AC-NMR Documentation

Release 1.0 beta 2

Christian Meisenbichler

November 20, 2015

CONTENTS

AC-NMR is an NMR database application with the focus on frictionless NMR shift search with special regard to the interests of inorganic chemistry.

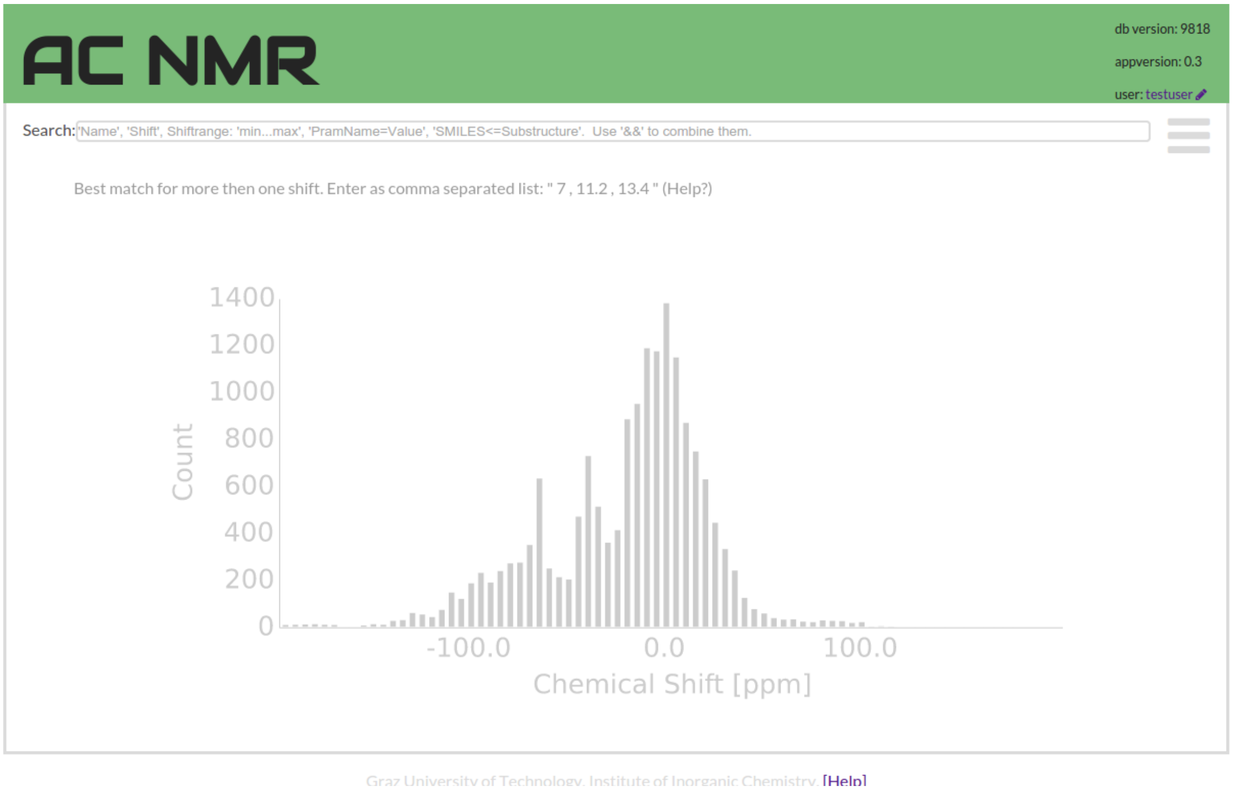
Back to the app.

Contents:

CHAPTER ONE

HOW TO SEARCH THE DB

The most important interface for all searching is the central search field which is almost always visible.



db version: 9818

appversion: 0.3

user: testuser

Search: 'Name', 'Shift', Shiftrange: 'min...max', 'PramName=Value', 'SMILES<=Substructure'. Use '&&' to combine them.

Best match for more then one shift. Enter as comma separated list: " 7 , 11.2 , 13.4 " (Help?)

Graz University of Technology, Institute of Inorganic Chemistry [Help]

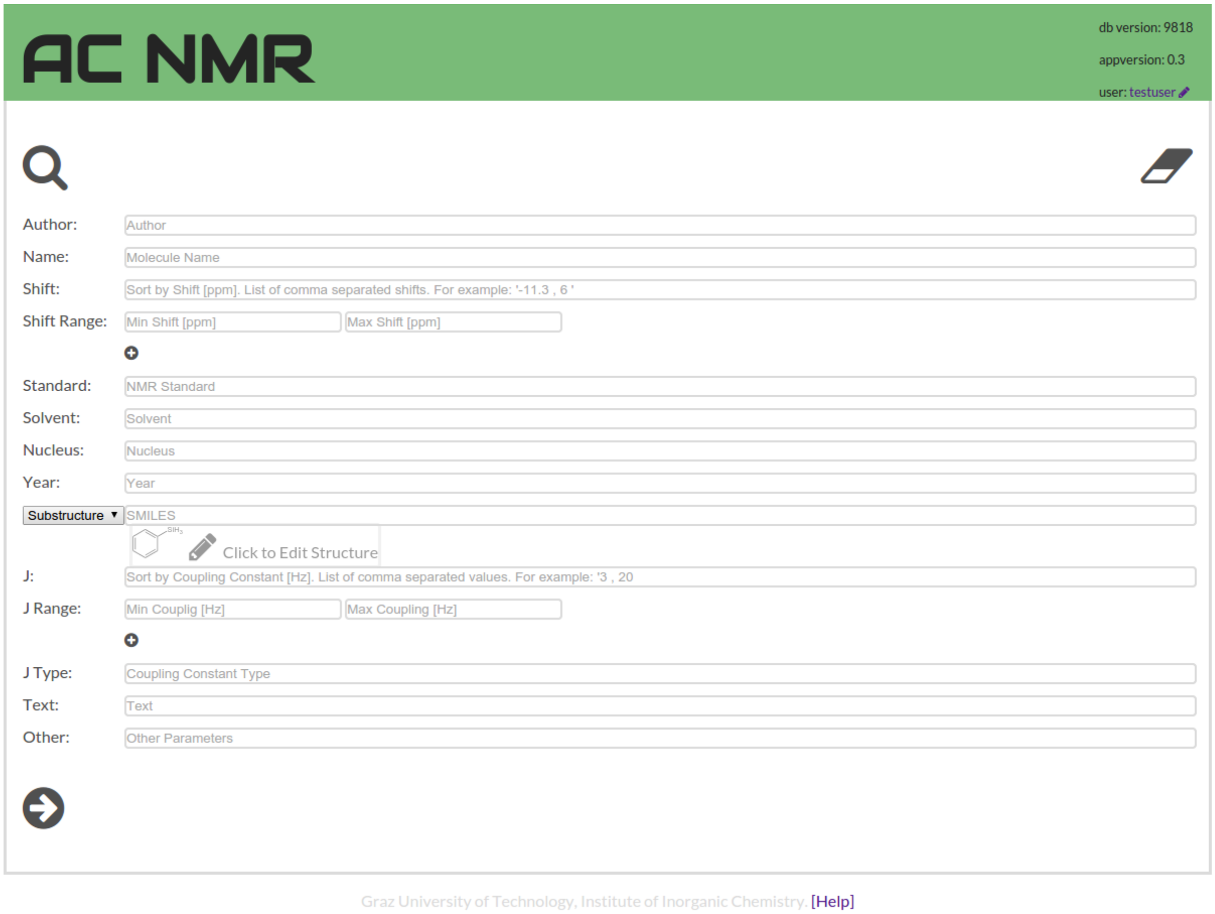
The search field tries to do the most intuitive thing for substring and shift searches. But also it provides a simple syntax to compose more complex query combinations.

If you enter a number value into the search field you find the compounds with a shift close to this number. If you enter a word or words the search performs a substring search.

This substring search is quite stubborn as it can only find exact matches. In order to help the user entering a successful query the input field will complete user input by offering suitable words that occur in the data. This substring search, searches compound names, publication authors and titles, also InChI codes or data-document IDs.

1.1 Detailed Search

The other user interface element designed to assist with search is the search detail form. It is accessible through the symbol with the three horizontal bars next to the search field.



db version: 9818

appversion: 0.3

user: testuser

Author:	Author
Name:	Molecule Name
Shift:	Sort by Shift [ppm]. List of comma separated shifts. For example: '-11.3 , 6 '
Shift Range:	Min Shift [ppm] Max Shift [ppm]
Standard:	NMR Standard
Solvent:	Solvent
Nucleus:	Nucleus
Year:	Year
Substructure ▾	SMILES
	Click to Edit Structure
J:	Sort by Coupling Constant [Hz]. List of comma separated values. For example: '3 , 20
J Range:	Min Couplig [Hz] Max Coupling [Hz]
J Type:	Coupling Constant Type
Text:	Text
Other:	Other Parameters

Graz University of Technology, Institute of Inorganic Chemistry. [Help]

If you already had a query in the search field, the input is not erased but decomposed into the search components and the form is filled with the values. In the other direction, if you have entered a set of parameters into the detailed search form, and you proceed to the searching, the parameters get concatenated into a single query string. Which reflects one of the design principles of the ACNMR data base that the query producing the results is always visible, and part of the URL.

In this form you find a button to edit a query structure for substructure or similarity search.

1.2 Combined Search

For combining two or more conditions the "&&" operator may be used to do this.

```
-2.3 ... -1.2 && 10.0 ... 13.4
```

This combines two shift-range searches, such that it returns only compounds that have shifts in both ranges.

You can combine shift searches like that:

```
7 && 11 && 14
```

This will sort the results by the sum of the smallest distances. It will however, use all available shift values of the entry, also across nucleus and solvents. It is therefore possible to refine the results by adding more constraints:

```
-121.3 && 4 && nucleus=Si
```

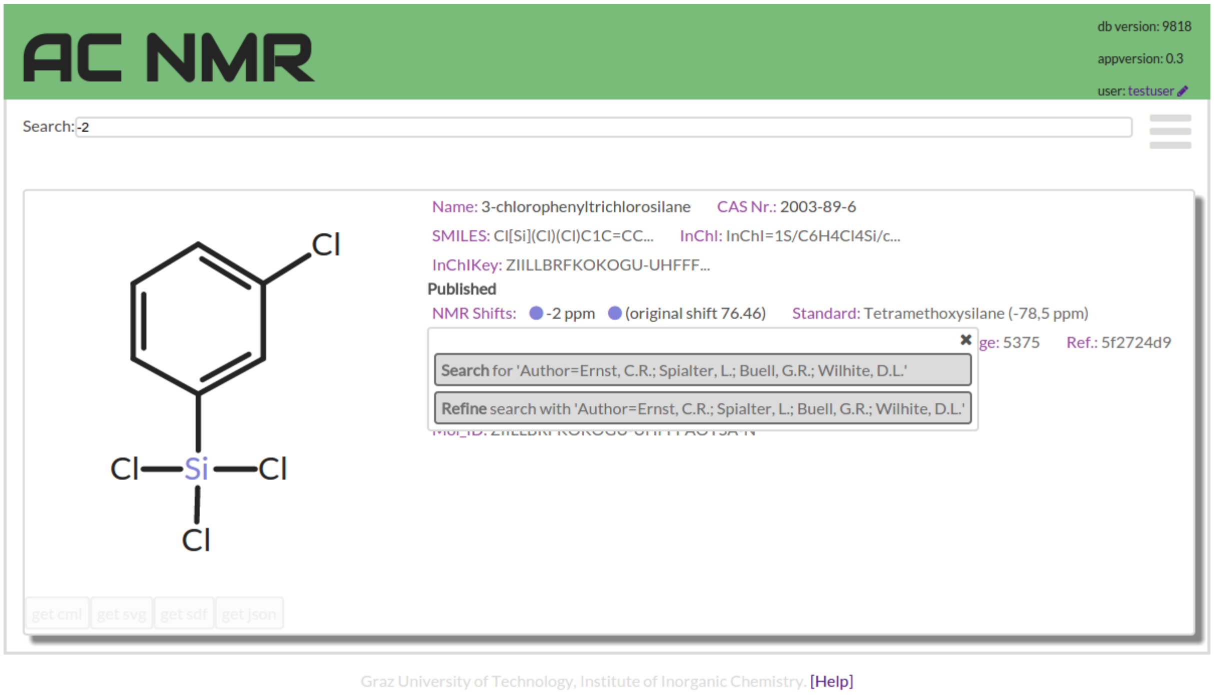
As an alternative syntax for shift sorting, it is possible to enter a list of shifts as comma separated list:

```
7.0, 11.2,  14.0
```

1.3 Parameter Context Menu

For refining the search by specifying additional conditions the parameter context menu is useful for exploring and narrowing the search.

In the panel displaying the molecule and shifts, most parameters have a context menu, which is accessible by double click:



AC NMR

db version: 9818

appversion: 0.3

user: testuser

Search: -2

Name: 3-chlorophenyltrichlorosilane CAS Nr.: 2003-89-6

SMILES: Cl[Si](Cl)(Cl)C1C=CC... InChI: InChI=1S/C6H4Cl4Si/c...

InChIKey: ZIILLBRFKOKOGU-UHFFF...

Published

NMR Shifts: ● -2 ppm ● (original shift 76.46) Standard: Tetramethoxysilane (-78,5 ppm)

ge: 5375 Ref.: 5f2724d9

Search for 'Author=Ernst, C.R.; Spialter, L.; Buell, G.R.; Wilhite, D.L.'

Refine search with 'Author=Ernst, C.R.; Spialter, L.; Buell, G.R.; Wilhite, D.L.'

Mol_ID: ZIILLBRFKOKOGU-UHFFFAOYSA-N

Graz University of Technology, Institute of Inorganic Chemistry. [Help]

This feature is intended to help you compile and discover queries.

1.4 Queryparameters

The parameter name are case insensitive and there are a few aliases for easier use.

The special value “undefined” lets you search for fields that are not defined yet. Example: doi=undefined

1.4.1 Mol and Literature

author= Substring in literature author field. Example: author=Marsmann

boilingpoint= Substring in boiling point field. Example: boilingpoint=20

CAS_RN= Substring in CAS_RN. Example: CAS_RN=1206-46-8

Comment= Substring in comment. Example: comment=Si assignment

DOCTYPE= Substring in literature type like journal, book, article. Example: DOCTYPE=article

DOI= Substring in literature DOI. Example: DOI=http://dx.doi.org/10.1016/s0022-328x(00)82875-2

Editor= Substring in literature editor. Example: Editor=Rappoport

FORMULA= Substring in sum formular. Example: FORMULA=C74

InChI= Substring in International Chemical Indentifier. Example: inchi=C74

InChI<=valid inchi> Substring search. Example: InChI<=InChI=1S/C5H12Si/c1-5-6(2,3)4/h5H,1H2,2-4H3

InChI~=valid inchi> Similarity search. Example: InChI~=InChI=1S/C5H12Si/c1-5-6(2,3)4/h5H,1H2,2-4H3

InChIKey= Substring. Example: InChIKey=GCSJLQSCSDMKTP-UHFFFAOYSA-N

Keyword Substring in literature key words. Example: keyword=silanes

MSPeakListPeaks= Substring. Example: MSPeakListPeaks=1531

MSPeakListSourceID Substring. Example: MSPeakListSourceID=EI MS

MeltingPoint= Substring. Example: MeltingPoint=20

Mol_ID= exact Mol id. Example: mol_id=DZNQKXKZINSVPH-UHFFFAOYSA-N

Mol_Name= Substring in Compound Name. Example: Mol_Name=cyclopentadienyl

Page= Substring literature page. Example: Page=205

REFNO= Substring in literature ref ID. Example: REFNO=f92be0

SMILES= Substring Example: smiles=C([SiH3])=C

SMILES<=<SMILES> Substructure search. Example: smiles<=C([SiH3])=C

SMILES~=<SMILES> Similarity Search. Example:smiles~=C([SiH3])=C

Source= Substring in Source. eg. journal name. Example: Source=Organometallics

Stereoinformation=true Marker for stereo Molecules. (Debugging help) Example: Stereoinformation=true

Title Substring literature title. Example: title=study

Volume= Substring literature volume. Example: Volume=42

YEAR= Order by distance to year published. Example: year=2004

date= Order by last edited close to date. Give as ISO date string. Example: date=2007-04-05T14:30

error= Substring error messages. Example: Error=smiles

owner Substring last user editing. Example: owner=me

1.4.2 NMR Parameters

Shift=<number> Sort by NMR Shift values close to number. Example: shift=0.9

Shift=<number1>...<number2> Shifts in range. Example: shift=0.9...0.95

Nucleus= Substring in NMR Nucleus field. Example: nucleus=29Si

Method= Substring in Method. Example: method=INDOR

J=<number> Sort by Coubling constant closest to number. Example: j=5.4

J=<number1>...<number2> Coupling Constant in range. Example: j=5.4...5.5

jtype= Substring Coupling type. Example: jtype= 2J(Si-P)

Temp= Substring in NMR Sample Temperature. Example: temp=-20

1.5 Design Principles

The single most important principle, the application tries hard not to break, is that any resource, that can be displayed, also has an URL that contains all the information required to display it again.

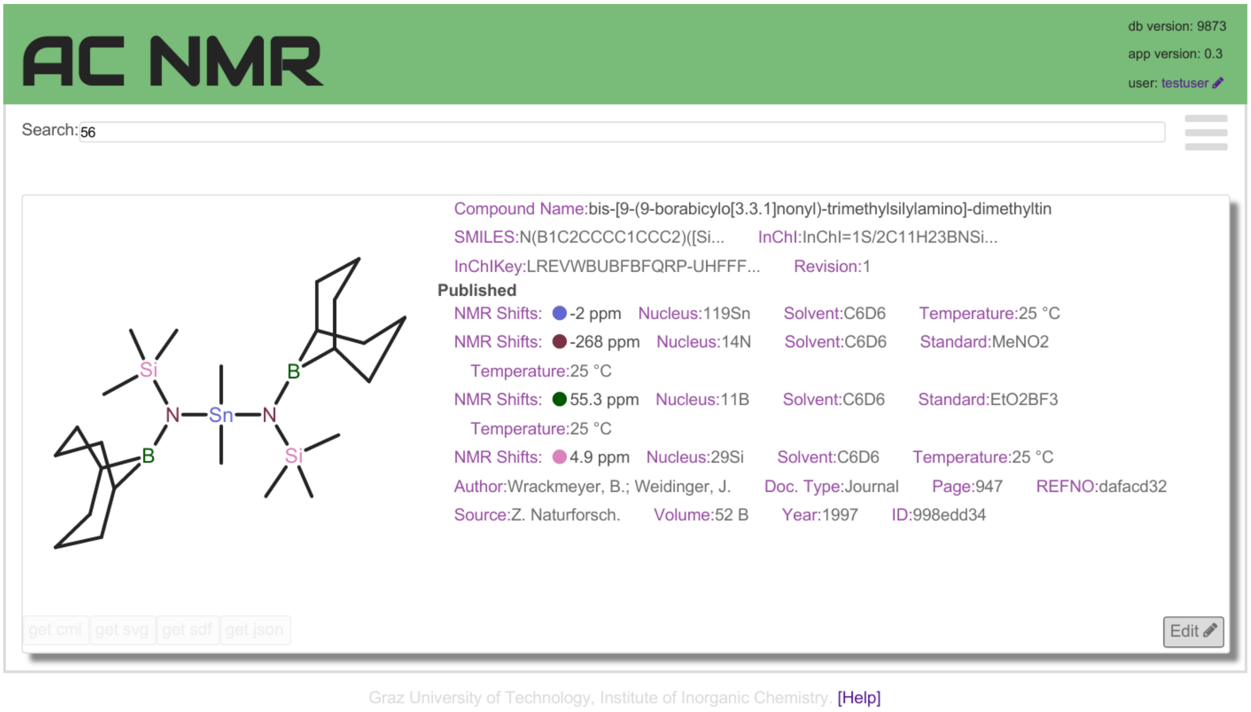
Also the search field is used to display the query, that led to the results. This is for example the main reason the system uses SMARTS for substructure search. Because, SMARTS encodes a chemical structure in a short string that can be part of the URL, and thus can be shown in the search field.

CHAPTER

TWO

HOW TO EDIT ENTRIES

Users with editor privilledges can edit datasets. To do this you may create a new dataset from the user profile. page or find the edit button on the displayed molecule. This will open the editor form.



AC NMR

db version: 9873

app version: 0.3

user: testuser

Search: 56

Compound Name: bis-[9-(9-borabicylo[3.3.1]nonyl)-trimethylsilylamino]-dimethyltin

SMILES: N(B1C2CCCC1CCC2)([Si...   InChI: InChI=1S/2C11H23BNSi...

InChIKey: LREVWBUBFBFQRP-UHFFF...   Revision: 1

Published

NMR Shifts: ● -2 ppm   Nucleus: 119Sn   Solvent: C6D6   Temperature: 25 °C

NMR Shifts: ● -268 ppm   Nucleus: 14N   Solvent: C6D6   Standard: MeNO2   Temperature: 25 °C

NMR Shifts: ● 55.3 ppm   Nucleus: 11B   Solvent: C6D6   Standard: EtO2BF3   Temperature: 25 °C

NMR Shifts: ● 4.9 ppm   Nucleus: 29Si   Solvent: C6D6   Temperature: 25 °C

Author: Wrackmeyer, B.; Weidinger, J.   Doc. Type: Journal   Page: 947   REFNO: dafacd32

Source: Z. Naturforsch.   Volume: 52 B   Year: 1997   ID: 998edd34

Graz University of Technology, Institute of Inorganic Chemistry. [Help]

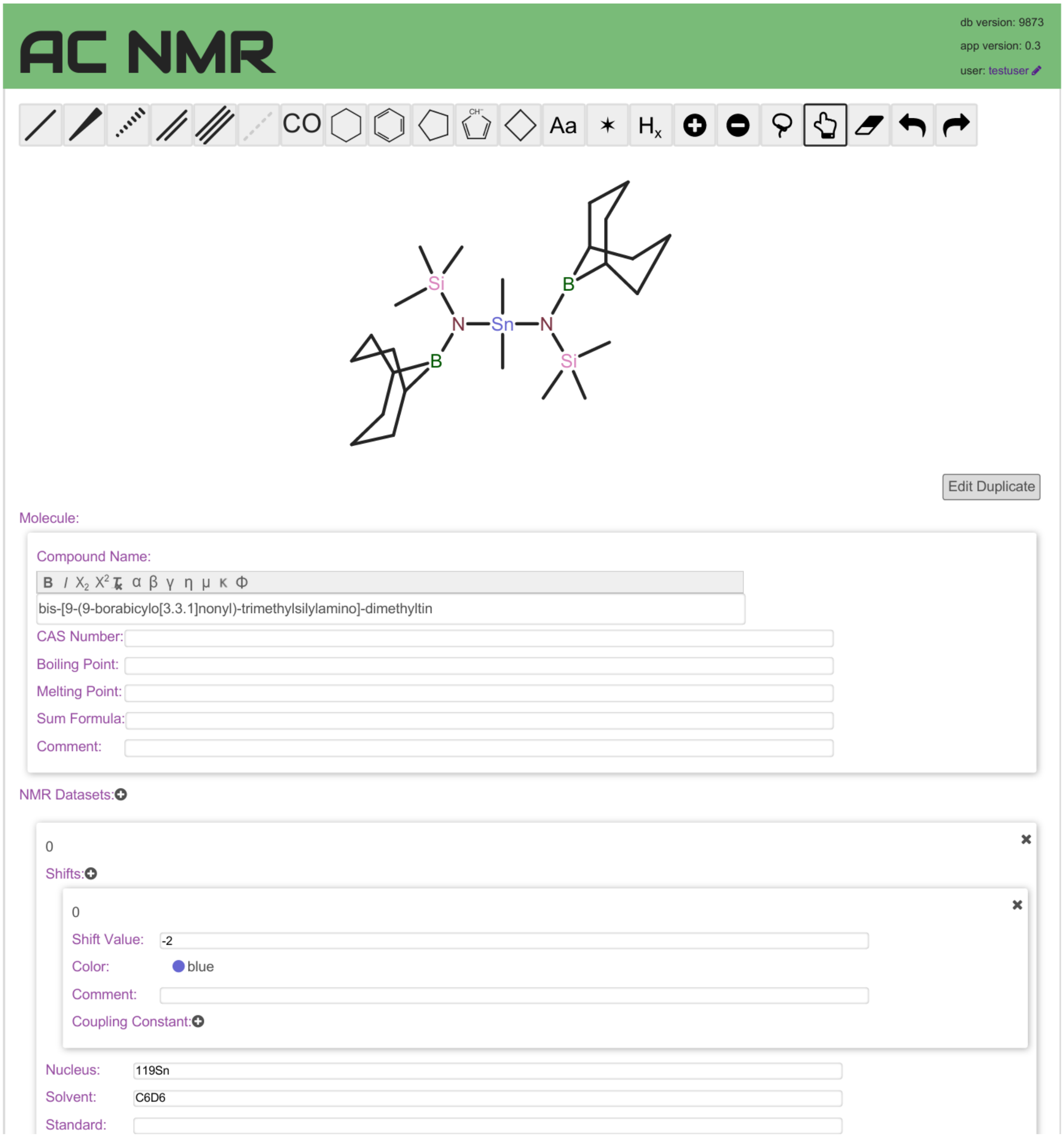
2.1 Editor Form

The editor form allows to edit all features in the database.

At the top is the *structure editor* to edit the molecular structure and colour code of the atom labels. Below there is a dynamic form which allows to add as many NMR schifts or couplings as needed.

The molecule name and coupling fields are rich text that allow bold, italic, super- and subscript formatting. Thera are also a few greek letters to chose from. If you need to use another greek or other letter just find the right unicode character and paste it in.

The most bottom part is the literature reference.



db version: 9873

app version: 0.3

user: testuser

Molecule:

Compound Name: bis-[9-(9-borabicylo[3.3.1]nonyl)-trimethylsilylamino]-dimethyltin

CAS Number:

Boiling Point:

Melting Point:

Sum Formula:

Comment:

NMR Datasets:

0

Shifts:

0

Shift Value: -2

Color: ● blue

Comment:

Coupling Constant:

Nucleus: 119Sn

Solvent: C6D6

Standard:

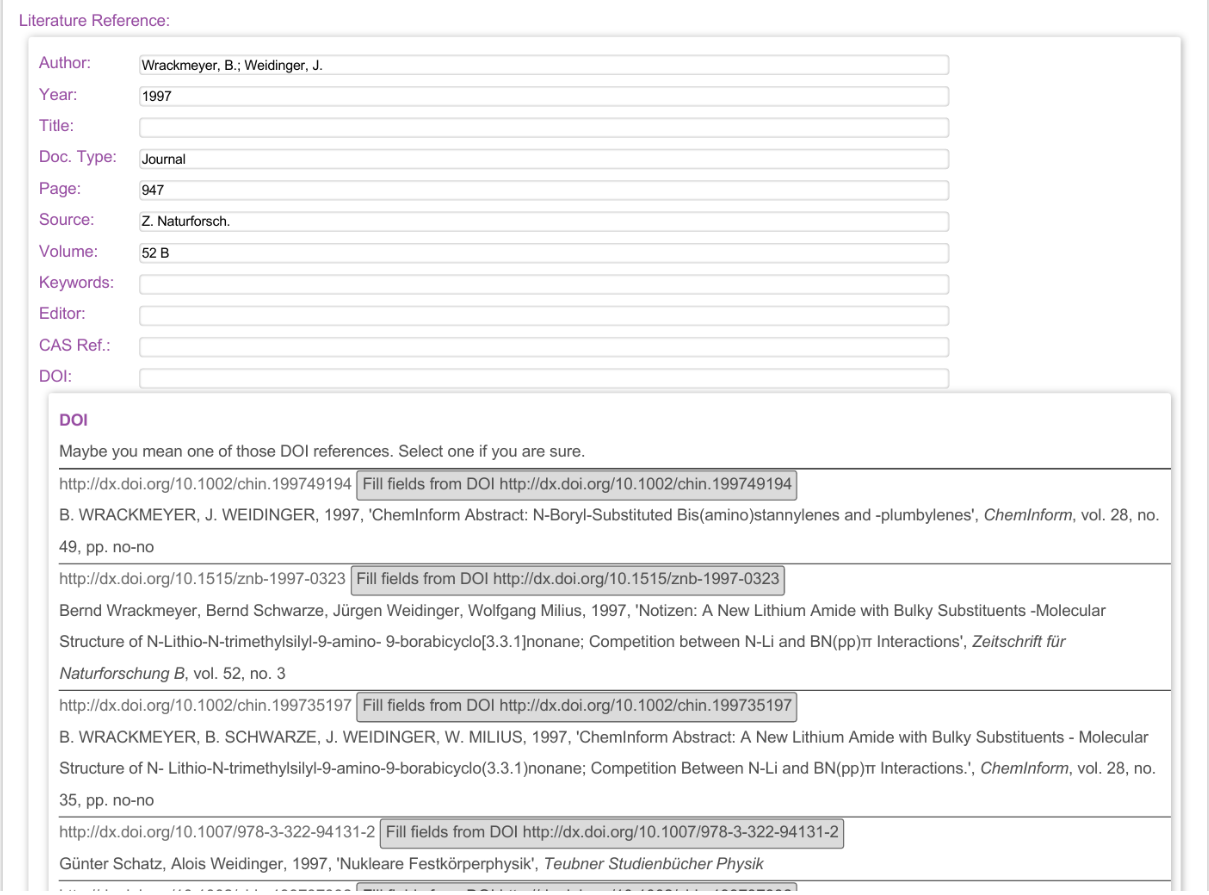
2.2 Create Duplicate

In many situations on might want to start from an available enry and create a new entry fom it, e.g. if you enter a list of compounds from the same literature source. For this purpose the editor has a button called "Edit Duplicate".

2.3 Literature Reference and DOI Editor

The literature reference editor form will look up DOIs on http://crossref.org when you change the Author, Year, Page, Source, Volume, or DOI field. Below the form you will find the hits which may help you to fill the rest of the fields.

As long as there is a value in the DOI field all the other values are ignored for the search. When you paste a DOI into the DOI field and leave the field, it will only use the new DOI to find the other metadata.



Literature Reference:

Author:	Wrackmeyer, B.; Weidinger, J.
Year:	1997
Title:	
Doc. Type:	Journal
Page:	947
Source:	Z. Naturforsch.
Volume:	52 B
Keywords:	
Editor:	
CAS Ref.:	
DOI:	

DOI

Maybe you mean one of those DOI references. Select one if you are sure.

http://dx.doi.org/10.1002/chin.199749194 Fill fields from DOI http://dx.doi.org/10.1002/chin.199749194

B. WRACKMEYER, J. WEIDINGER, 1997, 'ChemInform Abstract: N-Boryl-Substituted Bis(amino)stannylenes and -plumbylenes', *ChemInform*, vol. 28, no. 49, pp. no-no

http://dx.doi.org/10.1515/znb-1997-0323 Fill fields from DOI http://dx.doi.org/10.1515/znb-1997-0323

Bernd Wrackmeyer, Bernd Schwarze, Jürgen Weidinger, Wolfgang Milius, 1997, 'Notizen: A New Lithium Amide with Bulky Substituents -Molecular Structure of N-Lithio-N-trimethylsilyl-9-amino- 9-borabicyclo[3.3.1]nonane; Competition between N-Li and BN(pp)π Interactions', *Zeitschrift für Naturforschung B*, vol. 52, no. 3

http://dx.doi.org/10.1002/chin.199735197 Fill fields from DOI http://dx.doi.org/10.1002/chin.199735197

B. WRACKMEYER, B. SCHWARZE, J. WEIDINGER, W. MILIUS, 1997, 'ChemInform Abstract: A New Lithium Amide with Bulky Substituents - Molecular Structure of N- Lithio-N-trimethylsilyl-9-amino-9-borabicyclo(3.3.1)nonane; Competition Between N-Li and BN(pp)π Interactions.', *ChemInform*, vol. 28, no. 35, pp. no-no

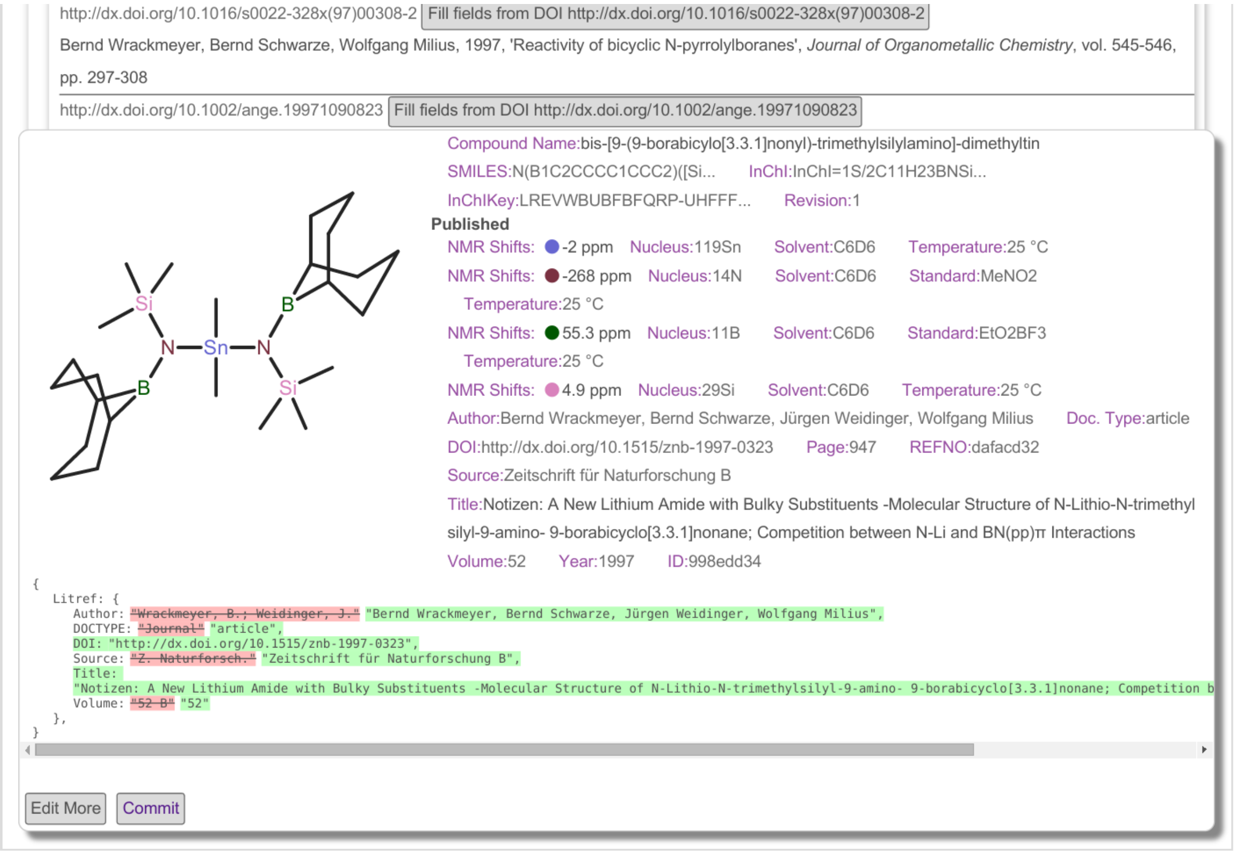
http://dx.doi.org/10.1007/978-3-322-94131-2 Fill fields from DOI http://dx.doi.org/10.1007/978-3-322-94131-2

Günter Schatz, Alois Weidinger, 1997, 'Nukleare Festkörperphysik', *Teubner Studienbücher Physik*

2.4 Preview, Commit or Delete

At the bottom of the page are two buttons one is to prewiew the data once more before you commit the changes to the database. And the other is to mark the entry as deleted, which will remove it from the searchresults but it will remain in the history.

The preview will display the data in the same way as in the search results and allows to review the changes. The next step is to commit the changes.


Si
N
Sn
N
B
B
Si
Fill fields from DOI http://dx.doi.org/10.1016/s0022-328x(97)00308-2
Fill fields from DOI http://dx.doi.org/10.1002/ange.19971090823
Edit More
Commit

http://dx.doi.org/10.1016/s0022-328x(97)00308-2

Bernd Wrackmeyer, Bernd Schwarze, Wolfgang Milius, 1997, 'Reactivity of bicyclic N-pyrrolylboranes', *Journal of Organometallic Chemistry*, vol. 545-546, pp. 297-308

http://dx.doi.org/10.1002/ange.19971090823

Compound Name:bis-[9-(9-borabicyclo[3.3.1]nonyl)-trimethylsilylamino]-dimethyltin

SMILES:N(B1C2CCCC1CCC2)([Si... InChI:InChI=1S/2C11H23BNSi...

InChIKey:LREVWBUBFBFQRP-UHFFF... Revision:1

Published

NMR Shifts: ● -2 ppm Nucleus:119Sn Solvent:C6D6 Temperature:25 °C

NMR Shifts: ● -268 ppm Nucleus:14N Solvent:C6D6 Standard:MeNO2 Temperature:25 °C

NMR Shifts: ● 55.3 ppm Nucleus:11B Solvent:C6D6 Standard:EtO2BF3 Temperature:25 °C

NMR Shifts: ● 4.9 ppm Nucleus:29Si Solvent:C6D6 Temperature:25 °C

Author:Bernd Wrackmeyer, Bernd Schwarze, Jürgen Weidinger, Wolfgang Milius Doc. Type:article

DOI:http://dx.doi.org/10.1515/znb-1997-0323 Page:947 REFNO:dafacd32

Source:Zeitschrift für Naturforschung B

Title:Notizen: A New Lithium Amide with Bulky Substituents -Molecular Structure of N-Lithio-N-trimethylsilyl-9-amino- 9-borabicyclo[3.3.1]nonane; Competition between N-Li and BN(pp)π Interactions

Volume:52 Year:1997 ID:998edd34

```
{
   Litref: {
      Author: ~~"Wrackmeyer, B.; Weidinger, J."~~ "Bernd Wrackmeyer, Bernd Schwarze, Jürgen Weidinger, Wolfgang Milius",
      DOCTYPE: ~~"Journal"~~ "article",
      DOI: "http://dx.doi.org/10.1515/znb-1997-0323",
      Source: ~~"Z. Naturforsch."~~ "Zeitschrift für Naturforschung B",
      Title:
      "Notizen: A New Lithium Amide with Bulky Substituents -Molecular Structure of N-Lithio-N-trimethylsilyl-9-amino- 9-borabicyclo[3.3.1]nonane; Competition b
      Volume: ~~"52 B"~~ "52"
   },
}
```

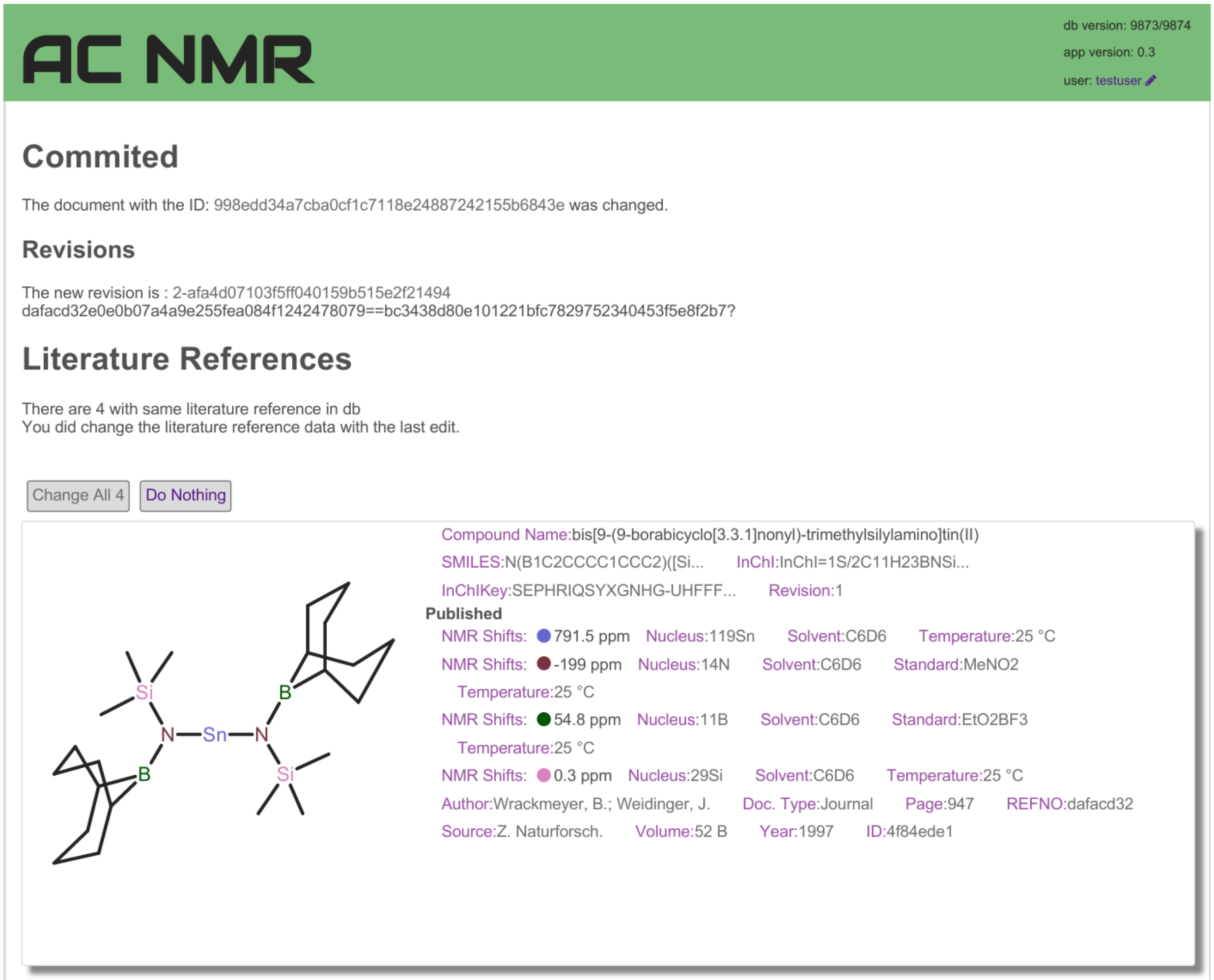
2.5 Commit Summary

After committing you will see a commit sumary with information on the revision and document ID. If you have changed the literature reference and there are other compounds in the database which had the same reference you may choose to change the others too. This is usefull if you add a DOI or fix a typo. If you want to create a new separete reference for the new entry than do nothing.

2.6 Recent Changes

On the profile page, reachble via the userame link in the header, you will find a putton to review the history of changes in the "Recent Changes" view. This lists all the documents which where changed in revers chronologic order. If you need to revert to an older version of a document open the document history view via the "History" button in the left lower corner of the molecule pannel.

Each of the versions has an "Edit this Version" button which allows you to edit and save an older version of the document.



Commited

The document with the ID: 998edd34a7cba0cf1c7118e24887242155b6843e was changed.

Revisions

The new revision is : 2-afa4d07103f5ff040159b515e2f21494
dafacd32e0e0b07a4a9e255fea084f1242478079==bc3438d80e101221bfc7829752340453f5e8f2b7?

Literature References

There are 4 with same literature reference in db
You did change the literature reference data with the last edit.

Compound Name:bis[9-(9-borabicyclo[3.3.1]nonyl)-trimethylsilylamino]tin(II)

SMILES:N(B1C2CCCC1CCC2)([Si... InChI:InChI=1S/2C11H23BNSi...

InChIKey:SEPHRIQSYXGNHG-UHFFF... Revision:1

Published

NMR Shifts: ● 791.5 ppm Nucleus:119Sn Solvent:C6D6 Temperature:25 °C

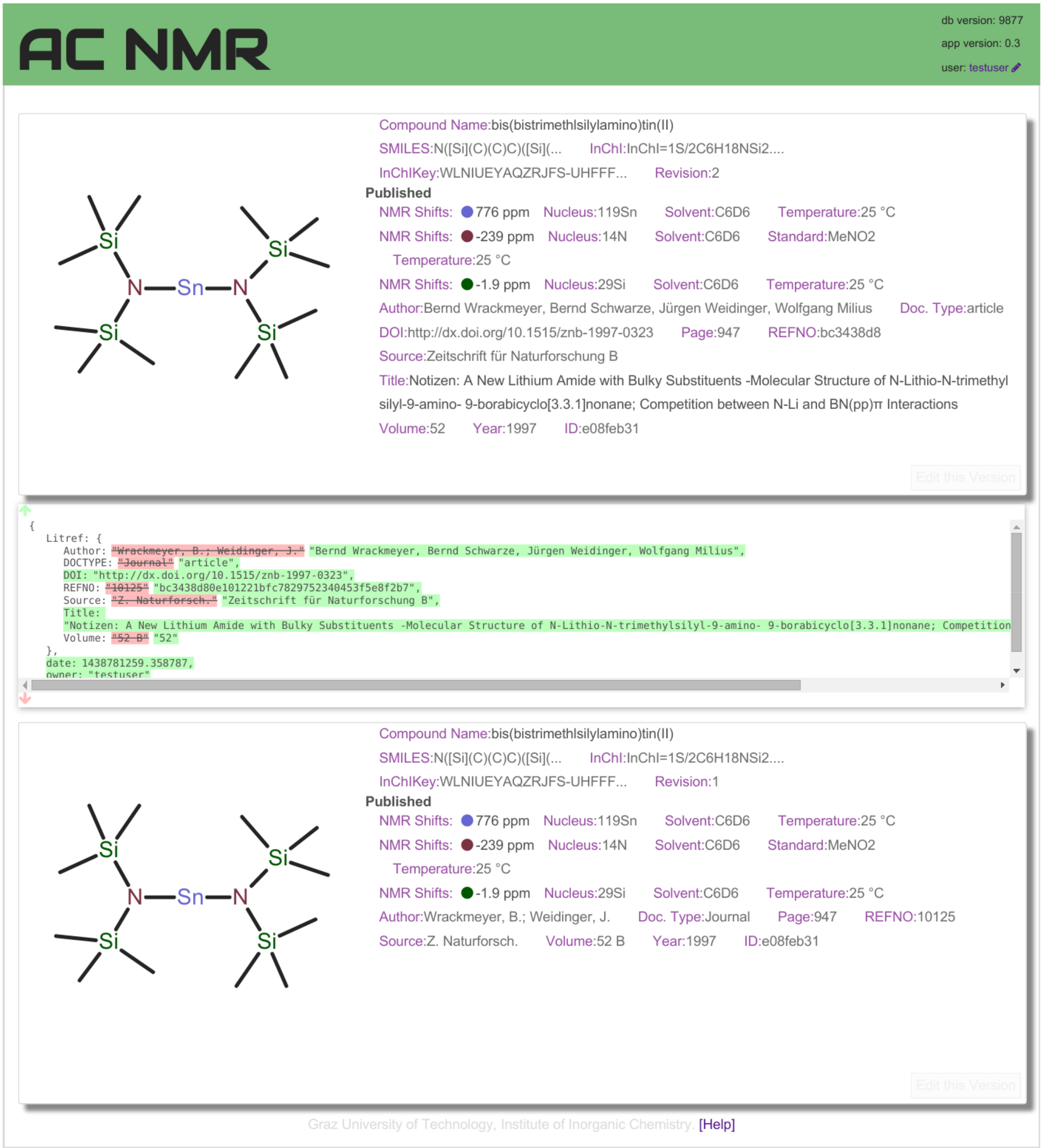
NMR Shifts: ● -199 ppm Nucleus:14N Solvent:C6D6 Standard:MeNO2 Temperature:25 °C

NMR Shifts: ● 54.8 ppm Nucleus:11B Solvent:C6D6 Standard:EtO2BF3 Temperature:25 °C

NMR Shifts: ● 0.3 ppm Nucleus:29Si Solvent:C6D6 Temperature:25 °C

Author:Wrackmeyer, B.; Weidinger, J. Doc. Type:Journal Page:947 REFNO:dafacd32

Source:Z. Naturforsch. Volume:52 B Year:1997 ID:4f84ede1



Compound Name:bis(bistrimethlsilylamino)tin(II)

SMILES:N([Si](C)(C)C)([Si](... InChI:InChI=1S/2C6H18NSi2....

InChIKey:WLNIUEYAQZRJFS-UHFFF... Revision:2

Published

NMR Shifts: ● 776 ppm Nucleus:119Sn Solvent:C6D6 Temperature:25 °C

NMR Shifts: ● -239 ppm Nucleus:14N Solvent:C6D6 Standard:MeNO2 Temperature:25 °C

NMR Shifts: ● -1.9 ppm Nucleus:29Si Solvent:C6D6 Temperature:25 °C

Author:Bernd Wrackmeyer, Bernd Schwarze, Jürgen Weidinger, Wolfgang Milius Doc. Type:article

DOI:http://dx.doi.org/10.1515/znb-1997-0323 Page:947 REFNO:bc3438d8

Source:Zeitschrift für Naturforschung B

Title:Notizen: A New Lithium Amide with Bulky Substituents -Molecular Structure of N-Lithio-N-trimethyl silyl-9-amino- 9-borabicyclo[3.3.1]nonane; Competition between N-Li and BN(pp)π Interactions

Volume:52 Year:1997 ID:e08feb31

```
{
   Litref: {
      Author: ~~"Wrackmeyer, B.; Weidinger, J."~~ "Bernd Wrackmeyer, Bernd Schwarze, Jürgen Weidinger, Wolfgang Milius",
      DOCTYPE: ~~"Journal"~~ "article",
      DOI: "http://dx.doi.org/10.1515/znb-1997-0323",
      REFNO: ~~"10125"~~ "bc3438d80e101221bfc7829752340453f5e8f2b7",
      Source: ~~"Z. Naturforsch."~~ "Zeitschrift für Naturforschung B",
      Title:
      "Notizen: A New Lithium Amide with Bulky Substituents -Molecular Structure of N-Lithio-N-trimethylsilyl-9-amino- 9-borabicyclo[3.3.1]nonane; Competition
      Volume: ~~"52 B"~~ "52"
   },
   date: 1438781259.358787,
   owner: "testuser"
```

Compound Name:bis(bistrimethlsilylamino)tin(II)

SMILES:N([Si](C)(C)C)([Si](... InChI:InChI=1S/2C6H18NSi2....

InChIKey:WLNIUEYAQZRJFS-UHFFF... Revision:1

Published

NMR Shifts: ● 776 ppm Nucleus:119Sn Solvent:C6D6 Temperature:25 °C

NMR Shifts: ● -239 ppm Nucleus:14N Solvent:C6D6 Standard:MeNO2 Temperature:25 °C

NMR Shifts: ● -1.9 ppm Nucleus:29Si Solvent:C6D6 Temperature:25 °C

Author:Wrackmeyer, B.; Weidinger, J. Doc. Type:Journal Page:947 REFNO:10125

Source:Z. Naturforsch. Volume:52 B Year:1997 ID:e08feb31

CHAPTER

THREE

MOLECULE DIAGRAMS

3.1 Atoms

The molecule diagrams are rendered with a custom molecule renderer that emphasizes its purpose of representing the data in the database and not illustration of the chemical properties. Therefore it does not show radicals or lone pairs but rather represents an atom with only 3 possible properties:

- element type
- hydrogen count
- charge

If a node in the molecule diagram has an implicit hydrogen atom associated with it, it will always be shown except for saturated carbon atoms in a chain.

The silicon atom in this structure has no hydrogen connected to it and only two bonds therefore it is two valued:

In the next structure Silicon has an implicit hydrogen:

Carbon monoxide is complicated to represent. Usually one would make the diagram like:

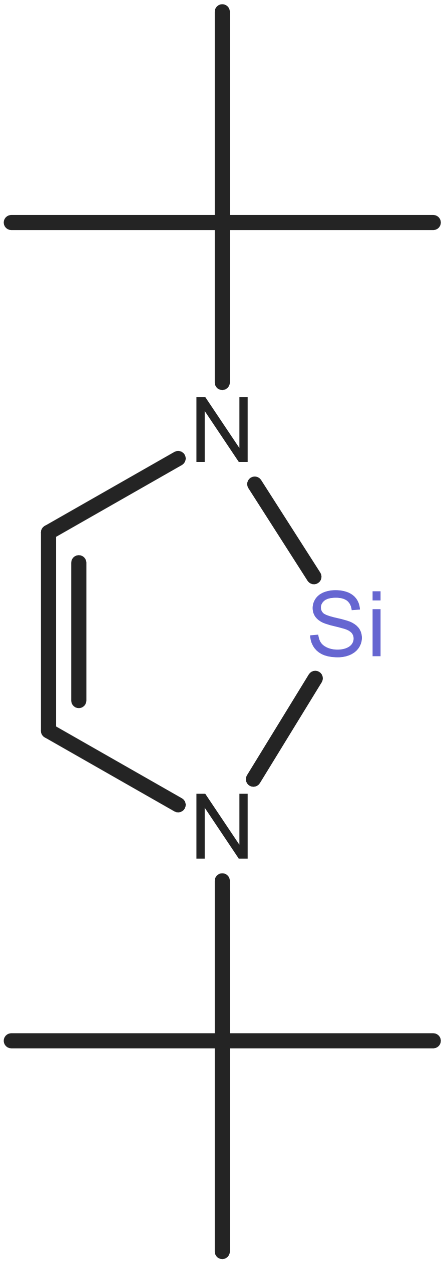
But the viewer and editor can collapse this structure to a simple “CO”.

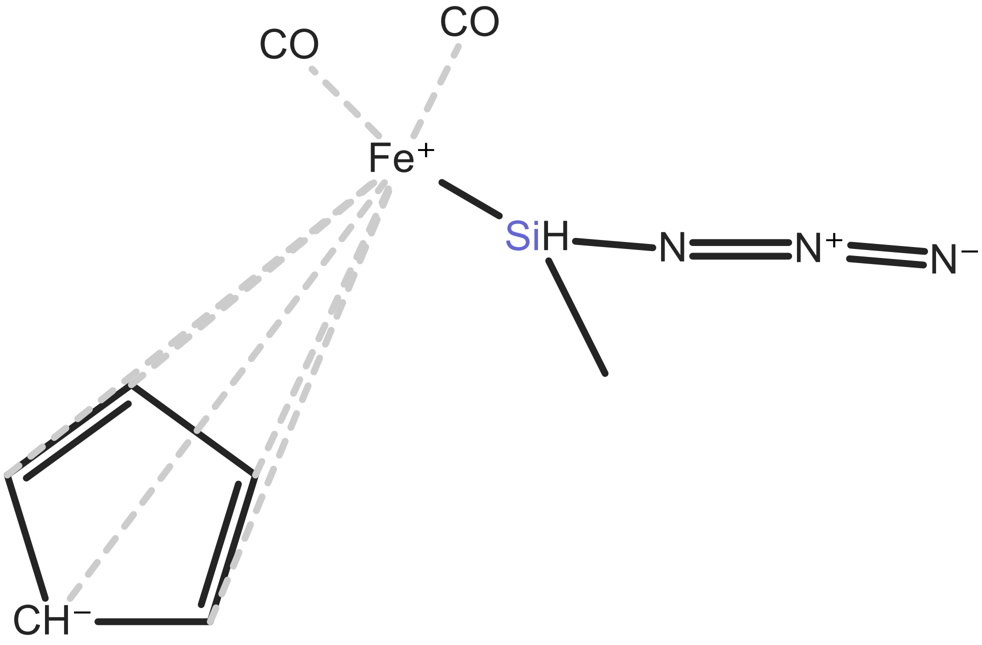
3.2 Bonds

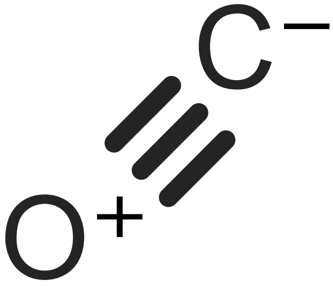
Also the way to represent bonds is limited. There are only 4 types of bonds:

- single bonds
- double bonds
- triple bonds
- coordinative bonds

So aromatic bonds have to be represented as conjugated bonds.

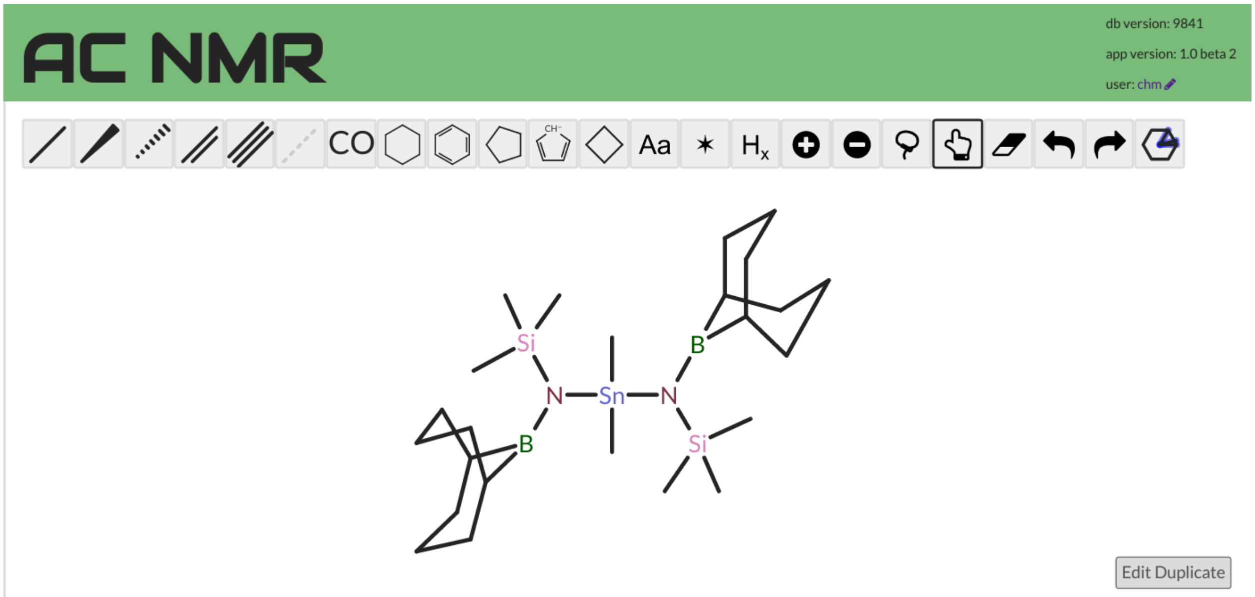

N
Si
N


CO
CO
Fe+
SiH
N
N+
N−
CH−


C−
O+

CHAPTER FOUR

MOLECULE EDITOR



The ACNMR app comes with its own molecule editor. It is based on the same SVG code as we use for the display. It is however, a data entry tool not an illustration tool. You can only create objects on the canvas that are also represented in the database.

Special features for this database include the ability to color atom labels with color codes as used for the NMR shifts data and to make “*” pseudo atoms to represent polymers.

4.1 The Hand Tool

The default mode of the molecule editor is to use the hand tool. The hand tool is able to manipulate all properties of the molecule. The features include:

Pan Dragging the background allows moving the canvas

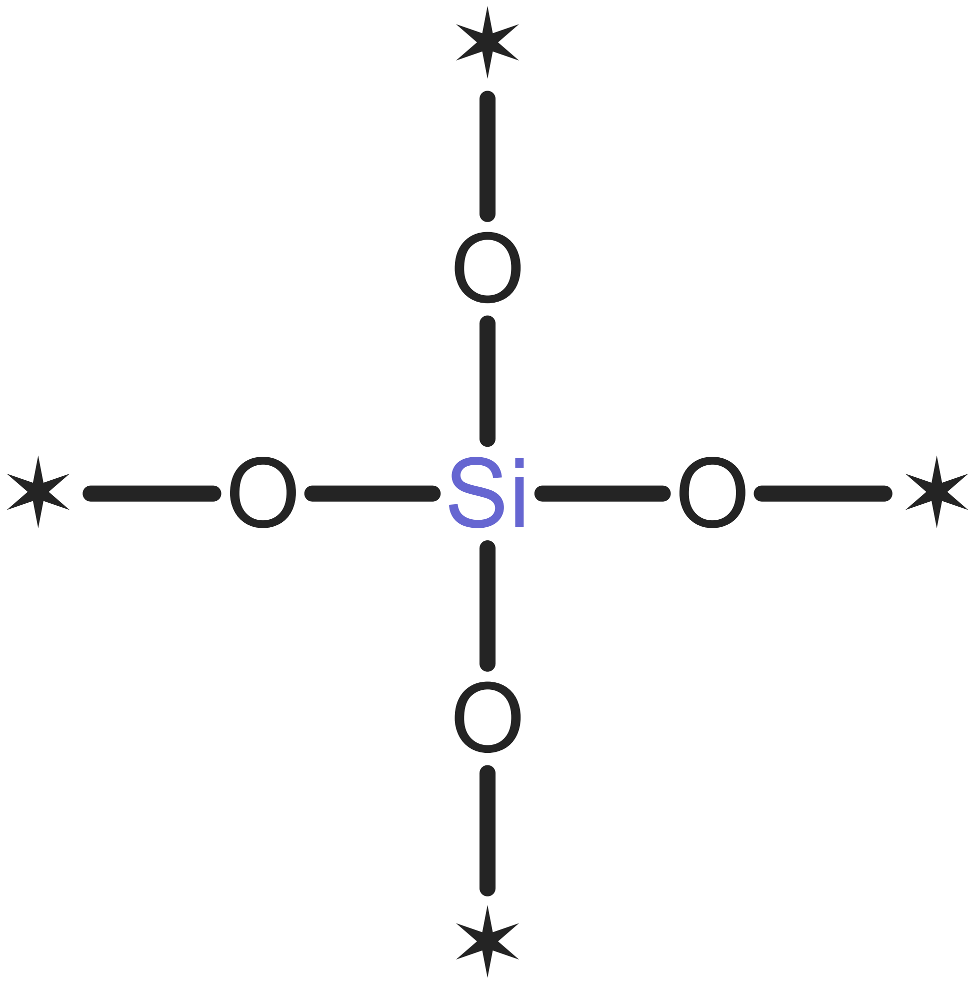
Drag node Hovering over a node will allow to grab it with the mouse and drag it around

Drag bond Bonds can also be grabbed and dragged

Change color Clicking a label or node will open a color select menu to choose a color code

Keyboard shortcuts allow to extent the hand. So far as you can do almost anything.

Shift click Add single bond with C atom to selected atom



Hover over node + letter Toggle through elements starting with the typed letter

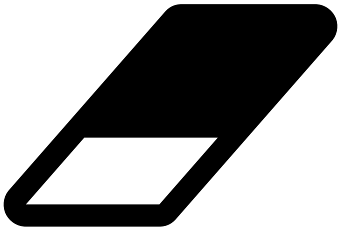
Hover over node + “+” or “-“ Change charge

Hover over node + ”.” Change hydrogen count

Hover over bond + number Change bond type. Repeat to change direction.

Hover and press Del Delete node or bond

4.2 The Erase Tool



With the erase tool active you can delete single nodes or bonds

For deleting more than one use the lasso tool

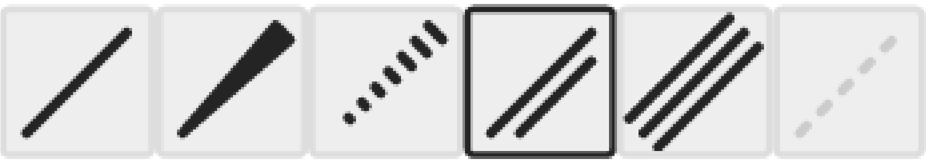
4.3 The Lasso Tool

The lasso tool is for manipulating a group of nodes. In this mode you have two ways to select nodes. You can circle the nodes with lasso or shift click on nodes to add them to the selection. On selected atoms you can do the following operations.

Drag Drag the crossed arrows to drag the selected atoms or grab one selected node and drag all the nodes.

Rotate Drag the rotating arrow symbol to rotate selected structure.

Delete Press Del to delete selected atoms.



4.4 The Bond Tools

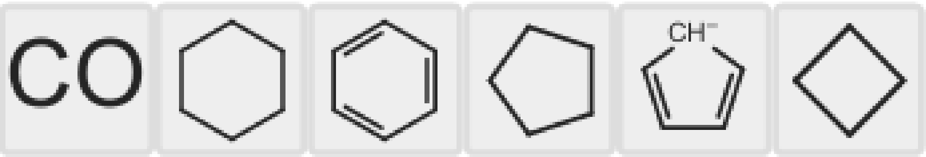
There are buttons to add C atoms with bonds of an specific order or change the order of bonds in the drawing. The modes it has are the following.

Hover over node click Add C with bond into the direction where there is the most space.

Hover over node drag Add C with bond and drag around

Hover bond click Change bond order. Repeat to change direction.

4.5 The add Fragment Tools



There are a collection of rings and other fragments that can be added via a special tool. The fragments may be added in one of these ways.

Click on canvas Insert fragment here and drag.

Click on node. Attach fragment here and rotate in order to use the direction with the most space. After inserting you may rotate it around the attachment point by dragging.

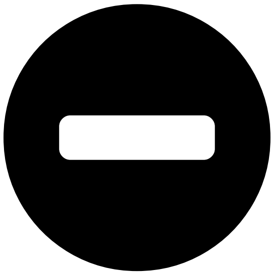
Click on bond Insert fragment by aligning it with the bond. After inserting you may flip the alignment side by dragging the mouse around.

4.6 The Hydrogen Count Tool

$$H_x$$

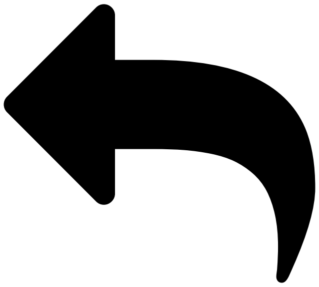
Hover over atom and click to change hydrogen count. Same as hover and press ”.”

4.7 The Charge Tools



Hover over atom and click to change charge.Same as hover and press “+” or “-“

4.8 Undo, Redo buttons



Buttons to undo and redo edits. Ctrl-Z for undo and Ctrl-Y for redo work too.

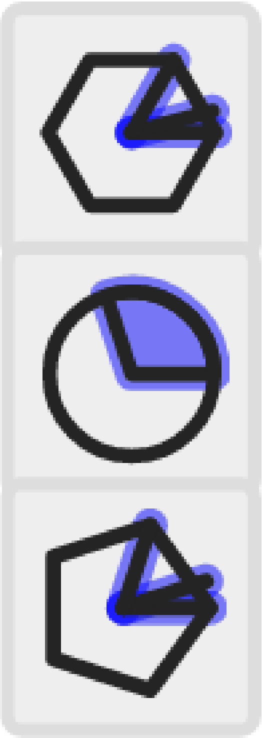
4.9 Cut and Paste

Ctrl-C and Ctrl-V work within one editor as expected on selected groups of atoms.

4.10 Mark All Atoms In Editor

Ctrl-A will mark all atoms in the editor.

4.11 Snap Mode



When moving fragments around the editor offeres guides to simplify matching angles and lengths. The three modes available are:

Hexagon Mode: Angles snap to 90/6° angles.

Pentagon Mode: Angles snap to 90/5° angles.

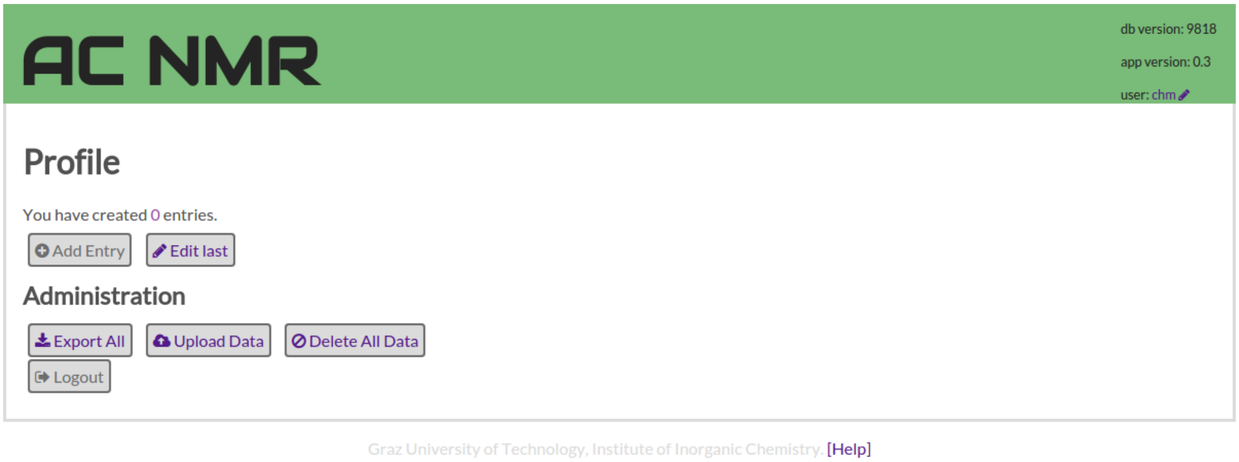
Circle Mode: Snap disabled.

CHAPTER FIVE

HOW TO IMPORT AND EXPORT DATA

5.1 Export

Users with administrator privileges can export the data on their profile page. This is reachable via the user name link in the header.



Profile

You have created 0 entries.

Administration

Graz University of Technology, Institute of Inorganic Chemistry. [Help]

The `Export All` link will start a download of the data as JSON. The format contains one JSON object per line. This is because it is much easier to parse it item by item then the whole thing at once. Use the save as function of your browser to save a copy on your disk.

5.2 Import/Upload

There are three ways of importing Data, you may use the *Bulk Import* features or the *Import or Upload single Entry* . Both are reachable vie the `Upload Data` Button on the profile Page.

5.3 Import or Upload single Entry

On the profile bage you will find the `Upload Data` Button, which will lead you to an upload form. After Uploading the file you can select the `Single Json` Option which will open an *Editor Form* to edit and commit the file to the database.

5.4 Bulk Import

You can import small junks of new data into the database by using the `Upload Data` but as this will make it hard to track the changes. It is recommended to reset the database first and re-import a consistent data set.

You can upload Scidex SDF data or the internal JSON lines format from an `Export All` download. You will have to chose which one it is, before the next step, which is checking the data. This will take a few minutes. In case of JSON lines format it will check the data for conformity with the current schema and in case of Scidex SDF it will parse the SDF to extract the data.

During this step, you will see the number of items already processed end eventual warnings and error messages. The warnings will include if no INCHI could be computed from the SDF, which is the case, if there are any special atoms in it like "*" or "R". If no SMILES code could be computed the structure in question is seriously non conform. But entries with no atoms and bonds in the structure part, will also trigger this error. Do not close the window and wait until you are being offered to commit the data.

Committing will send the data to the database and the import is done. It remains to wait until the changes are processed and the new version number shows up in the header.

5.5 Reset and Import

The `Delete All Data` resets the database. Before you do such a thing make sure you did export and save a backup of the current state before you delete anything.

If you deleted the data, you will recognize that you still can see and search the database after you deleted it which is confusing. This is because what really happened is, we reseted the version number to 0, and deleted the CouchDB database.

This means as soon as you import any new data, the version of the CouchDB database changes and it will trigger a rebuild of the search tables. The import of the new data is complete if the version number in the header is the same as the CouchDB version. During the time the search tables are rebuilding the version field will show both versions as `searchtable_version/CouchDB_version`

5.6 Version Numbers

The version number comes from the CouchDB version system. As it is an append only data store, any document added removed or changed will increase the version by one. Version 0 therefore must be an empty database.

CHAPTER

SIX

TECHNOLOGY

The ACNMR app is written in Python using the Flask framework. For chemical data processing it uses the CML parser and molecule analysis from Indigo the similarity and substructure search from Rdkit.

The core is the data model as described in *The NMR Document Object JSON Format*. This data model represents an entry in the database. It contains one molecular structure, one literature reference, and NMR data sets from this literature source. Entries with the same molecule are grouped in the detailed view of the entry but stored as independent entries. For grouping identical molecules, the system uses InChI Keys where applicable and arbitrary Molecule IDs if the structure can not represented as InChI.

As one literature reference may contain multiple compounds the literature references can also be duplicates across entries. Literature references are uniquely identified by a hash because we prefer global unique IDs over serial IDs in this system.

The data format is JSON which is a very versatile format especially for use in web applications. These files are stored in Couchdb which is an append only NoSQL database with powerful replication features. This is the authoritative data base. It manages the history of documents and replication or synchronization with other instances of the ACNMR application. Its job is, to have a consistent collection of a large number of ACNMR documents safely on the disk at all time. That it is designed to do. It is however, not suited to do complex dynamic queries, this is the job of SQL Databases.

For this purpose when you start the ACNMR WSGI server, it creates a process that connects to the CouchDB and listens for changes. This process will fetch the changes and write the data into the SQL Database distributed over a couple of tables suitable to perform the searches on the data. While doing that also the substructure and similarity finger-prints will be calculated and stored in suitable tables.

This SQL database currently is Sqlite. The changes monitor process fetches the changes since the last version, computes the changes and saves them to the SQL data base.

CHAPTER

SEVEN

SYSTEM INSTALLATION

7.1 Source code

If you have access the source code can be obtained by cloning from http://ac-software.tugraz.at/git/acnmr.git:

```
git clone http://ac-software.tugraz.at/git/acnmr.git
```

7.2 Dependencies

To run an instance of the ACNMR Server you need the following prerequisites.

- **A web server that can host WSGI apps.**
  - apache
  - gunicorn
- sqlite.
- Couchdb
- **Python with modules**
  - Rdkit.
  - Indigo
  - sqlite3
  - json
  - jsonschema
  - couchdb
  - flask

User authentication is not handled on flask level. For user authentication the authentication has to be provided by the web server e.g. simple authentication or some kind of single-sign-on system.

7.3 WSGI Config

The configuration of the ACNMR server is done by an WSGI file. This file has to be provided to the web server to tell what code should be executed and with which environment.

```
import sys,os
# The working directory of the app
HOME='/home/appadmin/acnmrdev'
sys.path.insert(0, HOME)
# add RDKit dir to python path
sys.path.insert(0, '/opt/RDKit_2014_09_1')

# this is used to sign cookies should be  random string that is kept secret
os.environ['SINMRSESSTOKEN']= '17486e6e'
# environment variabled that are needet for RDKit to work
os.environ['RDBASE']='/opt/RDKit_2014_09_1/'
# couchdb connection data
os.environ['COUCHDBNAME']='acnmr'
os.environ['COUCHDBURL']='http://127.0.0.1:5984/'
os.environ['COUCHUSER']='username'
os.environ['COUCHSECRET']='secret'
# name of the sqlite database file
os.environ['DATABASE']='NMR.db'
# An alternative title tag
os.environ['APPTITLE']='Sandbox'
# username of the superuser. There must be one, or no one can add other users
os.environ['ACNMROWNER']="superuser"
# change directory
os.chdir(HOME)
# get the changes monitor module
import acnmr.couchchanges   as couchchanges
# start changes monitor process
couchchanges.start()
# import the app module that will be run by the WSGI container
from acnmr.server import app as application
```

7.4 Configure Apache

Apache is not the fastest web server, but it offers a wide variety of authentication abilities and therefore it is the web server we choose.

7.4.1 Set global environment variables for RDkit

In order for the RDKit module to find the dynamic libraries one must set the library path environment variable. This cannot be set in the WSGI file but must be set in the Apache environment variables section.

Add the location where your installation of RDKit resides to the Apache `envvars` file, or where appropriate:

```
export LD_LIBRARY_PATH=/opt/RDKit_2014_09_1/lib
```

7.4.2 Configure Apache Virtual Host

As the traffic might go through the Internet and you will have some kind of authentication HTTPS is pretty much mandatory. Here is an example configuration for apache2 to setup a virtual host for HTTPS.

```
<VirtualHost    ac-appserver.tugraz.at:443>
   ServerAdmin christian.meisenbichler@tugraz.at
   ServerName      ac-appserver.tugraz.at
   ErrorLog /var/log/apache2/acnmrdev-error.log
   CustomLog /var/log/apache2/acnmrdev-access.log combined
   # set the WSGI variables
   WSGIDaemonProcess acnmrdev user=appadmin group=appadmin threads=1
   WSGIScriptAlias / /home/appadmin/acnmrdev.wsgi
   # working directory
   <Directory /home/appadmin/acnmrdev/>
      WSGIProcessGroup acnmrdev
      WSGIScriptReloading On
      LimitRequestBody 0
   </Directory>
   # you really must use SSL
   SSLEngine On
   SSLCertificateFile /etc/apache2/ssl/crt/ac-appserver.tugraz.at.pem
   SSLCertificateKeyFile /etc/apache2/ssl/key/ac-appserver.tugraz.at.key
   SSLCertificateChainFile  /etc/apache2/ssl/chain-5955-shib-test.tugraz.at.pem
   AllowEncodedSlashes On
   ProxyRequests Off
   KeepAlive Off


   <Location />
      SSLRequireSSL On
      SSLVerifyClient Off
      SSLVerifyDepth 1
      SSLOptions +StdEnvVars +StrictRequire
      WSGIProcessGroup acnmrdev
      SSLRenegBufferSize 104860000
      WSGIPassAuthorization On
      # example configuration to use HTTP basic auth
      AuthType Basic
      AuthName "Authentication Required"
      AuthUserFile "/etc/htpasswd"
      Require valid-user
      Order allow,deny
      Allow from all
   </Location>
</VirtualHost>
```

CHAPTER

EIGHT

THE AC NMR DOCUMENT FORMAT

For the storage in the database and for an exchange file format the AC-NMR DB uses a JSON file format. This format encodes the structure formula and the literature reference.

The atom and bonds objects in this file format are modeled as closely as possible to the CML XML format in terms of label names and options.

8.1 The NMR Document Object JSON Format

The ‘*‘ signifies a required Field.

Data format for molecules and NMR data.

Type object

Contains _rev, _id, *docid*, *owner*, *date*, *molecule**, *NMRDatasets**, *sdf*, *Litref*, *Annotations*, *Error*

Required True

JSON Path

- *#*

Example JSON:

```
{
  "NMRDatasets": [],
  "molecule": {
    "Mol_ID": "",
    "bonds": [],
    "atoms": []
  }
}
```

8.1.1 _rev

Revision ID as used by the document store.

Type string

Required False

JSON Path

- *#* [’_rev‘]

Example JSON:

```
{"_rev": ""}
```

8.1.2 _id

Document ID GUID as used by the document store.

Type string

Required False

JSON Path

- *#* [’_id‘]

Example JSON:

```
{"_id": ""}
```

8.1.3 docid

Unique document ID. This is used as ID throughout the application.

Type string

Required False

JSON Path

- *#* [’*docid*‘]

Example JSON:

```
{"docid": ""}
```

8.1.4 owner

User who did last edit.

Type string

Required False

JSON Path

- *#* [’*owner*‘]

Example JSON:

```
{"owner": ""}
```

8.1.5 date

UTC Time of last edit as Unix time stamp. (seconds since epoc)

Type number

Required False

JSON Path

- *#* [’*date*‘]

Example JSON:

```
{"date": 0}
```

8.1.6 molecule

Contains molecule data. Data, like atoms, bonds and color coding.

Type object

Contains *InChI*, *InChIKey*, *Mol_ID**, *MolLinked*, *Mol_Name*, *CAS_RN*, *SMILES*, *Stereoinformation*, *BoilingPoint*, *MSPeakListPeaks*, *MSPeakListSourceID*, *MeltingPoint*, *FORMULA*, Comment, *atoms**, *bonds**

Required True

JSON Path

- *#* [’*molecule*‘]

Example JSON:

```
{"molecule": {"Mol_ID": "","bonds": [],"atoms": []}}
```

8.1.7 InChI

International chemical identifier.

Type string

Required False

JSON Path

- *#* [’*molecule*‘][’*InChI*‘]

Example JSON:

```
{"InChI": ""}
```

8.1.8 InChIKey

International chemical identifier hashed. If available, it will be used for finding data sets with the same chemical structure.

Type string

Required False

JSON Path

- *#* [’*molecule*‘][’*InChIKey*‘]

Example JSON:

```
{"InChIKey": ""}
```

8.1.9 Mol_ID

Arbitrary ID for molecule. This is used if InChI is not available.

Type string

Required True

JSON Path

- *#* [’*molecule*‘][’*Mol_ID*‘]

Example JSON:

```
{"Mol_ID": ""}
```

8.1.10 MolLinked

Flag for deciding if changes on this molecule structure should affect the others with the same ID

Type boolean

Required False

Default False

JSON Path

- *#* [’*molecule*‘][’*MolLinked*‘]

Example JSON:

```
{"MolLinked": false}
```

8.1.11 Mol_Name

Compound name.

Type string

Required False

JSON Path

- *#* [’*molecule*‘][’*Mol_Name*‘]

Example JSON:

```
{"Mol_Name": ""}
```

8.1.12 CAS_RN

Compound number from the American Chemical Society.

Type string

Required False

JSON Path

- *#* [’*molecule*‘][’*CAS_RN*‘]

Example JSON:

```
{"CAS_RN": ""}
```

8.1.13 SMILES

SMILES Code. This represents the connection graph of the molecule and is used for substructure and similarity search.

Type string

Required False

JSON Path

- *#* [’*molecule*‘][’*SMILES*‘]

Example JSON:

```
{"SMILES": ""}
```

8.1.14 Stereoinformation

Marker for molecules containing chiral structure.

Type boolean

Required False

JSON Path

- *#* [’*molecule*‘][’*Stereoinformation*‘]

Example JSON:

```
{"Stereoinformation": true}
```

8.1.15 BoilingPoint

Boiling point [°C].

Type string

Required False

JSON Path

- *#* [’*molecule*‘][’*BoilingPoint*‘]

Example JSON:

```
{"BoilingPoint": ""}
```

8.1.16 MSPeakListPeaks

Type string

Required False

JSON Path

- *#* [’*molecule*‘][’*MSPeakListPeaks*‘]

Example JSON:

```
{"MSPeakListPeaks": ""}
```

8.1.17 MSPeakListSourceID

Type string

Required False

JSON Path

- *#* [’*molecule*‘][’*MSPeakListSourceID*‘]

Example JSON:

```
{"MSPeakListSourceID": ""}
```

8.1.18 MeltingPoint

Melting point [°C].

Type string

Required False

JSON Path

- *#* [’*molecule*‘][’*MeltingPoint*‘]

Example JSON:

```
{"MeltingPoint": ""}
```

8.1.19 FORMULA

Sum formula.

Type string

Required False

JSON Path

- *#* [’*molecule*‘][’*FORMULA*‘]

Example JSON:

```
{"FORMULA": ""}
```

8.1.20 Comment

Arbitrary comment.

Type string

Required False

JSON Path

- *#* [’*molecule*‘][’Comment‘]

Example JSON:

```
{"Comment": ""}
```

8.1.21 atoms

Type array() items: {*2d*, *3d*, *elementType*, *hydrogenCount*, *formalCharge*, *role*, *properties*}

Required True

JSON Path

- *#* [’*molecule*‘][’*atoms*‘]

Example JSON:

```
{"atoms": []}
```

8.1.22 2d

2D atom coordinates for structure diagrams. [a.u]

Type array(2) items: number

Required True

JSON Path

- *#* [’*molecule*‘][’*atoms*‘][0][’*2d*‘]

Example JSON:

```
{"2d": []}
```

8.1.23 3d

3D atom coordinates for 3d structure (crystal-structure). [a.u]

Type array(3) items: number

Required False

JSON Path

- *#* [’*molecule*‘][’*atoms*‘][0][’*3d*‘]

Example JSON:

```
{"3d": []}
```

8.1.24 elementType

Symbol of chemical Element. Allowed are the symbols as defined in CML.

Type string

values [H, He, Li, Be, B, C, N, O, F, Ne, Na, Mg, Al, Si, P, S, Cl, Ar, K, Ca, Sc, Ti, V, Cr, Mn, Fe, Co, Ni, Cu, Zn, Ga, Ge, As, Se, Br, Kr, Rb, Sr, Y, Zr, Nb, Mo, Tc, Ru, Rh, Pd, Ag, Cd, In, Sn, Sb, Te, I, Xe, Cs, Ba, La, Ce, Pr, Nd, Pm, Sm, Eu, Gd, Tb, Dy, Ho, Er, Tm, Yb, Lu, Hf, Ta, W, Re, Os, Ir, Pt, Au, Hg, Tl, Pb, Bi, Po, At, Rn, Fr, Ra, Ac, Th, Pa, U, Np, Pu, Am, Cm, Bk, Cf, Es, Fm, Md, No, Lr, Rf, Db, Sg, Bh, Hs, Mt, Ds, Rg, Cn, Uuq, Uuh, Du, R]

Required True

JSON Path

- *#* [’*molecule*‘][’*atoms*‘][0][’*elementType*‘]

Example JSON:

```
{"elementType": "H"}
```

8.1.25 hydrogenCount

Hydrocen-count as defined in CML.

Type number

Required False

JSON Path

- *#* [’*molecule*‘][’*atoms*‘][0][’*hydrogenCount*‘]

Example JSON:

```
{"hydrogenCount": 0}
```

8.1.26 formalCharge

Formal charge as defined in CML.

Type number

Required False

JSON Path

- *#* [’*molecule*‘][’*atoms*‘][0][’*formalCharge*‘]

Example JSON:

```
{"formalCharge": 0}
```

8.1.27 role

The role is used to further describe ‘R’ or ‘Du’ pseudo Atoms. May be used to describe polymeres by using ‘*’

Type string

Required False

JSON Path

- *#* [’*molecule*‘][’*atoms*‘][0][’*role*‘]

Example JSON:

```
{"role": ""}
```

8.1.28 properties

Properties of the atom that are not directly related to the molecular structure, such as coloring, and other annotations.

Type object

Contains *color*

Required False

JSON Path

- *#* [’*molecule*‘][’*atoms*‘][0][’*properties*‘]

Example JSON:

```
{"properties": {}}
```

8.1.29 color

Color code of atom to hilight or mark atom or set of atoms.

Type string or null

Required False

JSON Path

- *#* [’*molecule*‘][’*atoms*‘][0][’*properties*‘][’*color*‘]

Example JSON:

```
{"color": null}
```

8.1.30 bonds

Array of bonds.

Type array() items: {*id*, *bond*, *order*, *stereoType*}

Required True

JSON Path

- *#* [’*molecule*‘][’*bonds*‘]

Example JSON:

```
{"bonds": []}
```

8.1.31 id

type object

Required False

JSON Path

- *#* [’*molecule*‘][’*bonds*‘][0][’*id*‘]

Example JSON:

```
{"id": {}}
```

8.1.32 bond

Indices of atoms which are connected via this bond. Index starts with 1.

Type array(2) items: number

Required True

JSON Path

- *#* [’*molecule*‘][’*bonds*‘][0][’*bond*‘]

Example JSON:

```
{"bond": []}
```

8.1.33 order

The type or ‘order’ of this bond. The bond order values are as defined in CML.

Type string

values [1, hbond, partial01, partial12, 2, partial23, 3, A]

Required True

JSON Path

- *#* [’*molecule*‘][’*bonds*‘][0][’*order*‘]

Example JSON:

```
{"order": "1"}
```

8.1.34 stereoType

Stereo-type as defined in CML.

Type string

values [C, T, W, H]

Required False

JSON Path

- *#* [’*molecule*‘][’*bonds*‘][0][’*stereoType*‘]

Example JSON:

```
{"stereoType": "C"}
```

8.1.35 NMRDatasets

Chemical shifts and couplings.

Type array() items: {*Shifts*, *Nucleus*, *Solvent*, *Standard*, *Method*, *Temp*}

Required True

JSON Path

- *#* [’*NMRDatasets*‘]

Example JSON:

```
{"NMRDatasets": []}
```

8.1.36 Shifts

List of shifts for from one nucleus, standard, solvent, temperature ... -combination.

Type array() items: {*Shift*, *ColorRef*, Comment, *J*}

Required False

JSON Path

- *#* [’*NMRDatasets*‘][0][’*Shifts*‘]

Example JSON:

```
{"Shifts": []}
```

8.1.37 Shift

Numerical value of chemical shift in [ppm].

Type number or object

Contains *from*, *to*

Required False

JSON Path

- *#* [’*NMRDatasets*‘][0][’*Shifts*‘][0][’*Shift*‘]

Example JSON:

```
{"Shift": null}
```

8.1.38 from

Type number

Required False

JSON Path

- *#* [’*NMRDatasets*‘][0][’*Shifts*‘][0][’*Shift*‘][’*from*‘]

Example JSON:

```
{"from": 0}
```

8.1.39 to

Type number

Required False

JSON Path

- *#* [’*NMRDatasets*‘][0][’*Shifts*‘][0][’*Shift*‘][’*to*‘]

Example JSON:

```
{"to": 0}
```

8.1.40 ColorRef

Colour label to match the colors in molecular structure.

Type string

Required False

JSON Path

- *#* [’*NMRDatasets*‘][0][’*Shifts*‘][0][’*ColorRef*‘]

Example JSON:

```
{"ColorRef": ""}
```

8.1.41 Comment

Comment for shift.

Type string

Required False

JSON Path

- *#* [’*NMRDatasets*‘][0][’*Shifts*‘][0][’Comment‘]

Example JSON:

```
{"Comment": ""}
```

8.1.42 J

List of coupling constants for this shift.

Type array() items: {*ftype*, *textval*, *type*, *unit*, *value*}

Required False

JSON Path

- *#* [’*NMRDatasets*‘][0][’*Shifts*‘][0][’*J*‘]

Example JSON:

```
{"J": []}
```

8.1.43 ftype

Coupling type including text formatting information.

Type string

Required True

Default J()

JSON Path

- *#* [’*NMRDatasets*‘][0][’*Shifts*‘][0][’*J*‘][0][’*ftype*‘]

Example JSON:

```
{"ftype": "J()"}
```

8.1.44 textval

Unparsed text value if availabe.

Type string

Required True

JSON Path

- *#* [’*NMRDatasets*‘][0][’*Shifts*‘][0][’*J*‘][0][’*textval*‘]

Example JSON:

```
{"textval": ""}
```

8.1.45 type

Coupling type whithout text formatting information.

Type string

Required True

JSON Path

- *#* [’*NMRDatasets*‘][0][’*Shifts*‘][0][’*J*‘][0][’*type*‘]

Example JSON:

```
{"type": ""}
```

8.1.46 unit

Unit of numeric value.

Type string

Required False

Default Hz

JSON Path

- *#* [’*NMRDatasets*‘][0][’*Shifts*‘][0][’*J*‘][0][’*unit*‘]

Example JSON:

```
{"unit": "Hz"}
```

8.1.47 value

Numeric value of coupling constant.[Hz]

Type number

Required True

JSON Path

- *#* [’*NMRDatasets*‘][0][’*Shifts*‘][0][’*J*‘][0][’*value*‘]

Example JSON:

```
{"value": 0}
```

8.1.48 Nucleus

Type of nucleus used in NMR, ‘29Si’, for example.

Type string

Required False

JSON Path

- *#* [’*NMRDatasets*‘][0][’*Nucleus*‘]

Example JSON:

```
{"Nucleus": ""}
```

8.1.49 Solvent

Solvent in which NMR measurement took place.

Type string

Required False

JSON Path

- *#* [’*NMRDatasets*‘][0][’*Solvent*‘]

Example JSON:

```
{"Solvent": ""}
```

8.1.50 Standard

Reference compount for shift.

Type string

Required False

JSON Path

- *#* [’*NMRDatasets*‘][0][’*Standard*‘]

Example JSON:

```
{"Standard": ""}
```

8.1.51 Method

NMR Method.

Type string

Required False

JSON Path

- *#* [’*NMRDatasets*‘][0][’*Method*‘]

Example JSON:

```
{"Method": ""}
```

8.1.52 Temp

Sample teperature. [°C]

Type string

Required False

JSON Path

- *#* [’*NMRDatasets*‘][0][’*Temp*‘]

Example JSON:

```
{"Temp": ""}
```

8.1.53 sdf

Field to store legacy sdf source if available.

Type string

Required False

JSON Path

- *#* ['*sdf*']

Example JSON:

```
{"sdf": ""}
```

8.1.54 Litref

Literature reference. Each data document may be associated with one literature reference.

Type object

Contains *LitrefLinked*, *Author*, *YEAR*, *Title*, *DOCTYPE*, *Page*, *REFNO**, *Source*, *Volume*, *Keyword*, *Editor*, *CASAN*, *DOI*

Required False

JSON Path

- *#* ['*Litref*']

Example JSON:

```
{"Litref": {"REFNO": ""}}
```

8.1.55 LitrefLinked

Flag for deciding if changes on this literature reference should be changed in all documents.

Type boolean

Required False

Default True

JSON Path

- *#* ['*Litref*']['*LitrefLinked*']

Example JSON:

```
{"LitrefLinked": true}
```

8.1.56 Author

Author.

Type string

Required False

Default

JSON Path

- *#* ['*Litref*']['*Author*']

Example JSON:

```
{"Author": ""}
```

8.1.57 YEAR

Year of publication.

Type number

Required False

JSON Path

- *#* [’*Litref*‘][’*YEAR*‘]

Example JSON:

```
{"YEAR": 0}
```

8.1.58 Title

Title.

Type string

Required False

JSON Path

- *#* [’*Litref*‘][’*Title*‘]

Example JSON:

```
{"Title": ""}
```

8.1.59 DOCTYPE

Journal, book, thesis ...

Type string

Required False

JSON Path

- *#* [’*Litref*‘][’*DOCTYPE*‘]

Example JSON:

```
{"DOCTYPE": ""}
```

8.1.60 Page

Page number.

Type string

Required False

JSON Path

- *#* [’*Litref*‘][’*Page*‘]

Example JSON:

```
{"Page": ""}
```

8.1.61 REFNO

This is an ID for the reference. Many documents may come from the same Literature reference.

Type string

Required True

JSON Path

- *#* [’*Litref*‘][’*REFNO*‘]

Example JSON:

```
{"REFNO": ""}
```

8.1.62 Source

Journal name, for example.

Type string

Required False

JSON Path

- *#* [’*Litref*‘][’*Source*‘]

Example JSON:

```
{"Source": ""}
```

8.1.63 Volume

Journal or book volume.

Type string

Required False

JSON Path

- *#* [’*Litref*‘][’*Volume*‘]

Example JSON:

```
{"Volume": ""}
```

8.1.64 Keyword

List of keywords seperated by ‘ ‘ or ‘,’.

Type string

Required False

JSON Path

- *#* [’*Litref*‘][’*Keyword*‘]

Example JSON:

```
{"Keyword": ""}
```

8.1.65 Editor

Publication Editor

Type string

Required False

JSON Path

- *#* [’*Litref*‘][’*Editor*‘]

Example JSON:

```
{"Editor": ""}
```

8.1.66 CASAN

Chemical Abstract Service reference id of the American Chemical Society.

Type string

Required False

JSON Path

- *#* [’*Litref*‘][’*CASAN*‘]

Example JSON:

```
{"CASAN": ""}
```

8.1.67 DOI

Digital Object Identifier for the literature reference.

Type string

Required False

Default

JSON Path

- *#* [’*Litref*‘][’*DOI*‘]

Example JSON:

```
{"DOI": ""}
```

8.1.68 Annotations

This may contain arbitrary data and comments.

Type object

Contains

Required

False

JSON Path

- *#* ['*Annotations*']

Example JSON:

```
{"Annotations": {}}
```

8.1.69 Error

Processing errors.

Type string

Required False

JSON Path

- *#* ['*Error*']

Example JSON:

```
{"Error": ""}
```

Get this guide as PDF.