aRTist API

2.12

Generated by Doxygen 1.8.17

Wed Mar 9 2022 17:24:24

Chapter 1

aRTist API Documentation

Scripting on aRTist - this document describes commands available in the aRTist console. The console can be opend from the tools menu ("`Tools*Show Console`").

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 3

Namespace Documentation

3.1 aRTist Namespace Reference

General behavior of program and GUI.

Functions

- GetVersion ?param?
- GetPid
- SetInfo text
- SetStatus text
- ShowDock
- HideDock
- ToggleToolbar
- ToggleDock
- ToggleSetup
- shutdown args
- RegisterInitCommand cmd ?priority?
- RegisterExitCommand cmd ?priority?

3.1.1 Detailed Description

General behavior of program and GUI.

3.1.2 Function Documentation

3.1.2.1 GetPid()

`aRTist::GetPid`

Return the aRTist process ID.

3.1.2.2 GetVersion()

```
aRTist::GetVersion
              ?param?
```

Return the aRTist version info.

- `param` If any, first letter of `param` is interpreted to return copyright string on 'c', version date on 'd', or product version on 'p'. Otherwise full version number is returned.

3.1.2.3 HideDock()

```
aRTist::HideDock
```

Hide docking area.

3.1.2.4 RegisterExitCommand()

```
aRTist::RegisterExitCommand
              cmd ?priority?
```

3.1.2.5 RegisterInitCommand()

```
aRTist::RegisterInitCommand
              cmd ?priority?
```

3.1.2.6 SetInfo()

```
aRTist::SetInfo
              text
```

3.1.2.7 SetStatus()

```
aRTist::SetStatus
              text
```

3.1.2.8 ShowDock()

```
aRTist::ShowDock
```

Show docking area.

3.1.2.9 shutdown()

```
aRTist::shutdown
              args
```

Quit aRTist execution, but only, if no calculation is running.

- `args` Optional argumants are: -restart (Restart aRTist after quitting.), or -force (Quit regardless of running calculations.)

3.1.2.10 ToggleDock()

```
aRTist::ToggleDock
```

Toggle visibility of docking area.

3.1.2.11 ToggleSetup()

```
aRTist::ToggleSetup
```

Toggle parameter panel.

3.1.2.12 ToggleToolbar()

```
aRTist::ToggleToolbar
```

Toggle toolbar visibility.

3.2 Engine Namespace Reference

aRTist's computation engine.

Functions

- StartStopCmd

3.2.1 Detailed Description

aRTist's computation engine.

3.2.2 Function Documentation

3.2.2.1 StartStopCmd()

`Engine::StartStopCmd`

Start/Stop a simulation run. Executed on pressing the run button.

3.3 FileIO Namespace Reference

Open any file that aRTist can handle.

Functions

- GetStandardDir
- OpenAny fn ?type?
- OpenAnyGUI filename ?type?
- SaveAnyCmd

3.3.1 Detailed Description

Open any file that aRTist can handle.

- Scene files (load scene)
- CAD parts (add part)
- Spectra (change source spectrum)
- Images (show in ImageViewer)
- Projects (Open project)

Check first for file extension, then do trial loads

3.3.2 Function Documentation

3.3.2.1 GetStandardDir()

```
FileIO::GetStandardDir
```

Return working directory (standard path). Configured in setting 'Default directory'.

3.3.2.2 OpenAny()

```
FileIO::OpenAny
              fn ?type?
```

Load file `fn` (of any aRTist file type).

3.3.2.3 OpenAnyGUI()

```
FileIO::OpenAnyGUI
              filename ?type?
```

Open file `filename` (of any file type). Don't throw errors but display dialogs. Return value true: loaded successfully

3.3.2.4 SaveAnyCmd()

```
FileIO::SaveAnyCmd
```

Write aRTist data to file. Save the current project/image/view/part/spectrum/detector/scene to file.

3.4 Image Namespace Reference

Image handling and prosecing.

Functions

- GetNumberOfComponents img
- GetStatistics img ?ignore?
- GetMean img ?ignore?
- GetIntegral img ?ignore?
- SetName img name

3.4.1 Detailed Description

Image handling and prosecing.

3.4.2 Function Documentation

3.4.2.1 GetIntegral()

```
Image::GetIntegral
              img ?ignore?
```

Return the integral of the image values.

Parameters

img	Image object.
ignore	If not 0 (true), do not consider zero valued pixel.

Returns

Integral value (count ∗ mean).

3.4.2.2 GetMean()

```
Image::GetMean
              img ?ignore?
```

Return the mean of the image values.

Parameters

img	Image object.
ignore	If not 0 (true), do not consider zero valued pixel.

Returns

Mean value.

3.4.2.3 GetNumberOfComponents()

```
Image::GetNumberOfComponents
              img
```

Return the number of scalar components.

Parameters

img	Image object.

Returns

Number of scalar components.

3.4.2.4 GetStatistics()

```
Image::GetStatistics
              img ?ignore?
```

Return statistics of the image values.

Parameters

img	Image object.
ignore	If not 0 (true), do not consider zero valued pixel.

Returns

Statistics dictionary with min, max, mean, count, and standard deviation values.

3.4.2.5 SetName()

```
Image::SetName
              img name
```

Set image name.

Parameters

img	Image object.
name	Image name.

3.5 Materials Namespace Reference

Materials handling.

Functions

- getall
- exists args

3.5.1 Detailed Description

Materials handling.

3.5.2 Function Documentation

3.5.2.1 exists()

```
Materials::exists
              args
```

Check if the material is present.

Usage: exists <material>

Returns 1 or 0.

3.5.2.2 getall()

```
Materials::getall
```

Returns the list of materils.

3.6 Math Namespace Reference

Mathamatical constants and functions.

Functions

- DegToRad x
- RadToDeg x
- abs x
- atan2 x y

3.6.1 Detailed Description

Mathamatical constants and functions.

3.6.2 Function Documentation

3.6.2.1 abs()

```
Math::abs
              x
```

Alternative to 'abs()' which allows NaN as argument returning NaN.

- x Input value.

3.6.2.2 atan2()

```
Math::atan2
             x y
```

Returns arc tangens of x/y in -Pi to Pi.

- x
- y

3.6.2.3 DegToRad()

```
Math::DegToRad
             x
```

Convert angles from degrees to radians.

- x Value in degrees.

3.6.2.4 RadToDeg()

```
Math::RadToDeg
             x
```

Convert angles from radians to degrees.

- x Value in radians.

3.7 Modules Namespace Reference

Module infrastructure.

Functions

- Scan
- Load mdir
- Available name
- Invoke name args
- Set args
- Run name
- UpdateMenu m
- GetAll
- GetActive
- GetCallable
- Get args
- CreatePkg name ?pkg?

3.7.1 Detailed Description

Module infrastructure.

3.7.2 Function Documentation

3.7.2.1 Available()

```
Modules::Available
              name
```

Check if the module is present. Returns 1 or 0.

3.7.2.2 CreatePkg()

```
Modules::CreatePkg
              name ?pkg?
```

3.7.2.3 Get()

```
Modules::Get
              args
```

Get module properties. (To read out variables, see 'Set'.)

Usage: Get [<module>] [<key>]

3.7.2.4 GetActive()

```
Modules::GetActive
```

3.7.2.5 GetAll()

```
Modules::GetAll
```

Return list of all available modules.

3.7.2.6 GetCallable()

```
Modules::GetCallable
```

Return list of modules that are active and graphical, i.e. that can be run by the user.

3.7.2.7 Invoke()

```
Modules::Invoke
              name args
```

Execute module subroutine.

Usage: Invoke <module> <command> [<arguments>]

3.7.2.8 Load()

```
Modules::Load
              mdir
```

3.7.2.9 Run()

```
Modules::Run
              name
```

Run module with the name 'name'. This typically opens a GUI from the module in another toplevel.

3.7.2.10 Scan()

```
Modules::Scan
```

3.7.2.11 Set()

```
Modules::Set
              args
```

Get/Set module variables. Usage: Set <module> <variable> [

]

3.7.2.12 UpdateMenu()

```
Modules::UpdateMenu
              m
```

3.8 PartList Namespace Reference

Assambly list managemnet.

Functions

- Query attrs args
- Invoke id method ?args?
- GetPos id
- GetIDFromPos pos
- Count
- LoadPart fn material ?name? ?center?
- SaveSelection fname
- DeleteCmd
- CopyClipboard
- PasteClipboard
- CutCmd
- SelectAll
- UnselectAll
- Select id
- Unselect id

3.8.1 Detailed Description

Assambly list managemnet.

Display and manage the list of objects (parts, source, detector).

Deal with selection, show GUI to trigger geometric transformations.

3.8.2 Function Documentation

3.8.2.1 CopyClipboard()

`PartList::CopyClipboard`

Copy selected parts to clipboard.

3.8.2.2 Count()

`PartList::Count`

Returns the number of parts in assembly list including source and detector.

Returns

Number of parts.

3.8.2.3 CutCmd()

```
PartList::CutCmd
```

Cut out selected parts.

3.8.2.4 DeleteCmd()

```
PartList::DeleteCmd
```

Delete all selected parts.

3.8.2.5 GetIDFromPos()

```
PartList::GetIDFromPos
              pos
```

Get part's ID from position in assembly list.

Parameters

pos	Position in assembly list (range: 0 ... Count-1).

Returns

ID of part.

3.8.2.6 GetPos()

```
PartList::GetPos
              id
```

Get part's assembly list position.

Parameters

id	ID of the part.

Returns

Position of part (0 ... Count-1).

3.8.2.7 Invoke()

```
PartList::Invoke
              id method ?args?
```

Call method of part object.

Parameters

id	ID of part, or `parts`, `all`, `selection`.
method	GetSize, ...
args	

3.8.2.8 LoadPart()

```
PartList::LoadPart
              fn material ?name?  ?center?
```

Load part geometries from file.

Parameters

fn	Name of file to load.
material	Name of the material of the part.
name	Label to be shown in assembly list (default: file name).
center	Position the part at world's origin instead of the original coordinates in the file (default: actual setting of toolbar item "Center new parts").

Returns

ID of loaded part.

3.8.2.9 PasteClipboard()

```
PartList::PasteClipboard
```

Paste parts from clipboard.

3.8.2.10 Query()

```
PartList::Query
              attrs args
```

Retrieve information about parts in assembly list (Playing SQL). Source & detector are sorted in front.

Parameters

attrs	List of attributes (ID, Name, Material, Token, Obj, Visible).
args	• `-with-setup` include source and detector • `-only-selected` only selected parts • `-only-visible` only visible parts • `-where` expr return only parts matching expr, e.g. `1 set mat "Fe"; PartList::Query {ID Name Material} -where {Material==$mat}`

Returns

List of attribute values.

3.8.2.11 SaveSelection()

```
PartList::SaveSelection
              fname
```

Save selected part geometries to file.

Parameters

fname	Name of file to save. (∗.stl/∗.ply/∗.vtk, default: VTK format)

3.8.2.12 Select()

```
PartList::Select
              id
```

Select a part.

- `id` ID of the part to add to selection.

3.8.2.13 SelectAll()

```
PartList::SelectAll
```

Select all parts including source and detector.

3.8.2.14 Unselect()

```
PartList::Unselect
              id
```

Cancel selection of a part.

- `id` ID of the part to remove from selection.

3.8.2.15 UnselectAll()

```
PartList::UnselectAll
```

Cancel selection for all parts.

3.9 Table2Lists Namespace Reference

Read in tabulated values.

Functions

- Read ?table? ?fname? ?nRows? ?nCols?
- Show ?table?
- Release ?table?

3.9.1 Detailed Description

Read in tabulated values.

Provide table columns as lits.

3.9.2 Function Documentation

3.9.2.1 Read()

```
Table2Lists::Read
              ?table?  ?fname?  ?nRows?  ?nCols?
```

Read table from ASCII file.

Reads in an ASCII file and interprets it as a table with numerical values. Lines not starting with a numerical value will be skipped. The values must be separated by one of the following characters: comma, semicolon, space, or tab. The dot character is interpreted as decimal separator.

Parameters

table	Variable of the table. (default: ::Table)
fname	File name (default: aks for file name).
nRows	Max number of rows to read in (default: 0 <read all>).
nCols	Max number of columns to read in (default: 0 <read all>).

3.9.2.2 Release()

```
Table2Lists::Release
              ?table?
```

Delete table

3.9.2.3 Show()

```
Table2Lists::Show
              ?table?
```

Print list names

3.10 TkconClient Namespace Reference

Tkcon remote control over TCP sockets.

Functions

- start ?myport? ?myaddr?
- stop
- status

3.10.1 Detailed Description

Tkcon remote control over TCP sockets.

The aRTist console can be accessed by a telnet client application connecting to the activated socked, e.g. command line '# telnet localhost:3658'. While Windows Command Prompt telnet client seams not to work, CYGWIN/Linux telnet or PuTTY are nice clients for testing.

Warning

There is no access control to the socket!

Since

2.10.0

3.10.2 Function Documentation

3.10.2.1 start()

```
TkconClient::start
              ?myport?  ?myaddr?
```

Starts offering of a TCP socket for remote access to the TCL console.

- `port` Port number. The default value for `port` is `3658`.
- `myaddr` Hostname/IP address of the machine this service is started on. The default value for `myaddr` is `localhost`. If the port can't be opened, additional tries with incremented port numbers will be made.

3.10.2.2 status()

```
TkconClient::status
```

Returns the socket status.

3.10.2.3 stop()

```
TkconClient::stop
```

Shut down the server socket for remote access, and the current client connection if there exists one.

Note

To shut down a client connection from the remote side, the word `bye` can be sent as a pseudo-command.

3.11 Voxelizer Namespace Reference

Voxelize a facetted part geometry.

Functions

- GetDataType
- SetDataType ?Type?
- GetFileType
- SetFileType ?Type?
- Voxelize ?VoxelSize?

3.11.1 Detailed Description

Voxelize a facetted part geometry.

By positioning the source far from the geometry a near parallel beam is prepared. Then the geometry is scanned by shifting the detector in steps of voxel layer distance. The difference of penetrated length from the previous step is stored as voxel layer.

This is a optional feature in Tools (Tools==>Voxelize). Further information is available from artist@bam.de.

3.11.2 Function Documentation

3.11.2.1 GetDataType()

```
Voxelizer::GetDataType
```

Get data type of output volume (8bit (default), 16bit ,32bit, float).

3.11.2.2 GetFileType()

```
Voxelizer::GetFileType
```

Get file type for output volume (RAW (default), TIFFs).

3.11.2.3 SetDataType()

```
Voxelizer::SetDataType
              ?Type?
```

Set data type of output volume.

Parameters

Type	Voxel data type (8bit (default), 16bit ,32bit, float).

3.11.2.4 SetFileType()

```
Voxelizer::SetFileType
              ?Type?
```

Set file type for output volume.

Parameters

Type	File type (RAW (default), TIFFs).

3.11.2.5 Voxelize()

```
Voxelizer::Voxelize
              ?VoxelSize?
```

Voxelize a selected part. Considers the volume of the first selected part only.

Voxel value unit is dependent on DataType: Float: material length in Z direction in mm 32bit: material length in Z direction in µm (integer) 8bit, 16bit: percentage of material filling, 0 (empty) . . . 100 (entirely filled)

Parameters

VoxelSize	Voxel size in mm (default: -1; use actual size of detector pixels as voxel size).

3.12 XRayProject Namespace Reference

Functions for reading / writing project files.

Functions

- SaveProjectCmd
- LoadProject fn

3.12.1 Detailed Description

Functions for reading / writing project files.

Project files (.aRTist) are zip archives with a certain structure:

main.scene – scenefile folder1/ folder2/ ...

Registered handlers take care of a certain folder upon saving, the handlers are called before the main.scene file is written with the foldername and a dictionary of component->temporary path mappings It must return a similar list with mappings. If the pathnames are unchanged, they are assumed to be accepted from the handler. If they are different, the files pointed to are included.

upon loading, the handlers are notified after the main.scene file is loaded

3.12.2 Function Documentation

3.12.2.1 LoadProject()

```
XRayProject::LoadProject
              fn
```

Load project from file `fn` (.aRTist).

3.12.2.2 SaveProjectCmd()

```
XRayProject::SaveProjectCmd
```

Save project to file (.aRTist). Pops up a dialog asking for a file name, in case of a new project.

3.13 XSource Namespace Reference

Source spectra handling.

Functions

- ComputeSpectrum
- CheckSpectrum ?parent?
- ClearSpot
- LoadSpot ?fname? ?size?

3.13.1 Detailed Description

Source spectra handling.

3.13.2 Function Documentation

3.13.2.1 CheckSpectrum()

```
XSource::CheckSpectrum
              ?parent?
```

Reduce the number of bins, if it exceeds the recomended amount.

If <parent> is provided, a graphical dialog will pop up berore changing the spectra.

3.13.2.2 ClearSpot()

```
XSource::ClearSpot
```

Delete spot profile image.

3.13.2.3 ComputeSpectrum()

```
XSource::ComputeSpectrum
```

Run spectrum model

Call XRayTools to compute and load a X-ray spectrum.

Parameters and defaults:

- Xsource(AngleIn) =
- Xsource(FilterMaterial) = void
- Xsource(FilterThickness) = 0
- Xsource(Filter2Material) = void
- Xsource(Filter2Thickness) = 0
- Xsource(Filter3Material) = void
- Xsource(Filter3Thickness) = 0
- Xsource(Resolution) = 1.0
- Xsource(TargetAngle) = 45
- Xsource(TargetMaterial) = W
- Xsource(TargetThickness) = 1
- Xsource(Transmission) = 0
- Xsource(Tube) = Mono
- Xsource(Voltage) = 150
- Xsource(WindowMaterial) = Al
- Xsource(WindowThickness) = 4

3.13.2.4 LoadSpot()

```
XSource::LoadSpot
              ?fname?  ?size?
```

Read focal spot image form file.

Parameters

fname	Image file path. If empty a dialog will ask for the file name.
size	If "yes", the spot size is adapted to the size of the image. With "ask" this is queried with a dialog.

Index