cadence®

Allegro®
RF-PCB IFF Import User Guide

Product Version 23.1
September 2023

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

Contents

Preface

About This User Guide

The *RF-PCB IFF Import User Guide:*

- Explains how to use the RF-PCB IFF Import feature to import a radio-frequency (RF) design into a Design Entry HDL schematic
- Lists the different ways of starting the import
- Describes the user interface
- Lists the different symbol types
- Describes how to change the symbol types
- Explains the different ways of specifying the JEDEC_TYPE values for symbols

Before you use the RF-PCB IFF Import, you should be familiar with using the Design Entry HDL and Allegro RF PCB layout tools.

Finding Information in This User Guide

This user guide covers the following topics:

See...	For Information About...
Importing IFF Files	Describes the features of RF-PCB IFF Import and the basic steps for adding an RF design to an Allegro Design Entry HDL schematic. Also compares the standard IFF Import and the RF-PCB IFF Import.
RF-PCB IFF Import User Interface	Describes the RF-PCB IFF Import interface

Related Documentation

You can also refer to the following documentation to know more about related tools and methodologies:

Design Entry HDL

- For information on the new features available in the current release, see *Allegro Design Entry HDL: What's New in Release 17.2*.
- For learning Design Entry HDL, see *Allegro Design Entry HDL Tutorial*.

Related Tools and Flows

- For information on various PCB design working environments, such as a team of designers working on a Design Entry HDL project, implementing FPGAs in designs, working with high-speed constraints, importing IFF files for radio-frequency designs, and reusing existing modules, see *Allegro PCB Design Flows*.
- For learning how to create new Design Entry HDL projects and make various settings for them, see the *Project Manager User Guide*.
- For learning how to use the Design Entry HDL utilities, such as CRefer, Archiver and BOM, see the *Design Entry HDL Utilities User Guide*.
- To know more about RF-PCB, see *Allegro PCB Editor User Guide: Working with RF PCB*.
- To know more about the component libraries delivered with Allegro RF PCB, see *Allegro RF PCB Library Reference*.

Typographic and Syntax Conventions

This list describes the syntax conventions used for this user guide:

`literal`	Nonitalic words indicate keywords that you must enter literally. These keywords represent command (function, routine) or option names.
`argument`	Words in italics indicate user-defined arguments for which you must substitute a name or a value.
|	Vertical bars (OR-bars) separate possible choices for a single argument. They take precedence over any other character.
`[ ]`	Brackets denote optional arguments. When used with OR-bars, they enclose a list of choices. You can choose one argument from the list.
`{ }`	Braces are used with OR-bars and enclose a list of choices. You must choose one argument from the list.

1

Importing IFF Files

Introduction

Some designers create Radio Frequency (RF) designs with products, such as the Keysight Advanced Design System (ADS) product from Keysight Technologies, Inc., and then need to include this RF design as a block into a larger Allegro Design Entry HDL schematic.

With the RF PCB released on SPB 15.7, Cadence offered a powerful tool to do the RF design on PCB. RF PCB also offers a bidirectional interface in the layout side with ADS.

In this RF design flow using the RF PCB product, designers start a design from Design Entry HDL with the parameterized RF library, and then transfer the design to Allegro PCB Design to do the layout with RF PCB tools.

However, currently there are some designers who start their design and perform the simulation in ADS. Using the RF-PCB IFF Import in Design Entry HDL, you can now import RF designs into Design Entry HDL schematics.

Once the RF design is ready, it is exported to an Intermediate File Format (IFF) file, and this IFF file is imported into a Design Entry HDL schematic.

Design Entry HDL supports two IFF interfaces for importing an IFF file into a schematic:

- Standard IFF Import

  This is the default IFF import option available with Design Entry HDL. For information on the standard IFF import, see *Chapter 4: Importing Radio-Frequency Designs in the Cadence document PCB Design Flows Guide*.

- RF-PCB IFF Import

  This is the new IFF Import, which is described in this document.

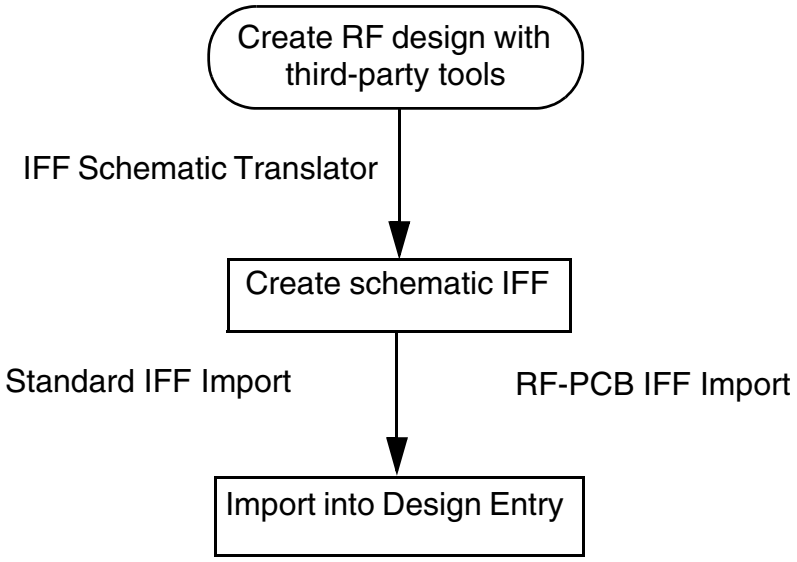
Important

All the IFF Import features are available only in the XL tiers. Therefore, RF-PCB IFF Import is available with the following licenses:

- Concept_HDL_expert
- Allegro_Design_Editor_620
- PCB_librarian_expert
- PCB_design_expert
- SPECCTRAQuest_EE

See "Comparing IFF Import and RF-PCB IFF Import" on page 30 to know more about the differences between the two IFF import features.

The following flowchart depicts how the IFF import works.

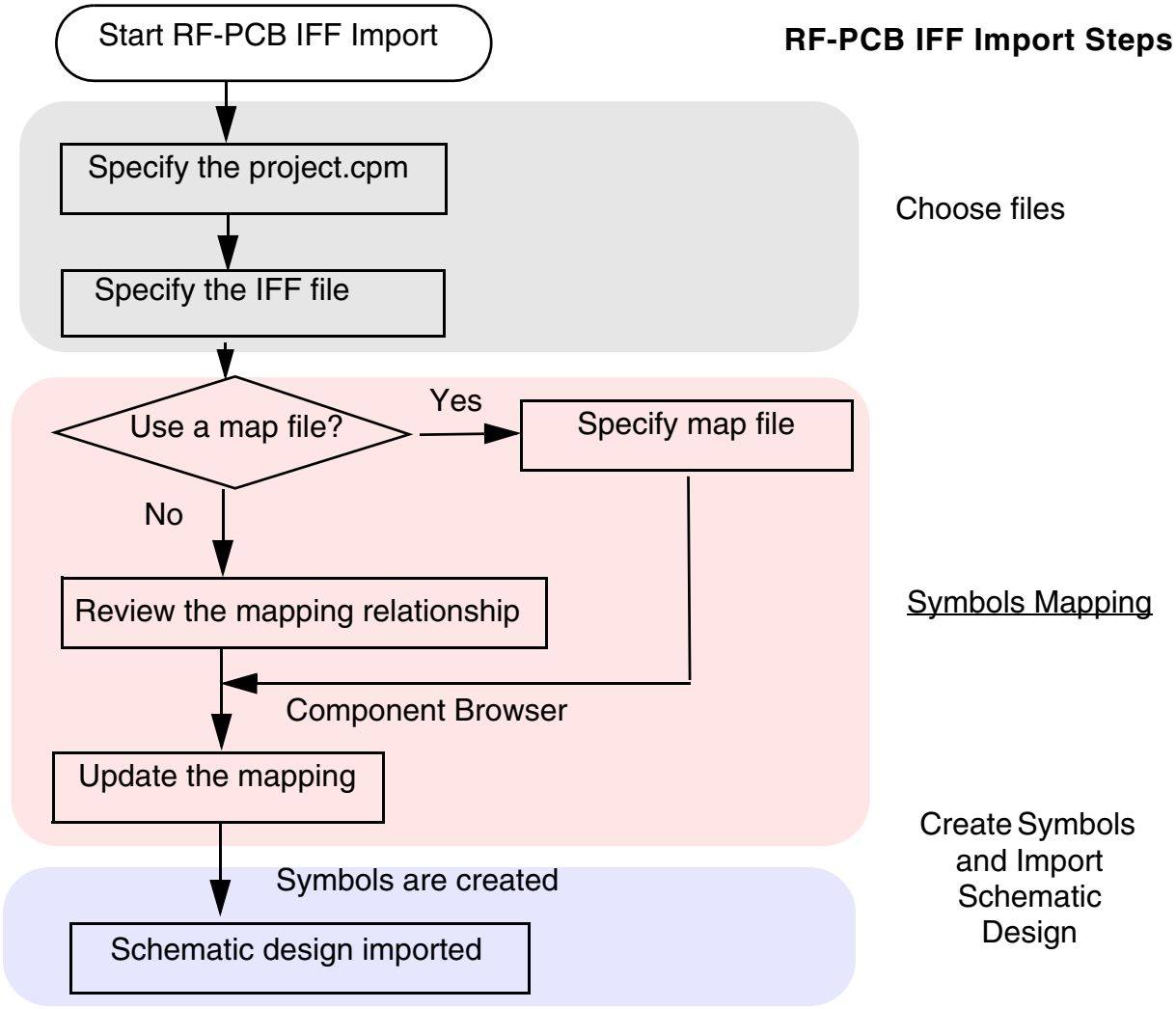


RF-PCB IFF Import Overview

RF-PCB IFF Import has a wizard interface that comprises the following steps:

- Step 1 *Choose files*
- Step 2 *Symbols Mapping*
- Step 3 *Create Symbols and Import Schematic Design*

The following figure shows how the RF-PCB IFF Import works and what part of the process maps to each step of the wizard.



Before you import an IFF file into a Design Entry HDL schematic, you need to understand how the mapping information is created, modified, and stored. The section, Symbols Mapping Overview, explains how mapping works.

Symbols Mapping Overview

The dictionary files available at `<installation_hierarchy>\share\cdssetup\concept` stores the details of the RF components and their corresponding Cadence component values. Three types of dictionary file exist:

- `rfpcb.dic:` Stores the details of the ADS components and their corresponding Cadence component values.
- `rfpcbmwo.dic:` Stores the details of the MWO components and their corresponding Cadence component values.
- `rfpcbunified.dic:` Stores the details of the Unified components and their corresponding Cadence component values.

For each component, its type is also specified. The possible symbol types in this file are:

- Parameterized
- Simulation-only
- Discrete
- IC
- IO
- Power

RF-PCB IFF Import reads the `.dic` file and the input IFF file and creates a basic mapping for you. You can then change the symbol mapping information as applicable for your design.

Mapping Considerations

Some important points to remember are:

- Parameterized Symbols

  For the ADS-compatible parameterized symbols, the importer finds the local library and updates the properties of the instances **without** creating a new local library. For these symbols, all the fields are filled automatically and set as read-only.

- Simulation-Only Symbols

  Simulation-only components, such as substrates and variables, are used for simulation only. Therefore, for these simulation-only symbols, the library and cell fields are blank and disabled. These are not a part of the mapping and are not transferred to the back-end and have no footprint on the back-end.

- Discrete

  For all symbols that are not `Parameterized`, the importer sets the default type to Discrete.

- Hierarchical Components

  Hierarchical components do not need symbol mapping. However, a new library cell named as <*`hierarchical component name`*> is created. Additionally, a symbol view is created instead of the chips view.

  The corresponding circuit page information for this component from the IFF file is written to a `.csa` file, which is actually the schematic page for this hierarchical component.

- Root Design

  The root design from the IFF file is treated as a hierarchical component and a block is used to denote its symbol. By expanding the block, Design Entry HDL brings up a new page with detailed elements of this design.

How Mapping Works

Once you specify the name of the project file and the IFF file, RF-PCB IFF Import reads the IFF file and the `.dic` file, and collects all the symbol definitions.

1. For `Parameterized` symbols, the importer updates the local library with the properties of the instances without creating a new local library. For these symbols, all the fields are automatically filled and set as read-only.
2. The information for the `Discrete` symbols is populated.
3. All the remaining symbols are set as `Discrete`.

   You can modify the mapping for these symbols. The possible options are:

   - Simulation-only

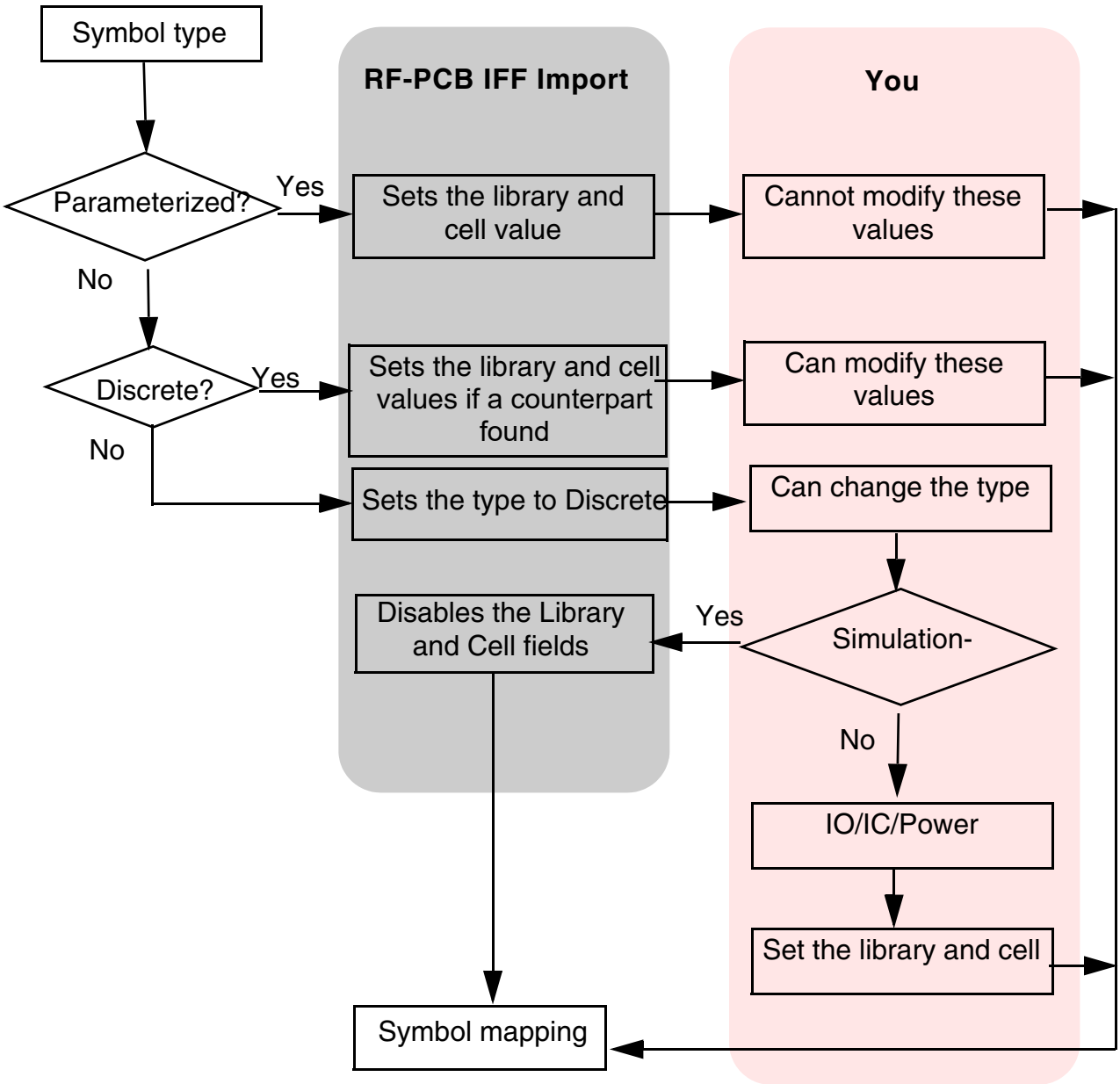
     Are not carried to the back-end.

   - IO

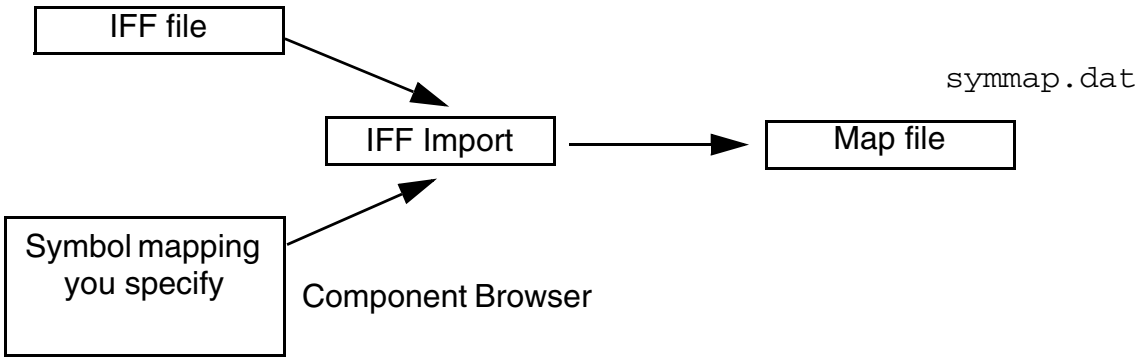
- IC
- Power

You can map all categories of symbols **except** `Parameterized` and `Simulation-only` to a local library. The importer searches the local libraries as defined in `cds.lib` for counterparts. If a counterpart is found, it is used as the default one to be mapped. Otherwise, there is no default local library specified.

The following flowchart depicts how the initial mapping is derived by RF-PCB IFF Import and how you can change the mapping information:



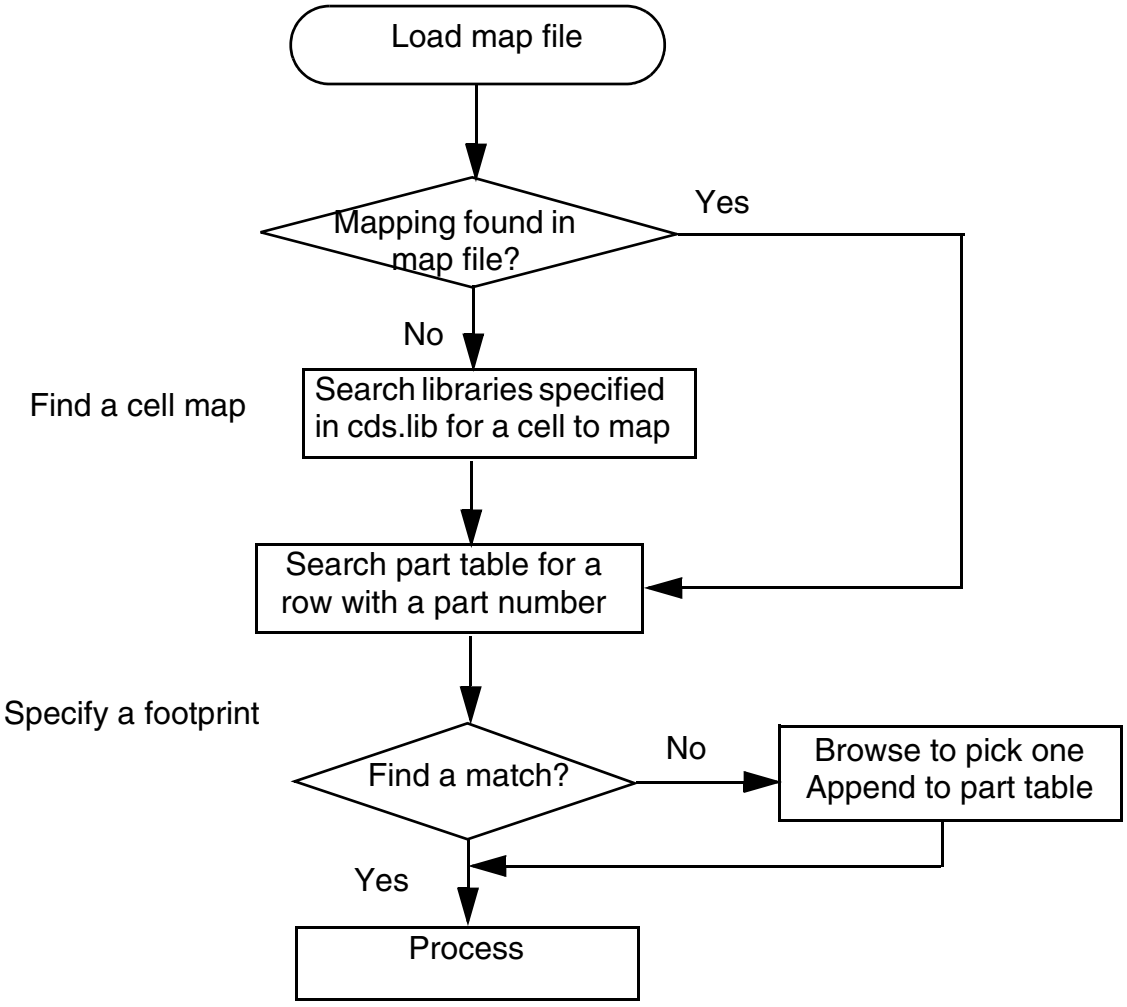
The mapping you set is stored in a map file. Later, you can direct the importer to use this map file for mapping the symbols.



After you have specified the *<project>*.cpm and IFF files, the *Symbols Mapping* screen of the RF-PCB IFF Import wizard shows the current mapping information for the IFF being imported.

For the non-Parameterized components, IFF Import maps them to local libraries. You can choose a value in the local library for each ADS non-Parameterized component individually. This mapping relationship is stored in a local map file called symmap.dat.

The following figure shows how these symbols are processed.



Importing an IFF File into a Design Entry HDL Schematic

To import the schematic IFF file into a `<project>.cpm`, follow these steps:

1. Start RF-PCB IFF Import.

   You can access the RF-PCB IFF Import from:

   - Project Manager

     Choose *File – Import IFF – RF PCB*.

   - Design Entry HDL

     Choose *RF-PCB – Import IFF*.

   - Command line

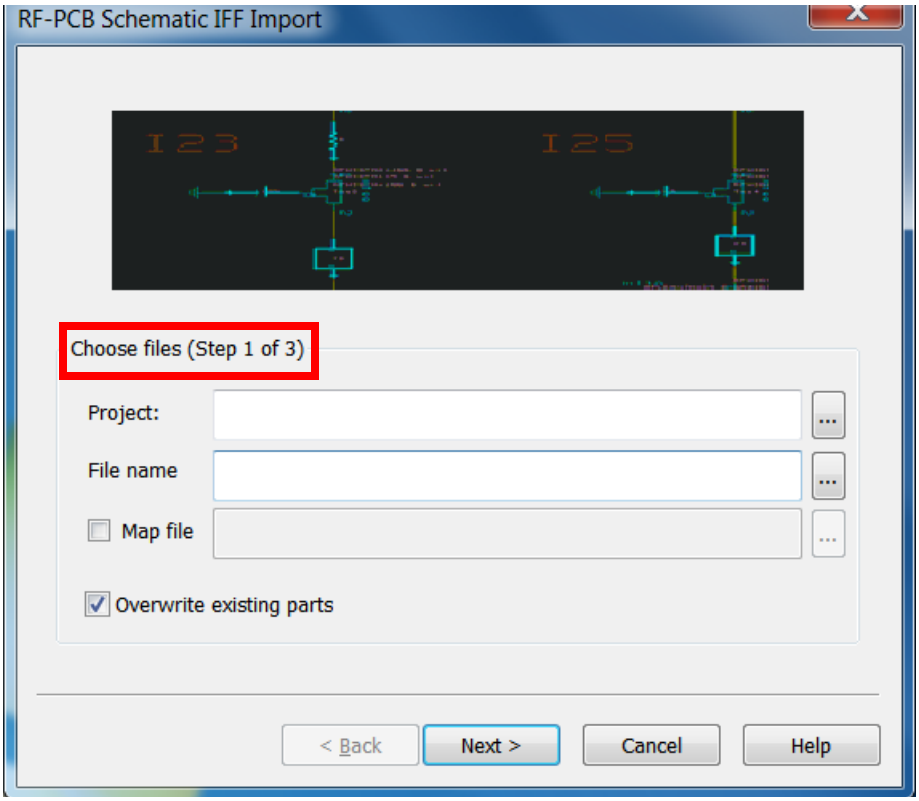
     Type `rfpcbiff2hdl` and press *Enter*.

     The complete syntax of the command is:

     ```
     rfpcbiff2hdl [-proj proj_file] [-product license_string] [-iff
     iff_file] [-lm layermap_file] [-mpssession session_name] [-sm
     symmbolmap_file] [-overwrite]
     ```

     See "Running RF-PCB IFF Import from the Command Line" on page 38 for more details.

The RF-PCB IFF Import wizard starts. The *Choose files* screen appears.



See “Choose files” on page 32 for details of this screen.

2. Specify the name of the project.

   You can use the browse button to navigate to the folder containing the `<project>.cpm` file. If you type the name of the project, the importer looks for the `<project>.cpm` in the current folder only.

   If you launch RF-PCB IFF Import when Project Manager with a project file open, or from Design Entry HDL, this field shows the currently open project name and is unavailable for modifications.

3. Specify the name of the IFF file in the File name field.

   The IFF file contains the schematic RF design and could be created with a third-party RF design-creating software, such as ADS.

Note: ADS also exports layout IFF design, which are invalid input for RF-PCB IFF Import.

4. Select the Map File option button, if you have a map file.

   When you import an IFF file into a design cpm, you specify the mapping of the new symbols. This mapping information is stored in a map file.

   If you have done the mapping earlier and you are importing the same design with only slight modifications, you can use an existing map file. When you specify a mapping file, the second step of the RF-PCB IFF Import wizard displays the mapping information and you can retain or change the mapping as needed.

   See "Symbols Mapping Overview" on page 14 for more information on mapping.

5. Select the *Overwrite existing parts* check box, if you want to overwrite the existing parts in the specified library during import.

   By default, this is selected.

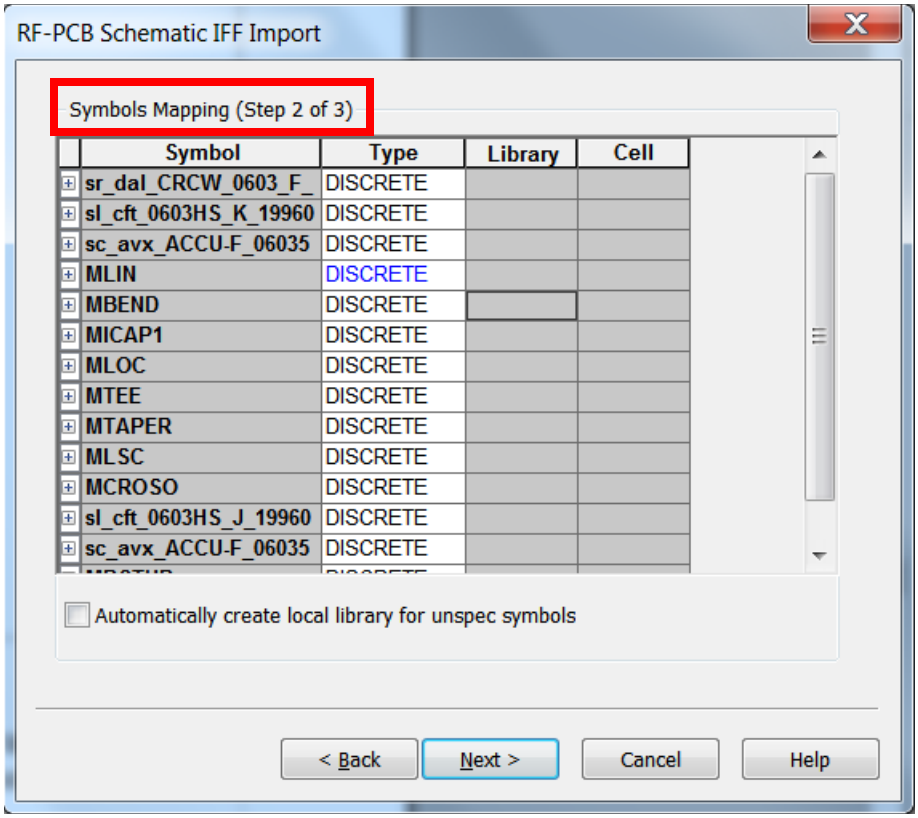
6. Click *Next*.

   The importer parses the specified IFF file and collects all the symbol definitions and component instances in the IFF file. If the IFF file is invalid, the importer moves to step 3 with an error message and then exits. Otherwise, if there are symbols that need mapping to local libraries, you proceed to step 2, *Symbols Mapping*, or move to step 3 to import the design directly.

   The *Symbols Mapping* screen of the RF-PCB IFF Import wizard is displayed.

   See "Symbols Mapping" on page 35 for details of this screen.

You can sort the display by any the displayed columns. Double-click the column header, to sort in the order of that column. To reverse the sort order, double-click the column header.



Some important things to note on this screen are:

- For the parameterized symbols, the importer updates the local library without creating a new local library. Therefore, for these symbols, all the fields are filled automatically and set as read-only.
- For the other symbols, importer searches the local libraries as defined in `cds.lib` for the counterparts.
  - If a counterpart is found, it will be used as a default one to be mapped.
  - Otherwise, there are no default local library specified.
- You can map symbols to another library or even delete their settings by pressing the `Delete` key on the keyboard at the cell to force the association with a new local library.

  See step 9 for details.
- You might need to set the JEDEC_TYPE value for some symbols

There may be some symbols that have the same part name but different part numbers.

See step 10 for details.

- The symbols that have no local library or cell associations, are treated as un-recognized symbols.

7. Specify the mapping information for the IFF file components.

In case you have specified a symbol map file, the mapping relationship is displayed as the initial setting. If necessary, you can change these settings and override the initial mapping.

In case you do not have any map file, you need to specify symbols or component instances that need mapping. RF-PCB IFF Import walks you through all the local libraries as specified in the `cds.lib`, and classifies the symbols.

A new mapping file, `symmap.dat`, is created and the mappings you specify are stored in it.

Initially, the mapping information, libraries and cells, are in black. If you make any changes, they are in red. This indicates that they have been modified.

For all discrete parts or IC/IO components, you need to specify the *JEDEC_TYPE*.

8. Change the symbol type of the Discrete symbols, as appropriate, to any of the following:

- Simulation-only
- IC
- IO
- Power

The editable symbol types are displayed with a white background. To change the symbol type, follow these steps:

a. Click the symbol type, such as *Discrete* or *Power.*

The other symbol types are displayed in a list.

b. Select the required symbol type.

9. Specify the library and cell for the symbol as applicable.

Some of the symbol names are in bold. For these, you need to specify mapping inputs, such as library name, cell name, or JEDEC_TYPE.

a. Double-click the symbol row.

The Universal Browser dialog box opens.

b. Select the library.

c. Select the cell.

d. Click *OK*.

10. Specify a value for the *JEDEC_TYPE* field as needed.

You can set the JEDEC_TYPE in any of the following ways:

- Type in a JEDEC_TYPE value

  See step 11 for details.

- Double-click the symbol to open the Universal Browser window and select a library and cell.

  This is the same as step 9.

- Double-click in the JEDEC_TYPE column to select a package symbol (.psm) file.

  See step 12 for details.

  Move to step 13 after you have set all the JEDEC_TYPE values.

11. To set the JEDEC_TYPE by typing in value, follow these steps. Otherwise skip to the next step.

a. Click in the JEDEC_TYPE column

a. Type in the value

12. To set the JEDEC_TYPE by using a .psm file, follow these steps. Otherwise skip to the next step:

b. Double-click in the *JEDEC_TYPE* cell for the symbol.

c. Navigate to the .psm file location.

d. Select a .psm file.

e. Click *OK.*

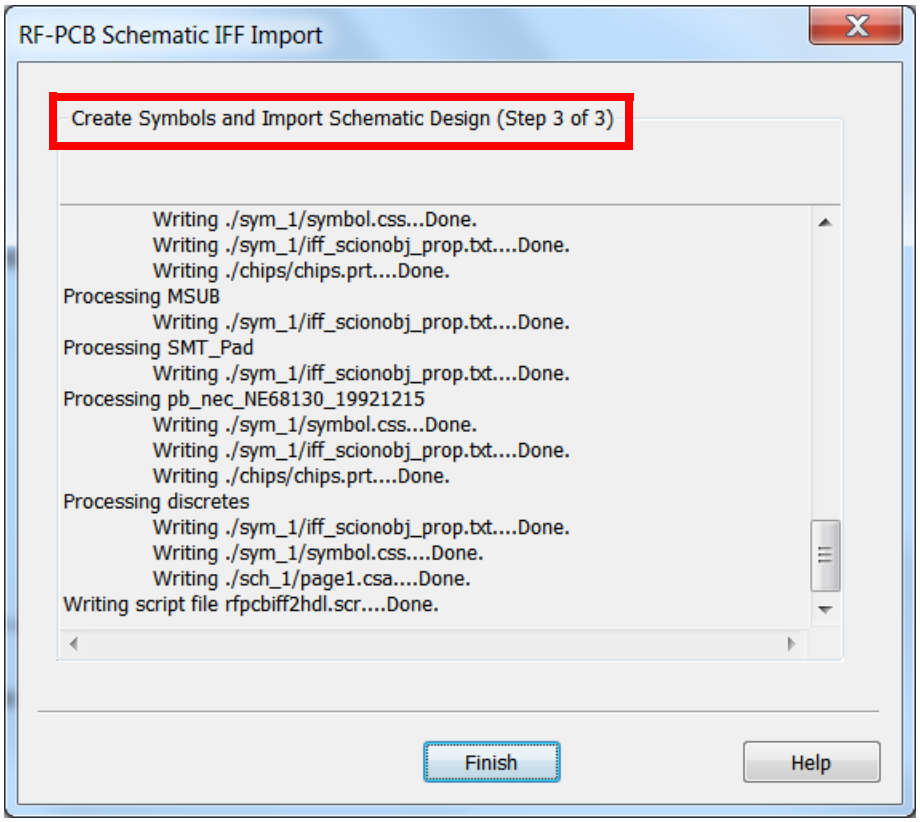
13. Ensure the *Automatically create local library for unspec symbols* check box is selected.

When you select this option, IFF import does not display warning messages for unrecognized symbols. However, if there is any unrecognized symbol that has no symbol page definition in the IFF file, no local library cell will be created.

Note: When you export an IFF file from a third-party tool, remember to include the symbol information in the IFF file.

14. Click *Next*.

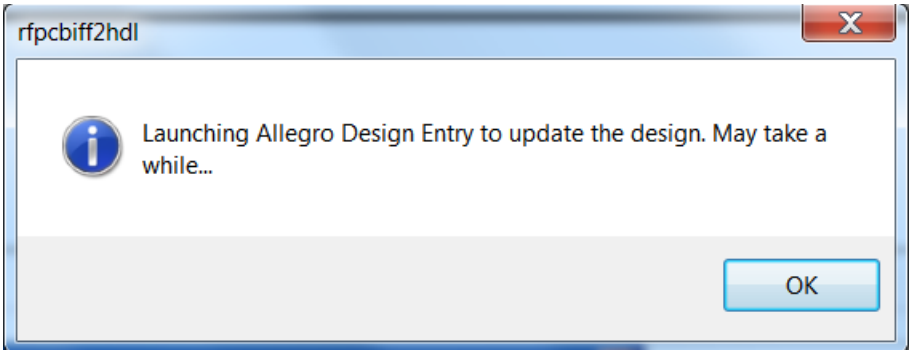
The *Create Symbols and Import Schematic Design* screen of the RF-PCB IFF Import wizard is displayed.



This shows the progress of the mapping.

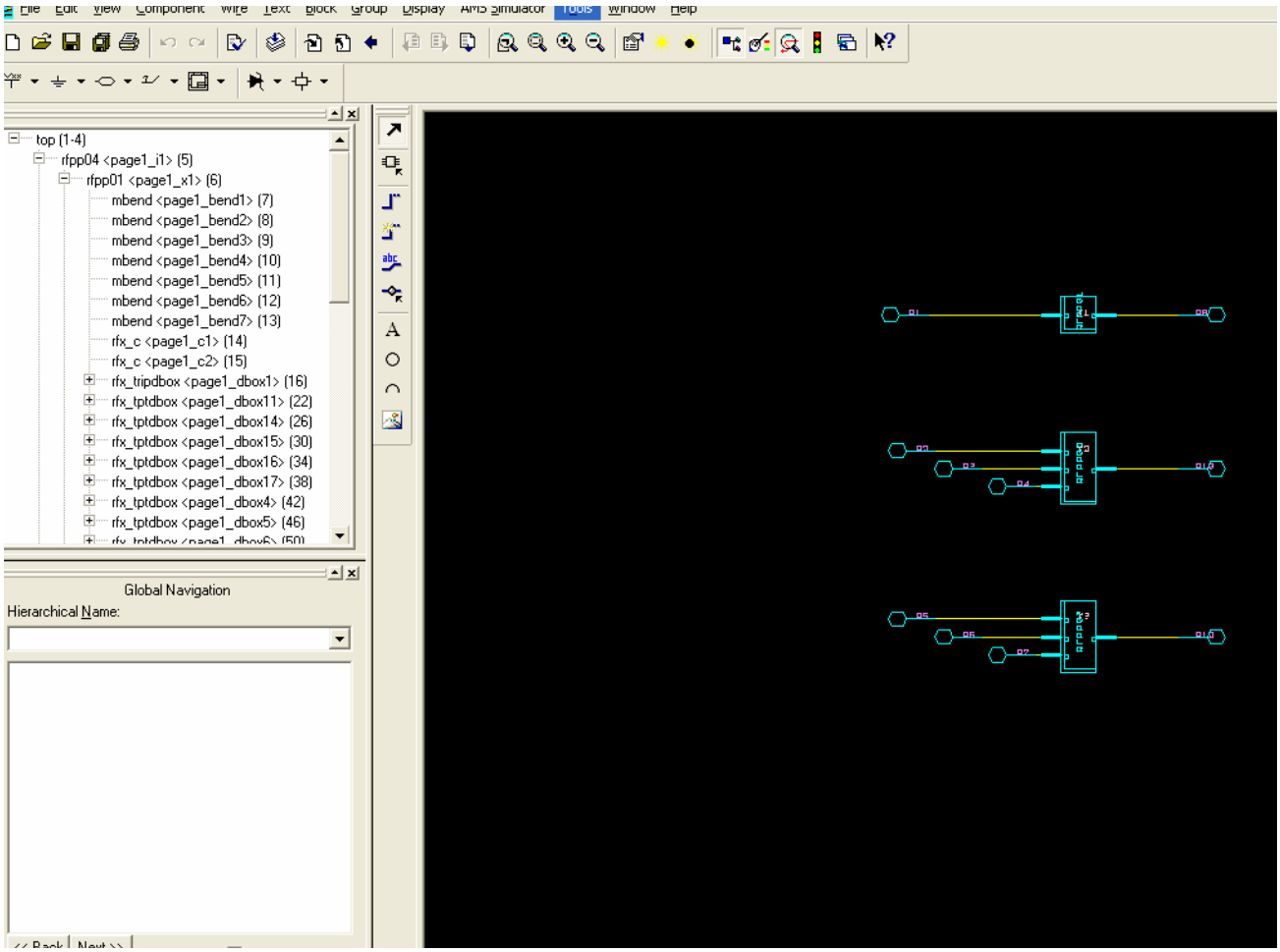
15. Click *Finish*.

Design Entry HDL is launched if it is not running:



16. Click *OK*.

Design Entry HDL executes a script file which completes the import. After completing the import, the RF design as placed on the schematic as a block.



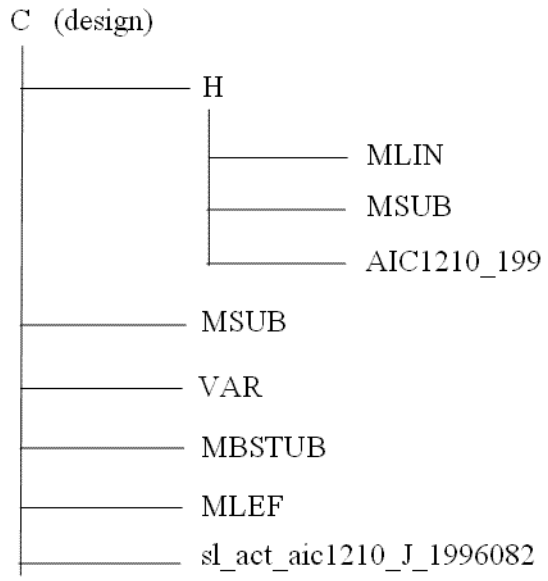
Directory Structure and Files

To understand the directory structure and files, consider the following example:

A project file named `a.cpm` locates at `D:\SPB_DATA`. An IFF file, named `b.iff`, to be imported locates at `D:\ ADS2005A`. The `b.iff` file structure is shown in figure 1-1. Its components are:

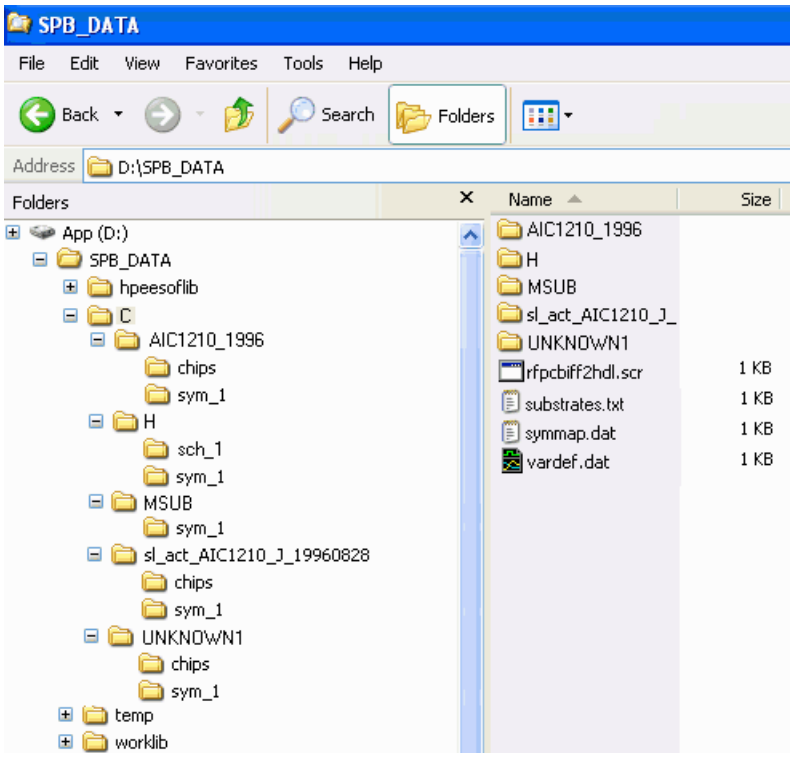
- A root design called `C`
- `MLIN`
- `MBSTUB`
- `MLEF`
- `AIC1210_1996`
- A hierarchical component `H`
- `VAR`
- `MSUB`

Figure 1-1 IFF File Structure



After importing the IFF file, a subdirectory `C` is created at the same directory as the `a.cpm` project. The structure of subdirectory `C` is shown in figure 1-2.

Figure 1-2 Structure of Added Components



Here is a description of some of the files and folders:

- The `vardef.dat` file stores the VAR definition and the `symmap.dat` file stores the symbol mapping relationship from Step 2 of the import process. Both these files are saved under `C`.
- For the hierarchical component `H`, a schematic page is added under the `sch_1` directory.
- For the `MSUB` substrate, the `iff_scionobj_prop.txt` file that stores the default simulation properties of the substrate is stored under the `sym_1` folder.
- The `substrates.txt` file stores all the properties of the substrates used in the IFF file is saved under `the C directory` as well. This is for future IFF export usage only.

Finally, the `C` directory is included in `cds.lib` as the local library to be used for the design.

Comparing IFF Import and RF-PCB IFF Import

The following table (1-1) shows you a comparison of the two IFF Import options:

Table 1-1 Available IFF Import options

Feature	Standard IFF Import	RF-PCB IFF Import
ADS discrete components	Use DEVICE property to map	The ADS discrete components are mapped to local libraries, which are created by the library translator in advance. Also supports the DEVICE property mapping.
Design consistency	RF components can not be passed to PCB Editor as components. Does not support RF components in the back-end.	All RF components with artworks can be passed to PCB Editor as components and they can be modified and be back-annotated to DE-HDL. This helps in keeping the front-end and back-end in sync.
Library	Does not understand parameters for ADS RF components	An RF-PCB library is used and RF-PCB defined properties are used to pass parameters for ADS-compatible parameterized components.
Library Translator	Requires RF-PCB Library Translator	Does not depend on RF-PCB Library Translator
Running the script file	Need to start Design Entry HDL and manually run the script	Launches Design Entry HDL automatically if it is not running, and the script is run on the background.

Feature	Standard IFF Import	RF-PCB IFF Import
Substrates and other simulation only components	Substrates are imported as regular components	Importer extracts information of substrates from the IFF file, but substrates and other simulation only components won't present on schematic
Variable/expression and hierarchical block	Variables are imported as regular components, importer doesn't understand the variables	Supports variable/expression and hierarchical block.

A

RF-PCB IFF Import User Interface

Launching RF-PCB IFF Import

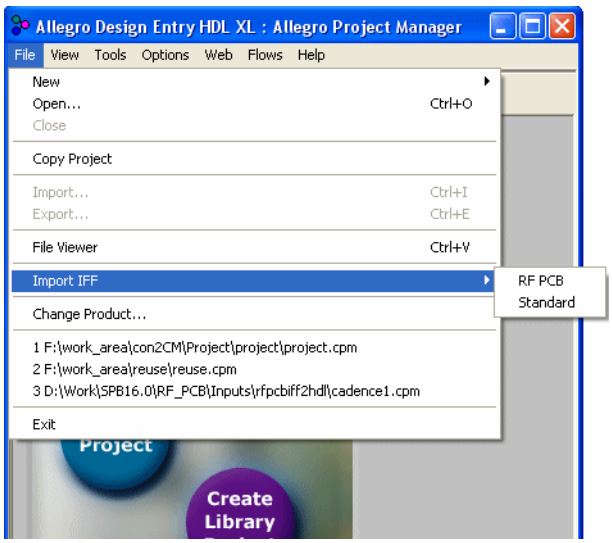
You can start RF-PCB IFF Import in any of the following ways:

- Run `rfpcbiff2hdl.exe`.

  See "Running RF-PCB IFF Import from the Command Line" on page 40 for details.

- Choose *RF-PCB – Import IFF* in Design Entry HDL.

- Choose *File – Import IFF – RF PCB* from Project Manager.



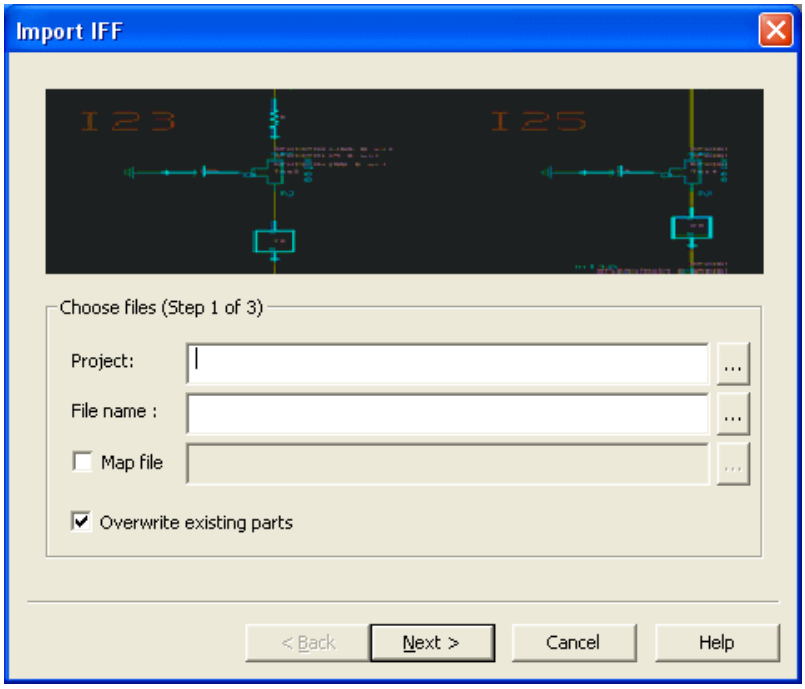
The RF-PCB IFF Import interface has the following steps:

- Step 1 *Choose files*
- Step 2 *Symbols Mapping*

- Step 3 *Create Symbols and Import Schematic Design*

See "RF-PCB IFF Import Overview" on page 11 for more information on how the importer works.

Choose files



In this screen you specify the following information:

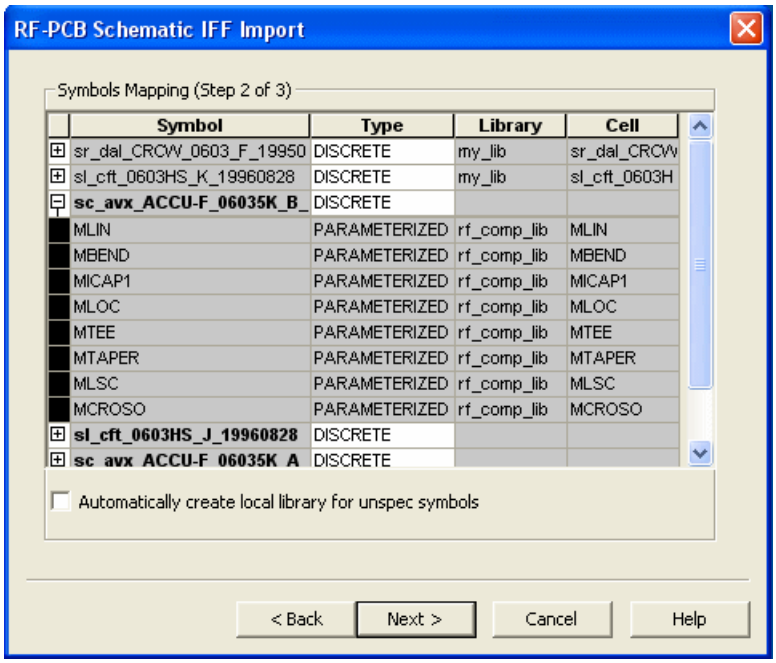
Option	Description
Project	The full name of project into which the RF design needs to be added. If you launch RF-PCB IFF Import from Design Entry HDL, this field shows the name of the currently open project and this field remains disabled. Similarly, if you open a project (.cpm) in Project Manager, and then launch RF-PCB IFF Import, the field is populated with the currently open project and remains disabled. If you launch RF-PCB from the command line or from Project Manager with no project file currently open, you need to specify the name of the project into which you plan to import an IFF file. To specify the cpm file, follow these steps: **1.** Click the browse button (...) to bring up a file browser with the file type set to project files, *.cpm. **2.** Navigate to the folder containing the required cpm file. **3.** Select the cpm file. **4.** Click *Open*.
Filename	The full name of the iff file to be imported. In case only a filename is specified the file will be searched for in the current working directory. To specify the IFF file, follow these steps: **1.** Click the browse button (...) to bring up a file browser with extension set to *.iff. **2.** Navigate to the folder containing the required iff file. **3.** Select the iff file. **4.** Click *Open*.

Option	Description
Map file	When you run the RF-PCB Import once for a project, a map file is created (`symmap.dat`). To use the same mapping information, you can direct RF-PCB IFF Import to use the map file. In this way you can avoid doing the mapping again in step 2. This is recommended when a design is imported with only slight changes.To use a map file, follow these steps: **1.** Select the *Map file* check box. **2.** Click the browse button (...) to navigate to the folder containing the map file. **3.** Select the map file. **4.** Click *Open*. To create a new map file or modify the mapping information, ensure the Map file check box is not selected. The text field and browse button for the map file are available only when the Map file check box is selected. See "Symbols Mapping Overview" on page 12 for details of the map file.
Overwrite existing parts	Select this check box to direct RF-PCB IFF Import to overwrite parts in the specified library during the import process. By default, the overwriting behavior is disabled.
Next	Parse the specified IFF file and move to the next step of the import.

When you click *Next*, RF-PCB IFF Import reads the IFF file and collects all the symbol definitions and component instances in the IFF file. If the IFF file is invalid, the importer moves to step 3 with an error message and then exit. Otherwise, if there are symbols that need mapping to local libraries, then move on to Symbol Mapping step or move on the step 3 to import design directly.

If you specify a map file, the mapping relationship described in the map file is used as initial mapping for the Symbol Mapping step. If the specified map file is invalid or inconsistent with the IFF file, it will just be ignored with a warning message.

Symbols Mapping



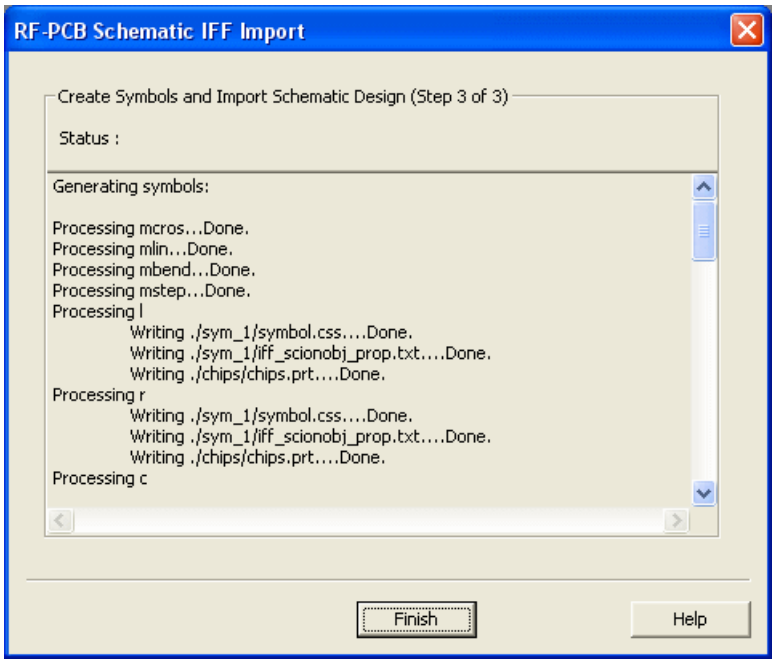
RF-PCB IFF Import reads the IFF file and collects all the symbol definitions and component instances in the IFF file. If there are symbols/component instances that need mapping, which happens when there is no initial symbol map file or the mapping relationship for the symbols is not defined in the map file yet, the importer traverses through all the local libraries specified in `cds.lib`, and classifies all the symbols.

The following information is displayed in this screen:

Option	Description
Symbol Mapping	The following information for each component is displayed: *Symbol* The name of the symbol.

Option	Description	
	Type	The type of the symbol. Each symbol is assigned an initial value according to the information from the IFF file. You can change it to another value from the list based on what best fits the symbol. The possible values for this field are: ■ Parameterized ■ Discrete ■ Simulation only ■ IC ■ IO ■ Power See "Symbols Mapping Overview" on page 12 for more information about the symbol types.
	Library	The name of the library the symbol is mapped to.
	Cell	The name of the cell the symbol is mapped to.
Automatically create local library symbols for unspec symbols	This option specifies if warning messages should be displayed for unrecognized symbols. By default, this check box is selected, which means that RF-PCB IFF Import automatically creates a local library for unspec symbols without prompting a warning message for the unrecognized symbols. However, if there is any unrecognized symbol has no symbol page definition in the IFF file, no local library cell will be created. If you do not select the *Automatically create local library for unspec symbols* option, the importer checks if there are any unspecified symbols when you click *Next*.	

Create Symbols and Import Schematic Design



This screen shows the status and progress of creating the symbols and importing the schematic design.

For example, at the symbol generating step, for a discrete symbol, say capacitor CR_05_1996 in ADS, is mapped to a local library `cr` and cell `CAP`, a new library cell named as `CR_05_1996` is created. The chips, symbol and part-table views of `CAP` are copied to the cell `CR_05_1996` as well.

While for a symbol that has no counterpart in the local library, a new library cell is mapped, and chips, symbol and part-table views are generated according to the symbol definition in the IFF file. When an instance of `CR_05_1996` is being imported, with the `00NFLJB` part-number, the Importer looks up the part-table. If part-number `100NFLJB` does not exist yet, a new row will be inserted into the part-table.

Running RF-PCB IFF Import from the Command Line

The complete syntax of the command is:

```
rfpcbiff2hdl [-proj proj_file] [-product license_string] [-iff iff_file] [-lm
layermap_file] [-mpssession session_name] [-sm symmbolmap_file] [-overwrite]
```

where

`[-proj project_file]`	The full name of project into which the RF design needs to be added.
`[-product <license_string>]`	The license to check out. RF PCB IFF Import is available only with the following: ■ Concept_HDL_expert ■ Allegro_Design_Editor_620 ■ PCB_librarian_expert ■ PCB_design_expert ■ SPECCTRAQuest_EE
`[-iff iff_file]`	The full name of the `iff` file to be imported. In case only a filename is specified the file will be searched for in the current working directory.
`[-lm layermap_file]`	The name of the layer map file created at the back-end.
`[-mpssession session_name]`	The name of the session.
`[-sm <symbolmap_file>]`	The name of the map file (`symmap.dat`).
`[-overwrite]`	Specify this to direct RF-PCB IFF Import to overwrite parts in the specified library during the import process. By default, the overwriting behavior is disabled.

Index

Symbols

A

B

C

D

F

H

I

J

K

L

M

O

P

R

S

T

U

V