

PSpice Library Browser Customization

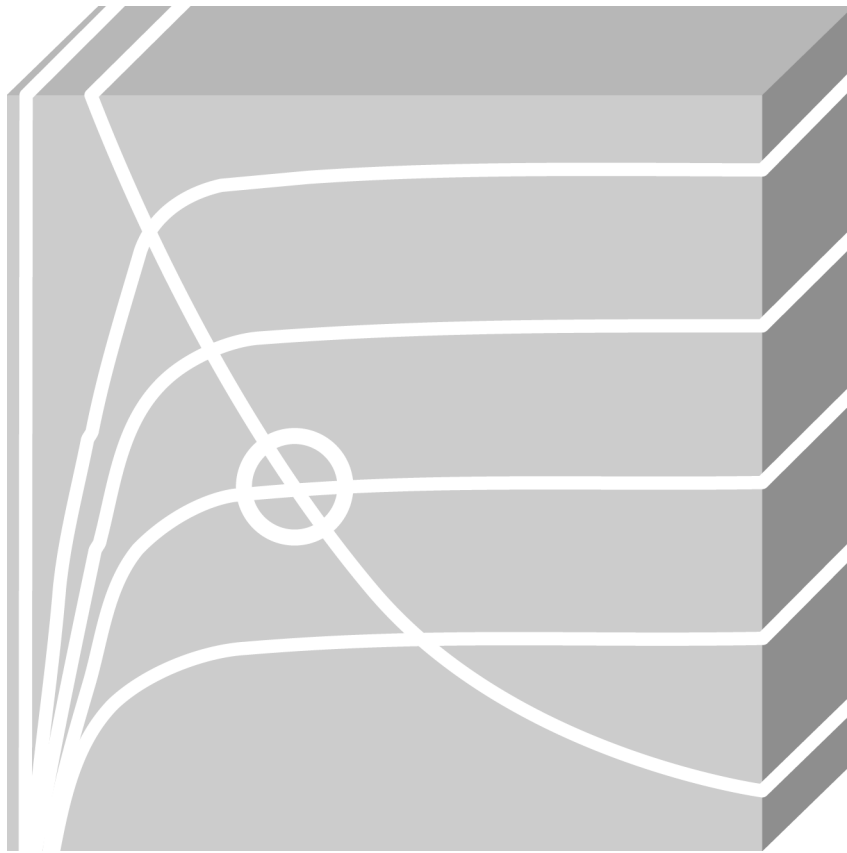


Table of Contents

- 1 Introduction 3
- 2 Location of the Database 3
- 3 PSpice Library Browser Customization..... 5
- 4 PSpice Library Browser Functionalities 10
- 5 Other Database 12
- 6 Modifications for SPB 17.2 12
- 6.1 Overview 12
- 6.2 How to Define the ODBC for the PSpice Library Browser DB 12
- 6.3 How to Customize PSpice Library Browser 13

1 Introduction

This document explains:

- How to change the default location of the Database
- How to customize the default Database including new components and new categories
- Extra Functionalities in PSpice Library Browser
- PSpice Library Browser features in the Release 17.2

The steps to be followed to customize PSpice library Browser in SPB 16.6 and SPB 17.2 are almost identical. The only difference is that release 17.2 is using 64bit architecture instead of 32bit, that is why the configuration of the ODBC Driver will be different (you will find the steps to define this ODBC in point 7). The rest of the points are the same for both options.

PSpice Library Browser is a container where it is possible to organize and classify into categories or libraries the components that can be used in PSpice simulations in order to make easier the search and use of these components in your designs. This option allows to compare properties of an amount of components in a glance. Moreover, it is possible to have all the components stored in a centralized server allowing to all local users to access to this data globally. The components included in this PSpice Library Browser at the beginning are the components that Cadence delivers by default.

2 Location of the Database

The default Cadence Database for PSpice Library Browser is located in

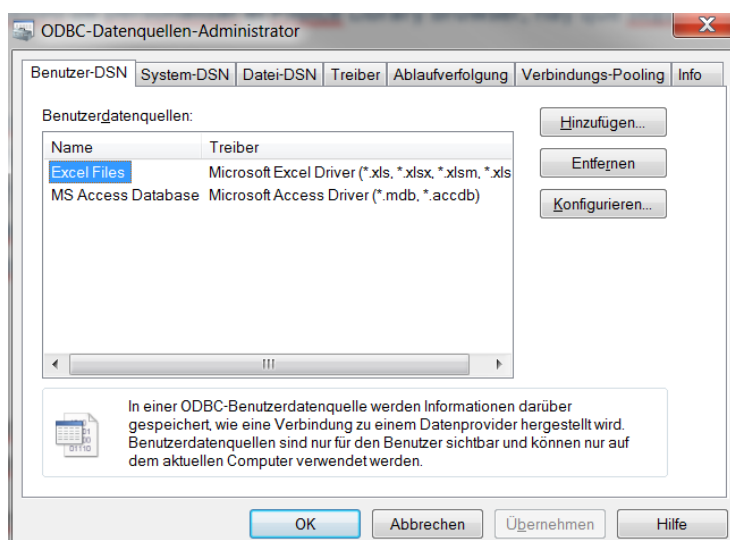
`%CDSROOT%/tools/capture/tclscripts/orPspiceModelBrowser`

If you want to use this Database in a server (globally), you have to indicate to OrCAD where PSpice Library Browser has to look for the Database. For that, follow the next steps.

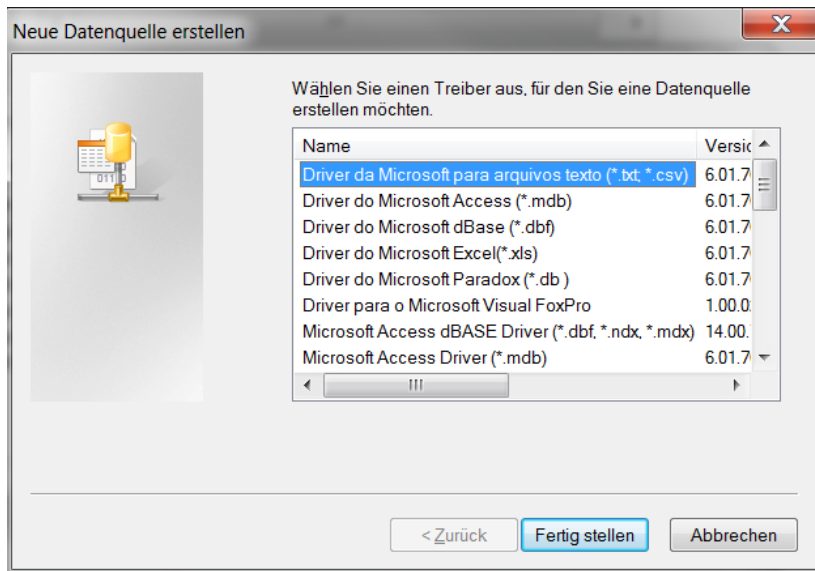
Note

This process is only for SPB 166. If you want to define it for SPB 172, go to point 7.

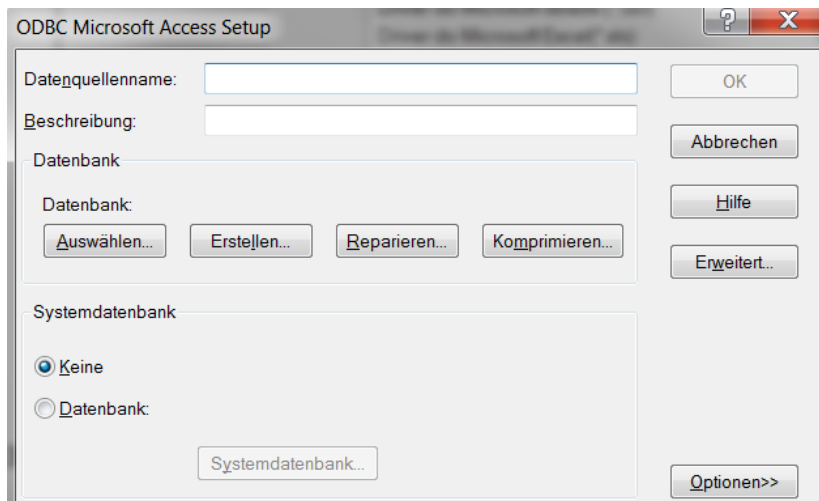
1. Go to the directory `C:\Windows\SysWOW64` and make double click on **odbcad32.exe**. A window will appear:



2. Select the tab System-DSN and click on **Hinzufügen (Add)**. Then select what kind of Database you want to use. In this case, it is a MS Access Database, that is why Microsoft Access Driver (*.mdb) is selected:



3. Write the name of the Data Source (for example TEST), and click on **Auswählen (select)** to choose your Database. Then click OK.



4. Once this process is done, you have to indicate to the software, where the Database is located and where the Cadence components and/or your components will be included (for example in the case you are using centralized server with all the components for your developers). For that, click on **Start > Computer > Properties > Advanced Properties > Environment Variables** and in System Variables, this variable must be defined:

Variable Name: CDN_PSPICE_OBDC_SRC

Value: TEST

Click **OK** until everything is closed. Up this moment, PSpice Library Browser will read the Database located in the directory you have specified.

3 PSpice Library Browser Customization

Each company has its own components organization. It might be that a particular company only uses components defined by default in the Cadence libraries, companies which only use components defined by themselves and in most of the cases companies which use not only components defined by Cadence but also components defined by themselves (found in Internet, delivered by manufacturers, etc.). In case you want to use components that are not included in the default Cadence libraries, you have to introduce them manually in the Database **cdn_osp_db.mdb** for SPB 166 or in **cdn_osp_db-db** for SPB 172. Both databases are structured identically. The only difference is that the first one is supported by Microsoft Access DB and the second one by SQLite DB.

This database is structured with the next tables and parameters inside of each of them:

- Category
 - Company_Id
 - Category_Id
 - Category_Name
 - Category_Description

- Category_part_link
 - Company_Id
 - Category_Id
 - Part_Number

- Footprint
 - Company_Id
 - Footprint_Name
 - Footprint_Pins
 - Footprint_PackageType

- Model
 - Company_Id
 - Model_Number
 - Model:Name
 - Model_Library
 - Model_Description
 - Model_LastUpdatedTime
 - Mfr_Name

- Part
 - Company_Id
 - Part_Number
 - Part_Name

- Part_Generic
- Part_Description
- Part_Library
- Mfr_Name
- Part_footprint_link
 - Company_Id
 - Part_Number
 - Footprint_Name
 - Default
- Part_model_link
 - Company_Id
 - Part_Number
 - Model_Number
- Part_symbol_link
 - Company_Id
 - Part_Number
 - Symbol_Number
- Property
 - Company_Id
 - Property_Name
 - Property_Value
 - Property_Value_Type
 - Object_Id
 - Object_Type
- Pspicetemplate_link
 - Company_Id
 - Symbol_Number
 - Model_Number
 - PSpiceTemplate
- Symbol
 - Company_Id
 - Symbol_Number
 - Symbol_Name
 - Symbol_Library
 - Symbol_Section
 - Symbol_View
 - PSpice Template

Note

Currently only the tables **Category**, **Category_Part_Link** and **Part** are connected with PSpice Library Browser. The other tables and their parameters are included to add functionalities in the future.

The question now is: **How can I add customized components in the Database to visualize them in the PSpice Library Browser?**

To add customized components to the PSpice Library Browser, follow the next steps with a description of each of them and a screenshot of an example using a customized component with a Part Number FC-RES-0001.

1. Add a part using part table:

Field Name	Data Type
Company_ID	Text (up to 16 char)
Part_Number	Text (up to 64 char)
Part_Name	Text (up to 64 char)
Part_Description	Text (up to 255 char)
Part_Library	Text (up to 64 char)

Note

- Part Number is the key in the part table.
- Part Name, Part Description, and Part Library are used by the application to search for a Part.

Example

Company_Id	Part_Number	Part_Name	Part_Generic	Part_Description	Part_Library	Mfr_Name
cadence	F78M12lopamp	F78M12	F78M12	3 Terminal Positive Voltage Regulator	opamp	
cadence	FA57SA50LCirf	FA57SA50LC	FA57SA50LC	57A, 500V, Single N-Channel HEXFET Power MOSFET	irf	International R
cadence	FB180SA10Iirf	FB180SA10	FB180SA10	180A, 100V, Single N-Channel HEXFET Power MOSFET	irf	International R
cadence	FC-RES-0001	RES_POTI_R	RES	TEST	FC_PASSIVE	FlowCAD
cadence	FCX458/ZTXIzetex	FCX458/ZTX	FCX458	225mA, 400V, NPN Bipolar Transistor Device	zetex	Zetex
cadence	FCX558/ZTXIzetex	FCX558/ZTX	FCX558	200mA, 400V, PNP Bipolar Transistor Device	zetex	Zetex
cadence	FCX591A/ZTXIzetex	FCX591A/ZTX		PNP Medium Power Bipolar Transistor	zetex	

2. Add a category using category table
If there is no related category of the new part already present in the database, you need to add a category. If you have the related category present in the database, you can skip this step.

Field Name	Data Type
Company_ID	Text (up to 16 char)
Category_ID	Number
Category_Parent ID	Number
Category_Name	Text (up to 128 char)
Category_Description	Text (up to 255 char)

Note

- Category ID is the key in the category table.
- Category ID, Category Parent ID, and Category Name are used by the application to search for a part.

Example

Company_Id	Category_Id	Category_Name	Category_Description
cadence	488	/null/Magnetics/Supermalloy Core/Toroid	<category desc>
cadence	489	/null/Magnetics/Ferrite Core/LP	<category desc>
cadence	491	/null/Magnetics/Magnesil Core/ETD	<category desc>
cadence	492	/null/ElectroMechanical/Sensors	<category desc>
cadence	493	/null/FlowCAD/Example	<category desc>
cadence	494	/null/ContiLib/Test	<category desc>
*	0		

3. Link the part to one or more categories using category_part_link table. In this step, you will link the new part with one or more categories.

Field Name	Data Type
Company_ID	Text (up to 16 char)
Category_ID	Number
Part_Number	Text (up to 64 char)

Note

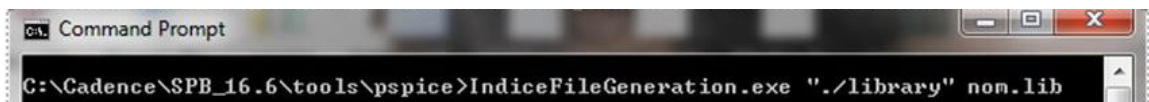
- Category ID must represent a category in the category table and Part Number must represent a part in the part table.
- Category ID and Part Number are the keys together in category_part_link table.
- Category ID and Part Number are used by the application to search for a part.
- A part can be linked to multiple categories.

Example

Company_Id	Category_Id	Part_Number	Zum Hinzufügen klicken
cadence	491	K44444EC!Cores	
cadence	491	K44949EC!Cores	
cadence	492	RESOLVER!anl_misc	
cadence	493	FC-RES-0001	
cadence	494	test	

Once the component is defined in the Database (at the moment the Company_Id cannot be changed), it is necessary to place the symbol (.olb) and the model library (.lib) in the proper directory:

- The model library has to be included in the directory where the PSpice libraries are located. Moreover, two additional steps must be followed:
 1. You need to add the libraries in nom.lib
 2. You need to regenerate the index for nom.lib. It can be done from command line which has written access on the drive, as shown below:



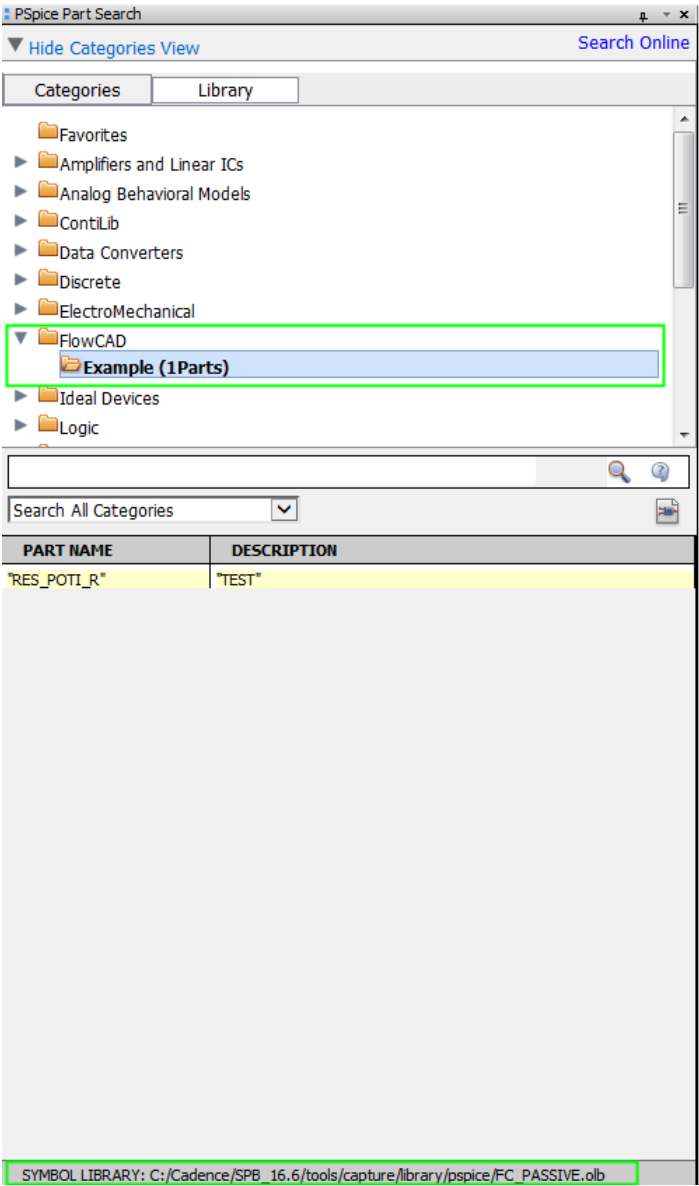
Note

If you do NOT include the lib in the nom.lib, you will have to add it in the simulation profile that you define for your design.

- The symbol library must be included well in the repository where Capture PSpice .olb files are available (local directory), or in a global directory modifying the next TCL function (this function is available in <install>\tools\capture\tclscripts\orPSpiceModelBrowser\tcl):

```
proc ::capPSpiceModelBrowser::SetIOlbPath { LibName } {
    set hier $::env(CDSROOT)
    set IOlbPath {}
    set libfolderPath {d:\olbpath1;d:\olbpath2;d:\olbpath3}
    set libfolderList [split $libfolderPath ";"]
    foreach IPath $libfolderList {
        if {[string first CDSROOT $IPath 0]==0} {
            set IPath [string replace $IPath 0 6 $hier]
        }
        set IOlbPath $IPath/$LibName.olb
        regsub -all "\\\" $IOlbPath "/" IOlbPath
        if {[file exists $IOlbPath] != 0 } {
            break
        }
    }
    if {[file exists $IOlbPath] == 0 } {
        set IOlbPath $LibName
    }
    return $IOlbPath
}
```

Finally, opening OrCAD and searching the defined component in PSpice Library Browser:



4 PSpice Library Browser Functionalities

- PSpice Part Search (Filter)
Learn to narrow your search criteria for better searches of parts.

Search Entry Example	To Search
optional amplifier	Parts with NAME or DESCRIPTION OR LIBRARY name containing “amplifier” and “operational”
name:AD-845	Parts with NAME containing “AD-485”
desc:amplifier	Parts with DESCRIPTION containing “amplifier”
lib:opamp	Parts with LIBRARY name containing “opamp”

name:AD-485 desc:amplifier operational	Parts with NAME containing "AD-845" AND DESCRIPTION containing "operational" AND "amplifier"
name:AD-845 lib:opamp	Parts with NAME containing "AD-845" AND LIBRARY name containing "opamp"
desc:amplifier lib:opamp	Parts with DESCRIPTION containing "amplifier" AND LIBRARY containing "opamp"
name:AD-845 des:rectifier amplifier lib:opamp	Parts with NAME containing "AD-845" AND DESCRIPTION containing "amplifier" AND "recitifier" AND LIBRARY name containing "opamp"

- Right Mouse Button (RMB) options

This option allows to add new operations when the RMB is clicked on the component that appear in the list of PSpice Library Browser. Some operations like opening a PDF of a particular component can be implemented. So that, first of all the menu tabs must be defined in the file Menu_Parts.txt and the operations for each tab must be defined in the file PSpiceModelBrowser.tcl found in

%CDSROOT%/tools/capture/tclscripts/orPSpiceModelBrowser/tcl.

- Code to define the tabs when clicking RMB on a component of the list is:

```
[
{"MenuName":"Custom Menu 1","Function":"Menu1Callback"},
{"MenuName":"Custom Menu 2","Function":"Menu2Callback"},
{"MenuName":"Custom Menu 3","Function":"Menu3Callback"}
]
```

- Code to define the operation to be realized when some of the tabs defined is clicked:

```
proc Menu1Callback {args} {
    puts $args
    puts Menu1Callback
}
proc Menu2Callback {args} {
    puts $args
    puts Menu2Callback
}
proc Menu3Callback {args} {
    puts $args
    puts Menu3Callback
}
```

5 Other Database

For those customers who do not use MS Access DB but another one like SQL or Oracle, it is important to remark that it is also possible to use these Databases. The issues that must be taken into account are:

- Vary the definition of the ODBC, where SQL or Oracle must be chosen instead of MS Access DB.
- Include the mandatory tables (Category, Category_Part_Link and Part) in the SQL or Oracle Database with the same structure.

6 Modifications for SPB 17.2

6.1 Overview

Up version 17.0 Cadence changed its SPB Tools from 32bit architecture to a 64bit architecture. Because of this change it is necessary to use 64bit ODBC Driver for the Database.

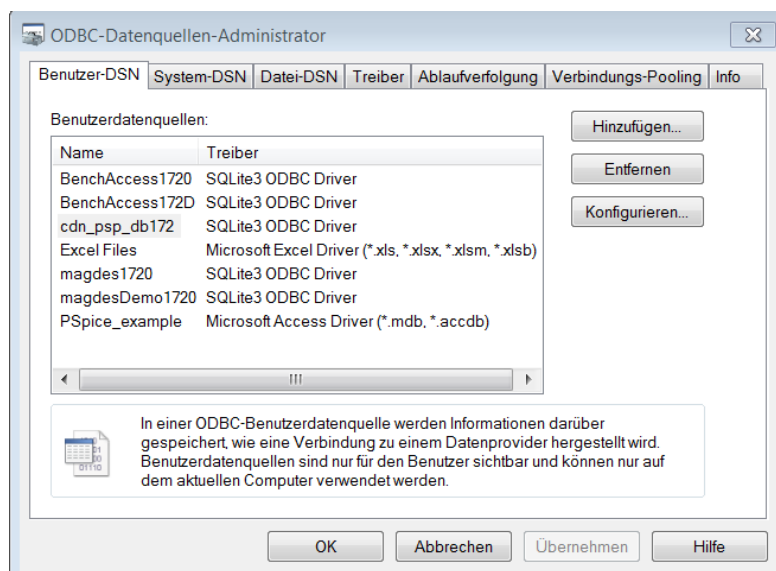
In order to work with a 64bit architecture Cadence made the decision to work with SQL Lite Database. To modify and customize the data in the same way it was explain in the previous points, you need to download SQLite (open source). The only difference is the connection of the ODBC Driver.

Note

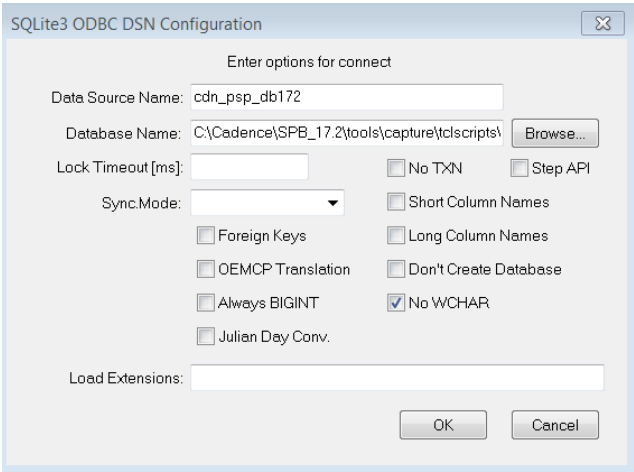
If you want to use your own DB which is in implemented using 32 bits and is also used for the CIS option, you have to convert this DB to 64bits. You can do it using the Application Note Access 32bit nach SQLite 64bit.

6.2 How to Define the ODBC for the PSpice Library Browser DB

To customize the location of the PSpice Library Browser DB it is no longer necessary to use the Environment Variable CDN_PSPICE_ODBC_SRC. The only thing you have to do is clicking on **System Control > Administration > Database (ODBC)**



and then make double click on `cdn_psp_db172` (default Cadence Database for PSpice Library Browser) and change the directory where you want to place it:



NOTE

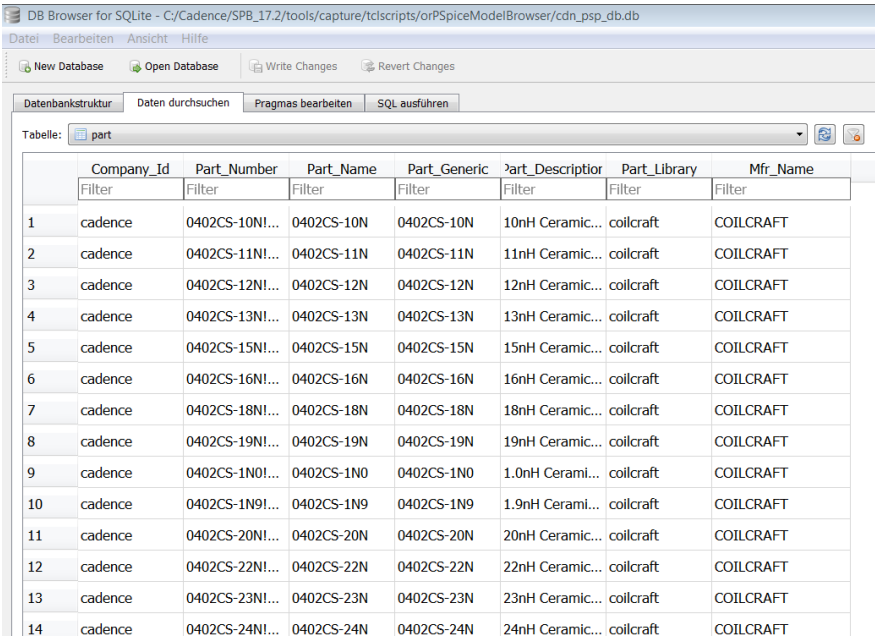
By default, this Database is called `cdn_psp_db172` and it is located in
`%CDSROOT%/tools/capture/tclscripts/orPSpiceModelBrowser/cdn_psp_db172`

Warning

If you want to use another Data Source Name, then you have to change the value of the environment variable `CDN_PSPICE_ODBC_SRC`.

6.3 How to Customize PSpice Library Browser

In order to customize the PSpice Library Browser including new components and / or new categories, you have to modify the Database. For that you could use SQLite, as it is an Open Source Database System. It is very easy to install and to use. There you can modify the tables **Category**, **Category_Part_Link** and **Part**. Please follow the modules 4, 5 and 6.



	Company_Id	Part_Number	Part_Name	Part_Generic	Part_Descriptor	Part_Library	Mfr_Name
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	cadence	0402CS-10N1...	0402CS-10N	0402CS-10N	10nH Ceramic...	coilcraft	COILCRAFT
2	cadence	0402CS-11N1...	0402CS-11N	0402CS-11N	11nH Ceramic...	coilcraft	COILCRAFT
3	cadence	0402CS-12N1...	0402CS-12N	0402CS-12N	12nH Ceramic...	coilcraft	COILCRAFT
4	cadence	0402CS-13N1...	0402CS-13N	0402CS-13N	13nH Ceramic...	coilcraft	COILCRAFT
5	cadence	0402CS-15N1...	0402CS-15N	0402CS-15N	15nH Ceramic...	coilcraft	COILCRAFT
6	cadence	0402CS-16N1...	0402CS-16N	0402CS-16N	16nH Ceramic...	coilcraft	COILCRAFT
7	cadence	0402CS-18N1...	0402CS-18N	0402CS-18N	18nH Ceramic...	coilcraft	COILCRAFT
8	cadence	0402CS-19N1...	0402CS-19N	0402CS-19N	19nH Ceramic...	coilcraft	COILCRAFT
9	cadence	0402CS-1N01...	0402CS-1N0	0402CS-1N0	1.0nH Cerami...	coilcraft	COILCRAFT
10	cadence	0402CS-1N91...	0402CS-1N9	0402CS-1N9	1.9nH Cerami...	coilcraft	COILCRAFT
11	cadence	0402CS-20N1...	0402CS-20N	0402CS-20N	20nH Ceramic...	coilcraft	COILCRAFT
12	cadence	0402CS-22N1...	0402CS-22N	0402CS-22N	22nH Ceramic...	coilcraft	COILCRAFT
13	cadence	0402CS-23N1...	0402CS-23N	0402CS-23N	23nH Ceramic...	coilcraft	COILCRAFT
14	cadence	0402CS-24N1...	0402CS-24N	0402CS-24N	24nH Ceramic...	coilcraft	COILCRAFT