Velocity Python API

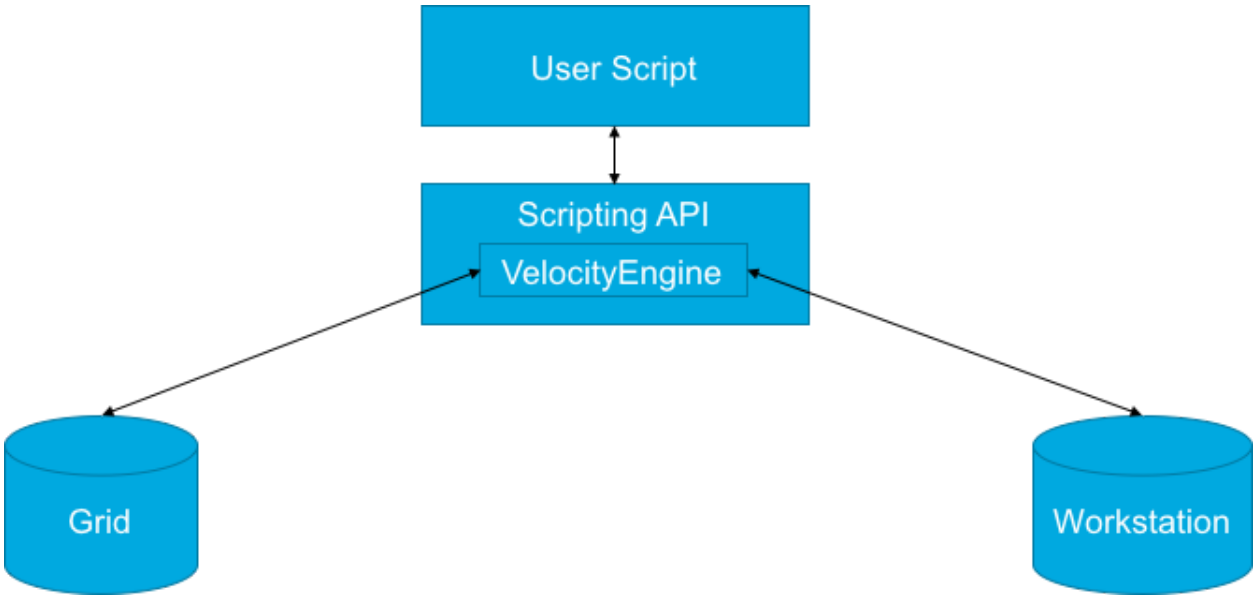
for Velocity 4.1.4

Last Edited 03/03/2021

Table of Contents

Architecture Overview

The Velocity scripting API is defined by an interface called the Velocity Engine. The engine is able to access both Grid and Workstation databases and functions as a client without a graphical user interface. The Scripting API exposes the engine to the particular scripting language (C# or Python), and user scripts written in the scripting language use the engine to drive Velocity functionality.



Return to Table of Contents

Command Prompt and Python Interpreter Basics

- Using Command Prompt vs Python Interpreter
    - >
        - Indicates that the following should be typed into the command prompt
    - >>>
        - Indicates that the following should be typed into the python interpreter
- Command Prompt Commands
    - > cd
        - Allows the user to change their working directory
        - Absolute Path
            - A path that starts from the root directory
            - This path must be used if your working directory is not part of the path to get to the desired directory
            - Example
                - > cd C:\absolute\path\to\directory
        - Relative Path
            - A path that starts from the working directory
            - This path can be used if your working directory is part of the path to get to the desired directory
            - Example
                - > cd \relative\path\to\directory
    - > python
        - Allows the user to open the python interpreter
        - If the following command is followed by a .py file, the file will be executed (otherwise, it will just open the python interpreter)
            - Example
                - > python my_code.py
    - Arrow Keys

        - The up and down arrow keys can be used to navigate through previously entered commands
- Python Interpreter Commands
    - >>> quit()
        - Allows you to exit the python interpreter and return to the command prompt

Download and Installation

This information is also included in the README file provided in the zip package. We recommend you follow the instructions in the file.

Velocity is packaged as a [Python wheel] (https://pypi.org/project/wheel/) that you can install into your environment. We currently require Python 2.7 and you must use the standard CPython interpreter. We support Windows and MacOS.

- Setting up python
    - Download latest version of Python 2.7 (currently 2.7.15)
        - https://www.python.org/downloads/windows/
        - Windows x86-64 MSI installer
- Setting up pip
    - Pip should already be downloaded with python
        - To check if pip is already downloaded
            - > pip --version
    - If pip is not already download
        - https://pip.pypa.io/en/stable/installing/
- Setting up virtualenv
    - To download virtualenv
        - > pip install virtualenv
    - To create a virtualenv (named “velocity-env”)
        - > virtualenv velocity-env
    - To activate the virtualenv
        - > velocity-env\Scripts\activate
- Downloading velocity
    - Download the Velocity API mac or win zip package for python provided
    - To install velocity
        - > pip install velocity-<version_number>.whl

Return to Table of Contents

Editing and Running Example Scripts

- Editing scripts to run
    - Open the script you want to change in your editor
    - Find the variable DB_NAME and change it to the database name
        - Example
          DB_NAME = 'VMS_GRID'
    - Find the DB_USER and DB_PASS and change it to your username and password
        - Example
          DB_USER = 'script'
          DB_PASS = 'script'
    - The DB_IP and DB_PORT should already be correct
        - DB_IP = '127.0.0.1'
        - DB_PORT = 57000
    - Save the file
- Run the script
    - Type the following into the command prompt
        - > python file_name.py
- Fixing errors
    - RuntimeError: Non-script users may not log in from scripts
        - Log into velocity and go to settings
        - Click on the users tab and create a new user
        - After creating the new user, change the user type to Script
        - Use the username and password created for this user for DB_USER and DB_PASS
    - RuntimeError: Could not connect to active database at : *DB_IP* (*DB_PORT*). DbName=*DB_NAME*
        - This means that the DB_NAME is incorrect
    - RuntimeError: No response from Grid database (*DB_IP*:*DB_PORT*/*DB_NAME*): Resource temporarily unavailable
        - This means that the DB_IP is incorrect or the DB_PORT is incorrect
    - RuntimeError: Could not find patient with id: *patientId*
        - The specific patient that the script is written for needs to be imported to velocity in order to get the script to run
        - The patientId specifies which patient the script is written for

Example Scripts

- BED.py
    - Runs Biological Effective Dose (BED) calculation
- Jacobian_structure_mask.py
    - Creates a deformable registration using default settings.
    - Uses the Jacobian and other volumetric information to create a structure mask that identifies rigid areas like bone that have been deformed under the registration.
    - Converts a structure mask into a structure.
    - Applies post processing operations like margin to create a new structure.
    - Creates a structure set with two structures described above.
- Liver_import.py
    - Tests import operations DICOM file import.
- Pet_ct_registrations.py
    - Tests running a manual registration, rigid registration, and performing a deformable registration using customized settings.
- Print_patient.py
    - Prints out all information and data on the patient.
- Structure_metrics.py
    - Copies structures using a deformable.
    - Calculates minimum, median, mean, and standard deviation values, comparing them across the deformable or with the proceeding rigid.

Return to Table of Contents

Writing Scripts

- Required components for writing velocity scripts
    - Import statements
        - import velocity
            - This statement will import all velocity classes
        - import atexit
            - This statement will import the atexit module, which assists with properly closing the program after the script has finished running
    - Attributes
        - DB_NAME
            - This attribute can be named whatever you choose, but it will represent the name of the database that is used to login to the velocity grid
            - This attribute should be a constant
                - Although it is not necessary put the name in all capitalization, it is suggested to use capitalization to signify that the attribute is a constant
            - Example
                - DB_NAME = r'VMS_GRID'
                    - The r is used in order to ensure that if there any escape characters, (a backslash) these characters will be registered as is, rather than as an escape character
        - DB_USER
            - This attribute can be named whatever you choose, but it will represent the username that is used to log into the velocity grid
            - The username should not be the generic username used to login to velocity, but should be the username that was created when you created a Script user for velocity
            - This attribute should be a constant
            - Example
                - DB_USER = 'script'
        - DB_PASS
            - This attribute can be named whatever you choose, but it will represent the password that is used to log into the velocity grid

- The password should not be the generic password used to login to velocity, but should be the password that was created when you created a Script user for velocity
- This attribute should be a constant
- Example
  - DB_PASS = 'script'

■ DB_IP
- This attribute can be named whatever you choose, but it will represent the IP address for the velocity grid
- Example
  - DB_IP = '127.0.0.1'

■ DB_PORT
- This attribute can be named whatever you choose, but it will represent the port number for the velocity grid
- Example
  - DB_PORT = 57000

■ PATIENT_ID
- This attribute can be named whatever you chose, but it will represent the id of the patient that is being used to run the script
- Example
  - PATIENT_ID = 'AW3Y6TA684'

○ Creating the velocity engine

■ e = velocity.VelocityEngine()
def orThrow(c, e=e):
    if not c or (hasattr(c, 'isValid') and not c.isValid()):
        raise RuntimeError(e.getErrorMessage())
- This block of code is used to create the velocity engine and checks any errors that occur when creating the engine
- Determines if the attempt to create the engine is valid using the isValid() method from the PatientDataItem class
- Uses the getErrorMessage() method from the VelocityEngine class if an error occurs when trying to create the engine

○ Logging in to and out of the velocity grid

■ orThrow(e.loginToGrid(DB_USER, DB_PASS, DB_IP, DB_PORT, DB_NAME))
atexit.register(e.logout)

        - Logs into your velocity grid using the loginToGrid() method from the VelocityEngine class and using your username, password, ip address, port number, and grid database
            - The username, password, etc. must be in the exact same order as listed above in order to successfully login
        - Logs out of the velocity grid after the script has finished running using the register() method from the atexit module
- Other components for writing velocity scripts
    - Based on what you want your script to test, you will most likely need to include other attributes, methods, and print statements
    - Most of the additional methods you will need to use will be from the VelocityEngine class; however, other methods may be needed
    - Additional attributes will need to be created in order to fill the parameters of the methods that you are testing
    - Scripts can be used to test a singular method to make sure it works properly, or used to test the interaction between multiple methods
    - Examine other scripts in order to get an idea of what is needed to write additional scripts
    - An example script with comments is provided for you on the next page to help visualize how to write your own scripts
        - This script can be copied and pasted into your editor as a .py file
        - A "#" is used to denote that the following information is a comment, not an actual part of the code

```
# The following script is used as an example to get you started.
# In order to create a script, you will need to create a .py file. You can name
# the .py file whatever you want. I have chosen to name my .py file MyScript.py.
# Place the MyScript.py script in the velocity-examples folder. Edit the parameters #
to reflect your Velocity settings (db path/ip, credentials, etc.) and run the
# script:
#      python MyScript.py

# The following script uses a patient that already exists in the database (to
# import a patient see the example liver_import.py), loads the patient, and then
# prints out the patient's full name and date of birth. The script tests the
# loadPatientByPatientId() and patientDataOps() methods from the VelocityEngine
# class, the getPatientByPatientId() method from the PatientDataOperations class,
# and the getFirstName() and getLastName() methods from the Patient class.

import velocity
import atexit

# Attributes
DB_NAME = r'VMS_GRID'
DB_USER = 'script'
DB_PASS = 'script'
DB_IP = '127.0.0.1'
DB_PORT = 57000

# Patient VMS123 must be loaded for the script to work
PATIENT_ID = "VMS123"

# Creates the velocity engine
e = velocity.VelocityEngine()
def orThrow(c, e=e):
       if not c or (hasattr(c, 'isValid') and not c.isValid()):
              raise RuntimeError(e.getErrorMessage())

# Login to velocity grid
orThrow(e.loginToGrid(DB_USER, DB_PASS, DB_IP, DB_PORT, DB_NAME))
# Logout of velocity grid when script is done running
atexit.register(e.logout)

# Loads the patient given the patient ID
orThrow(e.loadPatientByPatientId(PATIENT_ID))
print('Loaded patient: {}'.format(PATIENT_ID))
```

```
# Allows the script to use the patient data operations
patientDataOps = e.getPatientDataOperations()
# Assigns the patient to a variable called patient
patient = patientDataOps.getPatientByPatientId(PATIENT_ID)
# Determines the patient's full name
patientFullName = patient.getFirstName() + " " + patient.getLastName()
print('Patient {} is named {}.'.format(PATIENT_ID, patientFullName))
# Determines the patient's date of birth
patientDOB = patient.getDateOfBirth()
print('Patient {} was born {}.'.format(PATIENT_ID, patientDOB))
```

Viewing Documentation

- Entire Velocity Documentation
    - >>> import velocity
    - >>> help(velocity)
- Specific Class Documentation
    - >>> import velocity
    - >>>help(velocity.Class_Name)

Return to Table of Contents

__builtin__.SwigPyObject

- SwigPyObject is an implementation detail. All Velocity classes inherit SwigPyObject but scripting API users **should not use** SwigPyObject methods. These methods are:
    - acquire()
    - append()
    - disown()
    - next()
    - own()

Return to Table of Contents

Velocity Classes

- List of all classes
  - Base Registration Settings Structure
    - BaseImageBasedRegistrationSettingsStructure
      - BSplineDeformableRegistrationSettingsStructure
      - RigidRegistrationSettingsStructure
    - ManualRegistrationSettingsStructure
  - BoolList
  - CharList
  - CropSettings
  - DoubleList
  - ExportOperations
  - FloatList
  - Image2DList
  - ImportOperations
  - IntList
  - MarginSettings
  - MatrixR22d
  - MatrixR22i
  - MatrixR33d
  - MatrixR33f
  - MatrixR44d
  - MatrixR44f
  - PatientDataItem
    - Image2D
    - Patient
    - Plan
    - Registration
    - RegistrationCommissioningMarkerList
    - Structure
    - StructureSet
    - Volume
  - PatientDataOperations
  - PatientList
  - PlanList
  - RegistrationCommissioningMarkerLists
  - RegistrationList
  - RegistrationMarker

Return to Table of Contents

- RegistrationMarkerList
- RegistrationOperations
- ReportOperations
- ScaledVolumeData
- StringList
- StructureList
- StructureMap
- StructureOperations
- StructureSetList
- SurfaceDistanceMetrics
- SwigPyIterator
- TransformedRegistrationMarker
- VectorR2b
- VectorR2bList
- VectorR2d
- VectorR2dList
- VectorR2f
- VectorR2fList
- VectorR2i
- VectorR2iList
- VectorR2s
- VectorR2sList
- VectorR2ub
- VectorR2ubList
- VectorR2ui
- VectorR2uiList
- VectorR2us
- VectorR2usList
- VectorR3b
- VectorR3bList
- VectorR3d
- VectorR3dList
- VectorR3f
- VectorR3fList
- VectorR3i
- VectorR3iList
- VectorR3s
- VectorR3sList
- VectorR3ub

Return to Table of Contents

BaseImageBasedRegistrationSettingsStructure

- Attributes and data descriptors
    - preprocessingMethods
        - Attribute type
            - PreprocessingFilterMethod object values:
                - NoFilter = 0
                - MRCorrectionPrimary = 1
                - MRCorrectionSecondary = 2
                - CBCTCorrectionSecondary = 4
        - Description
            - In order to increase registration accuracy the user may choose to apply one or more pre-processing filters. These filters are modality specific and they are applied prior to execution of the registration algorithm.
    - primaryEndLevel, primaryStartLevel, secondaryEndLevel, secondaryStartLevel
        - Attribute type
            - Double
        - Description
            - These variables specify the dynamic ranges that will be used in the registration process. The values are expected to be given in the units specific to the volume modality. For CT or CBCT volumes – units are typically given in Hounsfield Units. Use a large range to use the entire range of the volume (e.g. Start Level = -1024, End Level = 3072). To focus on a specific tissue range the user can specify a smaller range, for example, “Start Level = -120” and “End Level = 400” HU.
    - roiEnd, roiStart
        - Attribute type
            - VectorR3d object
        - Description
            - The roiStart (lowest point) and roiEnd (highest point) define the Region of Interest (ROI) cuboid which will be used in the registration process. Note that the expected coordinates for the values are in the DICOM coordinates of the primary volume.

Return to Table of Contents

- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

BoundingRegion

- Attributes and data descriptors
    - isValid
        - Attribute type
            - bool
        - Description
            - Specifies if the region is valid
    - start
        - Attribute type
            - VectorR3d
        - Description
            - The starting corner, in the respective lower coordinate of each dimension.
    - end
        - Attribute type
            - VectorR3d
        - Description
            - The ending corner, in the respective higher coordinate of each dimension.

Return to Table of Contents

BSplineDeformableRegistrationSettingsStructure

- Attributes and data descriptors
    - numberOfMultiResolutionLevels
        - Attribute type
            - Integer
        - Description
            - Specifies the number of hierarchical levels that are to be used by the multi-resolution deformable image registration algorithm. Velocity uses 1 level for the single B-Spline DIR, 3 levels for the Multi-pass B-Spline, and 5 levels for the Extended Multi-pass DIR. Note, all other variables in the structure contain multiple values – one for each level in the hierarchical process (so the value of 'numberOfMultiResolutionLevels' determines the size of all other variables.
    - applyBoundaryContinuityConstrains
        - Type
            - vector < bool >
        - Description
            - Specifies whether to establish a smooth deformation vector field (DVF) at the boundary between the ROI and the background. if 'true' then the DVF will extend beyond the currently selected ROI so that a smooth transition region is constructed between the motion (ROI) and the non-motion (background) regions. Note, a smooth boundary is also constructed between overlapping ROIs in cases where the DIR is restarted and different ROIs are selected between restarts. The default value is set to 'false'.
    - applyTopologicalRegularizer
        - Type
            - vector < bool >
        - Description
            - Enables or disables the regularizer algorithm in the gradient descent optimizer. The default value is set to 'false'.
    - gradientMagnitudeTolerance
        - Attribute type

            - vector < double >
        - Description
            - Used by the optimizer algorithm as part of the early convergence criteria. If the maximum gradient computed during an iteration step is less than this threshold then the optimization process is terminated. The default value is 5e-21.
- gridCellSize
    - Attribute type
        - vector < VectorR3d >
    - Descripton
        - Used to define the B-Spline control lattice along with gridCellSizeType. This 3D lattice can be defined either by specifying the number of control points or by specifying a physical grid size.
- gridCellSizeType
    - Attribute type
        - vector < char >
    - Description
        - Used to define the B-Spline control lattice along with gridCellSize. To define the lattice by the total number of control points then specify a value of ‘n’ in in this variable followed by the number of controls points in “gridCellSize”. To define the lattice by an explicit physical grid size then specify a value of ‘p’ in “gridCellSizeType” followed by the physical space between control points (in meters) in “gridCellSize”. Specifying the number of control points to use may create an asymmetric grid spacing (depending on the ROI selected).Specifying a physical grid size will cause the control point count to be “adjusted” to adhere to this size (the originally specified ROI may be adjusted to meet the specified physical grid size requirements).
        - The default values are the following:

            - gridCellSizeType = {‘n’,’n’,’n’}
            - gridCellSize = { {5,5,5}, {10,10,10}, {15,15,15} }

- maximumNumberOfIterations

        - Attribute type
            - vector < int >
        - Description
            - Specifies the maximum number of iterations that may be used with the optimizer. The default value is 30. Note, the optimizer will terminate prior to reaching this maximum iteration count if any of the other convergence criteria are met.
    - maximumNumberOfConsecutiveOptimizerAttempts
        - Attribute type
            - vector < int >
        - Description
            - Used to establish additional termination criteria for the optimizer (this is separate from maximum number of iterations). During the convergence criteria, if no progress has occurred (the progress is defined by “metricValuePercentageDifference”) within the specified number of iteration attempts then we assume that we’ve reached convergence and the optimization process is terminated. Default value is 10.
    - metricValuePercentageDifference
        - Attribute type
            - vector < double >
        - Description
            - Used in the convergence criteria. This value is meant to identify that progress is being made during the optimization process. First the percentage difference of the metric results between the last two iteration steps in the optimization process are computed. If this percentage difference is greater or equal to this variable then we conclude that optimization progress is currently being made (the valid value range is between 0 ~ 100, the default value is ‘0’ (disabled)).
    - maximumStepLength
        - Attribute type
            - vector < double >
        - Description

            - Used to establish convergence criteria for the optimizer. The value specifies the maximum step length (in millimeters). The default value is 100.
    - minimumStepLength
        - Attribute type
            - vector < double >
        - Description
            - Used to establish convergence criteria for the optimizer. The value specifies the minimum step length (in millimeters). The default value is 1e-6.
    - numberOfHistogramBins
        - Attribute type
            - vector < int >
        - Description
            - It defines the number of bins in the histogram of volume values. The default is 50.
    - relaxationFactor
        - Attribute type
            - vector < double >
        - Description
            - Used to establish convergence criteria for the optimizer. The default value is 0.9 .
    - samplesDenominator
        - Attribute type
            - vector < int >
        - Description
            - Specifies a reduction in the total number of samples to be used in computing optimization metrics. For example, if the value is set to 1 then all of the pixels from the ROI are used, however, if the value is set to 2 then only half of the pixels from the ROI are used. The default value is 5.
    - topologicalRegularizerDistanceLimitingCoefficient
        - Attribute type
            - vector < VectorR3d >
        - Description
            - The regularization values used by the regularization algorithm in the gradient descent optimizer (these are used only if a value of ‘true’ is specified in the

corresponding level in “applyTopologicalRegularizer”). An example of an acceptable value is 0.333.

- Inherited data descriptors
    - From BaseImageBasedRegistrationSettingsStructure
        - preprocessingMethod
        - primaryEndLevel
        - primaryStartLevel
        - roiEnd
        - roiStart
        - secondaryEndLevel
        - secondaryStartLevel
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

RigidRegistrationSettingsStructure

- Attributes and data descriptors
    - disableRotationsX
        - Attribute type
            - Boolean
        - Description
            - Used to disable rotations along the X axis. Default value is false.
    - disableRotationsY
        - Attribute type
            - Boolean
        - Description
            - Used to disable rotations along the Y axis. Default value is false.
    - disableRotationsZ
        - Attribute type
            - Boolean
        - Description
            - Used to disable rotations along the Z axis. Default value is false.
    - disableTranslationsX
        - Attribute type
            - Boolean
        - Description
            - Used to disable translations along the X axis. Default value is false.
    - disableTranslationsY
        - Attribute type
            - Boolean
        - Used to disable translations along the Y axis. Default value is false.
    - disableTranslationsZ
        - Attribute type
            - Boolean
        - Description
            - Used to disable translations along the Z axis. Default value is false.
    - maximumNumberOfIterations

        - Attribute type
            - Integer
        - Description
            - The following variable specifies the maximum number of iterations that are to be used with the gradient descent optimizer. The default value is 45.
    - maximumStepLength
        - Attribute type
            - Double
        - Description
            - The maximum step length (in millimeters) that will be used with the gradient descent optimizer. The relaxation factor used in conjunction with the min/max values is set to 0.5. The default value is 7.
    - minimumStepLength
        - Attribute type
            - Double
        - Description
            - The minimum step length (in millimeters) that will be used with the gradient descent optimizer. The relaxation factor used in conjunction with the min/max values is set to 0.5. The default is 0.0001.
    - numberOfHistogramBins
        - Attribute type
            - Integer
        - Description
            - It defines the number of bins in the histogram of volume values. The default value is 25.
    - performInitialAutoAlignment
        - Attribute type
            - Boolean
        - Description
            - Prior to any rigid registration the user can specify to perform an initial auto-alignment between the two volumes (this is to quickly get the two volumes closer in alignment before executing the main rigid registration algorithm). The default value is ‘true’.
    - samplesDenominator
        - Attribute type

            - Integer
        - Description
            - The following variable defines a reduction in the total number of pixels from the currently specified ROI. For example, if the value is set to 1 then all of the pixels from the ROI are used, however, if the value is set to 2 then only half of the pixels from the ROI are used. The default value is 10.
- Inherited data descriptors
    - From BaseImageBaswedRegistrationSettingsStructure
        - preprocessingMethod
        - primaryEndLevel
        - primaryStartLevel
        - roiEnd
        - roiStart
        - secondaryEndLevel
        - secondaryStartLevel
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

ManualRegistrationSettingsStructure

- Attributes and data descriptors
    - registrationMatrix
        - Attribute type
            - MatrixR44d object
        - Description
            - Specifies a 4x4 transformation matrix for performing a manual alignment between the primary and secondary volumes. The matrix is expected in column-major order and in DICOM coordinates. The default value is the identity matrix.
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

Return to Table of Contents

DefaultRigidRegistrationSettings

- Attributes and data descriptors
    - performInitialAutoAlignment
        - Attribute type
            - Boolean
        - Description
            - Prior to any rigid registration the user can specify to perform an initial auto-alignment between the two volumes (this is to quickly get the two volumes closer in alignment before executing the main rigid registration algorithm). The default value is ‘true’.
    - preprocessingMethod
        - Attribute type
            - PreprocessingFilterMethod object values:
                - NoFilter = 0
                - MRCorrectionPrimary = 1
                - MRCorrectionSecondary = 2
                - CBCTCorrectionSecondary = 4
        - Description
            - In order to increase registration accuracy the user may choose to apply one or more pre-processing filters. These filters are modality specific and they are applied prior to execution of the registration algorithm. Default is NoFilter.
    - roiEnd, roiStart
        - Attribute type
            - VectorR3d object
        - Description
            - The roiStart (lowest point) and roiEnd (highest point) define the Region of Interest (ROI) cuboid which will be used in the registration process. Note that the expected coordinates for the values are in the DICOM coordinates of the primary volume.
        - Default value is the intersection of the volumes

Return to Table of Contents

DefaultBSplineDeformableRegistrationSettings

- type
  - Attribute type
    - DeformableRegistrationType object values
      - DeformableSinglePass = 1
      - DeformableMultiPass = 3
      - DeformableExtendedPass = 5
  - Default value is DeformableMultiPass
  - Description
    - Controls the type of deformable registration that will be performed.
- preprocessingMethod
  - Attribute type
    - PreprocessingFilterMethod object values:
      - NoFilter = 0
      - MRCorrectionPrimary = 1
      - MRCorrectionSecondary = 2
      - CBCTCorrectionSecondary = 4
  - Description
    - In order to increase registration accuracy the user may choose to apply one or more pre-processing filters. These filters are modality specific and they are applied prior to execution of the registration algorithm. Default is NoFilter.
- roiEnd, roiStart
  - Attribute type
    - VectorR3d object
  - Description
    - The roiStart (lowest point) and roiEnd (highest point) define the Region of Interest (ROI) cuboid which will be used in the registration process. Note that the expected coordinates for the values are in the DICOM coordinates of the primary volume.
  - Default value is the intersection of the volumes

Return to Table of Contents

BedDoseCalculationData

- Data descriptors
    - bedVariablesByStructure
        - Attribute Type
            - BedStructureVariablesList
        - Description
            - A list of BedStructureVariables objects
    - fractions
        - Attribute Type
            - integer
        - Description
            - Number of fractions in the plan
    - totalTimeTreatmentInSecs
        - Attribute Type
            - Double
    - kickOffTimeInSecs
        - Attribute Type
            - Double
    - double totalRepopTimeInSecs
        - Attribute Type
            - Double
    - _lambda
        - Attribute Type
            - Double
    - expDecayMode
        - Attribute Type
            - Boolean
    - xValue
        - Attribute Type
            - Double

Return to Table of Contents

BedBackgroundData

- Data descriptors
    - alphaBetaRatio
        - Attribute Type
            - Double
        - Description
            - alphaBetaRatio for the unspecified tissue
    - alpha
        - Attribute Type
            - Double
    - repopulationTime
        - Attribute Type
            - Double
    - repopulationSelected
        - Attribute Type
            - Double
    - mu
        - Attribute Type
            - Double

Return to Table of Contents

BedStructureVariables

- Data descriptors
    - structureName
        - Attribute Type
            - String
        - Description
            - Name of the structure
    - structureId
        - Attribute Type
            - Integer
        - Description
            - Database id for the structure
    - alphaBetaRatio
        - Attribute Type
            - Double
        - Description
            - alphaBetaRatio for the unspecified tissue
    - alpha
        - Attribute Type
            - Double
    - repopulationTime
        - Attribute Type
            - Double
    - repopulationSelected
        - Attribute Type
            - Double
    - mu
        - Attribute Type
            - Double

Return to Table of Contents

CropSettings

- Data descriptors
    - margin
        - Attribute Type
            - Double
        - Description
            - Margin value
    - marginDirection
        - RemoveInside = 0
        - RemoveOutside = 1
    - operation
        - CropMarginNone = 0
        - CropMarginGrow =1
        - CropMarginShrink = 2
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

Return to Table of Contents

ExportOperations

- Attributes
    - DICOM_REG
        - Value
            - 2
    - DICOM_RTPLAN
        - Value
            - 1
    - DICOM_RTSTRUCT
        - Value
            - 3
    - DICOM_VOLUME
        - Value
            - 0
- Other defined methods
    - documentNameStrFilenameFormatted(reportRecordId)
        - Return type
            - String
        - Description
            - Returns the name of the specified document as a formatted string
            - Calls getErrorMessage() method and returns an empty string if there is an error
    - exportDicomObject(objectType, objectUID, folderPath, overrideWithInfoFromDB = false, encodingType = ORIGINAL_ENCODING)
        - Return type
            - Boolean
        - Description
            - Exports a dicom object to the specified folder path
    - exportReportDocument(reportRecordId, baseFilenameExport, baseFolderPathExport)
        - Return type
            - Boolean
        - Description
            - Exports the report to the specified folder path
            - Returns false if there is an error
    - getErrorMessage()

Return to Table of Contents

        - Return type
            - String
        - Description
            - Returns a string displaying an error message
    - getLocalizedErrorMessage()
        - Return type
            - String
        - Description
            - Returns a string displaying an error message
    - patientIdStrFilenameFormatted()
        - Return type
            - String
        - Description
            - Returns the patient id as a formatted string
            - Calls the getErrorMessage() method and returns an empty string if an error occurs
    - registrationNameStrFilenameFormatted()
        - Return type
            - String
        - Description
            - Returns the name of the current registration object as a formatted string
            - Calls the getErrorMessage() method and returns an empty string if an error occurs
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

ImportOperations

- Other defined methods
    - getErrorMessage()
        - Return type
            - String
        - Description
            - Returns a string displaying an error message
    - getLocalizedErrorMessage()
        - Return type
            - String
        - Description
            - Returns a string displaying an error message
    - importDirectory(path, recursive)
        - Return type
            - Boolean
        - Description
            - Imports DICOM files
    - importEclipseTrePointSet(fileLocation, registrationCommissioningName)
        - Return type
            - Integer
        - Description
            - Imports the TRE Point Set from the Eclipse database and returns the record id
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

MarginSettings

- Data descriptors
    - marginDirection
        - MarginGrow = 0
        - MarginShrink = 1
    - marginType
        - MarginSymmetric = 0
        - MarginAsymmetric = 1
    - marginValue
    - marginValueNegX
    - marginValueNegY
    - marginValueNegZ
    - marginValueX
    - marginValueY
    - marginValueZ
- Constructors:
    - Default settings, symmetric with a margin of 0:
        - MarginSettings()
            - marginDirection = 0 (Grow)
            - marginType = 0 (Symmetric)
            - marginValue = 0
            - marginValueNegX = 0
            - marginValueNegY = 0
            - marginValueNegZ =0
            - marginValueX = 0
            - marginValueY = 0
            - marginValueZ = 0
    - Symmetric margin of the specified value:
        - MarginSettings(margin, direction)
            - marginDirection = direction (default 0)
            - marginType = 0 (Symmetric)
            - marginValue = margin
            - marginValueNegX = 0
            - marginValueNegY = 0
            - marginValueNegZ =0
            - marginValueX = 0
            - marginValueY = 0
            - marginValueZ = 0

Return to Table of Contents

    - Asymmetric margin with the specified values:
        - MarginSettings(marginValueX, marginValueY, marginValueZ, marginValueNegX,marginValueNegY, marginValueNegZ, direction)
            - marginDirection = direction (default 0)
            - marginType = 1 (Asymmetric)
            - marginValue = 0
            - marginValueNegX = marginValueNegX
            - marginValueNegY = marginValueNegY
            - marginValueNegZ = marginValueNegZ
            - marginValueX = marginValueX
            - marginValueY = marginValueY
            - marginValueZ = marginValueZ

- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

MatrixR22d

- Attributes
    - num_cols
        - Attribute type
            - Integer
            - Constant
        - Value
            - 2
    - num_rows
        - Attribute type
            - Integer
            - Constant
        - Value
            - 2
- Other defined methods
    - elements()
        - Return Type
            - E
        - Description
            - Returns a pointer to an array with the elements in the matrix
    - extractScale(scale)
        - Description
            - Extracts the scale from the matrix
    - extractTranslation(translation)
        - Description
            - Extracts the translation from the matrix
    - get()
    - identity()
        - Description
            - Sets the matrix back to identity
    - invert()
        - Description
            - Changes the matrix to its inverse
    - isRigid()
        - Return type
            - Boolean
        - Description

                - Determines if a matrix is a pure Rigid
        - set()
        - setElements(elementArray)
            - Description
                - Copies the elements from an array into the matrix
        - transpose(m)
            - Description
                - Returns the transpose of the matrix
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

MatrixR22i

- Attributes
    - num_cols
        - Attribute type
            - Integer
            - Constant
        - Value
            - 2
    - num_rows
        - Attribute type
            - Integer
            - Constant
        - Value
            - 2
- Other defined methods
    - elements()
        - Return Type
            - E
        - Description
            - Returns a pointer to an array with the elements in the matrix
    - extractScale(scale)
        - Description
            - Extracts the scale from the matrix
    - extractTranslation(translation)
        - Description
            - Extracts the translation from the matrix
    - get()
    - identity()
        - Description
            - Sets the matrix back to identity
    - invert()
        - Description
            - Changes the matrix to its inverse
    - isRigid()
        - Return type
            - Boolean
        - Description

                - Determines if a matrix is a pure Rigid
        - set()
        - setElements(elementArray)
            - Description
                - Copies the elements from an array into the matrix
        - transpose(m)
            - Description
                - Returns the transpose of the matrix
- Inherited methods
    - From __builtin__.SwigPyObject)
        - All __builtin__.SwigPyObject methods

MatrixR33d

- Attributes
    - num_cols
        - Attribute type
            - Integer
            - Constant
        - Value
            - 3
    - num_rows
        - Attribute type
            - Integer
            - Constant
        - Value
            - 3
- Other defined methods
    - elements()
        - Return Type
            - E
        - Description
            - Returns a pointer to an array with the elements in the matrix
    - extractScale(scale)
        - Description
            - Extracts the scale from the matrix
    - extractTranslation(translation)
        - Description
            - Extracts the translation from the matrix
    - get()
    - identity()
        - Description
            - Sets the matrix back to identity
    - invert()
        - Description
            - Changes the matrix to its inverse
    - isRigid()
        - Return type
            - Boolean
        - Description

                - Determines if a matrix is a pure Rigid
        - set()
        - setElements(elementArray)
            - Description
                - Copies the elements from an array into the matrix
        - transpose(m)
            - Description
                - Returns the transpose of the matrix
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

MatrixR33f

- Attributes
    - num_cols
        - Attribute type
            - Integer
            - Constant
        - Value
            - 3
    - num_rows
        - Attribute type
            - Integer
            - Constant
        - Value
            - 3
- Other defined methods
    - elements()
        - Return Type
            - E
        - Description
            - Returns a pointer to an array with the elements in the matrix
    - extractScale(scale)
        - Description
            - Extracts the scale from the matrix
    - extractTranslation(translation)
        - Description
            - Extracts the translation from the matrix
    - get()
    - identity()
        - Description
            - Sets the matrix back to identity
    - invert()
        - Description
            - Changes the matrix to its inverse
    - isRigid()
        - Return type
            - Boolean
        - Description

            - Determines if a matrix is a pure Rigid
    - set()
    - setElements(elementArray)
        - Description
            - Copies the elements from an array into the matrix
    - transpose(m)
        - Description
            - Returns the transpose of the matrix
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

MatrixR44d

- Attributes
    - num_cols
        - Attribute type
            - Integer
            - Constant
        - Value
            - 4
    - num_rows
        - Attribute type
            - Integer
            - Constant
        - Value
            - 4
- Other defined methods
    - elements()
        - Return Type
            - E
        - Description
            - Returns a pointer to an array with the elements in the matrix
    - extractScale(scale)
        - Description
            - Extracts the scale from the matrix
    - extractTranslation(translation)
        - Description
            - Extracts the translation from the matrix
    - get()
    - identity()
        - Description
            - Sets the matrix back to identity
    - invert()
        - Description
            - Changes the matrix to its inverse
    - isRigid()
        - Return type
            - Boolean
        - Description

                - Determines if a matrix is a pure Rigid
        - set()
        - setElements(elementArray)
            - Description
                - Copies the elements from an array into the matrix
        - transpose(m)
            - Description
                - Returns the transpose of the matrix
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

MatrixR44f

- Attributes
    - num_cols
        - Attribute type
            - Integer
            - Constant
        - Value
            - 4
    - num_rows
        - Attribute type
            - Integer
            - Constant
        - Value
            - 4
- Other defined methods
    - elements()
        - Return Type
            - E
        - Description
            - Returns a pointer to an array with the elements in the matrix
    - extractScale(scale)
        - Description
            - Extracts the scale from the matrix
    - extractTranslation(translation)
        - Description
            - Extracts the translation from the matrix
    - get()
    - identity()
        - Description
            - Sets the matrix back to identity
    - invert()
        - Description
            - Changes the matrix to its inverse
    - isRigid()
        - Return type
            - Boolean
        - Description

                - Determines if a matrix is a pure Rigid
        - set()
        - setElements(elementArray)
            - Description
                - Copies the elements from an array into the matrix
        - transpose(m)
            - Description
                - Returns the transpose of the matrix
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

PatientDataItem

- Other defined methods
    - getVelocityId()
        - Return Type
            - Integer
        - Description
            - Returns the velocity id value of the object
    - isValid()
        - Return Type
            - Boolean
        - Description
            - Determines if the object is a valid object
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

Image2D

- Other defined methods
    - getCreateDate()
        - Return type
            - String
        - Description
            - Returns the date that the image was created
            - Formats the date in ISO format
    - getEditDate()
        - Return type
            - String
        - Description
            - Returns the date that the image was edited
            - Formats the date in ISO format
    - getInstanceUID()
        - Return type
            - String
    - getModality()
        - Return type
            - String
    - getName()
        - Return type
            - String
        - Description
            - Returns the name
    - getPatient()
        - Return type
            - Patient object
        - Description
            - Returns the patient object
    - getSeriesUID()
        - Return type
            - String
    - getStudyDate()
        - Return type
            - String
        - Description
            - Returns the date of the study

                - Formats the date in ISO format
        - getStudyUID()
            - Return type
                - String
        - getType()
            - Return type
                - String
        - isLocked()
            - Return type
                - Boolean
            - Description
                - Determines if the object is locked
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods
    - From PatientDataItem
        - getVelocityId()
            - Return Type
                - Integer
            - Description
                - Returns the velocity id value of the object
        - isValid()
            - Return Type
                - Boolean
            - Description
                - Determines if the object is a valid object

Patient

- Other defined methods
    - getCreateDate()
        - Return type
            - String
        - Description
            - Returns the date that the record was created
            - Formats the date in ISO format
    - getDateOfBirth()
        - Return type
            - String
        - Description
            - Returns the patient's date of birth
            - Formats the date in ISO format
    - getEditDate()
        - Return type
            - String
        - Description
            - Returns the date that the record was edited
            - Formats the date in ISO format
    - getFirstName()
        - Return type
            - String
        - Description
            - Returns the patient's first name
    - getLastName()
        - Return type
            - String
        - Description
            - Returns the patient's last name
    - getPatientId()
        - Return type
            - String
        - Description
            - Returns the patient's id
    - getPlans()
        - Return type
            - vector < Plan >

    - getSex()
        - Return type
            - String
        - Description
            - Returns the patient's sex
    - getVolumes()
        - Return type
            - vector < Volume >
        - Parameters
            - modalityToGet
    - isLocked()
        - Return type
            - Boolean
        - Description
            - Determines if the object is locked
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods
    - From PatientDataItem
        - getVelocityId()
            - Return Type
                - Integer
            - Description
                - Returns the velocity id value of the object
        - isValid()
            - Return Type
                - Boolean
            - Description
                - Determines if the object is a valid object

Plan

- Other defined methods
    - getCreateDate()
        - Return type
            - String
        - Description
            - Returns the date the plan was created
            - Formats the date in ISO format
    - getDoseVolumeIds()
        - Return type
            - vector < int >
        - Description
            - Returns the id values of the dose volumes
    - getDoseVolumes()
        - Return type
            - vector < Volume >
        - Description
            - Returns the dose volumes
    - getEditDate()
        - Return type
            - String
        - Description
            - Returns the date that the plan was edited
            - Formats the date in ISO format
    - getInstanceUID()
        - Return type
            - String
        - Description
            - Returns the UID of the instance
    - getName()
        - Return type
            - String
        - Description
            - Returns the name of the plan
    - getSeriesUID()
        - Return type
            - String
        - Description

                - Returns the UID of the series
    - getStructureSet()
        - Return type
            - StructureSet object
        - Description
            - Returns the structure set
    - getStructureSetUID()
        - Return type
            - String
        - Description
            - Returns the UID of the structure sets
    - getStructureVelocityIds()
        - Return type
            - vector < int >
        - Description
            - Returns the velocity ids of the structures
    - getStructures()
        - Return type
            - vector < Structure >
        - Description
            - Returns the structures
    - getStudyDate()
        - Return type
            - String
        - Description
            - Returns the date the study was created
            - Formats the date in ISO format
    - getStudyUID()
        - Return type
            - String
        - Description
            - Returns the UID of the study
    - getVolume()
        - Return type
            - Volume object
        - Description
            - Returns the volume
    - getVolumeVelocityId()
        - Return type

                - Integer
            - Description
                - Returns the velocity id of the volume
        - isLocked()
            - Return type
                - Boolean
            - Description
                - Determines if the current plan is locked
- Inherited methods
    - From __builtin__.SwigPyObject)
        - All __builtin__.SwigPyObject methods
    - From PatientDataItem
        - getVelocityId()
            - Return Type
                - Integer
            - Description
                - Returns the velocity id value of the object
        - isValid()
            - Return Type
                - Boolean
            - Description
                - Determines if the object is a valid object

PlanOperations

- Other defined methods
    - makeAdaptivePlan(oldPlan, newPlanName, copiedStructures)
        - Return type
            - Plan object
        - Description
            - Creates a new plan linked to the secondary (adaptive) volume by copying the plan linked to the primary volume.
        - Preconditions
            - Primary volume should be linked to the plan to be copied. Secondary volume should be the adaptive volume.
            - Structures to be processed were copied from the primary to the secondary volume using *copyStructuresToSecondary* in StructureOperations.
        - Parameters
            - oldPlan
                - the old plan object to be copied. This plan must belong to the primary volume.
            - newPlanName
                - Name of new plan
            - copiedStructures
                - a map of structures (source id to new structure) that were copied from the primary volume on the secondary(adaptive) Volume.
    - makeContainerPlan(planName, treatmentMachine, structureSetId)
        - Return type
            - Plan object. Resampled dose will be set as secondary and new structure set structures loaded.
        - Description
            - Creates a new shell plan that links the dose (resampled) to the CT. The dose is resampled and linked to the new plan. The plan is linked to the structure set.
        - Preconditions
            - Primary volume should be a CT
            - Secondary volume should be the RTDOSE in the same FOR as primary volume. 'DICOM' registration will be loaded between the volumes.

Return to Table of Contents

        - Parameters
            - planName
                - Name of the plan created
            - treatmentMachine
                - The treatment machine that will be written in the plan
            - structureSetId
                - Optional - this is the id of the structure set that the new plan will point to. If none is given, the function creates a new structure set with an External body contour.

    - getErrorMessage()
        - Return type
            - String
        - Description
            - Returns an error message based on the error that has occurred
    - getLocalizedErrorMessage()
        - Return type
            - String
        - Description
            - Returns a localized error message based on the error that has occurred

Registration

- Other defined methods
    - getCreateDate()
        - Return type
            - String
        - Description
            - Returns the date that the registration was created
            - Formats the date in ISO format
    - getDicomRegistrationUID()
        - Return type
            - String
        - Description

                - Returns the UID of the DICOM registration
        - getEditDate()
            - Return type
                - String
            - Description
                - Returns the date that the registration was edited
                - Formats the date in ISO format
        - getInstanceUID()
            - Return type
                - String
            - Description
                - Returns the UID of the instance
        - getName()
            - Return type
                - String
            - Description
                - Returns the name of the registration
        - getSourceVolume()
            - Return type
                - Volume object
            - Description
                - Returns the source volume
        - getTargetVolume()
            - Return type
                - Volume object
            - Description
                - Returns the target volume
        - getType()
            - Return type
                - String
            - Description
                - Returns the registration type
        - isLocked()
            - Return type
                - Boolean
            - Description
                - Determines if the registration is locked
- Inherited methods
    - From __builtin__.SwigPyObject

        - All __builtin__.SwigPyObject methods
    - From PatientDataItem
        - getVelocityId()
            - Return Type
                - Integer
            - Description
                - Returns the velocity id value of the object
        - isValid()
            - Return Type
                - Boolean
            - Description
                - Determines if the object is a valid object

RegistrationCommissioningMarkerList

- Other defined methods
    - getCreateDate()
        - Return type
            - String
        - Description
            - Returns the date that the registration commissioning marker was created
            - Formats the date in ISO format
    - getName()
        - Return type
            - String
        - Description
            - Returns the name of the registration commissioning marker
    - getPrimaryVolume()
        - Return type
            - Volume object
        - Description
            - Returns the primary volume
    - getRegistrationMarkerList()
        - Return type
            - vector < RegistrationMarker >
        - Description
            - Returns the list of registration markers
    - getSecondaryVolume()
        - Return type
            - Volume object
        - Description
            - Returns the secondary volume
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods
    - From PatientDataItem
        - getVelocityId()
            - Return Type
                - Integer
            - Description

        - Returns the velocity id value of the object
- isValid()
    - Return Type
        - Boolean
    - Description
        - Determines if the object is a valid object

Structure

- Other defined methods
    - getCreateDate()
        - Return type
            - String
        - Description
            - Returns the date that the structure was created
            - Formats the date in ISO format
    - getEditDate()
        - Return type
            - String
        - Description
            - Returns the date that the structure was edited
            - Formats the date in ISO format
    - getFrameOfReferenceUID()
        - Return type
            - String
        - Description
            - Returns the UID of the frame of reference
    - getInstanceUID()
        - Return type
            - String
        - Description
            - Returns the UID of the instance
    - getName()
        - Return type
            - String
        - Description
            - Returns the name of the structure
    - getROINumber()
        - Return type
            - String
        - Description
            - Returns the ROI number of the structure
    - getStructureSet()
        - Return type
            - StructureSet object
        - Description

                - Returns the structure set
        - getType()
            - Return type
                - String
            - Description
                - Returns the type of the structure
        - getVolume()
            - Return type
                - Volume object
            - Description
                - Returns the volume of the structure
        - getVolumeId()
            - Return type
                - Integer
            - Description
                - Returns the id of the volume of the structure
        - getVolumetricSize()
            - Return type
                - Double
            - Description
                - Returns the size of the volume of the structure
        - isEditable()
            - Return type
                - Boolean
            - Description
                - Returns if the structure can be edited
        - isLocked()
            - Return type
                - Boolean
            - Description
                - Returns if the structure is locked
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods
    - From PatientDataItem
        - getVelocityId()
            - Return Type
                - Integer
            - Description

        - Returns the velocity id value of the object
- isValid()
    - Return Type
        - Boolean
    - Description
        - Determines if the object is a valid object

StructureMask

- Attributes and data descriptors
    - size
        - Attribute type
            - VectorR3i
        - Description
            - Size of the volumetric structure mask. Should match the size of the volume that it belongs to. Structures are usually created on the primary volume.
    - data
        - Type
            - vector < double >
        - Description
            - Binary values used to represent whether the pixel is in the structure or not.

Return to Table of Contents

StructureSet

- Other defined methods
    - getCreateDate()
        - Return type
            - String
        - Description
            - Returns the date that the structure set was created
            - Formats the date in ISO format
    - getEditDate()
        - Return type
            - String
        - Description
            - Returns the date that the structure set was edited
            - Formats the date in ISO format
    - getInstanceUID()
        - Return type
            - String
        - Description
            - Returns the UID of the instance
    - getName()
        - Return type
            - String
        - Description
            - Returns the name of the structure set
    - getSeriesDescription()
        - Return type
            - String
        - Description
            - Returns the description of the structure set series
    - getSeriesUID()
        - Return type
            - String
        - Description
            - Returns the UID of the series
    - getStructures()
        - Return type
            - vector < Structure >
        - Description

                - Returns the structures in the structure set
        - getStudyDate()
            - Return type
                - String
            - Description
                - Returns the date that the study was created
                - Formats the date in ISO format
        - getStudyUID()
            - Return type
                - String
            - Description
                - Returns UID of the study
        - getVolume()
            - Return type
                - Volume object
            - Description
                - Returns the volume of the structure set
        - getVolumeUID()
            - Return type
                - String
            - Description
                - Returns the UID of the volume
        - isLocked()
            - Return type
                - Boolean
            - Description
                - Returns if the structure set it locked
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods
    - From PatientDataItem
        - getVelocityId()
            - Return Type
                - Integer
            - Description
                - Returns the velocity id value of the object
        - isValid()
            - Return Type
                - Boolean

- Description
    - Determines if the object is a valid object

Volume

- Other defined methods
    - getAccessionNumber()
        - Return type
            - String
        - Description
            - Returns the accession number of the volume
    - getAllStructures()
        - Return type
            - vector < Structure >
        - Description
            - Returns all of the structures
    - getCreateDate()
        - Return type
            - String
        - Description
            - Returns the date that the volume was created
            - Formats the date in ISO format
    - getDicomPatientDateOfBirth()
        - Return type
            - String
        - Description
            - Returns the date that the patient was born
            - Formats the date in ISO format
    - getDicomPatientId()
        - Return type
            - String
        - Description
            - Returns the patient id
    - getDicomPatientName()
        - Return type
            - String
        - Description
            - Returns the name of the patient
    - getDicomPatientSex()
        - Return type
            - String
        - Description

                - Returns the sex of the patient
    - getDicomReferringPysicianName()
        - Return type
            - String
        - Description
            - Returns the name of the patient's physician
    - getEditDate()
        - Return type
            - String
        - Description
            - Returns the date that the volume was edited
            - Formats the date in ISO format
    - getFrameOfReferenceUID()
        - Return type
            - String
        - Description
            - Returns the UID of the frame of reference
    - getLinkedPlans()
        - Return type
            - vector < Plan >
        - Description
            - Returns the plans linked to the volume
    - getLinkedRegistrations()
        - Return type
            - vector < Registration >
        - Description
            - Returns the registrations linked to the volume
    - getModality()
        - Return type
            - String
        - Description
            - Returns the modality
    - getName()
        - Return type
            - String
        - Description
            - Returns the name of the volume
    - getPatient()
        - Return type

                - Patient object
            - Description
                - Returns the patient
    - getPlanUID()
        - Return type
            - String
        - Description
            - Returns the plan UID of the dose volumes
    - getAcquisitionDate()
        - Return type
            - String
        - Returns the AcquisitionDate of the volume
        - Formats the date in ISO format
    - getSeriesDate()
        - Return type
            - String
        - Description
            - Returns the date the series was created
            - Formats the date in ISO format
    - getSeriesNumber()
        - Return type
            - String
        - Description
            - Returns the number of the series
    - getSeriesTime()
        - Return type
            - String
        - Description
            - Returns the time the series was created
    - getSeriesType()
        - Return type
            - String
        - Description
            - Returns the type of the series
    - getSeriesUID()
        - Return type
            - String
        - Description
            - Returns the UID of the series

- getSourceType()
  - Return type
    - String
  - Description
    - Returns the type of the source
- getStructureSets()
  - Return type
    - vector < StructureSet >
  - Description
    - Return the structure sets of the volume
- getStudyDate()
  - Return type
    - String
  - Description
    - Returns the date the study was created
    - Formats the date in ISO format
- getStudyDescription()
  - Return type
    - String
  - Description
    - Returns the description of the study
- getStudyId()
  - Return type
    - String
  - Description
    - Returns the id of the study
- getStudyInstanceUID()
  - Return type
    - String
  - Description
    - Returns the UID of the study instance
- getStudyTime()
  - Return type
    - String
  - Description
    - Returns the time the study was created
- getVolumeUID
  - Return type
    - String

        - Description
            - Returns the UID of the volume
    - isLocked()
        - Return type
            - Boolean
        - Description
            - Returns if the volume is locked
    - registrationCommissioningMarkerLists()
        - Return type
            - vector < RegistrationCommissioningMarkerList >
        - Parameters
            - secondaryVolumeId
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods
    - From PatientDataItem
        - getVelocityId()
            - Return Type
                - Integer
            - Description
                - Returns the velocity id value of the object
        - isValid()
            - Return Type
                - Boolean
            - Description
                - Determines if the object is a valid object

PatientDataOperations

- Other defined methods
    - getErrorMessage()
        - Return type
            - String
        - Description
            - Returns an error message based on the error that has occurred
    - getLocalizedErrorMessage()
        - Return type
            - String
        - Description
            - Returns a localized error message based on the error that has occurred
    - getPatient(vscId)
        - Return type
            - Patient object
        - Description
            - Returns the patient of a given vsc id
    - getPatientByPatientId(patientId)
        - Return type
            - Patient object
        - Description
            - Returns the patient of a given patient id
    - getPatients()
        - Return type
            - vector < patientdata::Patient >
        - Description
            - Returns the patients
    - getPlan(velocityId)
        - Return type
            - Plan object
        - Description
            - Returns the plan of a given velocity id
    - getRegistration(velocityId)
        - Return type
            - Registration object
        - Description

                - Returns the registration of a given velocity id
        - getRegistrationCommissioningMarkerList(velocityId)
            - Return type
                - RegistrationCommissioningMarkerList object
            - Description
                - Returns the registration commissioning marker list of a given velocity id
        - getStructure(velocityId)
            - Return type
                - Structure object
            - Description
                - Returns the structure of a given velocity id
        - getStructureSet(velocityId)
            - Return type
                - StructureSet object
            - Description
                - Returns the structure set of a given velocity id
        - getStructureSetByUID(uid)
            - Return type
                - StructureSet object
            - Description
                - Returns the structure set of a given UID
        - getVolume(velocityId)
            - Return type
                - Volume object
            - Description
                - Returns the volume of a given velocity id
        - getVolumeByUID(uid)
            - Return type
                - Volume object
            - Description
                - Returns the volume of a given UID
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

RegistrationMarker

- Other defined methods
  - getName()
    - Return type
      - String
    - Description
      - Returns the name of the registration marker
  - getPrimaryLocation()
    - Return type
      - VectorR3d object
    - Description
      - Returns the primary location of the registration marker
  - getSecondaryLocation()
    - Return type
      - VectorR3d object
    - Description
      - Returns the secondary location of the registration marker
  - isConfirmed()
    - Return type
      - Boolean
    - Returns if the registration marker has been confirmed
- Inherited methods
  - From __builtin__.SwigPyObject
    - All __builtin__.SwigPyObject methods

Return to Table of Contents

RegistrationOperations

- Other defined methods
  - computeJacobian()
    - Return type
      - ScaledVolumeData object
    - Description
      - Calculates the Jacobian from loaded volumes and registrations
      - Will return an empty volume data if there is an error
  - createNewRegistration(name, progress = nullptr)
    - Return type
      - Registration object
    - Description
      - Creates a new rigid registration with a given name
  - exportRegistration(registrationId, progress = nullptr)
    - Description
      - Creates a DICOM file of the current registration
  - getErrorMessage()
    - Return value
      - String
    - Description
      - Returns an error message based on the error that has occurred
  - getLocalizedErrorMessage()
    - Return value
      - String
    - Description
      - Returns a localized error message based on the error that has occurred
  - getTransformedRegistrationCommissioningMarkers(markerListId)
    - Return type
      - pair < bool, vector < TransformedRegistrationMarker > >
    - Description
      - Returns a list of registration commissioned markers
      - The registration commissioned markers will be converted to coordinate systems
  - performBsplineRegistration(settings)
    - **Deprecated**

        - Return type
            - Boolean
        - Description
            - Performs a BSpline deformable registration of two loaded volumes
            - Requires coordinate and image values to exist in Velocity's coordinate space and units
            - Returns true if the deformable registration occurs successfully
    - performBsplineRegistrationDICOM(settings)
        - Return type
            - Boolean
        - Description
            - Performs a BSpline deformable registration of two loaded volumes
            - Requires coordinate and image values to exist in DICOM coordinate space and units
            - Returns true if the deformable registration occurs successfully
        - settings can be either
            - DefaultBSplineDeformableRegistrationSettings
                - Performs a deformable registration between the two loaded volumes in the engine using default values yielding very close results as the Velocity interface.
            - BSplineDeformableRegistrationSettingsStructure
                - Allows the user to manually control settings that impact the registration.
    - performManualAlignment(settings)
        - **Deprecated**
        - Return type
            - Boolean
        - Description
            - Performs a manual alignment of two loaded volumes
            - Requires coordinate and image values to exist in Velocity's coordinate space and units
            - Returns true if the manual alignment occurs successfully
    - performManualAlignmentDICOM(settings)
        - Return type

            - Boolean
        - Description
            - Performs a manual alignment of two loaded volumes
            - Requires coordinate and image values to exist in DICOM coordinate space and units
            - Returns true if the manual alignment occurs successfully
    - performRigidRegistration(settings)
        - **Deprecated**
        - Return type
            - Boolean
        - Description
            - Performs a rigid registration of two loaded volumes
            - Requires coordinate and image values to exist in Velocity's coordinate space and units
            - Returns true if the rigid registration occurs successfully
    - performRigidRegistrationDICOM(settings)
        - Return type
            - Boolean
        - Description
            - Performs a rigid registration of two loaded volumes
            - Requires coordinate and image values to exist in DICOM coordinate space and units
            - Returns true if the rigid registration occurs successfully
        - settings can be either
            - RigidRegistrationSettingsStructure
                - Allows the user to manually control settings that impact the registration.
            - DefaultRigidRegistrationSettings
                - Performs a rigid registration between the two loaded volumes in the engine using default values yielding very close results as the Velocity interface.
    - performBurnInStructureGuidedBsplineRegistrationDICOM(settings, burnInStructuresIds, burnInVoxelValues)
        - Return type
            - Boolean
        - Description
            - Performs a Burn-in Structure-Guided BSpline deformable registration between the two loaded volumes. If the current registration is a deformable object then it will be

restarted. If the current registration is not a deformable object then it will be converted into one by this process.
- Note, this function will restart a deformable object if such an object is currently loaded into memory.
- The roiStart and roiEnd values within *settings* are expected to be in DICOM coordinates of the currently set PRIMARY volume.
- The primaryStartLevel and primaryEndLevel values within *settings* are expected to be in units of the type that is currently set within the primary volume's histogram.
- The secondaryStartLevel and secondaryEndLevel values within *settings* are expected to be in units of the type that is currently set within the secondary volume's histogram.
- Note, the following two lists must have the same length: *'burnInstructureIds'* and *'burnInVoxelValues'*.
- settings(BSplineDeformableRegistrationSettingsStructure) - the deformable algorithm settings
- burnInStructureIds- the list of structure record Ids that are to be used in the Burn-in Structure-Guided DIR
- burnInVoxelValues - the list of values that are to be burned into the volumes (used in conjunction with burnInStructureIds)
- Returns true if successful

    - saveRegistration(progress = nullptr)
        - Return type
            - Registration object
        - Description
            - Saves loaded registration to the database
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

ReportOperations

- Other defined methods
    - appendToReport(htmlStreamDataToAppend, includePatientHeaderInformation = true)
        - Return type
            - Boolean
        - Description
            - Appends a specified html stream to the internal html report
            - Calls getErrorMessage() method and returns false if an error occurs
    - attachRegistrationCommissioningToReport(registrationCommissioningMarkerListRecord)
        - Return type
            - Boolean
        - Description
            - Attaches the specified registration commissioning marker to the internal report
            - Calls getErrorMessage() method and returns false if an error occurs
    - attachViewer3DToReport()
        - Return type
            - Boolean
        - Description
            - Attaches the viewer state to the internal report
    - clearReport()
        - Description
            - Clears the internal html report
    - getErrorMessage()
        - Return type
            - String
        - Description
            - Returns an error message based on the error that has occurred
    - getLocalizedErrorMessage()
        - Return type
            - String
        - Description

                - Returns a localized error message based on the error that has occurred
        - getReport()
            - Return type
                - String
            - Description
                - Returns the internal html report
        - loadPersistentCachedReport()
            - Return type
                - Boolean
            - Description
                - Loads persistent reports for the current patient from the database
        - loadReportFromDatabase(reportRecordId)
            - Return type
                - Boolean
            - Description
                - Loads the specified report from the database
        - persistCachedReport()
            - Return type
                - Boolean
            - Description
                - Puts the internal html report into a temporary file in the database
        - saveReportToDatabase(reportName)
            - Return type
                - Integer
            - Description
                - Saves the internal html report into the database under the specified name
                - Returns the id number of the report if the save is successful, otherwise -1 is returned
                - Clears the internal report if the save is successful
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

ScaledVolumeData

- Attributes and data descriptors
  - data
    - Attribute type
      - vector < double >
    - Description
      - Image volume pixel data, with dimension x*y*z given in size.
  - indexTranform
    - Attribute type
      - MatrixR44d object
    - Description
      - 4x4 matrix to scale the image volume. Units are in meters.
  - size
    - Attribute type
      - VectorR3i object
    - Description
      - Size of image volume
  - volume
    - Attribute type
      - Volume object
    - Description
      - Volume object metadata
- Inherited methods
  - From __builtin__.SwigPyObject
    - All __builtin__.SwigPyObject methods

Return to Table of Contents

StructureOperations

- Other defined methods
    - conformality(structureId1, structureId2, reporter = nullptr)
        - Return type
            - Double
        - Description
            - Calculates the conformality of two structures
    - copyStructuresToPrimary(secondaryStructureIds, structureSetId)
        - Return type
            - map < int, Structure >
        - Description
            - Copies the structures in the secondary volume to the primary volume
    - copyStructuresToSecondary(primaryStructureIds, structureSetId)
        - Return type
            - map < int, Structure >
        - Description
            - Copies the structures in the primary volume to the secondary volume
    - createStructure(structureSetId, name, type = “Organ”, color = nullptr)
        - Return type
            - Structure object
        - Description
            - Creates a new structure
            - Requires that the type parameter is a valid DICOM type
    - createStructureSet(name, onPrimary, studyUID = “”, studyDate = “”)
        - Return type
            - StructureSet object
        - Description
            - Creates a new structure set
    - createAutoExternal(name, origin)
        - Return type
            - StructureSet object
        - Description
            - Creates a new structure set on the primary Volume with an External body structure

Return to Table of Contents

        - Parameters
            - name
                - Name of the new structure set
            - origin
                - optional, default is ‘Internal’ (allows edits), use ‘External’ when you want to lock in the instanceUID of the RTSTRUCT and prevent edits. No other values are valid.
    - deleteStructures(structureIds, structureSetId, flags)
        - Return type
            - Boolean
        - Description
            - Deletes structures from a single structure set. Note: each structure must belong to the same structure set with state 'Internal' (editable)
        - Parameters
            - structureId
                - the internal id of the structures to delete. Locked structures will not be deleted.
            - structureSetId
                - database id of structure set
            - flags
                - use DeleteFlags to set appropriate flags for deletion exceptions. All flags are off by default.
    - crop(structureSetId, sourceStructureId, croppingStructureIds, cropSettings, targetName, targetColor = nullptr, reporter = nullptr)
        - Return type
            - Structure object
        - Description
            - Crops a structure
    - exportStructureSet(structureSetId, encodingType = “”, reporter = nullptr)
        - Return type
            - Boolean
        - Description
            - Creates the structure set as a RTSTRUCT DICOM file on the disk
    - getErrorMessage()
        - Return type

            - String
        - Description
            - Returns an error message based on the error that has occurred
    - getLocalizedErrorMessage()
        - Return type
            - String
        - Description
            - Returns a localized error message based on the error that has occurred
    - getStructureHistogram(structureId, onPrimary, bins)
        - Return type
            - pair < bool, vector < VectorR2d > >
            - a pair with the first slot indicating success or failure and second containing the histogram values
        - Description
            - Computes and returns the volume histogram of a given structure
            - structureId - the id of the structure to compute the histogram of
            - onPrimary - histogram will be created on the primary volume if true, on the secondary otherwise (default true)
            - Bins - the number of histogram bins to compute (default 65535)
    - getVolumetricStructure(structureId)
        - Return type
            - ScaledVolumeData object
        - Description
            - Returns the volumetric structure data
            - Requires that the structureId parameter is either the primary or secondary volume
            - Will change the secondary volume to the primary volume if the secondary volume is used
    - getStructureBoundingRegionDICOM(structureId)
        - Return type
            - BoundingRegion
        - Description
            - Determines the structure's bounding region and returns the starting and ending points of the bounding region in

the DICOM coordinates of the currently set PRIMARY volume regardless if the structure belongs to the currently set secondary volume (if any).
- Precondition: The associated volume of the specified structure is needs to be currently loaded.
- Returns the bounding region of the structure, with isValid = false on error. Note that the points are specified in the DICOM coordinates of the currently set PRIMARY volume regardless if the structure belongs to the currently set secondary volume.

- intersectStructures(structureSetId, structureIds, targetName, targetColor = nullptr, reporter = nullptr)
  - Return type
    - Structure object
  - Description
    - Creates a structure based on the intersection of multiple structures
- margin(structureSetId, sourceId, targetId, settings, reporter = nullptr)
  - Return type
    - Structure object
  - Description
    - Uses the margin operation on a structure
- saveStructureSets(structureSetId, encodingType = “”)
  - Return type
    - StructureSet object
  - Description
    - Saves changes to a structure set
    - Must be called after finishing a set of modifications to a structure set
- smooth(structureSetId, sourceStructureId, targetName, reporter = nullptr)
  - Return type
    - Structure object
  - Description
    - Creates a new structure after smoothing the current structure
- surfaceDistanceMetrics(structureId1, structureId2)
  - Return type
    - SurfaceDistanceMetrics object

        - Description
            - Calculates the surface distance between two structures
    - unionStructures(structureSetId, structureIds, targetName, targetColor = nullptr, reporter = nullptr)
        - Return type
            - Structure object
        - Description
            - Used the union from multiple structures to create a new structure
    - createStructureFromMask(structureSetId, mask, structureName, structureType, color)
        - Return type
            - Structure object
        - Description
            - Given a binary mask it converts it into a structure. The structure must belong to the current primary volume. The dimensions of the mask must be the same as the primary volume. The frame of reference and index transform of the structure will be taken from the primary volume.
            - structureSetId - the structure set of the new structure
            - mask.data (StructureMask) is be the binary mask used to create the structure
            - (optional)name - the structure name, default “StructureMask”
            - (optional) type - the DICOM type of the structure, default “Organ”
            - (optional) color - chosen color, otherwise one is chosen automatically

- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

SurfaceDistanceMetrics

- Attributes and data descriptors
    - hausdorffDistance
        - Attribute type
            - Double
    - isValid
        - Attribute type
            - Boolean
    - mean
        - Attribute type
            - Double
    - median
        - Attribute type
            - Double
    - min
        - Attribute type
            - Double
    - standardDeviation
        - Attribute type
            - Double
    - structureName1
        - Attribute type
            - String
    - structureName2
        - Attribute type
            - String
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

Return to Table of Contents

TransformedRegistrationMarker

- Attributes and data descriptors
  - deformedSecondaryInPrimaryDICOM
    - Attribute type
      - VectorR3d object
  - deformedSecondaryInScene
    - Attribute type
      - VectorR3d object
  - deformedSecondaryInSecondaryDICOM
    - Attribute type
      - VectorR3d object
  - isConfirmed
    - Attribute type
      - Boolean
  - name
    - Attribute type
      - String
  - primaryInIndex
    - Attribute type
      - VectorR3d object
  - primaryInPrimaryDICOM
    - Attribute type
      - VectorR3d object
  - primaryInScene
    - Attribute type
      - VectorR3d object
  - rigidSecondaryInPrimaryDICOM
    - Attribute type
      - VectorR3d object
  - rigidSecondaryInSecondaryDICOM
    - Attribute type
      - VectorR3d object
  - secondaryInIndex
    - Attribute type
      - VectorR3d object

- Inherited methods
  - From __builtin__.SwigPyObject

Return to Table of Contents

VectorR2b

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 2
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E >
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR2d

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 2
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR2f

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 2
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR2i

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 2
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR2s

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 2
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR2ub

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 2
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR2ui

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 2
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector

    - xy()
        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR2us

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 2
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR3b

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 3
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR3d

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 3
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR3f

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 3
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR3i

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 3
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR3s

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 3
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR3ub

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 3
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR3ui

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 3
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR3us

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 3
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR4b

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 4
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()
        - Return type

                - VectorR < 2, E >
            - Description
                - Returns the first two components of the vector
                - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR4d

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 4
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()
        - Return type

                - VectorR < 2, E >
            - Description
                - Returns the first two components of the vector
                - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR4f

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 4
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()
        - Return type

            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR4i

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 4
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR4s

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 4
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR4ub

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 4
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR4ui

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 4
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VectorR4us

- Attributes
    - SIZE
        - Attribute type
            - Integer
            - Constant
        - Value
            - 4
- Other defined methods
    - nonNegative()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is nonnegative
    - normalized(epsilon = vsc::VSC_EPSILON)
        - Return type
            - VectorR < N, E > object
        - Description
            - Returns a normalized vector
    - positive()
        - Return type
            - Boolean
        - Description
            - Determines if every element in the vector is positive
    - set(element)
        - Description
            - Changes all elements in the vector to the specified element
    - setElements(elementArray)
        - Description
            - Sets the elements in the vector equal to the elements in the array
            - The vector and array need to have the same number of elements
    - size()
        - Description
            - Determines the size of the vector
    - xy()

        - Return type
            - VectorR < 2, E >
        - Description
            - Returns the first two components of the vector
            - Sets undefined coordinates to 0
    - xyz()
        - Return type
            - VectorR < 3, E >
        - Description
            - Returns the first three components of the vector
            - Sets undefined coordinates to 0
    - xyzw()
        - Return type
            - VectorR < 4, E >
        - Description
            - Returns the first three components of the vector
            - If the w coordinate is undefined, it is set to 1
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VelocityEngine

- Attributes
    - VSC_PRIMARY_VALUE
        - Value
            - 0
    - VSC_SECONDARY_VALUE
        - Value
            - 1
- Other defined methods
    - deletePatient(patientId, forceDelete)
        - Return type
            - Boolean
        - Description
            - Deletes the patient with the specified patient id
            - If there are multiple patients with the same id, forceDelete must be true in order to delete the patients
            - Returns true if the patient is successfully deleted
    - deleteVolume(volumeId, flags)
        - Return type
            - Boolean
        - Description
            - Deletes a volume identified by database id
            - Returns true if the delete is successful
        - Parameters
            - volumeId
                - Database id of volume to be deleted
            - flags
                - use DeleteFlags to set appropriate flags for deletion exceptions. All flags are off by default.
    - deletePlan(planId, flags)
        - Return type
            - Boolean
        - Description
            - Deletes a plan identified by database id
            - Returns true if the delete is successful
        - Parameters
            - planId
                - Database id of plan to be deleted

            - flags
                - use DeleteFlags to set appropriate flags for deletion exceptions. All flags are off by default.
    - deleteRegistration(registrationId, flags)
        - Return type
            - Boolean
        - Description
            - Deletes a registration identified by database id
            - Returns true if the delete is successful
        - Parameters
            - registrationId
                - Database id of volume to be deleted
            - flags
                - use DeleteFlags to set appropriate flags for deletion exceptions. All flags are off by default.
    - deleteStructureSet(structureSetId, flags)
        - Return type
            - Boolean
        - Description
            - Deletes a structure set identified by database id
            - Returns true if the delete is successful
        - Parameters
            - structureSetId
                - Database id of structure set to be deleted
            - flags
                - use DeleteFlags to set appropriate flags for deletion exceptions. All flags are off by default.
    - getCurrentPatient()
        - Return type
            - Patient object
        - Description
            - Returns the patient that is currently loaded
            - Will print an error message if there is no patient loaded
    - getErrorMessage()
        - Return type
            - String
        - Description
            - Returns an error message based on the error that has occurred

            - If there is not an error, an empty string is returned
    - getExportOperations()
        - Return type
            - ExportOperations object
        - Description
            - Returns the export operations for the engine
            - Will return a null pointer if the engine is not connected to the database
    - getImportOperations()
        - Return type
            - ImportOperations object
        - Description
            - Returns the import operations for the engine
            - Will return a null pointer if the engine is not connected to the database
    - getLocalizedErrorMessage()
        - Return type
            - String
        - Description
            - Returns a localized error message based on the error that has occurred
            - If there is not a localized error, and empty string is returned
    - getPatientDataOperations()
        - Return type
            - PatientDataOperations object
        - Description
            - Returns the patient data operations for the engine
            - Will return a null pointer if the engine is not connected to the database
    - getPatientVolumeUIDs(patientID, modality)
        - Return type
            - vector < string >
        - Description
            - Returns the volumeUIDs of a specified patient in the form of a vector
    - getPrimaryVolume()
        - Return type
            - Volume object

        - Description
            - Returns the primary volume of the current patient
    - getRegistrationOperations()
        - Return type
            - RegistrationOperations object
        - Description
            - Returns the registration operations for the engine
            - Will return a null pointer if the engine is not connected to the database
    - getReportOperations()
        - Return type
            - ReportOperations object
        - Description
            - Returns the report operations for the engine
            - Will return a null pointer if the engine is not connected to the database
    - getSecondaryVolume()
        - Return type
            - Volume object
        - Description
            - Returns the secondary volume of the current patient
    - getStructureOperations()
        - Return type
            - StructureOperations object
        - Description
            - Returns the structure operations for the engine
            - Will return a null pointer if the engine is not connected to the database
    - getValueAtStructureCenter(volume, linearInterp = true)
        - Return type
            - Double
        - Description
            - Returns the value at the structure center of the primary and secondary volumes
    - getVolumeOperations()
        - Return type
            - VolumeOperations object
        - Description
            - Returns the volume operations for the engine

            - Will return a null pointer if the engine is not connected to the database
    - isLoggedIn()
        - Return type
            - Boolean
        - Description
            - Determines if the engine is currently logged in
    - loadPatientByPatientId(patientId)
        - Return type
            - Patient object
        - Description
            - Loads a patient based on the specified patient id
            - Returns true if the patient is loaded successfully
    - loadPatient(patientId)
        - Return type
            - Boolean
        - Description
            - Load a patient based on the specified internal database id
            - Returns true if the patient is loaded successfully
    - loadPlan(velocityId)
        - Return type
            - Plan object
        - Description
            - Loads a plan based on the specified patient id
    - loadPlanByUID(uid)
        - Return type
            - Plan object
        - Descritpion
            - Loads a plan based on the specified UID
    - loadPrimaryVolumeByUID(volumeUID)
        - Return type
            - Boolean
        - Description
            - Loads a primary volume based on the specified volume UID
            - Returns true if the volume is loaded successfully
    - loadPrimaryVolume(databaseId)
        - Return type
            - Boolean

- loadRegistration(registrationId)
  - Return type
    - Registration object
  - Description
    - Loads a registration based on the specified registration id
- loadRegistrationByName(name)
  - Return type
    - Registration object
  - Description
    - Loads a registration based on the specified registration name
- loadChainRegistration(registrationId1, registrationId2)
  - Return type
    - Boolean
  - Description
    - Loads a chain registration by combining two internal registration database ids.
    - This can be useful when resampling a dose on a different CT (CT-CT registration + DICOM)
- loadSecondaryVolumeByUID(volumeUID)
  - Return type
    - Boolean
  - Description
    - Loads a volume based on the specified volume UID
    - Returns true if the volume is loaded successfully
- loadSecondaryVolume(databaseId)
  - Return type
    - Boolean
- loadStructure(velocityId)
  - Return type
    - Structure object
  - Description
    - Loads a structure based on the specified velocity id
- loadStructureByName(name, structureSetUID)
  - Return type
    - Structure object
  - Description
    - Loads a structure based on the specified structure name

            - The structure must have a unique name in the set in order to be loaded properly
    - loadStructureByUID(uid)
        - Return type
            - Structure object
        - Description
            - Loads a structure based on the specified structure UID
    - loginToGrid(username, password, ip, port, dbLabel)
        - Return type
            - Boolean
        - Description
            - Logs into the the specified Grid database
            - Returns true if the specified Grid database is successfully logged into
            - All parameters must be valid in order to successfully log in
    - loginToWorkstation(username, password, path, force = false)
        - Return type
            - Boolean
        - Description
            - Logs into the specified workstation database
            - Returns true if the specified workstation database is successfully logged into
            - All parameters must be valid in order to successfully log in
    - logout()
        - Return type
            - Boolean
        - Description
            - Logs out of the database
            - Logging out makes all Operations class instances invalid
    - unloadPatient()
        - Return type
            - Boolean
        - Description
            - Unloads the current patient
            - Will cause the primary and secondary volumes to also unload
    - unloadPrimaryVolume()

        - Return type
            - Boolean
        - Description
            - Unloads the current primary volume
            - Will cause the secondary volume to unload
    - unloadSecondaryVolume()
        - Return type
            - Boolean
        - Description
            - Unloades the current secondary volume
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

VolumeOperations

- Other defined methods
    - createBEDose(data, background, EQDx)
        - Return type
            - Integer
        - Description
            - Carries out the BED/EQD2 scaling operation for the current secondary volume against the current primary volume
            - The currently loaded secondary must be a (PHYSICAL) dose
            - Pass in the BedDoseCalculationData object as data and BedBackgroundData object as background. These have alpha/beta ratios and other parameters for the calculation
            - EQDx is a boolean which enables the EQD2 scaling
            - Returns the volume id of the created volume, unless there is an error, in which -1 is returned
    - createResampledVolume(elementOperation, name = “”, scalingCoefficient = 1.0)
        - Return type
            - Integer
        - Description
            - Carries out the resampling operation defined by elementOperation against the current primary volume and secondary volumes
            - Returns the volume id of the created volume unless there is an error, in which -1 is returned
            - ElementOperation is an integer corresponding to:
                - VolumeResampleReplace = 0,
                - VolumeResampleAdd = 1
                - VolumeResampleSubtract = 2
                - VolumeResampleMultiply = 3
                - VolumeResampleDivide = 4
                - VolumeResampleAverage = 5
                - VolumeResampleMaximum = 6
                - VolumeResampleScale = 7
                - VolumeResampleFadeCorrection = 8

        - VolumeResampleReshape = 9
        - VolumeResampleMinimum = 10
- createResampledPrimaryVolume(elementOperation, name = “”, scalingCoefficient = 1.0)
    - Return type
        - Integer
    - Description
        - Performs the resampling of the specified element operation for the loaded primary volume
        - An error will occur if the specified element operation must be performed on multiple volumes
        - Returns the volume id of the created volume unless there is an error, in which -1 is returned
        - The only elementOperation available for single volume resampling is VolumeResampleScale = 7
- createResampledVolumeAggregate(elementOperation, volIds, flags = ResampleFlags::AllowNone);
    - Return type
        - Integer
    - Description
        - Performs the resampling of the specified element operation for the list of volume ids passed. One of the volumes must be set as primary for reference.
        - All volumes in volIds need to be of the same modality and in the same frame of reference.
        - Returns the volume id of the created volume unless there is an error, in which case -1 is returned
        - ElementOperation is an integer corresponding to:
            - VolumeResampleAdd = 1
            - VolumeResampleSubtract = 2
            - VolumeResampleAverage = 5
            - VolumeResampleMaximum = 6
            - VolumeResampleMinimum = 10
        - Flags is a value from the enum ResampleFlags, and all flags are off by default. These flags are used for determining the types of doses allowed in the resample operation and only apply to RTDOSE.
            - AllowNone = 0

        - Will not allow the resample operation to proceed if the doses are of different types (beam and fraction), belong to different plans, don't have the referred plan, don't have all beams/fractions included in the operation
    - AllowMixedDoseTypes = 1
        - Allow the resample operation to be performed on doses that are not all type beam, or all fraction
    - AllowMixedPlans = 1 << 1
        - Allow doses that don't refer to the same plan
    - AllowBeamDosesWithoutPlan = 1 << 2
        - Allow the resample operation to be performed on beam doses that are missing the referred plan
    - AllowIncompleteBeamDoses = 1 << 3
        - Allow the resample operation to be performed on beam doses that don't include all beams in the plan
    - AllowFractionDosesWithoutPlan = 1 << 4
        - Allow the resample operation to be performed on fraction doses that are missing the referred plan
    - AllowIncompleteFractionDoses = 1 << 5
        - Allow the resample operation to be performed on beam doses that don't include all fractions in the plan
    - AllowAll = (AllowIncompleteFractionDoses << 1) - 1
        - Allow all above conditions while resampling the dose files
- getVolumetricData(volumeId)
    - Return type
        - ScaledVolumeData
    - Description
        - Retrieves volume data in volumetric form. The volumeId must loaded as primary volume. On error, the data member of the volume data will be empty.
- getVolumeOverlapRegionDICOM()

        - Return Type
            - BoundingRegion
        - Description
            - Determines the overlapping region between the currently set Primary and Secondary volumes and returns the starting and ending points which define this overlapping region. Note that the start and end point coordinates will be given in the DICOM coordinate system of the currently set PRIMARY volume.
            - Note, if only one volume is currently loaded then that volume's region is returned.
            - Returns the overlapping region of the two volumes. Note that the points are specified in the DICOM coordinates of the currently set PRIMARY volume.
    - getErrorMessage()
        - Return type
            - String
        - Description
            - Returns an error message based on the error that has occurred
    - getLocalizedErrorMessage()
        - Return type
            - String
        - Description
            - Returns a localized error message based on the error that has occurred
- Inherited methods
    - From __builtin__.SwigPyObject
        - All __builtin__.SwigPyObject methods

Other

- Other defined data structures
    - DeleteFlags
        - AllowNone
            - No exceptions allowed
        - AllowDeleteFromSession
            - Allows delete when the object id is present in a session
        - AllowAll
            - All exceptions allowed
    - ResampleFlags
        - AllowNone
            - No exceptions allowed while resampling
        - AllowMixedDoseTypes
            - Doses in resample operation have different dose summation types (BEAM, FRACTION, PLAN)
        - AllowMixedPlans
            - Beam or fraction doses in resample operation are associated with different plan objects
        - AllowBeamDosesWithoutPlan
            - Referenced plan is not available for BEAM doses
        - AllowIncompleteBeamDoses
            - Resample operation doesn't include all beams in a plan
        - AllowFractionDosesWithoutPlan
            - Referenced plan is not available for FRACTION doses
        - AllowIncompleteFractionDoses
            - Resample operation doesn't include all fractions in a plan
        - AllowAll
            - All exceptions allowed