
Subject: DWAR file format

Posted by [catalyst](#) on Thu, 07 Mar 2019 12:36:24 GMT

[View Forum Message](#) <> [Reply to Message](#)

hey folks,

i've been looking for a format specification for the DWAR file format; so far, no luck . can somebody please direct me? thanks!

Subject: Re: DWAR file format

Posted by [thomas](#) on Thu, 07 Mar 2019 17:26:47 GMT

[View Forum Message](#) <> [Reply to Message](#)

there is no document that describes the entire file format, but there is open-source code that reads or writes it. In principal the files contains these sections:

- header with version and row count
 - column properties assigning properties to columns identified by names. These include
 - special types like idcode (encoded chemical structure), rxncode (encoded reaction), descriptors
 - parent-child relationships, e.g. to associate a descriptor column to a parent idcode column
 - detail type and access information in case the column contains references to details (images, html code)
 - others
 - the TAB delimited data table itself
 - optionally encoded embedded details
 - optionally encoded row lists
 - optionally embedded macros
 - the datawarrior template, which includes information about open views and filters
-

Subject: Re: DWAR file format

Posted by [catalyst](#) on Fri, 08 Mar 2019 15:43:03 GMT

[View Forum Message](#) <> [Reply to Message](#)

thanks for the quick reply! could you please point me to some examples? I'm particularly interested in how to put 3D conformer data into the file.

Thanks again for your help!

Subject: Re: DWAR file format

Posted by [thomas](#) on Sat, 09 Mar 2019 20:33:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

you could use something like the following Java method, which reads conformers from an SD-File, creates a CompoundTableModel, populates it with the conformers, and uses a

CompoundTableSaver

to write the table model into a native DataWarrior file.

If you need to write much larger files, this method may not be the appropriate approach, because of the large memory footprint. I suggest that you contact me on my idorsia e-mail address (see openmolecules.org about page) to find an optimal solution.

(the following depends on some files of the DataWarrior source code):

```
public static void createConformerDWARDemo() {
    final int ROW_COUNT = 1000;

    CompoundTableModel tableModel = new CompoundTableModel();
    tableModel.initializeTable(ROW_COUNT, 3);

    // this is a hack to prevent nullpointer exception and won't be necessary in the future
    tableModel.setDetailHandler(new CompoundTableDetailHandler(tableModel));

    final String IDENTIFIER_COLUMN_NAME = "ID";
    final String STRUCTURE_COLUMN_NAME = "Structure";

    // first column will contain the structure name/ID
    tableModel.setColumnName(IDENTIFIER_COLUMN_NAME, 0);

    // define second column to contain idcodes
    tableModel.setColumnName(STRUCTURE_COLUMN_NAME, 1);
    tableModel.setColumnProperty(1, CompoundTableConstants.cColumnPropertySpecialType,
    CompoundTableConstants.cColumnTypeIDCode);

    // define third column to contain the 3D-coordinates and make it a child of the structure column
    tableModel.setColumnName(CompoundTableConstants.cColumnType3DCoordinates, 2);
    tableModel.setColumnProperty(2, CompoundTableConstants.cColumnPropertySpecialType,
    CompoundTableConstants.cColumnType3DCoordinates);
    tableModel.setColumnProperty(2, CompoundTableConstants.cColumnPropertyParentColumn,
    STRUCTURE_COLUMN_NAME);

    // read some molecules from an SD-File that should contain 3-dimensional atom coordinates
    SDFFileParser parser = new SDFFileParser("someFileName.sdf");
    for (int row=0; row<ROW_COUNT && parser.next(); row++) {
        StereoMolecule mol = parser.getMolecule();

        Canonizer canonizer = new Canonizer(mol);
        String id = "ID-"+(row+1);
        String idcode = canonizer.getIDCode();
        String coords = canonizer.getEncodedCoordinates();

        tableModel.setTotalValueAt(id, row, 0);
    }
}
```

```
tableModel.setTotalValueAt(idcode, row, 1);
tableModel.setTotalValueAt(coords, row, 2);
}
    tableModel.finalizeTable(CompoundTableEvent.cSpecifierNoRuntimeProperties, null);

// the new JFrame() is also a hack, which can be replaced by null very soon
CompoundTableSaver saver = new CompoundTableSaver(new JFrame(), tableModel, null);
saver.saveNative(null, new File("someFileName.dwar"), false, false);
}
```
