GPUsOnMachine***>**

  Duplicate machine names are not permitted. The integer for <*CoresOnMachine*> must be greater than the integer for <*TasksOnMachine*>. If -auto is specified with a machine list, the number of tasks for each machine must be -1.

  Example:

```
"Orion:4:8:90%:1",
"Aries:3:12",
"Pluto:6:12",
```

> **Note:**
>
> Duplicate machines are not allowed when specifying these additional modifiers. The number of cores must be greater than the number of tasks.

- **-MachineList num= "<*numberofDistributedEngines*>"** – This format is used when a scheduler (such as LSF, PBS, SGE or HPC) is used to manage the jobs sent to a cluster of hosts. In a scheduler environment, you can specify the number tasks for distributed processing. In this case, you do not specify the machine names after the flag because the names are provided by the scheduler. For example, in the Windows HPC environment, you can write the number of tasks as follows:
  ```
  -MachineList num=4
  ```

Distributed setups can also take the following optional arguments. When these are not present, the behavior defaults to single-level distributed solutions with no change in order of precedence among possible distribution types.

The arguments are:

- **includeTypes= <*default*>|<*distribution type 1, distribution type 2*, ...>** – If included distribution types are specified, only the listed distribution types are enabled. If default is specified, the default set of enabled distribution types is used. To see valid distribution types for your design, click **Simulation** > **Analysis Config** to open the **Analysis Configuration** window and view the types on the **Job Distribution** tab.

  If the list contains any whitespace, it must be enclosed in quotation marks. For example:
  ```
  "includeTypes=Frequencies,Mesh Assembly"
  ```
- **excludeTypes= <*default*>|<*distribution type 1, distribution type 2*, ...>** – If excluded distributed types are specified, all distribution types except those listed will be enabled. If default is specified, the default set of enabled distribution types is used. To see a valid distribution types for your design, click **Simulation** > **Analysis Config** to open the **Analysis Configuration** window and view the types on the **Job Distribution** tab.

  If the list contains any whitespace, it must be enclosed in quotation marks.

For example:

`"excludeTypes=Frequencies,Mesh Assembly"`

- **maxLevels= <**1 | 2**>** – the maximum number of levels of job distribution (the current maximum is 2). See: Selecting Optimal Configurations for Distributed Analysis.
- **numLevel1=** – when two-level distribution is selected (**maxLevels=2**), this specifies the number of level 1 tasks.

-Auto [**NumDistributedVariations=<***num***>**]

This flag enables automatic HPC settings and must be used with one of the following options:

- **-machinelist list=**<*machine list*>, with tasks for each machine set to -1
- **-machinelist numcores=**<*num*>, under a scheduler

All design types being solved must support **-auto** or the solve will be aborted.

The NumDistributedVariations option can be used to specify the number of optimetrics variations to solve simultaneously. The default is to solve optimetrics variations sequentially.

Arguments with **-auto** in a scheduler environment:

- **numcores=**<*total number of cores*>

  Total number of cores, and can be used only with -auto and in a scheduler environment.

- **file=**"<*tmachine list file path*>"

  The specified file can contain line delimited machine specifiers as described above.

- **num=**<*tnum distributed tasks*>

  This is the total number of tasks and used only in a scheduler environment.

- **numgpus=**<*tnumber of GPUs to use*>

  This is the total number of GPUs and used only in a scheduler environment. numgpus must be combined with either **num=** or **numcores=**.

  You can also specify how a **-BatchSolve** distributes Optimetrics variations:

- **-auto** – Without additional parameters, the batch log file will specify that Optimetrics variations be solved sequentially. If -auto is specified with a machine list, the number of tasks for each machine must be -1.
- **-auto NumDistributedVariations=<***num***>** – You can specify an integer value greater than 1. This is the number of variations that will be solved in parallel.

- **-monitor** – during non-graphical analysis, you can monitor progress and messages. Progress, warning and info messages are logged to the standard output stream. Error and fatal messages are logged to the standard error stream. Schedulers intercept these streams and provide commands for display of this output. See individual scheduler documentation for specifics.

Run Command Examples

A distributed **-BatchSolve** of a specified design's Optimetrics setups, with a specified machine list:

```
C:\Program Files\ANSYS Inc\v252\AnsysEM\ansysedtng -distributed -
machinelist list="255.255.1.1,255.255.1.2" -batchsolve
myDesign:Optimetrics "C:\myProject.aedt"
```

A **-BatchSolve** of a specified design's nominal setups, run in a minimized window, with a specified log file:

```
ansysedtng -LogFile "H:\Logs\mylog.log" -BatchSolve
myDesign:Nominal "H:\Projects\MyProject.aedt"
```

Specifying Project Files

Specifying a project file opens that project when Electronics Desktop launches. If **-BatchSolve** is set, the project will also be solved.

You can specify an archive file instead of a project file. If **-Batchsolve** is set, the project will be automatically restored and solved. Otherwise, you are prompted for a restore location, and the project will be restored and opened.

When a **-Batchsolve** is being performed on an archive file, you may also specify **-archiveoptions**:

- **overwritefiles** – allows non-project/results-extracted files to overwrite existing files.
- **path= <***projectFilepath***>** – extracts the project file and associated files to the specified path. If not specified, the archive will be extracted into the same directory as the archive file.
- **repackageresults** – Add batchsolve results back to archive file.
- **winpath= <***windowsProjectFilepath***>** – specifies the Windows-specific path to the extracted project file. This is used when a batch job is to be run on a Linux system, but monitored on Windows.

Options

The following options can be associated with one or more of the run commands:

- **-autoextract** – exports profile (as text), convergence (as text), and report data (as CSV) for the requested project/design/setup in a batch job. Once the solve is complete, an export directory is created (for example, "Project1.aedtexport" for a project named "Project1.aedt") that contains a sub-directory for each design name. You can also specify `-autoextract “reports, fieldplots”` to also generate *.aedtplt files for each field plot and possible *.avz file (for import and display in Ensight) for all valid field plots. Export files reside within each design-name directory, and include setup name, design variation, job ID, and problem type, as applicable.

> **Note:**
>
> - The -autoextract option is only valid when used with -BatchSolve
> - The -autoextract option is automatically added for all Ansys Cloud Burst Computing jobs submitted from Electronics Desktop. There is also an additional "reports" and/or "fieldplots" option that immediately follow "-autoextract". This causes all reports to be exported as CSV files at the end of the batch solve, after the profile and convergence have been exported.
> - For example, you can specify `-autoextract “reports, fieldplots”` to also generate *.aedtplt files for each field plot and possible *.avz file for all valid field plots.

- **-batchoptions** – for batch jobs, specifies any of the options in **Tools** > **Options**. See additional information.
- **-distribute** – distributes a batch solve to multiple machines. This option must be combined with the **-BatchSolve** run command. See: Distributed Analysis.
- **-help** – opens a window displaying command line options. This can only be used without a run command.
- **-logfile <*filePath*>** – specifies a log file. If none is specified, <project_name>.log will be written to the <project_name>.batchinfo directory.
- **-monitor** – enables batch job output to standard output and standard error streams.
- **-ng** – runs Electronics Desktop in non-graphical mode.
- **-waitforlicense** – directs Electronics Desktop to wait for unavailable licenses.

-Batchoptions

All options that are specified through **Tools** > **Options** go into the user-level registry.

Note:

- Options are arranged as keys and values (in a structure similar to the *Windows Registry*). However, these options are not a part of the *Windows Registry* but are separately stored and maintained by the Ansys Electronics Desktop software.
- For access to options and functionality beyond what is directly accessible via the user interface or batch options, refer to the documentation of the **UpdateRegistry** tool. This tool is discussed in the following help topic and in the topics that follow it in the same branch of the product help:

  Setting or Removing Option Values in Configuration Files: UpdateRegistry Command

You can override the option registry entries via the **-batchoptions** command line. These overrides apply only to the current Desktop session. The registry setting overrides may be specified on the command line, or may be in a file with the file pathname specified on the command line. Batch jobs can be submitted from the command line or through Electronics Desktop's job submission window.

Note:

-batchoptions is only valid for batch jobs. It is ignored if you have not specified **-BatchSolve**.

-Batchoptions Examples

The batchoption **CreateStartingMesh** is available for the following products:

HFSS, Icepak, Q3D, Q2D, Maxwell3D, Maxwell2D, and Mechanical.

When this option is set, only the initial mesh and manual mesh operations are completed for the batch solution.

Note:

No adaptive meshing occurs for any Icepak or Mechanical solutions. Therefore, after solving, the final mesh will be identical to the starting mesh for these two design types.

This example enables CreateStartingMesh for HFSS, and runs a batch solution of the specified project:

```
ansysedtng -batchoptions "'HFSS/CreateStartingMesh'=1" -batchsolve
"D:\projects\MyProject.aedt"
```

See: Additional Examples of -Batchoptions Use.

Export Options Files

The **Tools** > **Options** > **Export Options Files** command writes XML files containing the options settings at all levels to the specified directory. This feature is intended to make it easier for different users to use Ansys Electromagnetics Suite 2025 R2 installed on shared directories or network drives. See: Example Uses for Export Options Features.

Running Ansys Electronics Desktop with a JSON File

Ansys Electronics Desktop includes line arguments that can be included in a JSON file and executed by entering the -json command line option. The -json option takes a JSON file as argument, and runs AEDT with the options specified within the file. All options in the -help text is supported. Options in the JSON file can be used together with options specified in the command line. If the same option is specified both in the command line and the JSON file, the option in the command line overrides the option in the JSON file.

> **Note:**
>
> The -json option functions with both the ansysedt and ansysedtng commands.

Enter the command as follows to execute the JSON file

```
ansysedt -json <filepath to the JSON file>
```

JSON File Format

Take the following into consideration when creating a JSON file.

- The JSON file must contain an object. Each option is a key value pair where the key is the option name (no dash) and the value is the argument.
- For options that do not take any arguments, such as -monitor, the value must be a bool - true or false to turn on or off the option, respectively.
- For options that take optional arguments, such as -logfile, the value can either be a bool or some other type (in this case, a string specifying the log file) depending on the option.
- Project/archive file and folder should be specified directly on the command line when using -json.
- See the example below for a JSON file containing all options in the -help text.

> **Note:**
>
> The example does not contain a valid combination of options but is solely used to show what type of argument each option takes.

```
{
    "archiveoptions":
    {
        "overwritefiles": true,
        "path": "\\\\dataserver\\home\\user\\OptimTee.aedt",
        "repackageresults": true,
        "winpath": "\\\\dataserver\\home\\user\\OptimTee.aedt"
    },
    "auto":
    {
        "numDistributedVariations": 2
    },
    "autoextract": "reports, fieldplots",
    "batchextract": "\\\\dataserver\\home\\user\\batchextract_
script.py",
    "batchoptionhelp": false,
    "batchoptions":
    {
        "HFSS/CreateStartingMesh": 1,
        "Desktop/ProjectDirectory": "\\\\dataserver\\home\\user"
    },
    "batchsave":
    {
        "autoheal": false,
        "cleanupmodel": true,
        "forcearchive": true,
        "norecurse": true,
        "outputfolder": "\\\\dataserver\\home\\user\\batchsave_
folder",
        "preservehistory": false,
        "previewonly": false
    },
    "batchsolve":
    {
        "designName": "TeeModel",
        "setupType": "Nominal",
        "setupName": "OptimTee_Analysis"
    },
    "distributed":
    {
        "excludeTypes":
        [
            "Frequencies",
            "Mesh Assembly"
        ],
```

```
        "includeTypes": [],
        "maxLevels": 2,
        "numLevel1": 2
    },
    "help": false,
    "iconic": false,
    "local": false,
    "logfile": "\\\\dataserver\\home\\user\\logfile.log",
    "machinelist":
    {
        "list":
        [
            {
                "machine": "sjohpcw2k16fe",
                "numTasks": -1,
                "numCores": 1,
                "RAMPercent": 90,
                "numGPUs": 1
            },
            {
                "machine": "dcuww10core01",
                "RAMPercent": 80
            }
        ]
    },
    "monitor": true,
    "ng": true,
    "runscript": "\\\\dataserver\\home\\user\\script.py",
    "runscriptandexit": "\\\\dataserver\\home\\user\\script.py",
    "scriptargs": "Design1 Setup1",
    "showmonitorjob": false,
    "showselectscheduler": false,
    "showsubmitjob": false,
    "waitforlicense": false
}
```

Running Ansys Electronics Desktop from a Windows Remote Terminal

When running EMIT from a remote terminal, there are some performance and behavior issues to consider. These issues are due to the interaction of bandwidth/opengl drivers/remote-terminal-protocol

- Showing axes when interactively drawing objects will slow the performance.
- Remote OpenGL performance will be slower in general. Graphics card and driver quality helps.
- All 3D windows will be closed when you switch from remote PC to a console or from a console to remote. This is to avoid display/opengl instability during the switch.
- Grid will not be turned off while viewing a plot from a remote desktop. The mouse over highlights on 2D plots may appear as not totally overlapping the line color or as thin dotted lines.

Using Windows HPC Commands

HPC Integration allows you to submit jobs directly using Ansys Electromagnetics command line arguments for batch solves. Ansys Electromagnetics products must be accessible from the same directory on all machines. The Ansys Electromagnetics command line syntax is documented here. You must pass in a -distributed flag as part of the Ansys Electromagnetics command line arguments if you want to run a distributed simulation.

For information concerning the supported HPC software, go the the Platform Support webpage and click the links to the following topics:

- *Ansys 2025 R2 - Job Schedulers and Queuing Systems Support*
- *Ansys 2025 R2 - Message Passing Interface Support for Parallel Computing*

Before running a job you must click **Tools** > **Job Management** > **Select Scheduler** and use the dialog box to designate the head node of a cluster. You can then click **Tools** > **Job Management** > **Submit Job** to submit the batch commands for the job.

Customizing Ansys Electronics Desktop with Ansys ACT

With Ansys ACT, you can create custom applications or extensions to customize supported Ansys products, including Electronics Desktop.

> **Note:**
>
> The **ACT Extensions** window and the design wizards it contains (*5G Wizard, HFSS Antenna Design Toolkit, HFSS-EMA3D Link,* and *Maxwell Eccentricities*) are only available for the Windows version of the Ansys Electronics Desktop software. These items are not available when the software is installed on a Linux platform.

An ACT guided process extension enables you to leverage both the functionality of Electronics Desktop and the scripting capabilities of the Workbench/AIM framework API. You can manipulate existing features and simulation components, organizing them as needed to produce a custom automated process. A guided process extension is exposed in Electronics Desktop as a wizard that provides step-by-step simulation guidance within the application workflow.

To access ACT functionality in Electronics Desktop, open the ACT Home page by clicking **View** > **ACT Extensions**.

For more information, see ACT Simulation Wizards and Electronics Desktop Wizard in the *Ansys ACT Developer's Guide*.

Installing PyAEDT (Beta)

PyAEDT is a Python library that interacts with the AEDT API to make scripting simpler for the end user. It supports all AEDT 3D products (HFSS, Icepak, Maxwell 3D, and Q3D Extractor), 2D tools, Ansys Mechanical, EMIT, Circuit tools like Nexxim, system simulation tools like Twin Builder, and layout tools like HFSS 3D Layout and EDB. Additionally, it enables the end user to have a CPython interface with AEDT. Its class and method structures simplify operation for the end user, enabling more Pythonic code while reusing information as much as possible across the various APIs.

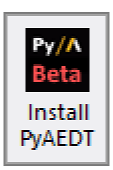
Documentation for PyAEDT can be found online at: https://aedt.docs.pyansys.com/version/stable/

Installing PyAEDT adds three items to the **Tools** > **Toolkit** > **PersonalLib** menu and to the **Automation** tab:

- **Console** – launches the PyAEDT console.
- **Jupyter Notebook** – launches Jupyter Notebook (a computational notebook) in an internet browser.
- **Run PyAEDT Script** – launches a file browser allowing you to select a Python script to run via PyAEDT.

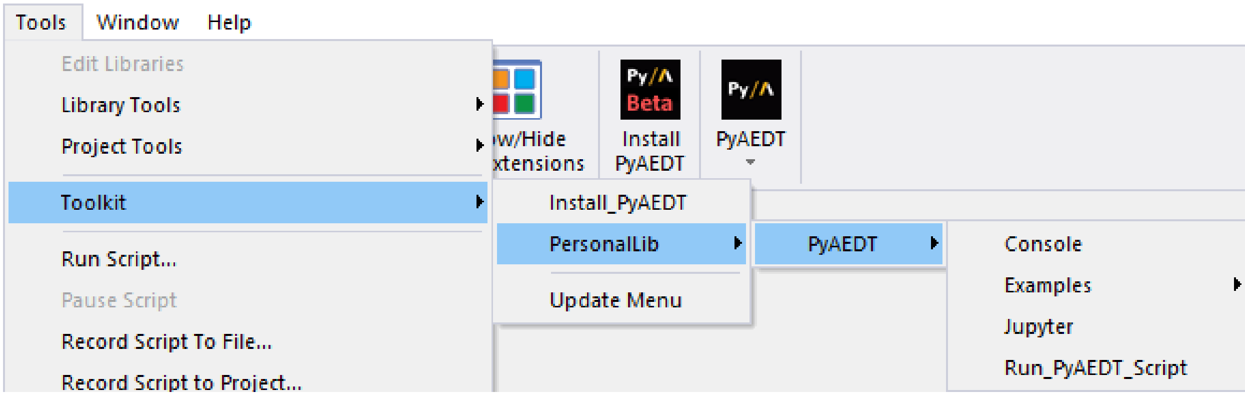
To install PyAEDT:

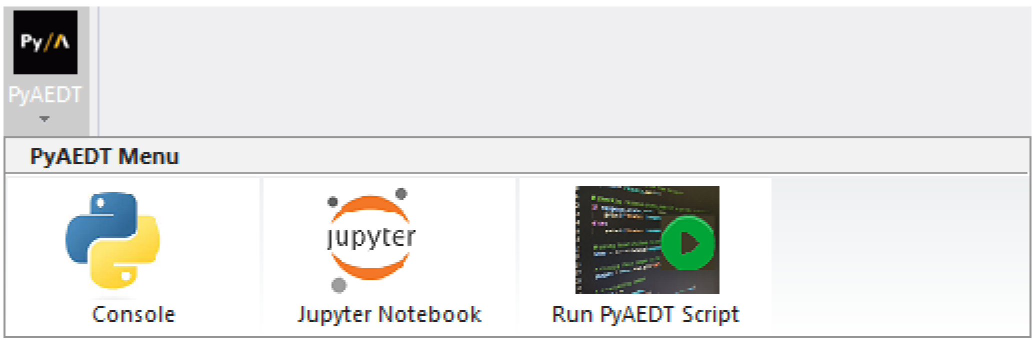
- From the **Automation** tab, click **Install PyAEDT**.



A web browser launches, and takes you to detailed installation instructions.

When installation is complete, the **Tools** > **Toolkit** > **PersonalLib** menu and the **Automation** tab update to display PyAEDT menu options:





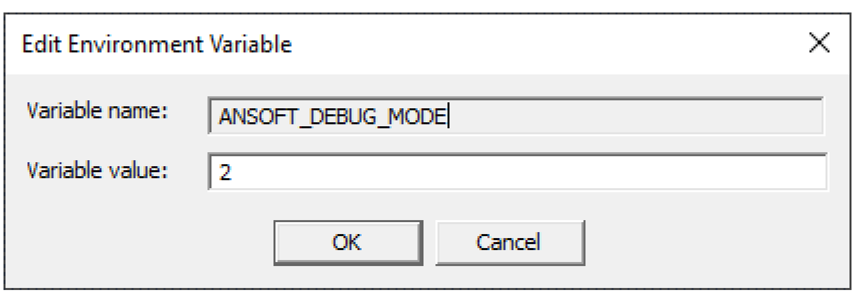
Debug Logging

Electronics Desktop provides error logging capabilities.

To begin logging errors:

1. Click **Tools** > **Debug Logging**.

   The **Edit Environment Variable** window appears, showing the ANSOFT_DEBUG_MODE variable.

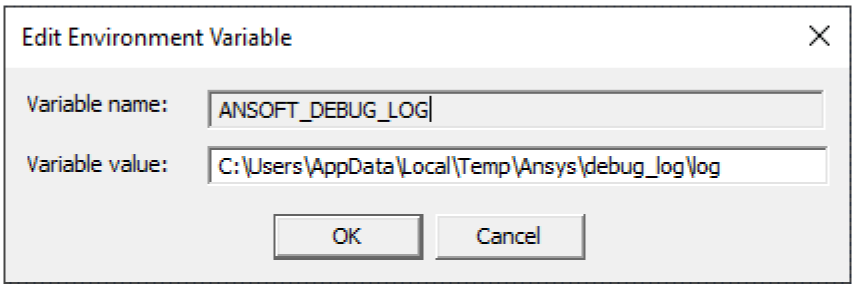


2. Enter a **Variable value**.

   The default value is 2, and should not be changed unless directed by technical support.

3. Click **OK**.

   The **Edit Environment Variable** window updates to display the ANSOFT_DEBUG_LOG variable.



4. Enter a folder path where log files (*.log) will be stored.
5. Click **OK**.

   Errors are logged to the specified folder until you disable logging.

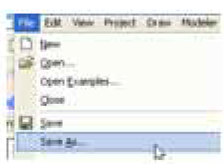
To stop logging errors:

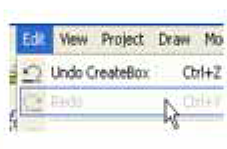
- Click **Tools** > **Stop Debug Logging**.

Working with Ansys Electronics Desktop Projects

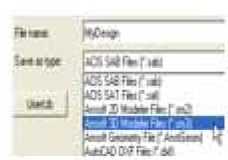
An Ansys Electronics Desktop project is a folder that includes one or more models, or *designs*. Each design ultimately includes a geometric model, its boundary conditions and material assignments, and field solution and post-processing information, or a schematic or netlist.

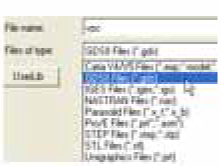

Project Files


Managing Projects

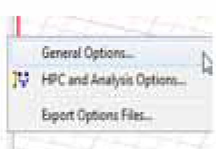

Undo and Redo

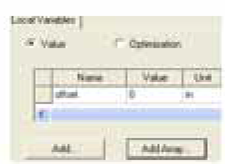

Validating Projects

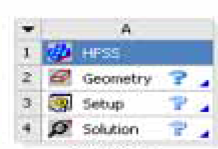

Exporting Models and Data


Importing Models and Data


Documentation and Printing


Setting Options


Variables


ANSYS Workbench Integration

A new project called *Projectn* is automatically created when the software is launched. By option, a design named Design*n* is automatically created for a new project. You can also open a new project by clicking **File** > **New**. In general, use the **File** menu commands to manage projects. If you move or change the names of files without using these commands, the software may not be able to find information necessary to solve the model.

> **Note:**
>
> Not all options and capabilities documented in this section apply to all design types.

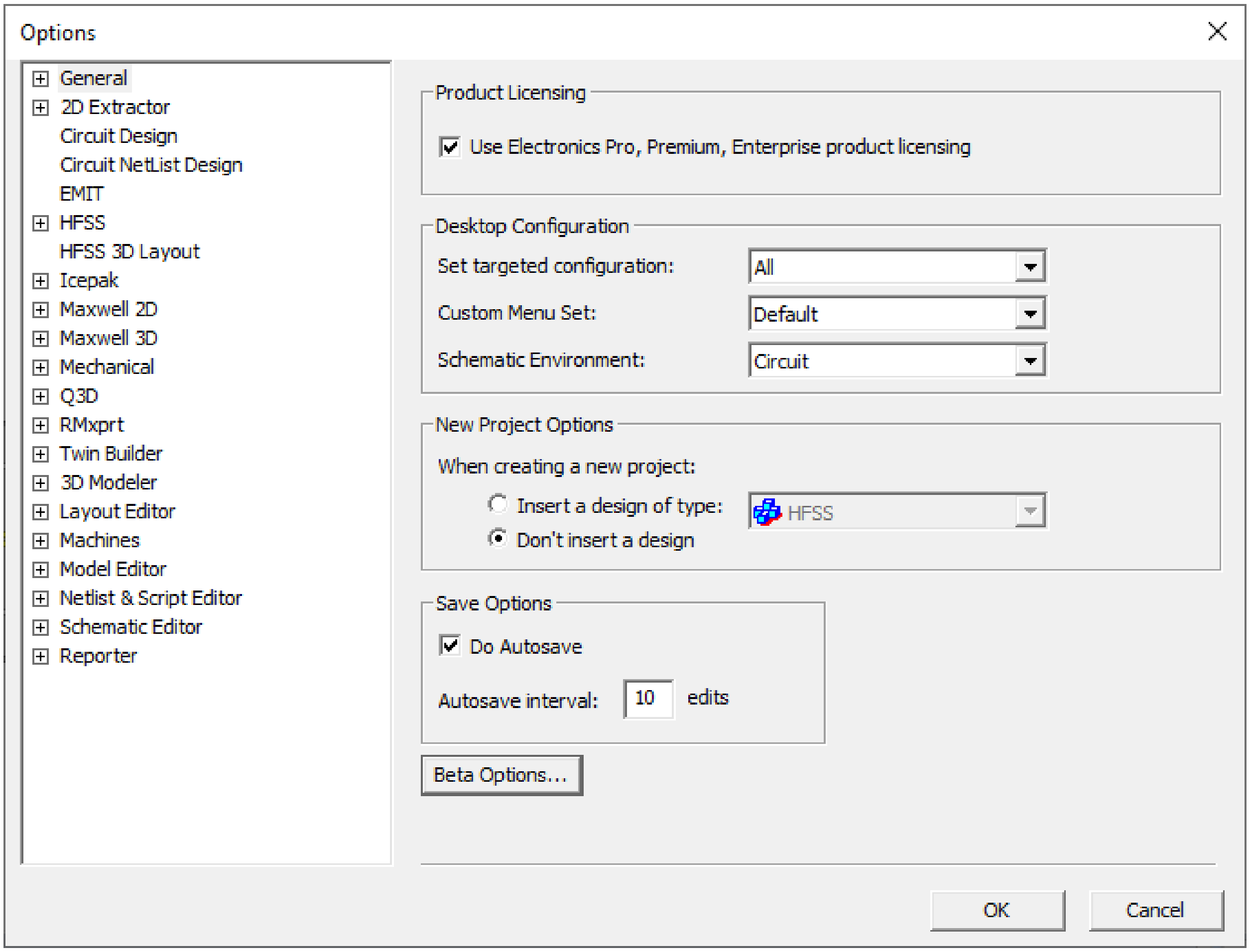
Ansys Electronics Desktop Files

When you create any project in Ansys Electronics Desktop, it is given an .aedt file extension and stored in the directory you specify. For Legacy projects, the transition is handled as follows:

- When opening a legacy project in interactive mode, a dialog box appears, informing you that the legacy file extension is no longer supported and that if you continue, the project will be converted to the new .aedt extension. If you agree, the project and results directory are renamed/moved to the new file extensions immediately. The read continues with the standard code for reading previous version projects.
- The Ansys Electronics Desktop 2025R1 version cannot open projects created with 2022R2 or earlier that were using ACIS modeler.
- Attempts to run batch solve or non-graphical with a legacy project return an error. There is no automatic/hidden conversion of file extensions. Note that the existing BatchSave command can be used to convert many projects to the new extension and version.
- Workbench integration has not been modified. It continues to open legacy projects without a warning and copies results from legacy to the new extension.

Setting Project Options

Tools > **Options** > **General Options** opens the **Options** window.



The left pane provides access to the following sets of options:

- General
- 2D Extractor
- Circuit Design
- Circuit Netlist Design
- HFSS
- HFSS 3D Layout
- Icepak
- Maxwell 2D
- Maxwell 3D
- Mechanical
- Q3D
- RMxprt

- Twin Builder
- 3D Modeler
- Layout Editor
- Machines
- Model Editor
- Netlist and Script Editor
- Schematic Editor
- Reporter

Use **Tools** > **HPC and Analysis Options** to specify active configuration per design type, queuing, distributed memory vendor, HPC licensing, select a MPI vendor, and whether to enable GPU for transient solves.

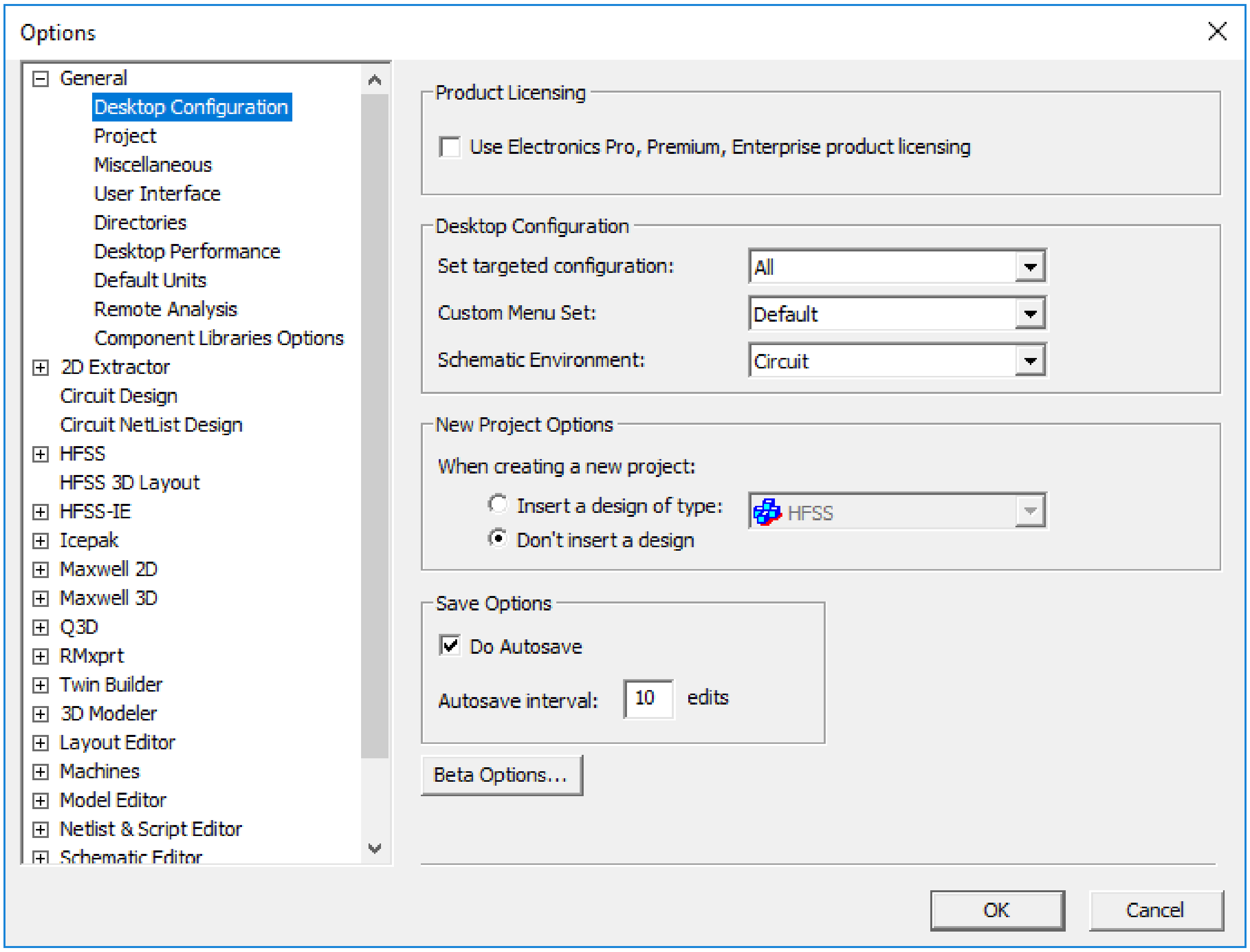
The **Tools** > **Options** > **Export Options Files** command writes xml files containing the Options settings at all levels to the specified directory. The **Tools** > **Options** > **Export Options** feature is intended to make it easier for different users to use Ansys Electromagnetics tools installed on shared directories or network drives. The Example Uses for Export Options Features section outlines some use cases enabled by this feature.

Setting General Options

To set general options in Ansys Electronics Desktop:

1. Click **Tools** > **Options** > **General Options**.

   The **Options** window opens with the Desktop Configuration options selected by default.



In the left pane, select the entries below **General** to display the associated options:

- Desktop Configuration
- Project
- Miscellaneous
- User Interface
- Directories
- Desktop Performance
- Default Units
- Remote Analysis
- Component Libraries Options

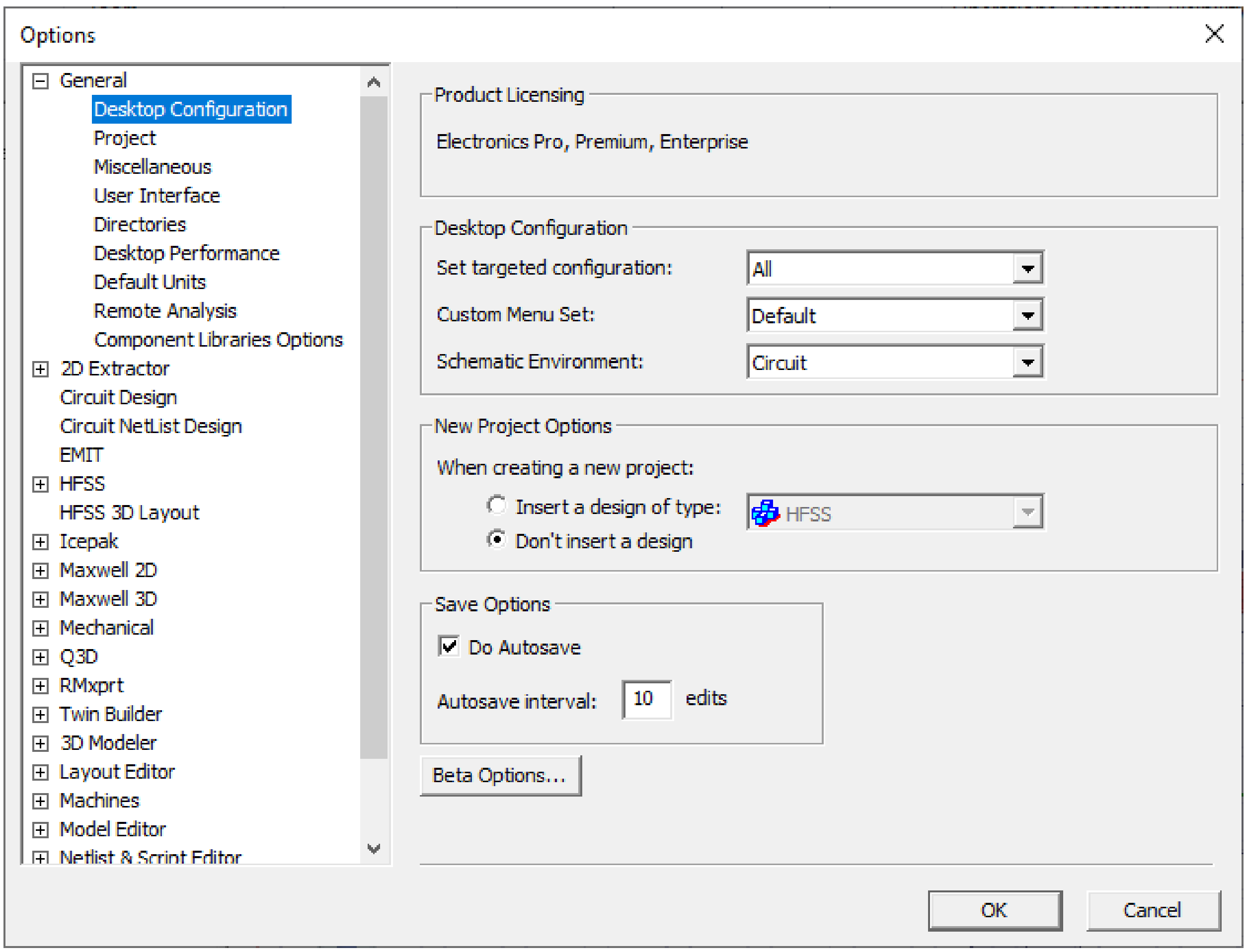
2. Click each entry and make the desired selections.

3. Click **OK**.

> **Note:**
>
> Not all option settings are applied immediately in the current design. Some settings take effect in newly created designs.

General Options: Desktop Configuration

Under General Options, **Desktop Configuration** options allow you to customize Electronics Desktop in a way that suits your work priorities. These include options for menus, new projects, save intervals, and beta features.



Ansys Electronics Desktop now uses Electronics Pro, Premium, Enterprise (PPE) product licensing. Legacy product licensing and DSO are no longer supported.

With PPE, HPC licensing is used to enable all cores, GPUs, and distributed tasks.

For more details, see Setting HPC and Analysis Options.

In the **Desktop Configuration** area, options include:

- **Set Targeted Configuration** – Choosing a Targeted Configuration changes the Custom Menu Set and Schematic Environment for your area of focus.

  Choices are:

  - **All** – Default option. Sets the **Custom Menu Set** to **Default**, the **Schematic Environment** to **Circuit**, and the default design type to **HFSS**.
  - **EM** – Electromagnetic focus. Sets the **Custom Menu Set** to **EM**, the **Schematic Environment** to **Twin Builder**, and the default design type to **Maxwell 3D**.
  - **RF** – Radio Frequency focus. Sets the **Custom Menu Set to** RF, the **Schematic Environment** to **Circuit**, and the default design type to **HFSS**.
  - **SI** – Signal Integrity focus. Sets the **Custom Menu Set** to **SI**, the **Schematic Environment** to **Circuit**, and the default design type to **HFSS 3D Layout**.
  - **Twin Builder** – Sets everything to default to **Twin Builder**.
- **Custom Menu Set** – Changes which menu options are available in Electronics Desktop.

  Choices are:

  - **Default** – All solvers appear on the **Project** menu.
  - **EM** – Only electromagnetics solvers appear on the **Project** menu.
  - **RF** – Only radio frequency solvers appear on the **Project** menu.
  - **RF.0** – All solvers appear on the **Project** menu. The **HFSS RF Setup** menu appears.
  - **SI** – Only signal integrity solvers appear on the **Project** menu.
  - **SI1.0** – Only signal integrity solvers appear on the **Project** menu. The **Import**, **Solution Setup**, **Automation**, and **Definitions** menus appear.
  - **SI2.0** – All solvers appear on the **Project** menu. The **HFSS SI Setup** menu appears.
  - **Twin Builder** – Only Twin Builder appears on the **Project** menu.
- **Schematic Environment** – Select **Circuit**, **Twin Builder**, or **Maxwell**.

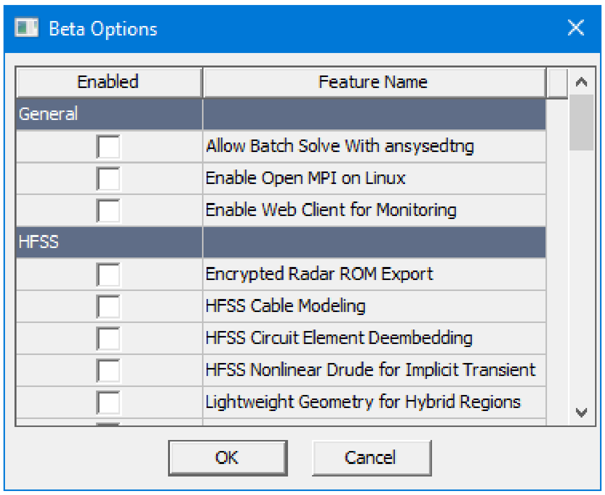
In the **New Project Options** area, options include:

- **When Creating a New Project** – Select whether to insert a design when creating a new project. If you choose to insert a design by default, use the drop-down menu to select the default design type.

In the **Save Options** area, options include:

- **Do Autosave** – Electronics Desktop has autosave enabled by default. Deselect this option to disable it.
- **Autosave Interval** – When **Do Autosave** is enabled, select the number of edits at which Electronics Desktop autosaves.

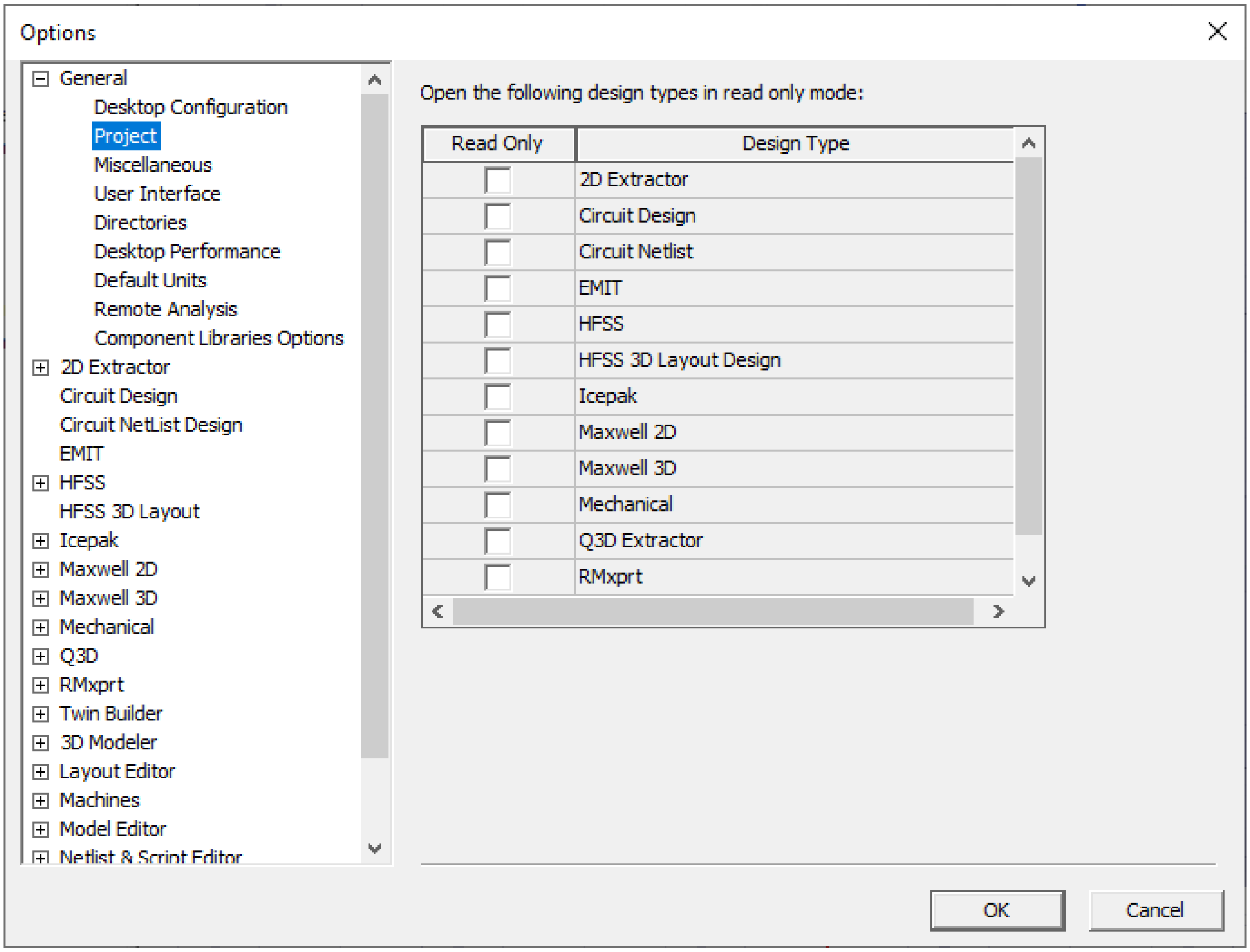
Click the **Beta Options** button to open a window listing beta options. You may need to scroll or size the window to view all options.



From there, enable or disable options, and click **OK**. You will need to restart to enable your selections.

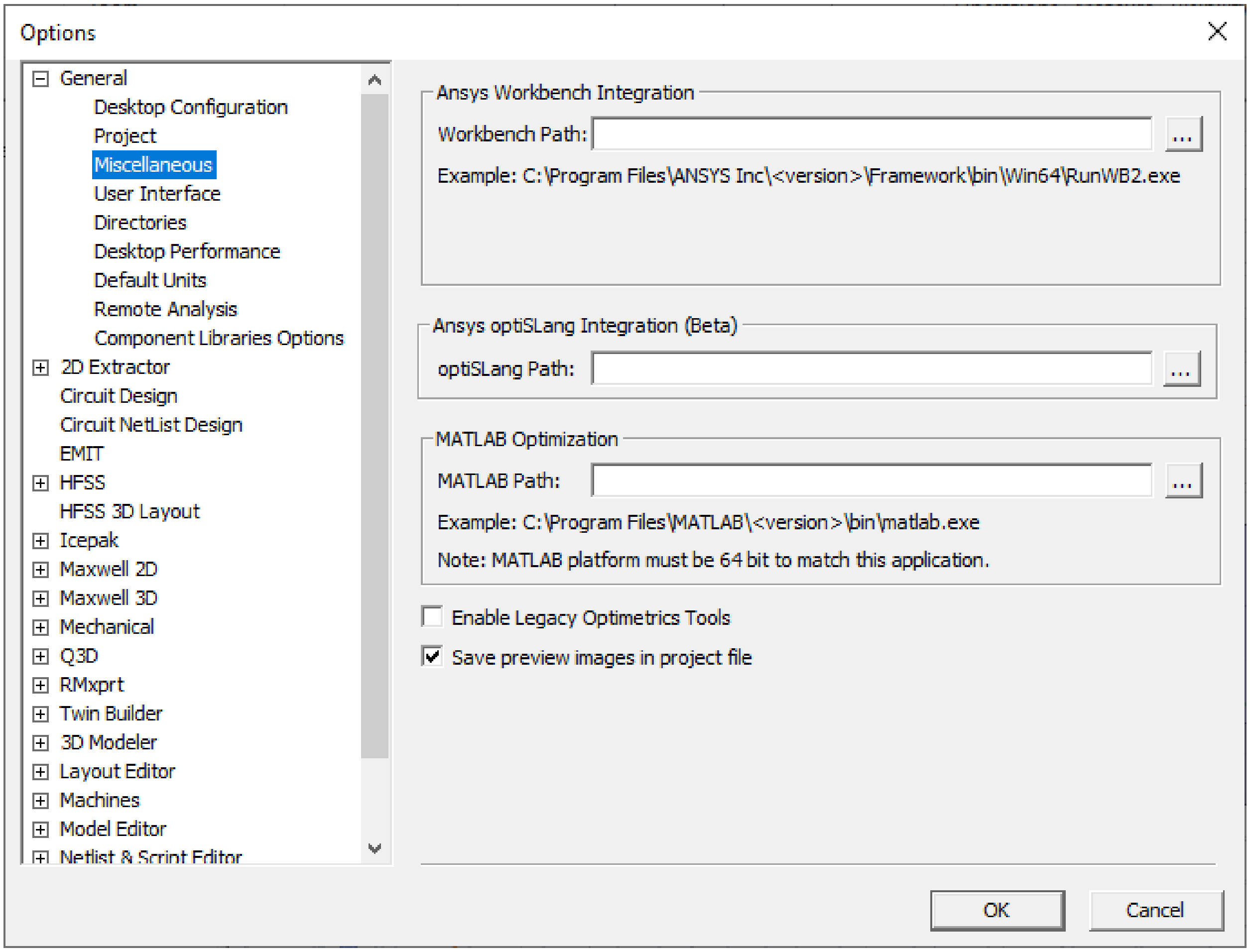
General Options: Project Options

Under General Options, **Project** options allow you to select design types to open in read-only mode. This can help prevent accidental changes.



General Options: Miscellaneous Options

Under General Options, **Miscellaneous** options allow you to specify paths for Ansys Workbench Integration, optiSLang path, MATLAB Optimization and to enable Legacy Optimetrics Tools.



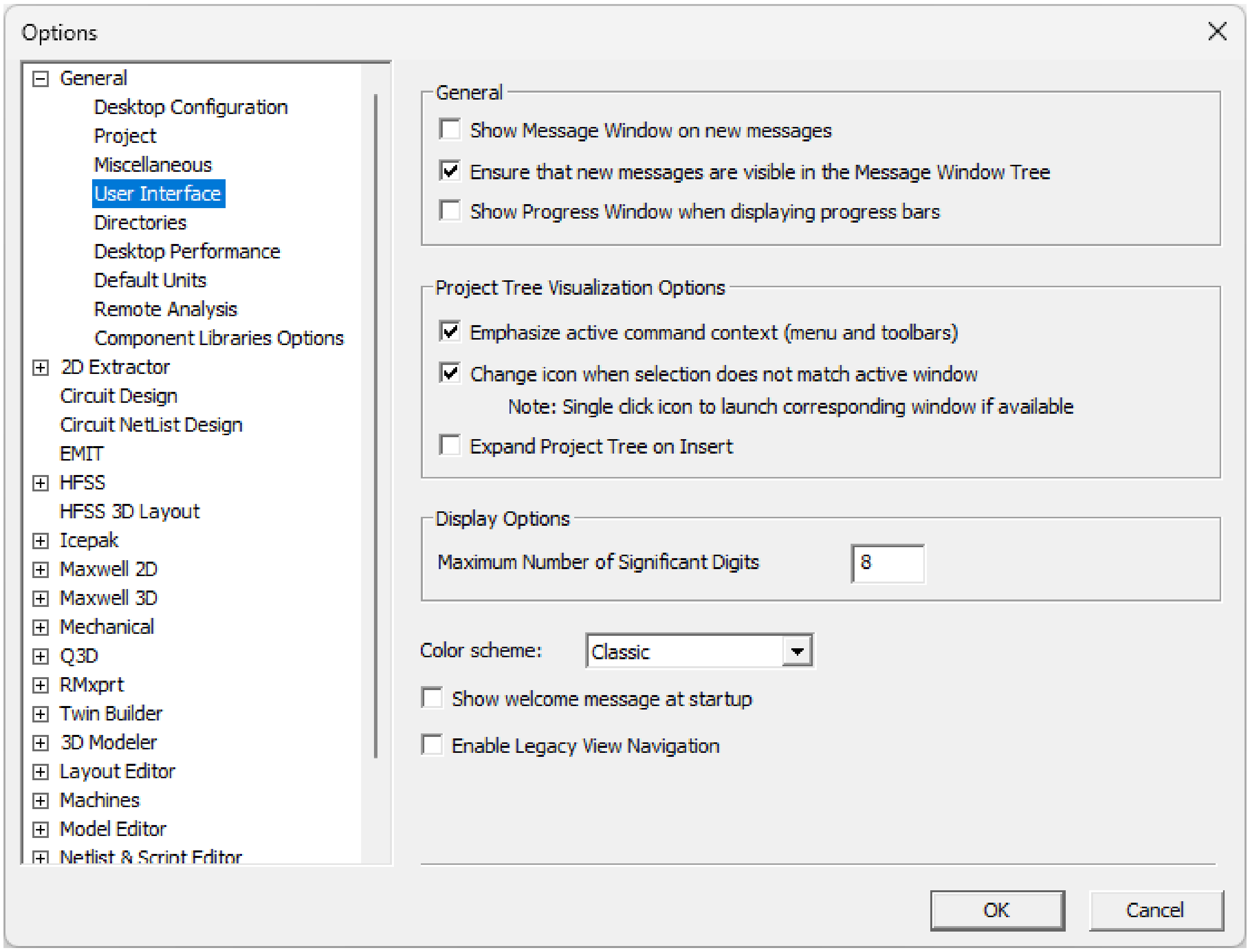
Ansys Workbench Integration allows you to specify the path to your Ansys Workbench installation. This path can be used in Optimetrics for connecting to the Design Xplorer.

MATLAB Optimization allows you to specify the path to your installation of MATLAB. MATLAB can be used as an Optimizer.

> **Note:**
>
> The platform for MATLAB and Ansys Electronics Desktop must match (for example, a 64 bit version of MATLAB).

General Options: User Interface Options

Under General Options, **User Interface** options allow you to change how Electronics Desktop displays messages, command text, the project tree, welcome messages, and more.



In the **General** area, options include:

- **Show Message Window on new messages** – when selected, the message window automatically opens if a message arrives.
- **Ensure that new messages are visible in the Message Window Tree** – when selected, the message window expands as needed to display messages.
- **Show Progress Window when displaying progress bars** – when selected, the progress window automatically opens while simulations are in progress.

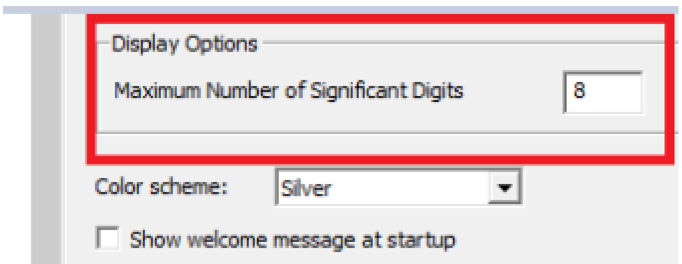
In the **Project Tree Visualization Options** area, options include:

- **Emphasize active command context** – when selected, active elements in the Project Tree display in bold text.
- **Change icon when selection does not match active window** – when selected, a small, window-shaped overlay icon displays in the corner of the selected Project Tree element. This icon changes when the data in the active window is unrelated to the

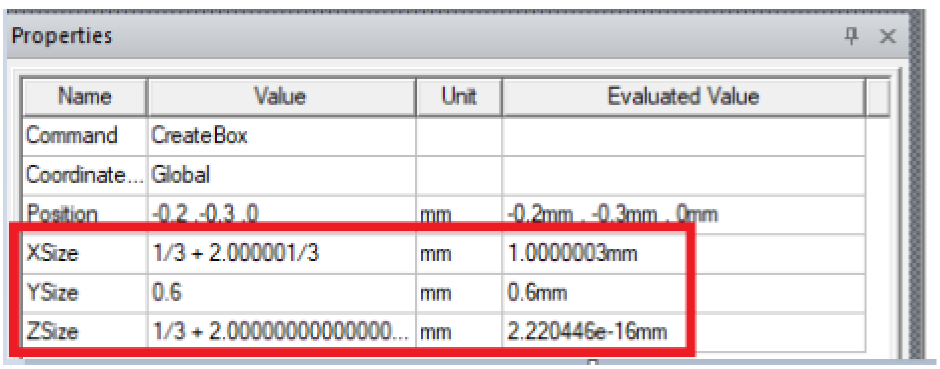
selected project item (data affecting the same model is considered to be related). Clicking the icon opens the window and brings it into focus.

- **Expand Project Tree on Insert** – when selected, the Project Tree automatically expands when you insert a new design.

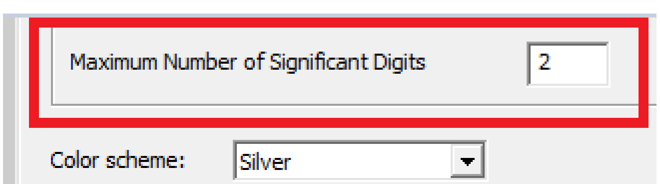
In the **Display Options** area, specify the **Maximum Number of Significant Digits** to display. The default is 8 and the maximum is 20. This affects the digits displayed in the **Solutions** dialog box, evaluated variable values, Animation dialog boxes, Optimetrics, Reports, and so forth. For example, here is the default.



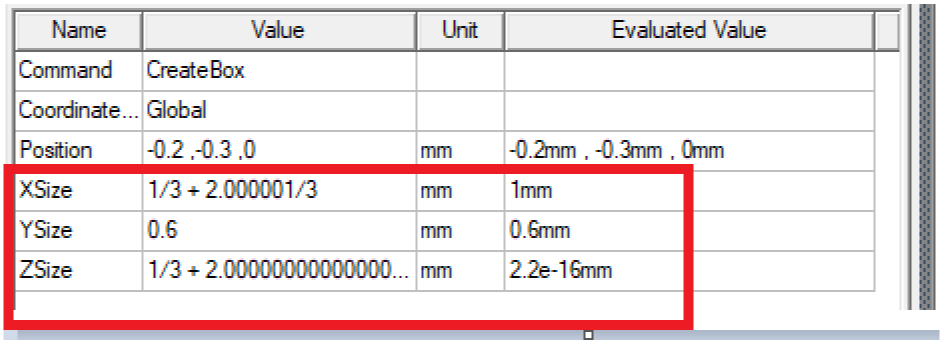
So some evaluated variable values with significant digits appear as follows.



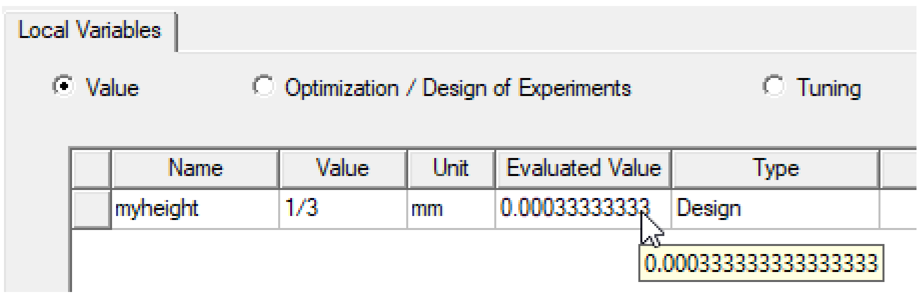
Change Precision = 2. This lets you display near-zero values in the Properties window as 0.



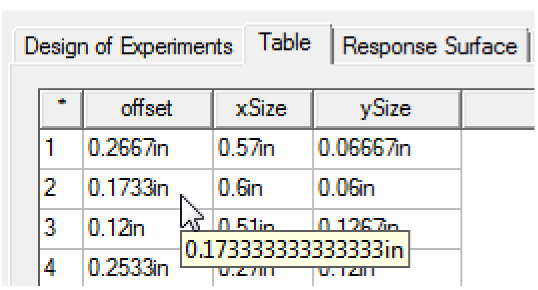
You can see that XSize is now rounded off to cleanly display 1mm. The decimal part of ZSize also shows cleaner value, but it is not rounded off to zero.



You can still see the full precision values in tooltips by holding a cursor over the displayed value.

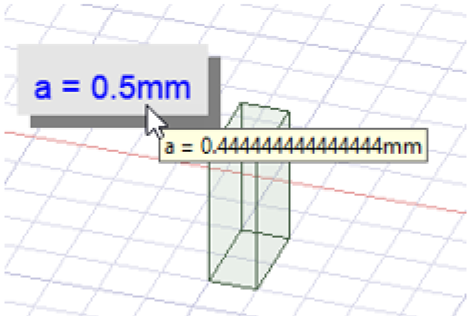


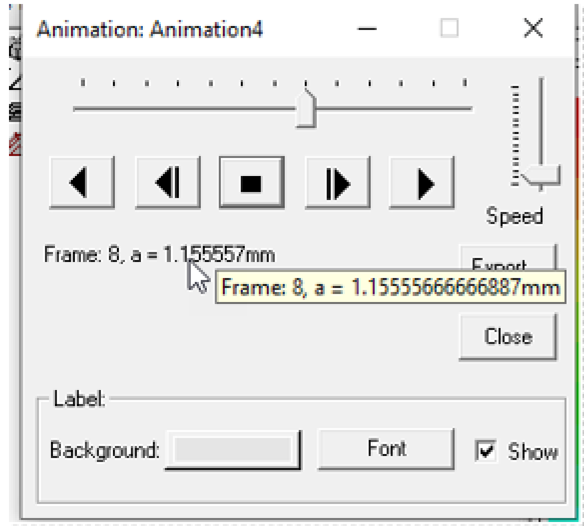
In the case of variable values, if you have assigned more significant digits, you will see these when editing the variable value. In the case of table displays of values, the tooltip display shows all available digits when the mouse pointer is over a result:



If you set the **Maximum Number of Significant Digits** to a lower value, Change Precision = 2. You can see that XSize is now rounded off to cleanly display 1mm. The decimal part of ZSize also shows cleaner value, but it is not rounded off to zero.

The tooltip functions to show internal digits in throughout the Ansys Electronics Desktop interface.





Select a **Color Scheme**. The choices are **Light (Beta)**, **Dark (Beta)**, or **Classic**: See Choosing a Color Scheme for more information and samples of each scheme.

Select **Show welcome message at startup** if you want to see a welcome message when Electronics Desktop opens.

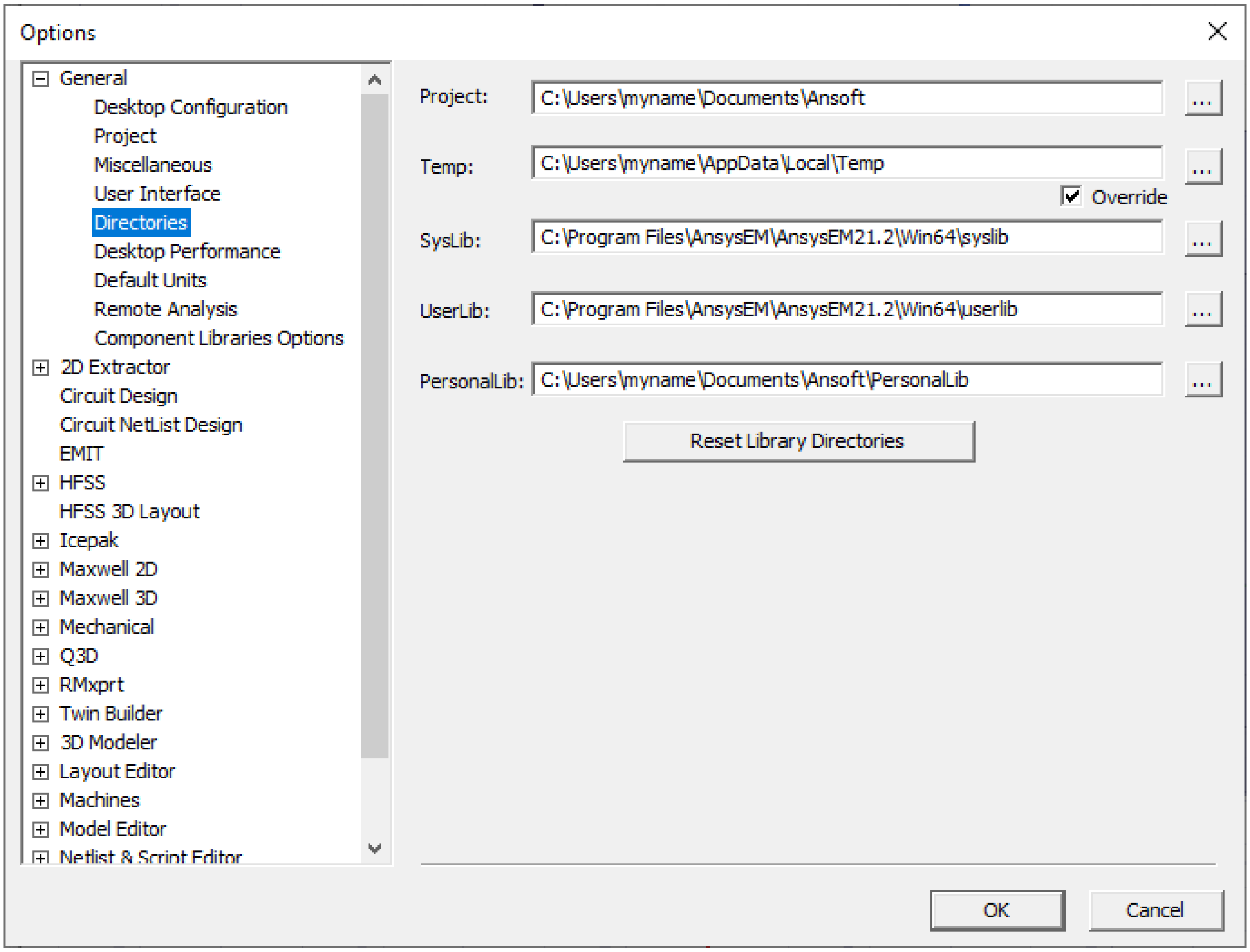
Enable legacy view navigation enables you to choose between two schemes for view navigation keyboard shortcuts and mouse button assignments, as follows:

- **Cleared:** Use only the view navigation mouse-button and hotkey assignments introduced in Ansys Electronics Desktop version 2024 R1 and applicable to subsequent versions too. Legacy (2023 R2 and earlier) mouse-button and hotkey assignments will *not* be recognized when this option is cleared.
- **Selected:** Both the legacy (2023 R2 and earlier) and current (2024 R1 and newer) mouse-button and hotkey assignments are supported, and either can be used for view navigation.

See Choosing the View Navigation Options for more information, including a detailed comparison of the two schemes.

General Options: Directories Options

Under General Options, **Directories** options allow you to get the paths for Project, Temp, SysLib, UserLib, and PersonalLib directories.



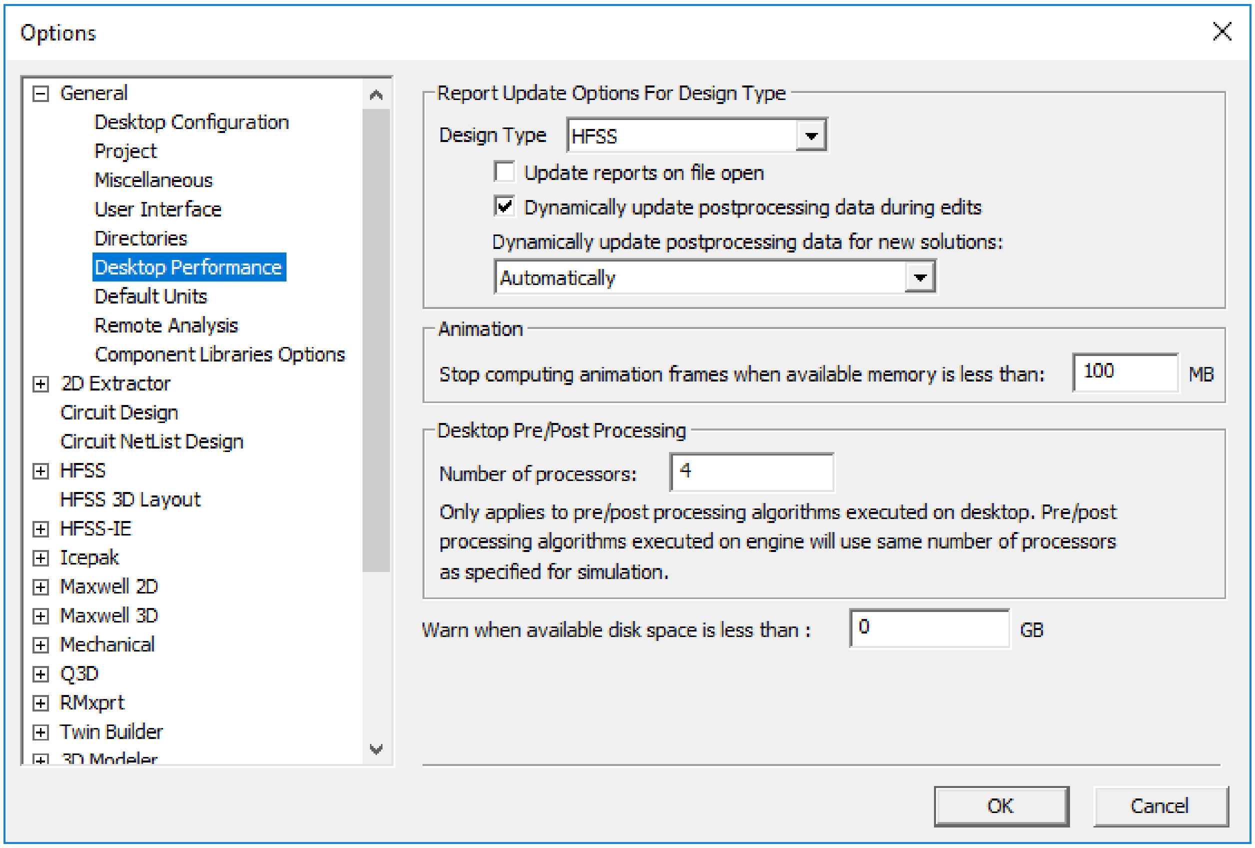
Use the **[...]** buttons to browse to paths and click **OK**. You will need to select **Override** to change the Temp directory.

If you need to restore defaults, click **Reset Library Directories**.

> **Note:**
>
> Your changes will be reflected in the User Defined Primitives menu on next startup.

General Options: Desktop Performance

Under General Options, **Desktop Performance** options allow you to change settings for animation, pre/post-processing, disk space warnings, and updating reports.



These options are set on the **Desktop Performance** panel under **General** in the **Options** window.

In the **Report Update Options for Design Type** area, select a **Design Type**. For each type, you can set the following options:

- **Update Reports on File Open** – when selected, reports are automatically updated whenever an existing file with solution data is opened for viewing/editing.
- **Dynamically Update Postprocessing Data During Edits** – when selected, report plots and overlays update as you edit their parameters.
- **Dynamically Update Postprocessing Data for New Solutions** – Because updating reports during analysis can impact solution time, you can specify how often your reports are updated:
  - **After Each Variation** – updates reports after analysis of each variation has been completed. Used for an Optimetric or parametric analysis.
  - **Automatically** – balances report and field plot updating with solution time.

    For Adaptive Passes, plots update at the end of each solution pass. For Last Adaptive or Transient, plots update at the end of the transient or adaptive solution.

For example, reports may be updated after each adaptive pass but field plots will not be updated until the solution is complete.

- **Immediately** – updates reports and plots as soon as data comes from the solver.

  This option will have the greatest impact on overall solution time, but affords the fastest updates to reports and field plots. Caution should be used in selecting this option. Some types of reports and field plots may take a long time to update, especially as mesh size increases.

- **Never** – updates reports only upon manual intervention. This prevents updates from impacting solution time.
- **On Completion** – updates reports once, when the solution completes.

> **Note:**
>
> Updates done on completion are done after the solve has been completed, and the time for that update is not included in the simulation profile.

In the **Animation** area, you can elect to **Stop computing animation frames when available memory is less than** a specified value, in MB (the default is 100MB). This setting is used to prevent problems related to low memory should an animation require large memory allocation.

In the **Desktop Pre/Post Processing** area, you can specify the **Number of Processors**. This option only affects pre- and post-processing (not solve or simulation). Pre-processing algorithms can take advantage of multiple processors for visualization and faceting of 3D models, model validation for 3D products, auto net identification for Q3D, and more. The default value is determined by the number of logical processors on the machine running Electronics Desktop:

- The default core usage per desktop session (UI + solve) is set to 2/3 of the logical processors on the machine.
- The cores for default local config is max{4, 1/3 of the logical processors on the machine}
- The default number of processors for pre/post is min{1, 2/3 of the logical processors - default_cores_for_local_config}
- When the core usage per desktop session cannot be evenly distributed between solve and pre/post, more cores will be assigned to solve.

Example 1:

- logical processors = 24
- cores usage per desktop session = 16
- local hpc set to 4 cores
- processors for pre/post set to 12

Example 2:

- logical processors = 20
- cores usage per desktop session = 13
- local hpc set to 4 cores
- processors for pre/post set to 9

Example 3:

- logical processors = 5
- cores usage per desktop session = 3
- local hpc set to 2
- processors for pre/post set to 1

You can also elect to **Warn when available disk space is less than** a specified value, in GB.

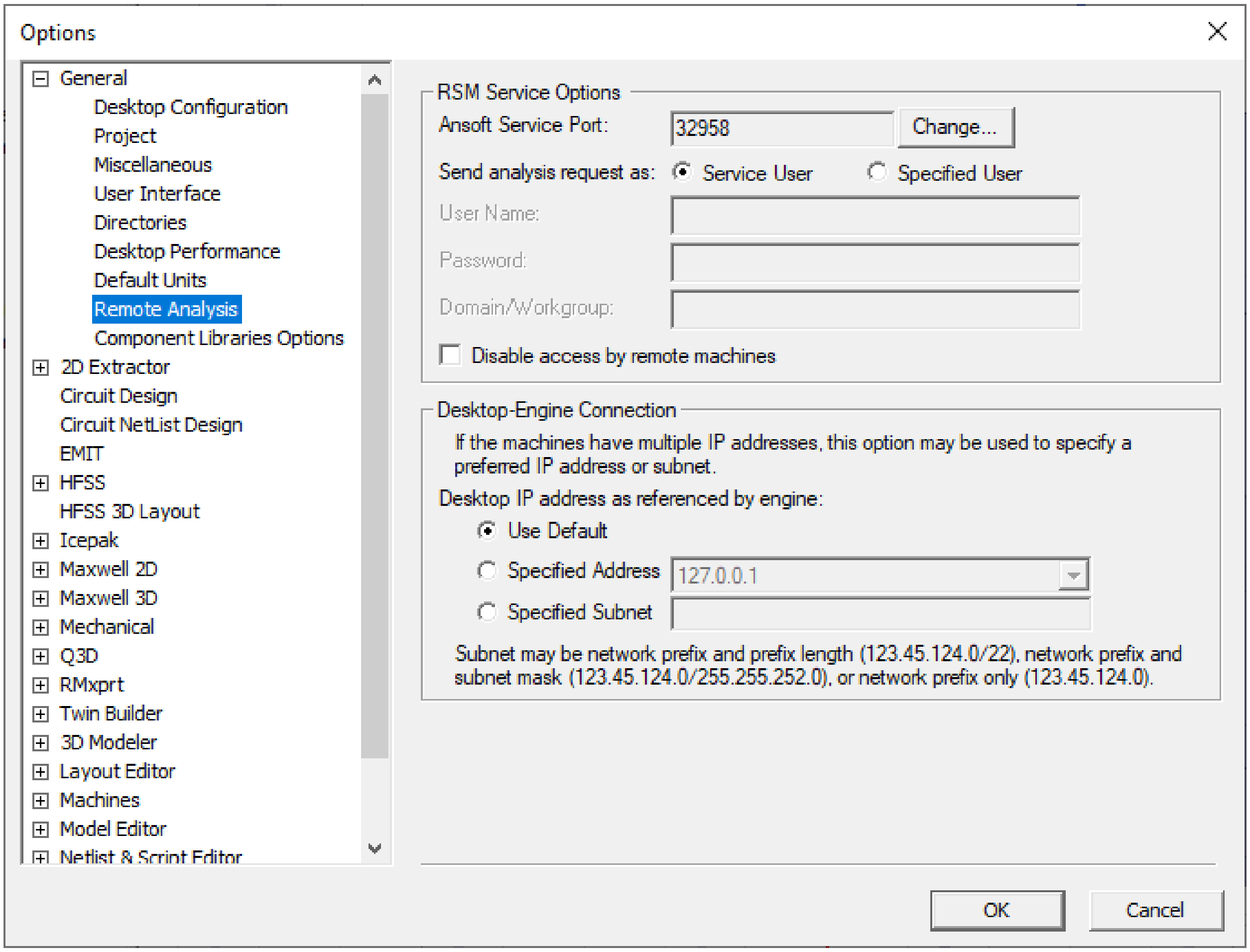
General Options: Default Units

Under General Options, **Default Units** options allow you to set default units for the following:

- Length
- Angle
- Time
- Temperature
- Torque
- Magnetic Induction
- Pressure
- Frequency
- Power
- Voltage
- Current
- Speed
- Mass
- Conductance
- Resistance
- Inductance
- Capacitance
- Force
- Angular Speed
- Magnetic Field Strength

General Options: Remote Analysis Options

Under General Options, **Remote Analysis** options allow you to launch all analyses as a service or specified user rather than as the current user.



In the **RSM Service Options** area, options include:

- **Ansoft Service Port** – Click **Change** to update the port number. Ansys Electromagnetics RSM Service should be running on this port for all distributed machines.
- **Send Analysis Request As** – Select either **Service User** or **Specified User**. Selecting **Specified User** enables the **User Name**, **Password**, and **Domain/Workgroup** fields.

> **Note:**
>
> If any of the remote machines is Linux-based, you must specify the current user.

- **Disable Access By Remote Machines** – If desired, select to disable access for remote machines.

When multiple IP addresses are available, the **Desktop-Engine Connection** area allows you to specify the preferred IP address for communication:

- **Use Default** – your system's default IP address.
- **Specified Address** – an IP address you specify.
- **Specified Subnet** – a subnet you specify. Subnet may be network prefix and prefix length (123.12.123.0/22), network prefix and subnet mask (123.12.123.0/255.255.252.0), or network prefix only (123.23.123.0).

Changing the Listening Port used by Ansys RSM Service

To change the listening port used by the RSM Service, you must change the ansoftrsmservice.cfg file, as follows:

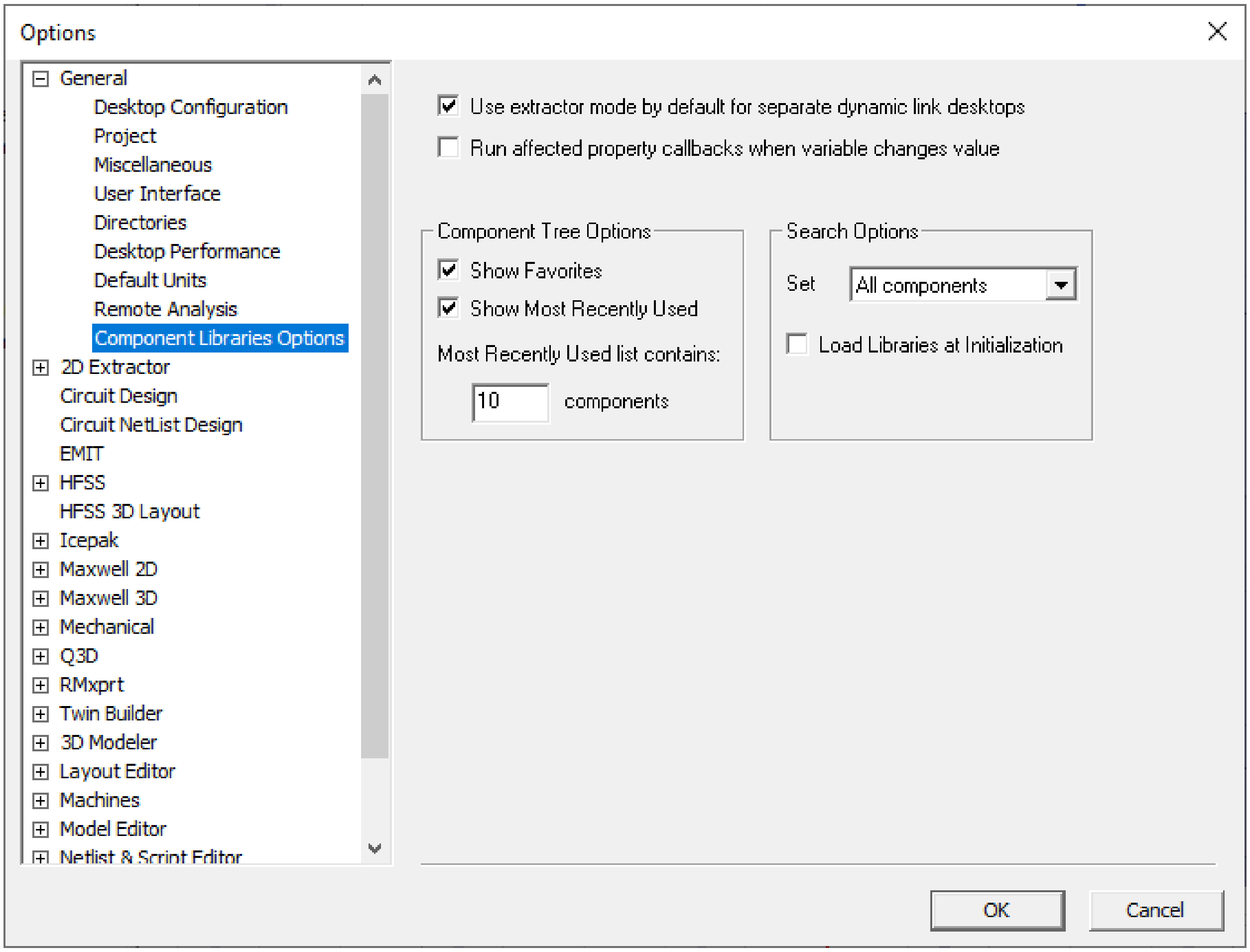
Specify the ListenPort within a 'CommDetails' block, which must be within a 'Default:CommDetails' block, which must be within the top level block of the file (the 'AnsoftCOMDaemon' block). The following example changes the listen port from 32958 to 32957, with these blocks at the beginning of the file:

```
$begin 'AnsoftCOMDaemon'
  $begin 'Default:CommDetails'
    $begin 'CommDetails'
      ListenPort='32957'
    $end 'CommDetails'
  $end 'Default:CommDetails'
  . . . .
$end 'AnsoftCOMDaemon'
```

For the second level block, ensure that there is a single colon character and no spaces or tabs separating the two parts of the block name 'Default:CommDetails'. The third level block, with name 'CommDetails' is also required. Use caution when editing this file by hand, because any typos in the block or value names may cause the data to be ignored.

General Options: Component Libraries Options

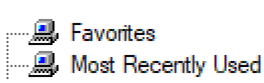
Under General Options, **Component Libraries** options allow you to change how Electronics Desktop handles components.



You can elect to **Use extractor mode by default for separate dynamic link desktops**, or to **Run affected property callbacks when variable changes value**.

In the **Component Tree Options** area, options include:

- **Show Favorites** – enables the **Favorites** folder in the **Component Libraries** tree.
- **Show Most Recently Used** – enables the **Most Recently Used** folder in the **Component Libraries** tree. Use the field to specify the number of components shown in the **Most Recently Used** folder. The default is 10.



In the **Search Options** area, options include:

- **Set** – specifies the search set when searching the Component Libraries. Choose to search **All components**, **Current list only**, or **Append to current list**.

- **Load Libraries at Initialization** – this option can slow initialization.

Setting HPC and Analysis Options

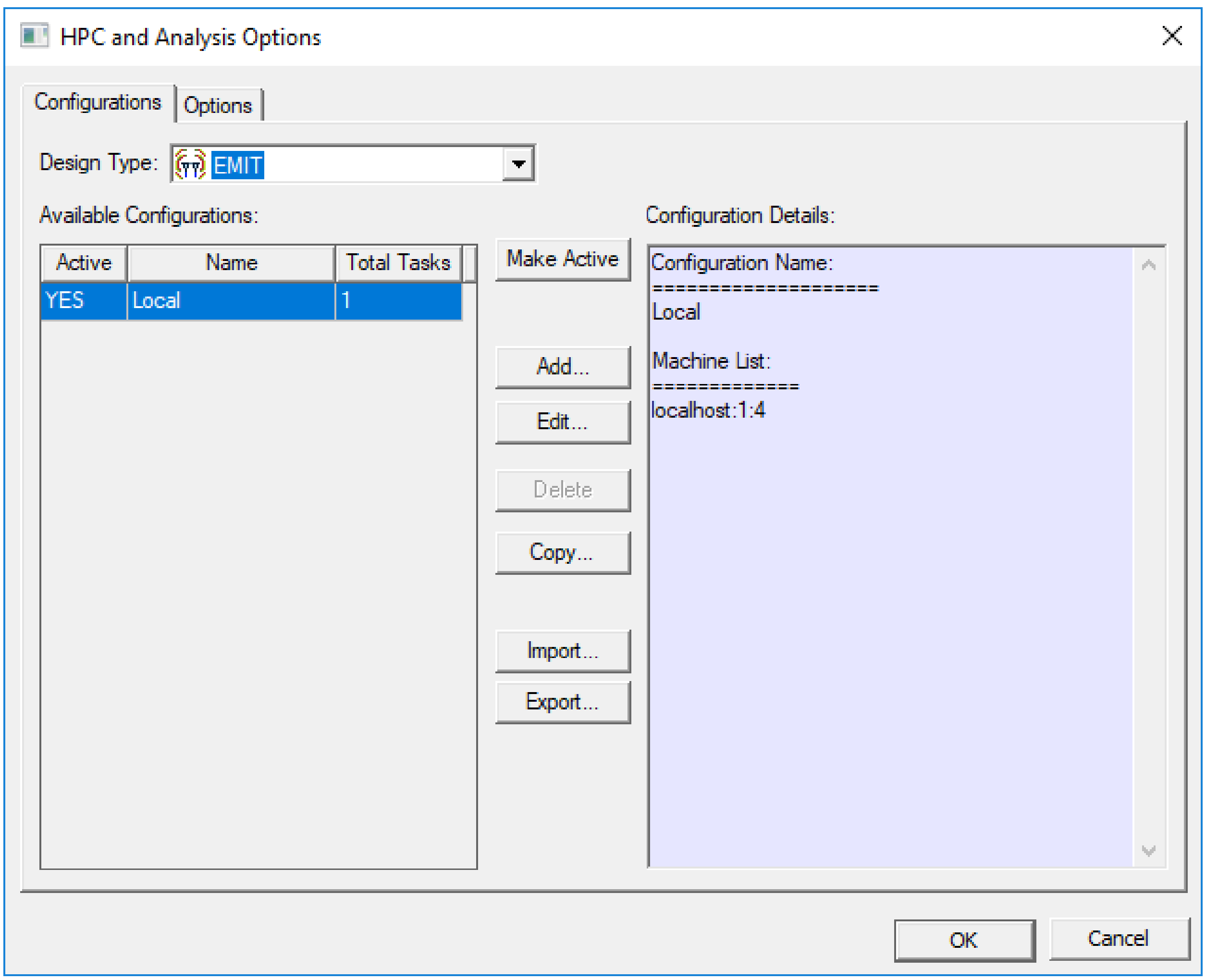
All analysis parameters are accessed via a single window. The machine list and options settings have been integrated into analysis configurations. The default configuration is for solving on a single, local machine. You can create many analysis configurations for remote and distributed solutions, and switch between them depending on the job being solved. Multiprocessing has been integrated into the machine lists.

To set HPC and Analysis Options:

1. Click **Tools** > **Options** > **HPC and Analysis Options**.

   You can also access HPC and Analysis Options using the **HPC Options** icon on the **Simulation** ribbon.

   The **HPC and Analysis Options** window appears, displaying the **Configurations** tab.

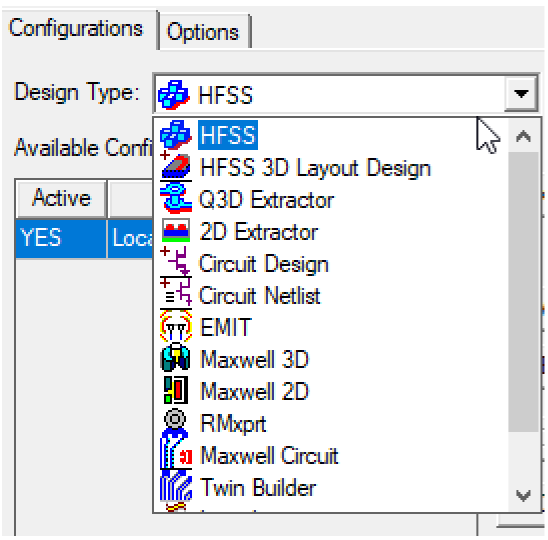


Configurations Tab

Available Configurations are described by **Name**, whether or not they are **Active**, and the **Total Tasks** the configuration can execute.Selecting a configuration from the list displays the details of that configuration in the **Configuration Details** panel.

From the **Configurations** tab, select the **Design Type** to display a list of available configurations for that type.

Configurations must be defined for all design types separately. To use similar analysis parameters for different design types, create separate analysis configurations for each design type. The active configuration is used when solving an analysis for that design type.



Selecting an Available Configuration

To activate a configuration, select it from the **Available Configurations** list and click **Make Active**. The active configuration will be indicated by a **YES** in the **Active** column.

Additional options include:

- **Add** – launches a dialog box to create a new analysis configuration.
- **Edit** – launches a dialog box to .
- **Delete** – deletes the currently selected analysis configuration(s).

> **Note:**
>
> You cannot delete the Local configuration.

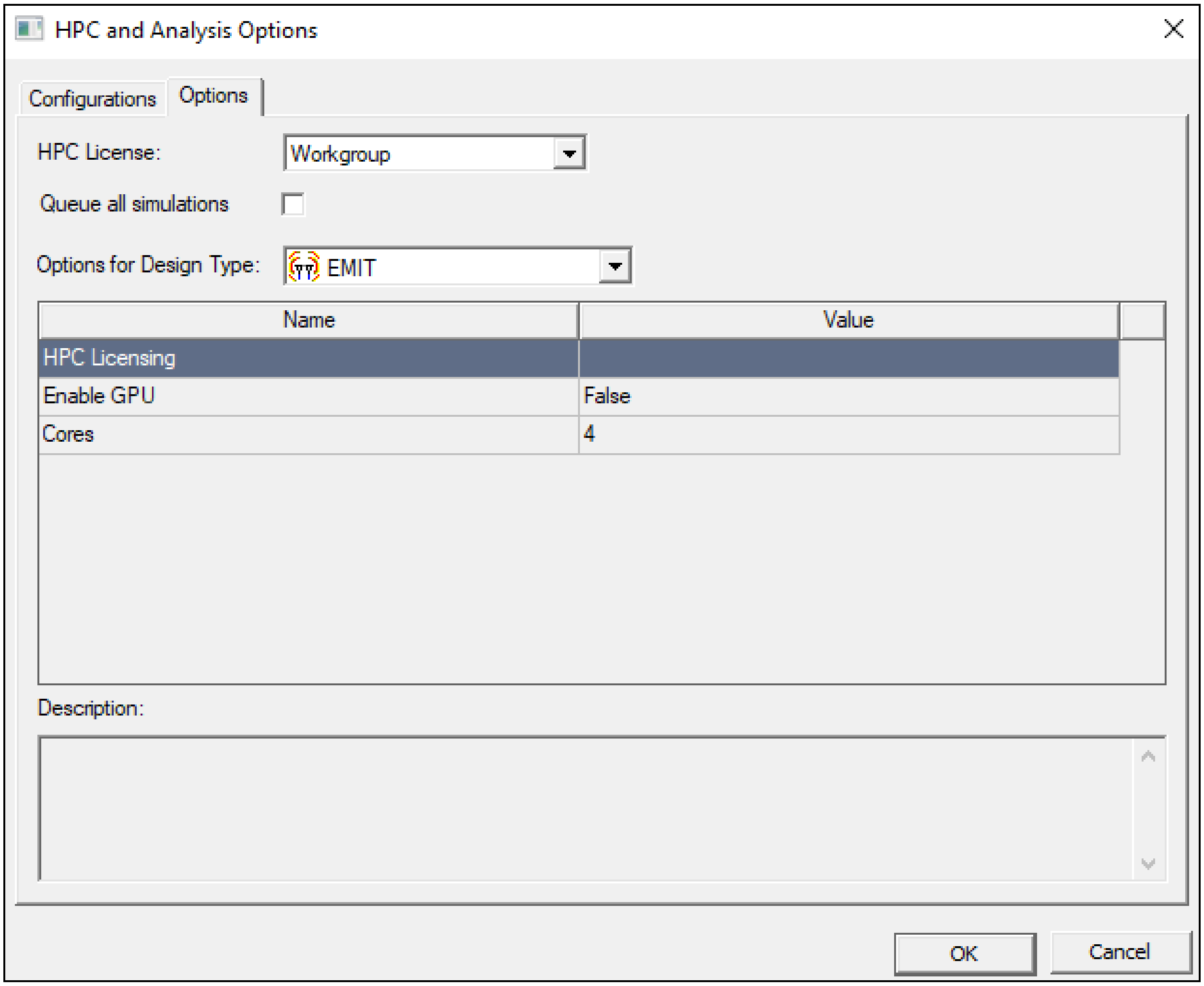
- **Copy** – creates a new analysis configuration, and . If the dialog box is canceled, the new analysis configuration is not created.
- **Import** – allows you to import an *.acf file to create an analysis configuration.

> **Note:**
>
> Importing analysis configurations always adds the imported analysis configurations to the current design type. If there is a name conflict between an imported analysis configuration and an existing analysis configuration, the imported configuration is renamed and a notification appears.

- **Export** – allows you to export the selected analysis configurations to an *.acf file. You can then import the configurations into a different design type, or import them on a different machine.

Options Tab

The **Options** tab in the **HPC and Analysis Options** dialog box contains general and product-specific settings.



> **Important:**
>
> Available options vary by design type. Certain items discussed on this page may not be visible, depending on the currently selected design type.

These options are not specified for, or saved as part of, the current analysis configuration. Instead, they are global and are always in effect for the given design type when both of the following conditions are true:

- A design of the matching design type is being solved.
- You have not specified corresponding overriding batch options on the command line.

From the **Options** tab, you can:

- Choose the HPC license type
- Enable queuing
- Specify Distributed Memory settings (for example, MPI for certain solvers)
- Set licensing options
- Enable GPU (for Transient, Matrix and SBR+ solves)
- Set the Default Process Priority

For **HPC License**, select **Auto**, **Workgroup** or **Pack**.

HPC licensing enables the use of cores and GPUs to accelerate simulations. In general, each core requires one unit of HPC, while each GPU requires eight units. The selected HPC license type determines which license is used, and how units of HPC are converted to license counts.

- **Workgroup** (formerly "pool") – One HPC workgroup license enables one unit of HPC.
- **Pack** – One HPC pack license enables eight units of HPC. Additional packs multiply by four, enabling 32, 128, 512,... , in the context of a single simulation.
- **Auto** – delegate the choice of Workgroup or Pack licensing to the Ansys Licensing Settings tool which is a separate application installed along with Electronics Desktop. When set to Auto, the default is to use HPC Pack first, then Workgroup.

Electronics Desktop products include four units of HPC for each licensed simulation. This means that up to four units can be used without requiring HPC licenses; license counting begins with the fifth unit. For example, a simulation that uses 36 cores requires 32 HPC units after subtracting the four included cores. This simulation will check out 32 HPC workgroup licenses, or two HPC pack licenses.

HPC licenses enable all parallel and distributed simulations, including distributed variations. Distributed variations require a single set of solver licenses, plus HPC to enable the variations.

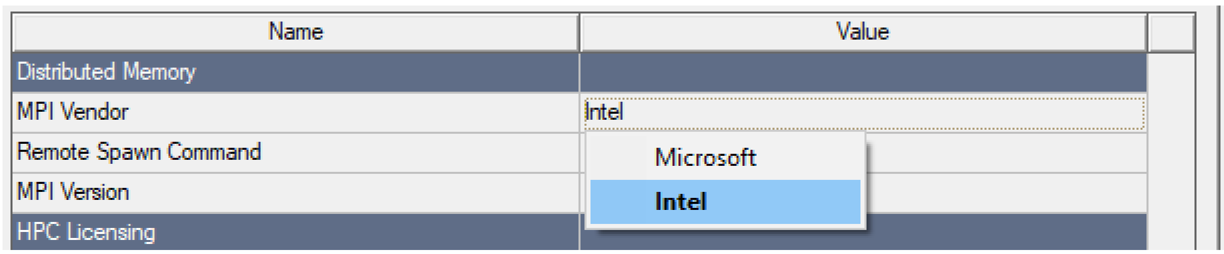
For HPC Workgroup, distributing *N* variations requires 8*(*N*–1) workgroup licenses and, together with the solver licenses, enables up to four HPC units per variation. Each additional set of *N* workgroup licenses will enable one additional HPC unit per variation. For HPC Pack, distributing *N* variations requires *N*–1 pack licenses and, together with the solver licenses, enables up to four HPC units per variation. Each additional set of *N* pack licenses will enable 8, 32, 128,... additional HPC units per variation.

Ansys licensing supports distributed simulations when Electronics Desktop is called from other Ansys tools, such as optiSLang and Workbench. In such cases, distributed design points (variations) generally use HPC counts as described above.

> **Note:**
>
> Licensing for some calling products may include some distributed design points, in which case the total required HPC will be reduced.

If the **Queue all simulations** check box is selected, the Desktop queues any active simulations for design types that have **Save before solving** turned off in the **General Options** and then processes them in order. You can view and change the queue by using Show Queued Simulations.

To configure options:



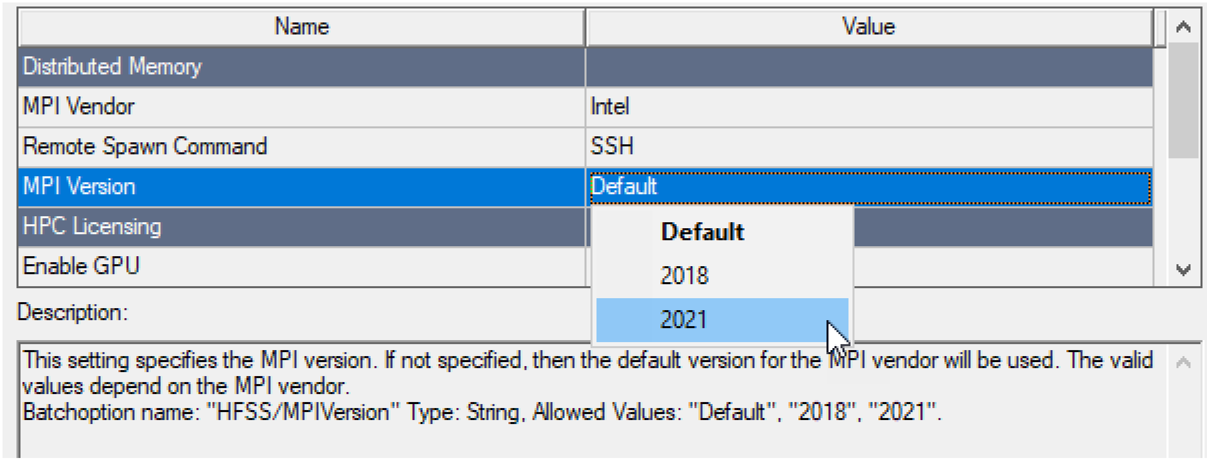
The solvers use the industry standard Message Passing Interface (MPI) and can perform solutions that distribute memory use across machines in a cluster or network. Memory used by the MPI-enabled solver is therefore limited by the set of machines that is available rather than the shared memory available on any single machine. This allows you to simulate larger structures and to optimally reconfigure the cluster of machines for the problem at hand. For solving on a single machine, MPI is not required, nor does it provide an advantage.

To use the distributed memory solution you will need to install MPI software from a supported third party vendor on all the machines you intend to use.

Depending on the MPI vendor, you may need to set passwords for authentication on the machines. Settings within each design type turn on distributed memory solutions and define the list of machines you intend to use.

You can specify the MPI version. If not specified, the default version for the MPI vendor will be used. This setting is ignored if there is only one supported version for the selected MPI Vendor. Multiple versions are only supported for Intel MPI, as follows:

- The value "Default" indicates that the default Intel MPI version should be used. This is Intel MPI 2021 in most cases.
- The value "2018" indicates that Intel MPI 2018 should be used.
- The value "2021" indicates that Intel MPI 2021 should be used.



Also see Setting Intel MPI Interconnect for more details.

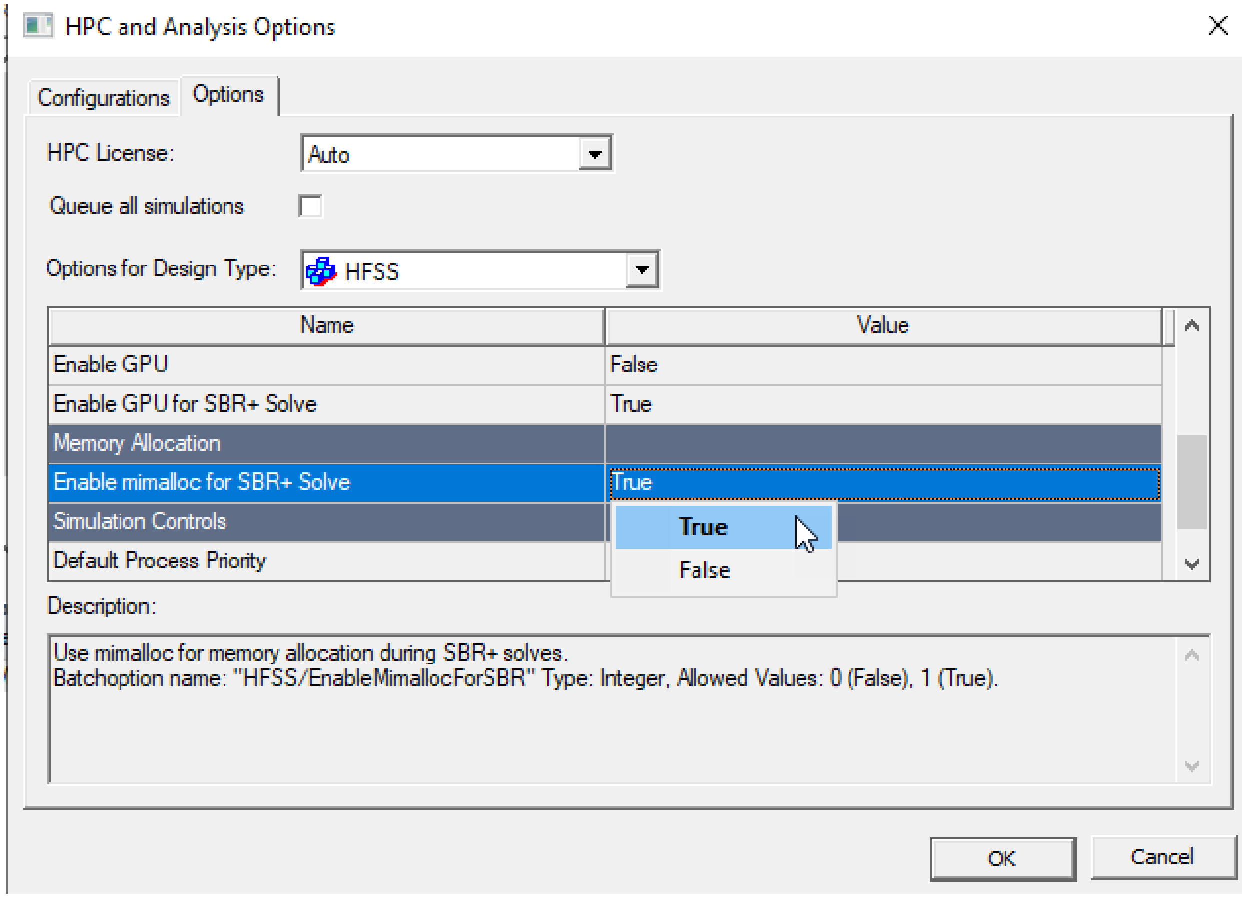
InfiniBand Support for Windows

By default, MPI vendors use the fastest interconnect by default (typically InfiniBand is faster than Ethernet). If you want to override the default behavior and force the use of Ethernet, you can set the ANSOFT_MPI_INTERCONNECT environment variable to "eth" for the job. Also see Setting Intel MPI Interconnect for more details.

1. For Linux authentication, specify the Remote Spawn command as RSH or SSH (the default).

   For details on the requirements for GPU use, see Transient GPU Acceleration

2. **Memory Allocation**. The HFSS SBR+ solver leverages mimalloc by default on Windows to deliver better performance, especially in multi-threaded environments. On certain systems, this might cause the HFSS SBR+ not to execute. In this case, users can use the default Windows memory allocation by setting the "Enable mimalloc for SBR+ solve" option to "False"."



3. **Simulation Controls**: Optionally, you can select one of the following from the **Default Process Priority** drop-down menu:
   - **Critical** (highest) (Not recommended)
   - **Above Normal** (Not recommended)
   - **Normal** (Default)
   - **Below Normal**
   - **Idle** (lowest)

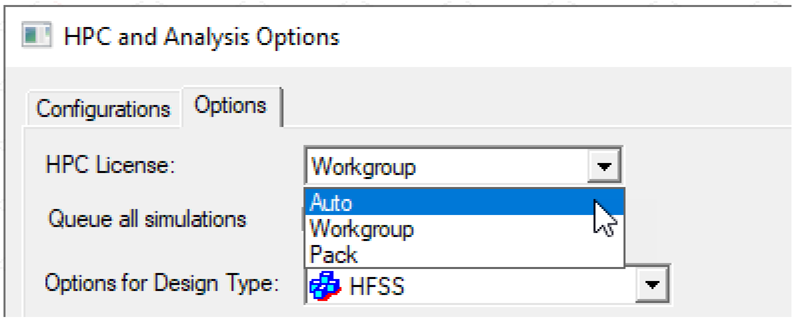
You can also set these values using Python scripts.

For information on editing configurations, see .

Licensing Settings Tool

HPC licensing includes a choice called “Auto”, along with Workgroup and Pack. The “Auto” choice is available in the HPC License combobox found on the **Options** tab of the **HPC and**

Analysis Options dialog box. If you make changes in the License Settings Tool, we recommend that you restart Electronics Desktop.

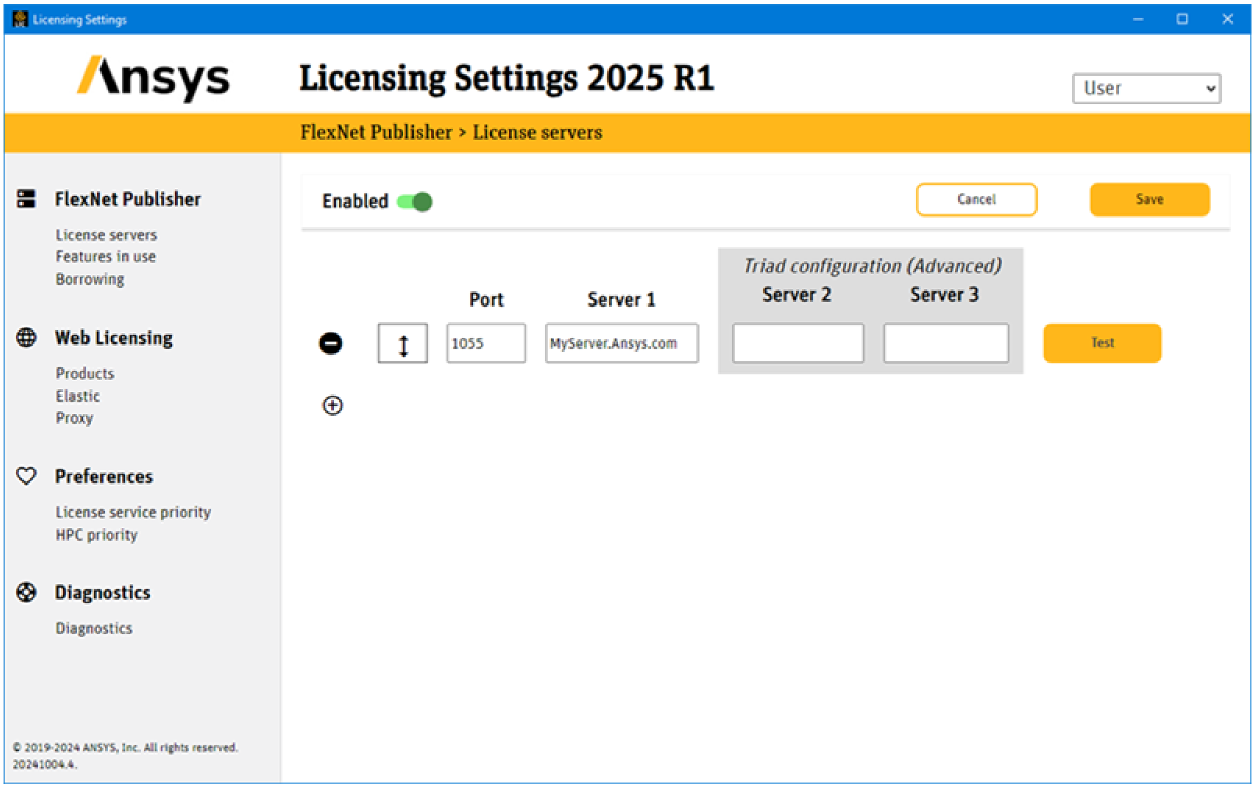


"Auto" means delegate the choice of Workgroup or Pack licensing to the Ansys Licensing Settings tool which is a separate application installed along with Electronics Desktop, accessible from the **Tools** menu. The HPC settings specified by the Licensing Settings tool are shared by other Ansys products. This provides you a single place where you can set the HPC preferences.

> **Note:** Important: HPC licensing preferences in the Licensing Settings tool are used only when Auto is selected in Electronics Desktop.

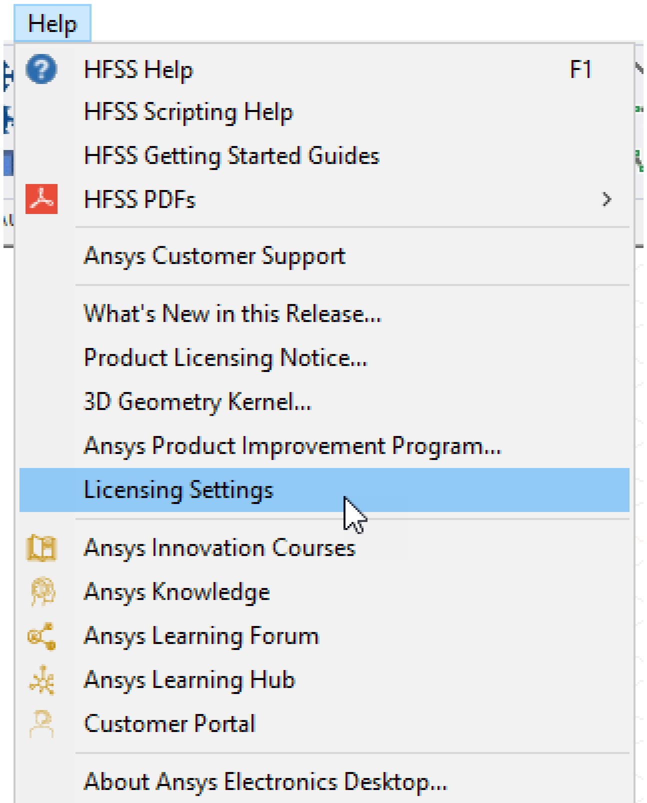
Launching the Licensing Settings Tool from the Main Menu

Click **Tools** > **Licensing Settings** to launch the Licensing Settings tool.



Licensing Settings Help

You can obtain help with the Licensing Settings tool from the **Licensing Settings** menu item in the **Help** menu.



License Fallbacks

When using Auto HPC licensing, in some cases if there aren't enough available licenses of the highest priority HPC license type, the lower priority HPC license type will be used. For instance, assume the user has set the priority order to "Ansys HPC, Ansys HPC Pack" in the Licensing Settings tool. If the user requests a nominal solve using 14 cores (10 Ansys HPC), but 10 Ansys HPC licenses aren't available, 2 Ansys HPC Pack licenses will be used if available.

Setting Options via Configuration Files

In addition to setting options from the Electronics Desktop user interface (UI), you can also set options in several configuration files. Option settings in configuration files may apply to all users or only to a specific user, and they may apply to all hosts or only to specific hosts. There are four levels of settings, listed below from most specific (highest precedence) to most general (lowest precedence):

- Host-dependent user options – apply to the specified user on the specified host only
- Host-independent user options – apply to the specified user on all hosts
- Host-dependent default options – apply to all users on the specified host
- Installation default – default for all users on all hosts

In the list above, settings at any level override settings at lower levels . If there is no setting in any file, the application default value is used. See UpdateRegistry for instructions on selecting these levels.

> **Important:**
>
> Options set from the Desktop UI override and update user settings in configuration files. Otherwise, the existing configuration file settings are used.

Behavior Examples

Consider running an application as user **jsmith** on host **host123**. If there is no host-dependent user setting for the *Expand Project Tree on Insert* option in the host-dependent user options config file for user **jsmith** on host **host123**, but there is a setting for the "Expand Project Tree on Insert" option in the host-independent user options config file for user **jsmith**, then the latter setting will be used if it is not overridden using the Desktop UI. Any settings in the host-dependent default options config file or the installation default config file are ignored.

As another example, consider running an application as user **jdoe** on host **host123**. If there is no setting for the *Expand Project Tree on Insert* option in the host-dependent user options config file for **jdoe** on **host123** or in the host-independent user options config file for user **jdoe** or in the host-dependent default options config file for host **host123**, then the value from the installation default config file are used, if present.

Rules for Modifying Option Settings

Option settings displayed in the Desktop UI follow the above rules. That is, if there is a setting in any of the option config files, then the setting from the highest priority config file is displayed in the Desktop UI. If there is no setting in any of the option config files, then the global default value is used. You can modify settings using the Options dialog boxes in the Desktop UI. If the dialog box is closed with the **Cancel** button, changes made on any of the tabs are discarded. If the dialog box is closed with the **OK** button, any settings that have been changed by the user are written to the host-dependent user options config file. The changed values written to this file are then used the next time that the application is run by the same user on the same host. The Desktop UI option settings are not written to any of the other option config files.

Configuration File Locations

The configuration files for host-dependent and installation **default** options reside at: "<*installation-directory*>\ANSYS Inc\v252\AnsysEM\config". The configuration files for host-dependent and host-independent **user** options reside in a subfolder of the user's Documents folder (for Windows) or the user's HOME folder (for Linux). See the tables below for specific Windows and Linux file names and paths.

Products with Multiple Desktop Versions

For products that have multiple Desktop versions, each will have a separate user-specific config directory, with a different value for the <*ApplicationName&Version*> directory name.

Table of Directories and Files

The following table shows the directories and files, where the *Level Name* is the name used to describe an options config file when using the UpdateRegistry tool.

Config File	**Level Name**	**File Name**	**Windows Directory Path**	**Linux Directory Path**
host-dependent user options	user_machine	<*hostname*>_user.XML	%UserProfile%\Documents\Ansoft\<*ApplicationName&Version*>\config	$HOME/Ansoft/<*ApplicationName&Version*>/config
host-independent user options	user	user.XML		
host-dependent default options	install_machine	<*hostname*>.XML	*"<Installation_Directory*>\ANSYS Inc*"<Version*>\AnsysEM\config"	*"<Installation_Directory*>/ansys_inc/*"<Version*>/AnsysEM/config"
installation default	install	default.XML		

Note:

- ***<hostname>*** is the name of the computer on which the Electronics Desktop software is installed
- **$HOME** is the user's home directory on Linux
- ***<ApplicationName&Version>*** is the product name (without spaces) followed by the four-digit year of the version, a decimal point, and the minor release number (such as ElectronicsDesktop2025.2)
- **%UserProfile%** is a Windows variable that represents the currently active user's profile (for example, C:\Users\JohnDoe)
- ***<Installation_Directory>*** is the root folder where the Ansys software is installed (typically, C:\Program Files, on Windows, or /opt, on Linux)
- ***<Version>*** is the lower-case letter "v" followed by last two digits of the product version's year and the minor release number, without a decimal point (such as v252)

The following table shows an example of specific file names and directory names for a typical Ansys Electronics Desktop installation on Microsoft Windows and on Linux. These are the files that apply to the current software version, user "**jsmith**," and hostname "**host123**":

<table>
<tr><th>Config File</th><th>Level Name</th><th>File Name</th><th>Windows Directory Path</th><th>Linux Directory Path</th></tr>
<tr><td>host dependent user options</td><td>user_machine</td><td>host123_user.XML</td><td rowspan="2">C:\Users\jsmith\Documents\Ansoft\ ElectronicsDesktop2025.2\config</td><td rowspan="2">/home/jsmith/Ansoft/ ElectronicsDesktop2025.2/config</td></tr>
<tr><td>host independent user options</td><td>user</td><td>user.XML</td></tr>
<tr><td>host dependent default options</td><td>install_machine</td><td>host123.XML</td><td rowspan="2">"C:\Program Files\ANSYS Inc\v252\ AnsysEM\config"</td><td rowspan="2">/opt/ansys_inc/v252 /AnsysEM/config</td></tr>
<tr><td>installation default</td><td>install</td><td>default.XML</td></tr>
</table>

> **Note:**
>
> As with the temporary file location configuration files, the settings in these options files have precedence in the following sequence:
>
> user_machine (highest precedence), user, install_machine, install (lowest precedence).
>
> The first time you start and exit the application, the file at the "user_machine" level is created (<*hostname*>_user.XML). The other files are only created if you do one of the following:
>
> - Use the Choose Registry Type command to switch to a machine-independent registry (user.XML), or
> - Use the UpdateRegistry tool to specify an option at the "user," "install_machine," or "install" level.
>
> If the temporary directory is set to an empty string in a configuration file, then that setting is ignored.

Setting or Removing Option Values in Configuration Files: UpdateRegistry Command

UpdateRegistry is a command line tool used to modify option settings in the options config files. You can use this command to add, change or remove settings from any of the option config files. This tool is included in the installation directory of each product. This feature makes it easier for different users to use Ansys Electromagnetics tools installed on shared directories or network drives.

The UpdateRegistry command has multiple command line formats, as shown below.

The following command line options are *mutually exclusive*:

- `-Set`
- `-Get`
- `-GetKeys`
- `-Delete`
- `-FromFile`

UpdateRegistry -Set Command

This command is used to add or modify an option setting in an option config file. If the option config file does not exist, it will be created. If the setting does not exist in the specified config file, it will be added. If the setting already exists in the specified config file, then the value will be changed to the specified value.

Example:

```
UpdateRegistry -Set -ProductName <name> -
RegistryKey <keyPath> -RegistryValue <value> [
-RegistryLevel <level> ]
```

Required:

`<name>`	The application or product name and version. For example, ElectronicsDesktop2025.2. If the name contains spaces, it must be quoted. The name can be found in the ProductList.txt file in the install directory: "...\ANSYS Inc\v252\AnsysEM\config\"
`<keyPath>`	The pathname of the option setting. This includes the same analysis-related registry keys and values that are displayed by the -batchoptions help. For example, Desktop/Settings/ProjectOptions/AnimationMemory
`<value>`	The new value of the option, typically a string or a number. If the value contains spaces, it must be quoted.

Optional:

`<level>`	When specifying `-RegistryLevel`, this is a string denoting which config file to modify. One of: install, install_machine, user, and user_machine. If the level is not specified, the user_machine (host-dependent user options) file is modified.

UpdateRegistry -Get Command

This command is used to view an option value in an option config file. If the setting exists in the specified config file or files, then the value, the value type and the config file where the value was found will be reported. If no value is found, then that will also be reported.

Example:

```
UpdateRegistry -Get -ProductName <name> -
RegistryKey <keyPath> [ -RegistryLevel <level>
]
```

Required:

`<name>`	The application or product name and version. For example, ElectronicsDesktop2025.2. If the name contains spaces, it must be quoted. The name can be found in the ProductList.txt file in the install directory: "...\ANSYS Inc\v252\AnsysEM\config\"
`<keyPath>`	The pathname of the option setting. This includes the same analysis-related registry keys and values that are displayed by the -batchoptions help. For example, Desktop/Settings/ProjectOptions/AnimationMemory

Optional:

`<level>`	When specifying `-RegistryLevel`, this is a string denoting which config file to modify. One of: install, install_machine, user, and user_machine. If the level is not specified, then all config files are searched in order of precedence.

UpdateRegistry -GetKeys Command

This command is used to view the allowed key names for all of the option settings, or to view a subset of the key names that match a string. For each key displayed, the current value, if any, is also reported. If a key has a value in multiple config files, then only the highest precedence value is reported.

Example:

```
UpdateRegistry -GetKeys [ <pattern> ] -
ProductName <name> [ -Case ]
```

Required:

`<name>`	The application or product name and version. For example, ElectronicsDesktop2025.2. If the name contains spaces, it must be quoted. The name can be found in the ProductList.txt file in the install directory: "...\ANSYS Inc\v252\AnsysEM\config\"

Optional:

`<pattern>`	If no pattern is specified, then all allowed key names are reported. If a pattern is specified, then only keys that match the pattern are shown. For example, Settings/Project. If the name contains spaces, it must be quoted. By default, the pattern match is case insensitive.

`-Case`	If this command line option is specified, then the pattern match is case sensitive.

UpdateRegistry -Delete Command

This command is used to remove an option setting from an option config file. If the setting does not exist in the specified config file, the file will not be changed. If the setting exists in the specified config file, then it will be removed. A setting may need to be removed from an option config file, to allow the setting from a lower priority file to be used by the application.

Example:

```
UpdateRegistry -Delete -ProductName <name> -
RegistryKey <keyPath> [ -RegistryLevel <level>
]
```

Required:

`<name>`	The application or product name and version. For example, ElectronicsDesktop2025.2. If the name contains spaces, it must be quoted. The name can be found in the ProductList.txt file in the install directory: "...\ANSYS Inc\v252\AnsysEM\config\"
`<keyPath>`	The pathname of the option setting. This includes the same analysis-related registry keys and values that are displayed by the -batchoptions help. For example, Desktop/Settings/ProjectOptions/AnimationMemory

Optional:

`<level>`	When specifying `-RegistryLevel`, this is a string denoting which config file to modify. One of: install, install_machine, user, and user_machine. If the level is not specified, the user_machine (host-dependent user options) file is modified.

UpdateRegistry -FromFile Command

You can use this form of the UpdateRegistry command to set multiple key-value pairs from a file with a single UpdateRegistry command. You specify the -FromFile command line option. This option must be followed by a filename. The file may contain multiple entries, where each entry contains a registry key and a registry value. The key-value pairs are added to the registry level specified by the -RegistryLevel command line option; if no -RegistryLevel is specified, then the default registry level (user_machine) is used.

UpdateRegistry File Format

Note:

Functionality featured in the example(s) in this section applies to multiple design types.

The file format is similar to the -batchoptions file format. An example UpdateRegistry file is shown below:

```
$begin 'AddEntries'
  'TempDirectory'='C:/temp/AnsysEM'
  'Desktop/Settings/ProjectOptions/HPCLicenseType'='Pool'
$end 'AddEntries'
```

Additional notes on the file format:

- The file may contain an arbitrary number of entries, one per line.
- Leading whitespace on each line is ignored. Spaces or tabs may be used to make the file more readable.

Registry key pathname:

- The registry key pathname appears before the equal sign "=" on each line.
- Each registry key pathname must be enclosed in single quotes.
- This includes the same analysis-related registry keys and values that are displayed by the -batchoptions help.

Registry value:

- The registry value appears after the equal sign on each line.
- Integral registry values must not be enclosed in quotes.
- All other registry values are treated as strings, and must be enclosed in single quotes.
- The forward slash "/" may be used as a directory separator on Windows and Linux. The back slash "\" may be used as a directory separator on Windows only.
- The back slash "\" is used as an escape character in the value string. That it, this character removes the special meaning of the following character.
- The single quote character normally ends the value string. The back slash may be used to remove this special meaning, and include a single quote in the string.
- To use a back slash as a directory separator on Windows, it must be escaped. That is, a double back slash "\\" is used to denote a single directory separator.

Alternative UpdateRegistry File Format:

- Analysis Configuration File format, which is exported from the HPC and Analysis Options dialog box.

> **Note:**
>
> If a current registry does not exist,Ansys Electronics Desktop can detect earlier minor versions of same application on the same machine. If such a registry exists (and does not involve -help, -batchoptionhelp, IsBatchMode(), -regserver, -unregserver, running a script, or non graphical mode), a prompt displays allowing you to port the registry from an earlier version.

Example Uses for Export Options Features

The **Tools** > **Options** > **Export Options** feature is intended to make it easier for different users to use Ansys Electromagnetics tools installed on shared directories or network drives. This section outlines some use cases enabled by this feature.

> **Note:**
>
> Functionality featured in the examples in this topic apply to multiple design types.

Options that Apply to All Users

In many cases, an Ansys Electromagnetics tool installation is administered and maintained by a single user or group and used by a number of other users or groups. The permissions of the Ansys Electromagnetics tool installation may be set so that the administrator may add, delete or modify files, but other users may only read or execute these files. The administrator may set the recommended option settings in the installation default config file and/or the host dependent default options config file. These config files reside within the installation directory hierarchy, and should generally have the same permissions as other Ansys Electromagnetics tool installation files. This allows that administrator to control these settings, but does not allow other users to add, remove, or change these settings.

Each user can override any of these settings, if needed. This may be done using the Desktop UI, which affects the host-dependent user options config file. It may also be done using the host-independent user options config file. If user has overridden an option setting in either of the user files, the user may revert back to the option settings provided by the administrator by removing the setting of the same option in the host-dependent user option config file and/or the host-independent user option config file.

For global defaults, the administrator may set a value in the installation default config file. These settings will to apply to all users on all hosts.

In some cases, there are significant differences between the capabilities of different hosts. The host-dependent default config file may be used to specify different default values on some hosts. Any setting in a host dependent default config file would affect all users running on the specified host. The installation default value is used if there is no value specified for the setting in the host-dependent default config file for the current host. Note that the host-dependent default config file is named <*hostname*>.xml, where *hostname* is the name of the host computer.

Example: Searching for a Registry Key Pathname

Both administrators and ordinary users may occasionally use the UpdateRegistry command line tool to add, change or delete settings. To use this tool, the registry key pathname must be known by the user. The -GetKeys option may be used to quickly search for a key pathname if some information is known about it. For example, if the administrator knows that there is a setting related to issuing warning messages when available disk space is low, but does not know the exact key name, the following command may list some of the keys related to disk space:

```
UpdateRegistry -GetKeys disk -ProductName ElectronicsDesktop2025.2
```

This will display a list of all keys that match the string "disk" case insensitively.

Typical output may look like the following:

```
Registry keys matching pattern <disk> case insensitively:
Desktop/Settings/ProjectOptions/DiskLimitForAbort: value is <0> at
level <user_machine>
```

Example: Setting an Installation Default Value

The normal default for the Options/General/Desktop Performance/Warn when available disk space is less than setting is 0 MB. If the administrator is concerned that running out of disk space might be a common problem, the administrator could set the installation default for the warn setting setting to 1000 MB, for example. This limit would then apply to all users running on all hosts. The administrator could use the following command to change this setting for Ansys Electronics Desktop:

```
UpdateRegistry -Set -ProductName ANSYSElectronicsDesktop2025.2
 -RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort
 -RegistryValue 1000
 -RegistryLevel install
```

Example: Setting a Host-Dependent Default Value

For this example, assume that all hosts have two cores, except for three hosts: bighost1, bighost2, and bighost3, which have eight cores each. The administrator has set the Desktop/Settings/ProjectOptions/NumberOfProcessors option value to 2 in the installation default config file. The administrator may set the Desktop/Settings/ProjectOptions/NumberOfProcessors option value to 8 in the host-dependent default config files for the three hosts having 8 cores: bighost1, bighost2 and bighost3. The administrator may log in to host bighost1 and run the following command to change this setting for the host-dependent default options config file for host bighost1:

```
UpdateRegistry -set -ProductName ElectronicsDesktop2025.2
 -RegistryKey Desktop/Settings/ProjectOptions/NumberOfProcessors
 -RegistryValue 8
 -RegistryLevel install_machine
```

To make this change for the other two hosts, the administrator would log in to bighost2 and bighost3, and run the same command on each of those hosts.

Example: Reverting from a User-Defined Option Value to the Administrator Default

Consider the case in which Electronics Desktop was installed and the administrator initially did not set a value for the Desktop/Settings/ProjectOptions/DiskLimitForAbort setting in the default installation config file. User jsmith (who always uses host jshost) wanted to be warned before disk space dropped to zero, so he set the Desktop/Settings/ProjectOptions/DiskLimitForAbort to 100 MB using the UI. This setting is recorded in the host dependent user options config file for host jshost and user jsmith. Now the administrator learns that many users are running into disk space issues, so that administrator sets the installation default value for the setting Desktop/Settings/ProjectOptions/DiskLimitForAbort to 1000 MB, as in the above example.

When user jsmith runs Electronics Desktop on host jshost, the disk limit is 100 MB, not 1000 MB, because the host-dependent user options config file overrides all of the other config files. User jsmith may revert to the administrator provided default by removing this setting from the host dependent user options config file for host jshost and user jsmith. The following command may be run by user jsmith on host jshost to remove this setting:

```
UpdateRegistry -Delete -ProductName ElectronicsDesktop2025.2
 -RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort
 -RegistryLevel user_machine
```

If user jsmith had added a value for this setting to the host-independent user options config file, then user jsmith would also run the following command to remove this setting from the host-independent user options config file:

```
UpdateRegistry -Delete -ProductName ElectronicsDesktop2025.2
 -RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort
 -RegistryLevel user
```

User Options and the UpdateRegistry Tool

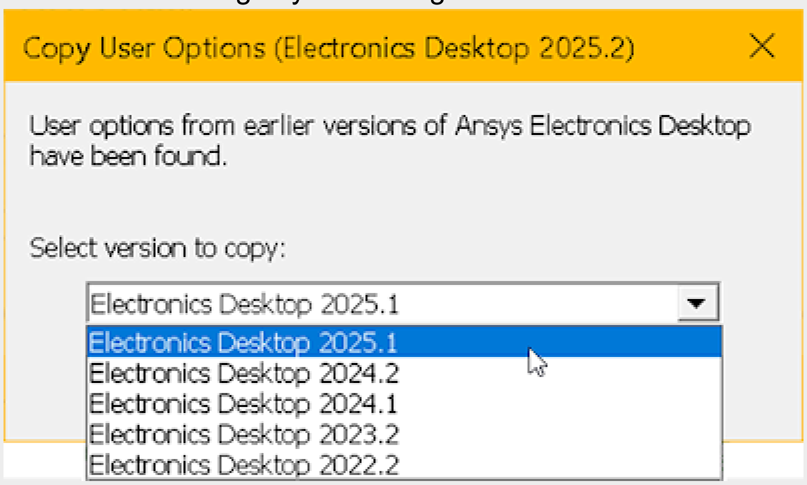
> **Note:**
>
> Functionality featured in the example(s) in this section applies to multiple design types.

When you change an option's value using the Ansys Electronics Desktop UI, the new value is stored in the currently active user options registry. You can use the UpdateRegistry tool to add, modify, or remove settings. However, you cannot use the Desktop UI to *remove* settings from the registry. *You must use the UpdateRegistry tool to do that.*

If you have not already created a user options configuration file (either host-specific or host-independent), before launching the Ansys Electronics Desktop application for the first time, all settings will come from the *host-specific default* options configuration file or the *installation default options* configuration file. In a host-specific user options configuration file, any settings applicable to a different host from the current one will not be carried over to the new host. This behavior may be inconvenient if the user has preferred option settings that differ from the default settings, which apply to all users, especially if the user runs the application on a number of different hosts. In this case, the user may choose to use a host-independent registry and set the option values in this type of options file. See Understanding Registry Tools for more information. Host-independent option values can be used on all new hosts, overriding any values set by the administrator to apply to all users. Changes made in the UI can affect the user's host-specific or host-independent registry, depending on the currently specified registry type.

Note:

If a current registry does not exist, the program detects earlier major and minor versions of same application on the same host. If such a registry exists (and does not involve -help, -batchoptionhelp, IsBatchMode(), -regserver, -unregserver, running a script, or non graphical mode), a prompt displays the earlier versions you can select, from which the registry will be migrated.



Copy over registry entries (both Windows and Ansys .xml files).

Example: Removing a Host-Dependent User Option Setting

For this example, user jsmith always uses host jshost to run Ansys Electronics Desktop. At some point, jsmith set the Autosave interval in **General Options** > **Project Options** tab to 1000 edits, and this value was written to the jsmith's host-dependent user options config file for host jshost. Now, jsmith wants to remove this setting and return to the default value of 10. User jsmith may run the following command on host jshost to remove the Desktop/Settings/ProjectOptions/AutoSaveInterval option value from this config file:

```
UpdateRegistry -Delete -ProductName ElectronicsDesktop2025.2
-RegistryKey Desktop/Settings/ProjectOptions/AutoSaveInterval
-RegistryLevel user_machine
```

Example: Getting a Value from a Specific Configuration File

In the previous example, the user jsmith may decide to check the Desktop/Settings/ProjectOptions/DiskLimitForAbort setting in the host-independent user configuration file before making any changes to this setting. The following command may be used to quickly view this setting before making the change:

```
UpdateRegistry -Get -ProductName ElectronicsDesktop2025.2 -
RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort -
RegistryLevel user
```

Example: Getting a Value Using Precedence Rules

In may cases, the user is more interested in the value of a setting that will be applicable when running the product than in the setting in a single configuration file. If the -Get option is used with no -RegistryLevel specified, then the value reported is the value found in the highest precedence configuration file. If the user jsmith is interested in the highest precedence value for the Desktop/Settings/ProjectOptions/DiskLimitForAbort setting, then the following command may be used to report this information:

```
UpdateRegistry -Get -ProductName ElectronicsDesktop2025.2 -
RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort
```

Example: Adding a Host-Independent User Option Setting

Consider the case in which there is no value set for the Desktop/Settings/ProjectOptions/DiskLimitForAbort setting for all users for Ansys Electronics Desktop. The default is then 0 MB. User jsmith uses a variety of hosts and wants to be warned whenever disk space drops to 250 MB on any host. User jsmith may use the following command to set the Desktop/Settings/ProjectOptions/DiskLimitForAbort option value to 250 MB for all hosts:

```
UpdateRegistry -set -ProductName ElectronicsDesktop2025.2
-RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort
-RegistryValue 250 -RegistryLevel user
```

Example: Setting a RegistryKey not defined in ElectronicsDesktopRegistrySyntax.xml

```
"C:\Program Files\ANSYS Inc\v252\AnsysEM">UpdateRegistry -Set -
ProductName ElectronicsDesktop2025.2 -RegistryKey "3D
Editors/Preferences/Geometry3D/q" -RegistryValue "m"
```

```
registry key is not defined in this product
```

```
Register value <3D Editors/Preferences/Geometry3D/q> is created with
value <m>
```

Example: Change a Newly Set Registry Key

```
"C:\Program Files\ANSYS Inc\v252\AnsysEM\">UpdateRegistry -Set -
ProductName ElectronicsDesktop2025.2 -RegistryKey "3D
```

```
Editors/Preferences/Geometry3D/q" -RegistryValue "n"
Register value <3D Editors/Preferences/Geometry3D/q> is set to <n>
```

If you attempt to set a registry key to the wrong data type, a message like the following appears.

```
Cannot set the registry value to different data type
The work around is delete it, then create it again
error on to set registry value at <3D
Editors/Preferences/Geometry3D/q>
```

Setting the Temporary Directory

The temporary directory may be viewed or set using the Electronics Desktop user interface (UI), or from the command line.

To set the directory via the UI:

1. Navigate to **Tools** > **Options** > **General Options**.
2. In the tree at the left side of the dialog box, expand the **General** branch and select **Directories**.
3. Use the **Temp** field and **Override** check box to enter a desired directory path.
   Values set in this manner are written to the user_machine level configuration file for the temporary directory. If the **Override** check box is cleared, clicking **OK** changes the user_machine level setting for the temporary directory to an empty string. This enables settings from the next highest precedence configuration file. The file that provides the currently active temporary directory setting is shown under the **Temp** edit box.

To set the temporary directory from the command line, using the -batchoptions command line option. See: Running Ansys Electronics Desktop from a Command Line and -Batchoptions Command Line Examples.

The temporary directory may be configured with an installation default value, as well as a host-dependent default value, a host-independent user-specified value and a host-dependent user-specified value. Temporary directory settings are stored in different files from the other option settings. These files are located in the same directories as the configuration files for the other option settings. The following table shows the directories and files used to store temporary directory settings.

Config File	Level Name	File Name	Windows Directory Path	Linux Directory Path
Host-dependent, user-specific	user_machine	<*hostname*>.cfg	%UserProfile%\Documents\Ansoft\<*ApplicationName&Version*	$HOME/Ansoft/<*ApplicationName&Version*>/config

Config File	Level Name	File Name	Windows Directory Path	Linux Directory Path
temporary directory				
Host-independent, user-specific temporary directory	user	default.cfg	>\config	
Host-dependent, default temporary directory	install_machine	<*hostname*>.cfg	"<*InstallationDirectory*>\ANSYS Inc\v252\AnsysEM\config"	<*InstallationDirectory*>/ansys_inc/v252/AnsysEM/config
Installation default temporary directory	install	default.cfg		

> **Note:**
>
> - ***<hostname>*** is the name of the computer on which the Electronics Desktop software is installed
> - **$HOME** is the user's home directory on Linux
> - ***<ApplicationName&Version>*** is the product name (without spaces) followed by the four-digit year of the version, a decimal point, and the minor release number (such as ElectronicsDesktop2025.2)
> - **%UserProfile%** is a Windows variable that represents the currently active user's profile (for example, C:\Users\JohnDoe)
> - ***<InstallationDirectory>*** is the root folder where the Ansys software is installed (typically, C:\Program Files, on Windows, or /opt, on Linux)
> - ***<Version>*** is the lower-case letter "v" followed by last two digits of the product version's year and the minor release number, without a decimal point (such as v252)
>
> As with other options, the settings in these files have precedence in the following sequence: user_machine (highest precedence), user, install_machine, install (lowest precedence). The installer creates the file at the "install" level (default.cfg). The first time you start and then exit the application, the file at the "user_machine" level is created (<*hostname*>.cfg). The other files are only created if you use the UpdateRegistry tool to specify an option at the "user" or "install_machine" level. If the temporary directory is set to an empty string in a configuration file, then that setting is ignored.

Temporary Directory Configuration File Format

This section describes the format of temporary directory configuration files. The format is the same for files at all four levels: user_machine, user, install_machine, and install. These files are text files, so any text editor may be used to modify or create them.

An example temporary directory configuration file is shown below:

```
$begin 'Config'
tempdirectory='C:/TEMP/AnsysEM'
$end 'Config'
```

The temporary directory specified by this configuration file is C:/TEMP/AnsysEM.

Important:

- The string containing the pathname of the temporary directory must be enclosed in single quotes.
- The forward slash (/) may be used as a directory separator on Windows and Linux. The backslash (\) may be used as a directory separator on Windows only.
- The backslash (\) is used as an escape character in the tempdirectory string. That it, this character removes the special meaning of the following character.
- The single quote character normally ends the tempdirectory string. The backslash may be used to remove this special meaning, and include a single quote in the string.
- To use a backslash as a directory separator on Windows, it must be escaped. That is, a double backslash "\\" is used to denote a single directory separator.
- On Windows, a UNC path normally begins with two backslash characters. In a tempdirectory string, each of these backslash characters must be doubled, so four consecutive backslashes "\\\\" are used in the config file.

Example: Config File with UNC

```
$begin 'Config'
tempdirectory='\\\\hostxyz\\TEMP\\abc'
$end 'Config'
```

Here, hostxyz is a host with a sharename TEMP having subdirectory abc used as the temporarydirectory. This shows that four backslashes are required for UNC names and that backslashes used as directory separators must be doubled.

Example: Config File with Single Quote

```
$begin 'Config'
tempdirectory='C:/TEMP/ab\'cd'
$end 'Config'
```

Temporary directory is C:/TEMP/ab'cd. This shows how to include a single quote in a tempdirectory pathname. It also shows that forward slashes may be used as directory separators on Windows.

UpdateRegistry: Setting or Removing Temporary Directory Values in Configuration Files

The UpdateRegistry command line tool, described above, may be used to view, add, change or remove the temporary directory setting from any of the temporary directory config files. The registry key for viewing or modifying the temporary directory is TempDirectory.

The -Get, -Set, and -Delete options are valid for viewing, adding, changing, or deleting a temporary directory setting.

The -GetKeys option does not list the temporary directory key.

Batchoptions Command Line Examples

The `-batchoptions` entries command line argument may be used to specify one or more batchoptions settings from the command line. To specify multiple entries using a single `-batchoptions` argument, the entries should be enclosed in double quotes. Alternatively, the batchoptions may be specified in a file using the `-batchoptions <filename>` command line argument format. In this case, the filename is an absolute or relative pathname of the file containing the batchoptions, as described above.

> **Important:**
>
> The two approaches may not be combined: either all batchoptions must be in a file or all batchoptions must be specified explicitly on the command line.

Batchoptions File Format

> **Note:**
>
> Functionality featured in the example in this section applies to multiple design types.

An example batchoptions file is shown below:

```
$begin 'Config'
  'Desktop/ProjectDirectory'='C:/test/projects'
  'Desktop/Settings/ProjectOptions/NumberofProcessors'=2
$end 'Config'
```

Additional notes on the file format:

- The file may contain an arbitrary number of batchoption entries, one per line.
- Leading whitespace on each line is ignored. Spaces or tabs may be used to make the file more readable.
- The **Registry Key** appears before the equals sign (=) on each line and must be enclosed in single quotes ('). The registry key includes the key path and the name of the registry value.
- The registry value (or option value) appears after the equals sign on each line.
- Registry keys are case-insensitive.

- There are two supported types of registry values—string and integer:
  - Each **string** value must be enclosed in single quotes (').
  - Do **not** enclose **interger** values in quotes.
- For file paths within string values, the forward slash (/) may be used as a directory separator on both Windows and Linux systems.
- Alternatively, on Windows only, the customary backslash (\ ) may be used as a directory separator. However, in string values, the backslash is used as an escape code for indicating special characters that cannot be typed directly (such as \n for a new line). Therfore, if you use the backslash as a directory path separator, each instance must be doubled (\\ ). An example is: '\\\\host3\\temp\\Ansoft'. In this case, each double backslash is interpreted as a single backslash (\\host3\temp\Ansoft).
- The single quote character (') normally ends a string value. If you need to include a single quote (or apostrophe) within a string, use the backslash-apostrophe (**\'**) escape sequence. For example, the string '%UserProfile%/Documents/Ansoft/John\'s_Files', is interpreted as: %UserProfile%/Documents/Ansoft/John's_Files.

Example -BatchOptions with -Remote (Windows)

> **Note:**
>
> Functionality featured in the examples in this section applies to multiple design types.

In this example, we run a batch HFSS analysis of project file project1.aedt. We want all temporary files and directories to be created in C:\temp\HFSS instead of using the default temporary directory. We decide that the analysis will be done on a remote host, at IP address 12.34.56.78. Because of limited memory on the remote host, we decide to run the analysis using only a single COM engine. Because the remote host has four cores, we decide to use four four processors for of the analysis. We can use the **-Remote** option to specify that there will be a single remote COM engine.

Here is a sample command line for this analysis, where the project file \\somehost\projects\project1.aedt is located in a shared directory specified using a UNC path:

```
ansysedt -BatchSolve -Remote -Machinelist list=12.34.56.78
 -batchoptions "'TempDirectory'='C:/temp/HFSS'
'Desktop/Settings/ProjectOptions/NumberofProcessors'=4"
\\somehost\projects\project1.aedt
```

An alternative is to use the **-Distributed** command line option. Because the -Machinelist list contains only one host, there is a single remote COM engine in this case also.

```
ansysedt -BatchSolve -Distributed -Machinelist list=12.34.56.78
 -batchoptions "'TempDirectory'='C:\\temp\\HFSS'
 'Desktop/Settings/ProjectOptions/NumberofProcessors'=4"
 \\somehost\projects\project1.aedt
```

The second line of the first example shows that you can use the forward slash "/" as a Windows directory separator within option value 'strings'. In this case, it is used in the TempDirectory path. You can also use the customary backslash "\" as a Windows directory separator, but it must be doubled to "\\" because the backslash is also an escape character within parameter strings. This usage is demonstrated in the second line of the second example, again in the TempDirectory path.

Example -Batchsolve with -Machinelist (Windows)

> **Note:**
>
> Functionality featured in the example(s) in this section applies to multiple design types.

Suppose that we want to run a batch HFSS analysis of project file project1.aedt. Because all of our hosts have multiples of 2 cores, we specify that we will use two threads for multiprocessing for both the distributed (HFSS/NumCoresPerDistributedTask) and non-distributed (Desktop/Settings/ProjectOptions/NumberOfProcessors) parts of the job. The analysis contains a sweep that will be distributed across three hosts: ahmed, bill, and catherine. The hosts ahmed and bill have four cores each, so we run two distributed COM engines on each of these hosts, each using two threads. Host catherine has only two cores, so we specify only one distributed COM engine on this host. This COM engine will also use two threads.

Here is a sample command line for this analysis, where the project file \\dennis\projects\project1.aedt is located in a shared directory specified using a UNC path:

```
ansysedt -BatchSolve -Distributed
-Machinelist list=ahmed,ahmed,bill,bill,catherine
-batchoptions "’Hfss/
‘Desktop/Settings/ProjectOptions/NumberOfProcessors’
‘Hfss/NumCoresPerDistributedTask’=2"
\\dennis\projects\project1.aedt
```

Example -Batchsolve with -Machinelist (Linux)

> **Note:**
>
> Functionality featured in the example(s) in this section applies to multiple design types.

In this example, we run a batch HFSS analysis of project file project2.aedt. We have four identical host computers—host1, host2, host3, and host4 for analysis, and each host has 4 cores. We do not use multiprocessing for the distributed analysis, so NumCoresPerDistributedTask=1. As each host has four cores, we specify multiprocessing using 4 threads for the non-distributed part of the analysis, so NumberOfProcessors=4. Because we do not use multiprocessing for the distributed analysis, we will run four distributed COM engines on each host, with a singe core available for each engine.

Here is a sample command line for this analysis, where the project file /home/jsmith/projects/project2.aedt is located in a shared directory:

```
hfss -BatchSolve -Distributed
-Machinelist file=/home/jsmith/hosts/list2
-batchoptions "HFSS/
'Desktop/Setings/ProjectOptions/NumberOfProcessors'=4
'HFSS/NumCoresPerDistributedTask'=1"
/home/jsmith/projects/project2.aedt
```

For this example, the hostnames are in the text file /home/jsmith/hosts/list2. The file contents are as follows:

```
host1
host1
host1
host1
host2
host2
host2
host2
host3
host3
host3
host3
host4
host4
host4
host4
```

Example -Batchsolve for Local (Windows)

> **Note:**
>
> Functionality featured in the example(s) in this section applies to multiple design types.

In this example, we run a batch analysis of project file testproject.adsn on the local host. We want all temporary files and directories created in directory C:\temp\ansysedt instead of using the installation default temporary directory. Because the local host has four cores, we decide to use four threads for multiprocessing, for both distributed and non-distributed parts of the analysis.

Here is a sample command line for this analysis, where the project file \\host123\projects\testproject.adsn is located in a shared directory specified using a UNC path:

```
ansysedt -BatchSolve -Local -batchoptions
"TempDirectory='C:/temp/ansysedt'
'HFSS/SolverOptions/NumProcessors'=4
'HFSS/SolverOptions/NumProcessorsDistrib'=4"
\\host123\projects\testproject.adsn
```

Note that the batchoptions pathnames 'HFSS/SolverOptions/NumProcessors' and 'HFSS/SolverOptions/NumProcessorsDistrib' must be in single quotes because they both contain embedded spaces.

Batch Options and Analysis Configurations in the Registry

Analysis configurations are used to specify machines and options for local, remote, and distributed analysis, including capabilities that are enabled by HPC licenses.

How Analysis Configurations are Stored in the Registry

A configuration contains information beyond the machine or machines to use for a solution. Examples are the number of processors to allocate to the analysis for each machine in the list, memory limits, directory locations for personal libraries and temporary files, and many other preferences.

Note:

- Options are arranged as keys and values (in a structure similar to the *Windows Registry*). However, these options are not a part of the *Windows Registry* but are separately stored and maintained by the Ansys Electronics Desktop software. For more information concerning the configuration files comprisisng the options registry, see the following topics:

  Setting Options Via Configuration Files
  Setting the Temporary Directory

- For access to options and functionality beyond what is directly accessible via the user interface or batch options, refer to the documentation of the **UpdateRegistry** tool. This tool is discussed in the following help topic and in the topics that follow it in the same branch of the product help:

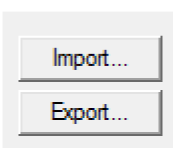
  Setting or Removing Option Values in Configuration Files: UpdateRegistry Command

General settings are associated with the Desktop application or 3D Editor. "HPCLicenseType" and "tempdirectory" values are at the root level of the registry. Other options are specific to a particular product or design type. For example, certain mesh, boundary, and memory limit settings as well as many other preferences are product-specific and are therefore associated only with the applicable design types. Such options appear with a consistent value name but in a different registry path for each applicable design type.

Copying a Configuration from one Design Type or Product to Another

To copy a configuration from one design type (or product) to another:

1. Click **Tools** > **Options** > **HPC and Analysis Options**. The *HPC and Analysis Options* dialog box appears.
2. On the **Configurations** tab of the **HPC and Analysis Options** dialog box, use the **Export…** button to export the configuration to a file.



3. Switch to the destination design type (or product) and use the **Import…** button to import the configuration data.

   Any data that is not applicable to the destination design type is ignored; any settings present in the destination design type that were not present in the source configuration will be assigned default values. The user may then edit the copy, as desired.

Using HPC and Analysis for Configurations

Due to the complexity of the registry values for the configurations we do not recommend directly editing these values using the *UpdateRegistry* tool. Instead, use the **HPC and Analysis Options** dialog box to edit or create a configuration. (See: Setting HPC and Analysis Options.) Configurations created or edited using the GUI are stored in the **user_machine** level of the registry. You can create a configuration for one of the other registry levels using the following steps:

1. Create the configuration using the GUI, then export the configuration to a file.
2. Delete the configuration using the GUI so that it will not be present in the **user_machine** level. Then, exit the GUI.
3. Use the **UpdateRegistry** tool to import the data into the desired registry level using the **-FromFile** option to specify the file exported via the GUI, and using the **-RegistryLevel** option to specify the registry level where the configuration is to be stored. For example, an administrator may use this approach to create a configuration at the **install** level that may be used by any user on any machine.

Batch Options

There is a large number of both general and product-specific options supported by the software. These options have evolved over time. Therefore, older batchoptions files may no longer be valid, and the options listed in the user interface of the current software may differ from earlier versions you have used. Additionally, there are options beyond those listed within the user interface (that is, the UI provides a subset of the available options). You can generate a more complete list of options by running the UpdateRegistry tool with the -GetKeys switch and piping the output to a text file, as detailed in the following Windows procedure:

1. In a command window, navigate to the following folder:

   "<*installation_directory*>\ANSYS Inc\v252\AnsysEM"

2. Type and enter the following command:

   UpdateRegistry -GetKeys -ProductName ElectronicsDesktop20*xx.y* **>** <*text_file_path*>**\Batchoptions.txt**

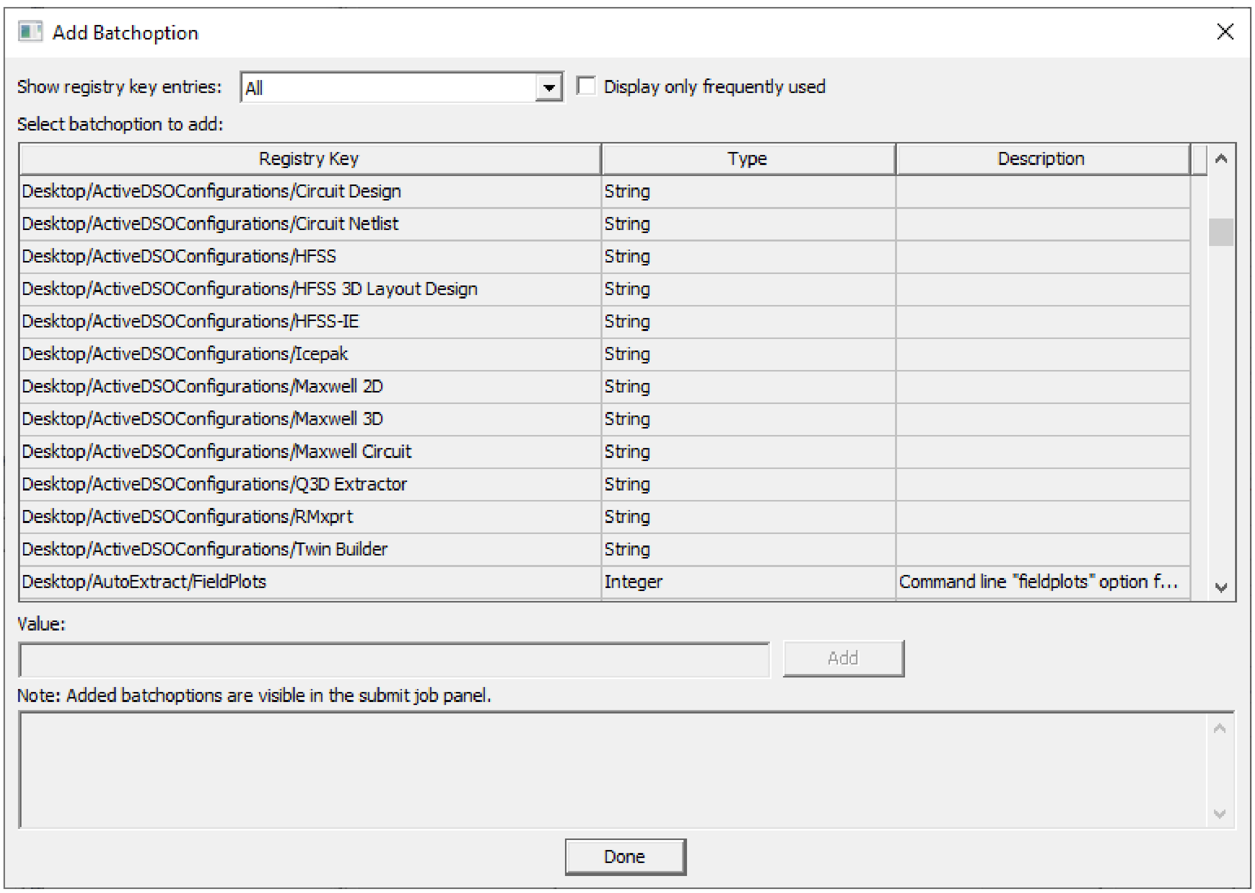
   Substitute the last two digits of the installed product version year and the minor release number for *xx.y* (such as **25.1**). Also, substitute the desired *text_file_path* (such as **%UserProfile%\Desktop**).

   The resulting **Batchoptions.txt** file will contain a nearly complete list of available options.

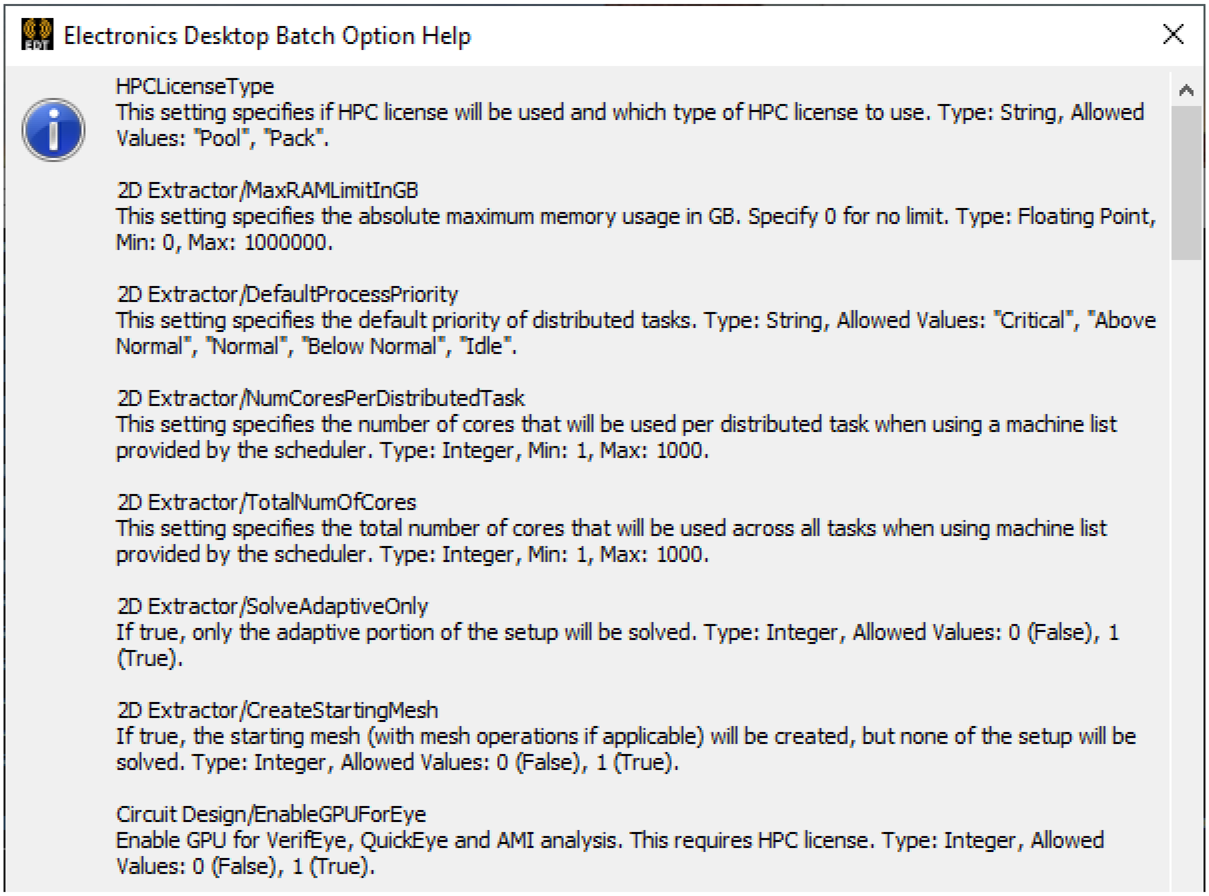
This procedure also works from the Linux <*installation_directory*>, which has the same downstream path (for example, under "opt/") as a Windows installation (typically under "C:\Program Files\").

> **Note:**
>
> For special product-specific options that are neither available from the GUI nor listed by the UpdateRegistry tool, see Special Batch Options. Even though the -GetKeys switch does not list these special options, you can still use the UpdateRegistry tool to set them.

When you submit jobs to a remote computer or cluster, you can specify batch options using the job submission GUI. When using the GUI, you select the batch options from a list and therefore avoid typographical errors. For the most commonly used batch options, there is detailed information about the allowed values. Click **Submit** on the **Simulation** ribbon tab to access the *Submit Job To* dialog box. Then, click **Add** to access the **Add Batchoption** dialog box pictured below:



To assist users who need to specify batch options and are unable to use the job submission GUI, a new help option has been added. If the Electronics Desktop application is launched with the **-batchoptionhelp** command line argument, a message box is displayed which lists and describes the most common design type-specific batch options:



Managing Projects and Designs

An Ansys Electronics Desktop *project* is essentially a folder that includes one or more models, or *designs*.

Some basic tasks that can be performed on projects and designs include:

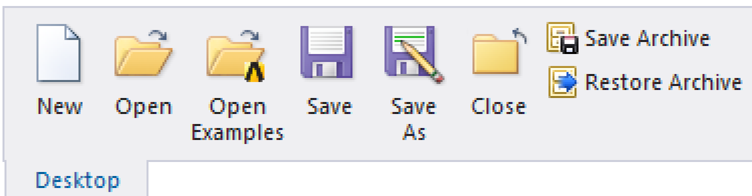
- Opening Projects
- Creating Projects
- Saving Projects
- Importing and Exporting Projects and Data
- Setting Project Options
- Validating Projects
- Copying and Pasting Projects and Designs
- Renaming Projects
- Deleting Projects or Designs
- Setting Read Only Designs

- Updating Design Components
- Undoing and Redoing Commands
- Inserting a Documentation File
- Saving Project Notes
- Printing

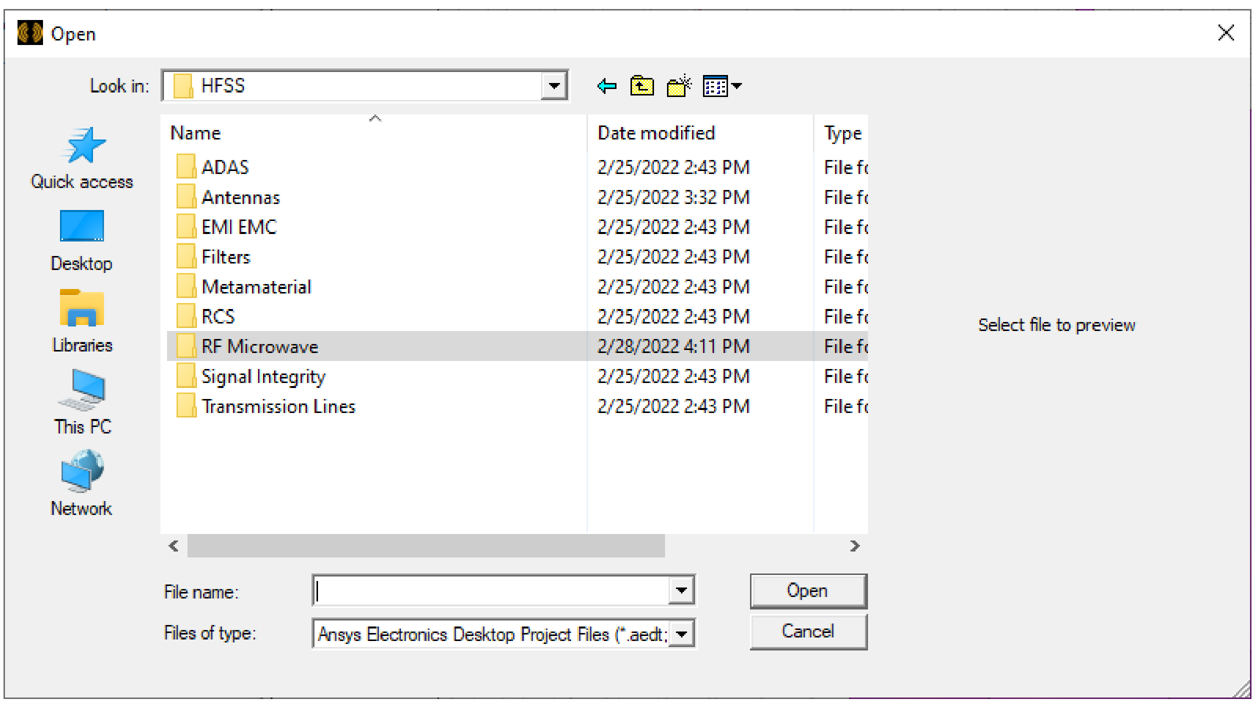
Opening Projects

Open a previously saved project using the **File** > **Open** command.

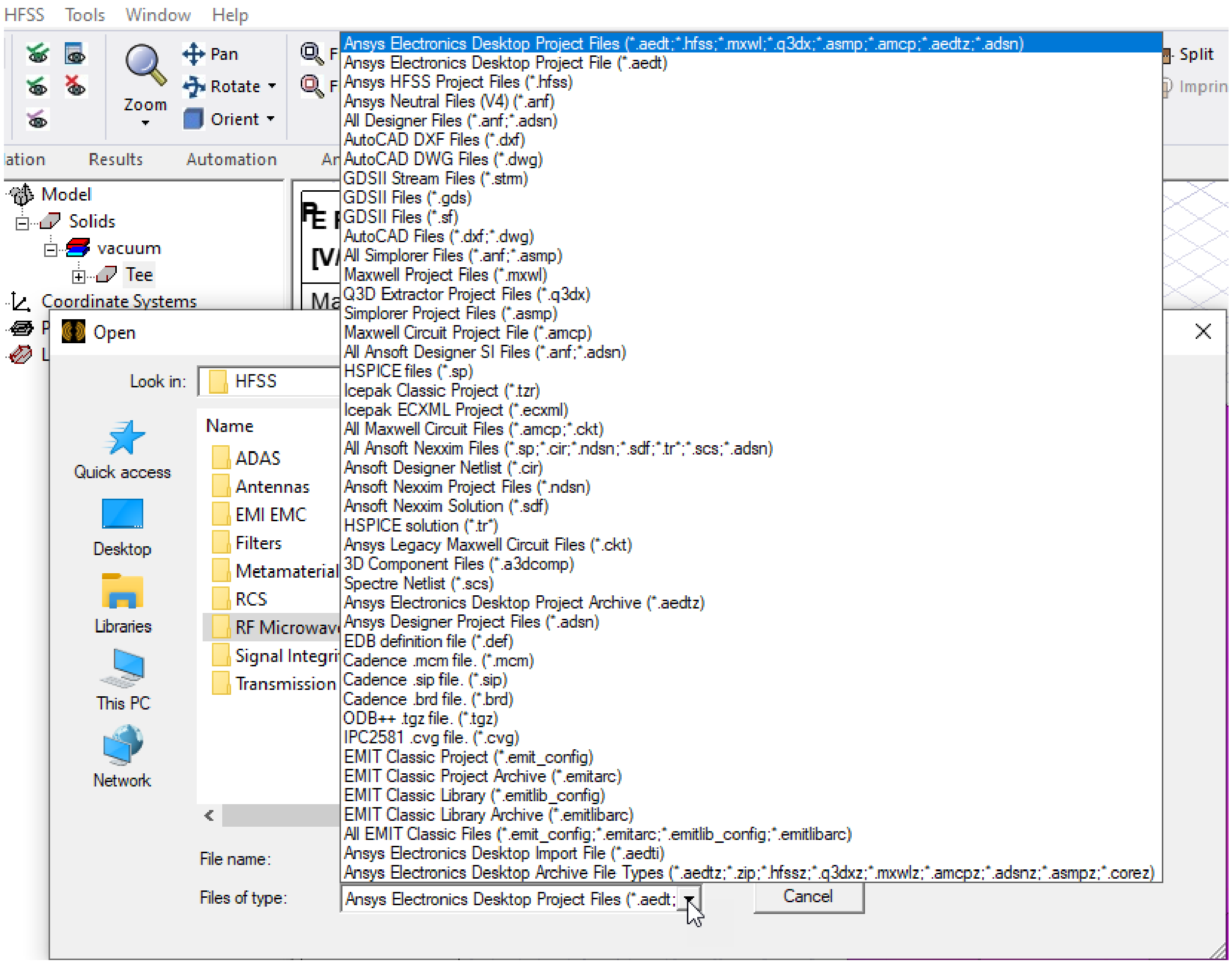
1. Click **File** > **Open** or click the **Open** icon on the **Desktop** tab of the ribbon.



2. Use the file browser to find the project file.

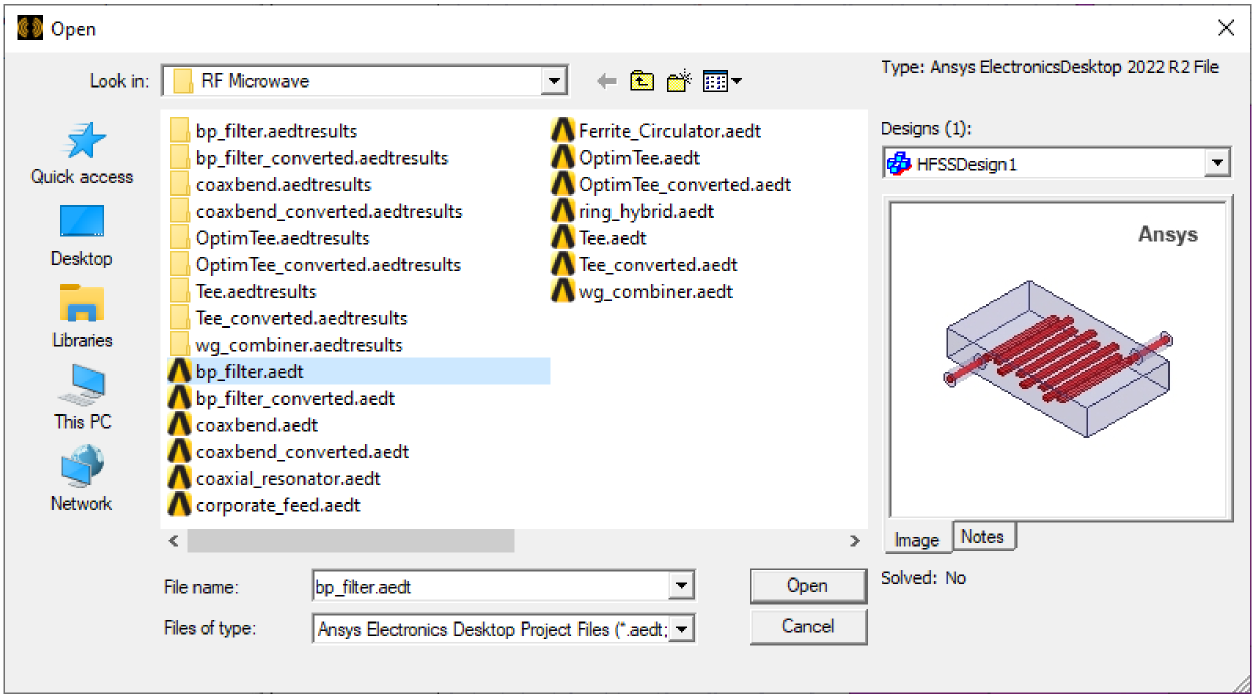


By default, only files that can be opened or translated display. *.aedt is the main default, although, you can also view several types of legacy files. The dropdown for Files of type lists the kinds file and archive formats for which you can select filters.



3. Most of these will be familiar to those seeking them. The Ansys Electronics Desktop Import file (*.aedti) format is for importing exported projects from ANSYS Discovery, an ANSYS Workbench tool. This format allows import of side by side *.aedti and the corresponding geometry definition *.sat into the Ansys Electronics Desktop, and creates a ready-to-solve HFSS project.

4. Select the file you want to open.



5. Click **OK**.

   The project information appears in the Project tree.

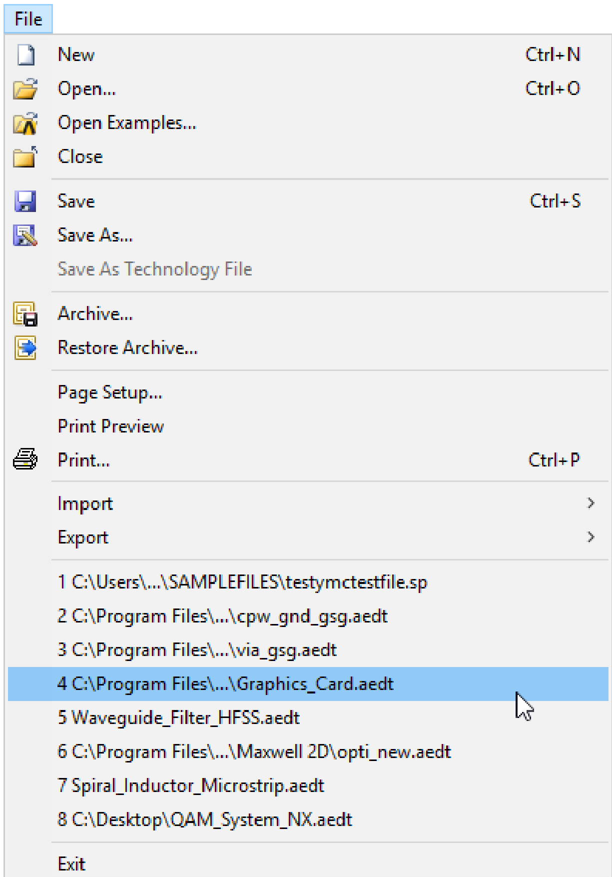
If you open another project without editing the automatically created project, the automatically created project is removed.

You can also open a saved project by:

- Dragging a project file icon to the Open icon
- Dragging a project file icon to the desktop
- Double-clicking a project file icon

Opening Recent Projects

Recently opened projects appear at the bottom of the **File** menu. To open one of these files, click its file name.

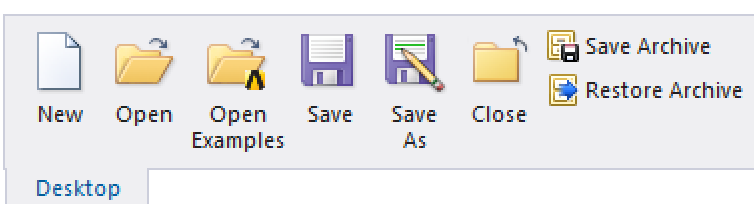


Opening Example Projects

You can directly access and open example projects included with product installation using **File** > **Open Examples**, or by clicking **Open Examples** on the **Desktop** tab of the ribbon.
See Example Projects for additional description of these files.

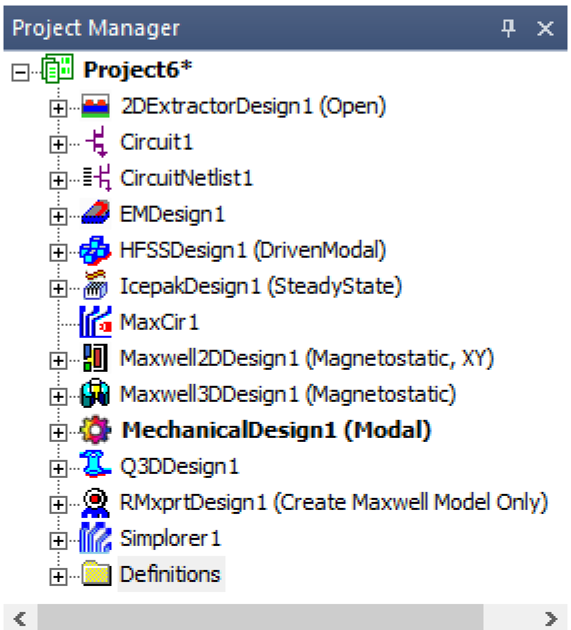
Creating Projects

To create a project, click **File** > **New**, or click the **New** icon on the **Desktop** tab of the ribbon.



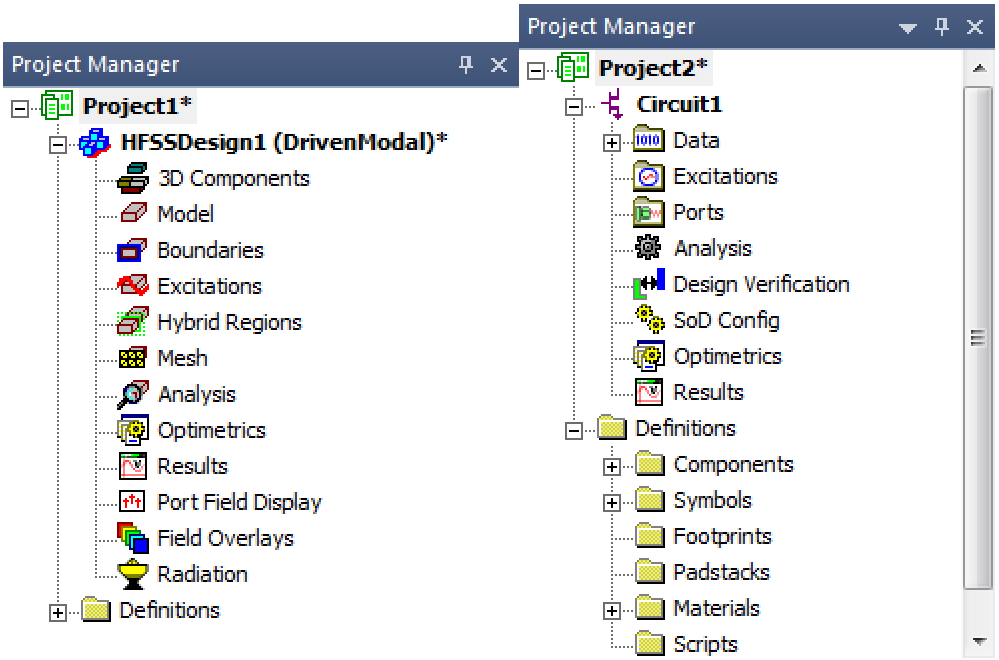
A new project is listed in the project tree. It is named Project*n* by default, where *n* is the order in which the project was added to the current project folder.

You can insert designs of any type into the project, where they are stored in the project tree.



The default name for each inserted project is <designType>*n*. You can also specify the name of the project when you save it using the **File** > **Save** or **File** > **Save As** commands.

You can view the contents of a project by clicking the plus sign (+) for each level of the hierarchy in the project or design.

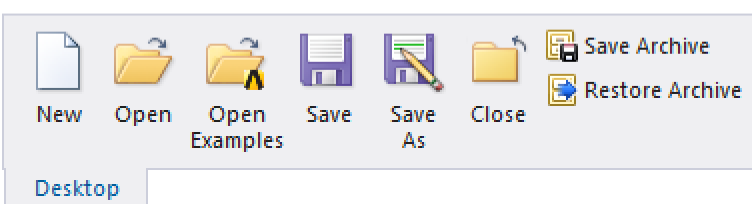


Saving Projects

Use the **File** > **Save As** command or select the **Desktop** tab of the Ribbon and click the **Save As** icon to:

- Save a new project.
- Save the active project with a different name or in a different location.
- Save the active project in another file format for use in another program.

Use the **File** > **Save** command or select the **Desktop** tab of the Ribbon and click the **Save** icon to save the active project.



> **Warning:**
>
> Be sure to save models periodically. Saving frequently helps prevent the loss of work if a problem occurs.
>
> Although Ansys Electronics Desktop has an auto-save feature, it may not save frequently enough for your needs.

Each solver has a Save Before Solving setting located in the **Tools** > **Options** window. This setting is enabled by default.

A prompt appears when you attempt to save a previously versioned file. If you agree to the prompt, the file is upgraded to the Ansys Electronics Desktop version in which you are running the software. In this case the file may no longer be compatible with previous versions. If you do not agree to the prompt, the file is not saved, so the file retains its previous compatibility.

If you have a simulation running, you see a warning that if you continue, Ansys Electronics Desktop will abort the simulation. If you OK the warning, Ansys Electronics Desktop aborts the simulation and saves the project.

Path Name Length Issues for Windows

For most Windows programs, the current directory pathname length is limited to 259 characters on startup. Thus, the pathname length of the directory containing installed programs should be limited to no more than 259 characters because double-clicking on an application in Windows Explorer will set the working directory to the directory containing the application when the application is started.

Win32 Long Paths Not Enabled

If win32 long paths are not enabled on the analysis host, then essentially all files are limited to a maximum absolute pathname length of 259 characters. Directories may be limited to a maximum absolute pathname length of about 246 characters.

Win32 Long Paths Are Enabled

If win32 long paths are enabled on the analysis host, then many files are not subject to the maximum absolute pathname length limit of 259 characters. Below is a partial list of files that are still subject to this limit even if win32 Long Paths are enabled:

- Project files
- Project Archive files
- Ansys EDB (Electronics Database) files, typically stored in the `ProjectName.aedb` folder

- The temporary directory and most temporary files

  Because some temporary files and directories have automatically generated names that are long, it is best to use a short pathname for the temporary directory.

  > **Important:**
  >
  > Although most temporary files are created within the temporary directory, there may be some temporary files within the project directory. As a result, the project file directory pathname should be well below the 259 character limit.

To enable Long Path support on Windows, both of the following requirements must be met:

- Long paths must be enabled on the machine

  Use either of two ways to do this:

  - **Via Registry Setting:** `HKLM\SYSTEM\CurrentControlSet\Control\FileSystem : LongPathsEnabled=1`
  - **Via Group Policy Tool:** The “Enable Win32 long paths” setting is in the folder: `Computer Configuration\Administrative Templates\System\Filesystem`

Changing Auto-Save Settings

Recent actions performed on the active project are stored in an auto-save file in case of a sudden workstation crash or other unexpected problem. The auto-save file is stored in the same directory as the project file and is named Project*n*.aedt.auto by default. Ansys Electronics Desktop automatically saves all data for the project to the auto-save file, except solution data.

By default, Ansys Electronics Desktop automatically saves project data after every 10 edits. An edit is any action performed that changes data in the project or the design, including actions associated with project management, model creation, and solution analysis.

After a problem occurs, you may be able to choose to re-open the original project file in an effort to recover the solution data, or to open the auto-save file. If the original file is not available, attempting to open the file provides a message that the auto-save is being used. If neither file is available, an error message displays.

To modify auto-save settings:

1. Click **Tools** > **Options** > **General Options**.

   The **Options** window appears.

2. Under **Desktop Configuration**, verify that **Do Autosave** is selected.

   This option is selected by default.

3. In the **Autosave interval** box, enter the number of *edits* that you want to occur between automatic saves. By default, this option is set to 10.

   > **Note:**
   >
   > Auto-save *always* increments forward; therefore, even when you undo a command, Ansys Electronics Desktop counts it as an edit.

4. Click **OK** to apply the specified auto-save settings.

   Once the specified number of edits is carried out, a "model-only" save will occur. This means that Ansys Electronics Desktop does not save solutions data or clear any undo/redo history. When Ansys Electronics Desktop auto-saves, an ".auto" extension is appended to the original project file name. For example, Project1.aedt will automatically be saved as Project1.aedt.auto.

> **Warning:**
>
> When you close or rename a project, Ansys Electronics Desktop deletes the auto-save file.

Save Before Solving Option

The **Save before solving** option forces a full save before running the solve and is enabled by default. For efficiency reasons, the project is saved only if it has been modified since its last save.

To change this setting, click **Tools** > **Options** to open the Options window. Select the **General** tab and use the **Save before solving** check box to toggle the option.

You can start a new solve while running another without having to abort the running solve. If you start a solve while another solve is running and the Save before solving option is set, Ansys Electronics Desktop asks if you want solve without saving first. This lets you perform multiple solves, and if you have not edited the project in between solves, crash recovery will work.

Recovering Project Data from an Auto-Save File

Following a sudden workstation crash or other unexpected problem, you can recover project data from its auto-save file.

> **Warning:** When you recover a project's auto-save file you *cannot* recover any solutions data; recovering an auto-save file means you will lose any solutions data that existed in the original project file.

To recover project data in an auto-save file, if Ansys Electronics Desktop containing a design has unexpectedly crashed:

1. Launch Ansys Electronics Desktop from your desktop.
2. Click **File** > **Open**.
3. Select the original Project*n*.aedt project file for which you want to recover its Project*n*.aedt.auto auto-save file.

   The **Crash Recovery** window appears, giving you the option to open the original project file or the auto-save file.
4. Select **Open project using autosave file** to recover project data in the auto-save file**, and then click OK.** Ansys Electronics Desktop replaces the original project file with the data in the auto-save file.

   Ansys Electronics Desktop immediately overwrites the original project file data with the auto-save file data, removing the results directory (solutions data) from the original project file as it overwrites to the auto-save file.

> **Warning:** If you choose to recover the auto-save file, you cannot recover the original project file that has been overwritten; recovering data in an auto-save file is *not* reversible.

Importing and Exporting Projects and Data

Ansys Electronics Desktop can import and export a variety of file types.

Importing Files

Object, material, and parameter names with non-ASCII characters are not allowed, and therefore, not allowed for data transfer. Such transfers fail and produce an error message.

For certain file types, the import dialog contains **Validation and Healing Options**, which are enabled by default.

These types are:

- ACIS SAB files (*.sab)
- ACIS SAT files (*.sat)
- Ansys 3D Modeler files (*.sm3)

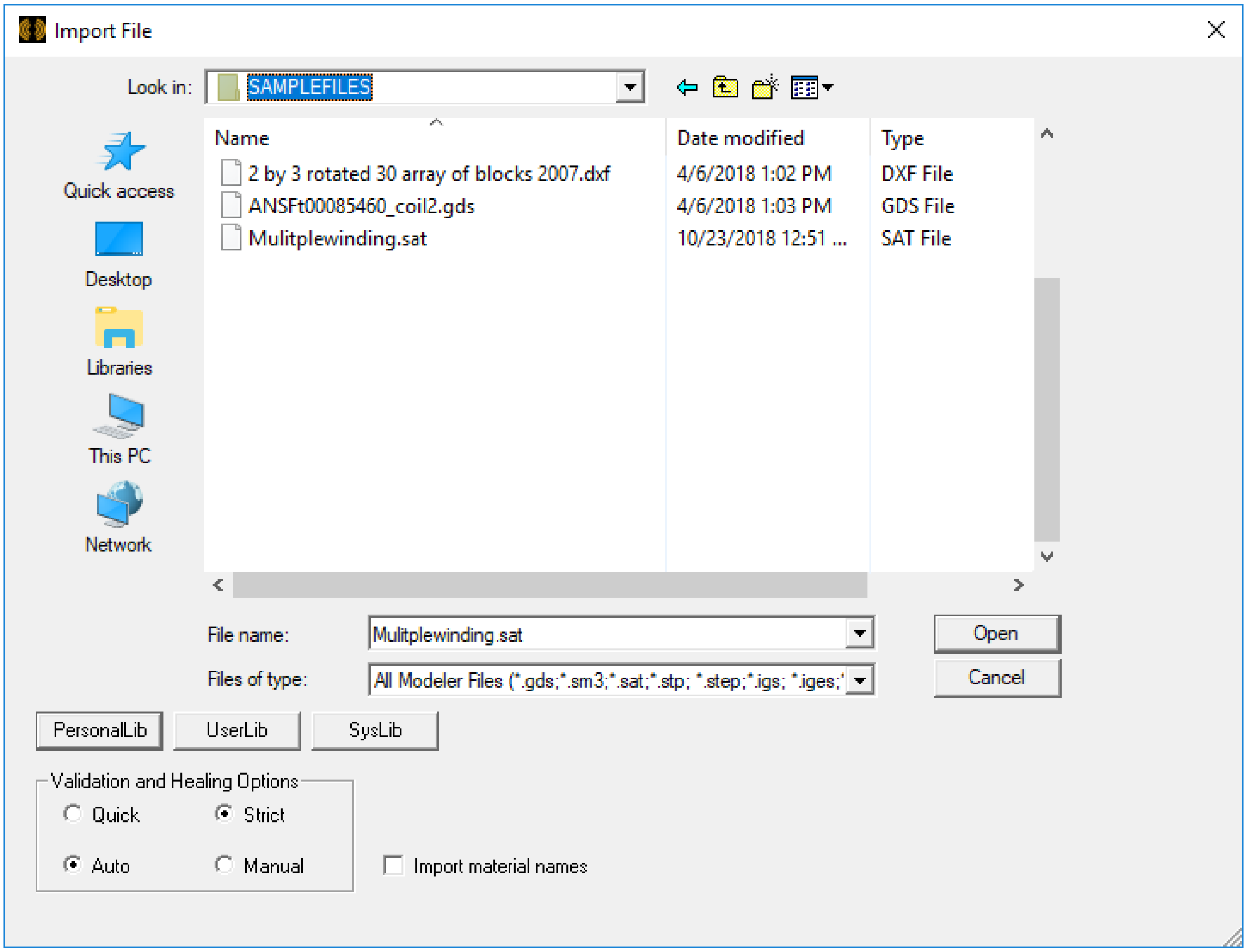
- Autodesk Inventor files (*.ipt, *.iam)
- CATIA V4/V5 files (*.exp, *.model, *.CATpart, *.CATproduct)
- Creo Parametric files (*.prt, *.asm)
- IGES files (*.iges, *.igs)
- JT files (*.jt)
- Nastran files (*,nas)
- NX files (*.prt)
- Parasolid files (*.x_t, and *.x_b)
- SOLIDWORKS Files (*.SLDPRT, *.SLDASM)
- STEP files (*.step, *.stp)

For more information, see Heal.

To import a file:

1. Click **Modeler > Import**.

   The **Import File** window appears.



If applicable to the file type, **Validation and Healing Options** are enabled by default. You can change them as desired.

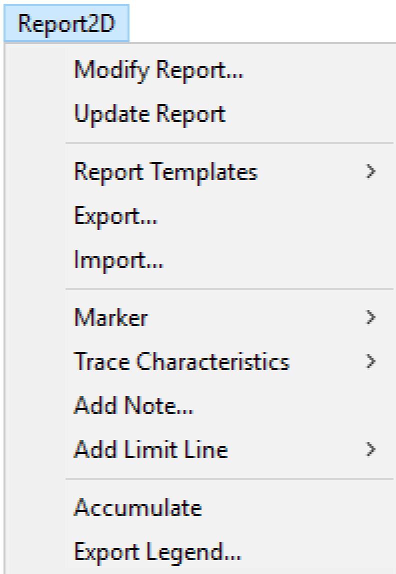
You can also use the check box to choose whether to **Import material names**.

2. Click **Open**.
3. Depending on the file type, additional options may appear. Set these as appropriate.
4. Click **OK**.

The file loads.

Importing Plot Data

When a report is open in Ansys Electronics Desktop, the **Report2D** menu appears.



> **Note:**
>
> You *must* have a report open in order to import plot data.

The **Report2D** > **Import Data** command lets you import plot data from comma delimited files (*.csv), tab delimited files (*.tab), Ansoft PlotData files (*.dat), post-processor files (*.txt), or Ansoft ReportData files (*.rdat).

1. From the top menu bar, click **Report2D** > **Import**.

   The **Open** window appears.

2. Browse for and select the desired file.
3. Click **Open** to import the file into the current report.

   The imported traces appear in the Project tree under the current report.

   > **Note:**
   >
   > For a report trace where the primary sweep is NOT the same as the x-component, an export report then import report may not produce the same curve. In this circumstance, only *.rdat import the same curve/trace. For other formats, the import produces two separated traces.

Exporting Files

You can export the following types of files from Ansys Electronics Desktop projects:

- 2D model files
- 3D model files
- Graphics files
- Reports as data or graphics files

Exporting Graphics Files

You can export to the following graphics formats:

Extension	Contents
.bmp	Bitmap files
.gif	Graphics Interchange Format files
.jpg	Joint Photographics Experts Group files
.png	Portable Network Graphics format files
.tiff	Tagged Image File Format files.
.wrl	Virtual Reality Modeling Language (VRML) files.
.gltf or **.glb**	Graphics Language Transmission Format (GLTF); only available on Windows

To export a Modeler image to a graphics format:

1. Click **Modeler** > **Export**.

   The **Export File** window appears.

2. Browse to and select a location to save your file.
3. In the **File Name** field, name the file.
4. From the **Save as Type** drop-down menu, select the desired image format.

   When you select an image format, **Image Export Options** appear. Note that this window does not appear for .gltf, .glb, or .wrl files, which are not image formats; if you are saving to one of these formats, skip to Step 6.

5. Choose your **Image Size**.

   Options include: Current View, User Defined (which allows you to specify **Width** and **Height**), Full Screen, 8K UHD, 4K UHD, 1080p HD, 720p HD, and 480p SD.

6. Click **Save**.

   The file is exported to the specified location with the appropriate file format.

You can also export an image file to a specified resolution using scripting commands. Fonts and line thickness are not scaled, only the image. You will have to iteratively increase font sizes until you find a suitable output.

Example Script:

```
oEditor.ExportModelImageToFile("C:/Users/Documents/highresexample_
image.jpg", 7680, 4320,

[

"NAME:SaveImageParams",

"ShowAxis:="       , "True",

"ShowGrid:="       , "True",

"ShowRuler:="      , "True",

"ShowRegion:="     , "Default",

"Selections:="     , ""

])
```

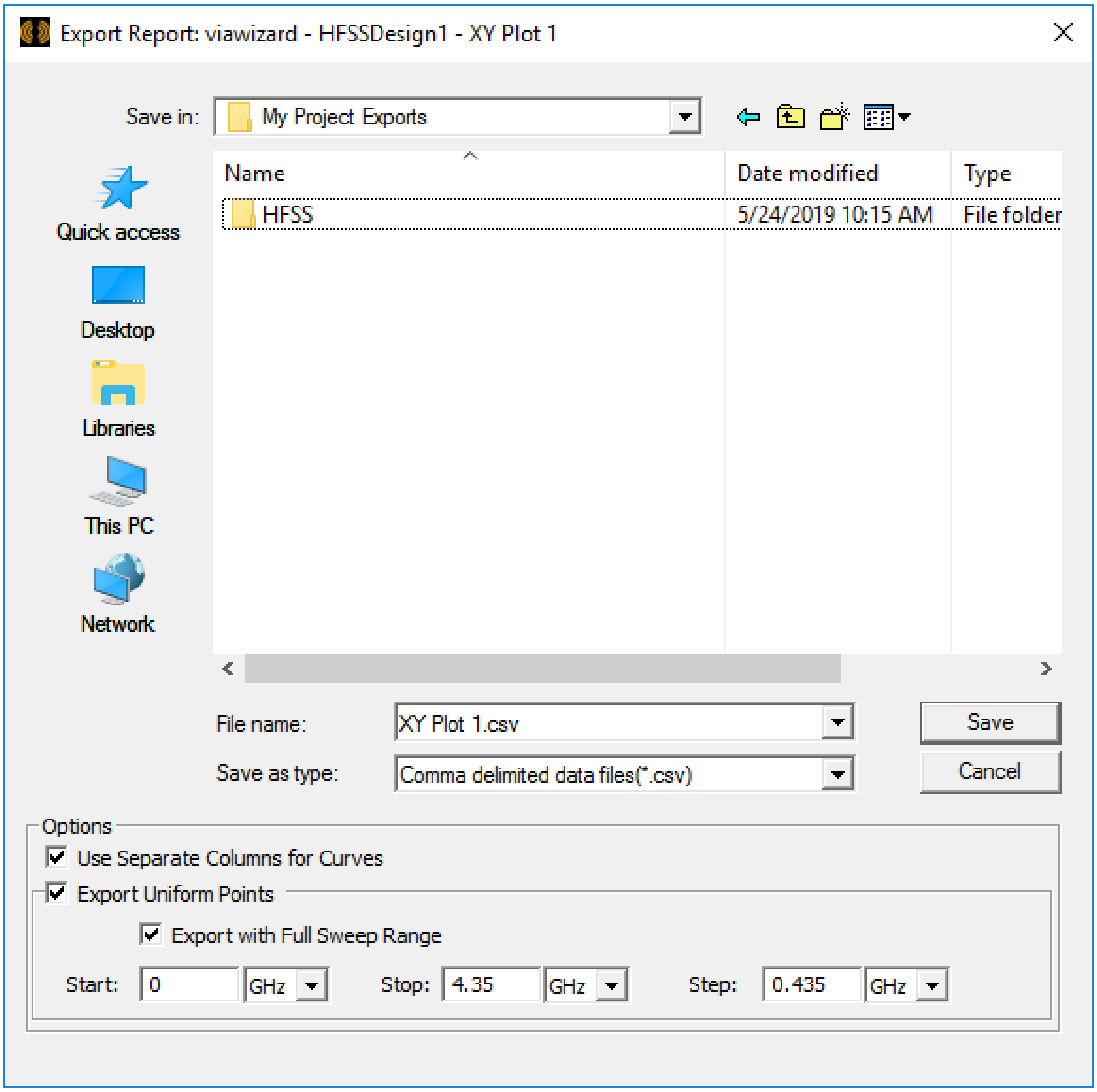
Exporting Reports

You can export reports in a variety of text and graphic formats. You must have an existing plot open to see the corresponding **Report2D** or **Report3D** menu.

1. Click **Report2D**> **Export** or **Report3D**> **Export**.

   Alternatively, right-click the plot and select **Export**.

   The *Export Report* dialog box appears.

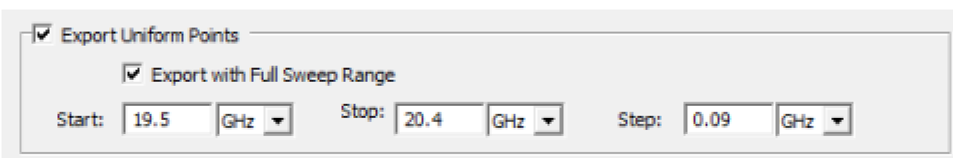


2. Use the file browser to find the directory where you want to save the file.
3. Type the name of the file in the **File name** field.
4. Select one of the following file formats from the **Save as type** drop-down menu:

Extension	Contents
.csv	Comma-delimited data file
.tab	Tab-separated file
.dat	Ansoft plot data file
.txt	Post processor format file

Extension	Contents
.rdat	Ansoft report data file
.emf	Microsoft EMF files
.gif	GIF files
.bmp	BMP files
.wrl	VRML files
.tif, **.tiff**	TIFF files
.jpg, **.jpeg**	JPEG files

- When exporting to *.csv or .tab files for an XY Plot, Eye Diagram or Data Table, a **Separate Columns for Curves** check box will be shown. Otherwise the check box will be hidden.
- When it is checked the data is exported with one column per curve, and variable values appear in the column's header cell.
- When it is unchecked the data is exported with one column per a trace, each variable has its own column.
- When the design has many variables, uncheck the check box may keep the column header from becoming too long and hard to read.
- When trace has more curves than number of sweep values, uncheck the check box to prevent the number of columns from being larger than the number of rows.
- When a report only has one trace or all trace have same variations, this option is unchecked by default. Otherwise, this feature is checked.
- For 2D reports, you can select **Export Uniform Points** and specify a full sweep range by editing start, stop, and step values.



5. Click **Save**. The report is exported to the specified location.

> **Note:**
>
> For a report trace where the primary sweep is NOT the same as the x-component, exporting and then importing a report may not produce the same curve. In this circumstance, only *.rdat format imports the same curve/trace. For other formats, the import produces two separated traces.

Exporting Options Files

Ansys Electronics Desktop can export user and host options files in XML format.

Exporting Options Files Using the Desktop UI

Export options files by selecting **Tools** > **Options** > **Export Options Files**. This brings up a browser dialog box to select the destination directory.

Click **Open** to copy all config files for the current user and current host to the specified directory. Config files for the install, install_machine, user, and user_machine levels will be copied, if they exist. One additional file, admin.XML, will also be copied to the destination directory; this file does not contain user configurable options.

Exporting Options Files Using a Script

A script command has been added that exports options config files:

ExportOptionsFiles

Syntax	`ExportOptionsFiles <DestinationDirectory>`
Return Value	None.
Parameters	BSTR <DestinationDirectory>
Python Example	`oDesktop.ExportOptionsFiles ('D:/test/export')`

Copying and Pasting a Project or Design

To **copy** a project or design:

1. In the Project Tree, select a project or design to enable the menu command **Edit** > **Copy**.
2. Click **Edit** > **Copy**.

   The project or design is copied for pasting.

To **paste** a project or design:

1. In the Project Tree, select a project or design to enable the menu command **Edit** > **Paste**.
2. Click **Edit** > **Paste**.

   The project or design is pasted under the selected project, and an icon is added to the project tree.

> **Note:**
>
> You can also use keyboard shortcuts.

Renaming Projects

In general, use the **File** menu commands to manage projects. If you move or change the names of files without using these commands, the software may not be able to find information necessary to solve the model.

To rename an existing, active project:

1. Select the project in the Project Tree.
2. Right-click and select **Rename**.

   This activates the text field for the project name.

3. Type the new project name and press **Enter**.

   The new project name appears in the directory and the project remains in the original location.

Archiving Projects

Use the **File** > **Archive** command to place a project and any other files related to the project that you want to include in a *.<product>z file or *.zip format archive. You can make notes about the contents of the archive and specify whether to include results and solutions files. The **Archive** command attempts to automatically detect the necessary files for linked projects and automatically include them in the archive. You can also add additional files to the archive, including results files, external files and projects. For example, if a project linked to the main project also has linked or associated files, you can add them.

Archive File Types

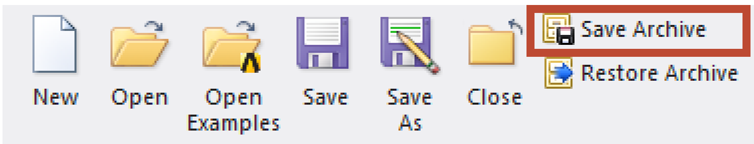
Internally, project archive files are .zip files, and are compatible with any program that can read .zip files. Project archive files have an extension that is unique for each product, and is generated by adding a 'z' to the project file extension (for example, *.aedtz, *.adsnz). This extension displays as the default when saving and restoring archive files.

Archive Preview

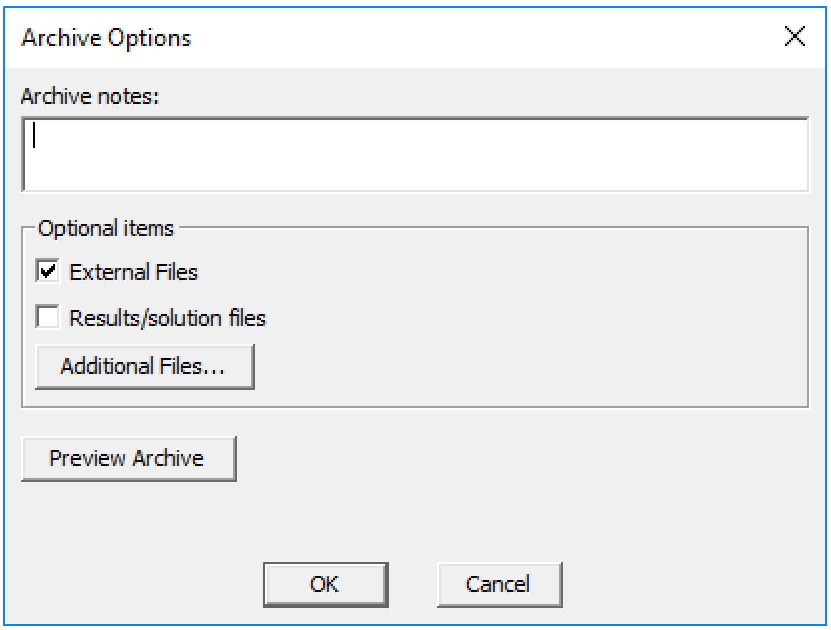
The **Archive** command includes a preview feature that lets you review the contents of a planned archive.

To archive the current project:

1. Click **File** > **Archive** or select the **Desktop** tab of the ribbon and click **Save Archive**.

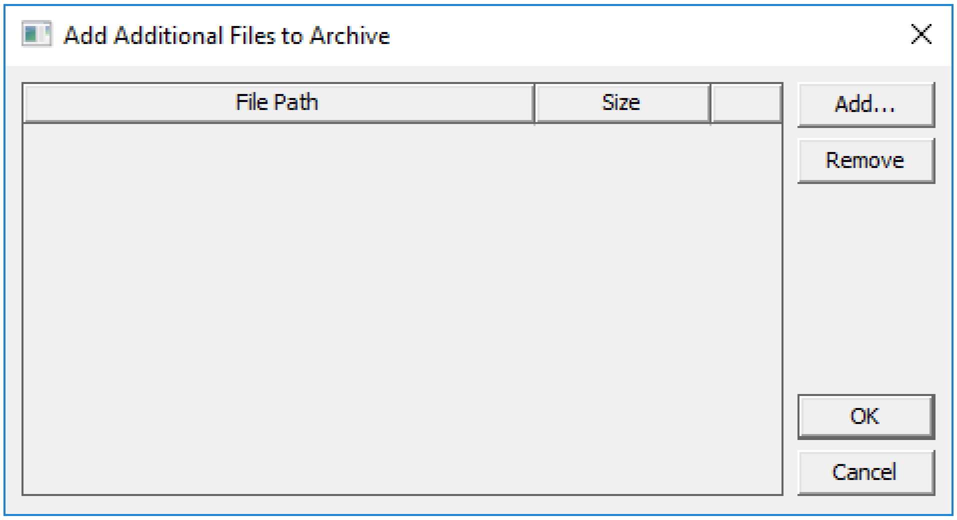


The **Archive Options** dialog box opens.



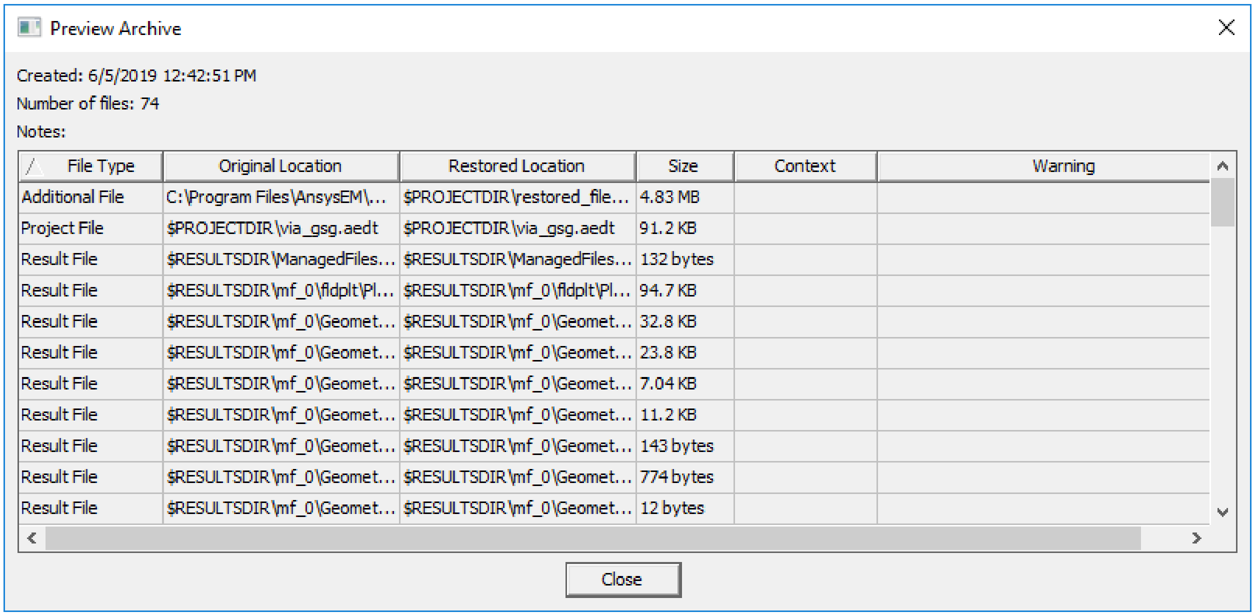
- **Archive Notes** – in this field, you can specify notes that will be visible when previewing the archive. These notes can be viewed from the preview dialog box without restoring the archive.
- **External Files** – selecting this check box causes all external files to be included in the archive. These include any existing files associated with the project, such as linked files, or files added through the **Project** > **Insert Documentation File** command or **Project** > **Data Set** command.
- **Results/Solution Files** – selecting this check box causes the entire results directory to be included in the archive. This may greatly increase the size of the archive file.

- **Additional Files** – clicking this opens the **Add Additional Files to Archive** dialog box.



  - Click **Add** to browse and locate additional files you want to include in the archive.
  - Select and the **Remove** any files listed.
  - Click **OK** to accept changes or **Cancel** any proposed changes.

2. Select any optional items, and make any desired notes in the text field.
3. When you have made your selections, click the **Preview Archive** button in the Archive Options dialog box to view archive contents, file types, the locations of the archive files, and the locations where restoring from archive would place them. Additionally, there may be context or warning information.



To read longer locations, you can drag the column header to expand them.

Previewing an archive before creating the archive can be helpful in order to see exactly what files will be included in an archive, as well as how those files are being relocated. Another purpose of previewing an archive is to view warnings and consider if any additional files need to be added to the archive.

4. When you are ready to create the archive, **Close** the preview and click **OK**.

   A browsing window appears.

5. Specify a name for your file and select the format you want to use from the **Save as type** drop-down menu.
6. Click **OK** to create the archive.

File Relocation

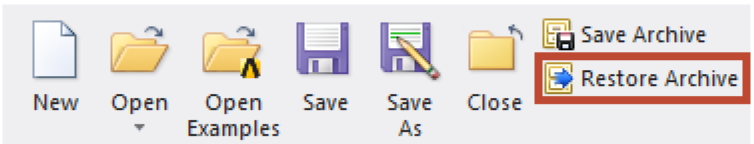
In a project to be archived, external files can be located anywhere on the user's system. One of the goals is for the restored project to be relatively self contained, and to NOT allow the restoring of an archived project to haphazardly write files anywhere on the restoring user's system.

To achieve this, it is sometimes necessary to change the location of files in the archived project so that the external files are located in the project directory. At archive time, any external files not located in the project directory are relocated to the restored_files subdirectory of the project directory in the archived project. Any external files located in the user library or system library will be relocated to the personal library directory. Note that the project file that is written into the archive will be updated to refer to the files at the new locations, and the original project file will remain unaltered.

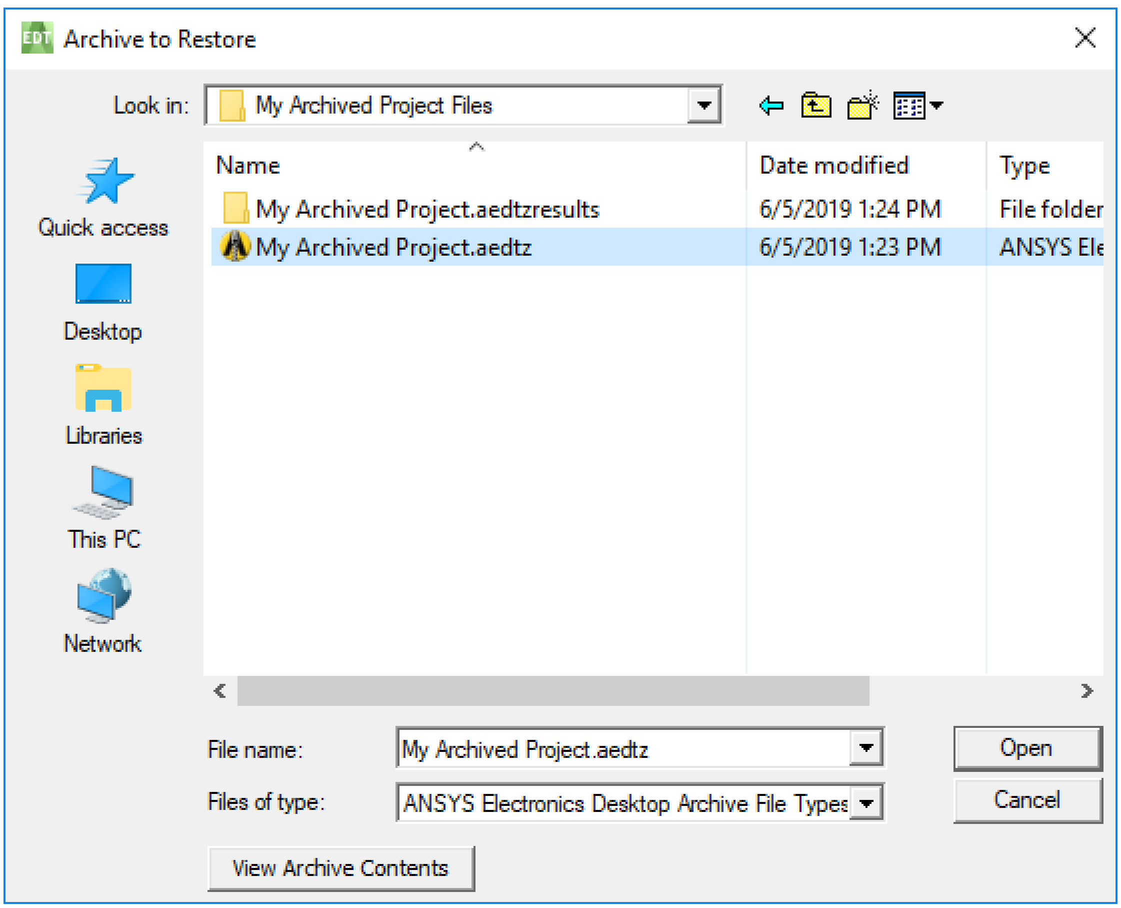
Restoring an Archived Project

To restore an existing archive created with **File** > **Archive...**, use the **File** > **Restore Archive** command.

1. Click **File** > **Restore Archive** or select the **Desktop** tab of the ribbon and click **Restore Archive**.

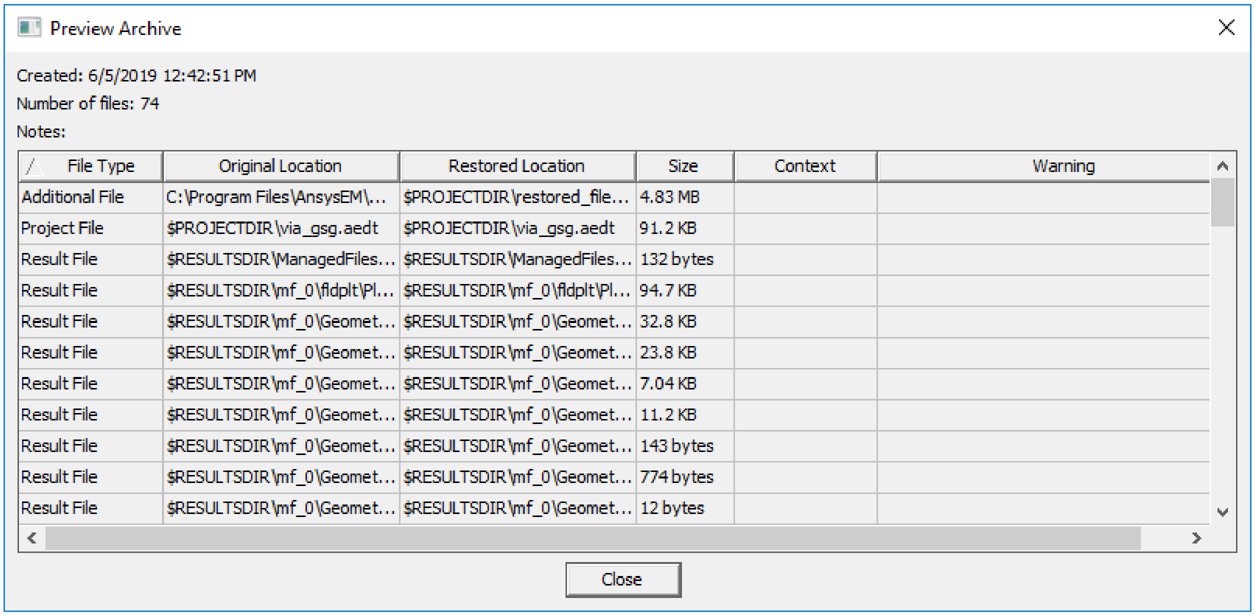


This displays an **Archive to Restore** browser window that lets you navigate your file system for *.aedtz, legacy *.<*solversuffix*>*z* or *.zip archive files.

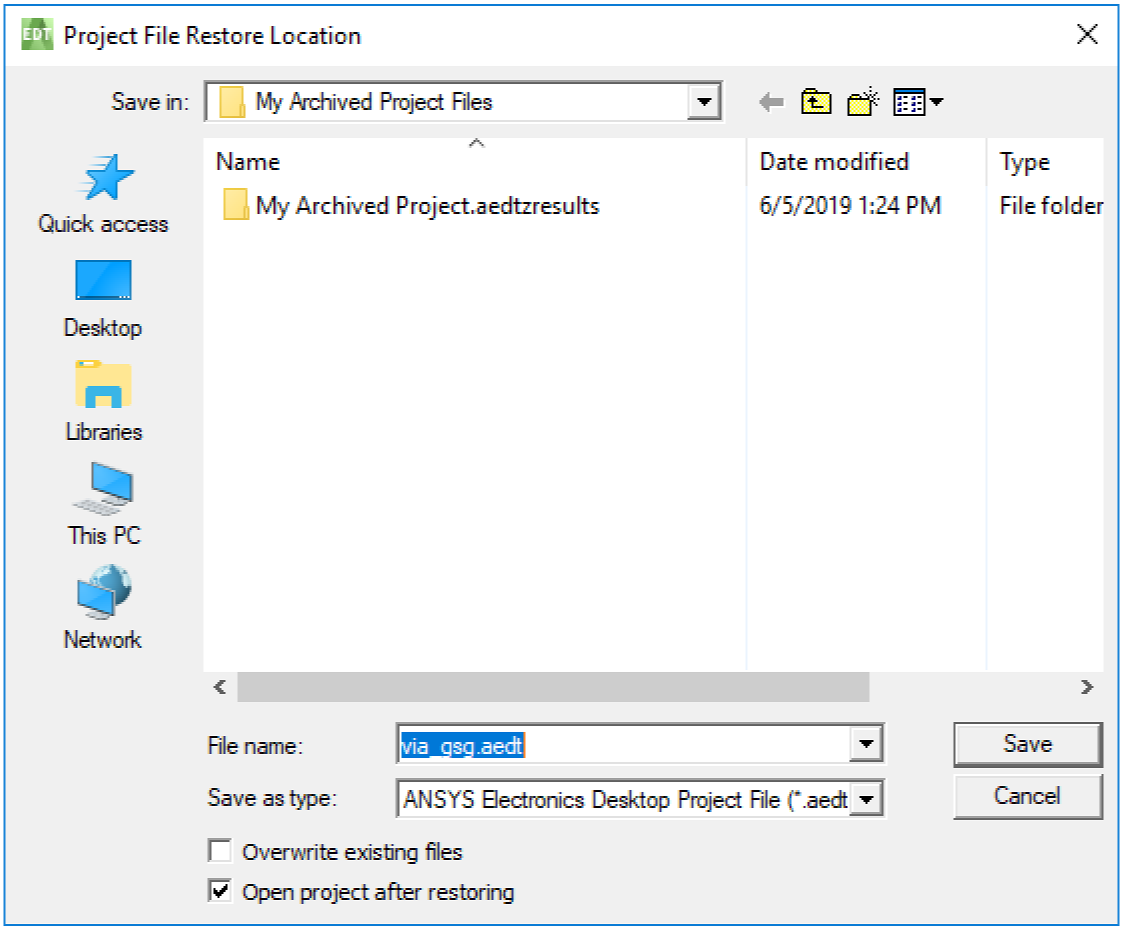


2. If you select a valid archive file, you can click **View Archive Contents** to preview the contents.

The **Preview Archive** dialog box lists all files in the archive, notes, the original and restored locations, and any warnings that were generated at archive time. These warnings may be useful to identify additional steps that are needed to update any files to refer to files which had to be manually added to the archive.



3. Click **Close** to exit the preview.
4. In the browsing window, click **Open** to change to a **Project File Restore Location** browser.

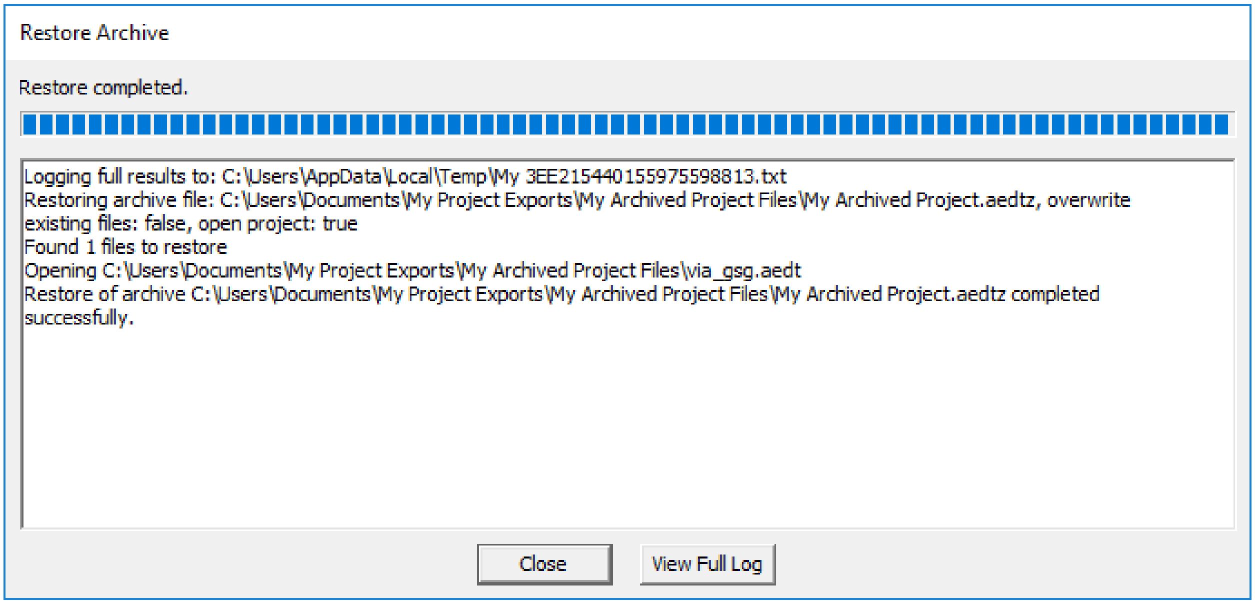


You can edit the file name, and select options:

- **Overwrite existing files** – If this check box is selected, restored files will automatically overwrite existing files during the restore process. If this button is deselected, existing files will not be modified.
- **Open project after restoring** – If this check box is selected, the project will be opened in this instance of the application after all files have been restored.

5. Click **Save** to restore the archived file.

   While restoring an archive, a dialog box displays showing restore results. A progress bar shows the relative progress, and the text window displays important information and warnings.



Electronics Desktop also generates a full log file that contains detailed information about the restore process. The first line in the text window displays the location of the full log file. After the restore has been completed, you can click **View Full Log** to open this log file, or use a text editor and open the file at the specified location.

Deleting a Project or Design

To delete a project or design:

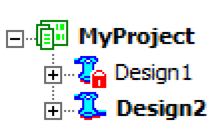
1. In the Project Tree, select a project or design.
2. Click **Edit** > **Delete**, or press the **Delete** key.
3. Confirm the warning box to complete the delete operation.

   The project or design is removed from the Project tree.

Setting Read Only Designs

Designs can be set as either Read Only or Full Access. Full Access is the default status of a design and allows you to modify the design. In Read Only mode, you may only run the design or link it to another design. Setting a design as Read Only protects it from accidental modification.

Read Only designs are marked with a red lock in the Project Manager:



To change a single design's designation:

1. In the **Project Manager**, right-click the design.
2. Select either **Convert to Read Only** or **Convert to Full Access**.

To change the designation of *every design* in a project:

1. In the **Project Manager**, right-click the project.
2. Select either **Convert All Designs to Read Only** or **Convert All Designs to Full Access**.

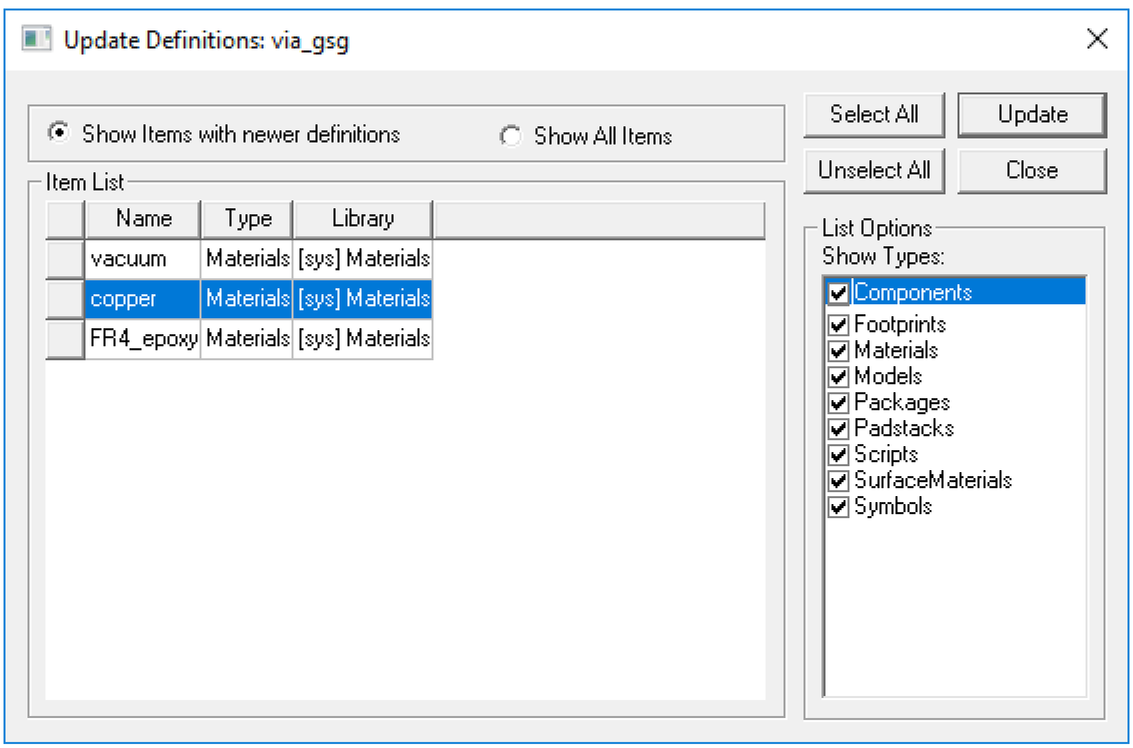
> **Note:**
>
> Read Only settings are not saved when you save the project file.

Updating Design Components

To update components defined in the current design:

1. Click **Tools** > **Project Tools** > **Update Definitions**.

   The **Update Definitions** dialog box appears.



2. Select either **Show Items with newer definitions** or **Show All Items**.

3. From the **Show Types** list in the **List Options** section, select the types of definitions you want to show in the **Item List**.
4. Select the item(s) you want to update from the **Item List**, or use **Select All.**
5. Click **Update**.

   A message appears telling you the update was successful.

6. Click **OK** to close the message.

When you are finished updating definitions, click **Close**.

Undoing and Redoing Commands

The **Edit** menu contains **Undo and** Redo commands.

Use the **Undo** command to cancel the last action you performed on the active project or design.

Use the **Redo** command to reperform an action you previously undid.

Both commands are useful for dealing with unintended actions related to project management, model creation, and post-processing.

To use **Undo** or **Redo**:

1. In the **Project Manager** window, do one of the following:
   - To undo or redo your last project-level action, such as inserting a design or adding project variables, click the project icon.
   - To undo or redo your last design-level action, such as drawing an object or deleting a field overlay plot, click the design icon.

   > **Note:**
   >
   > You cannot undo an analysis that you've performed on a model—that is, the **EMIT** > **Analyze** command.

2. Click **Edit** > [**Undo/Redo**], or click the **Undo** or **Redo** button (Undo Redo) on the **Desktop** tab.

   Your action is undone or redone.

> **Important:**
>
> When you save a project, Electronics Desktop clears the entire undo/redo history for the project and its designs.

Inserting a Documentation File

You may want to add a documentation file to the project tree.

1. Click **Project** > **Insert Documentation File**.

   This opens a file browser that lets you navigate your file system.

2. Select the documentation file and click **OK**.

   The documentation file is placed in the Project Tree.

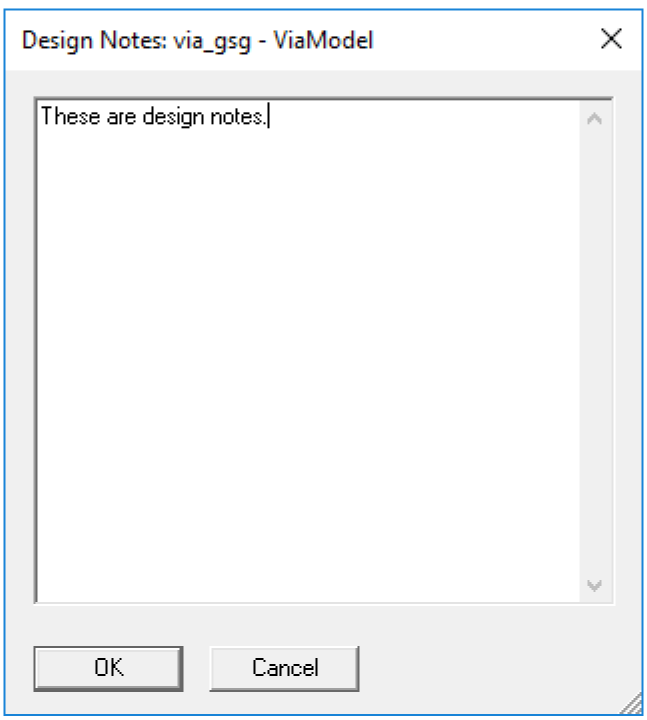
Working with Design Notes

You can save design notes, such as information about the creation date or a description of the device being modeled. This is useful for keeping a running log.

To add notes:

1. Click **EMIT** > **Edit Notes**.

   The **Design Notes** window appears.



2. Click in the window and type your notes.
3. Click **OK** to save the notes with the current project.

To edit notes, use the same window or double-click the **Notes** icon (Notes) in the Project Tree.

To delete the existing notes for a design:

1. Select the **Notes** icon in the Project Tree.
2. Click **Edit** > **Delete**, or right-click and select **Delete**.

   The Notes icon is removed from the Project Tree.

> **Note:**
>
> Notes are used to document aspects of designs only.
>
> For project-level documentation, you can insert a documentation file into a project with the **Project > Insert Documentation File** command.

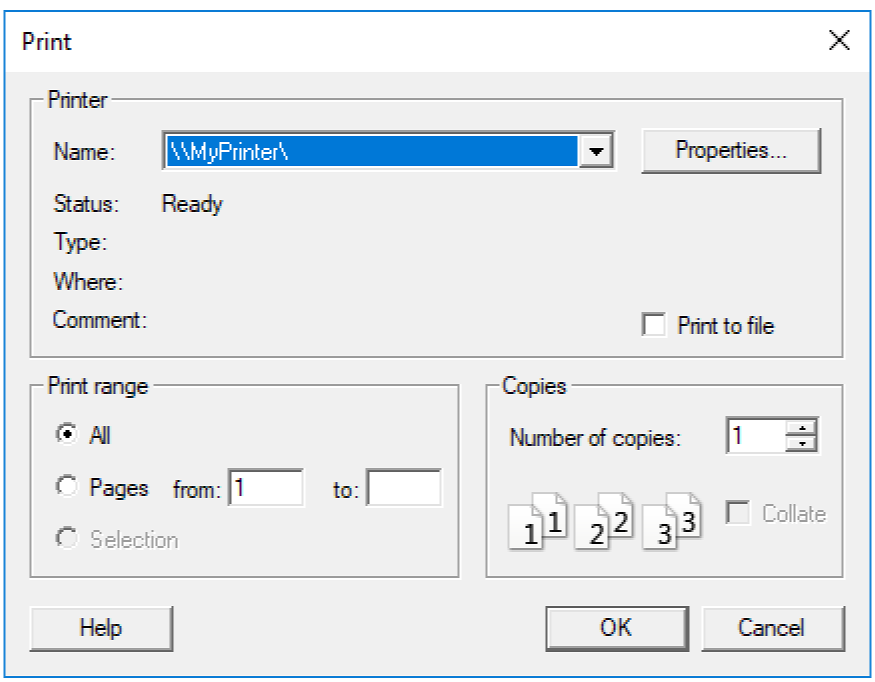
Printing

Electronics Desktop printing commands enable you to send an image of the active window to the printer.

To print the project:

1. Click **File** > **Print**.

   A dialog box similar to the following one appears:



2. Print options will vary based on your printer. Specify your desired options.
3. Click **OK** to print.

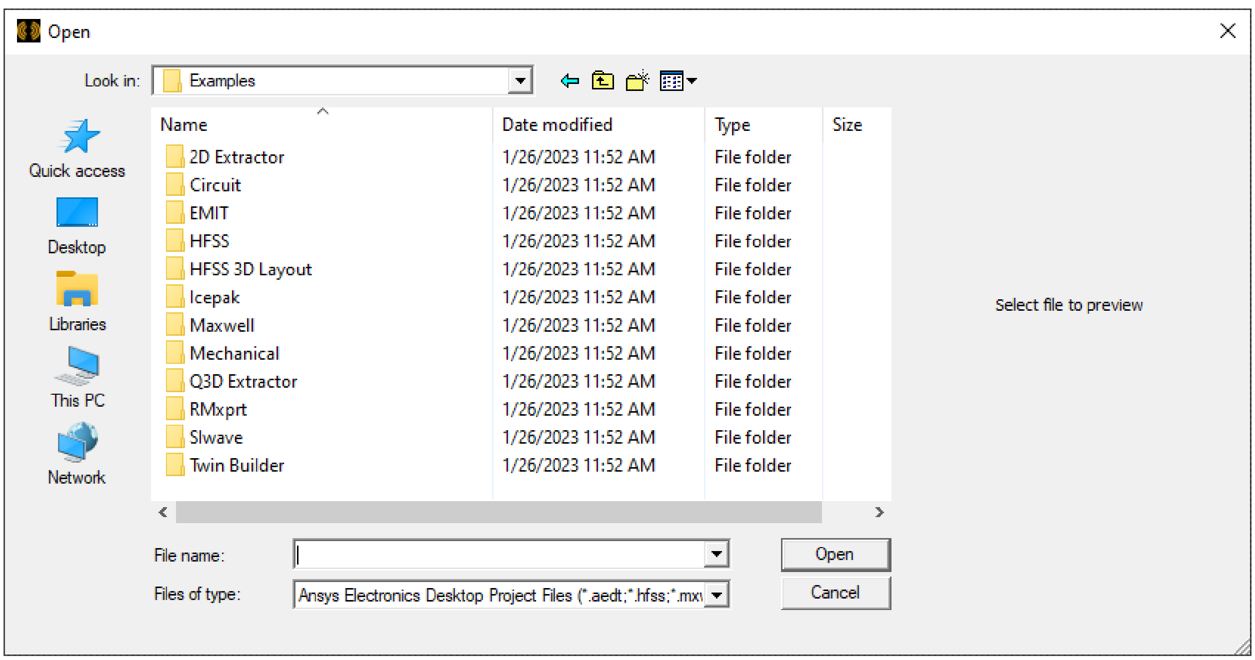
To change additional print settings, click **File** > **Page Setup**.

To preview what your printout will look like, click **File** > **Print Preview**.

> **Important:**
>
> To print from Ansys software on Linux, you must first configure a printer from the MainWin control panel.
>
> For more information, consult documentation for your specific instance of Linux.

Example Projects

Your Ansys Electronics Desktop installation includes an example directory containing projects folders for several kinds of designs.



Example projects are organized by the design type.

Several of these projects are associated with detailed getting started guides.

Working with Variables

A variable is a numerical value, mathematical expression, or mathematical function that can be assigned to a design parameter in **EMIT**. You can assign a variable to any dimension, material property, or output value. Variables are useful in the following situations:

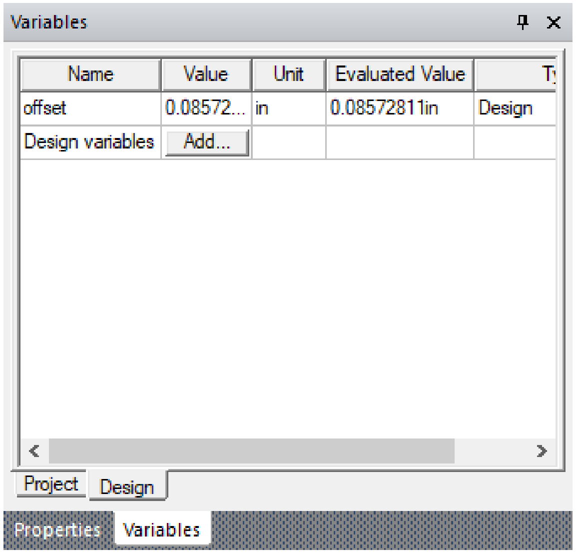
- You expect to change a parameter often.
- You expect to use the same parameter value often.
- You intend to animate a plot against a variable.

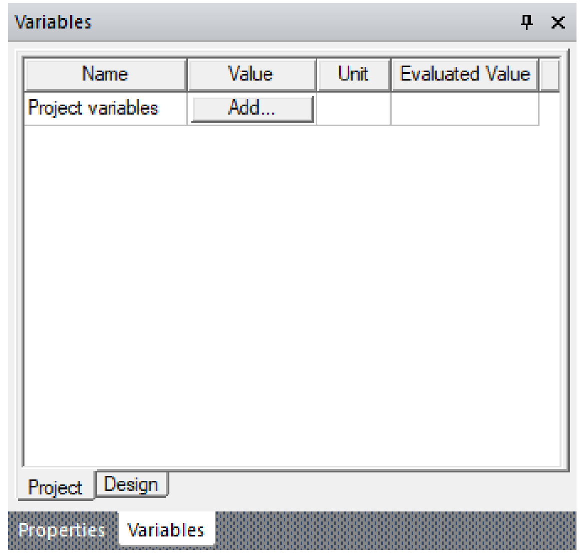
Variables can be regular variables, array index variables, or arrays. A project variable's value can reference other project variables or intrinsics. Values can be a number, variable, or mathematical expression.

There are two types of variables in EMIT:

Project Variables	A project variable can be assigned to any parameter value in the EMIT project in which it was created. EMIT differentiates project variables from other types of variables by prefixing the variable name with dollar sign symbol ($). You can manually include the $ in the project variable's name, or EMIT will automatically append the project variable's name after you define the variable. Project variables can be designated as Design, ArrayIndex or Separator variables .
Design (Local) Variables	A design variable can be assigned to any parameter value in the EMIT design in which it was created. From the Design Variables **Properties** dialog box, you can Add, Add Array, Edit, or Remove design variables. Design Variables can be designated as Design, ArrayIndex or Separator variables

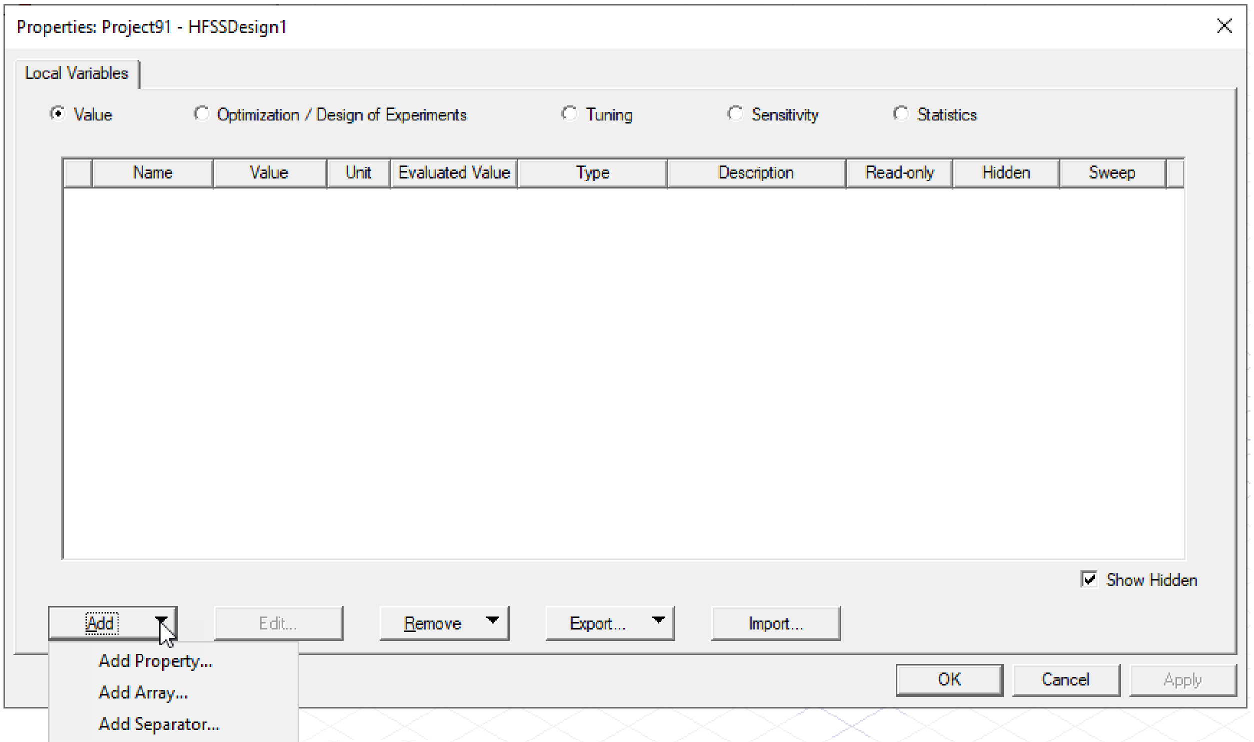
Clicking **View** > **Variables** brings up a dockable variable window that is associated with the active project and/or design. When there is an active project, there will be a corresponding project variable tab. When there is an active design, there will be a corresponding design variable tab. Each tab contains an **Add...** button allowing creation of new variable of this type. If variables exist for the Project or Design, they are shown in the corresponding tab.





Importing and Exporting Variables

Once you have defined Design and/or Project Variables, you can export them to a csv file, and import them to another project or design. If there are naming conflicts on import, these are flagged as errors. The **Properties** dialogs for Design Variables and Project variables include **Import...** and **Export...** buttons.



Working with Project Variables

Project variables are available across all hierarchical levels of a project. A project variable can be assigned to a parameter value in the EMIT project in which it was created. EMIT differentiates project variables from other types of variables by prefixing the variable name with the dollar sign symbol ($). You can manually include the $ in the project variable's name when you create it, or EMIT will automatically append it after you define the variable.

The **Project Variables** menu item allows you to access three tabs:

- **Project Variables** – variables, which you define, that apply to the entire project.
- **Intrinsic Variables** – pre-defined variables that cannot be changed. See: List of Intrinsic Variables.
- **Constants** – pre-defined constants that cannot be changed. See: List of Constants.

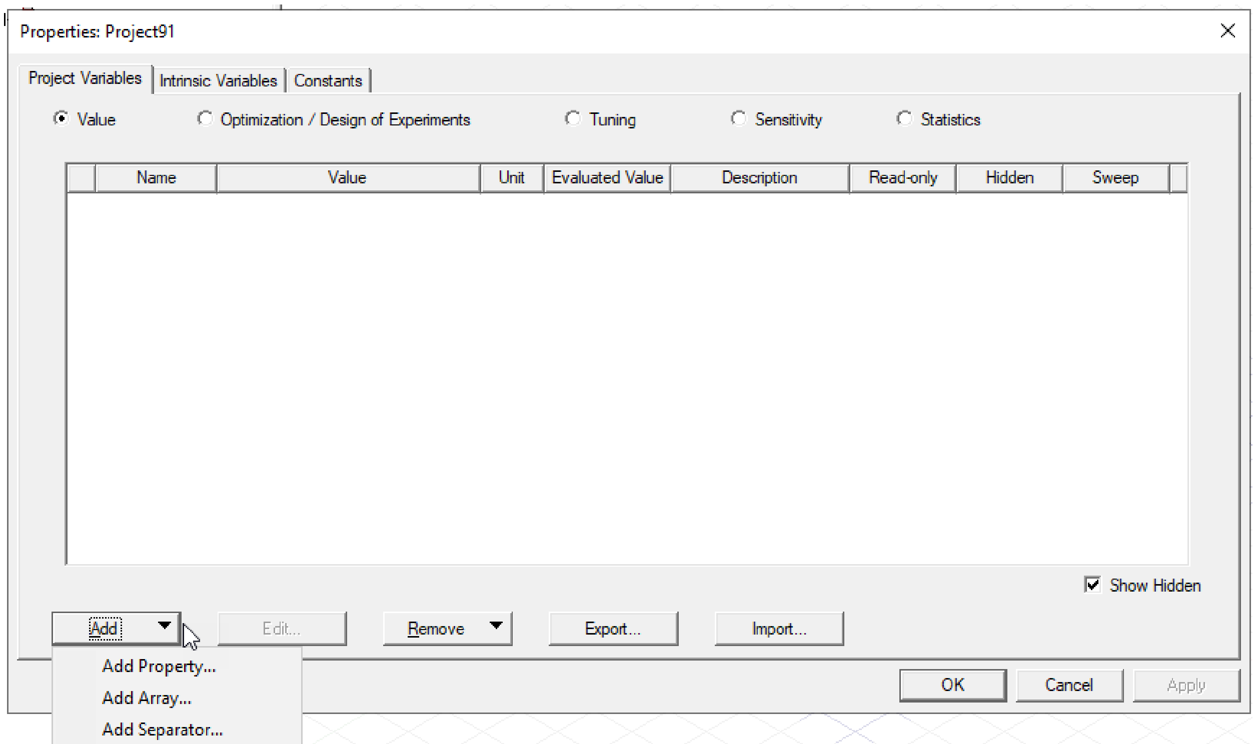
Adding a Project Variable

To define a Project Variable:

1. Access **Project Variables** one of three ways:
    - Click **Project** > **Project Variables**.
    - Right-click the project name in the Project Tree and select **Project Variables**.

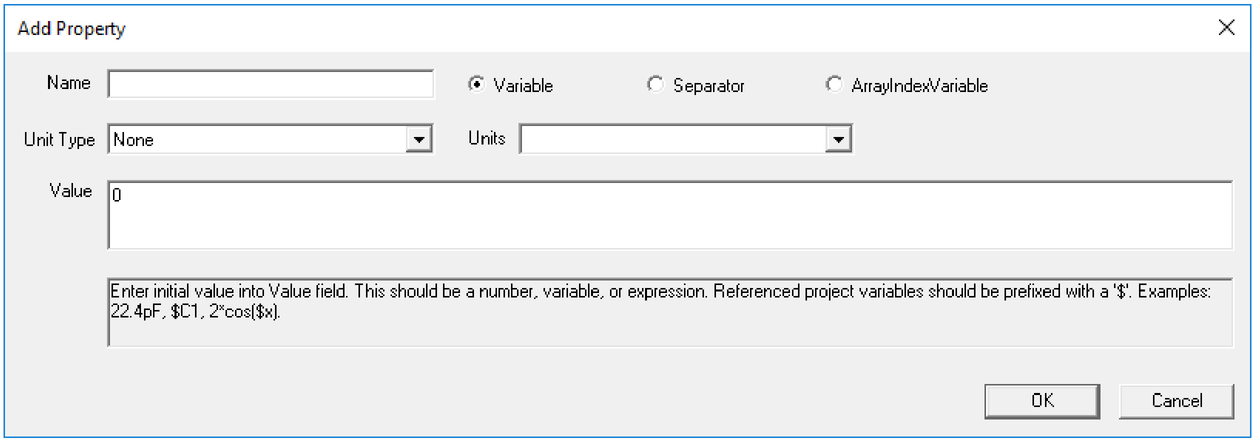
- From a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Project Variables**.

The **Properties** window appears, on the **Project Variables** tab.



2. From the **Project Variables** tab, click **Add**.

   The **Add Property** dialog box appears.



3. Enter information for the variable, as applicable:
   - **Name** – project variable names must start with the dollar sign symbol ($), followed by a letter. The name can contain alphanumeric characters and underscores (_). You cannot use the names of Intrinsic Variables or pre-defined Constants.
   - **Variable Type** – use the **Variable**, **Separator**, and **ArrayIndexVariable** radio buttons to select the variable type.

     Your selection impacts which properties you can edit:

Variable Type	Editable Properties
Variable	Unit Type, Units, Value.
Separator	Name. A separator variable provides a bolded name for a blank line to facilitate grouping variables in variable lists.
Array Index Variable	Associate Array Variable, Value

   - **Unit Type** – for Variables, use the drop-down menu to select a type from the list (for example, Charge, Density, Energy, ...). “None” is the default.

     When you select a Unit Type, the choices in drop-down menu for the Units text box adapt to that unit type. For example, selecting Length as the Unit Type causes the Unit menu to show a range of metric and english units for length. Similarly, if you select the Unit Type as Resistance, the Units drop down lists a range of standard Ohm units.
   - **Units** – for Variables, use the drop-down menu to select a unit of measure.
   - **Value** – for Variables and ArrayIndexVariables, enter a number, variable, or mathematical expression. The quantity entered will be the current (or default) value for the variable. If the mathematical expression includes a reference to an existing variable, this variable is treated as a dependent variable. The units for a dependent

variable will automatically change to those of the independent variable on which the value depends. Additionally, dependent variables, though useful in many situations, cannot be the direct subject of optimization, sensitivity analysis, tuning, or statistical analysis.

For ArrayIndexVariables, the index reference can be a constant (for example, 1), an index (for example, ii) or an expression (for example, ii + 1). This allows you to sweep the index and simulate for different values that are stored in the array variable itself. In particular, it also enables you to sweep different text strings. This allows you to set a property to different string values as the index is swept.

> **Warning:**
>
> If you include the variable's units in its definition (in the **Value** text box), do not include the variable's units when you enter the variable name for a parameter value.

- **Associate Array Variable** – for ArrayIndexVariables, select an associated array from the drop-down list. You must have previously created an array.

4. Click **OK**.

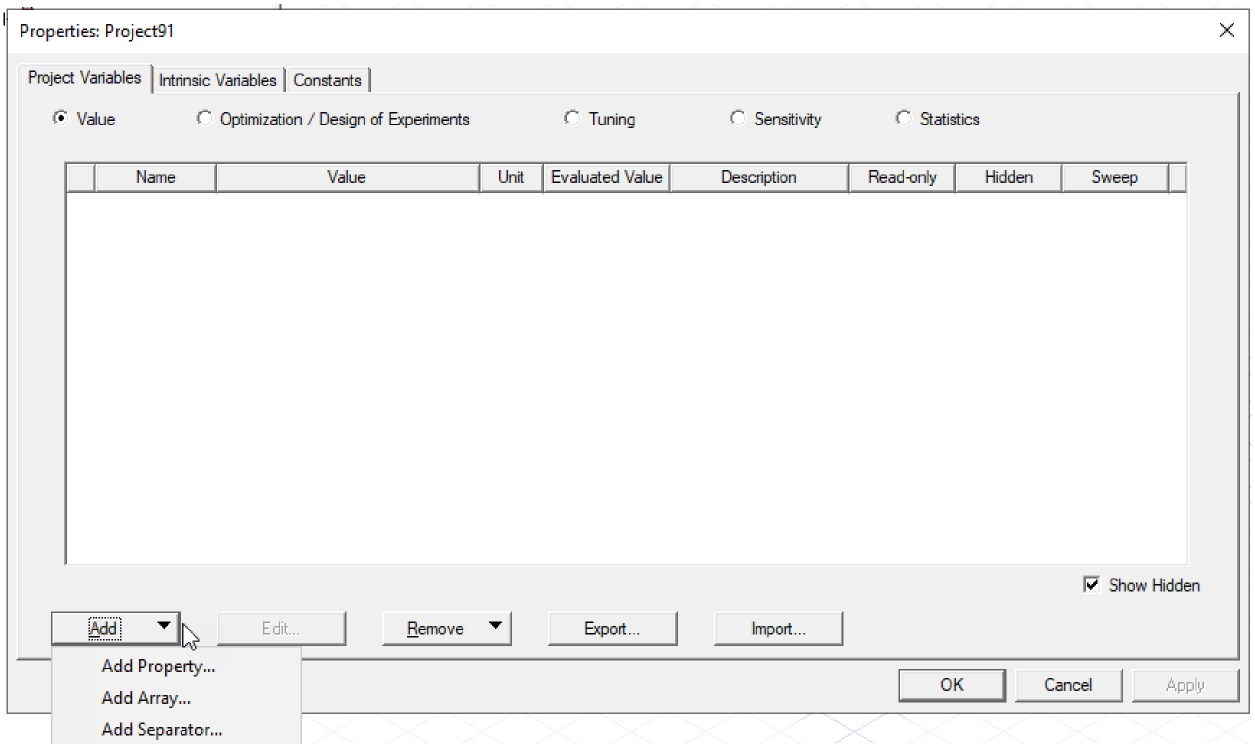
   The new variable appears in the list. You can sort project variables by clicking the Name column header. Clicking once sorts them in ascending order, noted by a triangle pointing up. Clicking against sorts in descending order, noted by a triangle pointing down. Clicking a third time sorts in original order, with no triangle.

5. If desired, use the check boxes to designate a variable as **Read-only**, **Hidden**, or **Sweep**.
   - **Read-only** – when selected, the variable's name, value, unit, and description cannot be modified.
   - **Hidden** – hidden variables do not appear in the **Properties** window unless **Show Hidden** is selected.
   - **Sweep** – allows you to designate variables to include in solution indexing as a way to permit faster post-processing. Variables with the Sweep check box cleared are not used in solution indexing. If a solution exists, selecting or clearing a variable's Sweep setting produces a warning that the change will invalidate existing solutions. To continue, click **OK** to dismiss the warning.
6. Click **Apply** to apply changes.
7. Click **OK** to exit the window.

The new variable can be assigned to a parameter value in the project in which it was created. You can enable defined variables for Optimization/Design of Experiments, Tuning, Sensitivity, or Statistics. See: Optimetrics.

Importing and Exporting Project Variables

Once you have defined Project Variables, you can export them to a csv file, and import them to another project or design. If there are naming conflicts on import, these are flagged as errors. The **Properties** dialogs for and Project variables include **Import...** and **Export...** buttons.



When you click **Export....** a dialog lets you name the file and select a location. When you **Save**, if the file exists, an overwrite prompt should pop up.

Deleting a Project Variable

To remove a Project Variable:

1. Access **Project Variables** as described above.

   The **Properties** window appears, on the **Project Variables** tab.

2. Remove one or more variables:
   - To remove a specific variable, select it and click **Remove** > **Remove Selected**.
   - To remove all unused variables, click **Remove** > **Remove All Unused**.
   - To force the removal of all unused variables, including those in the project's undo/redo history, click **Remove** > **Force Remove All Unused**.

Editing a Project Variable

To edit a Project Variable:

1. Access **Project Variables** as described above.

   The **Properties** window appears, on the **Project Variables** tab.
2. Select the variable you want to edit and click **Edit**.
3. Change the properties as desired and click **OK**.

Adding a Project Variable Array

You can define array variables that contain numbers or strings. Number array variables can be used in component property expressions, while string array variables can be used in certain component property values.

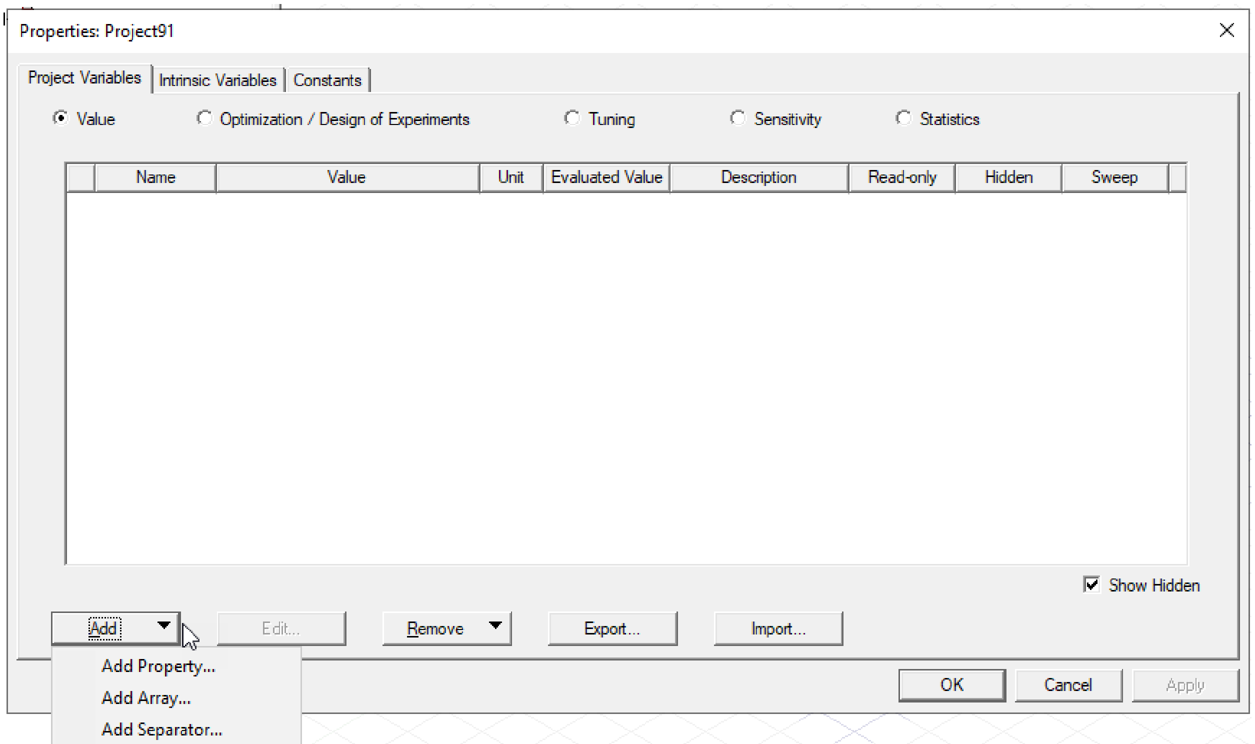
> **Note:**
>
> Text array variables can be used in certain component property values, but not all, and cannot be combined with operators to form more complex expressions.
>
> The index for a text array variable reference can either be a constant (1) or can be an index (ii) or even an expression (ii + 1). This allows you to sweep the index and simulate for different values that are stored in the array variable itself. In particular, it also enables you to sweep different text strings. This allows you to set a property to different string values as the index is swept. The following are properties that currently allow text array variables:
>
> - V_PRBS
> - V_PRBS_JITTER
> - V_PRBSD
> - V_PRBSG_JITTER
> - V_PSK
> - V_QAM
> - V_CPM

To define an array variable, you must first define a Project Variable Array:

1. Access **Project Variables** one of three ways:
   - Click **Project** > **Project Variables**.
   - Right-click the project name in the Project Tree and select **Project Variables**.

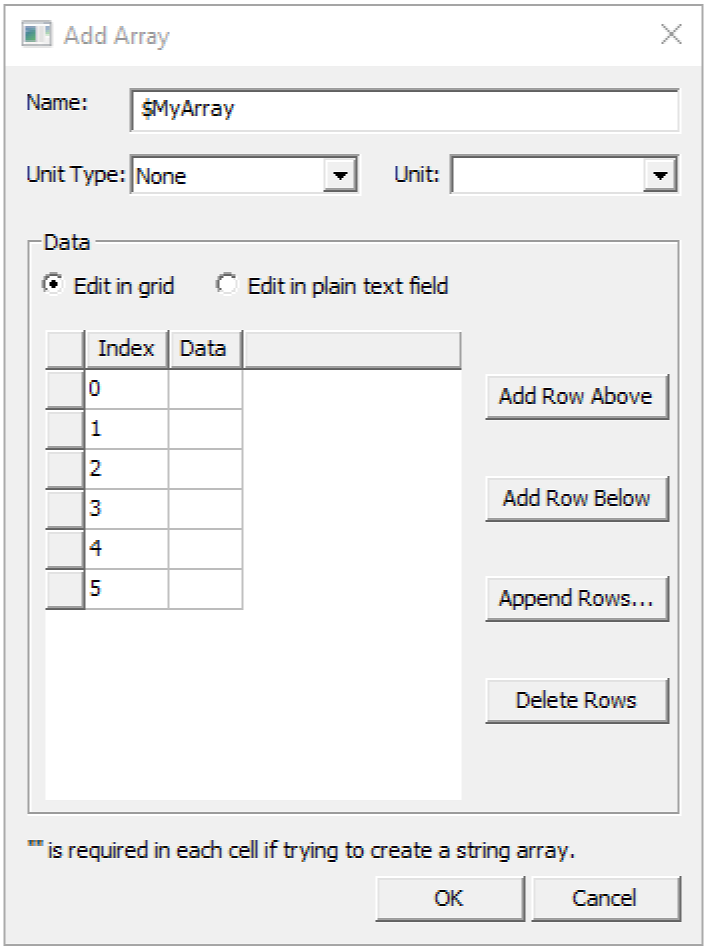
- From a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Project Variables**

The **Properties** window appears, on the **Project Variables** tab.



2. Click **Add Array** > **Add Array...**.

   The **Add Array** window appears.



The value grid displays each array item's **Index** number, and the **Data** associated with that index number.

3. Give the variable array a **Name**, and select a **Unit Type** and **Unit** from the drop-down menus.
4. Use the control buttons to add, delete, and reposition rows in the value grid at left. The default is **Edit in grid**, but you may select **Edit in plain text field**.

> **Note:**
>
> Quotation marks (“ ”) are *required* as delimiters when array values are entered in either the grid or text field.

5. When you have finished entering values, click **OK**.

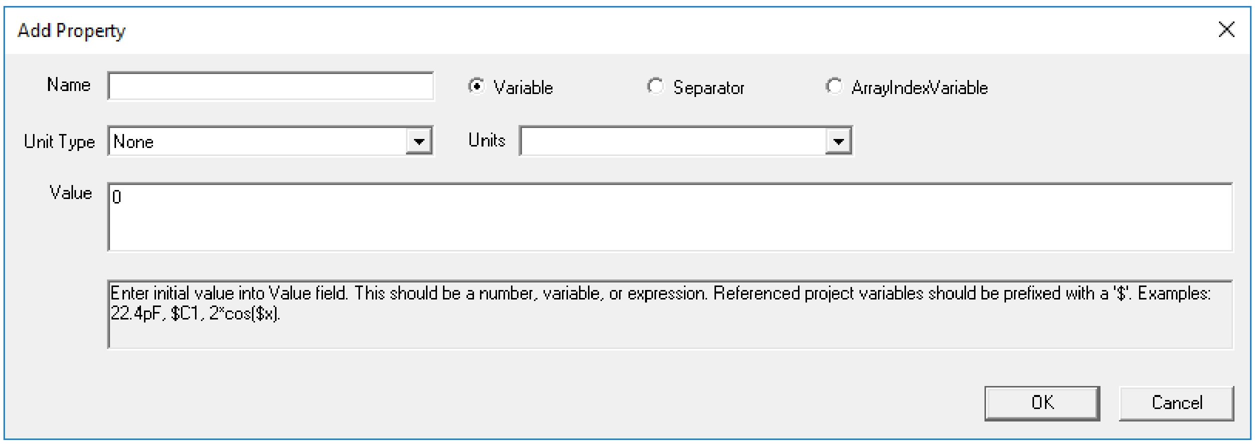
   The array appears in the **Project Variables** list.

You can add a variable to the array by adding an ArrayIndexVariable on the **Project Variables** tab.

1. Access **Project Variables** one of three ways:
   - Click **Project** > **Project Variables**.
   - Right-click the project name in the Project Tree and select **Project Variables**.
   - From the **Setup Optimization** dialog box, click **Edit Variables** > **Edit Project Variables**.

   The **Properties** window appears, on the **Project Variables** tab.
2. From the **Project Variables** tab, click **Add**.

   The **Add Property** dialog box appears.

3. Enter information for the variable, as applicable:
   - **Name** – project variable names must start with the dollar sign symbol ($), followed by a letter. The name can contain alphanumeric characters and underscores (_). You cannot use the names of Intrinsic Variables or pre-defined Constants.
   - **Variable Type** – select ArrayIndexVariable.
   - **Value** – enter a number, variable, or mathematical expression. The quantity entered will be the current (or default) value for the variable. If the mathematical expression includes a reference to an existing variable, this variable is treated as a dependent variable. The units for a dependent variable will automatically change to those of the independent variable on which the value depends. Additionally, dependent variables, though useful in many situations, cannot be the direct subject of optimization, sensitivity analysis, tuning, or statistical analysis.

     For ArrayIndexVariables, the index reference can be a constant (for example, 1), an index (for example, ii) or an expression (for example, ii + 1). This allows you to sweep the index and simulate for different values that are stored in the array variable itself. In particular, it also enables you to sweep different text strings. This allows you to set a property to different string values as the index is swept.

> **Warning:**
>
> If you include the variable's units in its definition (in the **Value** text box), do not include the variable's units when you enter the variable name for a parameter value.

- **Associate Array Variable** – for ArrayIndexVariables, select an associated array from the drop-down list. You must have previously created an array.

4. Click **OK**.

List of Intrinsic Variables

The Ansys Electronics Desktop software recognizes a set of intrinsic variables that can be used to define expressions. Intrinsic variable names are reserved and may not be used as user-defined variable names.

To view the list of Intrinsic Variables, click **Project** > **Project Variables** and select the **Intrinsic Variables** tab.

Name	Unit	Description
_Empty		Empty value, taken to be model default by simulator (Twin Builder)
_I1	mA	Armature current; **this variable is only used in RMxprt**.
_I2	mA	Exciting current; **this variable is only used in RMxprt**.
_I3	mA	Current; **this variable is only used in RMxprt**.
_I4 through _I9		Not used
_t,_u,_v		Variables to define parametric equation-based curve.
_V1 to **_V9**	mV	Port Voltage in user-defined model (V).
Ang	Ang	Angle. Postprocessing variable
Budget Index		Postprocessing variables; cannot be set by the user
Distance	mm	
Electrical Degree	deg	Electric degree of the rotating machine; cannot be set by the user.
F	GHz	Frequency of the circuit/system analysis
F1, **F2**, **F3**	GHz	Frequency tones 1, 2, 3 in harmonic balance analysis. (Hz).

Name	Unit	Description
FNoi	GHz	Offset noise frequency in harmonic balance noise analysis.
Freq	Hz	Frequency for analysis and postprocessing; cannot be set by the user
Hmax	ns	
Hmin	ns	
Ia, **Ib**	mA	Postprocessing variables; cannot be set by the user
Index		Identifier for a data point; cannot be set by the user
IWave Phi, **IWave Theta**	deg	Incident wave spherical coordinate variables. Phi is the angle from the origin in the z direction and Theta the angle from the x-axis.
Normalized Deformation, **Normalized Distance**		Postprocessing variables; cannot be set by the user
OP	mW	Postprocessing variables; cannot be set by the user
Pass		Post processing variables, not settable by the user.
Phase	deg	Angle of a complex number.
Phi	deg	The angle measured from the x-axis and can be from 0 to 360 degrees.
R	mm	R is the cylindrical coordinate system variable.
Rho		Rho is the spherical coordinate system variable.
Rspeed	rpm	Speed of the machine.
Spectrum	GHz	Postprocessing variable.
Temp	cel/deg	Analysis temperature (deg)
Tend	ns	
Theta	deg	The angle measured from the z-axis, which is the axis perpendicular to the plane of the work space, and must be from 0 to 180 degrees.
Time	ns	Time point in transient analysis
Time0	ns	Time 0 point in a transient analysis
Vac, **Vbe**, **Vce**, **Vds**, **Vgs**	mV	Post processing variables; cannot be set by the user
X,**Y**,**Z**	mm	Point coordinate variables in the modeler.
ZAng, **XRho**	deg	A spherical coordinate system; cannot be set by the user

List of Constants

To view the list of constants, click **Project** > **Project Variables** and select the **Constants** tab.

Name	Value	Description
abs0	-273.15	Absolute zero (C)
boltz	1.3806503e-23	Boltzmann constant (J/K)
c0	299792458	Speed of light in a vacuum (m/s)
e0	8.854187817e-12	Permittivity of vacuum (F/m)
elecq	1.602176462e-19	Electron charge (C)
eta	376.730313461	Impedance of vacuum (Ohm)
false	0	Boolean False
g0	9.80665	Acceleration due to gravity of Earth (m/s^2)
mathE	2.718281828	Euler's number (Napier's constant)
pi	3.1415926535898	Ratio of circumference to diameter
planck	6.6260755e-34	Planck's constant (m^2*kg/s)
true	1	Boolean True
u0	1.2566370614359e-06	Permeability of vacuum (H/m)

Working with Design Variables

Design Variables (Local Variables) are only available within the EMIT design for which they are defined. A design variable can be assigned to a parameter value in the design. You can also add a variable defined with an array of values. Choosing **Add Property** pops up an **Add Property** dialog with default radio button on Variable; choosing **Add Array** pops up an **Add Array** dialog; choosing **Add Separator** pops up an **Add Property** dialog with default radio button on Separator.

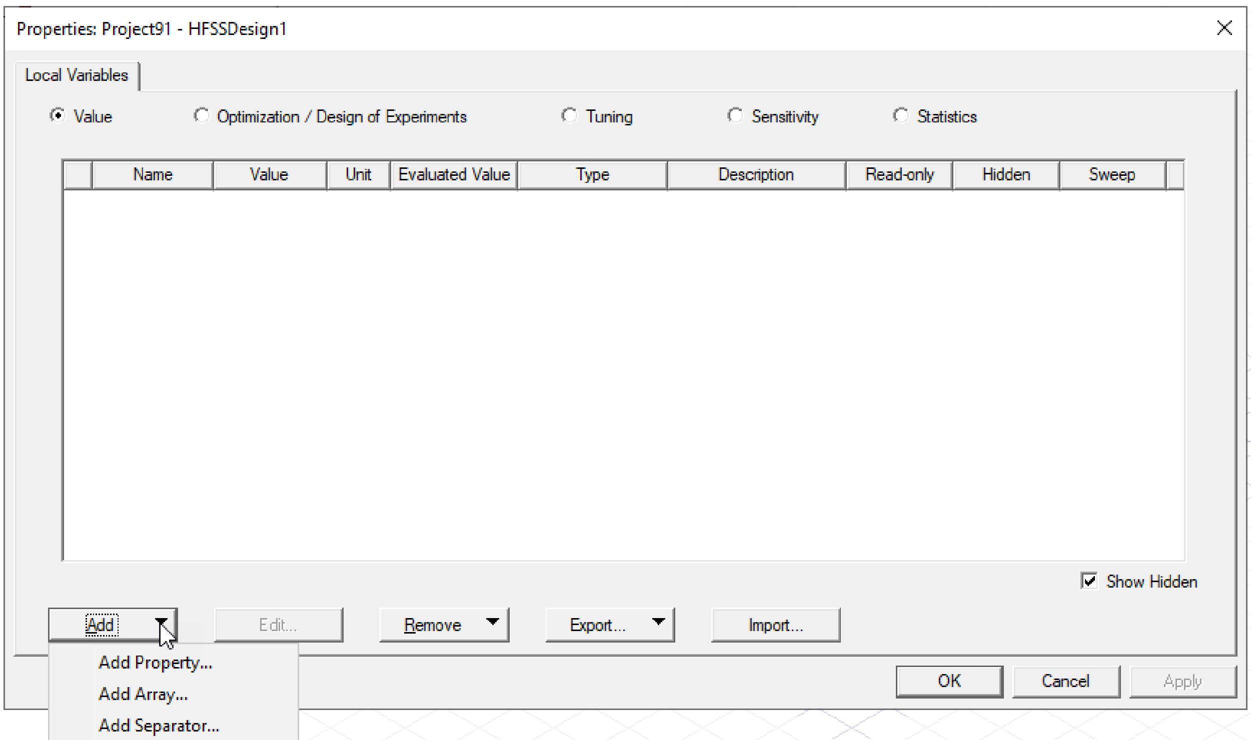
The **Design Properties** menu item allows you to access up to three tabs, depending on your design:

- **Parameter Defaults** – local variables with default values that can be overridden in instances of a design. For example, if three subcircuit instances contain a parameter default C1 that is defined as equal to 11.3pF, C1 may be overridden as 11.8pF in the first instance, overridden as 10.9pF in the second, and left at its default value of 11.3pF in the third. A property value that has been set by means of a parameter default is called a *passed parameter*. Parameter defaults can be defined from a component or at the design level.
- **Local Variables** – variables, which you define, that apply to the current design only. Local variables can be defined from a component or at the design level.
- **General** – pre-defined design variables that cannot be changed. You cannot use the names of these variables when creating a new variable.

> **Note:**
>
> Not all tabs appear for all designs.

Importing and Exporting Variables

Once you have defined Design Variables, you can export them to a csv file, and import them to another project or design. If there are naming conflicts on import, these are flagged as errors. The Properties dialogs for Design Variables include **Import...** and **Export...** buttons.



Defining Local Variables at the Design Level

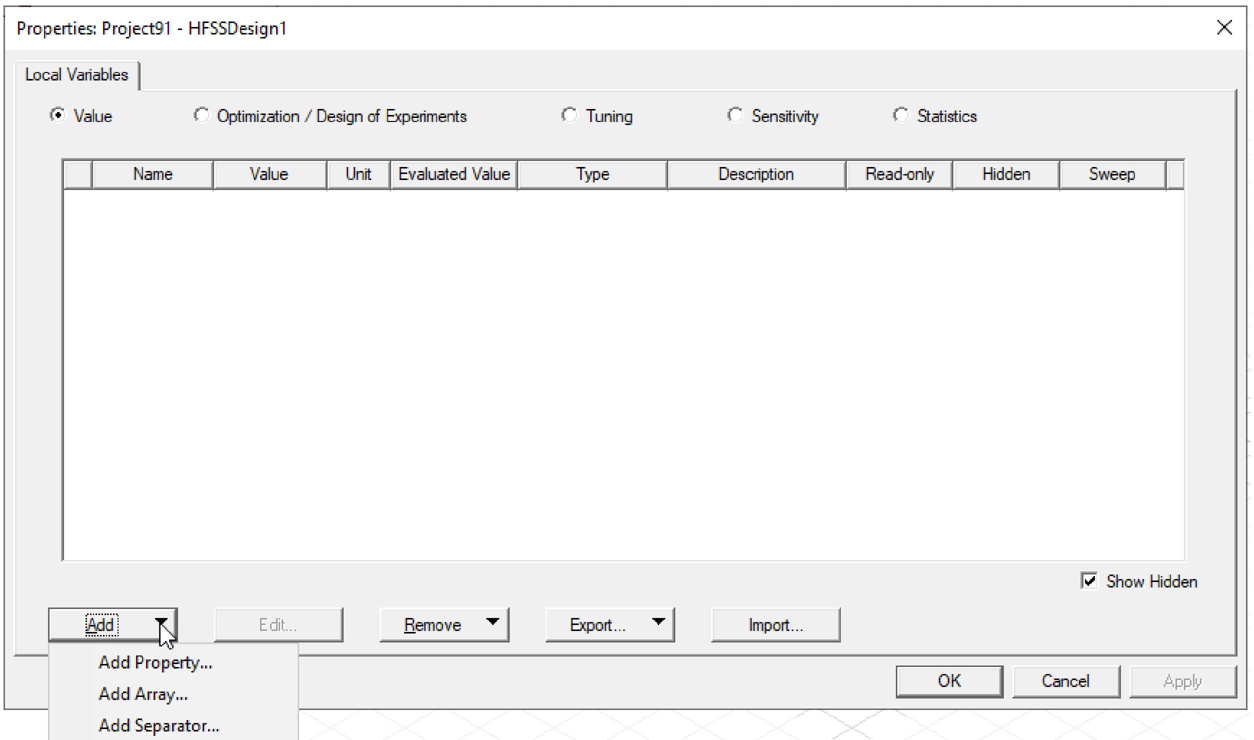
Adding a Local Variable

To define a Local Variable:

1. Access **Design Properties** one of three ways:
   - Click **EMIT** > **Design Properties**.
   - Right-click the design name in the Project Tree and select **Design Properties**.
   - From a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and

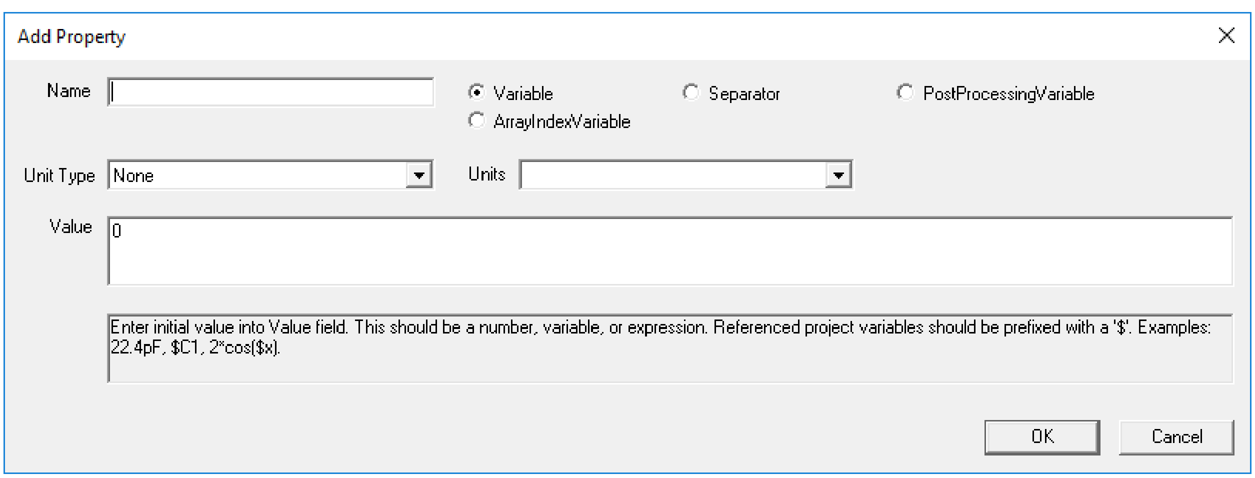
Design Xplorer Setup. Click **Edit Variables** and from the menu select **Edit Design Variables**

The **Properties** window appears with the **Local Variables** tab.



2. Click **Add**.

   The **Add Property** window appears.



3. Enter information for the variable, as applicable:
   - **Name** – the name must start with a letter and can contain alphanumeric characters and underscores (_). You cannot use any names already defined on the **General** tab.
   - **Variable Type** – use the **Variable**, **Separator**, **ArrayIndexVariable**, and **PostProcessingVariable** radio buttons to select the variable type.

     > **Note:**
     >
     > Not all variable types are available for all designs.

     Your selection impacts which properties you can edit:

Variable Type	Editable Properties
Variable	Unit Type, Units, Value.
Separator	Name. A separator variable provides a bolded name for a blank line to facilitate grouping variables in variable lists.
Array Index Variable	Associate Array Variable, Value
PostProcessingVariable	Unit Type, Units, Value.

   - **Unit Type** – for Variables and PostProcessingVariables, use the drop-down menu to select a type from the list (for example, Charge, Density, Energy, ...).
   - **Units** – for Variables and PostProcessingVariables, use the drop-down menu to select a unit of measure.
   - **Value** – for Variables, ArrayIndexVariables, and PostProcessingVariables, enter a number, variable, or mathematical expression. The quantity entered will be the current (or default) value for the variable. If the mathematical expression includes a reference to an existing variable, this variable is treated as a dependent variable. The units for a dependent variable will automatically change to those of the independent variable on which the value depends. Additionally, dependent variables, though useful in many situations, cannot be the direct subject of optimization, sensitivity analysis, tuning, or statistical analysis.

     For ArrayIndexVariables, the index reference can be a constant (for example, 1), an index (for example, ii) or an expression (for example, ii + 1). This allows you to sweep the index and simulate for different values that are stored in the array variable itself. In particular, it also enables you to sweep different text strings. This allows you to set a property to different string values as the index is swept.

> **Warning:**
>
> If you include the variable's units in its definition (in the **Value** text box), do not include the variable's units when you enter the variable name for a parameter value.

- **Associate Array Variable** – for ArrayIndexVariables, select an associated array from the drop-down list. You must have previously created an array.

4. Click **OK**.

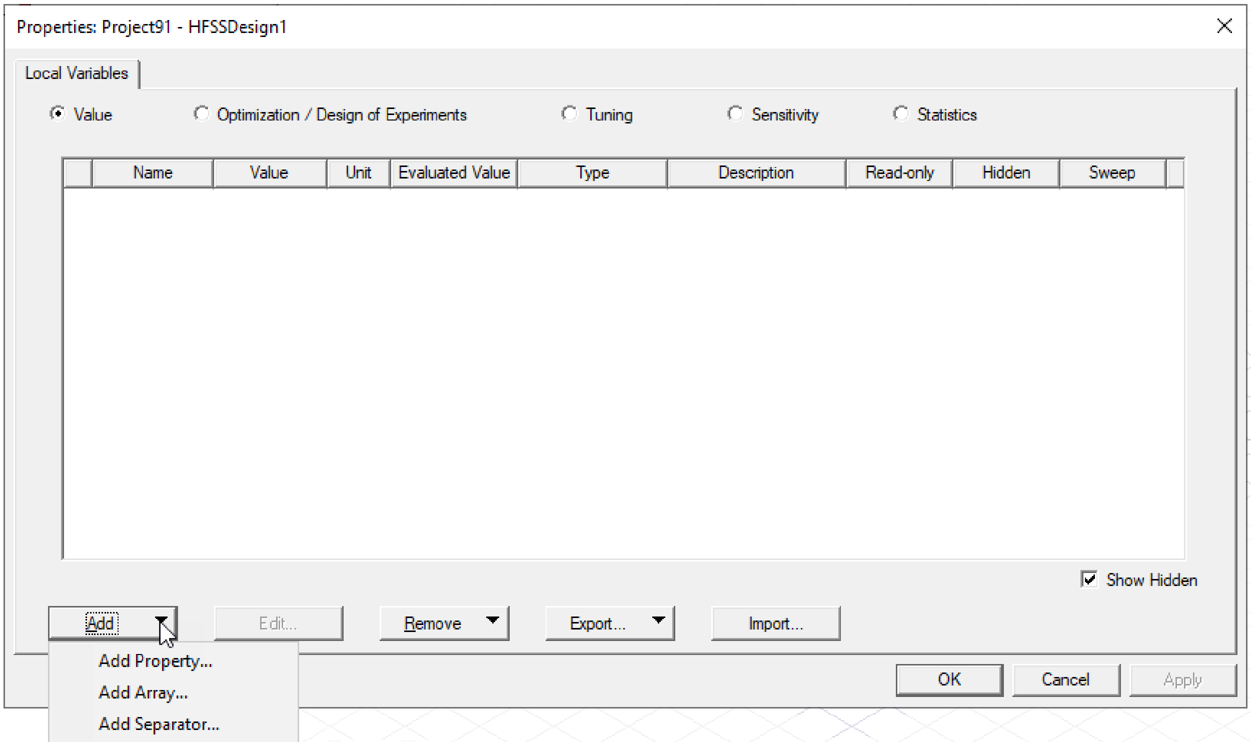
   The new variable appears in the list. You can sort local variables by clicking the Name column header. Clicking once sorts them in ascending order, noted by a triangle pointing up. Clicking against sorts in descending order, noted by a triangle pointing down. Clicking a third time sorts in original order, with no triangle.

5. If desired, use the check boxes to designate a variable as **Read-only**, **Hidden**, or **Sweep**.
   - **Read-only** – when selected, the variable's name, value, unit, and description cannot be modified.
   - **Hidden** – hidden variables do not appear in the **Properties** window unless **Show Hidden** is selected.
   - **Sweep** – allows you to designate variables to include in solution indexing as a way to permit faster post-processing. Variables with the Sweep check box cleared are not used in solution indexing. If a solution exists, selecting or clearing a variable's Sweep setting produces a warning that the change will invalidate existing solutions. To continue, click OK to dismiss the warning.
6. Click **Apply** to apply changes.
7. Click **OK** to exit the window.

The new variable can be assigned to a parameter value in the project in which it was created. You can enable defined variables for Optimization/Design of Experiments, Tuning, Sensitivity, or Statistics. See: Optimetrics.

Importing and Exporting Variables

Once you have defined Design Variables, you can export them to a csv file, and import them to another project or design. If there are naming conflicts on import, these are flagged as errors. The Properties dialogs for Design Variables include **Import...** and **Export...** buttons.



Deleting a Local Variable

To remove a Local Variable:

1. Access **Local Variables** as described above.
2. Remove one or more variables:
   - To remove a specific variable, select it and click **Remove** > **Remove Selected**.
   - To remove all unused variables, click **Remove** > **Remove All Unused**.
   - To force the removal of all unused variables, including those in the project's undo/redo history, click **Remove** > **Force Remove All Unused**.

Editing a Local Variable

To edit a Local Variable:

1. Access **Local Variables** as described above.
2. Select the variable you want to edit and click **Edit**.
3. Change the properties as desired and click **OK**.

Overriding a Local Variable

You can override a variable value from an analysis setup's **General** tab.

Select the **Override** check box next to the value you wish to override, and enter a new value in the **Value** field.

Design Variable	Override	Value	Units
padrad		0.6	mm
viarad		0.2	mm

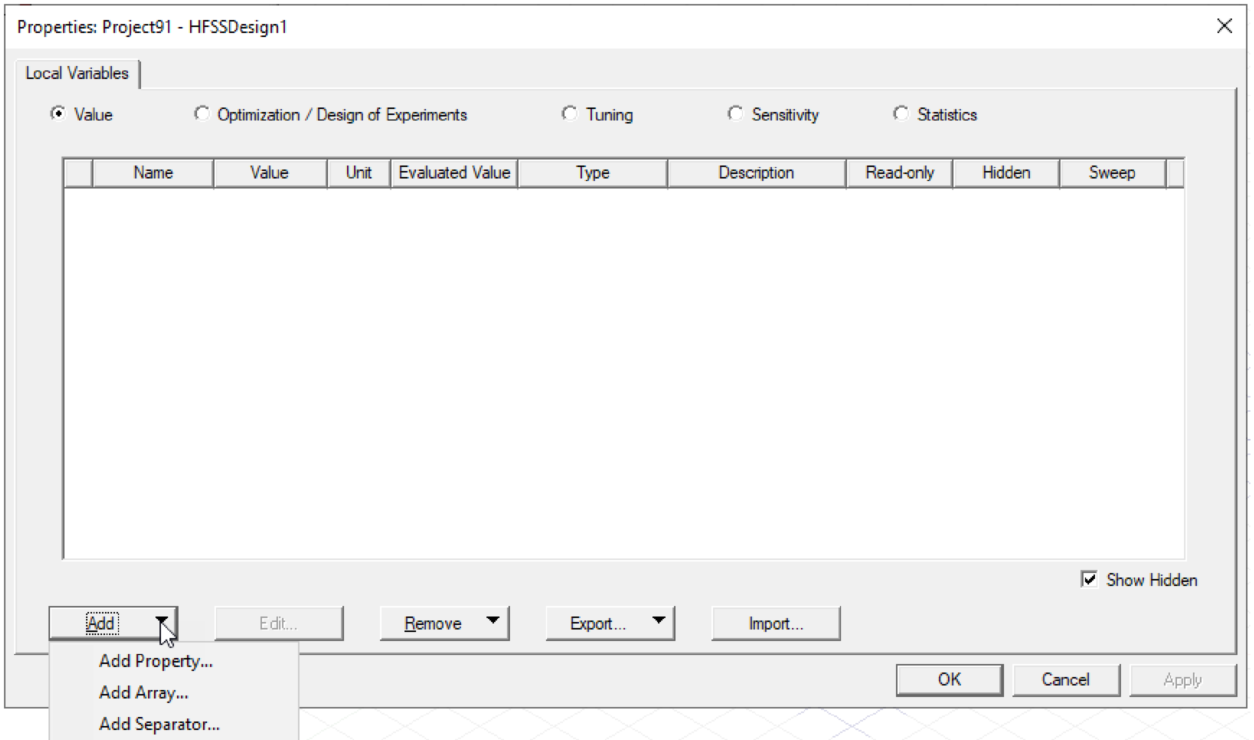
Defining Parameter Defaults at the Design Level

Adding a Parameter Default

To define a new parameter default at the design level:

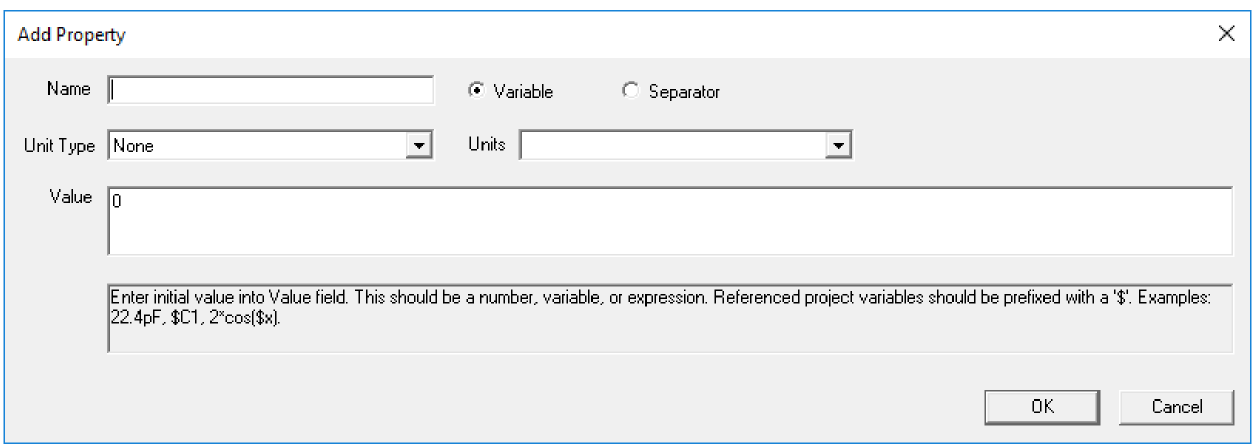
1. Access **Design Properties** one of three ways:
    - Click **EMIT** > **Design Properties**.
    - Right-click the design name in the Project Tree and select **Design Properties**.
    - From a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Design Variables.**

   The **Properties** window for design variables appears.



2. Click **Add...**.

   The **Add Property** window appears.



3. Enter information for the parameter default, as applicable:
   - **Name** – the name can contain alphanumeric characters and underscores (_). You cannot use any names already defined on the **General** tab.
   - **Variable Type** – select **Variable** to enter a value, or select **Separator** to provide a bolded name for a blank line.

Your selection impacts which properties you can edit:

Variable Type	Editable Properties
Variable	Unit Type, Units, Value.
Separator	Name. A separator variable provides a bolded name for a blank line to facilitate grouping variables in variable lists.

- **Unit Type** – for Variables, use the drop-down menu to select a type from the list (for example, Charge, Density, Energy, ...).
- **Units** – for Variables, use the drop-down menu to select a unit of measure.
- **Value** – for Variables, enter a valid numeric quantity. Alternately, you can assign a variable to a parameter by typing a variable name or mathematical expression in place of a parameter value.

4. Click **OK**.

   The new variable appears in the list. You can sort local variables by clicking the Name column header. Clicking once sorts them in ascending order, noted by a triangle pointing up. Clicking against sorts in descending order, noted by a triangle pointing down. Clicking a third time sorts in original order, with no triangle.

5. If desired, use the check boxes to designate a variable as **Read-only**, **Hidden**, or **Sweep**.
   - **Read-only** – when selected, the variable's name, value, unit, and description cannot be modified.
   - **Hidden** – hidden variables do not appear in the **Properties** window unless **Show Hidden** is selected.
6. Click **Apply** to apply changes.
7. Click **OK** to exit the window.

Deleting a Parameter Default

To remove a Parameter Default:

1. Access **Parameter Defaults** as described above.
2. Select a parameter from the list and click **Remove**.

Editing a Parameter Default

To edit a Parameter Default:

1. Access **Parameter Defaults** as described above.
2. Select the parameter you want to edit and click **Edit**.
3. Change the properties as desired and click **OK**.

Overriding a Parameter Default

You can override a parameter default from an analysis setup's **General** tab.

Select the **Override** check box next to the value you wish to override, and enter a new value in the **Value** field.

Design Variable	Override	Value	Units
padrad	☐	0.6	mm
viarad	☐	0.2	mm

Adding a Design Variable Array

You can define array variables that contain numbers or strings. Number array variables can be used in component property expressions, while string array variables can be used in certain component property values. Design variable arrays are only available within the EMIT design for which they are defined.

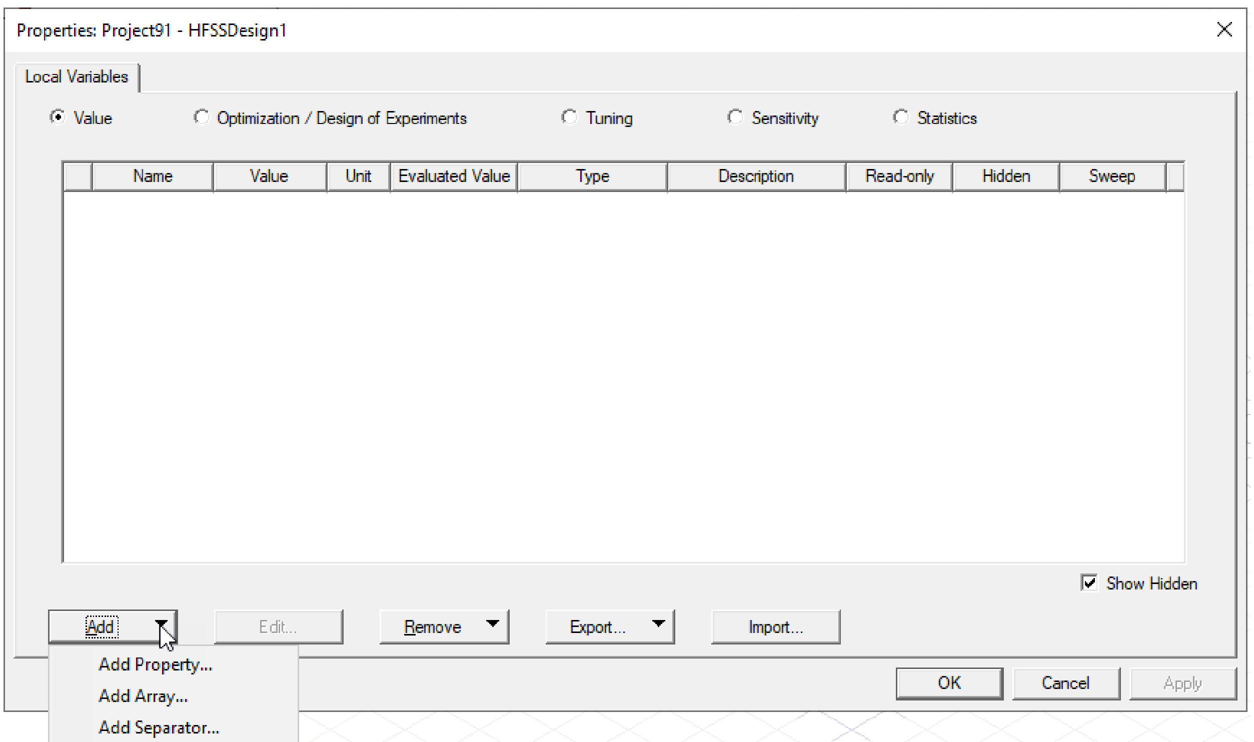
> **Note:**
>
> Text array variables can be used in certain component property values, but not all, and cannot be combined with operators to form more complex expressions.
>
> The index for a text array variable reference can either be a constant (1) or can be an index (ii) or even an expression (ii + 1). This allows you to sweep the index and simulate for different values that are stored in the array variable itself. In particular, it also enables you to sweep different text strings. This allows you to set a property to different string values as the index is swept. The following are properties that currently allow text array variables:
>
> - V_PRBS
> - V_PRBS_JITTER
> - V_PRBSD
> - V_PRBSG_JITTER
> - V_PSK
> - V_QAM
> - V_CPM

To define an array variable, you must first define a Local Variable Array:

1. Access **Local Variables** one of three ways:
   - Click **EMIT** > **Design Properties**.
   - Right-click the design name in the Project Tree and select **Design Properties**.
   - From a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Design Variables**

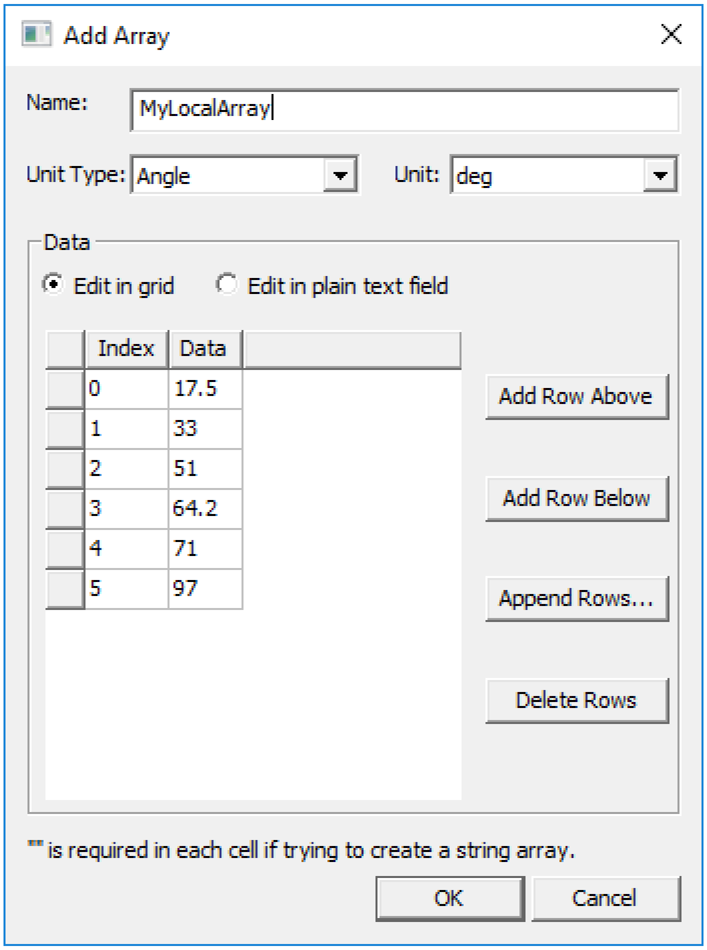
   The **Properties** window appears.
2. Select the **Local Variables** tab.

   The **Local Variables** tab displays.

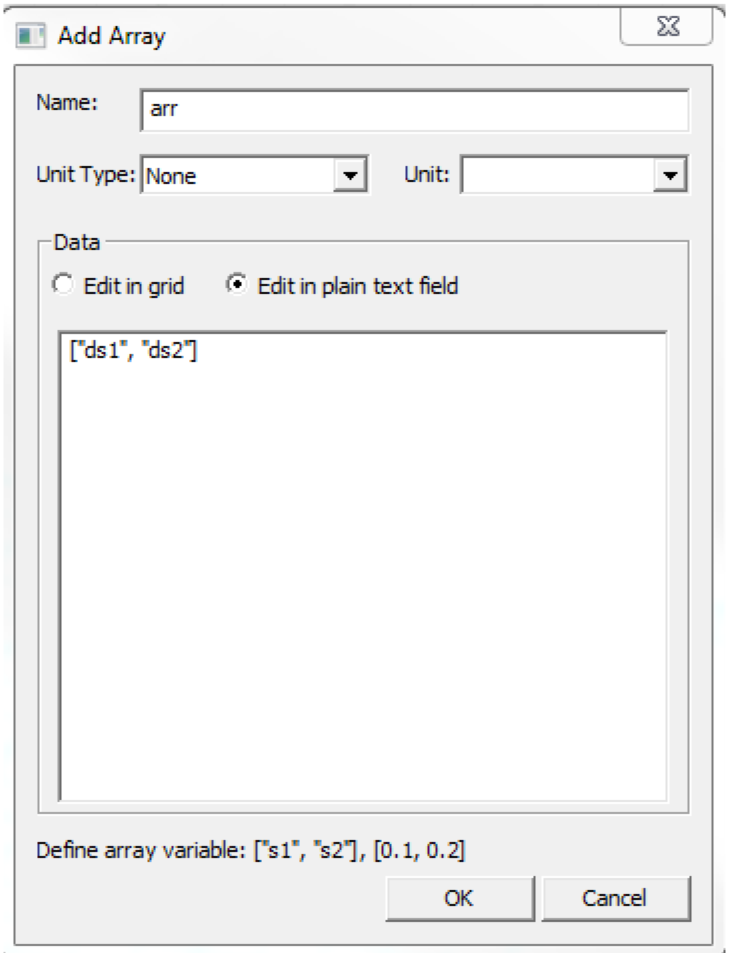
3. Click **Add Array**.

   The **Add Array** window appears.



The value grid displays each array item's **Index** number, and the **Data** associated with that index number.

If you elected to **Edit in plain text field** in the Add Array dialog box, the bracketed and comma delimited format is used.



4. Give the variable array a **Name**, and select a **Unit Type** and **Unit** from the drop-down menus.
5. Use the control buttons to add, delete, and reposition rows in the value grid at left. The default is **Edit in grid**, but you may select **Edit in plain text field**.

> **Note:**
>
> Quotation marks ("") are *required* as delimiters when array values are entered in either the grid or text field.

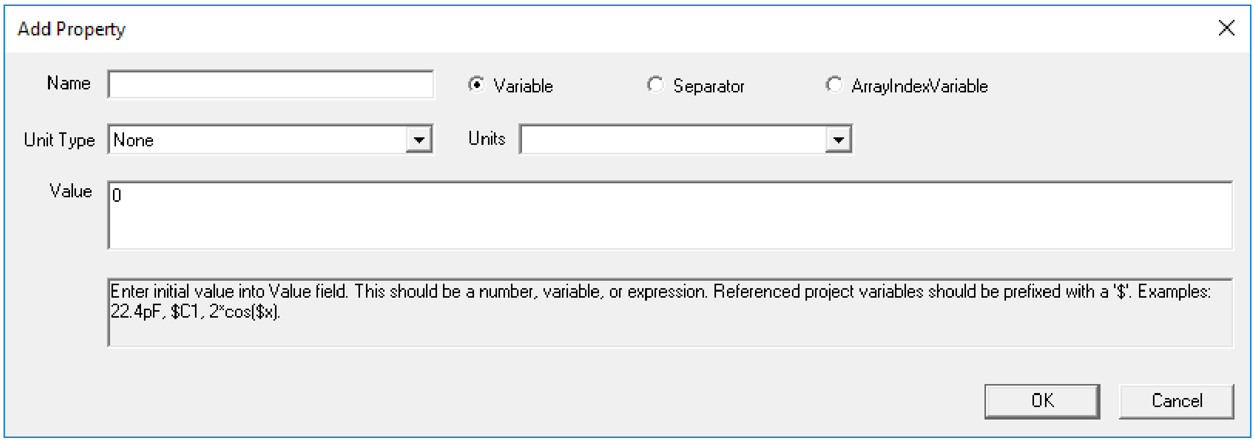
6. When you have finished entering values, click **OK**.

   The array appears in the **Local Variables** list.

You can add a variable to the array by adding an ArrayIndexVariable on the **Local Variables tab**.

1. Access **Local Variables** one of three ways:
   - Click **EMIT** > **Design Properties**.
   - Right-click the design name in the Project Tree and select **Design Properties**.
   - From the **Setup Optimization** dialog box, click **Edit Variables** > **Edit Design Variables**.
2. From the **Local Variables** tab, click **Add**.

   The **Add Property** dialog box appears.



3. Enter information for the variable, as applicable:
   - **Name** – the name can contain alphanumeric characters and underscores (_). You cannot use the names of Intrinsic Variables or pre-defined Constants.
   - **Variable Type** – select ArrayIndexVariable.
   - **Value** – enter a number, variable, or mathematical expression. The quantity entered will be the current (or default) value for the variable. If the mathematical expression includes a reference to an existing variable, this variable is treated as a dependent variable. The units for a dependent variable will automatically change to those of the independent variable on which the value depends. Additionally, dependent variables, though useful in many situations, cannot be the direct subject of optimization, sensitivity analysis, tuning, or statistical analysis.

     For ArrayIndexVariables, the index reference can be a constant (for example, 1), an index (for example, ii) or an expression (for example, ii + 1). This allows you to sweep the index and simulate for different values that are stored in the array variable itself. In particular, it also enables you to sweep different text strings. This allows you to set a property to different string values as the index is swept.

> **Warning:**
>
> If you include the variable's units in its definition (in the **Value** text box), do not include the variable's units when you enter the variable name for a parameter value.

- **Associate Array Variable** – for ArrayIndexVariables, select an associated array from the drop-down list. You must have previously created an array.

4. Click **OK**.

Converting Variables and Parameter Defaults

Design (Local) variables can be converted to parameter defaults, and vice versa.

To convert a parameter default or local variable:

1. Access **Design Properties** one of three ways:
   - Click **EMIT** > **Design Properties**.
   - Right-click the design name in the Project Tree and select **Design Properties**.
   - From the **Setup Optimization** dialog box, click **Edit Variables** > **Edit Design Variables**.

   The **Properties** window appears.
2. Select either the **Parameter Defaults** tab or the **Local Variables** tab, depending on which item you want to convert.
3. Select the variable/parameter and click **Convert to Parameter** or **Convert to Variable**.

   The item will move to the list on the appropriate tab.

Some special cases to consider:

- If a local variable references another local variable, it cannot be converted to a parameter default. You will see a pop-up message if you attempt to convert. This is because local variables can reference parameter defaults, but parameter defaults cannot reference local variables.
- Parameter defaults can be overridden and can have different values in each subcircuit, while a Local Variable cannot. If a parameter default has been overridden and you convert it to a variable, it will lose all overrides. Attempting to do so will display a pop-up message asking if you really want to do the conversion and lose the override values.

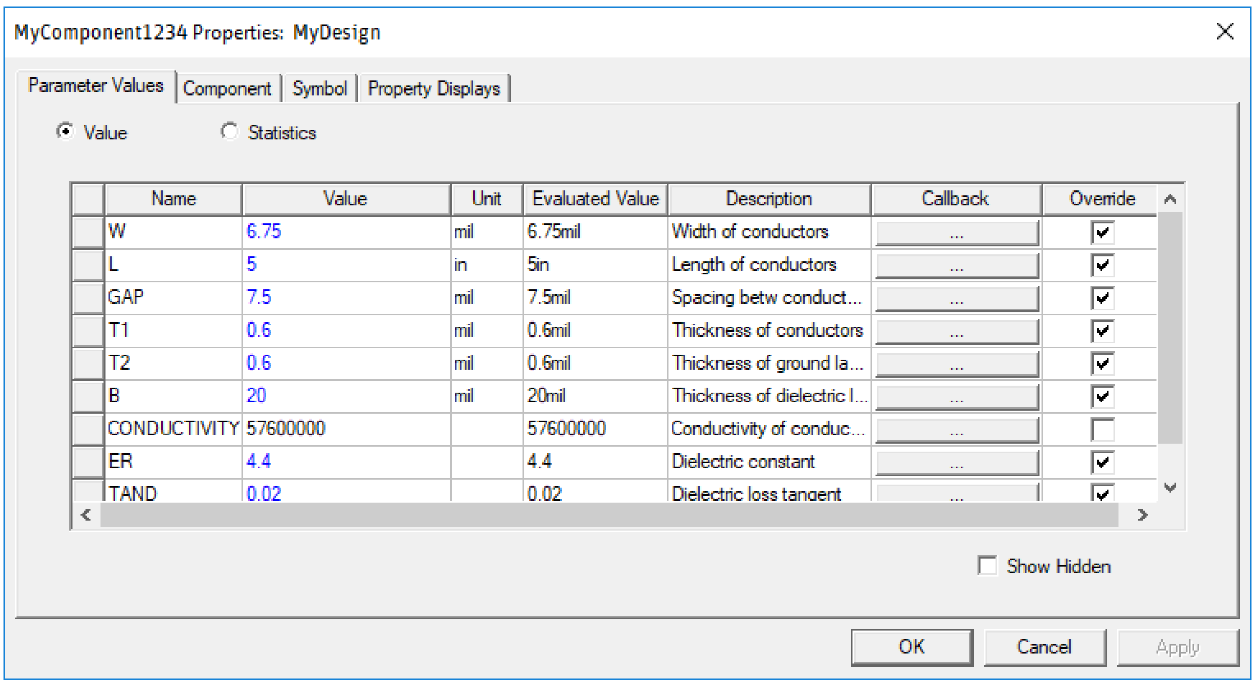
Adding Variables and Parameter Defaults from Components

To define a new variable or parameter default from a component:

1. Display a component's **Properties** window one of three ways:
   - Double-click the component.
   - Select the component and click **Edit** > **Properties**.
   - Select the component, right-click and select **Properties**.

   The **Properties** window appears.
2. Select the **Parameter Values** tab.

   The **Parameter Values** tab displays.



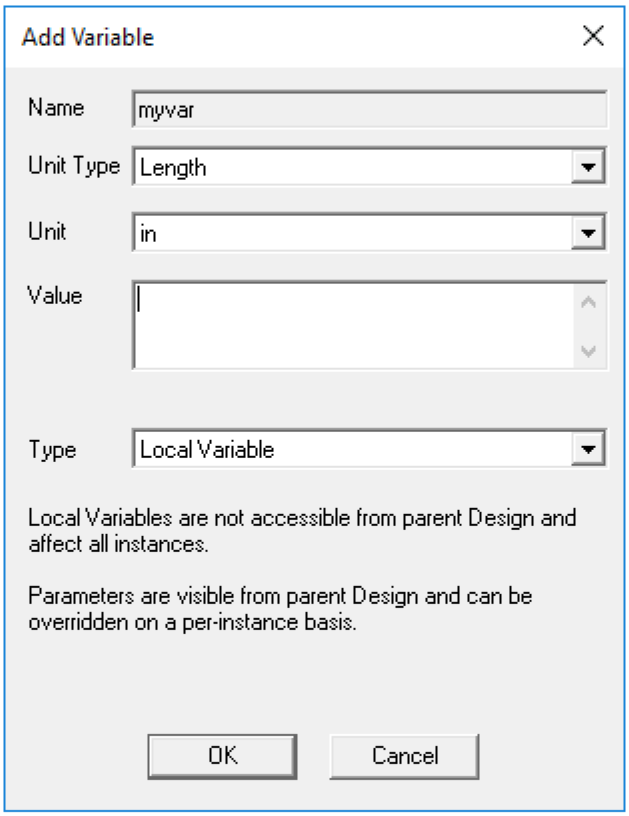
3. Select the **Value** radio button.
4. Select the **Value** field for the parameter you want to set equal to a local variable.
5. Type the variable name. The name may not be the same as an intrinsic variable or constant.

   Variable names may include alphanumeric characters and underscores (_).

> **Important:**
>
> If you wish to define a Project Variable, you must begin the name with a dollar sign ($).
>
> When you enter a name beginning with $, you will only be allowed to define it as a Project Variable.

6. Press **Enter**.

   The **Add Variable** window appears.



7. Select a **Unit Type** and **Unit** using the drop-down menus.
8. Enter a **Value**.

   The Value can be a number, variable, or mathematical expression. The quantity entered will be the current (or default) value for the variable. If the mathematical expression includes a reference to an existing variable, this variable is treated as a dependent variable. The units for a dependent variable will automatically change to those of the independent variable on which the value depends. Additionally, dependent variables, though useful in many situations, cannot be the direct subject of optimization, sensitivity analysis, tuning, or statistical analysis.

> **Warning:**
>
> If you include the variable's units in its definition in the **Value** text box, do not include the variable's units when you enter the variable name for a parameter value.

9. From the **Type** drop-down menu, select **Local Variable** or **Parameter Default**.

> **Note:**
>
> If you wish to define a Project Variable, your variable name must begin with a dollar sign ($).

10. Click **OK** to exit the **Add Variable** window.
11. Click **Apply** to save your changes.
12. Click **OK** to exit the **Properties** window.

Fixed vs. Non-Fixed Variables

By default, both Project variables and Design variables are included in the list of variables that index solution data. As a result, they are visible when selecting data to be displayed in plots.

However, if you define a variable as *fixed*, it is not swept and will not index solution data. Consequently, defining variables as fixed can speed up simulation—particularly if there are many variables to simulate.

To define a variable as fixed:

1. Open the **Properties** window for either Project or Design variables.
2. Clear the **Sweep** check box next to the variable you want to be fixed.

	Name	Value	Unit	Evaluated Value	Type	Description	Read-only	Hidden	Sweep
	wg_width	7.1	mm	7.1mm	Design		☐	☐	☑
	fillet_radius	0	mm	0mm	Design		☐	☐	☐

- If a **Sweep** check box is selected (default) the corresponding variable is NOT fixed.
- If a **Sweep** check box is cleared, the variable is fixed.

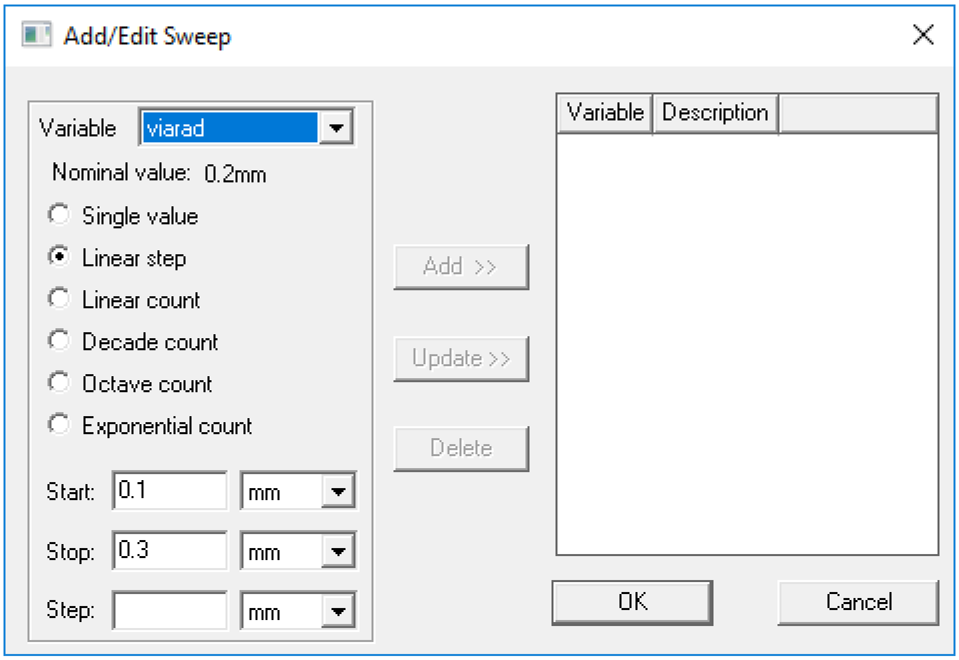
Non-Fixed Variable Sweeps

When setting up a sweep, the **Setup** window contains a list of variables being swept.

To add a non-fixed variable sweep:

1. Click **Add**.

   The **Add/Edit Sweep** window appears.



2. Use the **Variable** drop-down menu to select a non-fixed variable.
3. Select the desired settings and click **Add**.
4. Click **OK** to close the **Add/Edit Sweep** window.
5. Click **OK** to exit the **Setup**.

Orphaned Sweeps

If you create a simulation setup that contains a sweep of a variable, and then subsequently clear the variable's **Sweep** check box, the variable becomes fixed and, as a result, its sweep is "orphaned." The sweep is removed from the setup and the simulation runs as if the sweep did not exist.

- When you close the **Properties** window, a warning message is added to the **Message Window** for each orphaned sweep.

- If you edit an orphaned variable in the **Properties** window and reselect its **Sweep** box, the orphaned sweep will be restored to the setup list.
- If you double-click a setup that contains orphaned sweeps, a pop-up dialog box asks if you want to delete the orphaned sweeps. If you respond **Yes**, all orphaned sweeps are deleted when the **Properties** window is closed. Even if you reselect a **Sweep** check box, its orphaned sweep will NOT be restored. This action is undoable.
- Orphaned sweeps are not written to disk when the project is saved, so once you save a project and close it, any orphaned sweeps are permanently lost.

Choosing a Variable to Optimize

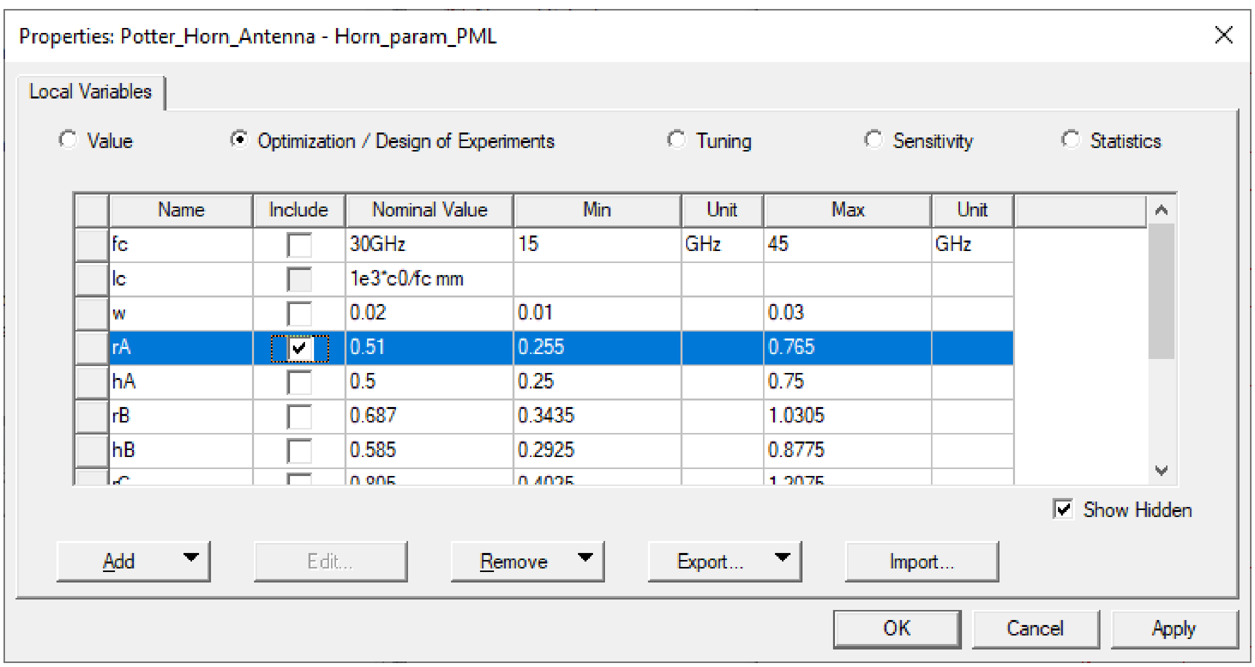
Before a variable can be optimized, you must specify that you intend for it to be used during an optimization analysis in the **Properties** window.

To specify that a variable is to be used during optimization:

1. Navigate either to the Project Variables **Properties** window (**Project** > **Project Variables**), or to the Design Variables **Properties** window (**EMIT** > **Design Properties**).

   The **Properties** window appears.

2. Use the tabs to navigate to the variable you want to optimize.
3. Select the **Optimization / Design of Experiments** radio button.
4. Use the **Include** check boxes to select variables for optimization.



> **Important:**
>
> Dependent variables cannot be optimized.
>
> Variables containing complex numbers cannot be used in an Optimetrics sweep or for optimization, statistical, sensitivity, and tuning setups.

The selected variable(s) will be available for optimization in an Optimetrics setup defined in the current design or project.

5. If desired, use the **Min** and **Max** fields to override the default minimum and maximum values that Optimetrics will use for the variable in every optimization analysis. During optimization, the optimizer will not consider variable values that lie outside of this range.

Choosing a Variable to Tune

Before a variable can be tuned, you must specify that you intend for it to be tuned in the **Properties** window.

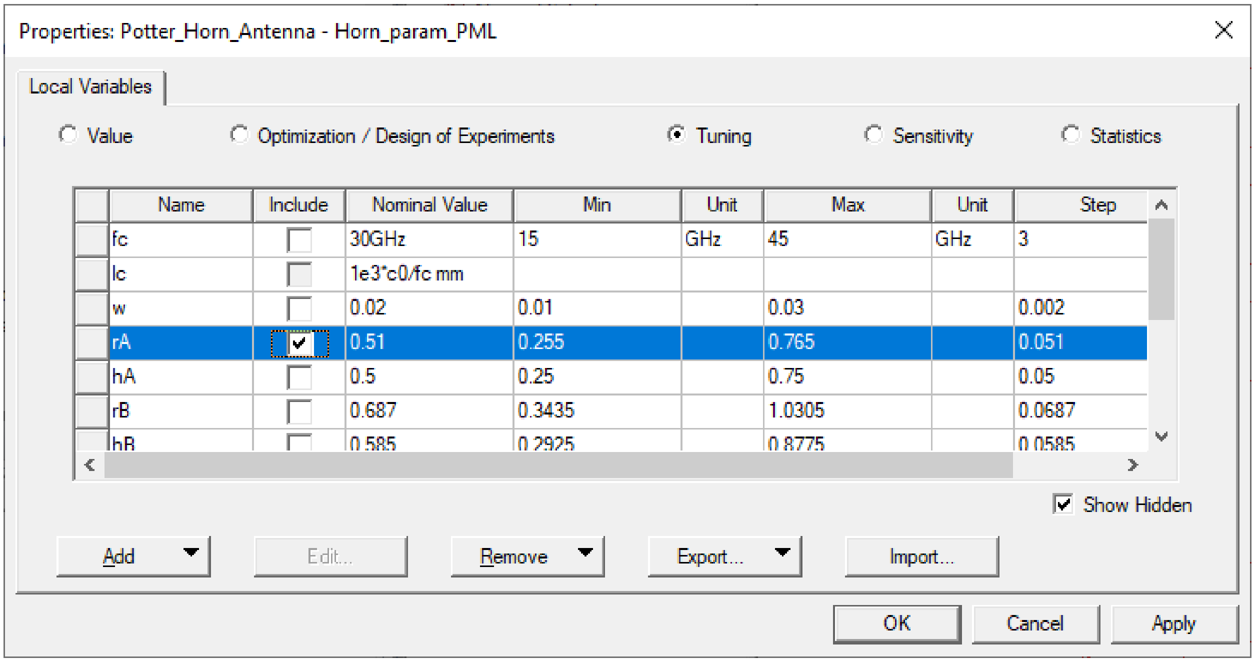
To specify that a variable is to be tuned:

1. Navigate either to the Project Variables **Properties** window (**Project** > **Project Variables**), or to the Design Variables **Properties** window (**EMIT** > **Design Properties**).

   The **Properties** window appears.

2. Use the tabs to navigate to the variable you want to optimize.
3. Select the **Tuning** radio button.
4. Use the **Include** check boxes to select variables for tuning.

> **Important:**
>
> Dependent variables cannot be tuned.
>
> Variables containing complex numbers cannot be used in an Optimetrics sweep or for optimization, statistical, sensitivity, and tuning setups.



The selected variable(s) will be available for tuning.

Including a Variable in a Sensitivity Analysis

Before a variable can be included in a sensitivity analysis, you must specify that you intend for it to be used during a sensitivity analysis in the **Properties** window.

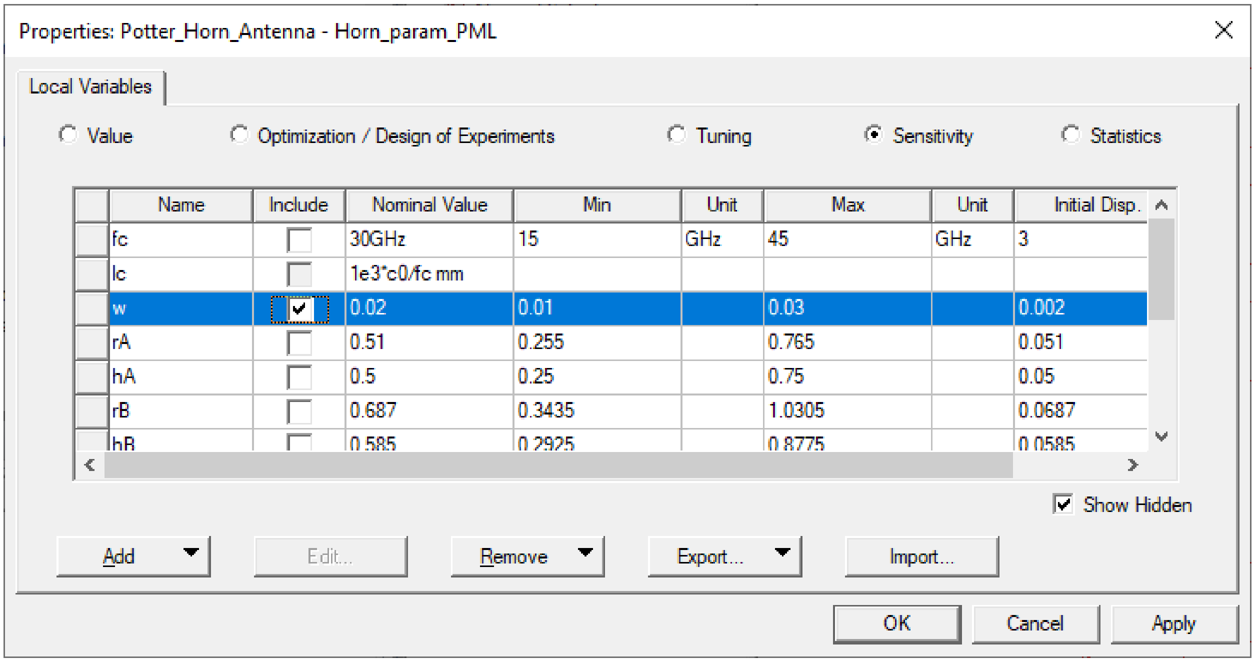
To specify that a variable is to be used during sensitivity analysis:

1. Navigate either to the Project Variables **Properties** window (**Project** > **Project Variables**), or to the Design Variables **Properties** window (**EMIT** > **Design Properties**).

   The **Properties** window appears.

2. Use the tabs to navigate to the variable you want to use.
3. Select the **Sensitivity** radio button.

4. Use the **Include** check boxes to select variables for sensitivity analysis.



Important:

Dependent variables cannot be included in a sensitivity analysis.

Variables containing complex numbers cannot be used in an Optimetrics sweep or for optimization, statistical, sensitivity, and tuning setups.

The selected variable(s) will be available for sensitivity analysis in the current design or project.

5. If desired, use the **Min** and **Max** fields to override the default minimum and maximum values that Optimetrics will use for the variable in every sensitivity analysis. During analysis, the optimizer will not consider variable values that lie outside of this range.
6. If desired, use the **Initial Disp.** field to override the default initial displacement value that Optimetrics will use for the variable in every sensitivity analysis. During analysis, Optimetrics will not consider a variable value for the first design variation that is greater than this step size away from the starting variable value.

Including a Variable in a Statistical Analysis

Before a variable can be included in a statistical analysis, you must specify that you intend for it to be used during a statistical analysis in the **Properties** window.

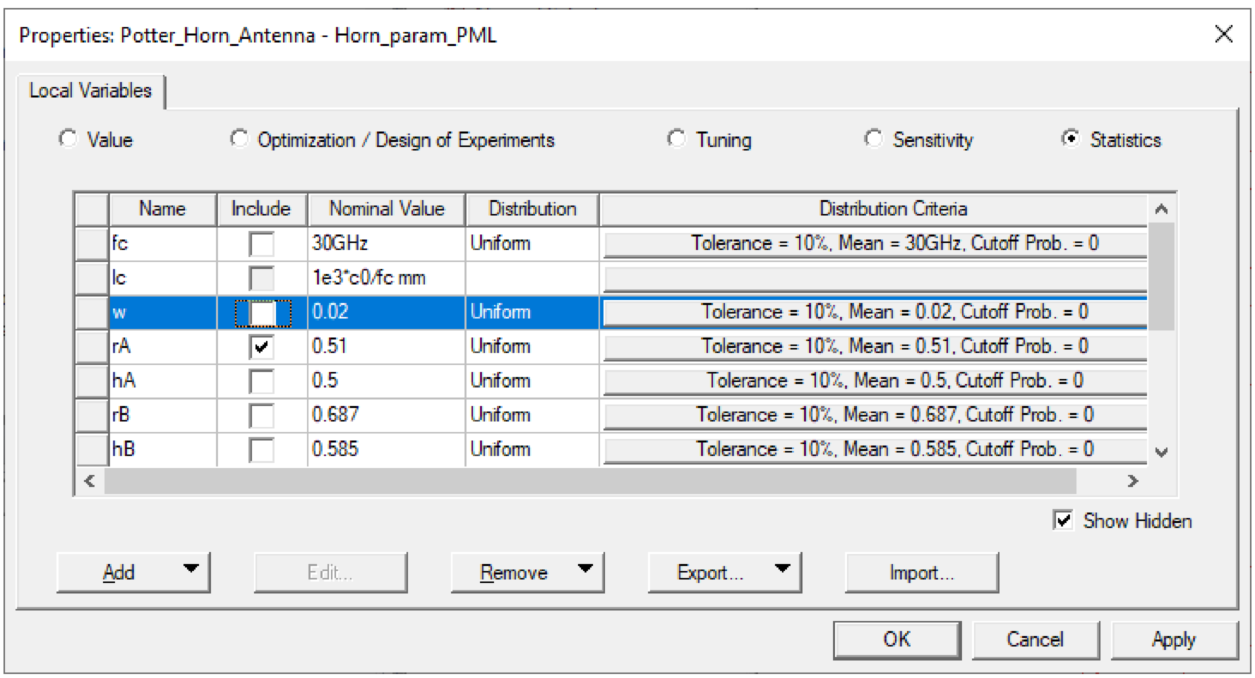
To specify that a variable is to be used during statistical analysis:

1. Navigate either to the Project Variables **Properties** window (**Project** > **Project Variables**), or to the Design Variables **Properties** window (**EMIT** > **Design Properties**).

   The **Properties** window appears.

2. Use the tabs to navigate to the variable you want to use.
3. Select the **Statistics** radio button.
4. Use the **Include** check boxes to select variables for statistical analysis.

   > **Important:**
   >
   > Dependent variables cannot be included in a statistical analysis.
   >
   > Variables containing complex numbers cannot be used in an Optimetrics sweep or for optimization, statistical, sensitivity, and tuning setups.

   The selected variable(s) will be available for statistical analysis in the current design or project.

5. If desired, click the **Distribution Criteria** button to override the distribution criteria that Optimetrics will use for the variable in every statistical analysis.

Copying and Pasting a Variables List

From either the Project or Design variables window, you can copy a list of variables and their values to the clipboard. You can then paste them in a plain text, tab-separated format. This is useful for creating documentation.

To copy and paste the variables list:

1. Navigate to the Project Variables or Design Variables **Properties** window.
2. Right-click and select **Copy to Clipboard**.
3. Open a text editor and **paste**.

The pasted variables appear in a tab separated column format. Fields that do not contain values are left blank.

Name $width	Value 14.8570192	Unit mm	"Evaluated Value 14.8570192mm	"Description	Read-only false	Hidden false
$length	7.824547736	mm	7.824547736mm		false	false
$height	0.45*$width		6.68565864mm		false	false

Working with Datasets

Datasets are collections of plotted data points that can be extrapolated into an equation based on the piecewise linear makeup of the plot. Each plot consists of straight line segments whose vertices represent their end points. A curve is fitted to the segments of the plot and an expression is derived from the curve that best fits the segmented plot. The created expression is then used in piecewise linear Intrinsic Functions. You can add datasets at either the project level or the design level. They can be for various purposes, including to define frequency-dependent port impedances or frequency-dependent global variables.

Project-level datasets are typically used for defining material properties at the project level (applicable to all designs in the project).

Design-level datasets can be used in geometry entities like part commands, coordinate systems, points, and planes. Design-level datasets do not work with equation-based surfaces or curves. Design-level datasets can be used directly with piecewise linear functions in expressions or indirectly through variables that can refer to the dataset.

Design-level datasets can also be used in the following operations:

- **Creating or editing geometry** – when a geometry uses a dataset directly, edit dataset invalidates the solution. When a geometry uses a variable that is defined by dataset, editing the dataset does not invalidate the solution.

- **Creating an Animation** – based on a variable which can index in datasets.
- **Copying and Pasting Geometry** – if a part refers to a design dataset, it will be pasted to destination design.

Datasets are provided for EMIT designs.

The **Datasets** dialog box provides a browsable listing of all datasets currently defined for the project or design. A preview window displays a plot of the currently selected dataset. Controls allow you to **Add**, **Edit**, **Remove**, and **Clone** datasets; to **Import** and **Export** characteristic data; and to launch the **SheetScan** tool to extract data from graphics such as data sheets.

To access the **Datasets** window:

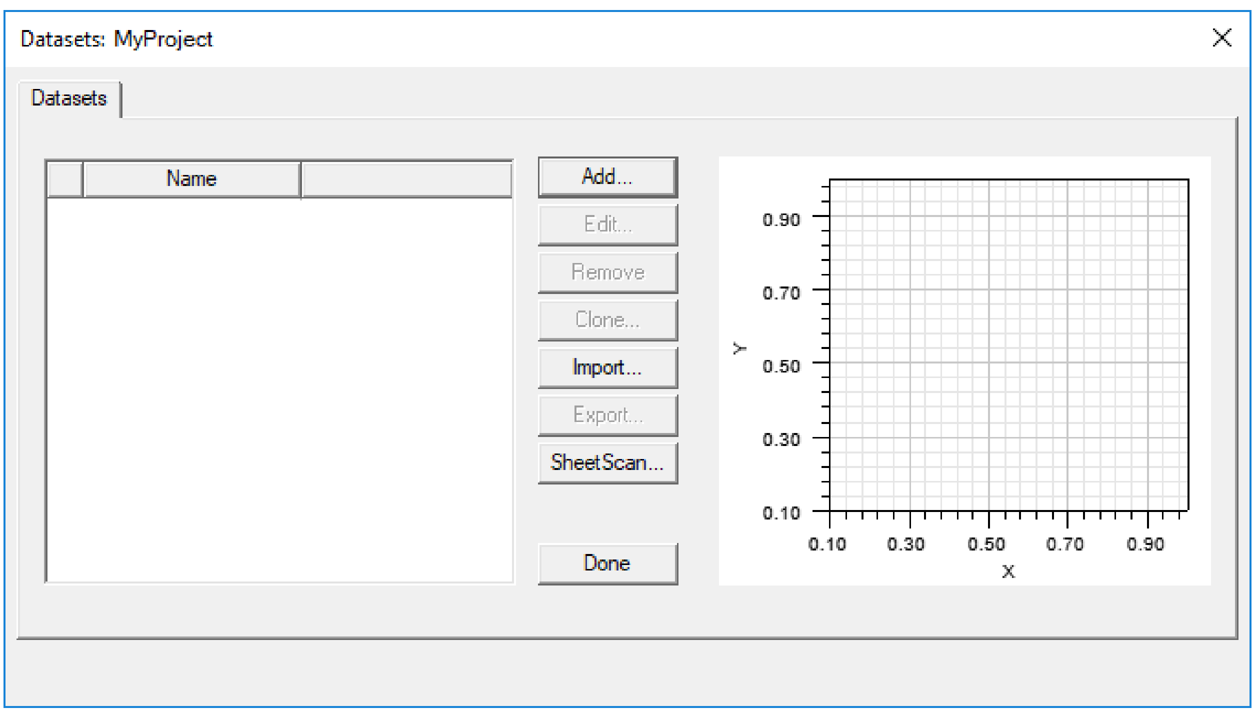
- For project-level datasets, including 3D datasets, click **Project** > **Datasets**.
- For design-level datasets, click **EMIT** > **Design Datasets**.

Importing Datasets

To import a dataset from a file:

1. Access the **Datasets** window:
    - For project-level datasets, click **Project** > **Datasets**.
    - For design-level datasets, click **EMIT** > **Design Datasets**.

    The **Datasets** window appears.



2. Click **Import**.
3. In the file browser that appears, use the drop-down menu to select the file format of the file you wish to import.

> **Note:**
>
> The table below lists the file types supported for *Project level datasets*. For *Design level datasets*, only **.tab** files are supported.

.mdx, .mda	Twin Builder Characteristic format
.xls, .xlsx	Microsoft Excel
.txt	text file
.csv	Comma-separated value
.out	Maxwell SPICE (read-only - reads data inside the KW_ DATA section)
.cfg	Comtrade (IEEE Std C37-111-1999)
.dat	TEK Oscilloscope
.tab	tab delimited data files

a. If you select an **.mdx**, **.mda**, **.out**, **.cfg**, or **.tab** file, the data is imported immediately into **Datasets** window.

b. If you select an **.xls** or **.xlsx** file containing multiple sheets, a **Table Properties** dialog box appears, where you can choose the desired sheet from a drop-down menu. Otherwise, selecting an .xls or .xlsx file imports the data immediately.

> **Note:**
>
> Microsoft Excel is required to import .xls or .xlsx files.
>
> Only the first two columns of data are imported, the left-most column containing the X-coordinate values. The x-coordinate values for successive data points must increase within ten significant digits. Non-numeric entries are assigned a value of zero.
>
> The first row of data is assumed to contain column headings and is ignored.

c. Selecting a **.txt** or **.csv** file opens an **Import** dialog box where you can specify settings for reading the data in the file for import. You can choose the separator(s) and decimal symbol, as well as the line at which to begin the import. The dialog box shows both the original text and the text as it would appear when imported based on the current import settings.

4. When satisfied with the import settings, click **OK** to import the data.

   Imported datasets take the name of the imported file. If a dataset of the same name already exists, a number will be appended to the end of the new dataset (for example, a duplicate 'dataset' becomes 'dataset1').

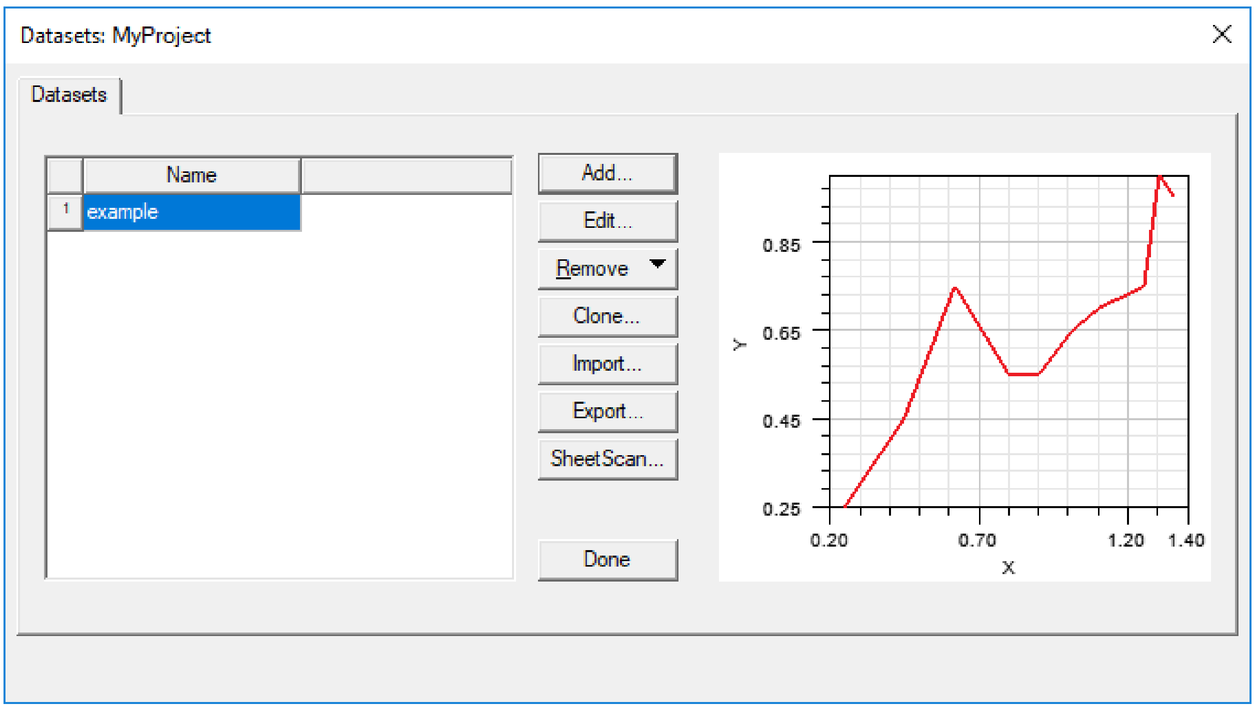
   Once the data has been imported, you can edit, remove, clone, or export the dataset. You can also change plot display properties.

Editing Datasets

To modify an existing dataset:

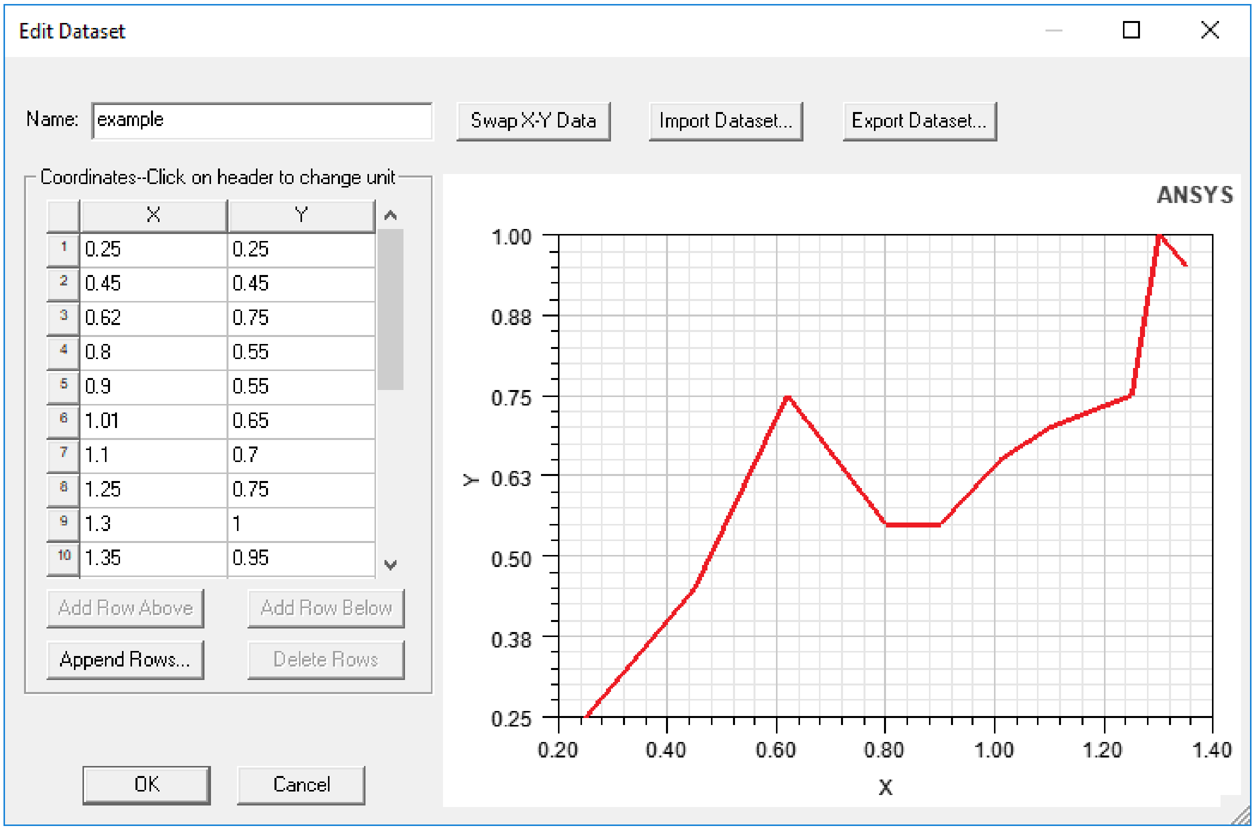
1. Access the **Datasets** window:
   - For project-level datasets, click **Project** > **Datasets**.
   - For design-level datasets, click **EMIT** > **Design Datasets**.

   The **Datasets** window appears, listing any project or design-level datasets.



2. Click **Edit**.

   The **Edit Dataset** window appears.



3. Change the **Name** and **X/Y** coordinates as desired.
4. When you are finished, click **OK** to exit the **Edit Dataset** window.
5. Click **Done** to exit the **Datasets** window.

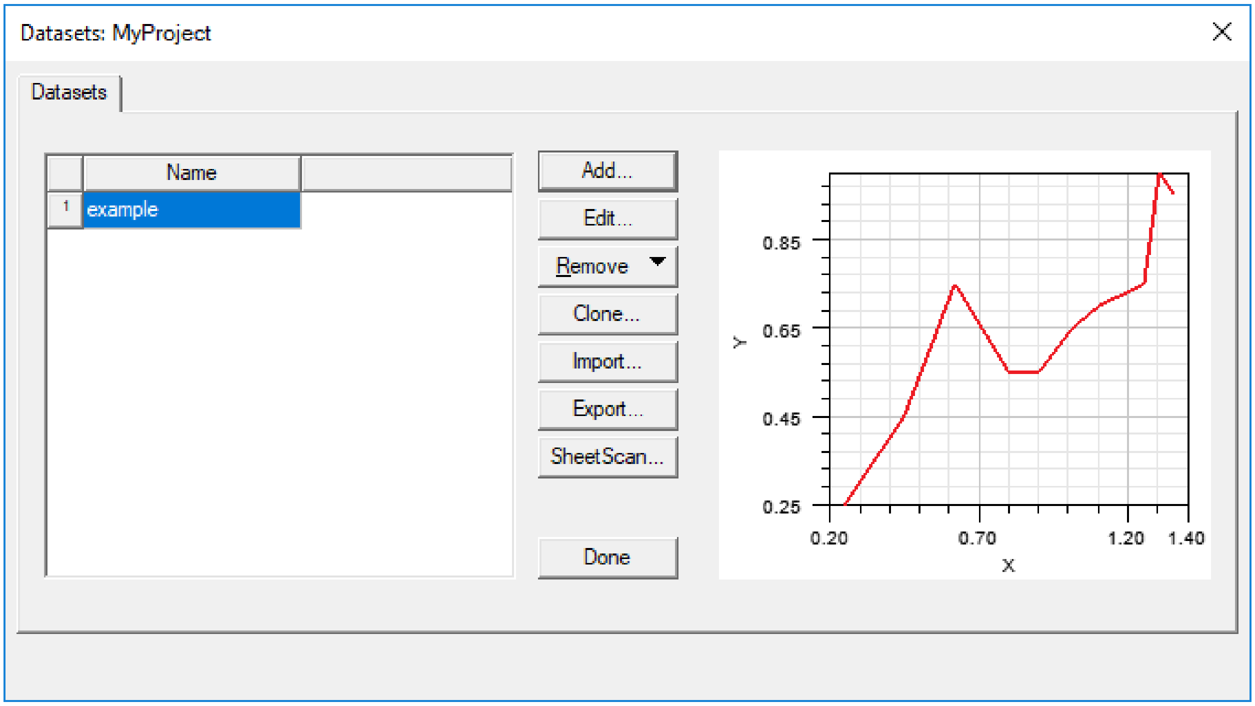
Cloning Datasets

Cloning a dataset creates an identical copy of an existing dataset. The clone can then be modified as necessary.

To clone a dataset:

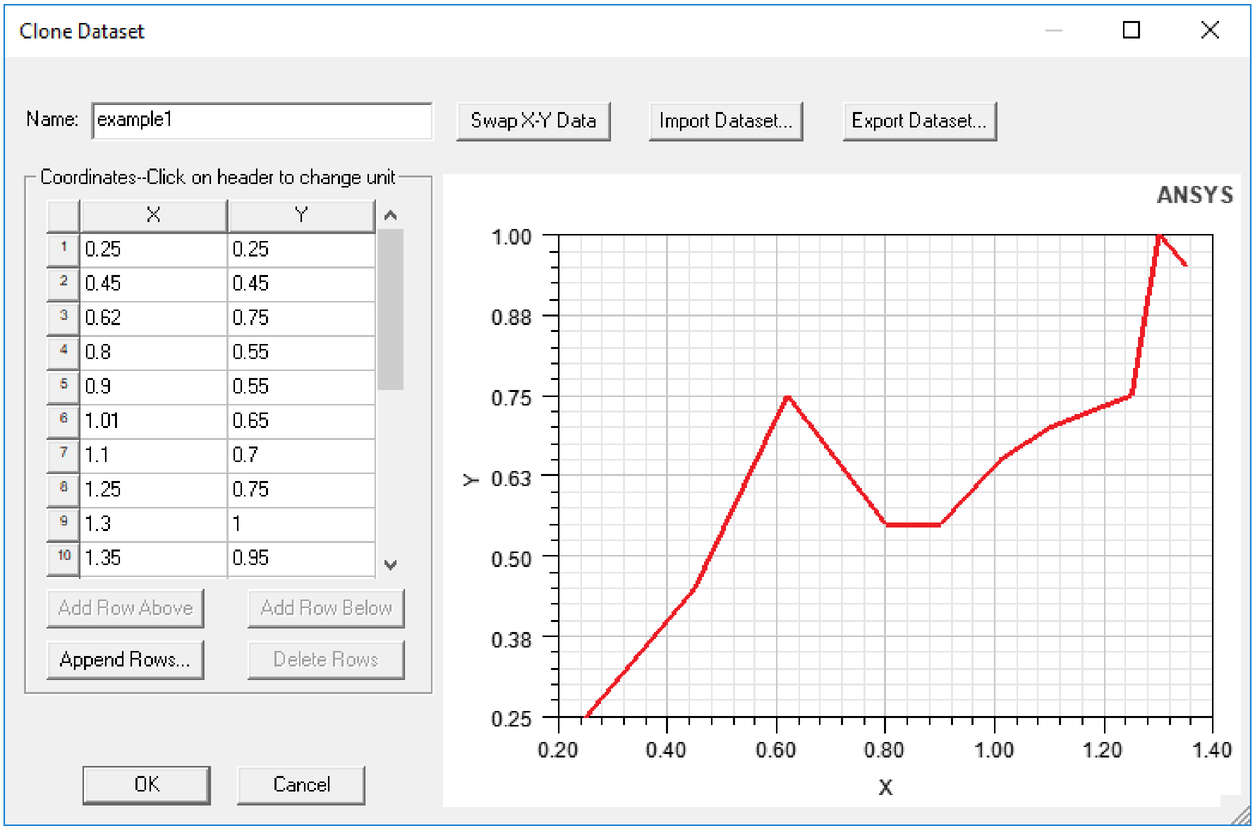
1. Access the **Datasets** window:
   - For project-level datasets, click **Project** > **Datasets**.
   - For design-level datasets, click **EMIT** > **Design Datasets**.

   The **Datasets** window appears, listing any project or design-level datasets.



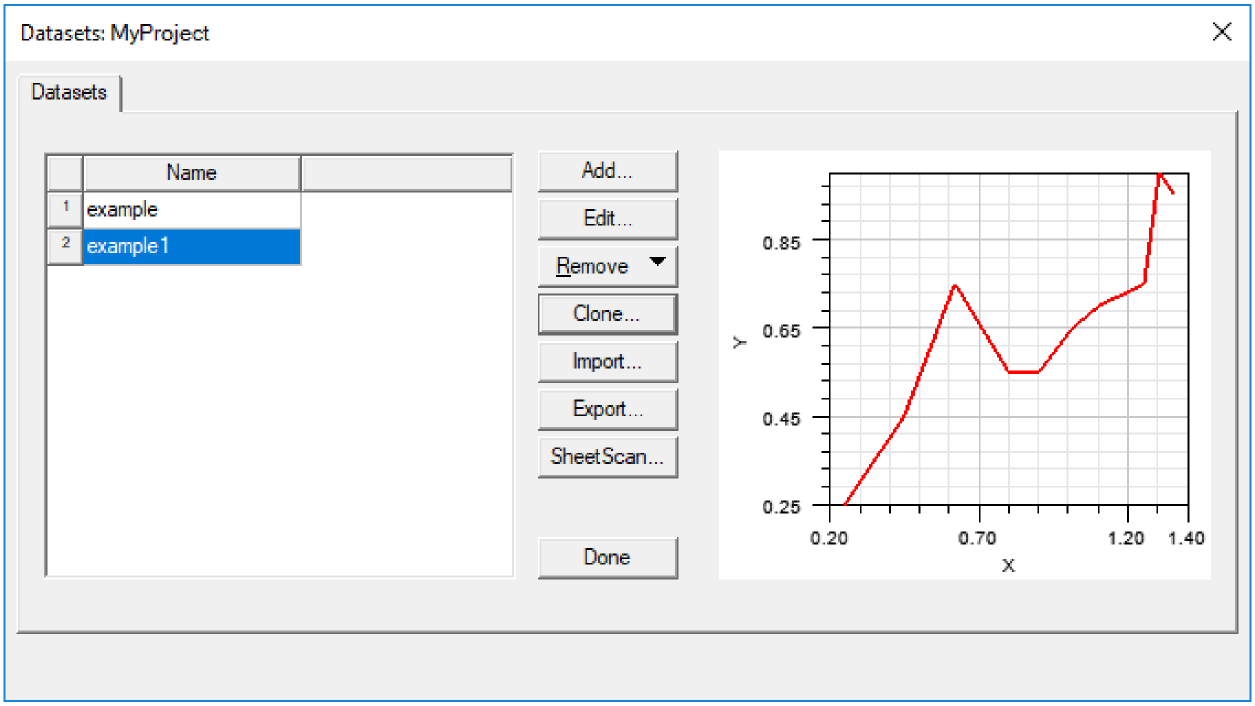
2. Click **Clone**.

   The **Clone Dataset** window appears. The default name is the original dataset name with a number appended to the end.



3. Change the **Name** and **X/Y** coordinates as desired.
4. When you are finished, click **OK** to exit the **Clone Dataset** window.

   The cloned dataset appears in the list.



5. Click **Done** to exit the **Datasets** window.

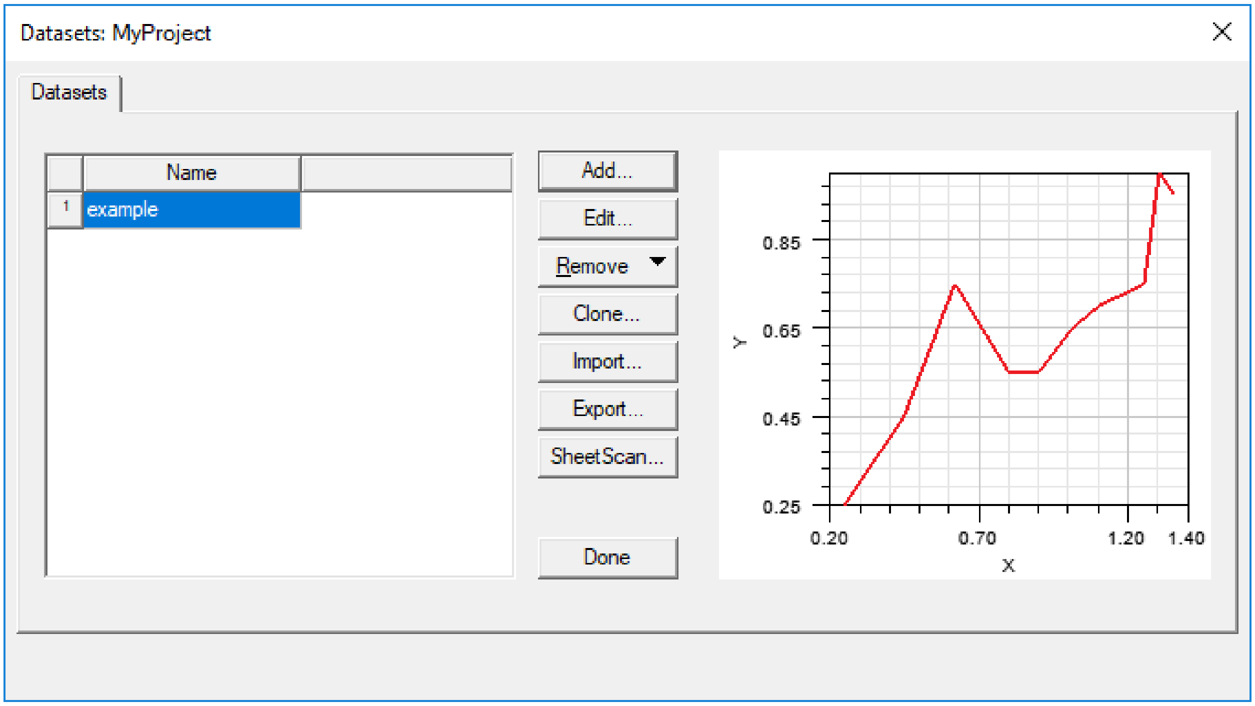
Exporting Datasets

You can export datasets in **.tab** format.

To export a dataset:

1. Access the **Datasets** window:
   - For project-level datasets, click **Project** > **Datasets**.
   - For design-level datasets, click **EMIT** > **Design Datasets**.

   The **Datasets** window appears, listing any existing datasets.



2. Select a dataset and click **Export**.
3. Browse to the location where you would like to save the *.tab file.
4. Enter a name for the file.
5. Click **Save**.

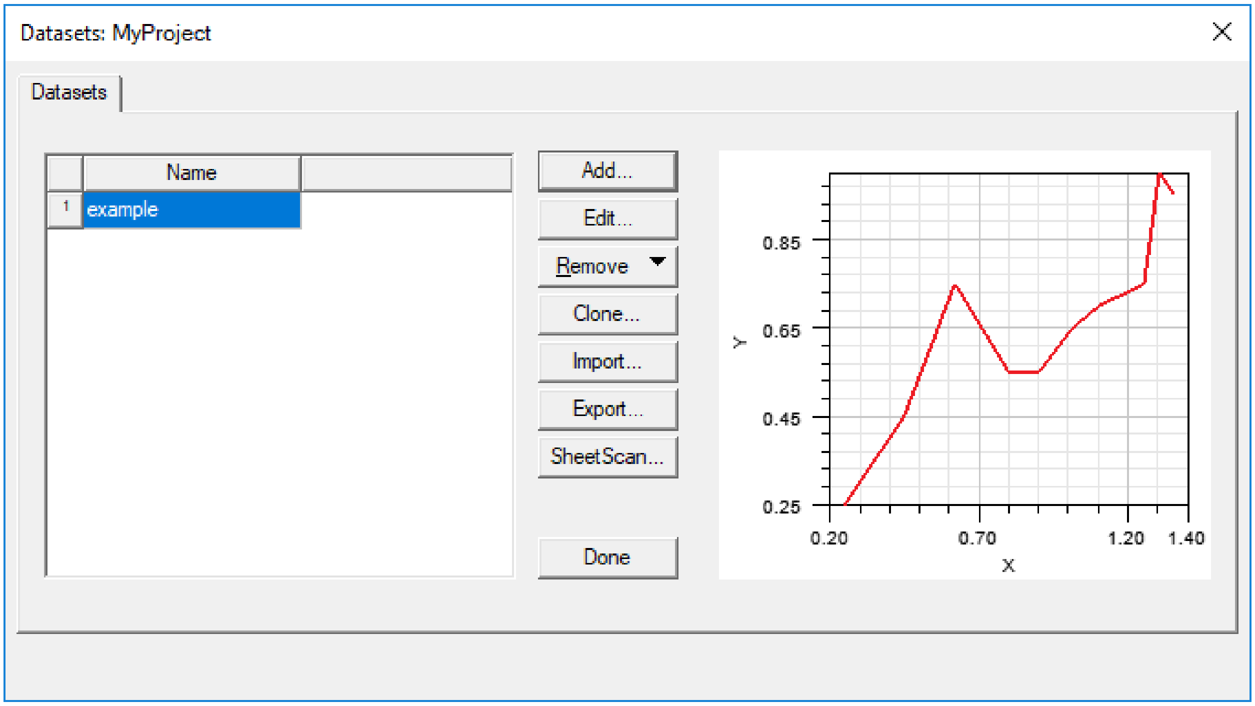
   The dataset file is saved to the specified location for later import.

Removing Datasets

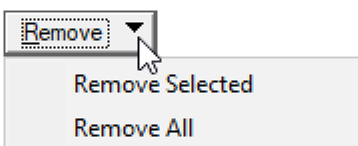
To remove a previously added dataset:

1. Access the **Datasets** window:
   - For project-level datasets, click **Project** > **Datasets**.
   - For design-level datasets, click **EMIT** > **Design Datasets**.

   The **Datasets** window appears, listing any existing datasets.



2. Select a dataset and click **Remove**.
3. Select either **Remove Selected** to remove the selected dataset or **Remove All** to remove all project or design-level datasets.



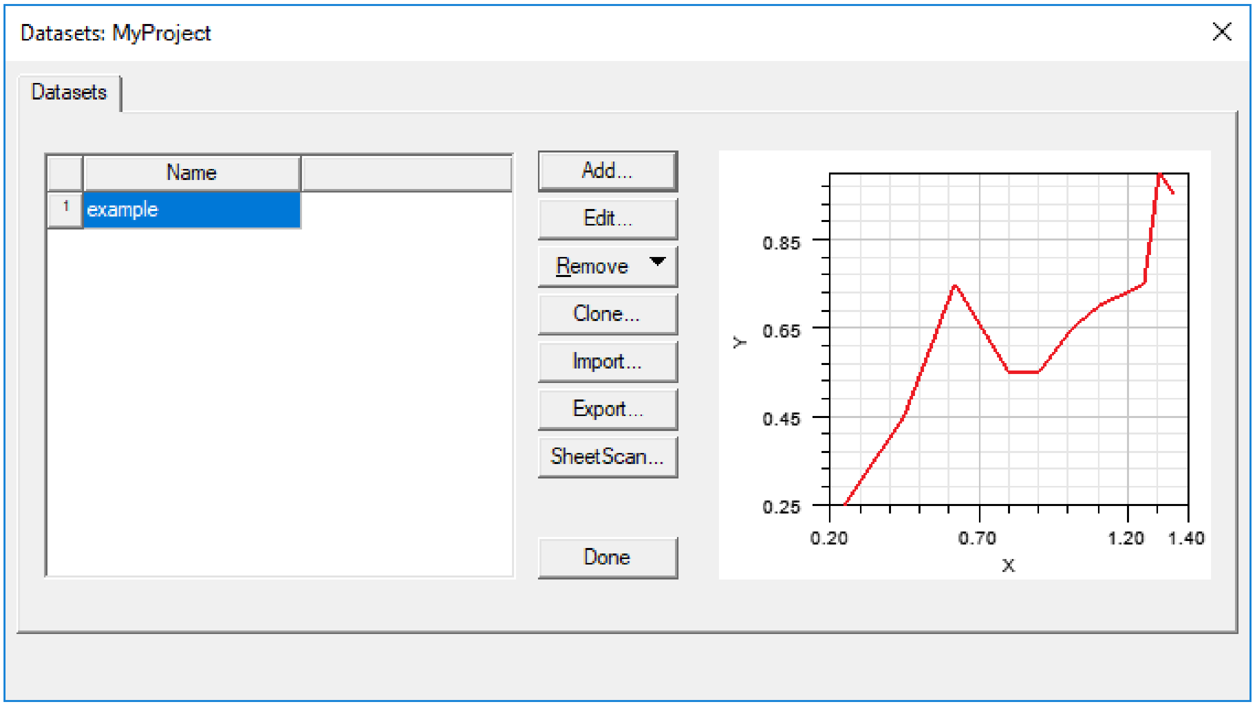
The selected dataset(s) are deleted from the list.

> **Note:**
>
> The **Remove All** option does not work across dataset types. That is, selecting **Remove All** from the **Project Datasets** window does not remove design-level datasets, and vice versa.

4. Click **Done** to exit the **Datasets** window.

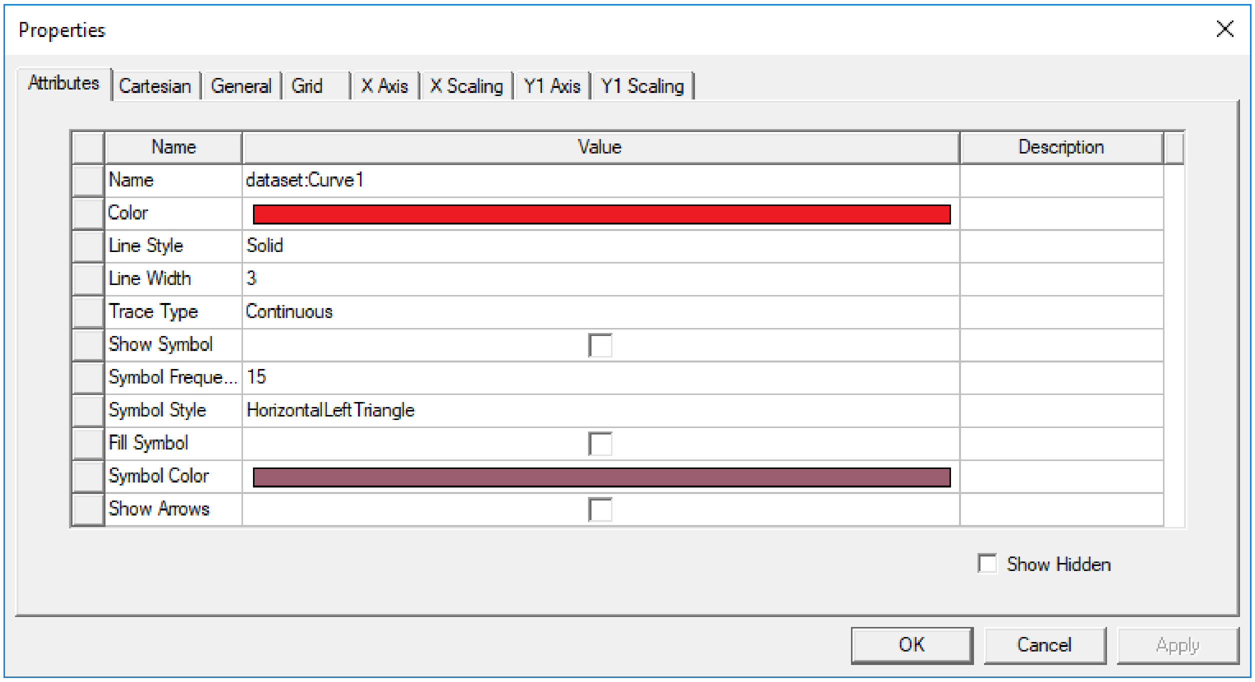
Changing Dataset Plot Properties

From the **Datasets** window, you can change the appearance of a dataset's plot.



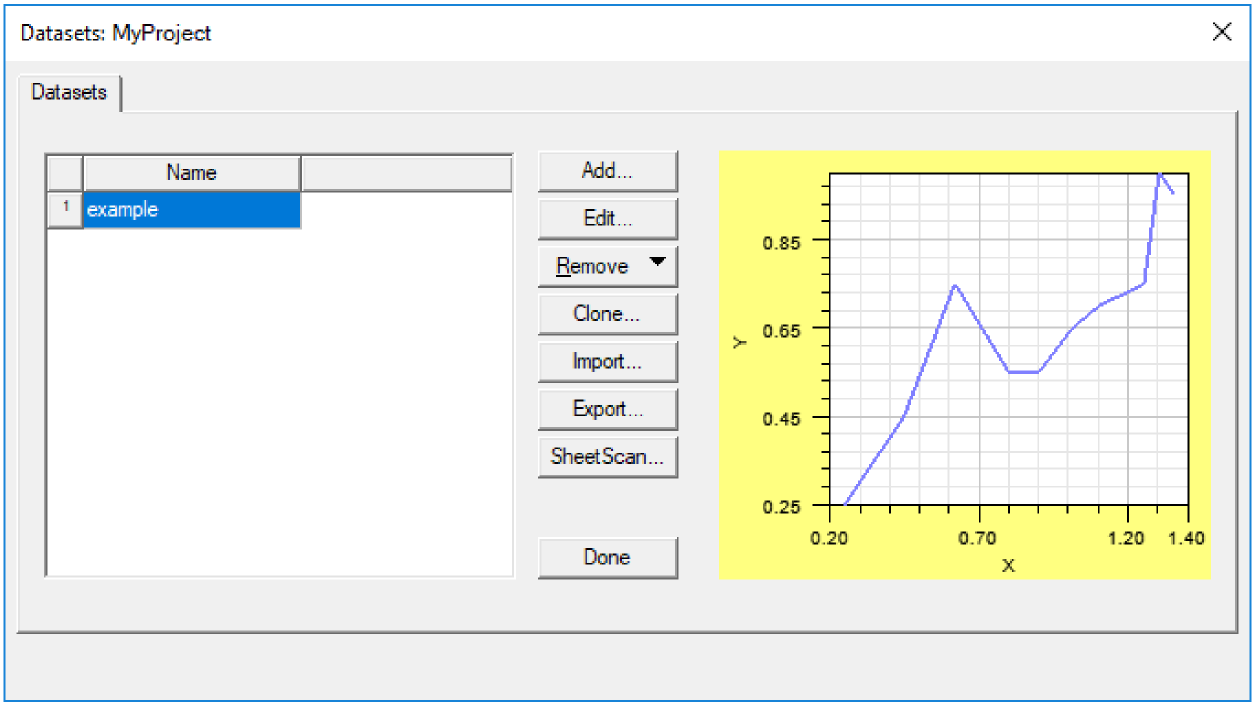
To change a plot's appearance:

1. Double-click a plot element (for example, a line, axis, or title) to open that element's **Properties** window.



2. Set the colors, fonts, and styles as applicable and as desired.
3. Click **Apply**.
4. Click **OK** to exit the **Properties** window.

   The plot now displays as specified.



Using SheetScan

SheetScan allows you to extract characteristics data from graphics such as data sheets which have been scanned and saved in any of the following formats: .bmp, .dib, .jpg, .gif, .tif, .tga, .pcx, .htm, or .html.

Access SheetScan from the **Datasets** window:

- For project-level datasets, click **Project** > **Datasets**.
- For design-level datasets, click **EMIT** > **Design Datasets**.

In addition to importing graphic files directly, SheetScan also can be used to browse the Internet for datasheet information and transfer a snapshot of the web page to the SheetScan editor where you can map axes on the image as an overlay. You can then manually add datapoints to approximate the characteristic curve(s) on the datasheet.The sampled data can then be converted to Ansys Electronics Desktop format, and the extracted data exported to an Ansys Electronics Desktop dataset or saved to a tab-delimited file.

The process for creating a dataset using SheetScan involves four basic operations:

- Loading a datasheet into SheetScan.
- Defining a coordinate system for the imported datasheet picture.

- Defining a characteristic curve using the datasheet picture as reference.
- Exporting the characteristic curve data to a file or to a dataset.

SheetScan Menus and Settings

In the SheetScan window, various menus and toolbars allow you to access SheetScan functionality.

The top menu bar contains the following submenus:

- **File** – allows you to open, close, save, and print files.
- **Edit** – provides access to cut, copy, and paste functions.
- **View** – allows you to select which toolbars appear in the SheetScan window (see below).
- **Options** – provides access to document settings.
- **Picture** – provides options for loading a picture or HTML file into SheetScan, and for deleting a picture.

> **Note:**
>
> The **Delete** action cannot be undone. If you delete a picture from the SheetScan editor, you must reload it from the source file or Internet web page.

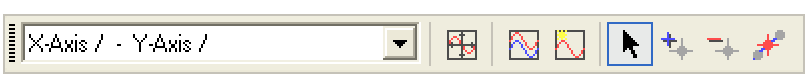
- **Coordinate System** – provides options for creating and editing a coordinate system.
- **Curve** – provides options that allow you to define a curve.
- **Window** – provides options for rearranging the currently open SheetScan windows (for example, overlapping and tiling).
- **Help** – accesses the Electronics Desktop online help.

SheetScan View Options

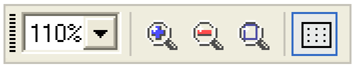
You can toggle the following toolbars via **View** > **Toolbar**:

- **Standard** – provides access to basic functions such as file Open and Save, Cut, Copy, Paste, Print, and Help.

- **Curve** – contains a drop-down menu allowing you to select a curve and tools for working with curve values.

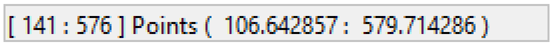


- **Zoom** – provides tools for scaling the current view.

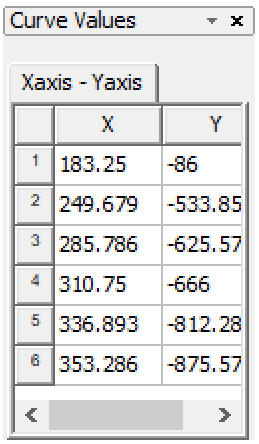


From the **View** menu, you can also toggle the following:

- **Status Bar** – displays, at the bottom of the screen, the current cursor coordinates.



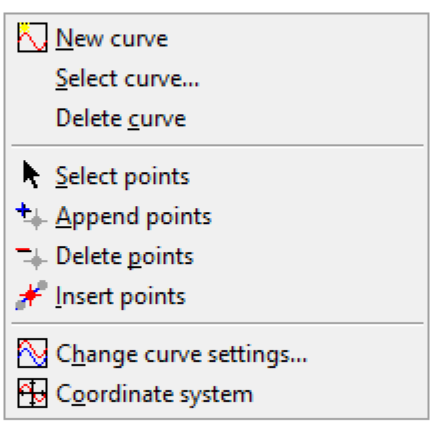
- **Curve Values** – dockable window that displays the data points used while creating a characteristic curve. Data for each curve on a sheet is displayed on its own tab. You can manually change the X and Y values in the table to fine-tune the characteristic curve.



- **Grid** – displays the coordinate grid you created over the sheet.

SheetScan Right-click Menu

Once you have loaded a sheet, you can right-click in the SheetScan window to access curve and coordinate system options:



SheetScan Settings

Default settings can be changed from **Options** > **Settings**.

The **Settings** dialog box contains three tabs:

- **Document** – allows you to set the width and height of the sheet created when a picture is imported. You can either enter the dimensions manually, or allow SheetScan to adapt the dimensions to the picture being loaded.
- **Axis** – allows you to set the name, units, scaling, and offset for the X and Y axis. You can also select **Monotonicity in X** to prevent adding consecutive data points whose X-values are not increasing.
- **Representation** – allows you to choose whether to connect points on the characteristic curve and to choose the connecting line color. You can also decide whether to display markers and choose marker colors.

> **Note:**
>
> You can also override the default settings on the **Axis** and **Representation** tabs for individual curves (See: Defining a Characteristic Curve in SheetScan).

Loading a Datasheet Picture into SheetScan

By default, SheetScan opens a new, blank datasheet editing window.

There are two ways to load a datasheet picture into the editor: directly, or by using the HTML Viewer.

Loading a Datasheet Picture Directly

1. Browse directly to the datasheet picture file by choosing **Picture** > **Load Picture** to open a file browser window.
2. When you have located the desired file, click **OK** to load the image into the SheetScan editor. Supported file types include: .bmp, .dib, .jpg, .gif, .tif, .tga, .pcx, .htm, and .html.

Loading a Datasheet Picture Using the HTML Viewer

1. Choose **Picture** > **Internet** to open the HTML Viewer.
2. Browse the Internet for the desired datasheet.
3. Resize the HTML Viewer window and adjust its scrollbars until the desired portion of the datasheet is in view.

4. Click the **To SheetScan** button () to copy the visible contents into the SheetScan editor window.

> **Note:**
>
> To hide the datasheet picture, select **View** > **Picture**.

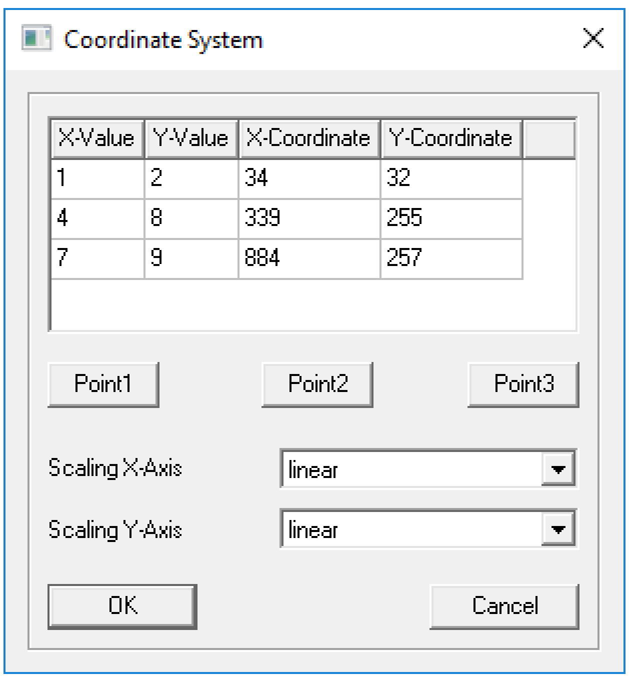
After loading a datasheet picture into the SheetScan editor, the next step is defining a coordinate system for the imported picture.

Defining a SheetScan Coordinate System

You can define a coordinate system for a graph on a datasheet picture that you have previously loaded into the SheetScan editor.

To define the coordinate system:

1. Select **Coordinate System** > **New** to open the **Coordinate System** dialog box.



You can enter points manually or select them using a crosshair.

2. To activate the crosshair cursor, click **Point1**.

The cursor transforms into a crosshair.

3. Position the cursor over a corner of the datasheet graph and click the left mouse button.

   The **Coordinate System** dialog box reappears, displaying the X- and Y-Coordinate values for the chosen point.

4. Enter the X-Value and Y-Value for this point. Typically, these values will correspond to the values taken from the axis scale values on the datasheet.
5. Select the desired scaling (linear, logarithmic, or decibel) for both the X and Y axes.
6. Repeat the above steps for **Point2** and **Point3**.
7. Click **OK**.

   The grid is placed over the graphic.

   > **Note:**
   >
   > - You can edit the grid after placement by selecting **Coordinate System** > **Properties**, by clicking the coordinate system icon on the Curve toolbar (), or by right-clicking in the SheetScan editing window and selecting **Coordinate System**.
   > - You can hide the grid by selecting **View** > **Grid**.

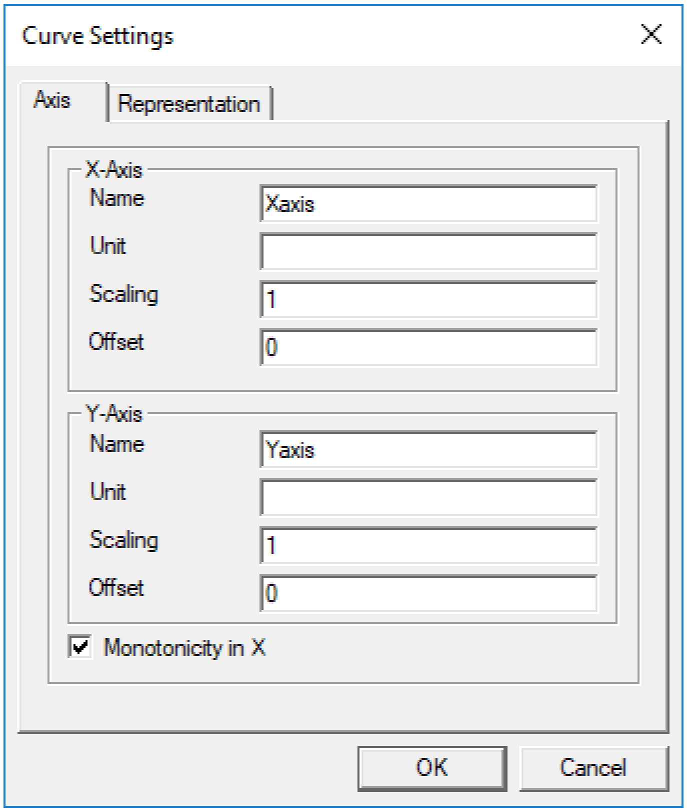
8. You can now define a Characteristic Curve.

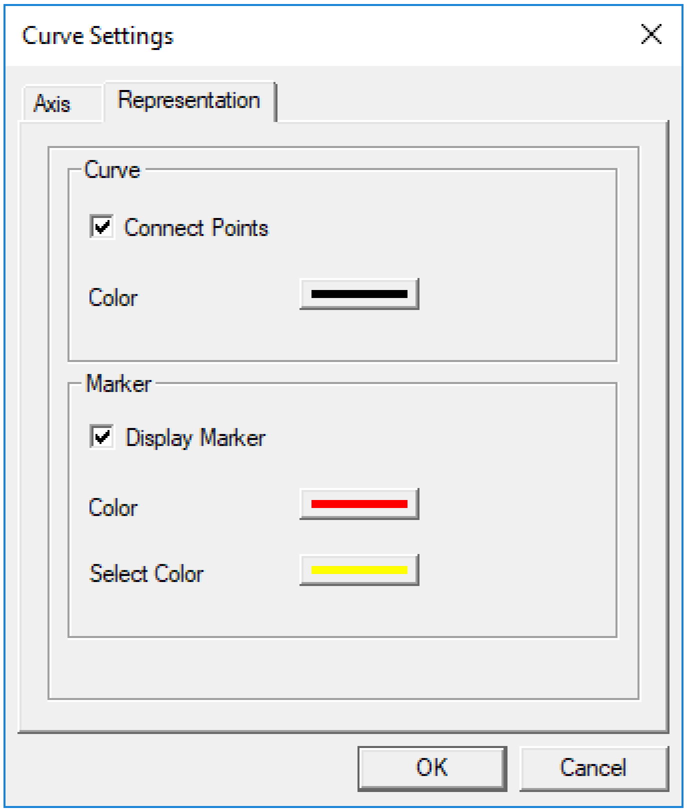
Defining a Characteristic Curve in SheetScan

Once you have loaded a datasheet picture in the editor and have defined a coordinate system, you can define one or more characteristic curves as follows:

1. Choose **Curve** > **New**.

   The **Curve Settings** dialog box opens, containing the **Axis** and **Representation** tabs.


Curve Settings
Axis
Representation
X-Axis
Name
Xaxis
Unit
Scaling
1
Offset
0
Y-Axis
Name
Yaxis
Unit
Scaling
1
Offset
0
Monotonicity in X
OK
Cancel



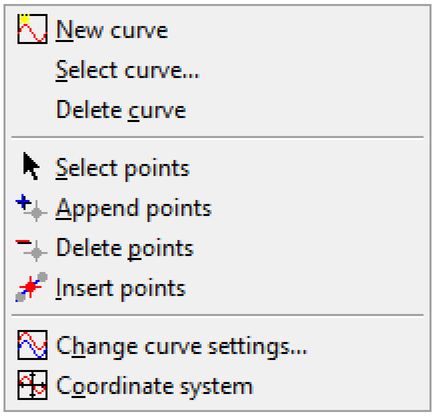
2. Use the fields on the **Axis** tab to define the curve's properties. Use the buttons on the **Representation** tab to change the curve's appearance.
3. Click **OK**.

   The cursor changes to a crosshair.
4. Click the points of the characteristic which you want to capture for the dataset. The points are connected automatically.
5. Repeat steps 1 through 4 for each additional characteristic curve you wish to define.

After characteristic curves have been defined, you can perform various operations on them. See: *Performing Operations on SheetScan Curves.*

Performing Operations on SheetScan Curves

Use the **Curve** menu (from the toolbar or by right-clicking in the sheet) to perform actions on SheetScan curves.



You can perform the following actions:

- **New Curve** – allows you to define a new characteristic curve.
- **Select Curve** – opens the **Select Curve** dialog box, listing all active curves. Select the desired curve and click **OK**.
- **Delete Curve** – deletes the selected curve.

> **Warning:**
>
> You cannot undo this action. If you delete a curve and its data points from the SheetScan editor, you must reconstruct it manually.

- **Select Points** – select this option, then click a point to select it. **Ctrl+click** allows you to select multiple points.
- **Append Points** – select this option, then click to add data points to the end of a curve.
- **Delete Points** – select this option, then click a data point to remove it from the curve.
- **Insert Points** – select this option, then click to insert new data points between existing data points.
- **Change Curve Settings** – opens the **Curve Settings** dialog box, where you can edit the initial curve settings.
- **Coordinate System** – opens the Coordinate System dialog box, where you can edit the coordinate system settings.

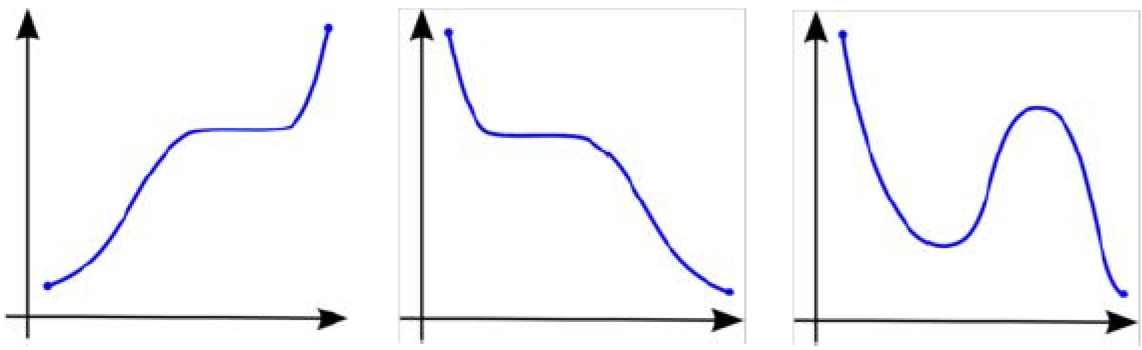
Checking for Monotonicity in X

EMIT requires that characteristics be monotonically increasing along the X-axis. In other words, successive data points must have increasing X-values, while Y-values may both increase and decrease.

To check for monotonicity in X from SheetScan:

1. Select **Curve** > **Check Monotonicity**.

   If the characteristic curve is monotonically increasing in X-value, the check completes without notice. Typical examples of curves that meet monotonicity criteria are shown below.



2. If the characteristic curve is not monotonically increasing in X-value, a dialog box displays informing you that errors were found. Click **Yes** to have SheetScan automatically correct the errors.

Importing Characteristic Data into SheetScan

SheetScan supports data import from the following file types:

- Twin Builder Characteristic (*.mdx, *.mda)
- Microsoft Excel (*.xls, *.xlsx)
- Text (*.txt)
- Comma Separated Value (*.csv)
- Spice (*.out)
- Comtrade (*.cfg)
- TEK Oscilloscope (*.dat)

> **Note:**
>
> Before importing characteristic data, you must define a coordinate system.

To import characteristic curve data into SheetScan:

1. Click **File** > **Import**.
2. In the file **Open** dialog box, select the desired data file and click **Open**.
   a. Selecting an **.xls** or **.xlsx** file containing multiple sheets opens a **Table Properties** dialog box where you can choose the desired sheet from a drop-down menu.

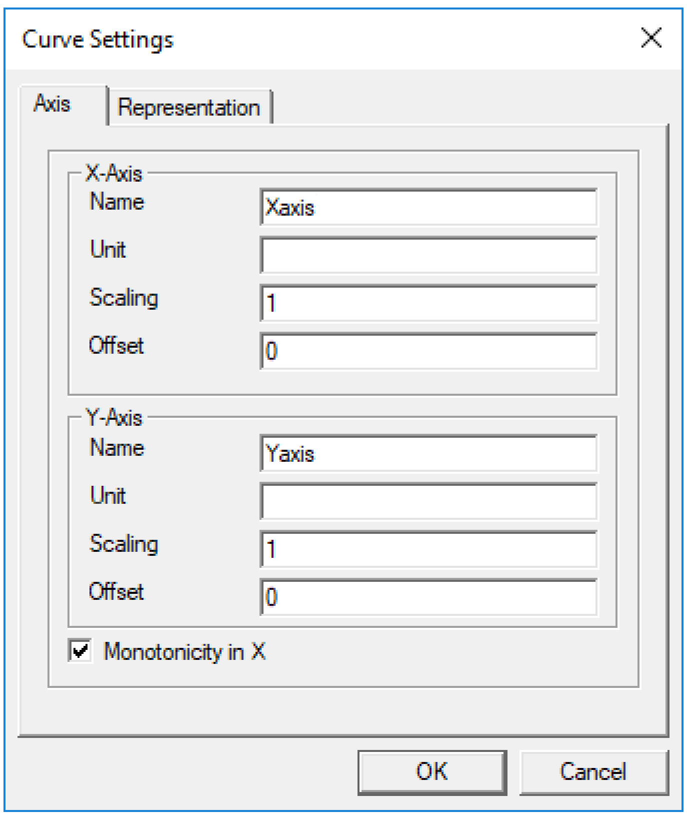
Otherwise, selecting an **.xls** or **.xlsx** file imports the data immediately into the **Add Dataset** dialog box.

> **Note:**
>
> - Only the first two columns of data are imported, the left-most column containing the X-coordinate values. The x-coordinate values for successive data points must increase within ten significant digits. Non-numeric entries are assigned a value of zero.
> - The first row of data is assumed to contain column headings and is ignored.

b. Selecting a **.txt** or **.csv** file opens an **Import** dialog box in which you can specify how to settings for reading the data in the file for import. You can choose the separator(s) and decimal symbol, as well as the line at which to begin the import. The dialog box shows both the original text and the text as it would appear when imported based on the current import settings.

When satisfied with the import settings, click **OK** to import the data.

The **Curve Settings** dialog box opens.



3. Change **Curve Settings** as needed and click **OK** to complete the data import.

   The new characteristic curve is added to the current SheetScan sheet.

Exporting SheetScan Data

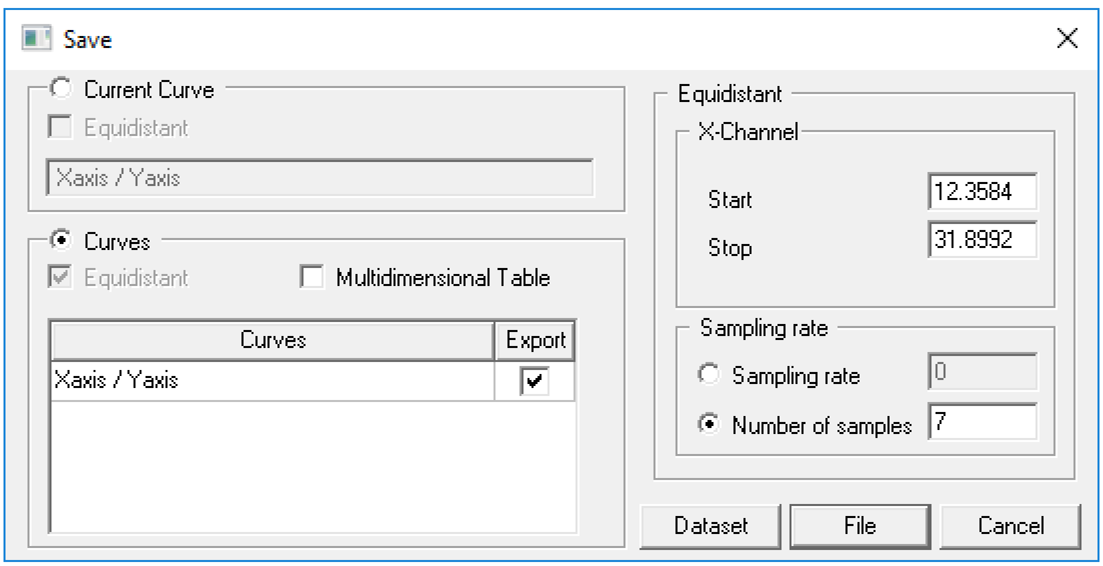
After defining a coordinate system and adding a curve, you can export SheetScan curve data directly to:

- Electronics Desktop Dataset
- Twin Builder Characteristic (*.mdx) file
- Comma Separated Value (*.csv) file
- Comtrade (*.cfg) file

To export curve data:

1. Click **File** > **Export**.

   The **Save** dialog box appears.



2. In the **Save** dialog box, choose **Current Curve** (default) to export current curve data, or **Curves** to choose the curve(s) whose data you wish to export. Choosing **Curves** reveals a list box showing all of the curves available for export.

   > **Note:**
   >
   > The **Multidimensional Table** option is not currently supported.

3. Choose **Equidistant** if you want to set the **Start** and **Stop X-Channel** values and a **Sample Rate** or **Number of samples** for the exported dataset(s).
4. Click **Dataset** to export curve data directly to the project's dataset file, or click **File** to export as one of the supported file types.

Defining an Expression

Expressions are mathematical descriptions that typically contain Intrinsic Functions, such as sin (x), and arithmetic operators, such as +, -, *, and /, as well as defined variables. For example, you could define: x_size = 1mm, y_size = x_size + sin(x_size). Defining one variable in terms of another makes it a dependent variable. Dependent variables, though useful in many situations, cannot be the subject of optimization, sensitivity analysis, tuning, or statistical analysis.

The Constants tab of the **Project Variables** window lists all available pre-defined constants. These may not be reassigned a new value.

Numerical values may be entered in Ansys shorthand for scientific notation. For example, $5x10^7$ could be entered as **5e7**.

Valid Operators for Expressions

The operators that can be used to define an expression or function are performed in a sequence.

The following table lists valid operators and the sequence in which they are accepted (listed in decreasing precedence):

()	Parenthesis	1
!	Not	2
^ (or **)	Exponentiation (If you use "**" for exponentiation, as in previous software versions, it is automatically changed to "^".)	3
-	Unary minus	4
*****	Multiplication	5
/	Division	5
+	Addition	6
-	Subtraction	6
==	Equals	7
!=	Not equals	7
>	Greater than	7
<	Less than	7
>=	Greater than or equal to	7
<=	Less than or equal to	7
&&	Logical and	8

||	Logical or	8

Using Intrinsic Functions in Expressions

EMIT recognizes a set of intrinsic trigonometric and mathematical functions that can be used to define expressions. Intrinsic function names are reserved and may not be used as variable names.

> **Note:**
>
> If units are not specified, all trigonometric functions interpret their arguments as radians. Likewise, return values of inverse trigonometric functions are output in radians. When the argument to a trigonometric expression is a variable, the units of the variable are assumed to be radians. To have values interpreted in degrees, supply the argument with the unit name **deg**.

The following intrinsic functions may be used to define expressions:

Function	Description	Syntax
abs	Absolute value (|x|)	abs(x)
sin	Sine	sin(x)
cos	Cosine	cos(x)
tan	Tangent	tan(x)
asin	Arcsine	asin(x)
acos	Arccosine	acos(x)
atan	Arctangent (in range of -90 to 90 degrees)	atan(x)
atan2	Arctangent (in range of -180 to 180 degrees)	atan2(y,x)
asinh	Hyperbolic Arcsine	asinh(x)
atanh	Hyperbolic Arctangent	atanh(x)
sinh	Hyperbolic Sine	sinh(x)
cosh	Hyperbolic Cosine	cosh(x)
tanh	Hyperbolic Tangent	tanh(x)
even	Returns 1 if integer part of the number is even; returns 0 otherwise.	even(x)
odd	Returns 1 if integer part of the number is odd; returns 0 otherwise.	odd(x)
sgn	Sign extraction	sgn(x)

exp	Exponential (e^x)	exp(x)
pow	Raise to power (x^y)	pow(x,y)
if	If	if(cond_exp,true_exp,false_exp)
pwl	Piecewise Linear. (**pwl** can be used with datasets for Design Variables but not for Project variables).	pwl(dataset_exp,variable)
pwl_periodic	Piecewise Linear for periodic extrapolation on x.	pwl_periodic (dataset_exp,variable)
sqrt	Square Root	sqrt(x)
ln	Natural Logarithm (The "log" function has been discontinued. If you use "log(x)" in an expression, the software automatically changes it to "ln (x)".)	ln(x)
log10	Logarithm base 10	log10(x)
int	Truncated integer function	int(x)
nint	Nearest integer	nint(x)
max	Maximum value of two parameters	max(x,y)
min	Minimum value of two parameters	min(x,y)
mod	Modulus	mod(x,y)
rem	Returns the fractional part of a decimal number such that rem (x) = x-int(x)	rem(x)
clp	Implements smooth interpolation employing weighted impact of all points of the dataset (not just the closest one). See formula below. **Note:** If used with a large 3D dataset, clp function will degrade.	clp (*datasetName* , X,Y,Z)

clp Formula

$$clp\left(Dataset, X, Y, Z\right) = \begin{cases} DatasetValue_i \text{ if } r_i\left(X, Y, Z\right) < r_{rounding} \\ \Sigma_i \frac{DatasetValue_i}{r_i^4} \Big/ \Sigma_i \frac{1}{r_i^4} \quad otherwise \end{cases}$$

$$r_i\left(X,Y,Z\right) = \sqrt{\left(X - DatasetX_i\right)^2 + \left(Y - DatasetY_i\right)^2 + \left(Z - DatasetZ_i\right)^2}$$

Using Piecewise Linear Functions in Expressions

The following piecewise linear intrinsic functions are accepted in expressions:

- **`pwl (dataset_expression, variable)`** – piecewise linear along the X-axis and returns a corresponding Y value.
- **`pwlx (dataset_expression, variable)`** – piecewise linear with linear extrapolation on x returns a corresponding Y value.
- **`pwl_periodic (dataset_expression, variable)`** – also interpolates along the X-axis, but periodically.

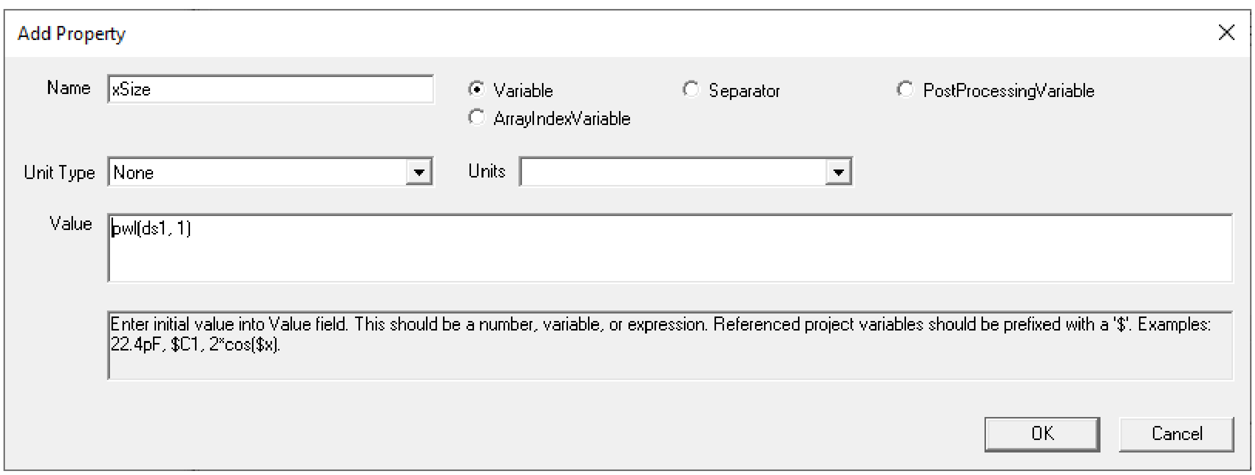
You can use **pwl** in an expression that uses datasets for such things as a frequency dependent material property (See: Adding Datasets).

For example, you can specify BulkConductivity as:

```
pwlx($ds1, Freq)
```

You can create a design variable representing a dimension `xSize` as `pwl(ds1, 1)` where `ds1` is a dataset.

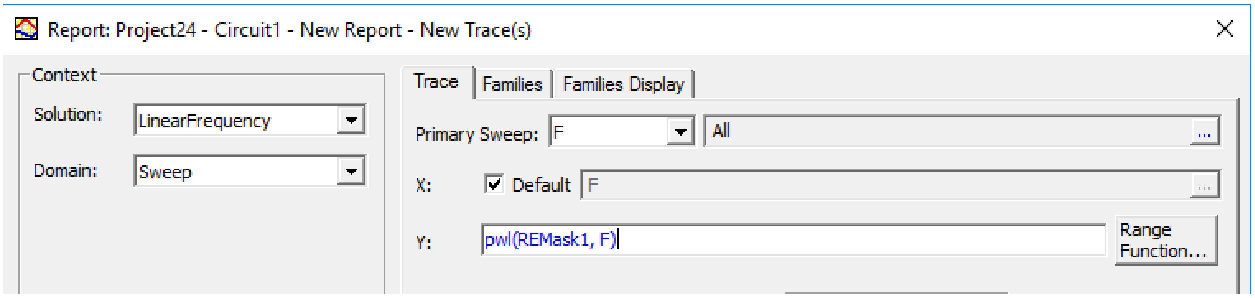
Doing so looks like this:



After the dataset is configured, the **Properties** window shows the new variable.

To use datasets in reports, create a report using **pwl(dataset_expression,**`variable`**)** where `variable` is the primary sweep in the report.

For example:



> **Note:**
>
> pwl can be used with datasets for Design Variables but not for Project variables.

Using Dataset Expressions

A dataset is a collection of data. It can take the following form:

$\$ds1((x_0,y_0),\ ...,\ (x_n,y_n))$

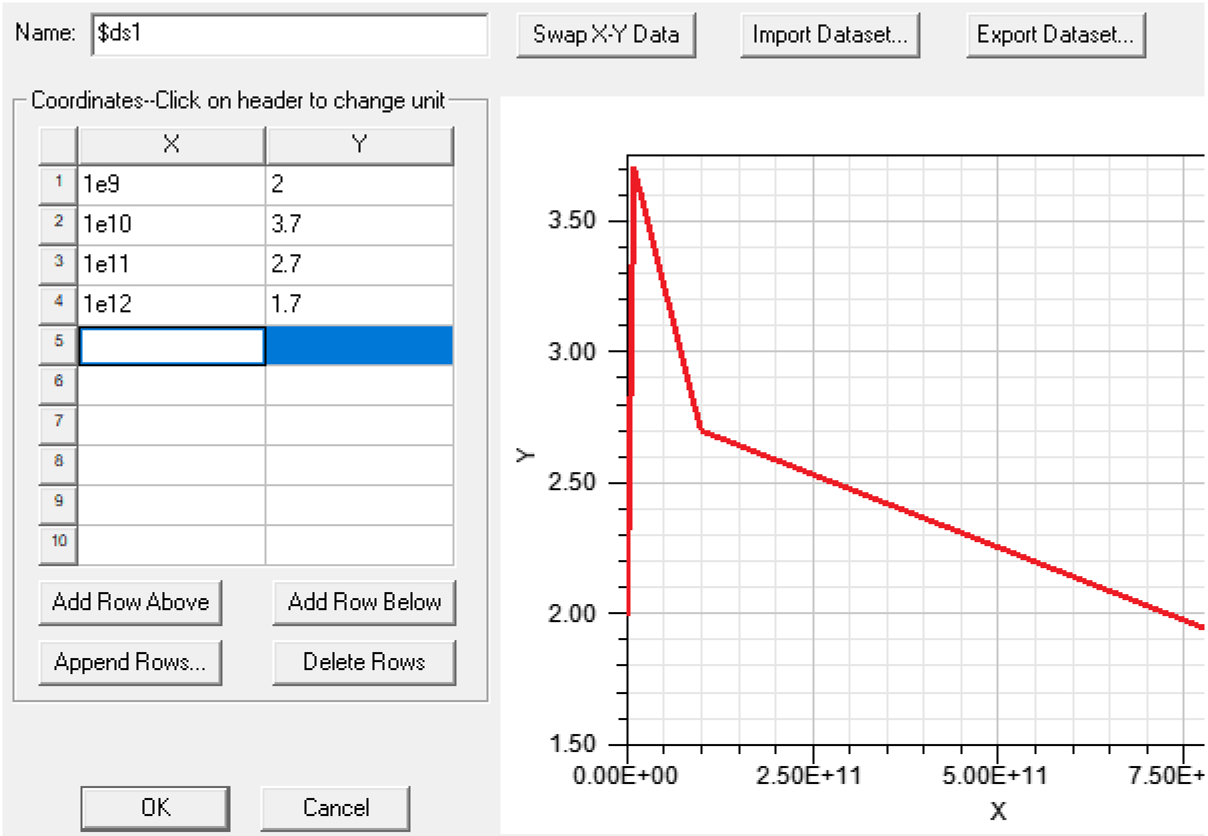
Once created, a dataset (such as $ds1) may be used as the first parameter to piecewise linear (**pwl**, **pwlx**, and **pwl_periodic**) functions, and may also be assigned to variables, in which case the variable may be used as the second parameter to **pwl**, **pwlx**, and **pwl_periodic** functions.

You can generate a dataset using a series of points in a plot on the **Datasets** dialog box (See: Adding Datasets). Each plot consists of straight line segments whose vertices represent their end points. A curve is fitted to the segments of the plot. This curve, which best fits the segmented plot, consists of the co-ordinates used in the creation of the dataset. The dataset may then be used in the piecewise linear intrinsic functions.

> **Note:**
>
> The following is an example showing how to make a material property frequency dependent using a dataset as the first parameter to a **pwl** function. The values used are arbitrary.

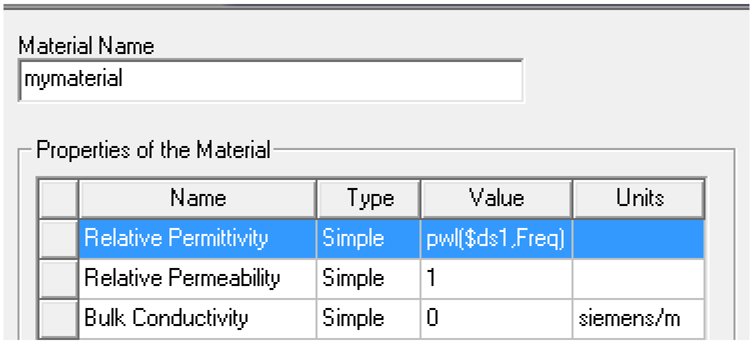
1. For a project-level dataset, click **Project** > **Datasets**. For a design-level dataset, click **EMIT** > **Design Datasets**.
2. Click **Add**.
3. Set the **Name** field and **Coordinates** as desired and click **OK**.

   The dataset is created.



> **Note:**
>
> By default, the dollar sign ($) is assigned to a project dataset even if you do not use one while naming it.

4. Go to **Tools** > **Edit Libraries** > **Materials** to open the **Edit Libraries** dialog box.
5. Click **Add Material**.
6. Type in the piecewise linear function and use the dataset $ds1, as shown in the following figure.



7. Click **OK**.

Handling Delta Temperature Units in Expressions

When Temperature and Delta Temperature quantities are used as operands to plus or minus operations in an expression, they are handled specially. The biggest difference is an automatic unit change of the resulting values based on the units of the operands.

- When two Temperature quantities are being subtracted (either the first or both operands is/are temperature quantities), the result value has Delta Temperature units.
- When a Delta Temperature quantity is added or subtracted from a Temperature quantity, the result value has Temperature units.
- Adding or subtracting two Delta Temperature quantities results in a quantity with Delta Temperature units.
- Subtracting two temperature quantities is an observable behavior change when compared to earlier releases of Ansys Electronics Desktop.

Temperature Units

Celsius Family

- cel, delta_cel (Legacy name for delta_cel, celdiff continues to be supported)

Kelvin Family

- mkel, delta_mkel (milli Kelvin) (Legacy name for delta_mkel, mkeldiff continues to be supported)
- ckel, delta_ckel (centi Kelvin)
- dkel, delta_dkel (deci Kelvin)
- kel, delta_kel (Legacy name for delta_kel, keldiff continues to be supported)

Farenheit Family

- fah, delta_fah

Temperature-related Use Cases

The high-level legal use cases (in unit types) are:

- Temperature - Temperature = DeltaTemperature
- Temperature +/- DeltaTemperature = Temperature
- DeltaTemperature +/- DeltaTemperature = DeltaTemperature

The minor variations in these use cases specify how the actual units (Celsius is a unit of the Temperature unit type) are handled.

Temperature - Temperature

If they have the same unit, then the Delta Temperature unit will be a matching one.

For example:

- 10 cel - 5 cel = 5 delta_cel
- 100 kel - 90kel = 10 delta_kel
- 10 fah - 1 fah = 9 delta_fah

If they have different units, they are converted to the default units for temperature difference.

For example:

- 10cel - 1 kel = 9 delta_kel

Temperature +/- DeltaTemperature

The resulting temperature quantity will retain the units of the Temperature quantity (first operand)

DeltaTemperature +/- DeltaTemperature

- If they have the same unit, the resulting Delta Temperature quantity will retain that.
- If they have different units, the resulting Delta Temperature quantity will have the default Delta Temperature units
- 5 delta_cel + 10 delta_cel = 15 delta_cel
- 10 delta_cel + 1 delta_kel = 11 delta_kel

All other use cases in a plus or minus arithmetic operation are physically meaningless and revert to previous behavior, where Electronics Desktop converts quantities to their SI values and then operates on the plain numbers.

- Temperature + Temperature
- Temperature +/- non-Temperature/non-Delta Temperature
- Delta Temperature +/- non-Temperature/non-Delta Temperature
- non-Temperature/non-Delta Temperature +/- Delta Temperature
- non-Temperature/non-Delta Temperature +/- Temperature

Defining Mathematical Functions

A mathematical function is an expression that references another defined variable. A function's definition can include both expressions and variables.

The following mathematical functions may be used to define expressions:

<table>
<tr><td>Basic Functions</td><td>/, +, -, *, mod (modulus), ** (exponentiation), - (Unary minus), == (equals), ! (not), != (not equals), > (greater than), < (less than), >= (greater than equals), <= (less than equals), && (logical and), || (logical or)</td></tr>
<tr><td>Intrinsic Functions</td><td>if, abs, exp, pow, ln (natural log), log10 (log to the base 10), sqrt</td></tr>
<tr><td>Trigonometric Expressions</td><td>sin, cos, tan, asin, acos, atan, sinh, cosh, tanh</td></tr>
</table>

The predefined variables X, Y, Z, Phi, Theta, R, and Rho must be entered as such. X, Y, and Z are the rectangular (Cartesian) coordinates. Phi, Theta, and Rho are the spherical coordinates. R is the cylindrical radius, and Rho is the spherical radius.

If you do not specify units, all trigonometric expressions expect their arguments to be in radians, and the inverse trigonometric functions' return values are in radians. If you want to use degrees, you must supply the unit name **deg**. When the argument to a trigonometric expression is a variable, the units are assumed to be radians. These function names are reserved and may not be used as variable names.

As far as expression evaluation is concerned: units are conversion factors (that is, from the given unit to SI). Note also that the evaluated value of an expression) is always interpreted as in SI units.