

The CARA/Lua Programmers Manual

Amendment 1

to

Document NMR.014

2015-10-01

compatible with CARA 1.9.1.4

<http://www.cara.nmr.ch>

Copyright (c) 2003-2015 by Rochus Keller
All rights reserved

THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Introduction.....	3
Referenced Documents.....	3
Class Object.....	3
Object:setAttr.....	3
Class Repository.....	3
Repository:createResidueType.....	3
Repository:removeResidueType.....	3
Class ResidueType.....	3
ResidueType:createAtom.....	3
ResidueType:createAtomGroup.....	4
ResidueType:linkAtoms.....	4
ResidueType:removeAtom.....	4
ResidueType:removeAtomGroup.....	4
ResidueType:unlinkAtoms.....	4
Class StarBlock.....	4
StarBlock:getLoop.....	4
StarBlock:getLoopCount.....	4
StarBlock:getSaveFrame.....	4
StarBlock:getSaveFrames.....	5
StarBlock:getValue.....	5
StarBlock:getValues.....	5
StarBlock:isEmpty.....	5
Class StarFile.....	5
StarFile:getDataBlock.....	5
StarFile:getDataBlocks.....	5
StarFile:getError.....	5
StarFile:getGlobalBlock.....	5
StarFile:isNewSyntax.....	6
StarFile:isValid.....	6
Class StarLoop.....	6
StarLoop:getHeader.....	6
StarLoop:getHeaderCount.....	6
StarLoop:getTable.....	6
Class StarRef.....	6
StarRef:getBlock.....	6
StarRef:getFrame.....	7
StarRef:getItem.....	7
StarRef:getKey.....	7
StarRef:getSource.....	7
Class StarTable.....	7
StarTable:getRow.....	7
StarTable:getRowCount.....	7
StarTable:getTable.....	7
StarTable:getValue.....	7
Library star.....	8
star.openDocument.....	8
star.parseDocument.....	8

Introduction

This document is an amendmend to the CARA/Lua Programmers Manual Version 1.9.1 2015-10-01 reflecting the cumulative additions to the CARA/Lua API since its publication.

Referenced Documents

- [9] Spadaccini, N. and Hall, S. R.: Extensions to the STAR File Syntax. J. Chem. Inf. Model., 52 (8), 1901-1906 (2012)

Class Object

Object:setAttr

This method allows to store custom values in CARA objects. These values are persistent i.e. saved and loaded with the Repository.

Class Repository

Repository:createResidueType

Parameters

Nr.	Type	Opt.	Description
1	string		Full name
2	string		Short name (must be unique within Repository)
3	string		Letter

Returns

Nr.	Type	Opt.	Description
1	ResidueType		

Repository:removeResidueType

Parameters: ResidueType

Returns: none

Removes the ResidueType from the Repository. This is only possible if it is not referenced by a Residue, candidate Assignment or SystemType.

Class ResidueType

ResidueType:createAtom

Parameters

Nr.	Type	Opt.	Description
1	string		Atom name (HA, HB, etc.)
2	string		Isotope symbol (H1, N15, C13, etc.)
3	number	x	Number of atoms represented by this Atom (default 1)
4	AtomGroup	x	The AtomGroup to which the Atom belongs

Returns

Atom

ResidueType:createAtomGroup

Parameters: string (atom group name)

Returns: AtomGroup

ResidueType:linkAtoms

Parameters

<i>Nr.</i>	<i>Type</i>	<i>Opt.</i>	<i>Description</i>
1	Atom		
2	Atom		

Returns

none.

ResidueType:removeAtom

Parameters: Atom

Returns: none

ResidueType:removeAtomGroup

Parameters: AtomGroup

Returns: none

ResidueType:unlinkAtoms

Parameters

<i>Nr.</i>	<i>Type</i>	<i>Opt.</i>	<i>Description</i>
1	Atom		
2	Atom		

Returns

none.

Class StarBlock

This class represents either a `global_`, `data_` or `save_` cell of a STAR file (see [9]).

StarBlock:getLoop

Parameters: number (array index, default 1)

Returns: StarLoop

StarBlock:getLoopCount

Parameters: none

Returns: number (array length)

StarBlock:getSaveFrame

Parameters: string or QString (framecode)

Returns: StarBlock or nil

StarBlock:getSaveFrames

Parameters: none

Returns: Table[QString (framecode), StarBlock]

StarBlock:getValue

Parameters: string or QString (tag of the data item)

Returns: string or QString or Table or StarRef or nil

In 1991 format STAR files, all values are plain strings. The 2012 STAR format supports an extended character set, compound data values and reference tables (see [9] for details). Single 2012 format data values are represented by QString. Compound table and list values are both mapped to Lua tables. Reference tables are represented by StarRef objects. Note that QString objects can easily be converted to plain Lua strings using the built-in `toString()` Lua function.

StarBlock:getValues

Parameters: none

Returns: Table[QString (tag), string or QString or Table or StarRef]

See StarBlock:getValue for more information about values.

StarBlock:isEmpty

Parameters: none

Returns: boolean (true if the block doesn't contain any data items, save frames or loops)

Class StarFile

This class represents a STAR file (see [9]). See Fehler: Referenz nicht gefunden for how StarFile objects are created.

StarFile:getDataBlock

Parameters: string or QString (blockcode)

Returns: StarBlock or nil

StarFile:getDataBlocks

Parameters: none

Returns: Table[QString (blockcode), StarBlock]

StarFile:getError

Parameters: none

Returns: string (the error message of the parser when `isValid()` returns false)

See also StarFile:isValid.

StarFile:getGlobalBlock

Parameters: none

Returns: StarBlock

There is always a global StarBlock even if the STAR file didn't contain a `global_cell`. See also `StarBlock.isEmpty`.

StarFile:isNewSyntax

Parameters: none

Returns: boolean (true ... 2012 syntax, see [9], false ... 1991 syntax)

StarFile:isValid

Parameters: none

Returns: boolean (false in case of syntax errors, true otherwise)

Note that in case of false the object can nevertheless contain data. The parser runs up to the first syntax error and returns all data up to then.

Class StarLoop

This class represents the contents of a looped list (i.e. `loop_`) in a STAR file (see [9]). A loop is essentially a data table with header labels. STAR supports nested loops in which case there is more than one header level and potentially a (sub-) table per table row. In the BMRB file format (2.1 as well as 3.1 versions) there are solely loops with one level.

StarLoop:getHeader

Parameters: number (index, default 1)

Returns: Table[number (array index), QString]

StarLoop:getHeaderCount

Parameters: none

Returns: number (number of header levels)

StarLoop:getTable

Parameters: none

Returns: StarTable

Class StarRef

This class represents a STAR file data reference-value of the 2012 format version (see [9]). It is not used in 1991 format files.

StarRef:getBlock

Parameters: none

Returns: QString

StarRef:getFrame

Parameters: none

Returns: QString or Table

StarRef:getItem

Parameters: none

Returns: QString or Table

StarRef:getKey

Parameters: none

Returns: QString

StarRef:getSource

Parameters: none

Returns: QString

Class StarTable

This class represents a table of data values (i.e. *loop packet*) corresponding to a header level in a loop (see [9]).

StarTable:getRow

Parameters: number (array index)

Returns: Table[number (column), string or QString or Table or StarRef]

See StarBlock:getValue for more information about values.

StarTable:getRowCount

Parameters: none

Returns: number

StarTable:getTable

Parameters: number (array index, row number)

Returns: StarTable or nil

In case of nested loops each row of a table might itself contain a StarTable.

StarTable:getValue

Parameters

Nr.	Type	Opt.	Description
1	number		Row
2	number		Column

Returns

string or QString or Table or StarRef or nil

Library star**star.openDocument**

Parameters

<i>Nr.</i>	<i>Type</i>	<i>Opt.</i>	<i>Description</i>
1	string		Path to the STAR file
2	string	x	Encoding of the star file (default "UTF-8")
3	boolean	x	true ... 2012 STAR format, false (default) ... 1991 STAR format. Note that the BMRB format (2.1 as well as 3.1 versions) uses the 1991 STAR format only.
4	boolean	x	true (default) ... throw Lua error, false ... return in any case

Returns

StarFile

star.parseDocument

Parameters

<i>Nr.</i>	<i>Type</i>	<i>Opt.</i>	<i>Description</i>
1	string or QString		STAR formatted string (ASCII or UTF-8)
2	boolean	x	true ... 2012 STAR format, false (default) ... 1991 STAR format
3	boolean	x	true (default) ... throw Lua error, false ... return in any case

Returns

StarFile