



## Entity Package

<<Interface>>

### IEntityGroup

extends IEntity,  
IDictionary<\_Ty>

+virtual get CHILD\_TAG() -> string



For type check

(typeof obj is IEntityGroup)

Entity Group

\_Ty := implements IEntity

### EntityArray

extends Vector<\_Ty>  
implements IEntityGroup

+virtual get CHILD\_TAG() -> string

+EntityArray()

+virtual construct(XML)

+virtual toXML() -> XML

\_Ty := implements IEntity

### EntityList

extends List<\_Ty>  
implements IEntityGroup

+virtual get TAG() -> string

+EntityList()

+virtual construct(XML)

+virtual toXML() -> XML

Inherits to share same interface

<<Interface>>

### IEntity

+virtual get TAG() -> string

+virtual construct(XML)

+virtual key() -> any

+virtual toXML() -> XML

### Entity

implements IEntity

+Entity()

+virtual toXML() -> XML

## Entity is

To standardize expression method of data structure.

Provides I/O interfaces to/from XML and Invoke.

## Composite relationship

enable to realize 1:N hierarchical relationship

If EntityGroup's children type is same with the EntityGroup, it will be the 1:N recursive hierarchical relationship like a Tree

## When "Hierarchical Relationship"

Compose the data class(entity) having children by inheriting EntityGroup and terminate the leaf node by inheriting Entity.

Just define the XML I/O only for each variables, then about the data I/O, all will be done

## Utility Interfaces

<<Interface>>

### IHTMLEntity

#virtual get CSS() -> string

#virtual get HEADER() -> string

+toTR(...any[]) -> string

+toTH(...any[]) -> string

+toTD(any) -> string

+virtual toHTML() -> string

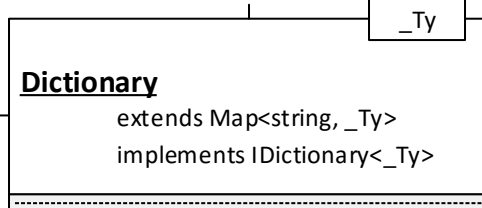
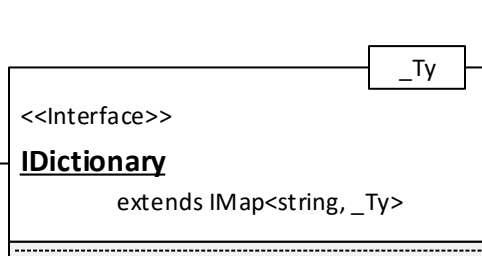
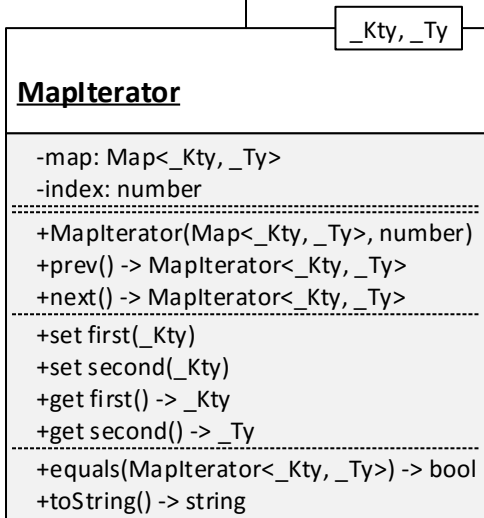
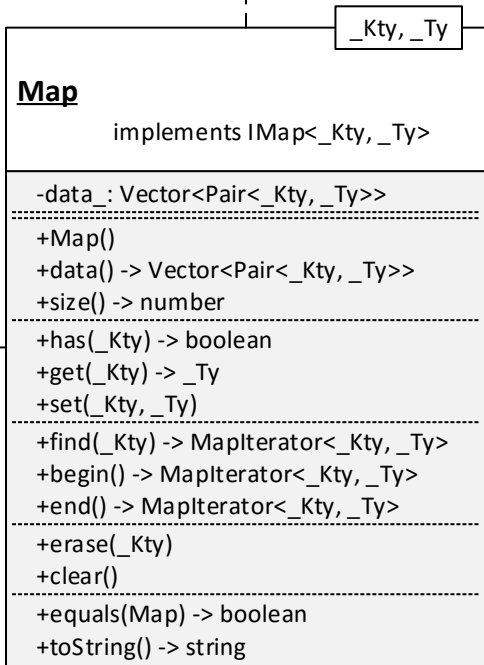
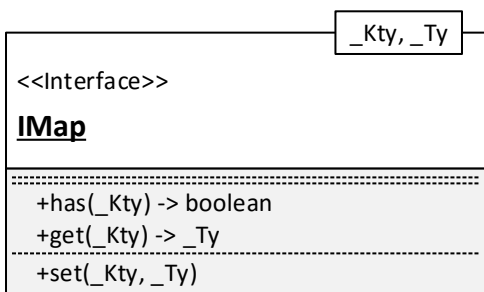
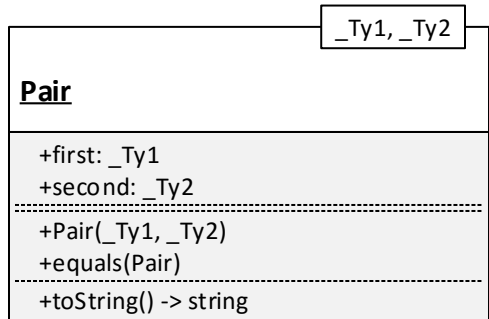
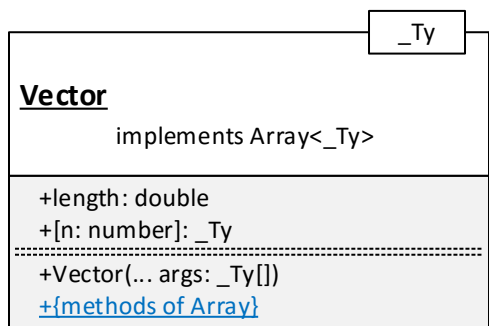
<<Interface>>

### ISpriteEntity

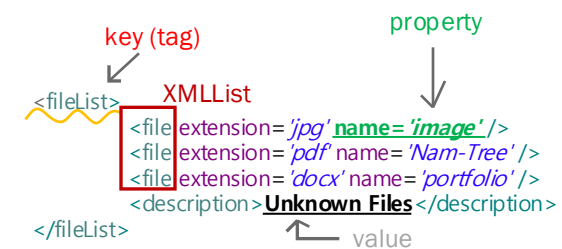
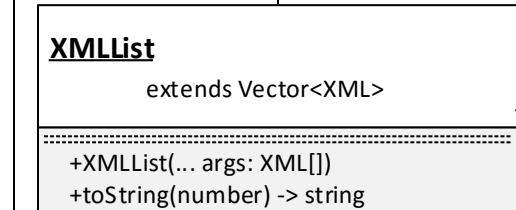
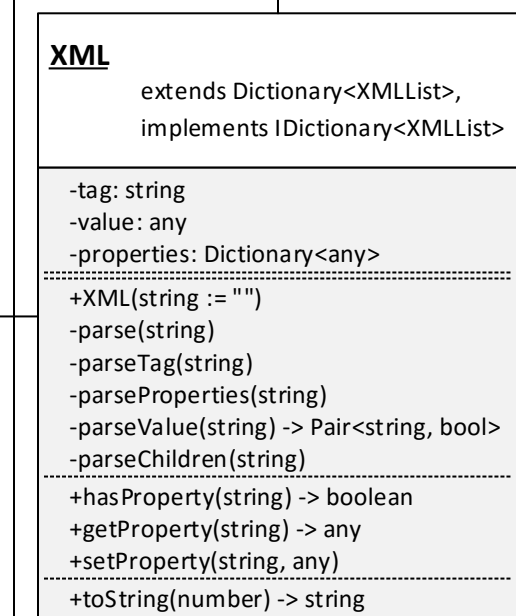
+virtual toSprite() -> Sprite

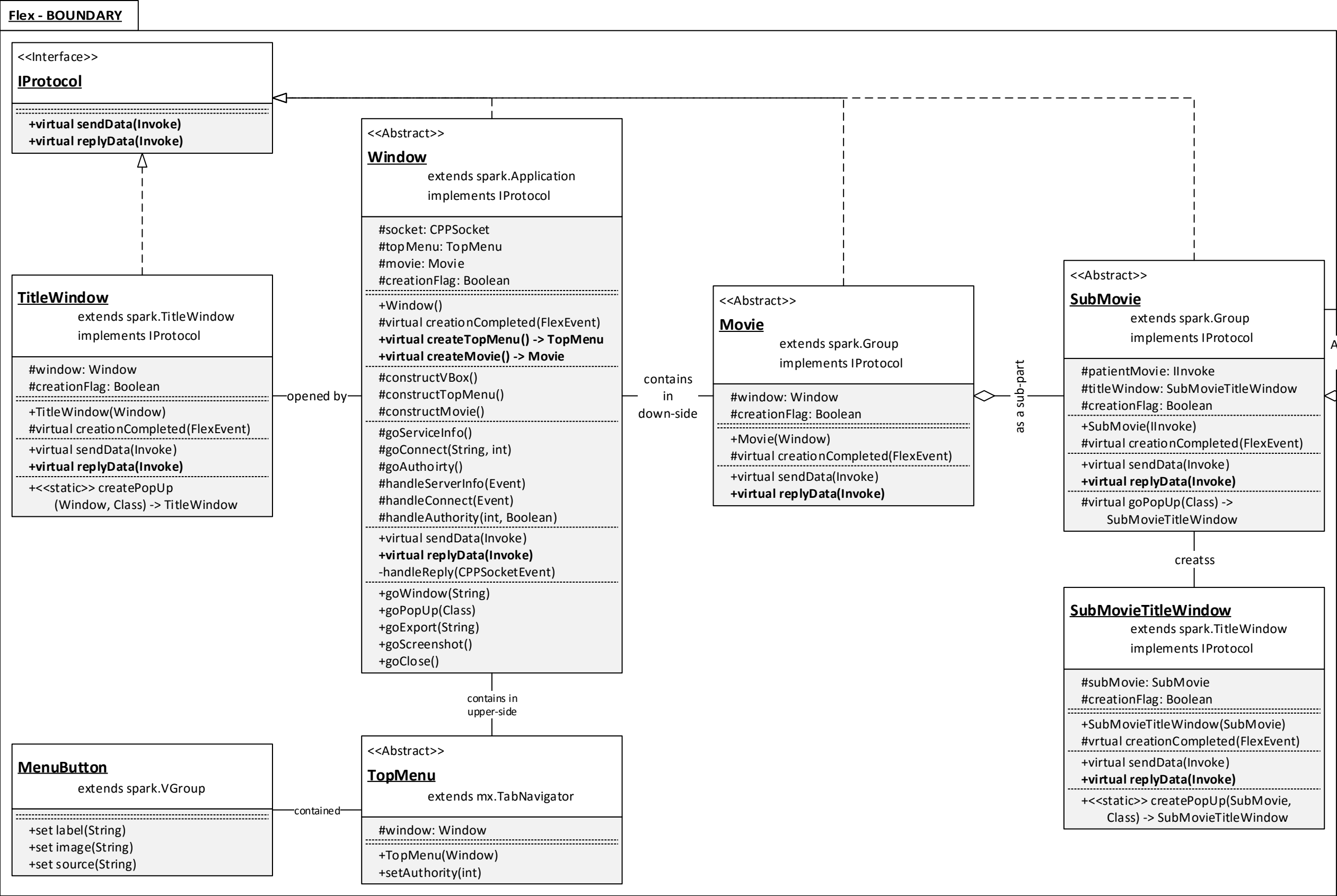
TypeScript's library

Like STL

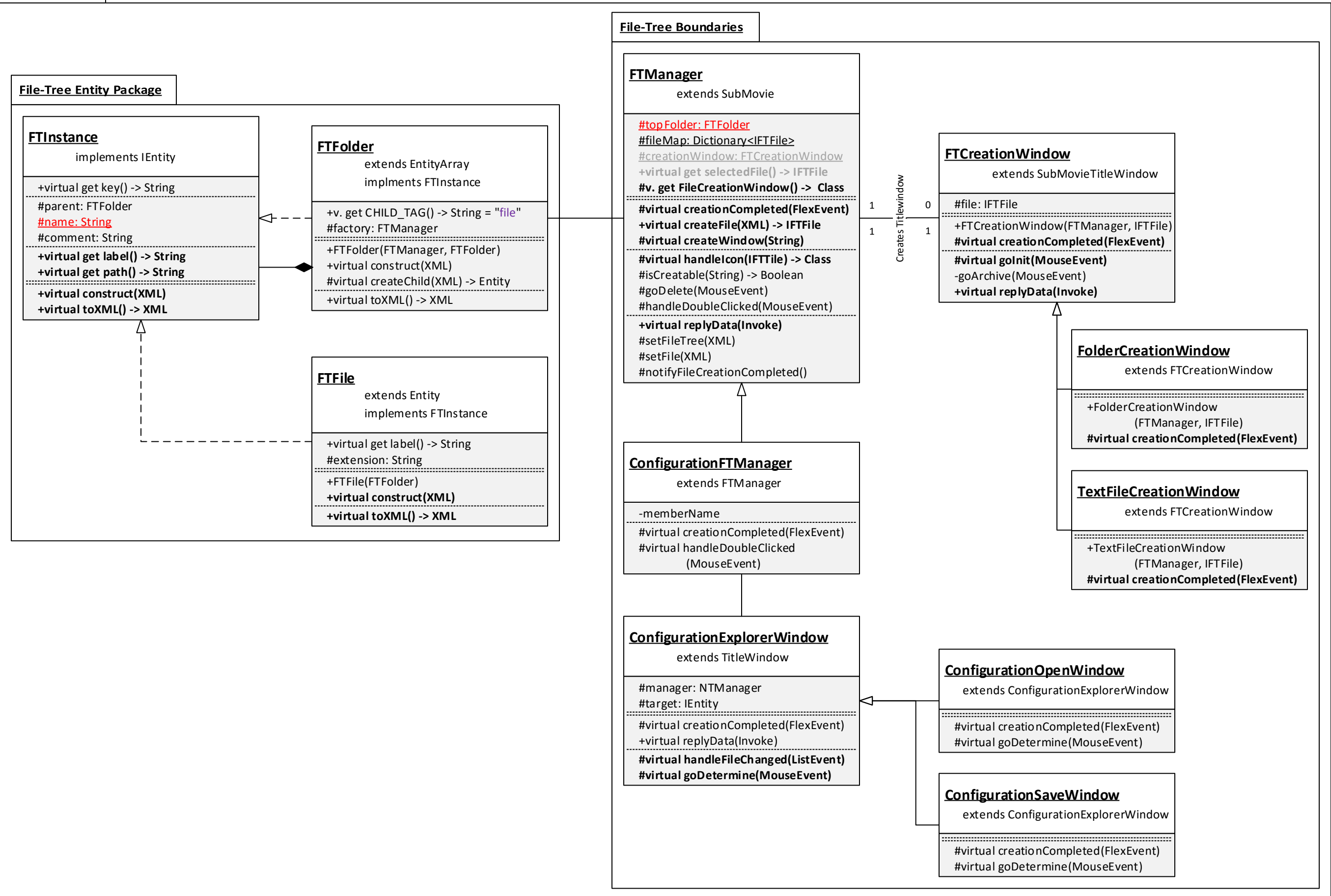


XML Package

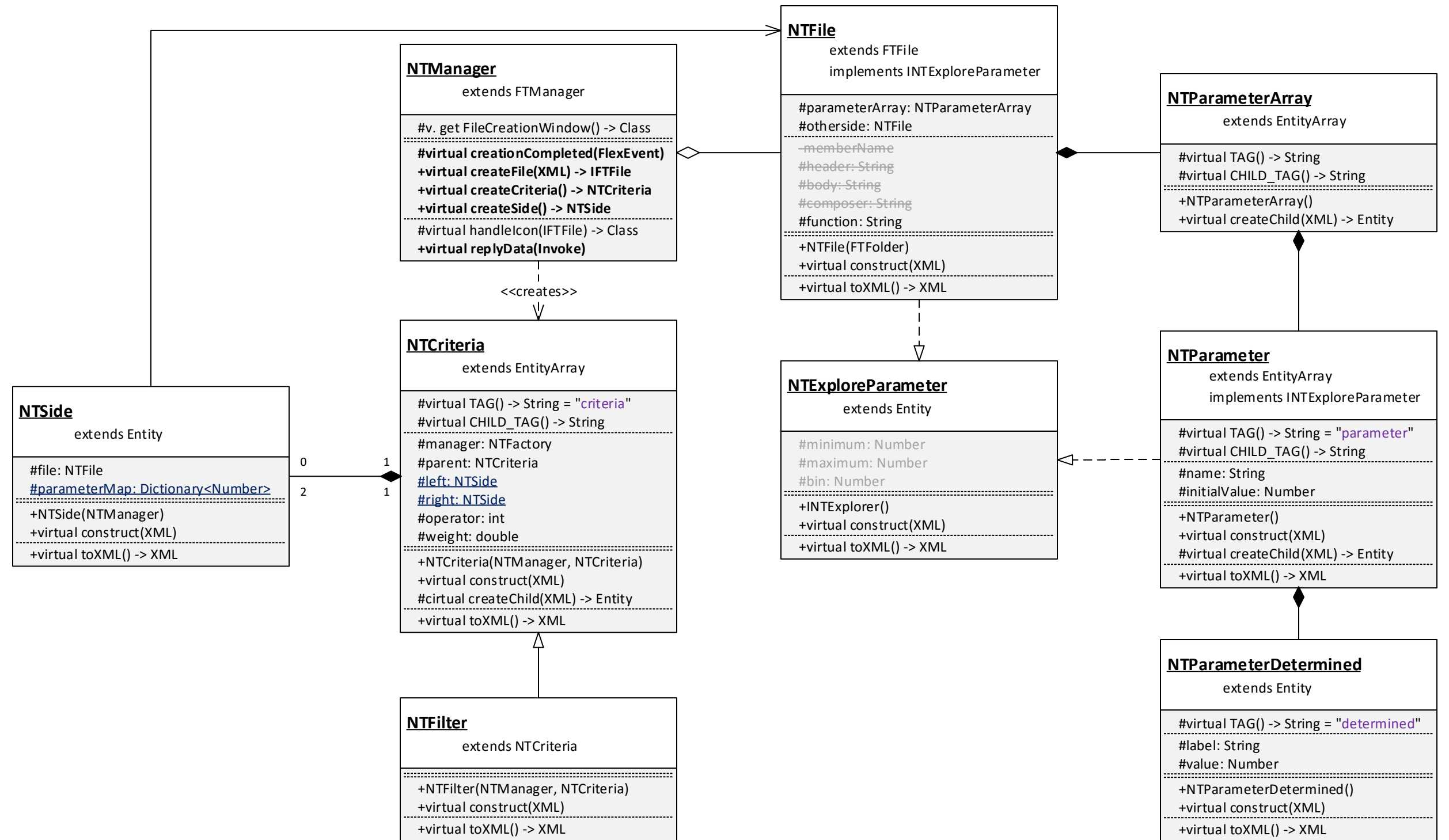




## Flex - FileTree



### Flex NamTree Entity



## Flex - NamTree UI

### NTManager

extends FTManager

```
#v. get FileCreationWindow() -> Class
#virtual creationCompleted(FlexEvent)
+virtual createFile(XML) -> IFTFile
+virtual createCriteria() -> NTCriteria
+virtual createSide() -> NTSide
#virtual handleIcon(IFTFile) -> Class
+virtual replyData(Invoke)
```

creates window

### NTCreationWindow

extends SubMovieTitleWindow

```
#get parameters() -> NTParameterArray
#get determined() -> NTParameter
+NTCreationWindow(FTManager, IFile)
#virtual creationCompleted(FlexEvent)
#virtual goInit(MouseEvent)
+virtual replyData(Invoke)
```

### NTFileParameterGrid

extends mx.HGroup

```
#window: NTCreationWindow
#parameterGrid: mx.AdvancedDataGrid
#determinedGrid: mx.AdvancedDataGrid
#creationCompleted(FlexEvent)
-add(MouseEvent)
-addDetermined(MouseEvent)
-remove(MouseEvent)
-removeDetermined(MouseEvent)
```

contains

be contained in right side

### NTMovie

extends SubMovie

```
#v. get criteriaGrid() -> NTCriteriaGrid
#manager: NTManager
#criteriaGrids: Vector<NTCriteriaGrid>
#leftSideContainer: NTSideContainer
#rightSideContainer: NTSideContainer
#virtual creationCompleted(FlexEvent)
#virtual constructUI()
#virtual createManager() -> NTManager
#virtual createCriteriaGrid()
-> NTCriteriaGrid
#virtual createSideContainer()
-> NTSideContainer
-goApplyFile(enum := DIRECTION)
-itemGridChanged(ListEvent)
```

contains in bottom side

### NTSideContainer

extends mx.VGroup

```
#direction: enum {LEFT, RIGHT}
#parameterGrid: mx.AdvancedDataGrid
#side: NTSide
#virtual creationCompleted(FlexEvent)
+refresh()
+setSide(NTSide)
-valueChanged(Event)
```

### NTCriteriaGrid

extends mx.AdvancedDataGrid

```
#manager: NTManager
#topCriteria: NTCriteria
#changeHandler: Function
+get selectedCriteria() -> NTCriteria
#virtual creationCompleted(FlexEvent)
-initialize(MouseEvent)
-goImport(MouseEvent, NTCriteria)
-goExport(MouseEvent, NTCriteria)
-addFilter(MouseEvent)
-addCriteria(MouseEvent)
-removeCriteria(MouseEvent)
```

Sometimes can be multiple

