

TstML

Generated by Doxygen 1.8.13

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	Class Documentation	5
3.1	TstML::Executor::AbstractExecutionContext Class Reference	5
3.1.1	Detailed Description	5
3.1.2	Member Function Documentation	5
3.1.2.1	createNodeExecutor()	6
3.2	TstML::Executor::AbstractNodeExecutor Class Reference	6
3.2.1	Detailed Description	7
3.2.2	Member Function Documentation	7
3.2.2.1	abort()	7
3.2.2.2	context()	8
3.2.2.3	executionFinished()	8
3.2.2.4	node()	8
3.2.2.5	pause()	8
3.2.2.6	resume()	8
3.2.2.7	start()	9
3.2.2.8	stop()	9
3.2.2.9	waitForFinished()	9
3.3	TstML::Executor::AbstractNodeExecutorFactory Class Reference	9

3.4	TstML::Executor::DefaultNodeExecutor::Concurrent Class Reference	10
3.4.1	Detailed Description	11
3.4.2	Member Function Documentation	11
3.4.2.1	abort()	11
3.4.2.2	pause()	11
3.4.2.3	resume()	11
3.4.2.4	start()	12
3.4.2.5	stop()	12
3.5	TstML::Executor::ExecutionStatus Class Reference	12
3.5.1	Detailed Description	13
3.5.2	Member Function Documentation	13
3.5.2.1	Aborted()	13
3.5.2.2	AlreadyStarted()	13
3.5.2.3	FailedToCreateExecutor()	13
3.5.2.4	Finished()	13
3.5.2.5	isFailed()	14
3.5.2.6	message()	14
3.5.2.7	NotStarted()	14
3.5.2.8	operator" " ()	14
3.5.2.9	Paused()	14
3.5.2.10	Running()	15
3.5.2.11	Started()	15
3.5.2.12	type()	15
3.6	TstML::Executor::Executor Class Reference	15
3.6.1	Detailed Description	16
3.6.2	Member Function Documentation	16
3.6.2.1	abort()	16
3.6.2.2	pause()	16
3.6.2.3	resume()	16
3.6.2.4	start()	16

3.6.2.5	stop()	16
3.6.2.6	waitForFinished()	17
3.7	TstML::Executor::NodeExecutorRegistry Class Reference	17
3.7.1	Detailed Description	17
3.7.2	Member Function Documentation	17
3.7.2.1	create()	17
3.7.2.2	registerNodeExecutor()	18
3.8	TstML::Executor::DefaultNodeExecutor::Sequence Class Reference	18
3.8.1	Detailed Description	19
3.8.2	Member Function Documentation	19
3.8.2.1	abort()	20
3.8.2.2	pause()	20
3.8.2.3	resume()	20
3.8.2.4	start()	20
3.8.2.5	stop()	20
3.9	TstML::TSTModuleSingleton Class Reference	21
3.10	TstML::TSTNode Class Reference	22
3.10.1	Detailed Description	23
3.10.2	Member Function Documentation	23
3.10.2.1	addNode()	24
3.10.2.2	fromJson()	24
3.10.2.3	getParameter()	24
3.10.2.4	hasParameter()	24
3.10.2.5	load()	24
3.10.2.6	model()	25
3.10.2.7	nodes()	25
3.10.2.8	removeNode()	25
3.11	TstML::TSTNodeIO Class Reference	25
3.12	TstML::TSTNodeModel Class Reference	26
3.12.1	Detailed Description	27

3.12.2	Member Function Documentation	28
3.12.2.1	createNode()	28
3.12.2.2	name()	28
3.12.2.3	type()	28
3.13	TstML::TSTNodeModelsRegistry Class Reference	29
3.13.1	Detailed Description	30
3.13.2	Member Function Documentation	30
3.13.2.1	load()	30
3.13.2.2	loadDirectory()	30
3.13.2.3	model()	31
3.13.2.4	models()	31
3.13.2.5	print()	31
3.13.2.6	setQmlModuleUri()	31
3.13.3	Property Documentation	31
3.13.3.1	qmlModuleUri	31
3.14	TstML::TSTNodeParameterModel Class Reference	32
Index		35

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

TstML::Executor::AbstractExecutionContext	5
TstML::Executor::AbstractNodeExecutor	6
TstML::Executor::DefaultNodeExecutor::Concurrent	10
TstML::Executor::DefaultNodeExecutor::Sequence	18
TstML::Executor::AbstractNodeExecutorFactory	9
TstML::Executor::ExecutionStatus	12
TstML::Executor::Executor	15
TstML::Executor::NodeExecutorRegistry	17
QObject	
TstML::TSTModuleSingleton	21
TstML::TSTNode	22
TstML::TSTNodeIO	25
TstML::TSTNodeModel	26
TstML::TSTNodeModelsRegistry	29
TstML::TSTNodeParameterModel	32

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

TstML::Executor::AbstractExecutionContext	5
TstML::Executor::AbstractNodeExecutor	6
TstML::Executor::AbstractNodeExecutorFactory	9
TstML::Executor::DefaultNodeExecutor::Concurrent	10
TstML::Executor::ExecutionStatus	12
TstML::Executor::Executor	15
TstML::Executor::NodeExecutorRegistry	17
TstML::Executor::DefaultNodeExecutor::Sequence	18
TstML::TSTModuleSingleton	21
TstML::TSTNode	22
TstML::TSTNodeIO	25
TstML::TSTNodeModel	26
TstML::TSTNodeModelsRegistry	29
TstML::TSTNodeParameterModel	32

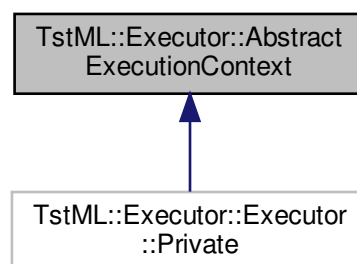
Chapter 3

Class Documentation

3.1 TstML::Executor::AbstractExecutionContext Class Reference

```
#include <AbstractExecutionContext.h>
```

Inheritance diagram for TstML::Executor::AbstractExecutionContext:



Public Member Functions

- virtual `AbstractNodeExecutor` * `createNodeExecutor` (const `TSTNode` * _node)=0

3.1.1 Detailed Description

Represent the execution context.

3.1.2 Member Function Documentation

3.1.2.1 createNodeExecutor()

```
virtual AbstractNodeExecutor* TstML::Executor::AbstractExecutionContext::createNodeExecutor (
    const TSTNode * _node ) [pure virtual]
```

Call this to create an executor for the given `_node`. This function will call

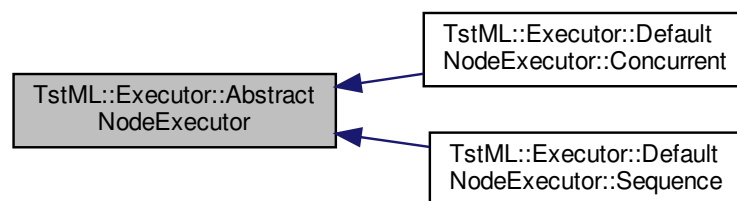
The documentation for this class was generated from the following files:

- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/Executor/AbstractExecutionContext.h
- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/Executor/AbstractExecutionContext.cpp

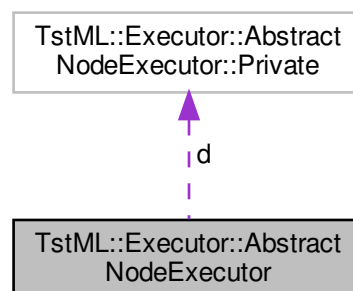
3.2 TstML::Executor::AbstractNodeExecutor Class Reference

```
#include <AbstractNodeExecutor.h>
```

Inheritance diagram for TstML::Executor::AbstractNodeExecutor:



Collaboration diagram for TstML::Executor::AbstractNodeExecutor:



Classes

- struct **Private**

Public Member Functions

- const [TSTNode](#) * [node](#) () const
- [AbstractExecutionContext](#) * [context](#) () const
- virtual [ExecutionStatus](#) [start](#) ()=0
- virtual [ExecutionStatus](#) [pause](#) ()=0
- virtual [ExecutionStatus](#) [resume](#) ()=0
- virtual [ExecutionStatus](#) [stop](#) ()=0
- virtual [ExecutionStatus](#) [abort](#) ()=0
- [ExecutionStatus](#) [waitForFinished](#) (unsigned long time=std::numeric_limits< unsigned long >::max()) const

Protected Member Functions

- **AbstractNodeExecutor** (const [TSTNode](#) *_node, [AbstractExecutionContext](#) *_context)
- **AbstractNodeExecutor** (Private *_d, const [TSTNode](#) *_node, [AbstractExecutionContext](#) *_context)
- void [executionFinished](#) (const [ExecutionStatus](#) &_status)

Protected Attributes

- Private *const **d**

3.2.1 Detailed Description

Base class to define the execution of a [TSTNode](#).

3.2.2 Member Function Documentation

3.2.2.1 abort()

```
virtual ExecutionStatus TstML::Executor::AbstractNodeExecutor::abort ( ) [pure virtual]
```

Called to abort the execution of a node,

Implemented in [TstML::Executor::DefaultNodeExecutor::Concurrent](#), and [TstML::Executor::DefaultNodeExecutor::Sequence](#).

3.2.2.2 context()

```
AbstractExecutionContext * AbstractNodeExecutor::context ( ) const
```

Returns

the [AbstractExecutionContext](#) associated with this node executor.

3.2.2.3 executionFinished()

```
void AbstractNodeExecutor::executionFinished (
    const ExecutionStatus & _status ) [protected]
```

Call this function when the execution is finished.

3.2.2.4 node()

```
const TstML::TSTNode * AbstractNodeExecutor::node ( ) const
```

Returns

the [TSTNode](#) associated with this node executor.

3.2.2.5 pause()

```
virtual ExecutionStatus TstML::Executor::AbstractNodeExecutor::pause ( ) [pure virtual]
```

Called to pause the execution of a node,

Implemented in [TstML::Executor::DefaultNodeExecutor::Concurrent](#), and [TstML::Executor::DefaultNodeExecutor::Sequence](#).

3.2.2.6 resume()

```
virtual ExecutionStatus TstML::Executor::AbstractNodeExecutor::resume ( ) [pure virtual]
```

Called to resume the execution of a node,

Implemented in [TstML::Executor::DefaultNodeExecutor::Concurrent](#), and [TstML::Executor::DefaultNodeExecutor::Sequence](#).

3.2.2.7 start()

```
virtual ExecutionStatus TstML::Executor::AbstractNodeExecutor::start ( ) [pure virtual]
```

Called to start the execution of a node.

It is expected that this function should return immediately, and it should not block.

Implemented in [TstML::Executor::DefaultNodeExecutor::Concurrent](#), and [TstML::Executor::DefaultNodeExecutor↵
::Sequence](#).

3.2.2.8 stop()

```
virtual ExecutionStatus TstML::Executor::AbstractNodeExecutor::stop ( ) [pure virtual]
```

Called to stop the execution of a node,

Implemented in [TstML::Executor::DefaultNodeExecutor::Concurrent](#), and [TstML::Executor::DefaultNodeExecutor↵
::Sequence](#).

3.2.2.9 waitForFinished()

```
ExecutionStatus AbstractNodeExecutor::waitForFinished (
    unsigned long time = std::numeric_limits<unsigned long>::max() ) const
```

Call this function to wait for the execution to finish

Parameters

<code>_timeout</code>	given in ms, the function will exit after the timeout, and return <code>ExecutionStatus::InProgress</code> .
-----------------------	--

The documentation for this class was generated from the following files:

- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/Executor/AbstractNodeExecutor.h
- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/Executor/AbstractNodeExecutor.cpp

3.3 TstML::Executor::AbstractNodeExecutorFactory Class Reference

Public Member Functions

- virtual [AbstractNodeExecutor](#) * **create** (const [TstML::TSTNode](#) * _node, [AbstractExecutionContext](#) * _↵
context) const =0

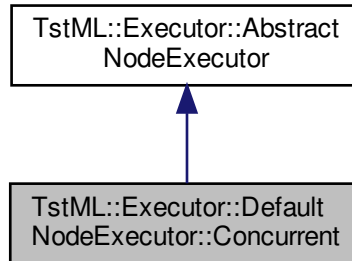
The documentation for this class was generated from the following files:

- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/Executor/AbstractNodeExecutorFactory.h
- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/Executor/AbstractNodeExecutorFactory.cpp

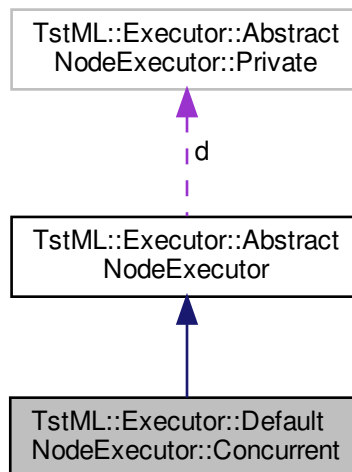
3.4 TstML::Executor::DefaultNodeExecutor::Concurrent Class Reference

```
#include <Concurrent.h>
```

Inheritance diagram for TstML::Executor::DefaultNodeExecutor::Concurrent:



Collaboration diagram for TstML::Executor::DefaultNodeExecutor::Concurrent:



Classes

- struct **Private**

Public Member Functions

- **Concurrent** (const [TSTNode](#) *_node, [AbstractExecutionContext](#) *_context)
- [ExecutionStatus start](#) () override
- [ExecutionStatus pause](#) () override
- [ExecutionStatus resume](#) () override
- [ExecutionStatus stop](#) () override
- [ExecutionStatus abort](#) () override

Additional Inherited Members

3.4.1 Detailed Description

Default implementation of a concurrent executor.

3.4.2 Member Function Documentation

3.4.2.1 abort()

[ExecutionStatus](#) Concurrent::abort () [override], [virtual]

Called to abort the execution of a node,

Implements [TstML::Executor::AbstractNodeExecutor](#).

3.4.2.2 pause()

[ExecutionStatus](#) Concurrent::pause () [override], [virtual]

Called to pause the execution of a node,

Implements [TstML::Executor::AbstractNodeExecutor](#).

3.4.2.3 resume()

[ExecutionStatus](#) Concurrent::resume () [override], [virtual]

Called to resume the execution of a node,

Implements [TstML::Executor::AbstractNodeExecutor](#).

3.4.2.4 start()

```
ExecutionStatus Concurrent::start ( ) [override], [virtual]
```

Called to start the execution of a node.

It is expected that this function should return immediately, and it should not block.

Implements [TstML::Executor::AbstractNodeExecutor](#).

3.4.2.5 stop()

```
ExecutionStatus Concurrent::stop ( ) [override], [virtual]
```

Called to stop the execution of a node,

Implements [TstML::Executor::AbstractNodeExecutor](#).

The documentation for this class was generated from the following files:

- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/Executor/DefaultNodeExecutor/Concurrent.h
- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/Executor/DefaultNodeExecutor/Concurrent.cpp

3.5 TstML::Executor::ExecutionStatus Class Reference

```
#include <ExecutionStatus.h>
```

Public Types

- enum **Type** {
 Invalid, **Started**, **Paused**, **Running**,
 Finished, **Aborted**, **Failed** }

Public Member Functions

- **ExecutionStatus** (const [ExecutionStatus](#) &)=default
- **ExecutionStatus** ([ExecutionStatus](#) &&)=default
- [ExecutionStatus](#) & **operator=** (const [ExecutionStatus](#) &)=default
- bool **operator==** (const [ExecutionStatus](#) &_rhs) const
- bool **operator!=** (const [ExecutionStatus](#) &_rhs) const
- const [ExecutionStatus](#) & **operator||** (const [ExecutionStatus](#) &_rhs) const
- Type **type** () const
- QString **message** () const
- bool **isFailed** () const

Static Public Member Functions

- static [ExecutionStatus Started](#) ()
- static [ExecutionStatus Running](#) ()
- static [ExecutionStatus Paused](#) ()
- static [ExecutionStatus Aborted](#) ()
- static [ExecutionStatus Finished](#) ()
- static [ExecutionStatus FailedToCreateExecutor](#) (const [TSTNode](#) *_node)
- static [ExecutionStatus AlreadyStarted](#) (const [TSTNode](#) *_node)
- static [ExecutionStatus NotStarted](#) (const [TSTNode](#) *_node)

3.5.1 Detailed Description

Class that holds the execution status

3.5.2 Member Function Documentation

3.5.2.1 Aborted()

```
static ExecutionStatus TstML::Executor::ExecutionStatus::Aborted ( ) [static]
```

Return the [ExecutionStatus](#) corresponding to a successfully aborted execution And it is the expected results of [AbstractNodeExecutor::abort](#).

3.5.2.2 AlreadyStarted()

```
ExecutionStatus ExecutionStatus::AlreadyStarted (
    const TSTNode * _node ) [static]
```

Return the [ExecutionStatus](#) corresponding to failing to start, because `_node` is already started.

3.5.2.3 FailedToCreateExecutor()

```
ExecutionStatus ExecutionStatus::FailedToCreateExecutor (
    const TSTNode * _node ) [static]
```

Return the [ExecutionStatus](#) corresponding to failing to create an executor for `_node`.

3.5.2.4 Finished()

```
static ExecutionStatus TstML::Executor::ExecutionStatus::Finished ( ) [static]
```

Return the [ExecutionStatus](#) corresponding to a successfully completed execution It is returned by [AbstractNode↵Executor::waitForFinished](#) when the execution is completed.

3.5.2.5 isFailed()

```
bool TstML::Executor::ExecutionStatus::isFailed ( ) const [inline]
```

Convenient function to check if the status is failed

3.5.2.6 message()

```
QString TstML::Executor::ExecutionStatus::message ( ) const [inline]
```

Returns

the message associated with the status, only valid if [type](#) is equal to `Type::Failed`.

3.5.2.7 NotStarted()

```
ExecutionStatus ExecutionStatus::NotStarted (
    const TSTNode * _node ) [static]
```

Return the [ExecutionStatus](#) corresponding to failing to act, because `_node` is not started.

3.5.2.8 operator" | " |()

```
const ExecutionStatus& TstML::Executor::ExecutionStatus::operator|| (
    const ExecutionStatus & _rhs ) const [inline]
```

If if this [ExecutionStatus](#) is invalid, it return `_rhs` otherwise it return this.

For instance:

```
(ExecutionStatus() or ExecutionStatus::Finished()) == ExecutionStatus::
    Finished()
(ExecutionStatus::Paused() or ExecutionStatus::Finished()) == ExecutionStatus::
    Paused()
```

3.5.2.9 Paused()

```
static ExecutionStatus TstML::Executor::ExecutionStatus::Paused ( ) [static]
```

Return the [ExecutionStatus](#) corresponding to a successfully paused execution And it is the expected results of [AbstractNodeExecutor::pause](#).

3.5.2.10 Running()

```
static ExecutionStatus TstML::Executor::ExecutionStatus::Running ( ) [static]
```

Return the [ExecutionStatus](#) corresponding indicating the node is currently running. It is returned by [AbstractNode::waitFinished](#) when the function timeout. And it is the expected results of [AbstractNodeExecutor::resume](#).

3.5.2.11 Started()

```
static ExecutionStatus TstML::Executor::ExecutionStatus::Started ( ) [static]
```

Return the [ExecutionStatus](#) corresponding to a started execution And it is the expected results of [AbstractNodeExecutor::start](#).

3.5.2.12 type()

```
Type TstML::Executor::ExecutionStatus::type ( ) const [inline]
```

Returns

the status type

The documentation for this class was generated from the following files:

- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/Executor/ExecutionStatus.h
- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/Executor/ExecutionStatus.cpp

3.6 TstML::Executor::Executor Class Reference

```
#include <Executor.h>
```

Classes

- struct **Private**

Public Member Functions

- **Executor** (TSTNode *_node, [NodeExecutorRegistry](#) *_registry)
- [ExecutionStatus](#) start ()
- [ExecutionStatus](#) pause ()
- [ExecutionStatus](#) resume ()
- [ExecutionStatus](#) stop ()
- [ExecutionStatus](#) abort ()
- [ExecutionStatus](#) waitFinished (unsigned long time=std::numeric_limits< unsigned long >::max()) const

3.6.1 Detailed Description

Handles the execution of a [TSTNode](#).

3.6.2 Member Function Documentation

3.6.2.1 abort()

`ExecutionStatus` `Executor::abort ( )`

Called to abort the execution of the tree,

3.6.2.2 pause()

`ExecutionStatus` `Executor::pause ( )`

Called to pause the execution of the tree,

3.6.2.3 resume()

`ExecutionStatus` `Executor::resume ( )`

Called to resume the execution of the tree,

3.6.2.4 start()

`ExecutionStatus` `Executor::start ( )`

Called to start the execution of the tree.

It is expected that this function should return immediately, and it should not block.

3.6.2.5 stop()

`ExecutionStatus` `Executor::stop ( )`

Called to stop the execution of the tree,

3.6.2.6 waitForFinished()

```
ExecutionStatus Executor::waitForFinished (
    unsigned long time = std::numeric_limits<unsigned long>::max() ) const
```

Wait for the execution to be finished

The documentation for this class was generated from the following files:

- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/Executor/Executor.h
- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/Executor/Executor.cpp

3.7 TstML::Executor::NodeExecutorRegistry Class Reference

```
#include <NodeExecutorRegistry.h>
```

Classes

- struct **Private**

Public Member Functions

- [AbstractNodeExecutor](#) * [create](#) (const [TstML::TSTNode](#) * _node, [AbstractExecutionContext](#) * _context)
- template<typename [T_](#)>
void **registerNodeExecutor** (const [TSTNodeModel](#) * _node_model)
- void [registerNodeExecutor](#) (const [TSTNodeModel](#) * _node_model, const [AbstractNodeExecutorFactory](#) * _
factory)

3.7.1 Detailed Description

Holds factories used to create [AbstractNodeExecutor](#) for the tst node of a tree.

3.7.2 Member Function Documentation

3.7.2.1 create()

```
AbstractNodeExecutor * NodeExecutorRegistry::create (
    const TstML::TSTNode * _node,
    AbstractExecutionContext * _context )
```

Create an [AbstractNodeExecutor](#) for the given _node.

Returns

a null pointer if there are no factories for the model of the _node.

3.7.2.2 registerNodeExecutor()

```
void NodeExecutorRegistry::registerNodeExecutor (
    const TSTNodeModel * _node_model,
    const AbstractNodeExecutorFactory * _factory )
```

Associate `_factory` to be used in `create` to create `AbstractNodeExecutor` for the given `_node_model`.

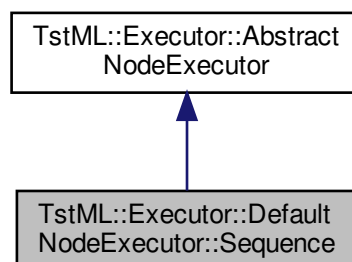
The documentation for this class was generated from the following files:

- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/Executor/NodeExecutorRegistry.h
- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/Executor/NodeExecutorRegistry.cpp

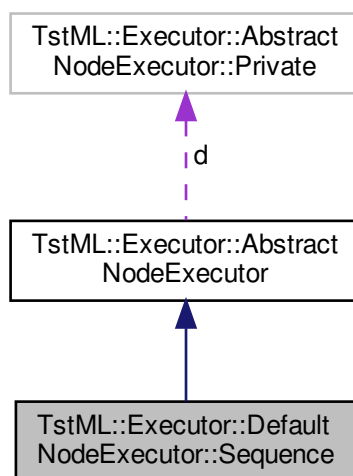
3.8 TstML::Executor::DefaultNodeExecutor::Sequence Class Reference

```
#include <Sequence.h>
```

Inheritance diagram for TstML::Executor::DefaultNodeExecutor::Sequence:



Collaboration diagram for TstML::Executor::DefaultNodeExecutor::Sequence:



Classes

- struct **Private**

Public Member Functions

- **Sequence** (const [TSTNode](#) *_node, [AbstractExecutionContext](#) *_context)
- [ExecutionStatus start](#) () override
- [ExecutionStatus pause](#) () override
- [ExecutionStatus resume](#) () override
- [ExecutionStatus stop](#) () override
- [ExecutionStatus abort](#) () override

Additional Inherited Members

3.8.1 Detailed Description

Default implementation of a sequence executor.

3.8.2 Member Function Documentation

3.8.2.1 abort()

`ExecutionStatus` `Sequence::abort ( )` `[override]`, `[virtual]`

Called to abort the execution of a node,

Implements [TstML::Executor::AbstractNodeExecutor](#).

3.8.2.2 pause()

`ExecutionStatus` `Sequence::pause ( )` `[override]`, `[virtual]`

Called to pause the execution of a node,

Implements [TstML::Executor::AbstractNodeExecutor](#).

3.8.2.3 resume()

`ExecutionStatus` `Sequence::resume ( )` `[override]`, `[virtual]`

Called to resume the execution of a node,

Implements [TstML::Executor::AbstractNodeExecutor](#).

3.8.2.4 start()

`ExecutionStatus` `Sequence::start ( )` `[override]`, `[virtual]`

Called to start the execution of a node.

It is expected that this function should return immediately, and it should not block.

Implements [TstML::Executor::AbstractNodeExecutor](#).

3.8.2.5 stop()

`ExecutionStatus` `Sequence::stop ( )` `[override]`, `[virtual]`

Called to stop the execution of a node,

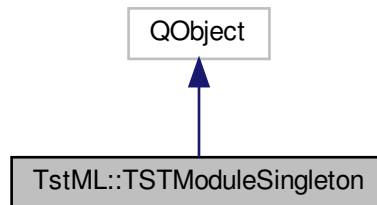
Implements [TstML::Executor::AbstractNodeExecutor](#).

The documentation for this class was generated from the following files:

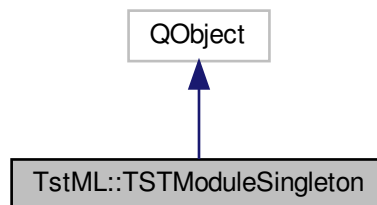
- `/courses/TDDE05/software/lrs-pkg/src/TstML/TstML/Executor/DefaultNodeExecutor/Sequence.h`
- `/courses/TDDE05/software/lrs-pkg/src/TstML/TstML/Executor/DefaultNodeExecutor/Sequence.cpp`

3.9 TstML::TSTModuleSingleton Class Reference

Inheritance diagram for TstML::TSTModuleSingleton:



Collaboration diagram for TstML::TSTModuleSingleton:



Public Member Functions

- **TSTModuleSingleton** (QHash< QString, [TSTNodeModelsRegistry](#) *> *_registry)
- Q_INVOKABLE [TstML::TSTNodeModel](#) * **model** (const QString &_uri, const QString &_name)

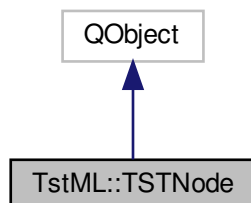
The documentation for this class was generated from the following files:

- `/courses/TDDE05/software/lrs-pkg/src/TstML/TstML/TSTModuleSingleton.h`
- `/courses/TDDE05/software/lrs-pkg/src/TstML/TstML/TSTModuleSingleton.cpp`

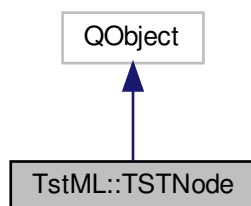
3.10 TstML::TSTNode Class Reference

```
#include <TSTNode.h>
```

Inheritance diagram for TstML::TSTNode:



Collaboration diagram for TstML::TSTNode:



Classes

- struct **Private**

Public Types

- enum **ParameterType** { **Common**, **Specific** }

Signals

- void **nodesChanged** ()
- void **colorChanged** ()
- void **labelChanged** ()
- void **dataChanged** ()
- void **parentChanged** ()

Public Member Functions

- **TSTNode** (const [TSTNodeModel](#) * _model=nullptr, QObject *parent=0)
- QList< [TSTNode](#) * > **nodes** () const
- Q_INVOKABLE void **addNode** ([TstML::TSTNode](#) * _tst_node)
- Q_INVOKABLE void **removeNode** ([TstML::TSTNode](#) * _tst_node)
- const [TSTNodeModel](#) * **model** () const
- QColor **color** () const
- QString **label** () const
- void **setLabel** (const QString & _label)
- Q_INVOKABLE [TstML::TSTNode](#) * **clone** () const
- QJsonObject **toJson** () const
- Q_INVOKABLE QString **toJsonString** () const
- Q_INVOKABLE QString **toQml** () const
- Q_INVOKABLE bool **save** (const QUrl & _url)
- Q_INVOKABLE QVariant **getParameter** (ParameterType _parameterType, const QString & _name) const
- Q_INVOKABLE void **setParameter** (ParameterType _parameterType, const QString & _name, const Q< Variant & _value)
- Q_INVOKABLE bool **hasParameter** (ParameterType _parameterType, const QString & _name) const
- Q_INVOKABLE void **removeParameter** (ParameterType _parameterType, const QString & _name)

Static Public Member Functions

- static [TSTNode](#) * **fromJson** (const QJsonObject & _value, const [TstML::TSTNodeModelsRegistry](#) * _registry)
- static [TSTNode](#) * **fromJson** (const QByteArray & _value, const [TstML::TSTNodeModelsRegistry](#) * _registry)
- static [TSTNode](#) * **fromQml** (const QString & _code, QQmlEngine * _engine, const QUrl & _url)
- static Q_INVOKABLE [TSTNode](#) * **load** (const QUrl & _url, const [TSTNodeModelsRegistry](#) * _registry, Q< QmlEngine * _engine=nullptr)

Properties

- [TstML::TSTNode](#) **parent**
- QList< QObject * > **nodes**
- QColor **color**
- QString **label**
- [TstML::TSTNodeModel](#) **model**
- QQmlListProperty< QObject > **data**

3.10.1 Detailed Description

Represents an instance of a node in a TST.

3.10.2 Member Function Documentation

3.10.2.1 addNode()

```
void TSTNode::addNode (
    TstML::TSTNode * _tst_node )
```

Add `_tst_node` to the list of children nodes. This class takes ownership of `_tst_node` (and set itself as the parent).

3.10.2.2 fromJson()

```
TSTNode * TSTNode::fromJson (
    const QByteArray & _value,
    const TstML::TSTNodeModelsRegistry * _registry ) [static]
```

Construct a `TSTNode` from a Json string

3.10.2.3 getParameter()

```
QVariant TSTNode::getParameter (
    ParameterType _parameterType,
    const QString & _name ) const
```

Returns

the value for the parameter named `_name` from the type `_parameterType`.

3.10.2.4 hasParameter()

```
bool TSTNode::hasParameter (
    ParameterType _parameterType,
    const QString & _name ) const
```

Returns

if the parameter named `_name` from the type `_parameterType` exists.

3.10.2.5 load()

```
TSTNode * TSTNode::load (
    const QUrl & _url,
    const TSTNodeModelsRegistry * _registry,
    QQmlEngine * _engine = nullptr ) [static]
```

Load a `TSTNode` from the file given in `_url` using the model registry `_registry`. `_engine` is optional and only needed when loading Tst nodes from QML files.

3.10.2.6 model()

```
const TSTNodeModel* TstML::TSTNode::model ( ) const
```

Returns

the model associated with this node.

3.10.2.7 nodes()

```
QList<TSTNode*> TstML::TSTNode::nodes ( ) const
```

Returns

the list of children nodes for this node.

3.10.2.8 removeNode()

```
void TSTNode::removeNode (
    TstML::TSTNode * _tst_node )
```

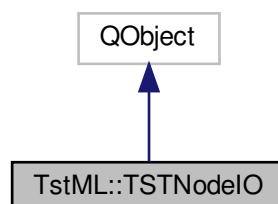
Remove `_tst_node` from the list of children nodes. The caller takes ownership of `_tst_node`, the parent is set to nullptr, it is the responsibility of the caller to delete `_tst_node`.

The documentation for this class was generated from the following files:

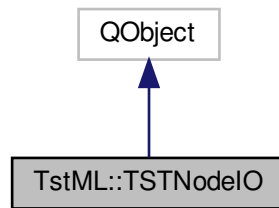
- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/TSTNode.h
- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/TSTNode.cpp

3.11 TstML::TSTNodeIO Class Reference

Inheritance diagram for TstML::TSTNodeIO:



Collaboration diagram for TstML::TSTNodeIO:



Public Member Functions

- **TSTNodeIO** (QObject *_parent=0)
- Q_INVOKABLE [TstML::TSTNode](#) * **load** (const QUrl &_url, QObject *_registry) const

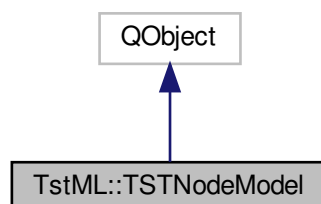
The documentation for this class was generated from the following files:

- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/TSTNodeIO_p.h
- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/TSTNodeIO.cpp

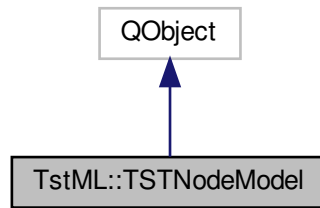
3.12 TstML::TSTNodeModel Class Reference

```
#include <TSTNodeModel.h>
```

Inheritance diagram for TstML::TSTNodeModel:



Collaboration diagram for TstML::TSTNodeModel:



Classes

- struct **Private**

Public Types

- enum **Type** : uint { **Terminal**, **NonTerminal** }

Public Member Functions

- **TSTNodeModel** (const QString &_name, Type _type, int _number_of_children, const QList< [TSTNodeParameterModel](#) * > &_parameters, [TSTNodeModelsRegistry](#) *parent)
- const [TSTNodeModelsRegistry](#) * **registry** () const
- QString **name** () const
- QColor **color** () const
- Type **type** () const
- int **numberOfChildren** () const
- Q_INVOKABLE [TstML::TSTNode](#) * **createNode** () const
- QList< [TSTNodeParameterModel](#) * > **parameters** () const

Properties

- QString **label**
- QColor **color**
- QList< QObject * > **parameterModels**
- int **numberOfChildren**

3.12.1 Detailed Description

Represent the model (or specification) of a type of TST Node.

3.12.2 Member Function Documentation

3.12.2.1 createNode()

```
TSTNode * TSTNodeModel::createNode ( ) const
```

Create a node with default value

3.12.2.2 name()

```
QString TSTNodeModel::name ( ) const
```

Returns

the name of the model

3.12.2.3 type()

```
TSTNodeModel::Type TSTNodeModel::type ( ) const
```

Returns

the type of the model

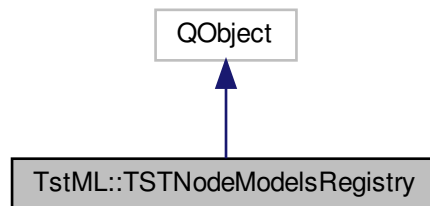
The documentation for this class was generated from the following files:

- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/TSTNodeModel.h
- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/TSTNodeModel.cpp

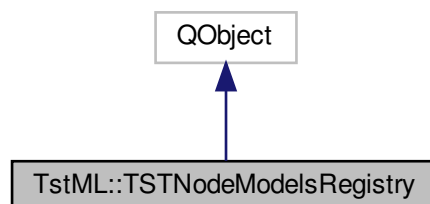
3.13 TstML::TSTNodeModelsRegistry Class Reference

```
#include <TSTNodeModelsRegistry.h>
```

Inheritance diagram for TstML::TSTNodeModelsRegistry:



Collaboration diagram for TstML::TSTNodeModelsRegistry:



Classes

- struct **Private**

Signals

- void **modelsChanged** ()
- void **commonParametersChanged** ()
- void **qmlModuleUriChanged** ()

Public Member Functions

- **TSTNodeModelsRegistry** (QObject *parent=0)
- QList< [TSTNodeModel](#) * > [models](#) () const
- void [print](#) () const
- const [TSTNodeModel](#) * [model](#) (const QString &_name) const
- QString [qmlModuleUri](#) () const
- void [setQmlModuleUri](#) (const QString &_uri)
- Q_INVOKABLE bool [load](#) (const QUrl &_url)
- Q_INVOKABLE bool [load](#) (const QString &_filename)
- Q_INVOKABLE bool [loadFromString](#) (const QByteArray &_rawdata)
- bool [loadFromJson](#) (const QJsonDocument &_rawdata)
- Q_INVOKABLE bool [loadDirectory](#) (const QUrl &_url)
- Q_INVOKABLE bool [loadDirectory](#) (const QString &_dirname)

Properties

- QList< QObject * > **models**
- QList< QObject * > **commonParameters**
- QString [qmlModuleUri](#)

3.13.1 Detailed Description

Registry of [TSTNodeModel](#) used to define TST.

3.13.2 Member Function Documentation

3.13.2.1 load()

```
bool TSTNodeModelsRegistry::load (
    const QUrl & _url )
```

Load the JSON TST node definition

3.13.2.2 loadDirectory()

```
bool TSTNodeModelsRegistry::loadDirectory (
    const QUrl & _url )
```

Load the JSON TST config node from a directory

3.13.2.3 model()

```
const TSTNodeModel * TSTNodeModelsRegistry::model (
    const QString & _name ) const
```

Returns

the model named _name

3.13.2.4 models()

```
QList<TSTNodeModel*> TstML::TSTNodeModelsRegistry::models ( ) const
```

Returns

all the loaded models

3.13.2.5 print()

```
void TSTNodeModelsRegistry::print ( ) const
```

Print on the terminal the content of the registry

3.13.2.6 setQmlModuleUri()

```
void TSTNodeModelsRegistry::setQmlModuleUri (
    const QString & _uri )
```

Set the Uri used as QML module

3.13.3 Property Documentation

3.13.3.1 qmlModuleUri

```
QString TSTNodeModelsRegistry::qmlModuleUri [read], [write]
```

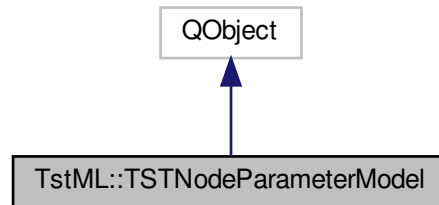
This property can be used to enable the generation of a qml module for TSTs (the value can be set only once!)

The documentation for this class was generated from the following files:

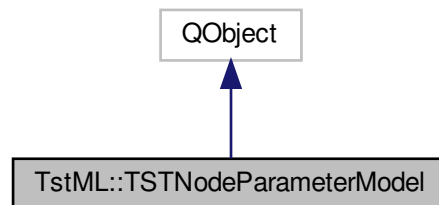
- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/TSTNodeModelsRegistry.h
- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/TSTNodeModelsRegistry.cpp

3.14 TstML::TSTNodeParameterModel Class Reference

Inheritance diagram for TstML::TSTNodeParameterModel:



Collaboration diagram for TstML::TSTNodeParameterModel:



Classes

- struct **Private**

Public Member Functions

- **TSTNodeParameterModel** (const QString &_name, const QString &_type, const QString &_description, bool _required, QObject *parent=0)
- QString **name** () const
- QString **type** () const
- QString **description** () const
- bool **required** () const
- QVariant **defaultValue** () const

Properties

- QString **name**
- QString **type**
- QString **description**
- bool **required**
- QVariant **defaultValue**

The documentation for this class was generated from the following files:

- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/TSTNodeParameterModel.h
- /courses/TDDE05/software/lrs-pkg/src/TstML/TstML/TSTNodeParameterModel.cpp

Index

abort
 TstML::Executor::AbstractNodeExecutor, 7
 TstML::Executor::DefaultNodeExecutor::Concurrent, 11
 TstML::Executor::DefaultNodeExecutor::Sequence, 19
 TstML::Executor::Executor, 16
Aborted
 TstML::Executor::ExecutionStatus, 13
addNode
 TstML::TSTNode, 23
AlreadyStarted
 TstML::Executor::ExecutionStatus, 13

context
 TstML::Executor::AbstractNodeExecutor, 7
create
 TstML::Executor::NodeExecutorRegistry, 17
createNode
 TstML::TSTNodeModel, 28
createNodeExecutor
 TstML::Executor::AbstractExecutionContext, 5

executionFinished
 TstML::Executor::AbstractNodeExecutor, 8

FailedToCreateExecutor
 TstML::Executor::ExecutionStatus, 13
Finished
 TstML::Executor::ExecutionStatus, 13
fromJson
 TstML::TSTNode, 24

getParameter
 TstML::TSTNode, 24

hasParameter
 TstML::TSTNode, 24

isFailed
 TstML::Executor::ExecutionStatus, 13

load
 TstML::TSTNode, 24
 TstML::TSTNodeModelsRegistry, 30
loadDirectory
 TstML::TSTNodeModelsRegistry, 30

message
 TstML::Executor::ExecutionStatus, 14
model
 TstML::TSTNode, 24
 TstML::TSTNodeModelsRegistry, 30
models
 TstML::TSTNodeModelsRegistry, 31

name
 TstML::TSTNodeModel, 28
node
 TstML::Executor::AbstractNodeExecutor, 8
nodes
 TstML::TSTNode, 25
NotStarted
 TstML::Executor::ExecutionStatus, 14

operator | |
 TstML::Executor::ExecutionStatus, 14

pause
 TstML::Executor::AbstractNodeExecutor, 8
 TstML::Executor::DefaultNodeExecutor::Concurrent, 11
 TstML::Executor::DefaultNodeExecutor::Sequence, 20
 TstML::Executor::Executor, 16
Paused
 TstML::Executor::ExecutionStatus, 14
print
 TstML::TSTNodeModelsRegistry, 31

qmlModuleUri
 TstML::TSTNodeModelsRegistry, 31

registerNodeExecutor
 TstML::Executor::NodeExecutorRegistry, 17
removeNode
 TstML::TSTNode, 25
resume
 TstML::Executor::AbstractNodeExecutor, 8
 TstML::Executor::DefaultNodeExecutor::Concurrent, 11
 TstML::Executor::DefaultNodeExecutor::Sequence, 20
 TstML::Executor::Executor, 16
Running
 TstML::Executor::ExecutionStatus, 14

setQmlModuleUri
 TstML::TSTNodeModelsRegistry, 31
start
 TstML::Executor::AbstractNodeExecutor, 8

- TstML::Executor::DefaultNodeExecutor::Concurrent, 11
- TstML::Executor::DefaultNodeExecutor::Sequence, 20
- TstML::Executor::Executor, 16
- Started
 - TstML::Executor::ExecutionStatus, 15
- stop
 - TstML::Executor::AbstractNodeExecutor, 9
 - TstML::Executor::DefaultNodeExecutor::Concurrent, 12
 - TstML::Executor::DefaultNodeExecutor::Sequence, 20
 - TstML::Executor::Executor, 16
- TstML::Executor::AbstractExecutionContext, 5
 - createNodeExecutor, 5
- TstML::Executor::AbstractNodeExecutor, 6
 - abort, 7
 - context, 7
 - executionFinished, 8
 - node, 8
 - pause, 8
 - resume, 8
 - start, 8
 - stop, 9
 - waitForFinished, 9
- TstML::Executor::AbstractNodeExecutorFactory, 9
- TstML::Executor::DefaultNodeExecutor::Concurrent, 10
 - abort, 11
 - pause, 11
 - resume, 11
 - start, 11
 - stop, 12
- TstML::Executor::DefaultNodeExecutor::Sequence, 18
 - abort, 19
 - pause, 20
 - resume, 20
 - start, 20
 - stop, 20
- TstML::Executor::ExecutionStatus, 12
 - Aborted, 13
 - AlreadyStarted, 13
 - FailedToCreateExecutor, 13
 - Finished, 13
 - isFailed, 13
 - message, 14
 - NotStarted, 14
 - operator | |, 14
 - Paused, 14
 - Running, 14
 - Started, 15
 - type, 15
- TstML::Executor::Executor, 15
 - abort, 16
 - pause, 16
 - resume, 16
 - start, 16
 - stop, 16
 - waitForFinished, 16
- TstML::Executor::NodeExecutorRegistry, 17
 - create, 17
 - registerNodeExecutor, 17
- TstML::TSTModuleSingleton, 21
- TstML::TSTNode, 22
 - addNode, 23
 - fromJson, 24
 - getParameter, 24
 - hasParameter, 24
 - load, 24
 - model, 24
 - nodes, 25
 - removeNode, 25
- TstML::TSTNodeIO, 25
- TstML::TSTNodeModel, 26
 - createNode, 28
 - name, 28
 - type, 28
- TstML::TSTNodeModelsRegistry, 29
 - load, 30
 - loadDirectory, 30
 - model, 30
 - models, 31
 - print, 31
 - qmlModuleUri, 31
 - setQmlModuleUri, 31
- TstML::TSTNodeParameterModel, 32
- type
 - TstML::Executor::ExecutionStatus, 15
 - TstML::TSTNodeModel, 28
- waitForFinished
 - TstML::Executor::AbstractNodeExecutor, 9
 - TstML::Executor::Executor, 16