

MagicDrawCompPlugin

AUTHOR
Version 4.3
Tue Aug 9 2016

Table of Contents

Table of contents

MagicDraw Component Plugin

Introduction

This is the documentation for the Magic Draw UML to XML generator plugin.

Currently, the program can create XML files from component models and subsystem models. Subsystems within subsystems along side multiple instances of subsystems and models are supported. Additionally, exceptions are thrown if there is something the program cannot handle and error messages are displayed onto the console.

Notes

- If there are multiple subsystems in a model, make sure to create a stereotype "Top" and assign it to the "Top" level subsystem. This will ensure proper naming of the XML files.
 - A simple example of how to implement multiple subsystems and multiple instances of subsystems in a model can be found at tests/testMultipleSubsystems.mdzip.
- Please make sure to give instances of subsystems unique names.
- Velocity (an Apache template engine) is being used to generate the XML files. These files can be found in the root of the MagicDrawPlugin folder and have .vm extensions.

Using The Plugin

- Load a project into MagicDraw.
- Ensure that the project either only has zero or one subsystems, or has the "Top" stereotype applied to one of the subsystems if many exist.
- Click on the "Component Autocoder" tab in the top toolbar.
- Click on "Auto Generate Component/Port/Topology XML".
- Check the MagicDraw console to see if any errors were generated.
- If not, the XML files can be found in <projectDir>/AutoXML/

Program Flow

Menu Configuration

These files take care of the menu configuration:

MainMenuConfigurator.java - Creates "Component Autocoder" tab in the toolbar.

IsfXMLAutocoderAction.java - Creates buttons within the "Component Autocoder" tab.

IsfAbout.java - Creates the modal element that displays the about information.

IsfXmlAction.java - Action object that is called when "Auto Generate Component/Port/Topology XML" is clicked. Processes ports, components, and the entire topology.

IsfCommandLine.java - Used to handle command line protocol.

IsfTopAction.java - NOT USED.

IsfComponentAction.java - NOT USED.

LoadIDConfigAction.java - Creates button to load id configuration file within the "Component Autocoder" tab.

Processing

ProcessISFProject.java - Object processes both components and ports at the same time.

Component

ISFComponent.java - Helper object for component processing.

IsfCompXmlWriter.java - Writes the processed components to XML documents.

Port

ISFPort.java Helper object for port processing.

IsfPortXmlWriter.java - Writes the processed ports to XML documents.

Topology

ProcessISFTopology.java - This object processes the model to create the topology XML diagrams. It can handle multiple subsystem within subsystems as well and multiple instantiations of subsystems and components.

ISFSubsystem.java - Contains many helper functions and objects used when process the model for the topology diagram.

IsfSubXmlWriter.java - Writes the processed model to an XML document.

Exceptions

These are the Exception objects of this plugin:

ComponentException.java - Thrown during component processing.

PortException.java - Thrown during port processing.

ConnectorException.java - Thrown during the topology processing

LoadIDEException.java - Thrown during LoadIDConfig processing.

Other

Utils.java - Includes simple functions to help reduce code size (IE throwing ConnectorExceptions or raising warnings). **LoadIDConfig.java** - Loads id configuration file into the model and over writes component base and window ids with ones found in the file.

About the Documentation

More resources can be found in the base MagicDrawCompPlugin/doc/ or at MagicDrawCompPlugin/README.txt

To update the documentation to reflect code changes:

- This Introduction area can be edited in the top comment block in **MagicDrawCompPlugin/src/gov/nasa/jpl/componentaction/MainMenuConfigurator.java**.
- Make sure Doxygen and Graphviz are both installed.

- Open the Doxygen Gui.
- Under the "File" tab at the top, click "Open" and select the file titled Doxyfile in the MagicDrawCompPlugin directory.
- Modify the settings (IE version number) in the GUI.
- To regenerate the documentation files, click on the "Run" tab in the GUI and click "Run doxygen".
- To update the PDF found in the doc folder, go to doc/rtf and convert refman.rtf to refman.doc using Microsoft Word, and then convert the .doc file to .pdf.

Contact

Until the plugin is updated even further, I will be able to answer questions any time. My name is Saikiran Ramanan and I worked on the plugin as an intern during the Summer of 2016. My email address is saiquiranra@gmail.com

Namespace Index

Packages

Here are the packages with brief descriptions (if available):

gov	10
gov.nasa	11
gov.nasa.jpl	12
gov.nasa.jpl.componentaction	13
gov.nasa.jpl.componentaction.isfxmlwriter	14

Hierarchical Index

Class Hierarchy

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

gov.nasa.jpl.componentaction.ISFSubsystem.componentType	17
gov.nasa.jpl.componentaction.LoadIDConfig.configItem	19
Exception	
gov.nasa.jpl.componentaction.ComponentException	15
gov.nasa.jpl.componentaction.ConnectorException	21
gov.nasa.jpl.componentaction.LoadIDEException.....	108
gov.nasa.jpl.componentaction.PortException	123
gov.nasa.jpl.componentaction.ISFComponent	27
gov.nasa.jpl.componentaction.isfxmlwriter.IsfCompXmlWriter	47
gov.nasa.jpl.componentaction.ISFPort	49
gov.nasa.jpl.componentaction.isfxmlwriter.IsfPortXmlWriter	57
gov.nasa.jpl.componentaction.ISFSubsystem	59
gov.nasa.jpl.componentaction.isfxmlwriter.IsfSubXmlWriter.....	88
gov.nasa.jpl.componentaction.LoadIDConfig	99
gov.nasa.jpl.componentaction.Utils.Pair< A, B >	114
gov.nasa.jpl.componentaction.Partition.....	118
gov.nasa.jpl.componentaction.ISFSubsystem.physicalConnectionType	120
gov.nasa.jpl.componentaction.ISFComponent.portObject	125
gov.nasa.jpl.componentaction.ProcessISFProject	128
gov.nasa.jpl.componentaction.ProcessISFTopology	137
gov.nasa.jpl.componentaction.ISFComponent.referencePort	161
gov.nasa.jpl.componentaction.ISFSubsystem.topologyModel.....	186
gov.nasa.jpl.componentaction.Utils.....	188
AMConfigurator	
gov.nasa.jpl.componentaction.MainMenuConfigurator	110
CommandLine	
gov.nasa.jpl.componentaction.IsfCommandLine	25
MagicDrawTestCase	
gov.nasa.jpl.componentaction.TestISFEceptions	162
MDAction	
gov.nasa.jpl.componentaction.IsfAbout	23
gov.nasa.jpl.componentaction.IsfComponentAction	45
gov.nasa.jpl.componentaction.IsfTopAction.....	91
gov.nasa.jpl.componentaction.IsfXmlAction	95
gov.nasa.jpl.componentaction.LoadIDConfigAction.....	105

Plugin	
gov.nasa.jpl.componentaction.IsfXmlAutocoderAction	97
PrintStream	
gov.nasa.jpl.componentaction.MpmcsErrorStream	112

Class Index

Class List

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

gov.nasa.jpl.componentaction.ComponentException	15
gov.nasa.jpl.componentaction.ISFSubsystem.componentType	17
gov.nasa.jpl.componentaction.LoadIDConfig.configItem	19
gov.nasa.jpl.componentaction.ConnectorException	21
gov.nasa.jpl.componentaction.IsfAbout	23
gov.nasa.jpl.componentaction.IsfCommandLine	25
gov.nasa.jpl.componentaction.ISFComponent	27
gov.nasa.jpl.componentaction.IsfComponentAction	45
gov.nasa.jpl.componentaction.isfxmlwriter.IsfCompXmlWriter	47
gov.nasa.jpl.componentaction.ISFPort	49
gov.nasa.jpl.componentaction.isfxmlwriter.IsfPortXmlWriter	57
gov.nasa.jpl.componentaction.ISFSubsystem	59
gov.nasa.jpl.componentaction.isfxmlwriter.IsfSubXmlWriter	88
gov.nasa.jpl.componentaction.IsfTopAction	91
gov.nasa.jpl.componentaction.IsfXmlAction	95
gov.nasa.jpl.componentaction.IsfXmlAutocoderAction	97
gov.nasa.jpl.componentaction.LoadIDConfig	99
gov.nasa.jpl.componentaction.LoadIDConfigAction	105
gov.nasa.jpl.componentaction.LoadIDException	108
gov.nasa.jpl.componentaction.MainMenuConfigurator	110
gov.nasa.jpl.componentaction.MpmcsErrorStream	112
gov.nasa.jpl.componentaction.Utils.Pair< A, B >	114
gov.nasa.jpl.componentaction.Partition	118
gov.nasa.jpl.componentaction.ISFSubsystem.physicalConnectionType	120
gov.nasa.jpl.componentaction.PortException	123
gov.nasa.jpl.componentaction.ISFComponent.portObject	125
gov.nasa.jpl.componentaction.ProcessISFProject	128
gov.nasa.jpl.componentaction.ProcessISFTopology	137
gov.nasa.jpl.componentaction.ISFComponent.referencePort	161
gov.nasa.jpl.componentaction.TestISFEExceptions	162
gov.nasa.jpl.componentaction.ISFSubsystem.topologyModel	186
gov.nasa.jpl.componentaction.Utils	188

File Index

File List

Here is a list of all files with brief descriptions:

src/gov/nasa/jpl/componentaction/ComponentException.java	196
src/gov/nasa/jpl/componentaction/ConnectorException.java	197
src/gov/nasa/jpl/componentaction/IsfAbout.java	198
src/gov/nasa/jpl/componentaction/IsfCommandLine.java	199
src/gov/nasa/jpl/componentaction/ISFComponent.java	200
src/gov/nasa/jpl/componentaction/IsfComponentAction.java	201
src/gov/nasa/jpl/componentaction/ISFPort.java	202
src/gov/nasa/jpl/componentaction/ISFSubsystem.java	203
src/gov/nasa/jpl/componentaction/IsfTopAction.java	204
src/gov/nasa/jpl/componentaction/IsfXmlAction.java	205
src/gov/nasa/jpl/componentaction/IsfXmlAutocoderAction.java	206
src/gov/nasa/jpl/componentaction/LoadIDConfig.java	210
src/gov/nasa/jpl/componentaction/LoadIDConfigAction.java	211
src/gov/nasa/jpl/componentaction/LoadIDEException.java	212
src/gov/nasa/jpl/componentaction/MainMenuConfigurator.java	213
src/gov/nasa/jpl/componentaction/MpmcsErrorStream.java	214
src/gov/nasa/jpl/componentaction/Partition.java	215
src/gov/nasa/jpl/componentaction/PortException.java	216
src/gov/nasa/jpl/componentaction/ProcessISFProject.java	217
src/gov/nasa/jpl/componentaction/ProcessISFTopology.java	218
src/gov/nasa/jpl/componentaction/TestISFEExceptions.java	219
src/gov/nasa/jpl/componentaction/Utils.java	220
src/gov/nasa/jpl/componentaction/isfxmlwriter/IsfCompXmlWriter.java	207
src/gov/nasa/jpl/componentaction/isfxmlwriter/IsfPortXmlWriter.java	208
src/gov/nasa/jpl/componentaction/isfxmlwriter/IsfSubXmlWriter.java	209

Namespace Documentation

Package gov

Packages

- package nasa

Package gov.nasa

Packages

- package **jpl**

Package gov.nasa.jpl

Packages

- package **componentaction**

Package gov.nasa.jpl.componentaction

Packages

- package **isfxmlwriter**

Classes

- class **ComponentException**
- class **ConnectorException**
- class **IsfAbout**
- class **IsfCommandLine**
- class **ISFComponent**
- class **IsfComponentAction**
- class **ISFPort**
- class **ISFSubsystem**
- class **IsfTopAction**
- class **IsfXmlAction**
- class **IsfXmlAutocoderAction**
- class **LoadIDConfig**
- class **LoadIDConfigAction**
- class **LoadIDEception**
- class **MainMenuConfigurator**
- class **MpmcsErrorStream**
- class **Partition**
- class **PortException**
- class **ProcessISFProject**
- class **ProcessISFTopology**
- class **TestISFExceptions**
- class **Utils**

Detailed Description

Id

ActionTypesExample.java 52410 2007-10-05 06:51:43Z donsim
Copyright (c) 2002 NoMagic, Inc. All Rights Reserved.

Package gov.nasa.jpl.componentaction.isfxmlwriter

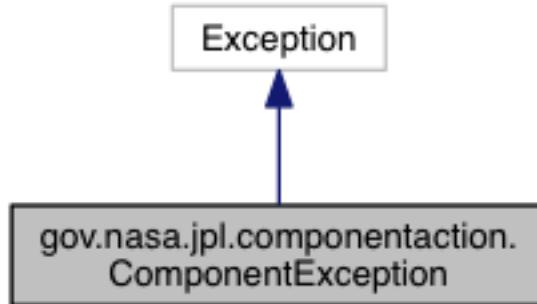
Classes

- class **IsfCompXmlWriter**
- class **IsfPortXmlWriter**
- class **IsfSubXmlWriter**

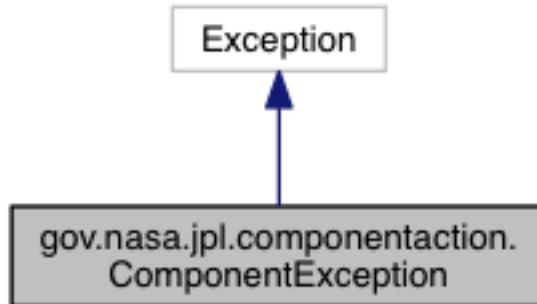
Class Documentation

gov.nasa.jpl.componentaction.ComponentException Class Reference

Inheritance diagram for gov.nasa.jpl.componentaction.ComponentException:



Collaboration diagram for gov.nasa.jpl.componentaction.ComponentException:



Public Member Functions

- `ComponentException (String message)`

Static Private Attributes

- `static final long serialVersionUID = 1L`

Detailed Description

Component Exception shell class. This just serves as a wrapper that can be used to differentiate between where an error happens.

Constructor & Destructor Documentation

`gov.nasa.jpl.componentaction.ComponentException.ComponentException (String message)`

Member Data Documentation

```
final long gov.nasa.jpl.componentaction.ComponentException.serialVersionUID = 1L [static],  
[private]
```

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

- [src/gov/nasa/jpl/componentaction/ComponentException.java](#)

gov.nasa.jpl.componentaction.ISFS subsystem.componentType Class Reference

Public Member Functions

- `componentType ()`
- `String getName ()`
- `String getNameSpace ()`
- `String getType ()`
- `String getBaseID ()`
- `String getInstanceWindow ()`
- `String getXMLLocation ()`

Package Attributes

- `String name`
 - `String nameSpace`
 - `String type`
 - `String baseID`
 - `String instanceWindow`
 - `String XMLLocation`
-

Detailed Description

Used as a dataObjet within the componentMap. The objects within this object are used to describe a leaf component. The methods are pretty self-explanatory.

Constructor & Destructor Documentation

`gov.nasa.jpl.componentaction.ISFS subsystem.componentType.componentType ()`

Member Function Documentation

`String gov.nasa.jpl.componentaction.ISFS subsystem.componentType.getBaseID ()`

`String gov.nasa.jpl.componentaction.ISFS subsystem.componentType.getInstanceWindow ()`

`String gov.nasa.jpl.componentaction.ISFS subsystem.componentType.getName ()`

`String gov.nasa.jpl.componentaction.ISFS subsystem.componentType.getNameSpace ()`

`String gov.nasa.jpl.componentaction.ISFS subsystem.componentType.getType ()`

`String gov.nasa.jpl.componentaction.ISFS subsystem.componentType.getXMLLocation ()`

Member Data Documentation

String gov.nasa.jpl.componentaction.ISFSubsystem.componentType.baseID [package]

String gov.nasa.jpl.componentaction.ISFSubsystem.componentType.instanceWindow [package]

String gov.nasa.jpl.componentaction.ISFSubsystem.componentType.name [package]

String gov.nasa.jpl.componentaction.ISFSubsystem.componentType.nameSpace [package]

String gov.nasa.jpl.componentaction.ISFSubsystem.componentType.type [package]

String gov.nasa.jpl.componentaction.ISFSubsystem.componentType.XMLLocation [package]

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

- src/gov/nasa/jpl/componentaction/ISFSubsystem.java

gov.nasa.jpl.componentaction.LoadIDConfig.configItem Class Reference

Public Member Functions

- **configItem** (String compType, String instName, int bID, int wRange)
- int **hashCode** ()
- boolean **equals** (Object obj)
- String **toString** ()

Package Attributes

- String **componentType**
 - String **instanceName**
 - int **baseID**
 - int **windowRange**
-

Detailed Description

Used to store information for each configuration item from the config file.

Constructor & Destructor Documentation

gov.nasa.jpl.componentaction.LoadIDConfig.configItem (String *compType*, String *instName*, int *bID*, int *wRange*)

Here is the caller graph for this function:

Member Function Documentation

boolean gov.nasa.jpl.componentaction.LoadIDConfig.configItem.equals (Object obj)

Here is the call graph for this function:

int gov.nasa.jpl.componentaction.LoadIDConfig.configItem.hashCode ()

String gov.nasa.jpl.componentaction.LoadIDConfig.configItem.toString ()

Member Data Documentation

int gov.nasa.jpl.componentaction.LoadIDConfig.configItem.baseID [package]

String gov.nasa.jpl.componentaction.LoadIDConfig.configItem.componentType [package]

String gov.nasa.jpl.componentaction.LoadIDConfig.configItem.instanceName [package]

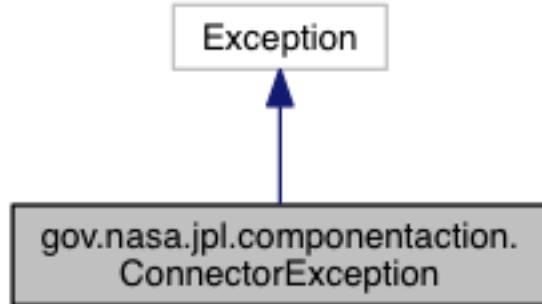
int gov.nasa.jpl.componentaction.LoadIDConfig.configItem.windowRange [package]

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

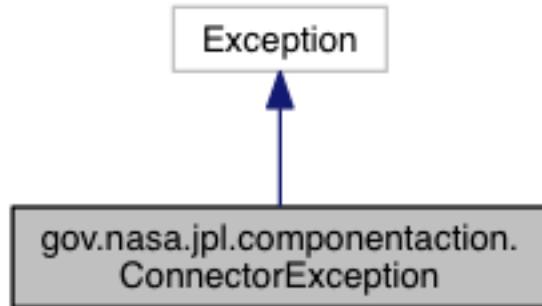
- `src/gov/nasa/jpl/componentaction/LoadIDConfig.java`

gov.nasa.jpl.componentaction.ConnectorException Class Reference

Inheritance diagram for gov.nasa.jpl.componentaction.ConnectorException:



Collaboration diagram for gov.nasa.jpl.componentaction.ConnectorException:



Public Member Functions

- **ConnectorException** (String message)

Static Private Attributes

- static final long serialVersionUID = 1L

Detailed Description

Connector Exception shell class. This just serves as a wrapper that can be used to differentiate between where an error happens.

Constructor & Destructor Documentation

gov.nasa.jpl.componentaction.ConnectorException.ConnectorException (String message)

Member Data Documentation

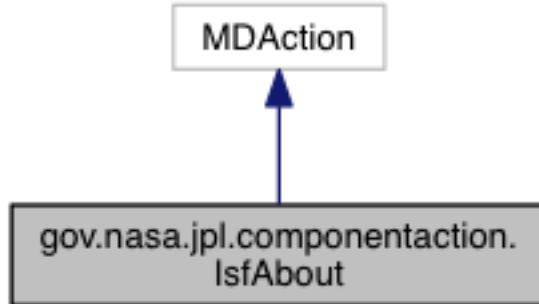
```
final long gov.nasa.jpl.componentaction.ConnectorException.serialVersionUID = 1L [static],  
[private]
```

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

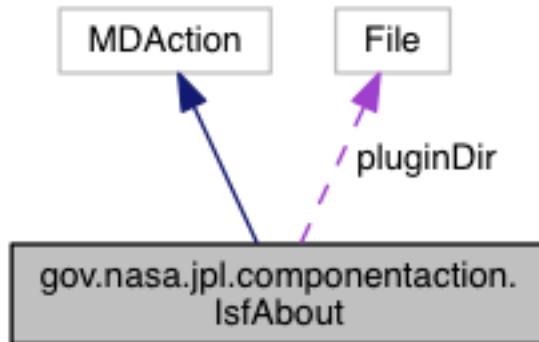
- [src/gov/nasa/jpl/componentaction/ConnectorException.java](#)

gov.nasa.jpl.componentaction.IsfAbout Class Reference

Inheritance diagram for gov.nasa.jpl.componentaction.IsfAbout:



Collaboration diagram for gov.nasa.jpl.componentaction.IsfAbout:



Public Member Functions

- **IsfAbout** (String id, String name, File **pluginDir**)
- void **actionPerformed** (ActionEvent e)

Package Attributes

- File **pluginDir**

Static Private Attributes

- static final long **serialVersionUID** = -6790954285526957354L

Detailed Description

Action class for information regarding the plugin

Constructor & Destructor Documentation

gov.nasa.jpl.componentaction.IsfAbout.IsfAbout (String *id*, String *name*, File *pluginDir*)

Member Function Documentation

void gov.nasa.jpl.componentaction.IsfAbout.actionPerformed (ActionEvent e)

Used to display content in the "About" tab. Edit this for tab changes.

Here is the caller graph for this function:



Member Data Documentation

File **gov.nasa.jpl.componentaction.IsfAbout.pluginDir** [package]

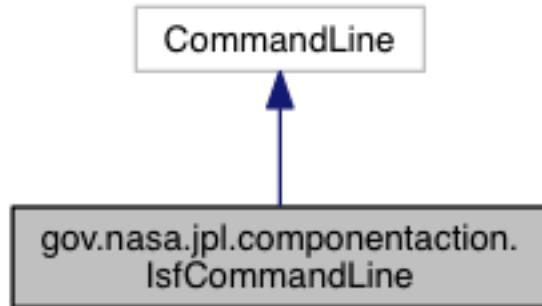
final long gov.nasa.jpl.componentaction.IsfAbout.serialVersionUID = -6790954285526957354L [static], [private]

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

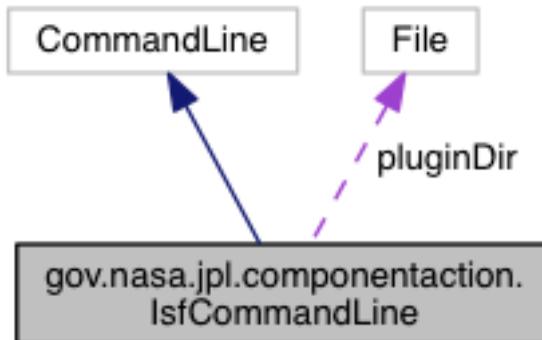
- [src/gov/nasa/jpl/componentaction/IsfAbout.java](#)

gov.nasa.jpl.componentaction.IsfCommandLine Class Reference

Inheritance diagram for gov.nasa.jpl.componentaction.IsfCommandLine:



Collaboration diagram for gov.nasa.jpl.componentaction.IsfCommandLine:



Static Public Member Functions

- static void **main** (String[] args)

Protected Member Functions

- byte **execute** ()

Static Package Attributes

- static File **pluginDir**
- static String **projectFile**

Static Private Attributes

- static final String **PROJECT** = "project_name"
- static final String **PACKAGE** = "package"
- static final String **USAGE** = "isfpluginexec [options]\nBefore running on headless environment, use Xvfb to set up a virtual X11 display server\n/usr/bin/Xvfb :1 -screen 0 1024x768x24\nexport DISPLAY=:1"
- static final String **COPYRIGHT** = "Copyright 2015 (California Institute of Technology)\nALL RIGHTS RESERVED. U.S. Government Sponsorship acknowledged."

Detailed Description

Class to run MagicDraw plugin from command line

```
Before running on headless environment, use Xvfb to set up a virtual X11 display server  
/usr/bin/Xvfb :1 -screen 0 1024x768x24 export DISPLAY=:1
```

To run the utility, use the script isfpluginexec.sh in the MagicDrawComdPlugin directory

Member Function Documentation

byte gov.nasa.jpl.componentaction.IsfCommandLine.execute () [protected]

Runs autocoder on the project specified

Here is the call graph for this function:

static void gov.nasa.jpl.componentaction.IsfCommandLine.main (String[] args) [static]

Reads the command line arguments and launches the program

Parameters:

<i>args</i>	application arguments.
-------------	------------------------

Member Data Documentation

**final String gov.nasa.jpl.componentaction.IsfCommandLine.COPYRIGHT = "Copyright 2015
(California Institute of Technology)\nALL RIGHTS RESERVED. U.S. Government Sponsorship
acknowledged." [static], [private]**

**final String gov.nasa.jpl.componentaction.IsfCommandLine.PACKAGE = "package" [static],
[private]**

File gov.nasa.jpl.componentaction.IsfCommandLine.pluginDir [static], [package]

**final String gov.nasa.jpl.componentaction.IsfCommandLine.PROJECT =
"project_name" [static], [private]**

String gov.nasa.jpl.componentaction.IsfCommandLine.projectFile [static], [package]

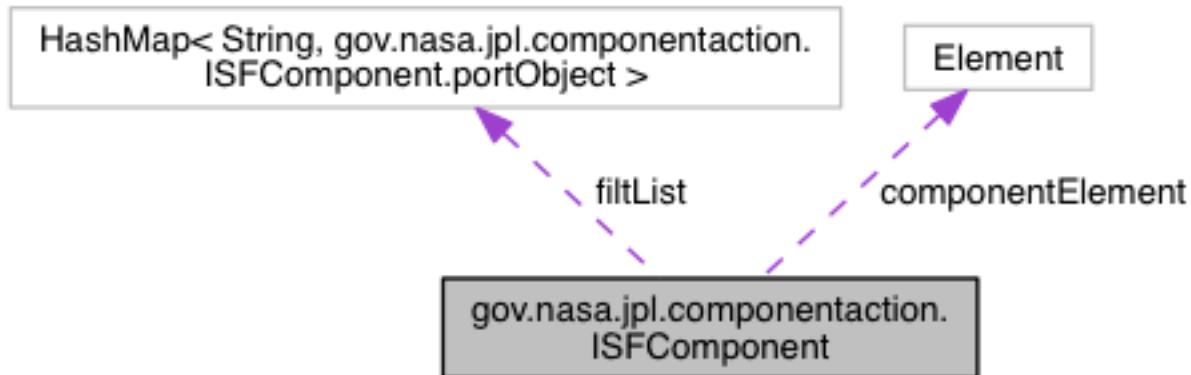
**final String gov.nasa.jpl.componentaction.IsfCommandLine.USAGE = "isfpluginexec
[options]\nBefore running on headless environment, use Xvfb to set up a virtual X11 display
server\n/usr/bin/Xvfb :1 -screen 0 1024x768x24\nexport DISPLAY=:1" [static], [private]**

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

- src/gov/nasa/jpl/componentaction/IsfCommandLine.java

gov.nasa.jpl.componentaction.ISFComponent Class Reference

Collaboration diagram for gov.nasa.jpl.componentaction.ISFComponent:



Classes

- class **portObject**
- class **referencePort**

Public Member Functions

- **ISFComponent (Element componentElement)**
- **void setStereotype (String compStereotype)**
- **void newAddPort (Element portElement) throws PortException**
- **void newRolePort (Element portElement) throws PortException**
- **String getPortName (Element portElement)**
- **String getPortStereotype (Element portElement)**
- **String getPortDataType (Element portElement) throws PortException**
- **String getPortDataTypeNamespace (Element portElement)**
- **HashMap< String, portObject > getPortHashList ()**
- **String getURI ()**
- **String getIncludeCmdXML ()**
- **String getIncludeEvrXML ()**
- **String getIncludeTlmXML ()**
- **String getIncludeParamXML ()**
- **String getIncludeInternalIFXML ()**
- **String getIncludeHdrFile ()**
- **String getNamespace ()**
- **String getName ()**
- **String getStereotype ()**
- **List< String > getPortDataList ()**
- **String getPortTypeProperty (Element portElement, String property)**
- **String getComponentProperty (Element portElement, String property)**
- **String getPortProperty (Element portElement, String property)**
- **void print ()**
- **void checkComponent () throws ComponentException**
- **void warnLog (String errStr)**
- **boolean isValid (String stereotype, String portType, boolean atLeastOne)**

Package Attributes

- `HashMap< String, portObject > filtList`
- `String IncludeCmdXML`
- `String IncludeTlmXML`
- `String IncludeEvrXML`
- `String IncludeParamXML`
- `String IncludeInternalIFXML`
- `String IncludeIncludeHdr`

Private Member Functions

- `String getDictionaryString (String includeString)`

Private Attributes

- `Element componentElement`
- `String compStereotype`
- `String URIName`

Detailed Description

This class encapsulates information pertaining to the component types and their ports in the MD model - ie active, passive or queued components

Constructor & Destructor Documentation

`gov.nasa.jpl.componentaction.ISFComponent.ISFComponent (Element componentElement)`

Instantiation for the **ISFComponent**.

Parameters:

<code>componentElement</code>	The element of the component that will be processed.
-------------------------------	--

Here is the call graph for this function:



Member Function Documentation

`void gov.nasa.jpl.componentaction.ISFComponent.checkComponent () throws ComponentException`

All passive components cannot have `async_input` ports All active components should have `sync_input` ports All queued components should have `sync` and `async` input ports

Exceptions:

<code>ComponentException</code>	
---------------------------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



String gov.nasa.jpl.componentaction.ISFComponent.getComponentProperty (Element portElement, String property)

Returns the property value of Component stereotype of the input element argument.

Parameters:

<i>portElement</i>	Element of the port type
<i>property</i>	String of value to be looked for in the stereotype Component

Returns:

The value of the attribute from property

Here is the caller graph for this function:



```
String gov.nasa.jpl.componentaction.ISFComponent.getDictionaryString (String  
includeString) [private]
```

Return the full dictionary path which includes the URI path specified in the package above the component.

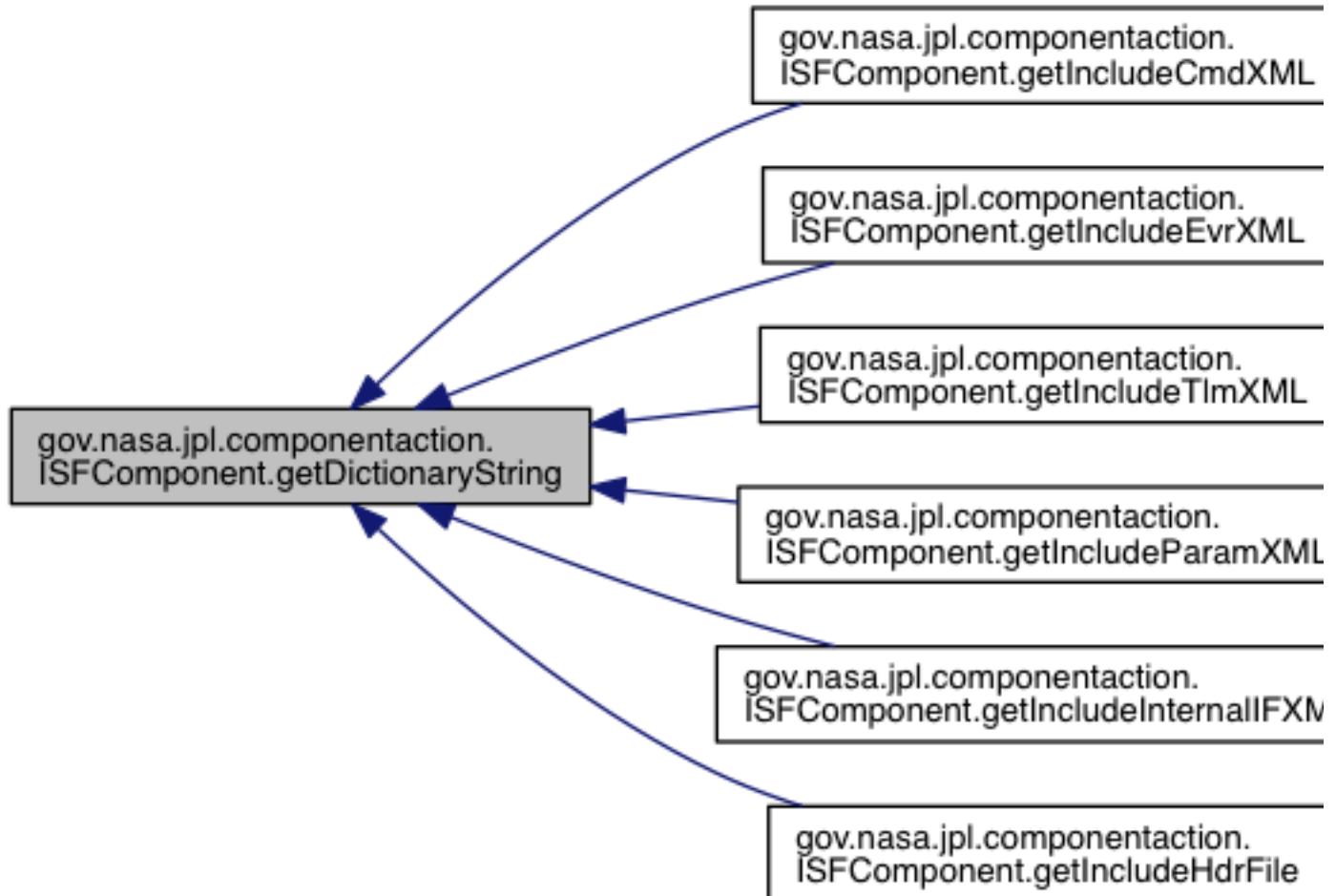
Parameters:

includeString	
---------------	--

Returns:

input string prepended by URIName

Here is the caller graph for this function:



```
String gov.nasa.jpl.componentaction.ISFComponent.getIncludeCmdXML ()
```

Prepends the IncludeCmdXML String with the URI Name

Returns:

address string

Here is the call graph for this function:



String gov.nasa.jpl.componentaction.ISFComponent.getIncludeEvrXML ()

Prepends the IncludeEvrXML String with the URI Name

Returns:

address string

Here is the call graph for this function:



String gov.nasa.jpl.componentaction.ISFComponent.getIncludeHdrFile ()

Prepends the IncludeIncludeHdr String with the URI Name

Returns:

address string

Here is the call graph for this function:



String gov.nasa.jpl.componentaction.ISFComponent.getIncludeInternalIFXML ()

Prepends the IncludeInternalIFXML String with the URI Name

Returns:

address string

Here is the call graph for this function:



String gov.nasa.jpl.componentaction.ISFComponent.getIncludeParamXML ()

Prepends the IncludeParamXML String with the URI Name

Returns:

address string

Here is the call graph for this function:



String gov.nasa.jpl.componentaction.ISFComponent.getIncludeTlmXML ()

Prepends the IncludeTlmXML String with the URI Name

Returns:

address string

Here is the call graph for this function:



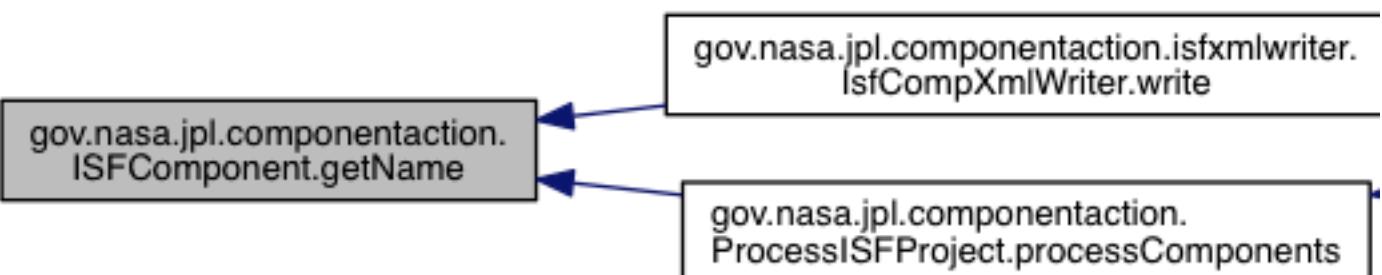
String gov.nasa.jpl.componentaction.ISFComponent.getName ()

Returns last part of the componentElement name

Returns:

name

Here is the caller graph for this function:



String gov.nasa.jpl.componentaction.ISFComponent.getNamespace ()

Get the component stereotype attribute called "Namespace"

Returns:

NameSpace string of the componentElement

Here is the call graph for this function:



List<String> gov.nasa.jpl.componentaction.ISFComponent.getPortDataList ()

Creates a list of strings based off the data type from filitList values

Returns:

A list of data type strings

String gov.nasa.jpl.componentaction.ISFComponent.getPortDataType (Element portElement) throws PortException

This returns the full path of the data type which includes the directory tree.

Parameters:

<code>portElement</code>	
--------------------------	--

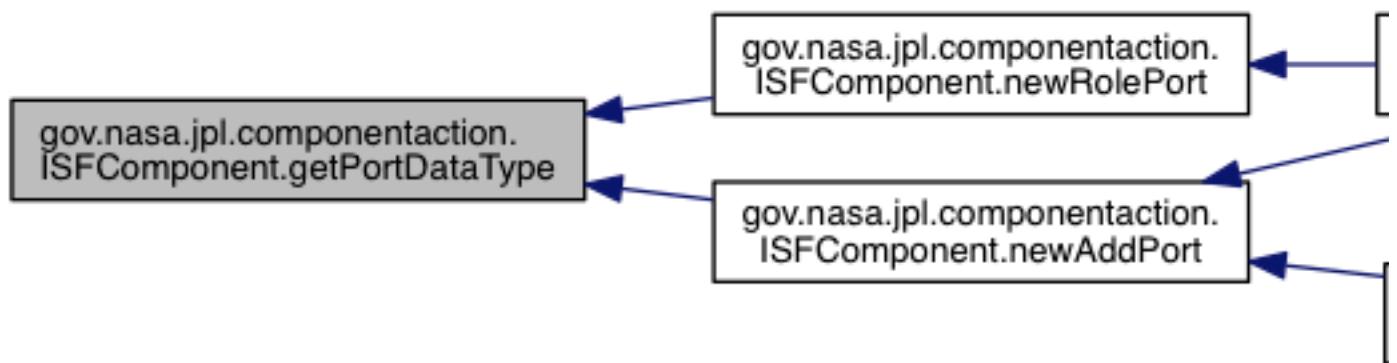
Returns:

port data type string

Exceptions:

<code>PortException</code>	
----------------------------	--

Here is the caller graph for this function:



String gov.nasa.jpl.componentaction.ISFComponent.getPortDataTypeNamespace (Element portElement)

This returns the port data type without the full path. It also tags on the Namespace.

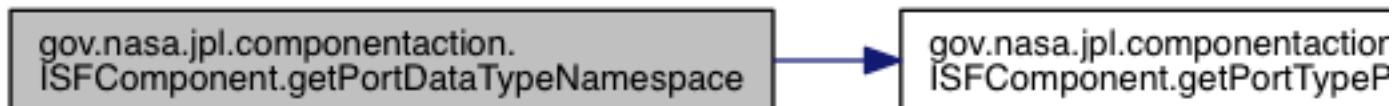
Parameters:

<code>portElement</code>	
--------------------------	--

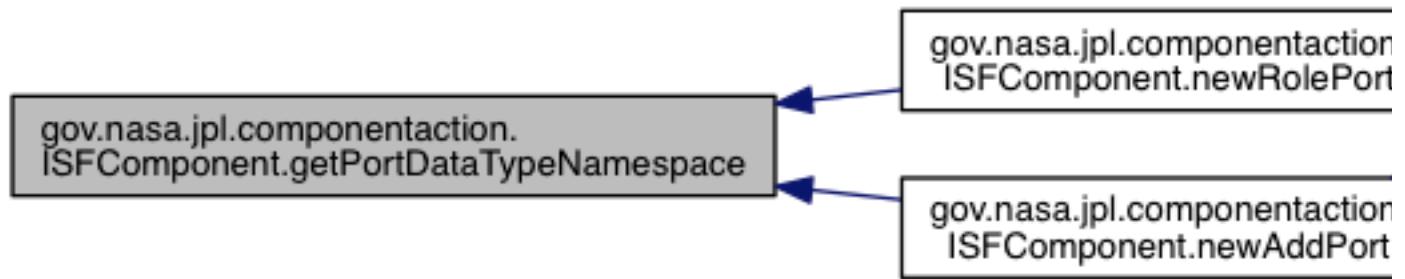
Returns:

`data type + '::' + name space`

Here is the call graph for this function:



Here is the caller graph for this function:



HashMap<String, portObject> gov.nasa.jpl.componentaction.ISFComponent.getPortHashList ()

Returns the filtList HashMap, a map with a string associated with a port object.

Returns:

HashMap

String gov.nasa.jpl.componentaction.ISFComponent.getPortName (Element portElement)

Returns the port name.

Parameters:

<i>portElement</i>	port which the name will be extracted from
--------------------	--

Returns:

The name of the port

String gov.nasa.jpl.componentaction.ISFComponent.getPortProperty (Element portElement, String property)

Returns the property value of *async_input* stereotype of the input element argument.

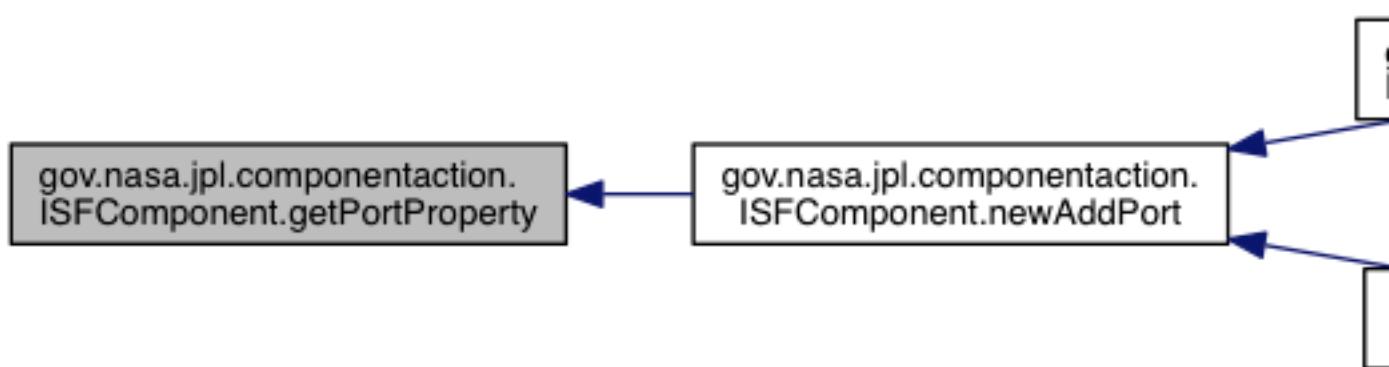
Parameters:

<i>portElement</i>	Element of the port type
<i>property</i>	String of value to be looked for in the stereotype <code>async_input</code>

Returns:

The value of the attribute from property

Here is the caller graph for this function:



String gov.nasa.jpl.componentaction.ISFComponent.getPortStereotype (Element *portElement*)

Returns the port stereotype.

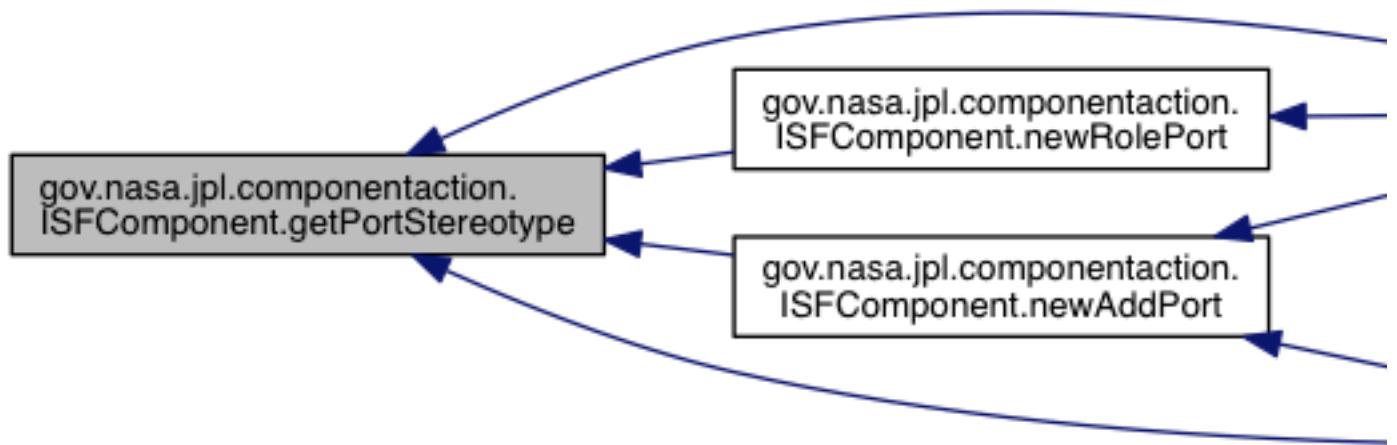
Parameters:

<i>portElement</i>	port which the stereotype will be extracted from
--------------------	--

Returns:

The stereotype of the port

Here is the caller graph for this function:



String gov.nasa.jpl.componentaction.ISFComponent.getPortTypeProperty (Element *portElement*, String *property*)

Returns the property value of PortType stereotype of the input element argument.

Parameters:

<i>portElement</i>	Element of the port type
<i>property</i>	String of value to be looked for in the stereotype PortType

Returns:

The value of the attribute from property

Here is the caller graph for this function:

String gov.nasa.jpl.componentaction.ISFComponent.getStereotype ()

Returns compStereotype

Returns:

the stereotype of the object

String gov.nasa.jpl.componentaction.ISFComponent.getURI ()

Returns the component's URI, which is the location in the tree hierarchy.

Returns:

string of position in hierarchy

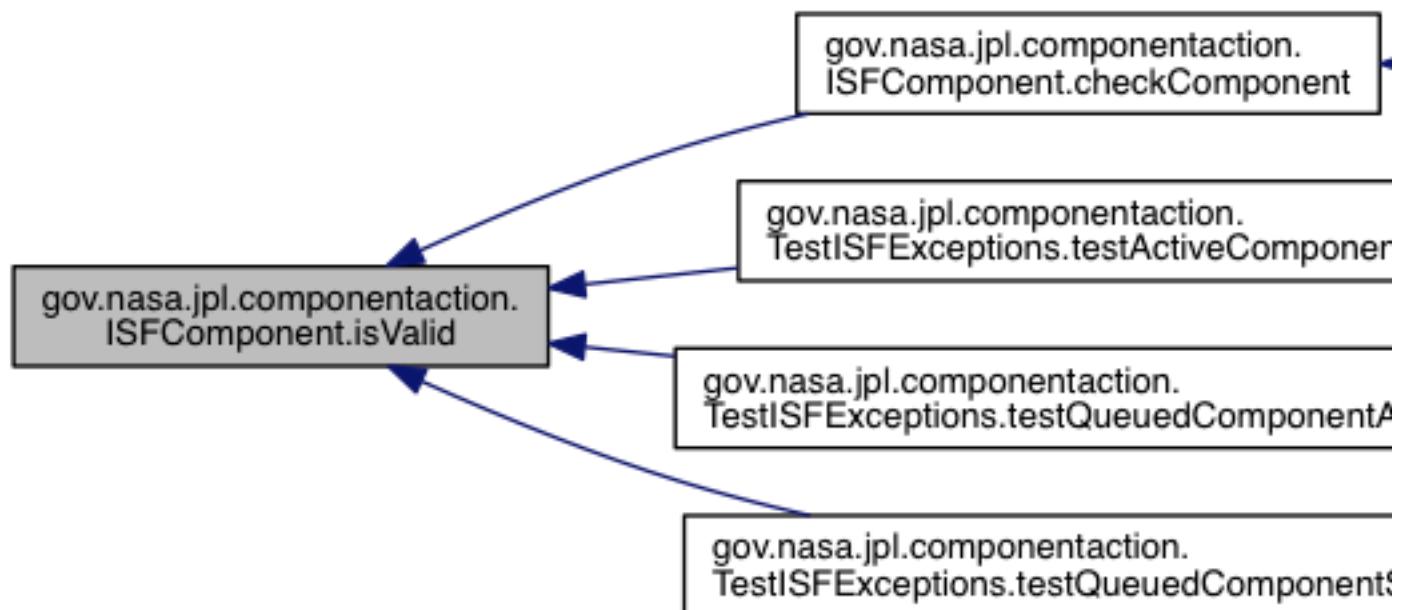
boolean gov.nasa.jpl.componentaction.ISFComponent.isValid (String stereotype, String portType, boolean atLeastOne)

Checks if a component is of a specific stereotype If so, checks if the component satisfies the requirement for a specific port type

Parameters:

<i>stereotype</i>	the stereotype the component should have
<i>portType</i>	the port type to check for
<i>atLeastOne</i>	if the component should have the port type

Here is the caller graph for this function:



void gov.nasa.jpl.componentaction.ISFComponent.newAddPort (Element *portElement*) throws PortException

Creates a new port object using the port Element inputed. This processes the port and can throw an exception if something in the port is illegal. It also adds the port into fillList.

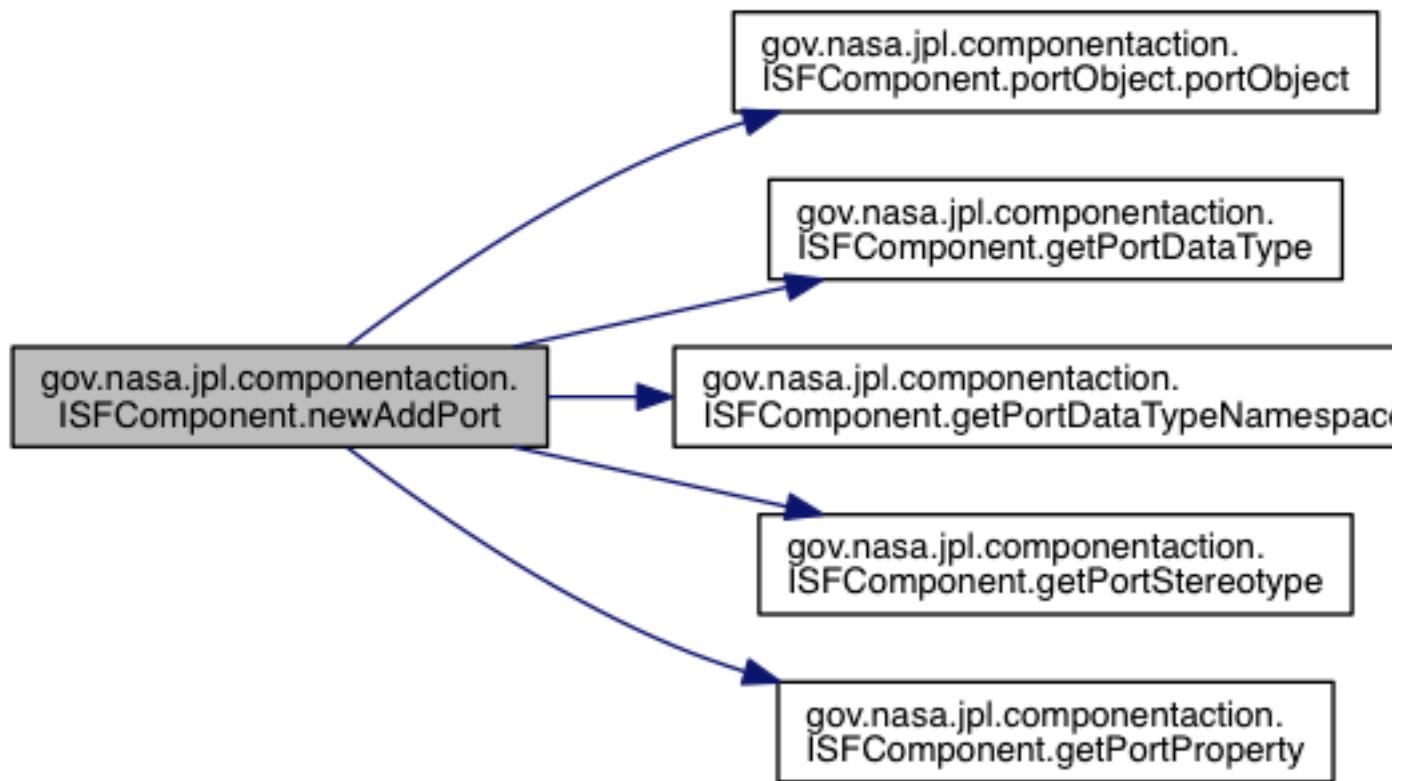
Parameters:

<i>portElement</i>	
--------------------	--

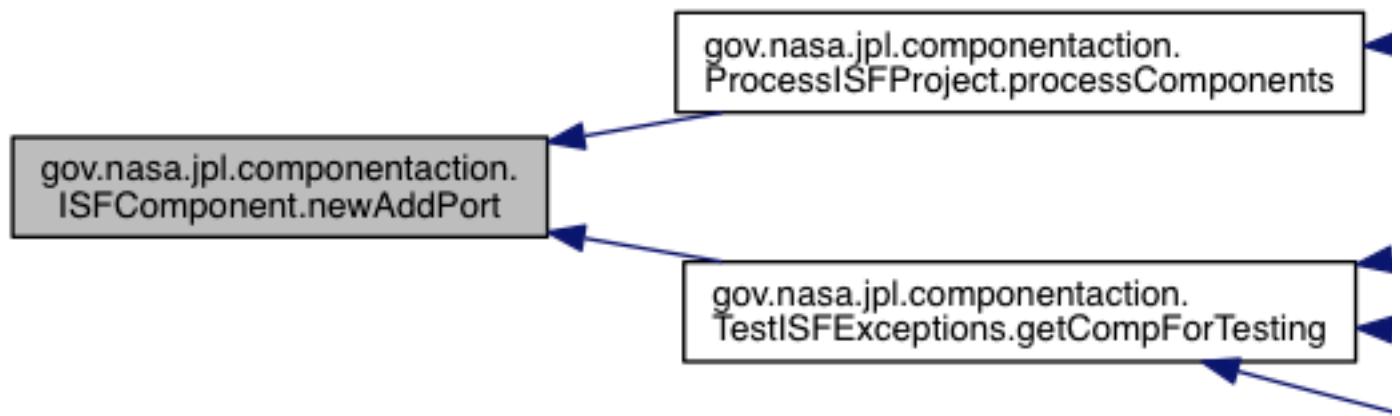
Exceptions:

<i>PortException</i>	
----------------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



void gov.nasa.jpl.componentaction.ISFComponent.newRolePort (Element *portElement*) throws PortException

Creates a new port object using the port Element inputed. This processes the port and can throw an exception if something in the port is illegal. It also adds the port into fillList.

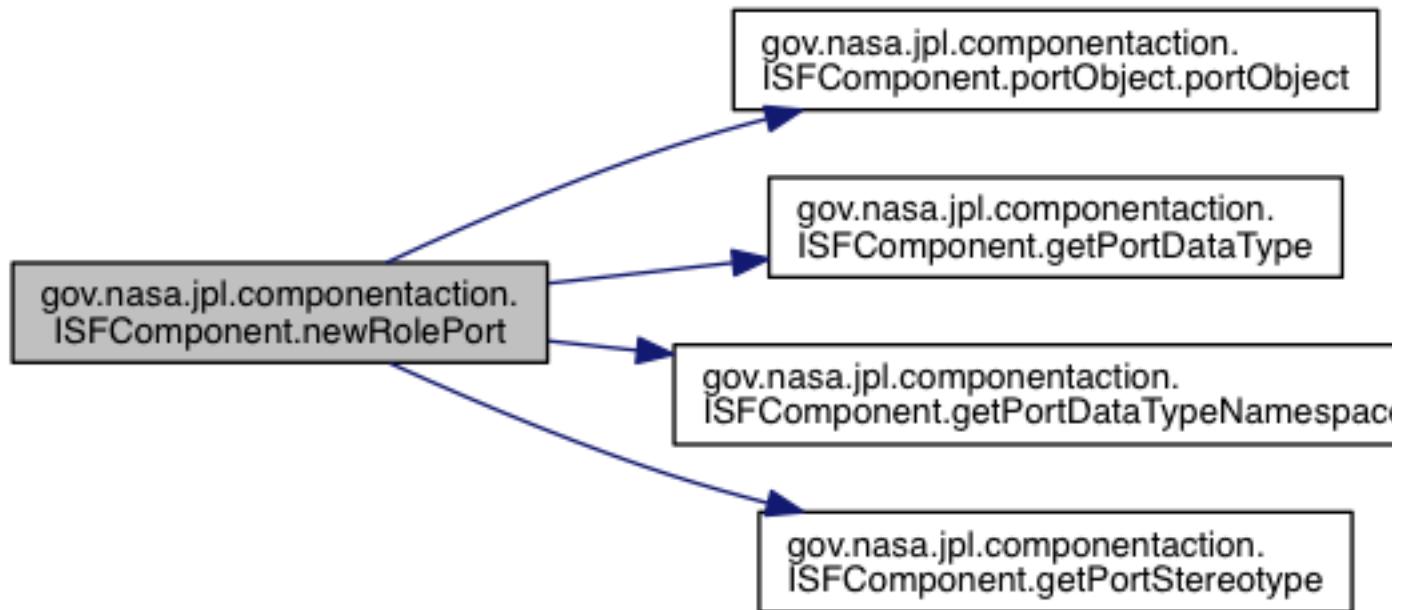
Parameters:

<i>portElement</i>	
--------------------	--

Exceptions:

<i>PortException</i>	
----------------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



```
void gov.nasa.jpl.componentaction.ISFComponent.print()
```

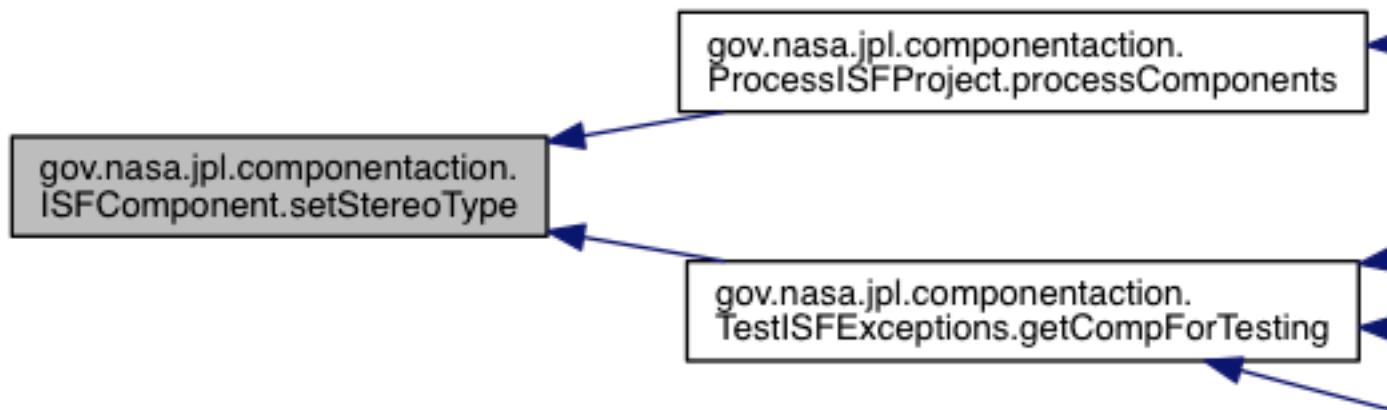
```
void gov.nasa.jpl.componentaction.ISFComponent.setStereoType (String compStereotype)
```

Sets the component stereotype string value

Parameters:

<i>compStereotype</i>	Stereotype string
-----------------------	-------------------

Here is the caller graph for this function:



```
void gov.nasa.jpl.componentaction.ISFComponent.warnLog (String errStr)
```

Prints a warning to the Java standard out as well as the MagicDraw console.

Parameters:

<i>errStr</i>	Message to be printed
---------------	-----------------------

Member Data Documentation

Element gov.nasa.jpl.componentaction.ISFComponent.componentElement [private]

String gov.nasa.jpl.componentaction.ISFComponent.compStereotype [private]

HashMap<String, portObject> gov.nasa.jpl.componentaction.ISFComponent.filtList [package]

String gov.nasa.jpl.componentaction.ISFComponent.IncludeCmdXML [package]

String gov.nasa.jpl.componentaction.ISFComponent.IncludeEvrXML [package]

String gov.nasa.jpl.componentaction.ISFComponent.IncludeIncludeHdr [package]

String gov.nasa.jpl.componentaction.ISFComponent.IncludeInternalIFXML [package]

String gov.nasa.jpl.componentaction.ISFComponent.IncludeParamXML [package]

String gov.nasa.jpl.componentaction.ISFComponent.IncludeTImXML [package]

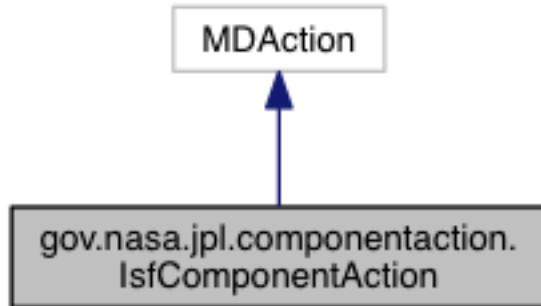
String gov.nasa.jpl.componentaction.ISFComponent.URIName [private]

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

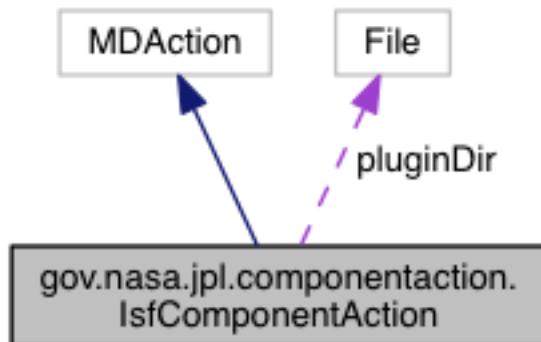
- src/gov/nasa/jpl/componentaction/ISFComponent.java

gov.nasa.jpl.componentaction.IsfComponentAction Class Reference

Inheritance diagram for gov.nasa.jpl.componentaction.IsfComponentAction:



Collaboration diagram for gov.nasa.jpl.componentaction.IsfComponentAction:



Public Member Functions

- **IsfComponentAction** (String id, String name, File **pluginDir**)
- void **actionPerformed** (ActionEvent e)

Package Attributes

- File **pluginDir**

Static Private Attributes

- static final long **serialVersionUID** = -6790954285526957354L

Detailed Description

Used to only generate Component and Port XML diagrams.

Constructor & Destructor Documentation

gov.nasa.jpl.componentaction.IsfComponentAction.IsfComponentAction (String *id*, String *name*, File *pluginDir*)

Member Function Documentation

void gov.nasa.jpl.componentaction.IsfComponentAction.actionPerformed (ActionEvent e)

See also:

java.awt.event.ActionListener::actionPerformed(java.awt.event.ActionEvent)

Here is the call graph for this function:

Member Data Documentation

File gov.nasa.jpl.componentaction.IsfComponentAction.pluginDir [package]

**final long gov.nasa.jpl.componentaction.IsfComponentAction.serialVersionUID = -
6790954285526957354L [static], [private]**

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

- src/gov/nasa/jpl/componentaction/**IsfComponentAction.java**

gov.nasa.jpl.componentaction.isfxmlwriter.IsfCompXmlWriter Class Reference

Static Public Member Functions

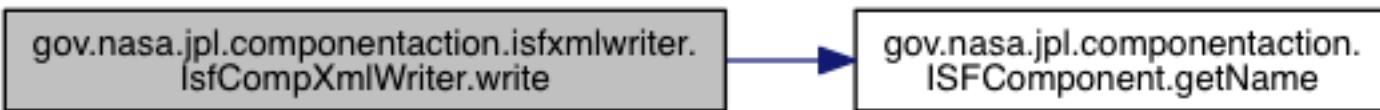
- static void **write** (ISFComponent comp, String fileName, String outDir, File pluginDir)

Member Function Documentation

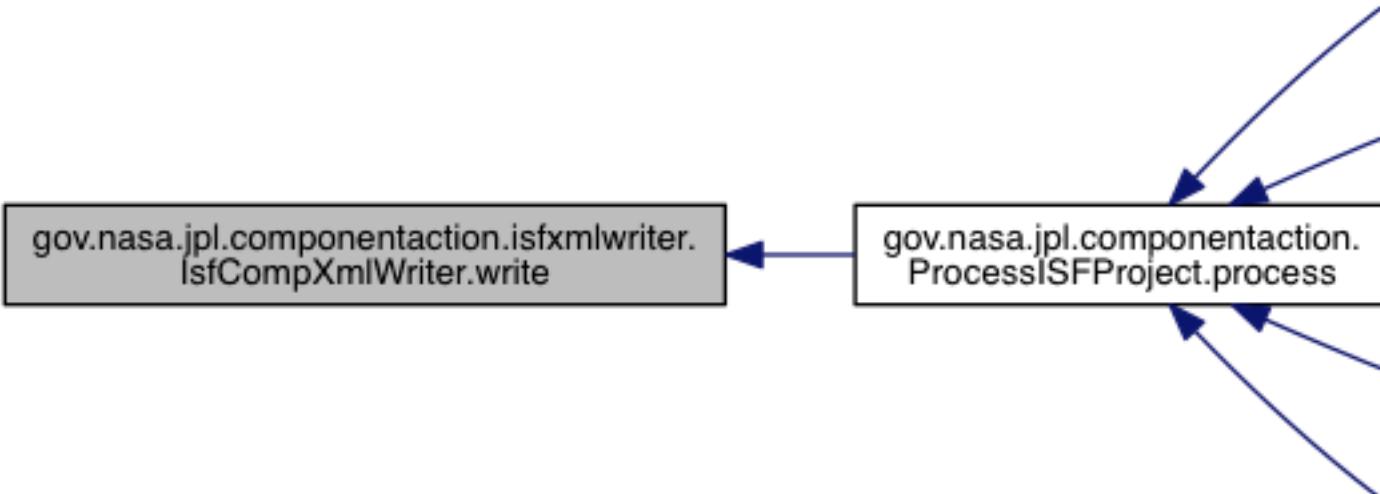
```
static void gov.nasa.jpl.componentaction.isfxmlwriter.IsfCompXmlWriter.write (ISFComponent  
comp, String fileName, String outDir, File pluginDir) [static]
```

Writes the component XML file.

Here is the call graph for this function:



Here is the caller graph for this function:

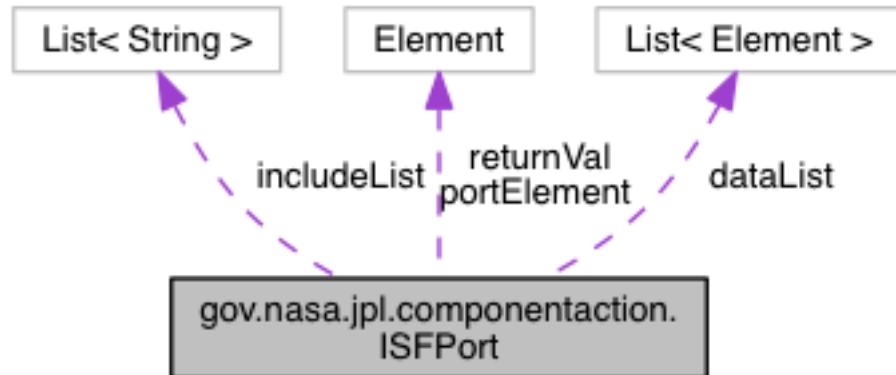


The documentation for this class was generated from the following file:

- [src/gov/nasa/jpl/componentaction/isfxmlwriter/IsfCompXmlWriter.java](#)

gov.nasa.jpl.componentaction.ISFPort Class Reference

Collaboration diagram for gov.nasa.jpl.componentaction.ISFPort:



Public Member Functions

- `ISFPort (Element portElement)`
- `String getNamespace ()`
- `void addData (Element dataElement)`
- `void addReturn (Element dataElement)`
- `boolean isReturnEnumeration (Element dataElement)`
- `boolean hasReturn ()`
- `Element getReturn ()`
- `String getReturnType (Element dataElement)`
- `String getName ()`
- `String[] getDataName (Element dataElement)`
- `String getDataType (Element dataElement)`
- `List< Element > getDataList ()`
- `List< String > getIncludeList ()`
- `List< EnumerationLiteral > getEnumItems (Element dataElement)`
- `List< EnumerationLiteral > getReturnEnumItems (Element dataElement)`
- `boolean isEnumeration (Element dataElement)`
- `String getPortTypeProperty (Element dataElement, String property)`
- `String getDataTypeProperty (Element portElement, String property)`
- `void print ()`

Private Attributes

- `Element portElement`
- `List< Element > dataList`
- `Element returnVal`
- `List< String > includeList`

Constructor & Destructor Documentation

`gov.nasa.jpl.componentaction.ISFPort.ISFPort (Element portElement)`

Creates an `ISFPort` object

Parameters:

<i>portElement</i>	Element of port type to be used in the creation of the object
--------------------	---

Member Function Documentation

void gov.nasa.jpl.componentaction.ISFPort.addData (Element *dataElement*)

Adds Element to the dataList and checks to make sure the datatype is not empty before adding it to the includeList.

Parameters:

<i>dataElement</i>	Element to be added
--------------------	---------------------

Here is the call graph for this function:



Here is the caller graph for this function:



```
void gov.nasa.jpl.componentaction.ISFPort.addReturn (Element dataElement)
```

Adds Element to the includeList if the datatype is not empty.

Parameters:

<i>dataElement</i>

Here is the call graph for this function:



Here is the caller graph for this function:



```
List<Element> gov.nasa.jpl.componentaction.ISFPort.getDataList ()
```

Returns the dataList

Returns:

List of Elements

`String [] gov.nasa.jpl.componentaction.ISFPort.getDataName (Element dataElement)`

Checks if the dataElement is a reference or pointer and returns a list of Strings where the first value is the pass type and the second value is the name of the dataElement.

Parameters:

<code>dataElement</code>	
--------------------------	--

Returns:

List of Strings

`String gov.nasa.jpl.componentaction.ISFPort.getDataType (Element dataElement)`

Returns the data type with the namespace property prepended to it.

Parameters:

<code>dataElement</code>	
--------------------------	--

Returns:

namespace String

Here is the call graph for this function:

**`String gov.nasa.jpl.componentaction.ISFPort.getDataTypeProperty (Element portElement, String property)`**

Returns the property value of DataType stereotype of the input element argument.

Parameters:

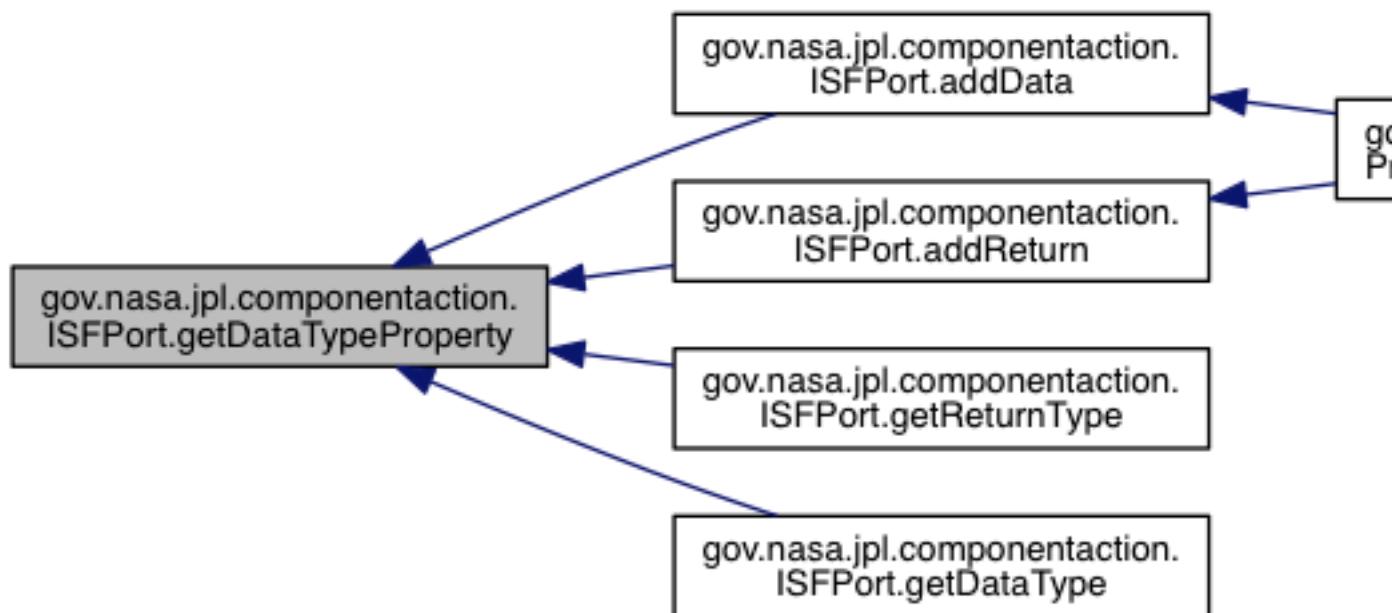
<code>dataElement</code>	The Element in question.
--------------------------	--------------------------

<code>property</code>	The parameter that is being accessed within the DataType stereotype
-----------------------	---

Returns:

the associated value of the property in the DataType stereotype

Here is the caller graph for this function:



List<EnumerationLiteral> gov.nasa.jpl.componentaction.ISFPort.getEnumItems (Element *dataElement*)

Returns the enumerated items of the datatype.

Parameters:

<i>dataElement</i>	
--------------------	--

Returns:

List of Enumeration Literal objects

List<String> gov.nasa.jpl.componentaction.ISFPort.getIncludeList ()

Returns the includeList

Returns:

List of Strings

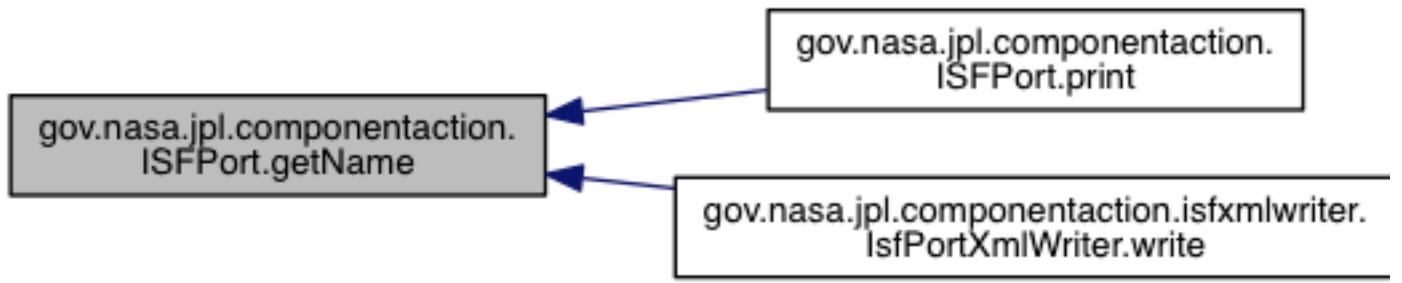
String gov.nasa.jpl.componentaction.ISFPort.getName ()

Returns the last part of the object name.

Returns:

String name

Here is the caller graph for this function:



String gov.nasa.jpl.componentaction.ISFPort.getNamespace ()

Return the port element's namespace

Returns:

namespace string

Here is the call graph for this function:



String gov.nasa.jpl.componentaction.ISFPort.getPortTypeProperty (Element *dataElement*, String *property*)

Returns the property value of the PortType's stereotype of the input element argument.

Parameters:

<i>dataElement</i>	The Element in question.
<i>property</i>	The parameter that is being accessed within the PortType stereotype

Returns:

the associated value of the property in the PortType stereotype

Here is the caller graph for this function:

**Element `gov.nasa.jpl.componentaction.ISFPort.getReturn ()`**

Returns the returnVal.

Returns:

returnVal

List<EnumerationLiteral> `gov.nasa.jpl.componentaction.ISFPort.getReturnEnumItems (Element dataElement)`

Get enumeration items from a Port's Enumeration return type

Parameters:

<code>dataElement</code>	
--------------------------	--

Returns:**String `gov.nasa.jpl.componentaction.ISFPort.getReturnType (Element dataElement)`**

Returns String based off the dataElement.

Parameters:

<code>dataElement</code>	Input Element
--------------------------	---------------

Returns:

ReturnType

Here is the call graph for this function:

**boolean `gov.nasa.jpl.componentaction.ISFPort.hasReturn ()`**

Checks if the returnVal is null.

Returns:

True if the ISFPort has a returnValue

boolean `gov.nasa.jpl.componentaction.ISFPort.isEnumeration (Element dataElement)`

Checks if the Type of the dataElement is an instance of Enumeration.

Parameters:

<code>dataElement</code>	
--------------------------	--

Returns:

boolean

boolean gov.nasa.jpl.componentaction.ISFPort.isReturnEnumeration (Element *dataElement*)

Checks if the Element object is an instance of Enumeration.

Parameters:

<i>dataElement</i>	Object to be checked
--------------------	----------------------

Returns:

True if the Element is an instance of Enumeration

void gov.nasa.jpl.componentaction.ISFPort.print ()

Prints the name of the ISFPort object.

Here is the call graph for this function:



Member Data Documentation

List<Element> gov.nasa.jpl.componentaction.ISFPort.dataList [private]

List<String> gov.nasa.jpl.componentaction.ISFPort.includeList [private]

Element gov.nasa.jpl.componentaction.ISFPort.portElement [private]

Element gov.nasa.jpl.componentaction.ISFPort.returnValue [private]

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

- src/gov/nasa/jpl/componentaction/ISFPort.java

gov.nasa.jpl.componentaction.isfxmlwriter.IsfPortXmlWriter Class Reference

Static Public Member Functions

- static void **write** (ISFPort port, String fileName, String outDir, File pluginDir)

Member Function Documentation

static void gov.nasa.jpl.componentaction.isfxmlwriter.IsfPortXmlWriter.write (ISFPort port, String fileName, String outDir, File pluginDir) [static]

Writes the Port XML files.

Parameters:

<i>port</i>	
<i>fileName</i>	
<i>outDir</i>	
<i>pluginDir</i>	

Here is the call graph for this function:



Here is the caller graph for this function:

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

- src/gov/nasa/jpl/componentaction/isfxmlwriter/**IsfPortXmlWriter.java**

gov.nasa.jpl.componentaction.ISFSubsystem Class Reference

Classes

- class **componentType**
- class **physicalConnectionType**
- class **topologyModel**

Static Public Member Functions

- static **componentType createComponent** (String name, String nameSpace, String type, String baseID, String instanceWindow, String xmlPath)
- static int **getSourceIndex** (Connector conn) throws ConnectorException
- static int **getTargetIndex** (Connector conn) throws ConnectorException
- static void **print** ()
- static String **getFileLocation** (Element instanceElement)
- static String **getBaseId** (Element instanceElement)
- static String **getInstanceWindow** (Element instanceElement)
- static String **getSourceComp** (Connector c) throws ConnectorException
- static String **getTargetComp** (Connector c) throws ConnectorException
- static String **getCompName** (ConnectorEnd conn) throws ConnectorException
- static String **getIDSourcePort** (Connector c) throws ConnectorException
- static String **getIDTargetPort** (Connector c) throws ConnectorException
- static String **getSourcePort** (Connector c) throws ConnectorException
- static String **getTargetPort** (Connector c) throws ConnectorException
- static String **getPortName** (ConnectorEnd conn) throws ConnectorException
- static String **getLastPartOfString** (String in)
- static String **getQualifiedPort** (ConnectorEnd connEnd)
- static String **getQualifiedSourcePort** (Connector c) throws ConnectorException
- static String **getQualifiedTargetPort** (Connector c) throws ConnectorException
- static String **getSourcePortType** (Connector c) throws ConnectorException
- static String **getTargetPortType** (Connector c) throws ConnectorException
- static ConnectorEnd **getSourceConnEnd** (Connector c) throws ConnectorException
- static ConnectorEnd **getTargetConnEnd** (Connector c) throws ConnectorException
- static ConnectorEnd **getConnEnd** (Connector c, boolean isSource) throws ConnectorException
- static List< Element > **getElementsOfStereotype** (String stereotype)
- static String **getNamespace** (Element componentElement)

Detailed Description

Helper class for **ProcessISFTopology**

Member Function Documentation

static componentType gov.nasa.jpl.componentaction.ISFSubsystem.createComponent (String name, String nameSpace, String type, String baseID, String instanceWindow, String xmlPath) [static]

Creates and returns new **componentType**

Parameters:

<i>name</i>	Component name
<i>nameSpace</i>	Component name space
<i>type</i>	Component Type
<i>baseID</i>	Base ID
<i>instanceWindow</i>	Instance Window ID

Returns:**componentType**

Here is the caller graph for this function:

```
static String gov.nasa.jpl.componentaction.ISFSubsystem.getBaseId (Element  
instanceElement) [static]
```

Returns the base ID of the element.

Parameters:

<i>instanceElement</i>	
------------------------	--

Returns:

Here is the caller graph for this function:



```
static String gov.nasa.jpl.componentaction.ISFSubsystem.getCompName (ConnectorEnd conn)
throws ConnectorException [static]
```

Returns the component name of the connectorEnd.

Parameters:

<i>conn</i>	
-------------	--

Returns:

Exceptions:

<i>ConnectorException</i>	
---------------------------	--

Here is the caller graph for this function:

```
static ConnectorEnd gov.nasa.jpl.componentaction.ISFSubsystem.getConnEnd (Connector c,
boolean isSource) throws ConnectorException [static]
```

Both getSourceConnEnd and getTargetConnEnd use this function. Incorporates error handling/checking.

The function checks the direction property of the first stereotype found in each port of the connector. Depending on what the connector can see within and around the system, the function will return what it sees as the source and what it sees as the target.

Parameters:

<i>c</i>	Connector
<i>isSource</i>	true if looking for source end, false if looking for target end

Returns:

Connector End

Exceptions:

<i>ConnectorException</i>	
---------------------------	--

Here is the call graph for this function:



Here is the caller graph for this function:


```
static List<Element> gov.nasa.jpl.componentaction.ISFSubsystem.getElementsOfStereotype  
(String stereotype) [static]
```

Returns a list of elements with the given stereotype String.

Parameters:

<i>stereotype</i>	String
-------------------	--------

Returns:

List of Objects

Here is the caller graph for this function:



```
static String gov.nasa.jpl.componentaction.ISFSubsystem.getFileLocation (Element  
instanceElement) [static]
```

Here is the caller graph for this function:

```
static String gov.nasa.jpl.componentaction.ISFSubsystem.getIDSourcePort (Connector c) throws  
ConnectorException [static]
```

Returns the generic ID of the source port on this Connector.

This ID is not unique among different instances of the same module.

Parameters:

c	Connector
---	-----------

Returns:

ID String

Exceptions:

ConnectorException	
--------------------	--

Here is the call graph for this function:



```
static String gov.nasa.jpl.componentaction.ISFSubsystem.getIDTargetPort (Connector c) throws  
ConnectorException [static]
```

Returns the generic ID of the target port on this Connector.

This ID is not unique among different instances of the same module.

Parameters:

<i>c</i>	Connector
----------	-----------

Returns:

ID String

Exceptions:

<i>ConnectorException</i>	
---------------------------	--

Here is the call graph for this function:



static String gov.nasa.jpl.componentaction.ISFSubsystem.getInstanceWindow (Element *instanceElement*) [static]

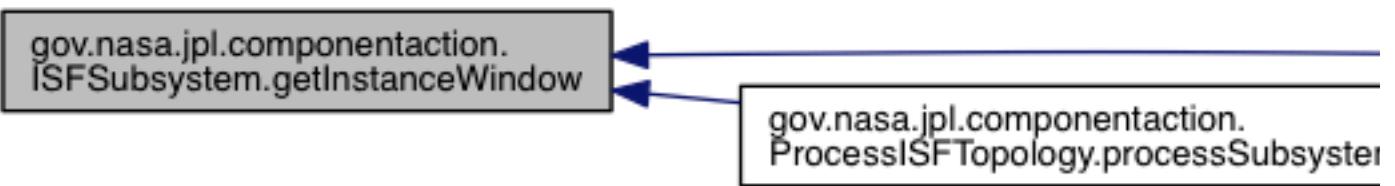
Returns the instance window of the element.

Parameters:

<i>instanceElement</i>	
------------------------	--

Returns:

Here is the caller graph for this function:



```
static String gov.nasa.jpl.componentaction.ISFSubsystem.getLastPartOfString (String  
in) [static]
```

Splits the *in* string with '' and returns the last values in the array.

Parameters:

<i>in</i>	
-----------	--

Returns:

Here is the caller graph for this function:

```
static String gov.nasa.jpl.componentaction.ISFSubsystem.getNamespace (Element  
componentElement) [static]
```

Returns the name space of the element.

Parameters:

<i>componentElement</i>	
-------------------------	--

Returns:

Here is the caller graph for this function:



```
static String gov.nasa.jpl.componentaction.ISFSubsystem.getPortName (ConnectorEnd conn)
throws ConnectorException [static]
```

Returns the generic name of the role of the ConnectorEnd.

Parameters:

<i>conn</i>	ConnectorEnd
-------------	--------------

Returns:

Name

Exceptions:

<i>ConnectorException</i>	
---------------------------	--

Here is the caller graph for this function:

```
static String gov.nasa.jpl.componentaction.ISFSubsystem.getQualifiedPort (ConnectorEnd
connEnd) [static]
```

Returns a string with the qualified name of the base port it is attached to, along with the qualified name of the instance of the port (if it exists).

This can be used to uniquely identify a port against other ports in a subsystem.

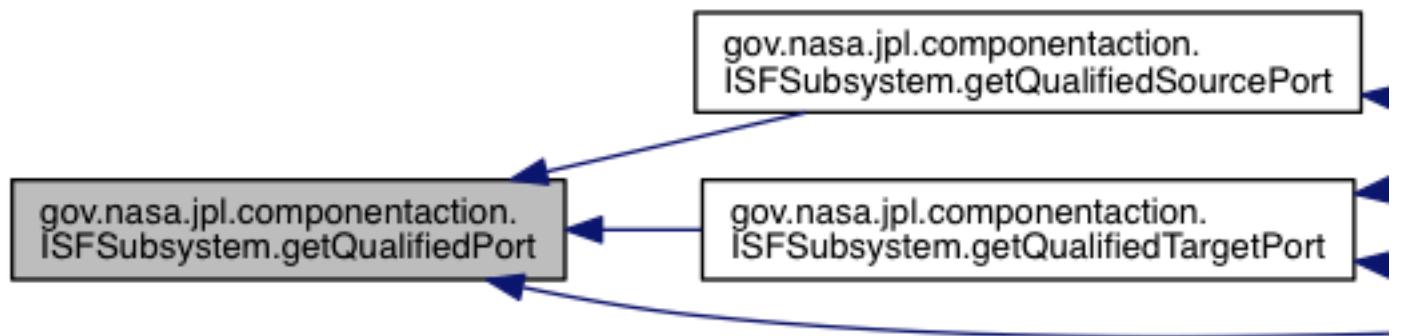
Parameters:

<i>connEnd</i>	ConnectorEnd
----------------	--------------

Returns:

String of the combined qualified name.

Here is the caller graph for this function:



```
static String gov.nasa.jpl.componentaction.ISFSubsystem.getQualifiedSourcePort (Connector c)
throws ConnectorException [static]
```

Returns the qualified name of the source port from the connector.

The qualified name describes the port by providing the port's name, the module it is attached to, and the location of the module.

Parameters:

c	Connector
---	-----------

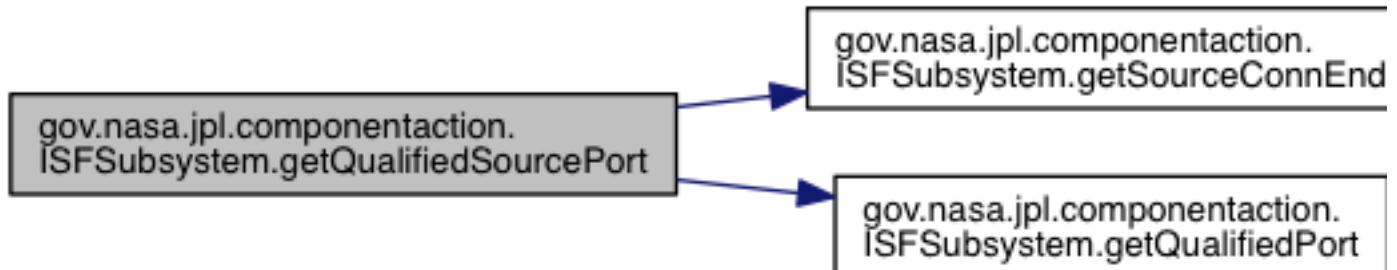
Returns:

String qualified name

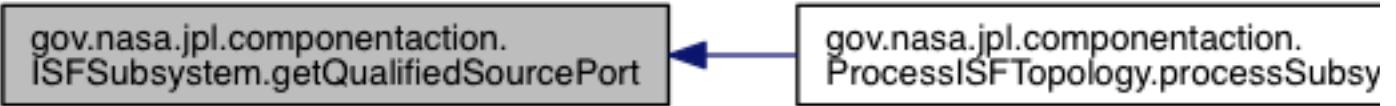
Exceptions:

ConnectorException

Here is the call graph for this function:



Here is the caller graph for this function:



```
static String gov.nasa.jpl.componentaction.ISFSubsystem.getQualifiedTargetPort (Connector c)
throws ConnectorException [static]
```

Returns the qualified name of the target port from the connector.

The qualified name describes the port by providing the port's name, the module it is attached to, and the location of the module.

Parameters:

c	Connector
---	-----------

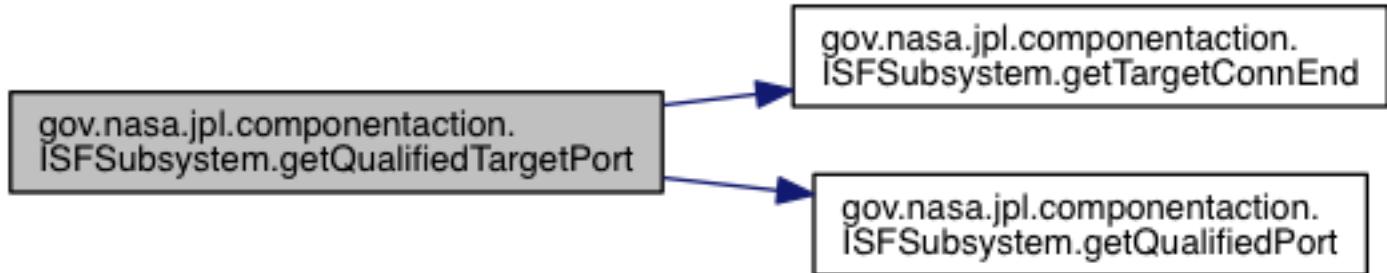
Returns:

String qualified name

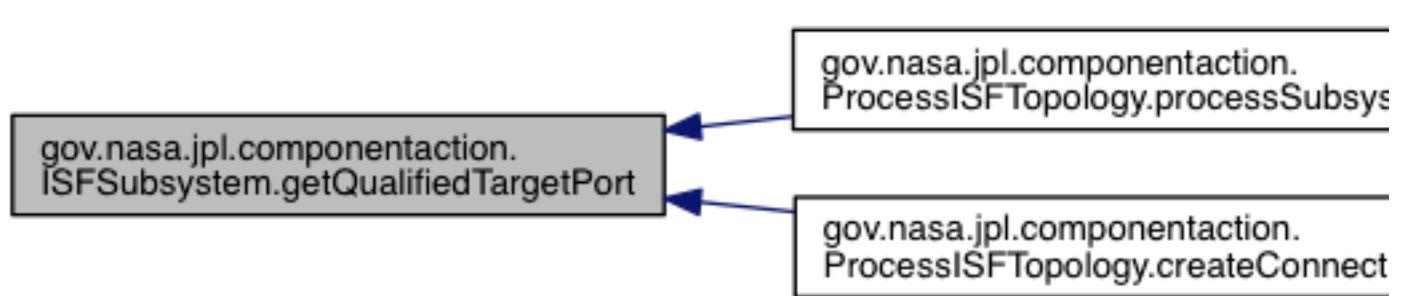
Exceptions:

ConnectorException

Here is the call graph for this function:



Here is the caller graph for this function:



```
static String gov.nasa.jpl.componentaction.ISFSubsystem.getSourceComp (Connector c) throws  
ConnectorException [static]
```

Returns the source name off the attached component.

Parameters:

c	Connector
---	-----------

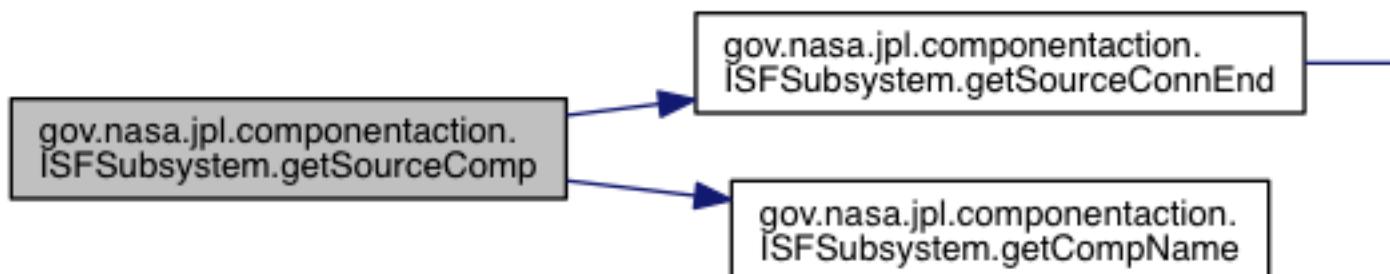
Returns:

Component Name

Exceptions:

<i>ConnectorException</i>	Will raise an exception if the target component does not exist (IE the source is connected to the outside of the subsystem)
---------------------------	---

Here is the call graph for this function:



```
static ConnectorEnd gov.nasa.jpl.componentaction.ISFSubsystem.getSourceConnEnd  
(Connector c) throws ConnectorException [static]
```

Checks the connectorEnd roles to match the returned value with the input (source) connector end.

This function does not check to see if both ConnectorEnds are different or have the same input/output types.

Parameters:

c	Connector
---	-----------

Returns:

Source ConnectorEnd

Here is the call graph for this function:



Here is the caller graph for this function:


```
static int gov.nasa.jpl.componentaction.ISFSubsystem.getSourceIndex (Connector conn) throws  
ConnectorException [static]
```

Returns the multiplicity of the source end of the connector.

Parameters:

<i>conn</i>	Connector
-------------	-----------

Returns:

multiplicity

Exceptions:

<i>ConnectorException</i>	
---------------------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



```
static String gov.nasa.jpl.componentaction.ISFSubsystem.getSourcePort (Connector c) throws  
ConnectorException [static]
```

Returns the generic name of source port on this Connector.

Parameters:

c	Connector
---	-----------

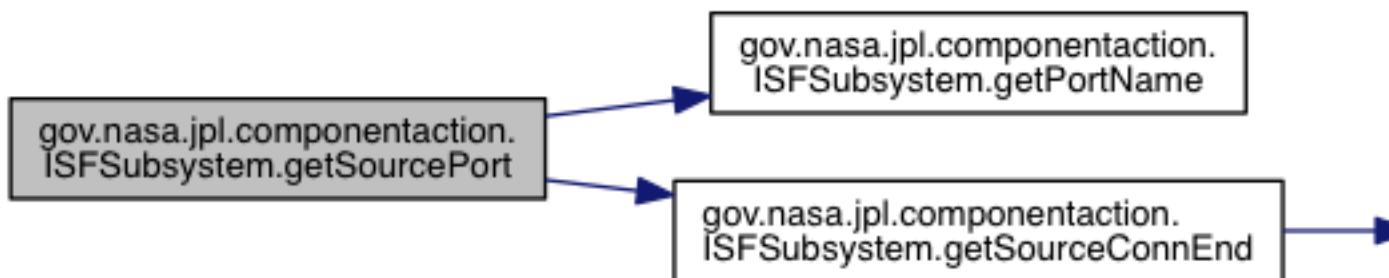
Returns:

Name

Exceptions:

ConnectorException	
--------------------	--

Here is the call graph for this function:



```
static String gov.nasa.jpl.componentaction.ISFSubsystem.getSourcePortType (Connector c)  
throws ConnectorException [static]
```

Returns the source port type from the Connector.

Parameters:

c	Connector
---	-----------

Returns:

Port Type

Exceptions:

ConnectorException	
--------------------	--

Here is the call graph for this function:



```
static String gov.nasa.jpl.componentaction.ISFSubsystem.getTargetComp (Connector c) throws  
ConnectorException [static]
```

Returns the target name off the attached component.

Parameters:

c	Connector
---	-----------

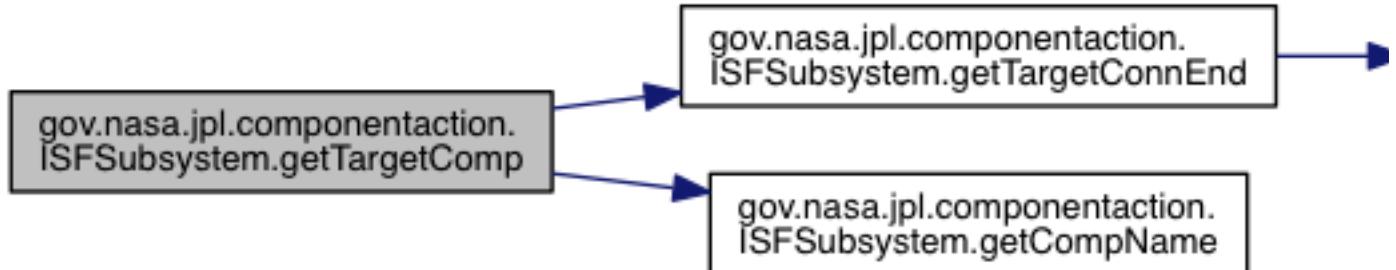
Returns:

Component Name

Exceptions:

<i>ConnectorException</i>	Will raise an exception if the target component does not exist (IE the target is connected to the outside of the subsystem)
<i>on</i>	

Here is the call graph for this function:



static ConnectorEnd gov.nasa.jpl.componentaction.ISFSubsystem.getTargetConnEnd (Connector c) throws ConnectorException [static]

Checks the connectorEnd roles to match the returned value with the output (target) connector end.

This function does not check to see if both ConnectorEnds are different or have the same input/output types.

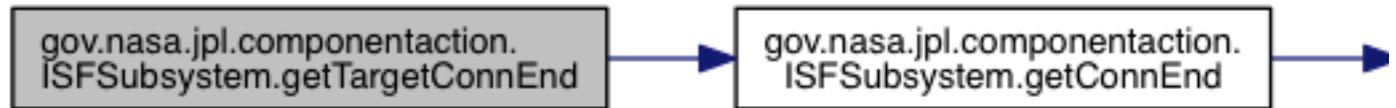
Parameters:

<i>c</i>	Connector
----------	-----------

Returns:

Target ConnectorEnd

Here is the call graph for this function:



Here is the caller graph for this function:


```
static int gov.nasa.jpl.componentaction.ISFSubsystem.getTargetIndex (Connector conn) throws  
ConnectorException [static]
```

Returns the multiplicity of the target end of the connector.

Parameters:

<i>conn</i>	Connector
-------------	-----------

Returns:

multiplicity

Exceptions:

<i>ConnectorException</i>	
---------------------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



```
static String gov.nasa.jpl.componentaction.ISFSubsystem.getTargetPort (Connector c) throws  
ConnectorException [static]
```

Returns the generic name of target port on this Connector.

Parameters:

c	Connector
---	-----------

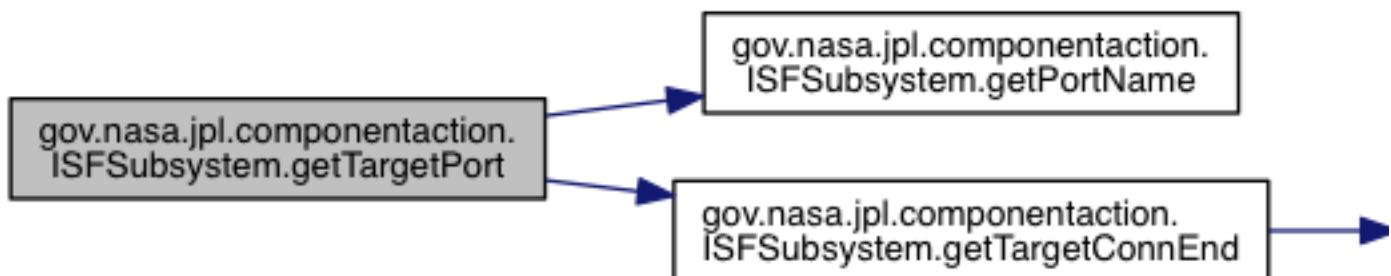
Returns:

Name

Exceptions:

ConnectorException	
--------------------	--

Here is the call graph for this function:



```
static String gov.nasa.jpl.componentaction.ISFSubsystem.getTargetPortType (Connector c)  
throws ConnectorException [static]
```

Returns the target port type from the Connector.

Parameters:

c	Connector
---	-----------

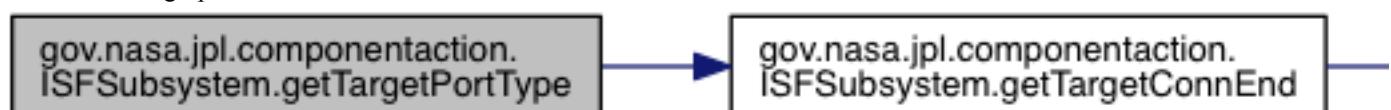
Returns:

Port Type

Exceptions:

ConnectorException	
--------------------	--

Here is the call graph for this function:



```
static void gov.nasa.jpl.componentaction.ISFSubsystem.print () [static]
```

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

- src/gov/nasa/jpl/componentaction/ISFSubsystem.java

gov.nasa.jpl.componentaction.isfxmlwriter.IsfSubXmlWriter Class Reference

Static Public Member Functions

- static void **write** (ISFSubsystem.topologyModel top, String deployment, String fileName, String outDir, File pluginDir)
-

Member Function Documentation

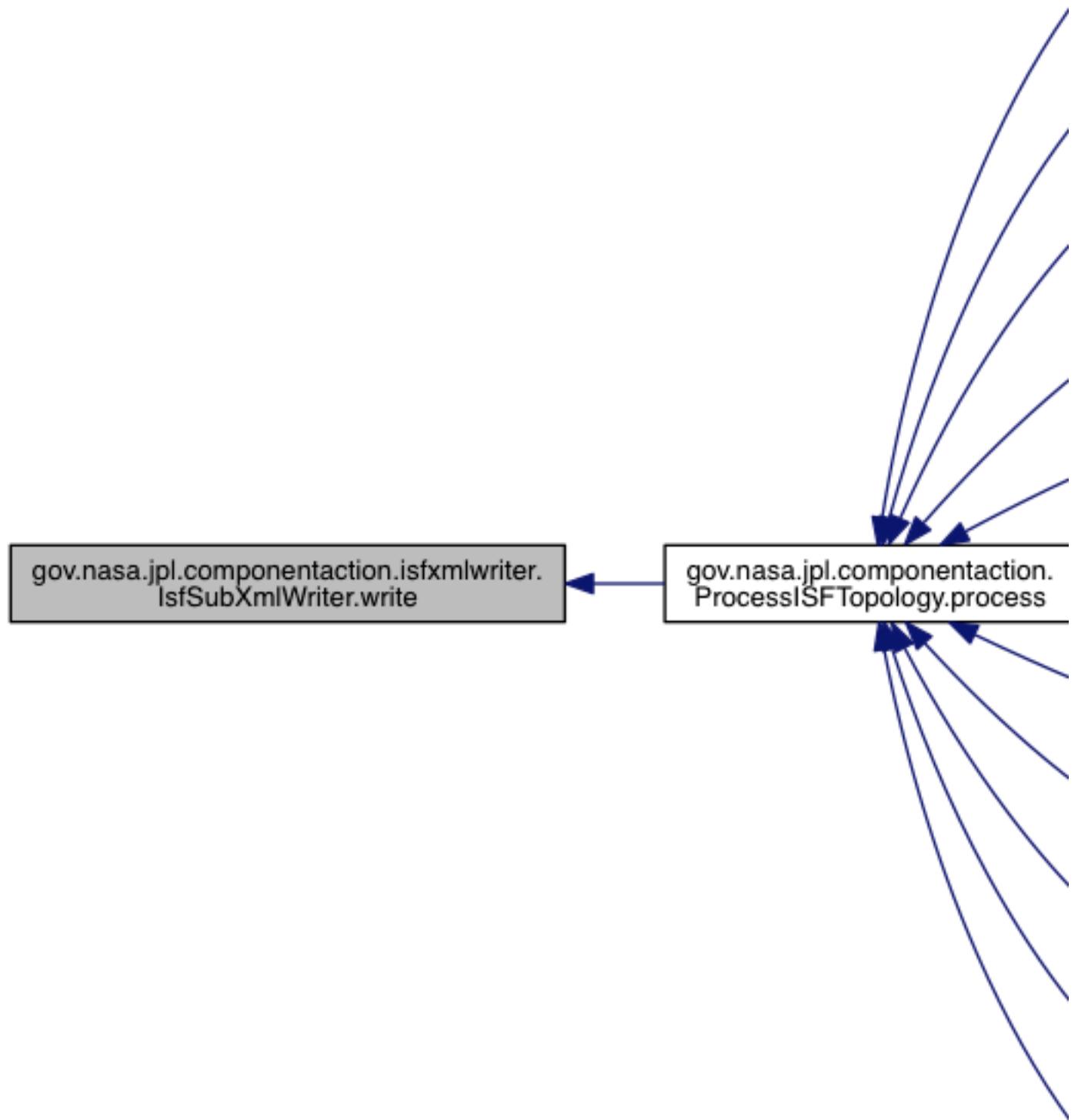
```
static void gov.nasa.jpl.componentaction.isfxmlwriter.IsfSubXmlWriter.write  
(ISFSubsystem.topologyModel top, String deployment, String fileName, String outDir, File  
pluginDir) [static]
```

Writes the subsystem topology XML file.

Parameters:

<i>top</i>	
<i>deployment</i>	
<i>fileName</i>	
<i>outDir</i>	
<i>pluginDir</i>	

Here is the caller graph for this function:

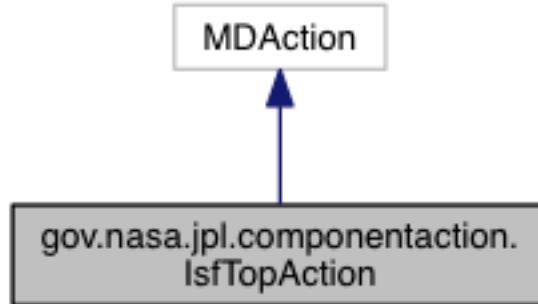


The documentation for this class was generated from the following file:

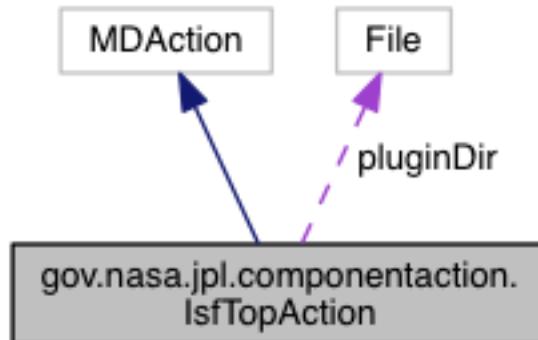
- [src/gov/nasa/jpl/componentaction/isfxmlwriter/IsfSubXmlWriter.java](#)

gov.nasa.jpl.componentaction.IsfTopAction Class Reference

Inheritance diagram for gov.nasa.jpl.componentaction.IsfTopAction:



Collaboration diagram for gov.nasa.jpl.componentaction.IsfTopAction:



Public Member Functions

- **IsfTopAction** (String id, String name, File **pluginDir**)
- void **actionPerformed** (ActionEvent e)

Package Attributes

- File **pluginDir**

Static Private Attributes

- static final long **serialVersionUID** = -6790954285526957354L

Detailed Description

Used to generate only the topology XML diagrams.

Constructor & Destructor Documentation

gov.nasa.jpl.componentaction.IsfTopAction.IsfTopAction (String *id*, String *name*, File *pluginDir*)

Member Function Documentation

void gov.nasa.jpl.componentaction.lsfTopAction.actionPerformed (ActionEvent e)

Runs the **ProcessISFProject.process** function.

See also:

`java.awt.event.ActionListener::actionPerformed(java.awt.event.ActionEvent)`

Here is the call graph for this function:

Member Data Documentation

File `gov.nasa.jpl.componentaction.IsfTopAction.pluginDir` [package]

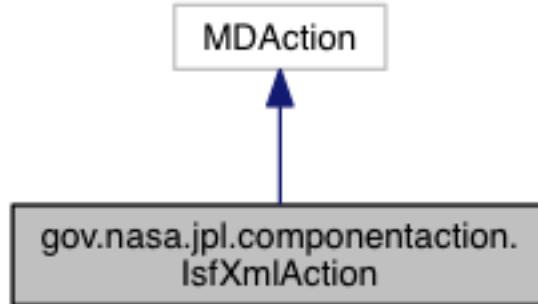
```
final long gov.nasa.jpl.componentaction.IsfTopAction.serialVersionUID = -  
6790954285526957354L [static], [private]
```

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

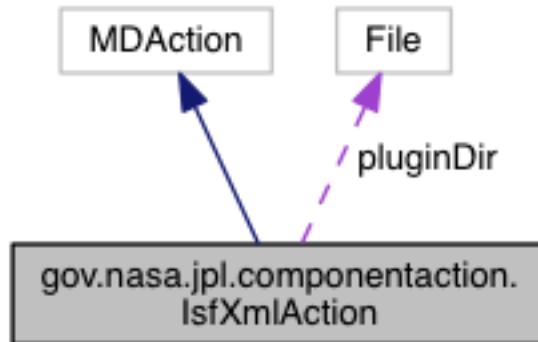
- `src/gov/nasa/jpl/componentaction/IsfTopAction.java`

gov.nasa.jpl.componentaction.IsfXmlAction Class Reference

Inheritance diagram for gov.nasa.jpl.componentaction.IsfXmlAction:



Collaboration diagram for gov.nasa.jpl.componentaction.IsfXmlAction:



Public Member Functions

- **IsfXmlAction** (String id, String name, File **pluginDir**)
- void **actionPerformed** (ActionEvent e)

Package Attributes

- File **pluginDir**

Static Private Attributes

- static final long **serialVersionUID** = -6790954285526957354L

Detailed Description

Used to generate component, port, and topology XML files.

Constructor & Destructor Documentation

gov.nasa.jpl.componentaction.IsfXmlAction.IsfXmlAction (String *id*, String *name*, File *pluginDir*)

Member Function Documentation

void gov.nasa.jpl.componentaction.IsfXmlAction.actionPerformed (ActionEvent e)

Runs the **ProcessISFProject.process** and the **ProcessISFTopology.process** functions.

See also:

`java.awt.event.ActionListener::actionPerformed(java.awt.event.ActionEvent)`

Here is the call graph for this function:

Member Data Documentation

File gov.nasa.jpl.componentaction.IsfXmlAction.pluginDir [package]

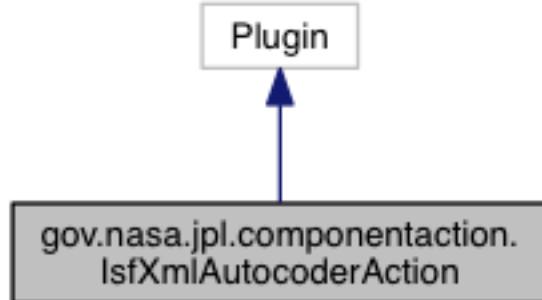
**final long gov.nasa.jpl.componentaction.IsfXmlAction.serialVersionUID = -
6790954285526957354L [static], [private]**

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

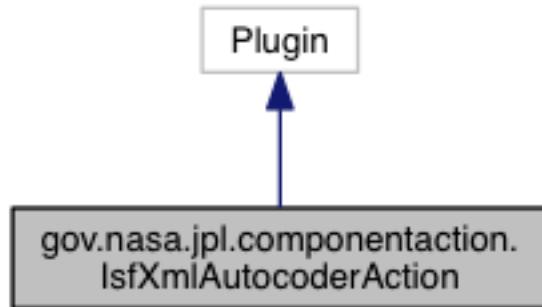
- `src/gov/nasa/jpl/componentaction/IssfXmlAction.java`

gov.nasa.jpl.componentaction.IsfXmlAutocoderAction Class Reference

Inheritance diagram for gov.nasa.jpl.componentaction.IsfXmlAutocoderAction:



Collaboration diagram for gov.nasa.jpl.componentaction.IsfXmlAutocoderAction:



Public Member Functions

- void **init ()**
- boolean **close ()**
- boolean **isSupported ()**

Private Member Functions

- NMAction **getSeparatedActions ()**

Detailed Description

Used to create generation/about tabs within MagicDraw. Buttons are attached to different action object.

Member Function Documentation

boolean gov.nasa.jpl.componentaction.IsfXmlAutocoderAction.close ()

See also:

[com.nomagic.magicdraw.plugins.Plugin::close\(\)](#)

NMACTION gov.nasa.jpl.componentaction.IsfXmlAutocoderAction.getSeparatedActions() [private]

Creates group of actions. This group is separated from others using menu separator (when it is represented in menu). Separator is added for group of actions in one actions category.

Here is the caller graph for this function:



void gov.nasa.jpl.componentaction.IsfXmlAutocoderAction.init()

See also:

[com.nomagic.magicdraw.plugins.Plugin::init\(\)](#)

Here is the call graph for this function:



boolean gov.nasa.jpl.componentaction.IsfXmlAutocoderAction.isSupported()

See also:

[com.nomagic.magicdraw.plugins.Plugin::isSupported\(\)](#)

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

- [src/gov/nasa/jpl/componentaction/IsfXmlAutocoderAction.java](#)

gov.nasa.jpl.componentaction.LoadIDConfig Class Reference

Collaboration diagram for gov.nasa.jpl.componentaction.LoadIDConfig:

Classes

- class **configItem**

Static Public Member Functions

- static void **process** (Project proj, File pluginDir) throws IOException
- static void **changeModel** (Element root, Project proj, Map< String, configItem > configMap) throws LoadIDEException
- static Map< String, configItem > **validateAndProcessConfig** (File config) throws LoadIDEException
- static File **getConfigFile** (File configFile) throws LoadIDEException

Private Member Functions

- **LoadIDConfig ()**

Static Private Attributes

- static Project **project**

Detailed Description

LoadIDConfig allows the user to specify a config CSV that will modify all the baseIDs and window ranges within the model. This allows users to have isfgen generate IDs and use them in the future by adding them into the models.

Exceptions are generated when anything does not conform to what is expected, including file types, file column amounts, and mismatches between the config file and the model.

The format of the config file is so:

component,instance,base_id,base_window_range

SignalGen,SG1,10,15

...

Constructor & Destructor Documentation

`gov.nasa.jpl.componentaction.LoadIDConfig.LoadIDConfig () [private]`

Member Function Documentation

`static void gov.nasa.jpl.componentaction.LoadIDConfig.changeModel (Element root, Project proj, Map< String, configItem > configMap) throws LoadIDEException [static]`

This function goes through the model and matches each component with its respective instance within the configMap.

If there isn't any discrepancies between the model and the config files, baseIDs and window ranges are set to the values from the config file.

All subsystems have their base IDs set to zero.

Parameters:

<code>root</code>	
<code>proj</code>	
<code>configMap</code>	

Exceptions:

<code>LoadIDEException</code>	
-------------------------------	--

Here is the caller graph for this function:

`static File gov.nasa.jpl.componentaction.LoadIDConfig.getConfigFile (File projectFile) throws LoadIDEException [static]`

Opens up a file chooser box for the user to select what config CSV file to choose.

The input argument `projectFile` is used to specify a starting path.

Parameters:

<code>projectFile</code>	
--------------------------	--

Returns:

A config file object

Exceptions:

<code>LoadIDException</code>	
------------------------------	--

Here is the caller graph for this function:

```
static void gov.nasa.jpl.componentaction.LoadIDConfig.process (Project proj, File pluginDir)
throws IOException [static]
```

`process` handles all the phases of this methodology.

Parameters:

<code>proj</code>	Project object
<code>pluginDir</code>	Directory to generate documents into

Exceptions:

<code>IOException</code>	
--------------------------	--

Here is the call graph for this function:

Here is the caller graph for this function:

```
static Map<String , configItem>
gov.nasa.jpl.componentaction.LoadIDConfig.validateAndProcessConfig (File config) throws
LoadIDException [static]
```

Validates and processes the input config file.

Checks if the file is valid, is a .CSV, and has four items per row.

This also constructs a map where the key is an instance name and the value is a **configItem** that holds the associated names and IDs.

Parameters:

<i>config</i>	
---------------	--

Returns:

Exceptions:

<i>LoadIDException</i>	
------------------------	--

Here is the call graph for this function:

Here is the caller graph for this function:

Member Data Documentation

Project `gov.nasa.jpl.componentaction.LoadIDConfig.project` **[static], [private]**

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

- [src/gov/nasa/jpl/componentaction/LoadIDConfig.java](#)

gov.nasa.jpl.componentaction.LoadIDConfigAction Class Reference

Inheritance diagram for gov.nasa.jpl.componentaction.LoadIDConfigAction:

Collaboration diagram for gov.nasa.jpl.componentaction.LoadIDConfigAction:

Public Member Functions

- **LoadIDConfigAction** (String id, String name, File **pluginDir**)
- void **actionPerformed** (ActionEvent e)

Package Attributes

- File **pluginDir**

Static Private Attributes

- static final long serialVersionUID = -6790954285526957354L
-

Detailed Description

Used to generate component, port, and topology XML files.

Constructor & Destructor Documentation

gov.nasa.jpl.componentaction.LoadIDConfigAction.LoadIDConfigAction (String id, String name, File pluginDir)

Member Function Documentation

void gov.nasa.jpl.componentaction.LoadIDConfigAction.actionPerformed (ActionEvent e)

Runs the **LoadIDConfig**

See also:

`java.awt.event.ActionListener::actionPerformed(java.awt.event.ActionEvent)`

Here is the call graph for this function:

Member Data Documentation

File `gov.nasa.jpl.componentaction.LoadIDConfigAction.pluginDir` [package]

```
final long gov.nasa.jpl.componentaction.LoadIDConfigAction.serialVersionUID = -  
6790954285526957354L [static], [private]
```

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

- `src/gov/nasa/jpl/componentaction/LoadIDConfigAction.java`

gov.nasa.jpl.componentaction.LoadIDEException Class Reference

Inheritance diagram for gov.nasa.jpl.componentaction.LoadIDEException:

Collaboration diagram for gov.nasa.jpl.componentaction.LoadIDEException:

Public Member Functions

- **LoadIDEException** (String message)

Static Private Attributes

- static final long serialVersionUID = 1L

Detailed Description

Shell exception for the **LoadIDConfig** class. This just serves as a wrapper that can be used to differentiate between where an error happens.

Constructor & Destructor Documentation

`gov.nasa.jpl.componentaction.LoadIDEException.LoadIDEException (String message)`

Member Data Documentation

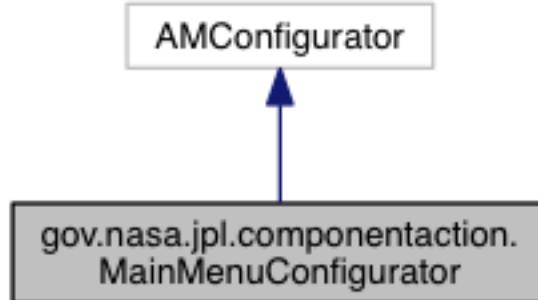
`final long gov.nasa.jpl.componentaction.LoadIDEException serialVersionUID = 1L [static],
[private]`

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

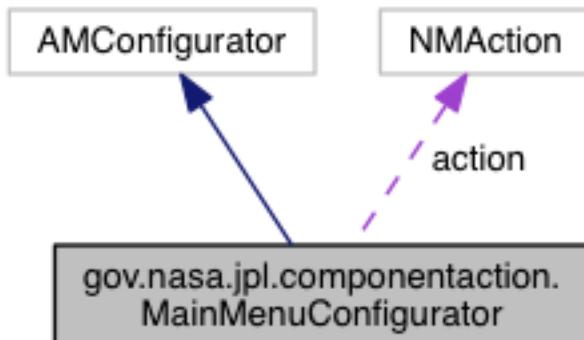
- `src/gov/nasa/jpl/componentaction/LoadIDEException.java`

gov.nasa.jpl.componentaction.MainMenuConfigurator Class Reference

Inheritance diagram for gov.nasa.jpl.componentaction.MainMenuConfigurator:



Collaboration diagram for gov.nasa.jpl.componentaction.MainMenuConfigurator:



Public Member Functions

- **MainMenuConfigurator (NMAction action)**
- void **configure** (ActionsManager mngr)
- int **getPriority ()**

Package Attributes

- String **MPMCS** ="Component Autocoder"

Private Attributes

- NMAction **action**

Detailed Description

Class for configuring main menu and adding new submenu.

Version:

Date

2007-10-05 09:51:43 +0300 (Fri, 05 Oct 2007)

Revision

52410

Author:

Donatas Simkunas

Constructor & Destructor Documentation

gov.nasa.jpl.componentaction.MainMenuConfigurator.MainMenuConfigurator (NMAction action)

Creates configurator.

Parameters:

<i>action</i>	action to be added to main menu.
---------------	----------------------------------

Member Function Documentation

void gov.nasa.jpl.componentaction.MainMenuConfigurator.configure (ActionsManager mngr)**See also:**com.nomagic.actions.AMConfigurator::configure(ActionsManager) Methods adds **action** to given manager Examples category.**int gov.nasa.jpl.componentaction.MainMenuConfigurator.getPriority ()**

Member Data Documentation

NMAction gov.nasa.jpl.componentaction.MainMenuConfigurator.action [private]

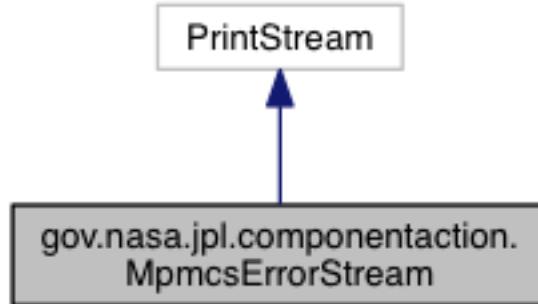
Action will be added to manager.

String gov.nasa.jpl.componentaction.MainMenuConfigurator.MPMCS ="Component Autocoder" [package]**The documentation for this class was generated from the following file:**

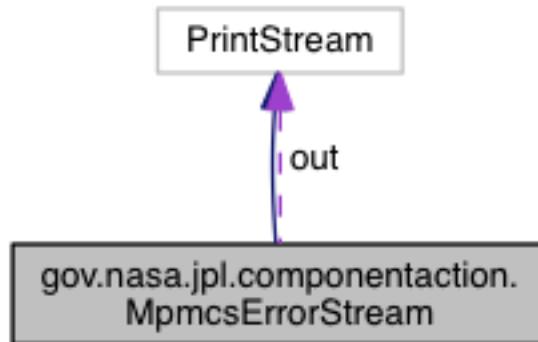
- src/gov/nasa/jpl/componentaction/MainMenuConfigurator.java

gov.nasa.jpl.componentaction.MpmcsErrorStream Class Reference

Inheritance diagram for gov.nasa.jpl.componentaction.MpmcsErrorStream:



Collaboration diagram for gov.nasa.jpl.componentaction.MpmcsErrorStream:



Public Member Functions

- **MpmcsErrorStream** (PrintStream out1, PrintStream out2)
- **MpmcsErrorStream** (OutputStream arg0)
- **MpmcsErrorStream** (String arg0) throws FileNotFoundException
- **MpmcsErrorStream** (File arg0) throws FileNotFoundException
- **MpmcsErrorStream** (OutputStream arg0, boolean arg1)
- **MpmcsErrorStream** (String arg0, String arg1) throws FileNotFoundException, UnsupportedEncodingException
- **MpmcsErrorStream** (File arg0, String arg1) throws FileNotFoundException, UnsupportedEncodingException
- **MpmcsErrorStream** (OutputStream arg0, boolean arg1, String arg2) throws UnsupportedEncodingException
- void **write** (byte buf[], int off, int len)
- void **flush** ()

Private Attributes

- PrintStream **out**

Constructor & Destructor Documentation

`gov.nasa.jpl.componentaction.MpmcsErrorStream.MpmcsErrorStream (PrintStream out1,
PrintStream out2)`

`gov.nasa.jpl.componentaction.MpmcsErrorStream.MpmcsErrorStream (OutputStream arg0)`

`gov.nasa.jpl.componentaction.MpmcsErrorStream.MpmcsErrorStream (String arg0) throws
FileNotFoundException`

`gov.nasa.jpl.componentaction.MpmcsErrorStream.MpmcsErrorStream (File arg0) throws
FileNotFoundException`

`gov.nasa.jpl.componentaction.MpmcsErrorStream.MpmcsErrorStream (OutputStream arg0,
boolean arg1)`

`gov.nasa.jpl.componentaction.MpmcsErrorStream.MpmcsErrorStream (String arg0, String arg1)
throws FileNotFoundException, UnsupportedEncodingException`

`gov.nasa.jpl.componentaction.MpmcsErrorStream.MpmcsErrorStream (File arg0, String arg1)
throws FileNotFoundException, UnsupportedEncodingException`

`gov.nasa.jpl.componentaction.MpmcsErrorStream.MpmcsErrorStream (OutputStream arg0,
boolean arg1, String arg2) throws UnsupportedEncodingException`

Member Function Documentation

`void gov.nasa.jpl.componentaction.MpmcsErrorStream.flush ()`

`void gov.nasa.jpl.componentaction.MpmcsErrorStream.write (byte buf[], int off, int len)`

Member Data Documentation

`PrintStream gov.nasa.jpl.componentaction.MpmcsErrorStream.out [private]`

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

- `src/gov/nasa/jpl/componentaction/MpmcsErrorStream.java`

gov.nasa.jpl.componentaction.Utils.Pair< A, B > Class Template Reference

Public Member Functions

- **Pair** (final A firstObj, final B secondObj)
- int **hashCode** ()
- boolean **equals** (Object obj)
- A **getFirst** ()
- B **getSecond** ()
- void **setFirst** (A first)
- void **setSecond** (B second)

Public Attributes

- A **first**
- B **second**

Private Member Functions

- Utils **getOuterType** ()

Detailed Description

Generic **Pair** class that can be used to hold two items of separate types

Parameters:

<A>	

Constructor & Destructor Documentation

gov.nasa.jpl.componentaction.Utils.Pair< A, B >.Pair (final A *firstObj*, final B *secondObj*)

Here is the caller graph for this function:

Member Function Documentation

boolean gov.nasa.jpl.componentaction.Utils.Pair< A, B >.equals (Object obj)

Here is the call graph for this function:

A gov.nasa.jpl.componentaction.Utils.Pair< A, B >.getFirst ()

Utils gov.nasa.jpl.componentaction.Utils.Pair< A, B >.getOuterType () [private]

Here is the caller graph for this function:

```
B gov.nasa.jpl.componentaction.Utils.Pair< A, B >.getSecond ()
```

```
int gov.nasa.jpl.componentaction.Utils.Pair< A, B >.hashCode ()
```

Here is the call graph for this function:

```
void gov.nasa.jpl.componentaction.Utils.Pair< A, B >.setFirst (A first)
```

```
void gov.nasa.jpl.componentaction.Utils.Pair< A, B >.setSecond (B second)
```

Member Data Documentation

A `gov.nasa.jpl.componentaction.Utils.Pair< A, B >.first`

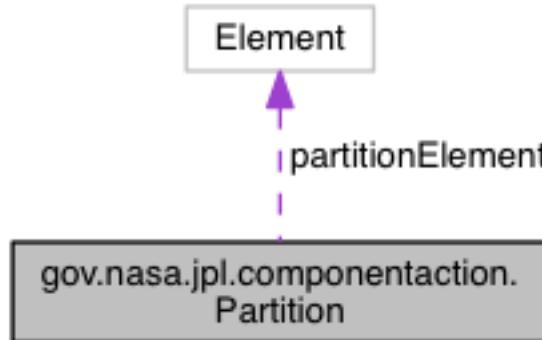
B `gov.nasa.jpl.componentaction.Utils.Pair< A, B >.second`

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

- `src/gov/nasa/jpl/componentaction/Utils.java`

gov.nasa.jpl.componentaction.Partition Class Reference

Collaboration diagram for gov.nasa.jpl.componentaction.Partition:



Public Member Functions

- `Partition (Element partitionElement)`
- `String getName ()`
- `String getDataName (Element dataElement)`
- `String getDataType (Element dataElement)`
- `void print ()`

Private Attributes

- `Element partitionElement`

Detailed Description

Creates the partition object.

Constructor & Destructor Documentation

gov.nasa.jpl.componentaction.Partition.Partition (Element *partitionElement*)

Assigns the input `partitionElement` to the instance variable `partitionElement`.

Parameters:

<code>partitionElement</code>	
-------------------------------	--

Member Function Documentation

String gov.nasa.jpl.componentaction.Partition.getDataName (Element *dataElement*)

Returns the third word in the name of the `dataElement` if the name was divided by spaces.

Parameters:

<code>dataElement</code>	
--------------------------	--

Returns:

the name of the dataElement

String gov.nasa.jpl.componentaction.Partition.getDataType (Element *dataElement*)

Returns the third word in the type of the dataElement if the type was divided by spaces.

Parameters:

dataElement

Returns:

the type of the dataElement

String gov.nasa.jpl.componentaction.Partition.getName ()

Returns the name of the partitionElement.

Returns:

name string

Here is the caller graph for this function:



void gov.nasa.jpl.componentaction.Partition.print ()

Prints the partition name to Java standard out and the MagicDraw console.

Here is the call graph for this function:



Member Data Documentation

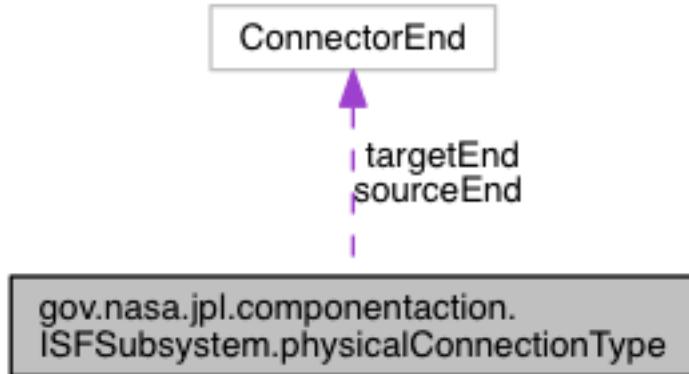
Element `gov.nasa.jpl.componentaction.Partition.partitionElement` [private]

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

- `src/gov/nasa/jpl/componentaction/Partition.java`

gov.nasa.jpl.componentaction.ISFSubsystem.physicalConnectionType Class Reference

Collaboration diagram for gov.nasa.jpl.componentaction.ISFSubsystem.physicalConnectionType:



Public Member Functions

- **physicalConnectionType** (ConnectorEnd **sourceEnd**, ConnectorEnd **targetEnd**, Integer **source_index**, Integer **target_index**)
- **physicalConnectionType** (ConnectorEnd **sourceEnd**, ConnectorEnd **targetEnd**, String **sourceName**, String **targetName**, Integer **source_index**, Integer **target_index**)
- String **toString** ()
- int **hashCode** ()
- boolean **equals** (Object obj)
- int **getSourcePortIndex** ()
- int **getTargetPortIndex** ()
- String **getSource** ()
- String **getTarget** ()
- String **getSourcePortName** ()
- String **getTargetPortName** ()
- String **getSourcePortType** ()
- String **getTargetPortType** ()

Package Attributes

- String **source**
- String **target**
- String **sourcePortName**
- String **targetPortName**
- String **sourcePortType**
- String **targetPortType**
- String **targetRoleParentName**
- String **sourceRoleParentName**
- ConnectorEnd **sourceEnd**
- ConnectorEnd **targetEnd**
- Integer **source_index**
- Integer **target_index**

Detailed Description

The **physicalConnectionType** is used to describe connections. Connector objects can not be re-defined through the API, so when we find a source port with it's final target port, we make one of these objects to describe the connection. This includes the names of the ports, the type, the parent name, the multiplicities, and the actual connectorEnd.

Constructor & Destructor Documentation

```
gov.nasa.jpl.componentaction.ISFSubsystem.physicalConnectionType.physicalConnectionType  
(ConnectorEnd sourceEnd, ConnectorEnd targetEnd, Integer source_index, Integer  
target_index)
```

```
gov.nasa.jpl.componentaction.ISFSubsystem.physicalConnectionType.physicalConnectionType  
(ConnectorEnd sourceEnd, ConnectorEnd targetEnd, String sourceName, String targetName,  
Integer source_index, Integer target_index)
```

Member Function Documentation

```
boolean gov.nasa.jpl.componentaction.ISFSubsystem.physicalConnectionType.equals (Object  
obj)
```

```
String gov.nasa.jpl.componentaction.ISFSubsystem.physicalConnectionType.getSource ()
```

```
int gov.nasa.jpl.componentaction.ISFSubsystem.physicalConnectionType.getSourcePortIndex ()
```

```
String gov.nasa.jpl.componentaction.ISFSubsystem.physicalConnectionType.getSourcePortName  
()
```

```
String gov.nasa.jpl.componentaction.ISFSubsystem.physicalConnectionType.getSourcePortType  
()
```

```
String gov.nasa.jpl.componentaction.ISFSubsystem.physicalConnectionType.getTarget ()
```

```
int gov.nasa.jpl.componentaction.ISFSubsystem.physicalConnectionType.getTargetPortIndex ()
```

```
String gov.nasa.jpl.componentaction.ISFSubsystem.physicalConnectionType.getTargetPortName  
()
```

```
String gov.nasa.jpl.componentaction.ISFSubsystem.physicalConnectionType.getTargetPortType  
()
```

```
int gov.nasa.jpl.componentaction.ISFSubsystem.physicalConnectionType.hashCode ()
```

```
String gov.nasa.jpl.componentaction.ISFSubsystem.physicalConnectionType.toString ()
```

Member Data Documentation

String gov.nasa.jpl.componentaction.ISFS subsystem.physicalConnectionType.source [package]

Integer

gov.nasa.jpl.componentaction.ISFS subsystem.physicalConnectionType.source_index [package]

ConnectorEnd

gov.nasa.jpl.componentaction.ISFS subsystem.physicalConnectionType.sourceEnd [package]

String

gov.nasa.jpl.componentaction.ISFS subsystem.physicalConnectionType.sourcePortName [package]

String

gov.nasa.jpl.componentaction.ISFS subsystem.physicalConnectionType.sourcePortType [package]

String

gov.nasa.jpl.componentaction.ISFS subsystem.physicalConnectionType.sourceRoleParentName [package]

String gov.nasa.jpl.componentaction.ISFS subsystem.physicalConnectionType.target [package]

Integer

gov.nasa.jpl.componentaction.ISFS subsystem.physicalConnectionType.target_index [package]

ConnectorEnd

gov.nasa.jpl.componentaction.ISFS subsystem.physicalConnectionType.targetEnd [package]

String

gov.nasa.jpl.componentaction.ISFS subsystem.physicalConnectionType.targetPortName [package]

String

gov.nasa.jpl.componentaction.ISFS subsystem.physicalConnectionType.targetPortType [package]

String

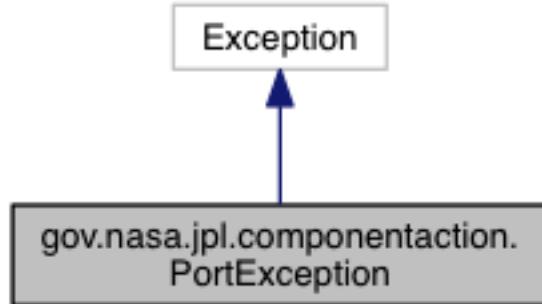
gov.nasa.jpl.componentaction.ISFS subsystem.physicalConnectionType.targetRoleParentName [package]

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

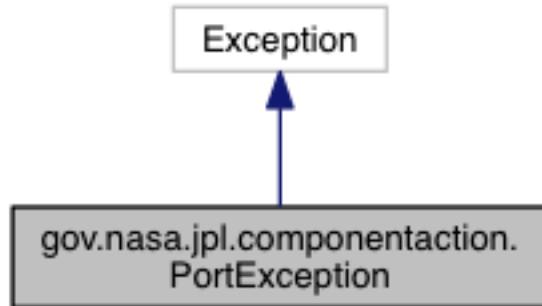
- [src/gov/nasa/jpl/componentaction/ISFS subsystem.java](#)

gov.nasa.jpl.componentaction.PortException Class Reference

Inheritance diagram for gov.nasa.jpl.componentaction.PortException:



Collaboration diagram for gov.nasa.jpl.componentaction.PortException:



Public Member Functions

- `PortException (String message)`

Static Private Attributes

- static final long `serialVersionUID = 1L`

Detailed Description

Port Exception shell class. This just serves as a wrapper that can be used to differentiate between where an error happens.

Constructor & Destructor Documentation

`gov.nasa.jpl.componentaction.PortException.PortException (String message)`

Member Data Documentation

```
final long gov.nasa.jpl.componentaction.PortException.serialVersionUID = 1L [static],  
[private]
```

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

- src/gov/nasa/jpl/componentaction/**PortException.java**

gov.nasa.jpl.componentaction.ISFComponent.portObject Class Reference

Public Member Functions

- int **getIndex** ()
- String **getStereotype** ()
- String **getDatatype** ()
- String **getDatatypeNamespace** ()
- String **getRole** ()
- String **getPriority** ()
- String **getFull** ()

Package Functions

- **portObject** ()

Package Attributes

- int **index**
- String **stereotype**
- String **datatype**
- String **datatypeNamespace**
- String **role**
- String **priority**
- String **full**

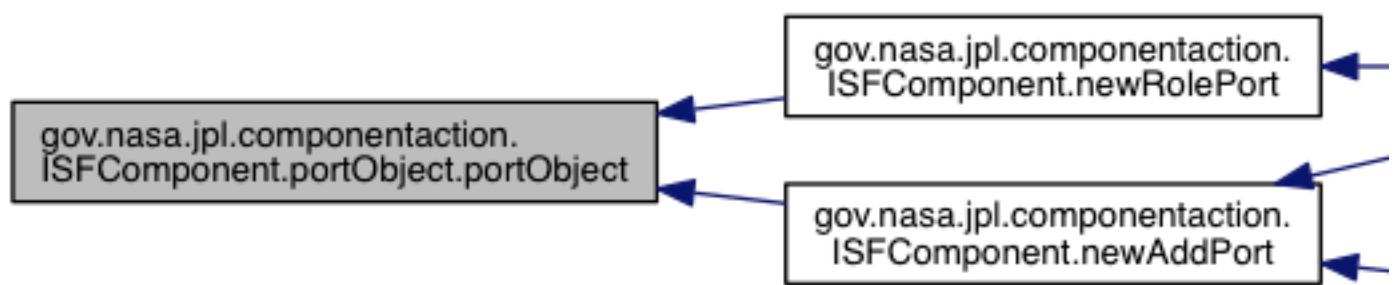
Detailed Description

Used to process ports in the component phase.

Constructor & Destructor Documentation

gov.nasa.jpl.componentaction.ISFComponent.portObject.portObject () [package]

Here is the caller graph for this function:



Member Function Documentation

`String gov.nasa.jpl.componentaction.ISFComponent.portObject.getDatatype ()`

`String gov.nasa.jpl.componentaction.ISFComponent.portObject.getDatatypeNamespace ()`

`String gov.nasa.jpl.componentaction.ISFComponent.portObject.getFull ()`

`int gov.nasa.jpl.componentaction.ISFComponent.portObject.getIndex ()`

`String gov.nasa.jpl.componentaction.ISFComponent.portObject.getPriority ()`

`String gov.nasa.jpl.componentaction.ISFComponent.portObject.getRole ()`

`String gov.nasa.jpl.componentaction.ISFComponent.portObject.getStereotype ()`

Here is the caller graph for this function:



Member Data Documentation

`String gov.nasa.jpl.componentaction.ISFComponent.portObject.datatype [package]`

`String gov.nasa.jpl.componentaction.ISFComponent.portObject.datatypeNamespace [package]`

`String gov.nasa.jpl.componentaction.ISFComponent.portObject.full [package]`

`int gov.nasa.jpl.componentaction.ISFComponent.portObject.index [package]`

`String gov.nasa.jpl.componentaction.ISFComponent.portObject.priority [package]`

`String gov.nasa.jpl.componentaction.ISFComponent.portObject.role [package]`

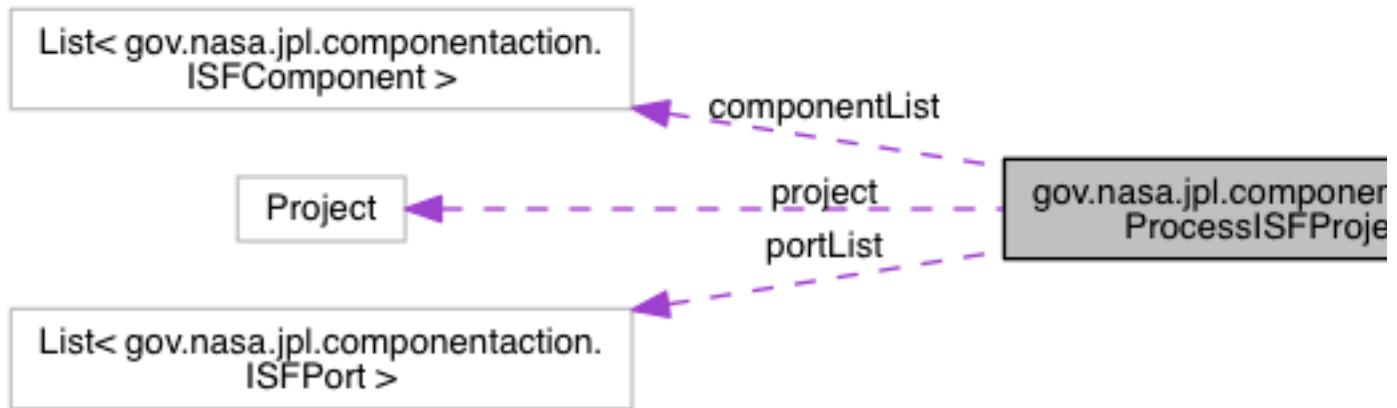
`String gov.nasa.jpl.componentaction.ISFComponent.portObject.stereotype [package]`

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

- `src/gov/nasa/jpl/componentaction/ISFComponent.java`

gov.nasa.jpl.componentaction.ProcessISFProject Class Reference

Collaboration diagram for gov.nasa.jpl.componentaction.ProcessISFProject:



Static Public Member Functions

- static void **process** (Project proj, File pluginDir) throws IOException
- static void **process** (Project proj, File pluginDir, String processPackage) throws IOException

Private Member Functions

- **ProcessISFProject ()**

Static Private Member Functions

- static void **processPorts** (List< Element > portElementList) throws PortException
- static void **processComponents** (List< Element > compElementList, String stereotype) throws PortException
- static List< Element > **getElementsOfStereotype** (String stereotype)
- static List< Element > **getElementsOfStereotypeAndPackage** (String stereotype, String processPackage)

Static Private Attributes

- static Project **project**
- static List< **ISFComponent** > **componentList**
- static List< **ISFPort** > **portList**

Detailed Description

This singleton class will process an ISF Project.

Author:

Garth Watney

Constructor & Destructor Documentation

`gov.nasa.jpl.componentaction.ProcessISFProject.ProcessISFProject () [private]`

Member Function Documentation

```
static List<Element> gov.nasa.jpl.componentaction.ProcessISFProject.getElementsOfStereotype  
(String stereotype) [static], [private]
```

Returns a list of elements with specified input stereotype string.

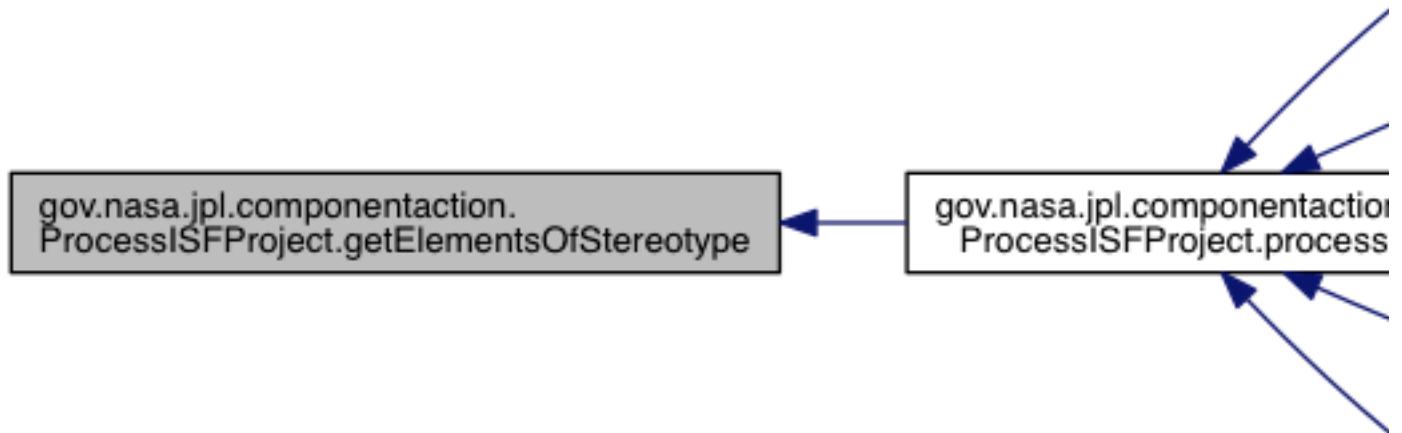
Parameters:

<i>stereotype</i>	stereotype string to look for
-------------------	-------------------------------

Returns:

list of Elements

Here is the caller graph for this function:



```
static List<Element>  
gov.nasa.jpl.componentaction.ProcessISFProject.getElementsOfStereotypeAndPackage (String  
stereotype, String processPackage) [static], [private]
```

Returns a list of elements with specified input stereotype string and within the processPackage.

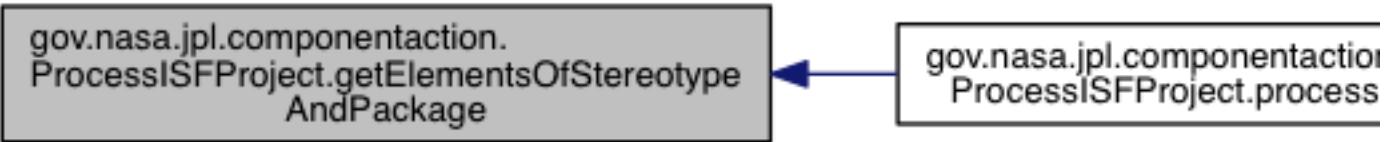
Parameters:

<i>stereotype</i>	stereotype string to look for
<i>processPackage</i>	package string to look for

Returns:

list of Elements

Here is the caller graph for this function:



```
static void gov.nasa.jpl.componentaction.ProcessISFProject.process (Project proj, File pluginDir) throws IOException [static]
```

Main function to process the project.

- Creates a directory to output XML files into if it does not already exist.
- Processes all the components in the model and generates an XML file for each of them.
- Processes all the ports in the model and generates an XML file to store all the port data.

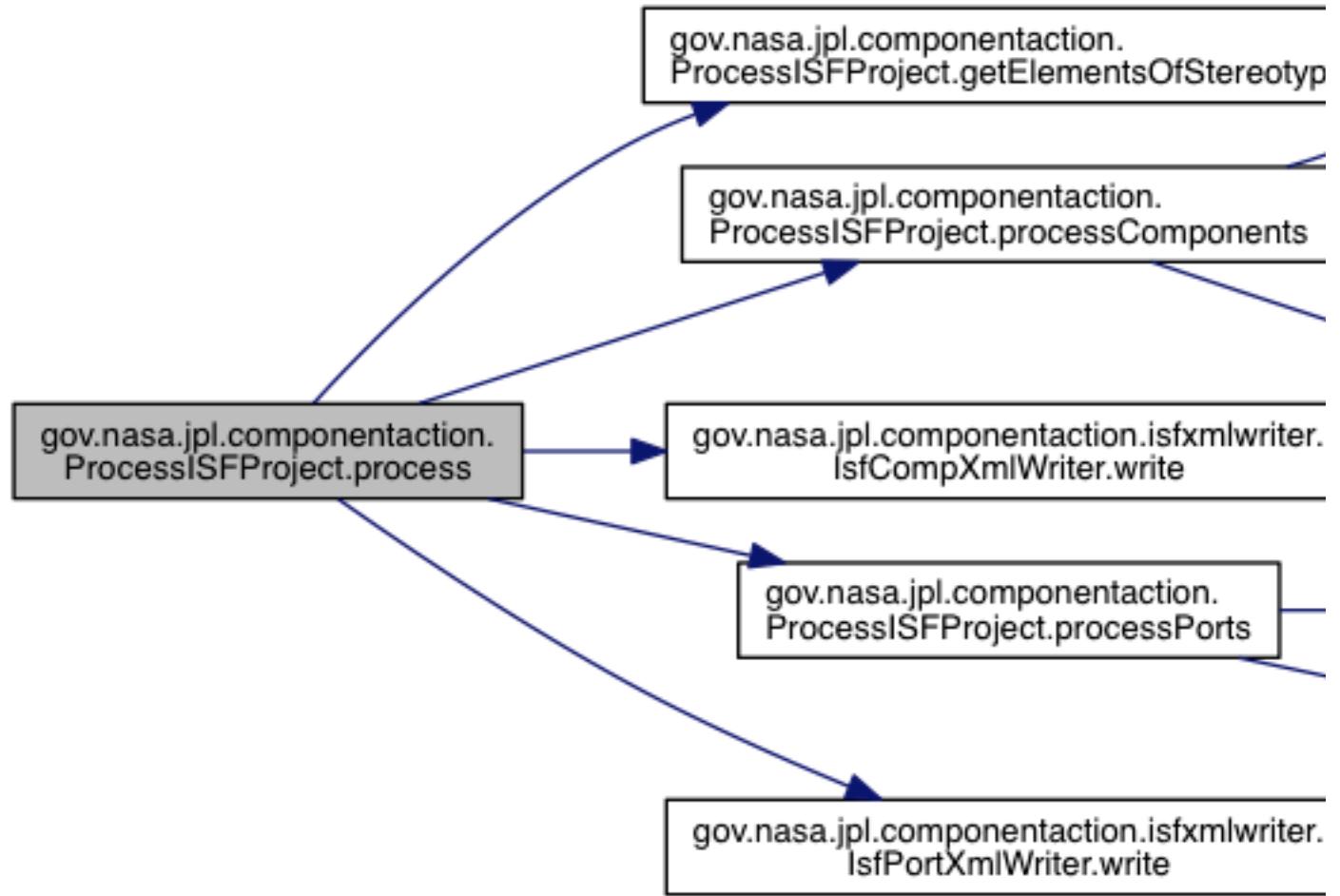
Parameters:

<i>proj</i>	Project object
<i>pluginDir</i>	Folder to create XML folder in

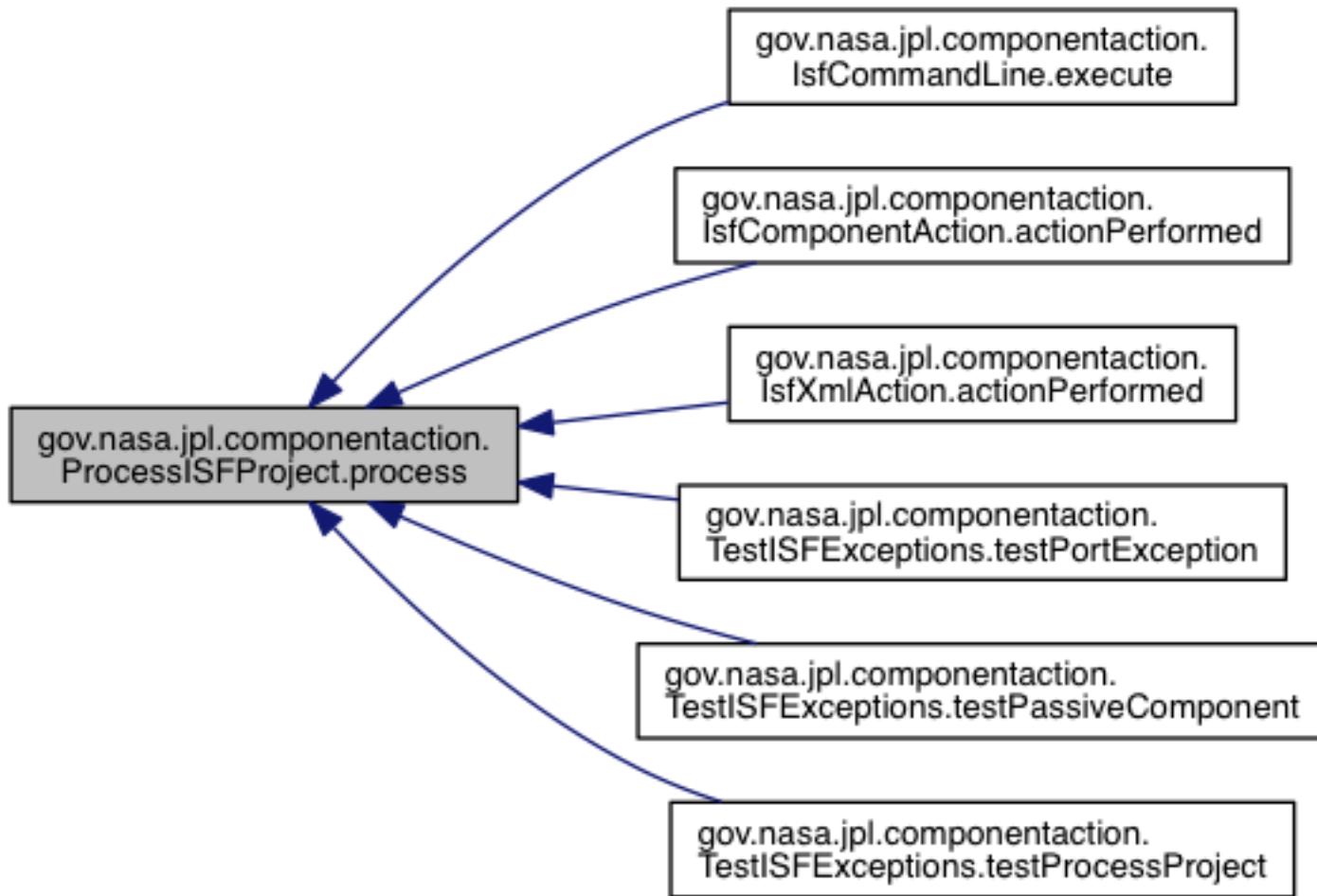
Exceptions:

<i>IOException</i>

Here is the call graph for this function:



Here is the caller graph for this function:



static void gov.nasa.jpl.componentaction.ProcessISFProject.process (Project *proj*, File *pluginDir*, String *processPackage*) throws IOException [static]

Function to process the project, except restricts component search to the processPackage.

- Creates a directory to output XML files into if it does not already exist.
- Processes all the components in the processPackage and generates an XML file for each of them.
- Processes all the ports in the processPackage and generates an XML file to store all the port data.

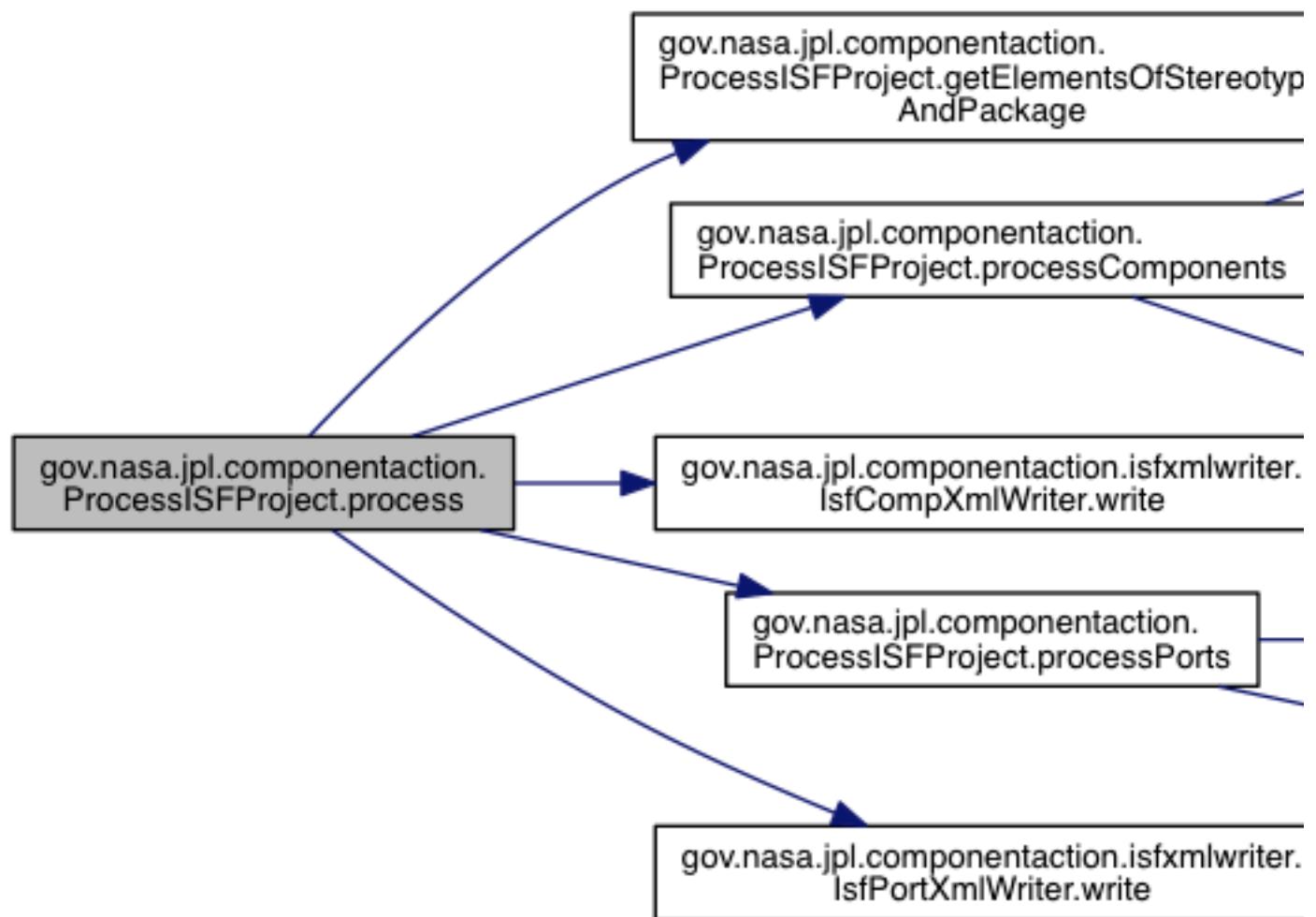
Parameters:

<i>proj</i>	Project object
<i>pluginDir</i>	Folder to create XML folder in
<i>processPackage</i>	String of the process package

Exceptions:

<i>IOException</i>

Here is the call graph for this function:



```
static void gov.nasa.jpl.componentaction.ProcessISFProject.processComponents (List< Element > compElementList, String stereotype) throws PortException [static], [private]
```

Creates **ISFComponent** objects for all components with associated ports and adds them to the componentList.

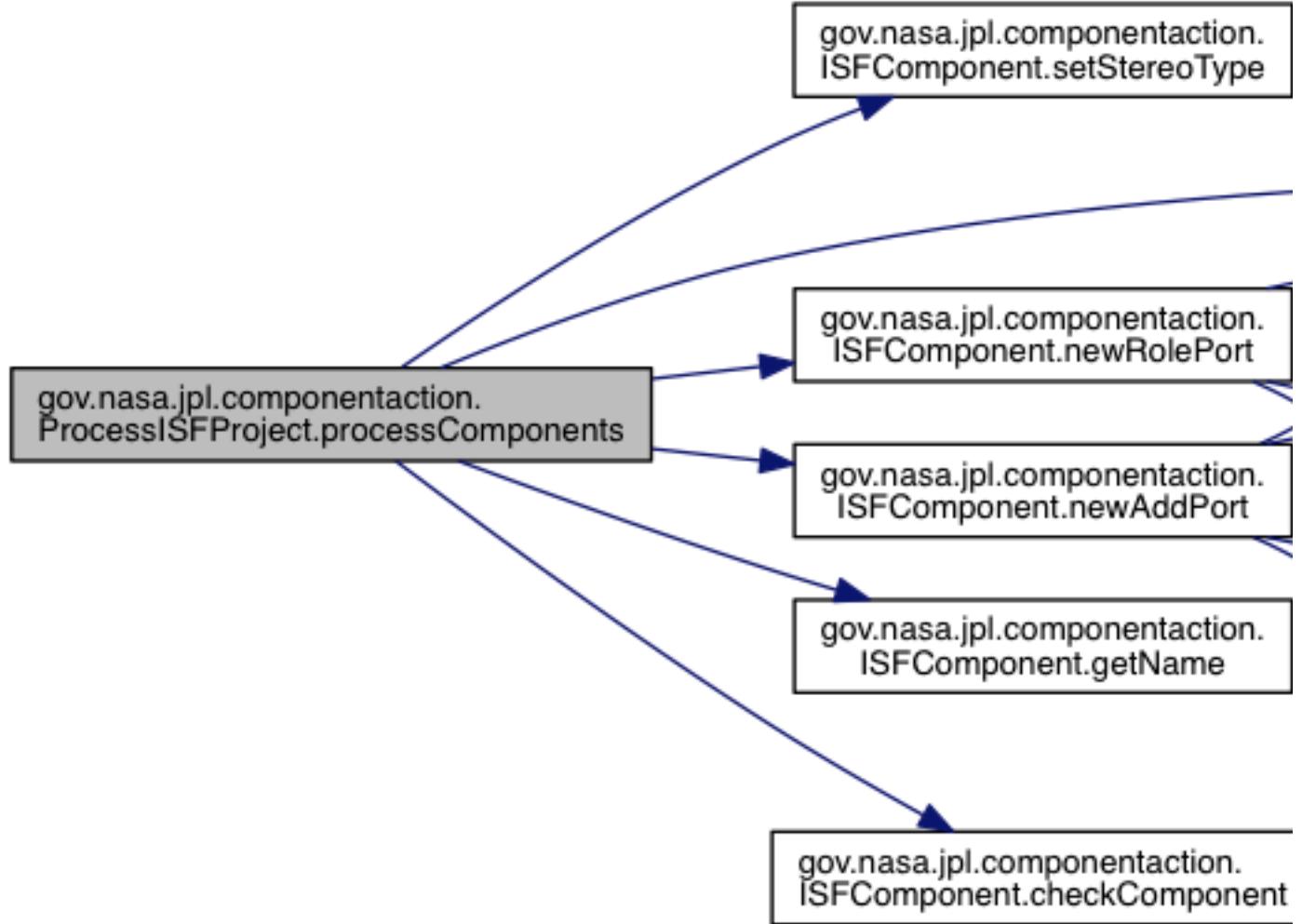
Parameters:

<i>compElementList</i>	
<i>stereotype</i>	

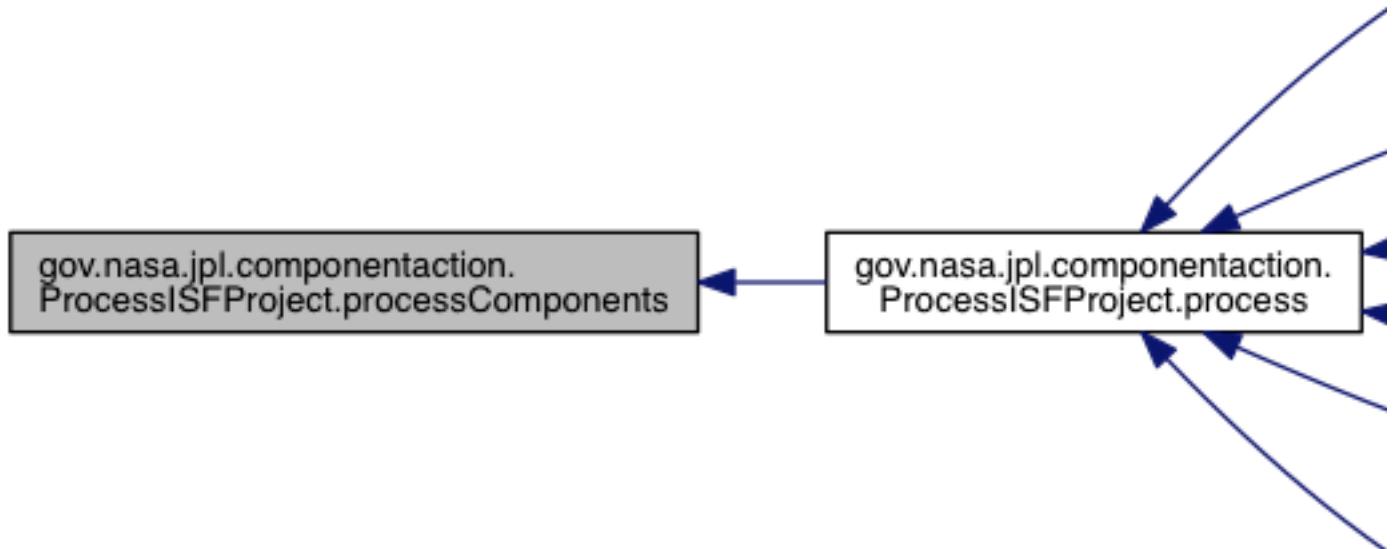
Exceptions:

<i>PortException</i>

Here is the call graph for this function:



Here is the caller graph for this function:



static void gov.nasa.jpl.componentaction.ProcessISFProject.processPorts (List< Element > portElementList) throws PortException [static], [private]

Converts ports to **ISFPort** objects with associated data and return elements and adds it to the portList.

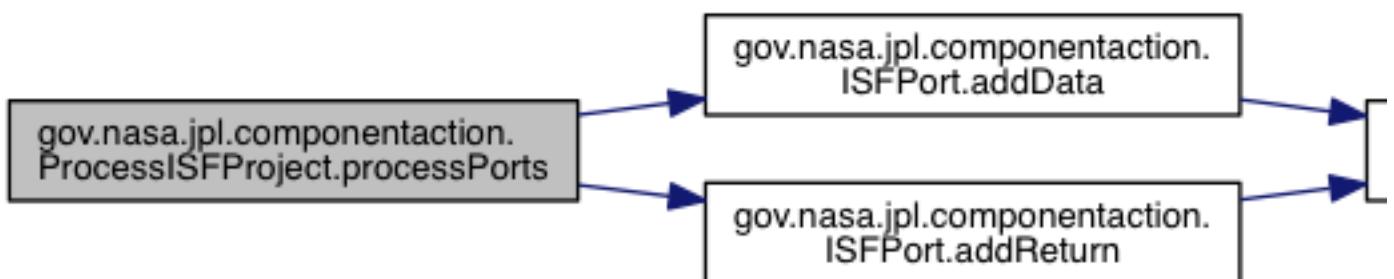
Parameters:

<code>portElementList</code>	List of ports
------------------------------	---------------

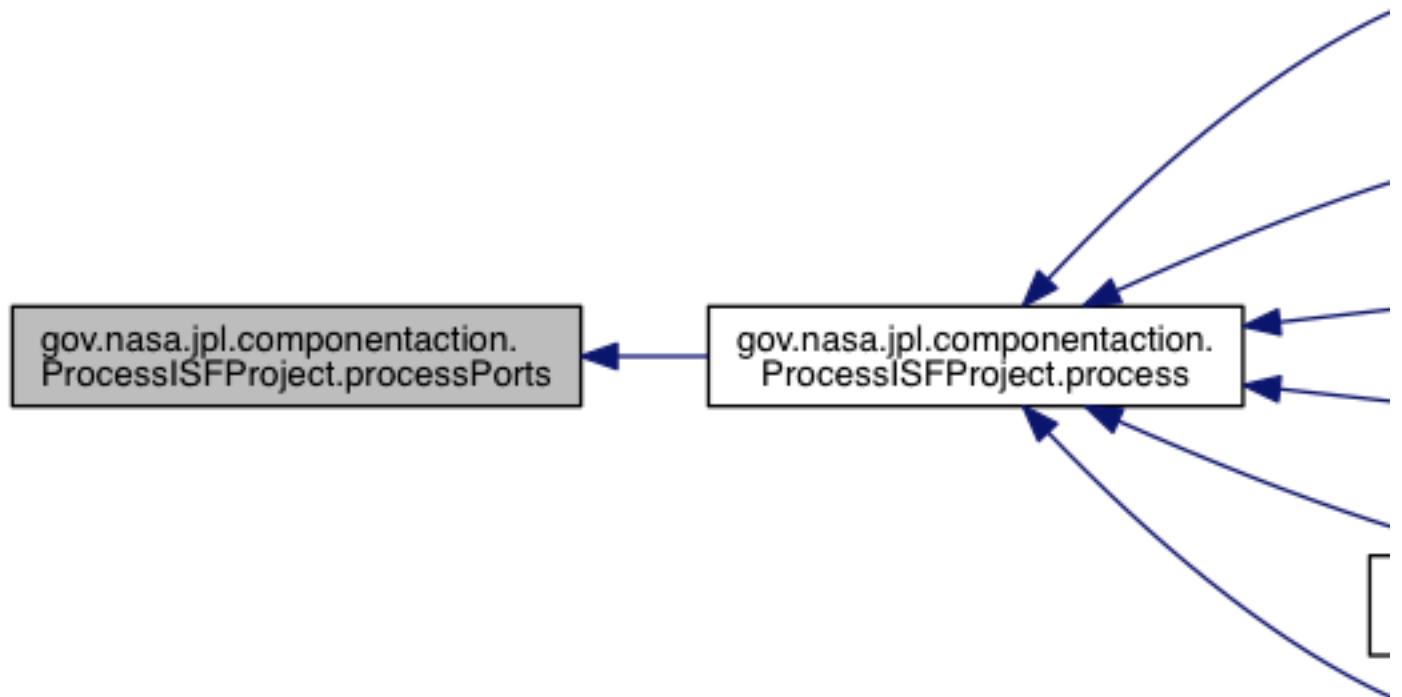
Exceptions:

<code>PortException</code>

Here is the call graph for this function:



Here is the caller graph for this function:



Member Data Documentation

`List<ISFComponent> gov.nasa.jpl.componentaction.ProcessISFProject.componentList[static], [private]`

`List<ISFPort> gov.nasa.jpl.componentaction.ProcessISFProject.portList[static], [private]`

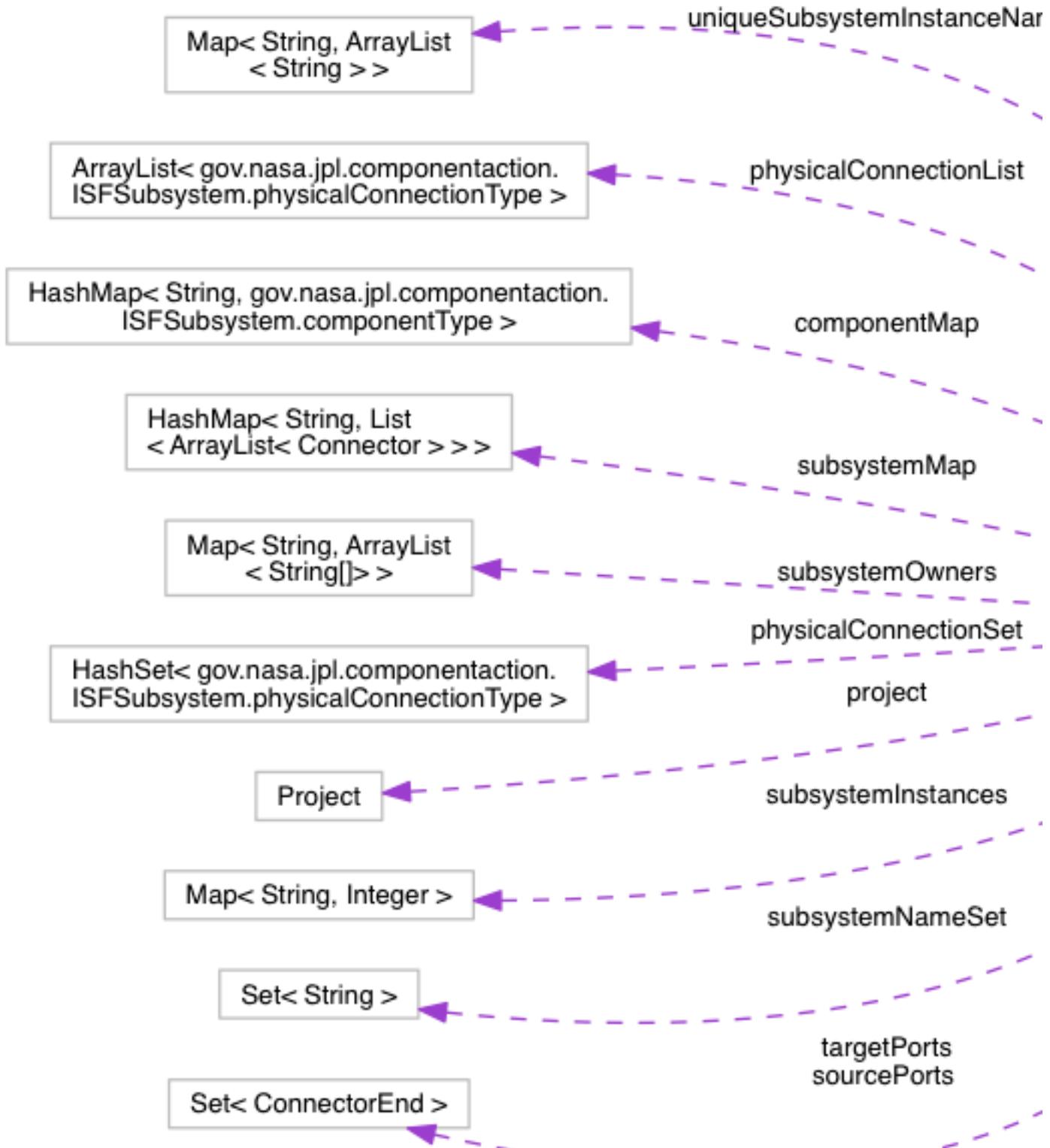
`Project gov.nasa.jpl.componentaction.ProcessISFProject.project[static], [private]`

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

- `src/gov/nasa/jpl/componentaction/ProcessISFProject.java`

gov.nasa.jpl.componentaction.ProcessISFTopology Class Reference

Collaboration diagram for gov.nasa.jpl.componentaction.ProcessISFTopology:



Static Public Member Functions

- static void **process** (Project proj, File pluginDir) throws IOException
- static int **generateNewMultiplicity** (Connector e, int currentMultiplicity, int previousConnectorTargetMult) throws ConnectorException

Static Public Attributes

- static HashSet< ISFSubsystem.physicalConnectionType > **physicalConnectionSet**
- static HashMap< String, ISFSubsystem.componentType > **componentMap**
- static ArrayList< ISFSubsystem.physicalConnectionType > **physicalConnectionList**
- static Map< String, ArrayList< String[] > > **subsystemOwners**
- static Map< String, Integer > **subsystemInstances**
- static Map< String,ArrayList< String > > **uniqueSubsystemInstanceNames**

Private Member Functions

- **ProcessISFTopology** ()

Static Private Member Functions

- static ArrayList< ISFSubsystem.physicalConnectionType > **indexChecking** (HashSet< ISFSubsystem.physicalConnectionType > pctSet) throws ConnectorException
- static void **checkPorts** () throws ConnectorException
- static List< Element > **processModel** (Element root) throws ConnectorException
- static boolean **isTopSubsystemNode** (String key)
- static void **processSubsystem** (Element subsystem) throws ConnectorException
- static void **createConnections** () throws ConnectorException

Static Private Attributes

- static Project **project**
- static Set< String > **subsystemNameSet**
- static HashMap< String, List< ArrayList< Connector > > > **subsystemMap**
- static Set< ConnectorEnd > **sourcePorts**
- static Set< ConnectorEnd > **targetPorts**

Detailed Description

This singleton class will process an ISF Project.

There are three main phases of the processing:

- PHASE ONE: Processing the model
 - A list of all subsystems is made.
 - A map of all the components is made.
 - A list of all the target ports is made.
 - A list of all the source ports is made.
 - A map of all the port ends with a list of associated connectors (one list going in, the other going out) is made.
- PHASE TWO: Creating connections
 - All full connections are found using each port in the source port list as a start point.
 - Further connections are found using the map of port ends with their associated connectors.

- Some connectors are ignored depending on the multiplicity of the previous source port. This multiplicity is recalculated for every connector.
- When reached a port that is not owned by a subsystem, the type is checked. If the source port and the target port type do not matched, that target port is ignored.
- This process goes on for each source port until either the target port is found, all the associated connectors have been done being searched, or a time out variable is reached.
- Each source and derived target connector ends are processed through the **ISFSubsystem** physicalConnectionType and added to a list.
-
- PHASE THREE: Checking ports and writing to XML
 - A map is created with the name of generic modules being associated with a list of the ports on that module.
 - This map is cross checked with all the instances of the modules. If something in the map does not exist within an instance, the user gets notified by a warning.
 - An **ISFSubsystem** topologyModel object is created using the pysicalConnectionType list and the map of all the components.
 - This object is passed to the IsfSubXmlWriter and is used to generate the flattened XML topology file.
-

Please note that if you have more than one subsystem to be processed, make sure one of them has a stereotype with the name "Top". This subsystem will be used for the name of the topology xml. If not, an exception will be thrown.

Exceptions can be generated at any time in this process.

Constructor & Destructor Documentation

gov.nasa.jpl.componentaction.ProcessISFTopology.ProcessISFTopology () [private]

Member Function Documentation

static void gov.nasa.jpl.componentaction.ProcessISFTopology.checkPorts () throws ConnectorException [static], [private]

PHASE THREE FUNCTION

Checks if ports on leaf modules have any connections. If they do not, a warning is brought up onto the console.

This function first generates a Map with a generic leaf module's name as a key to a list of the "qualified name" of each port on the module. The qualified name is used as an identifier for the generic module.

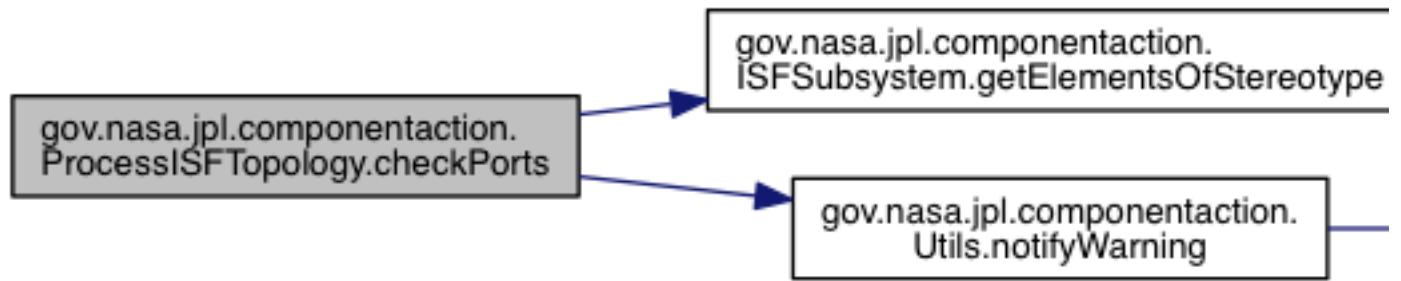
The function next iterates through all elements with the "PartProperty" stereotype to find "unique" leaf components. The function now uses the unique leaf components generic name to iterate through each of its' ports.

Each port is then checked to see if it exists as a key in the subsystemMap. If it doesn't exist, it means that this port has no connections and a warning flag is immediately notified.

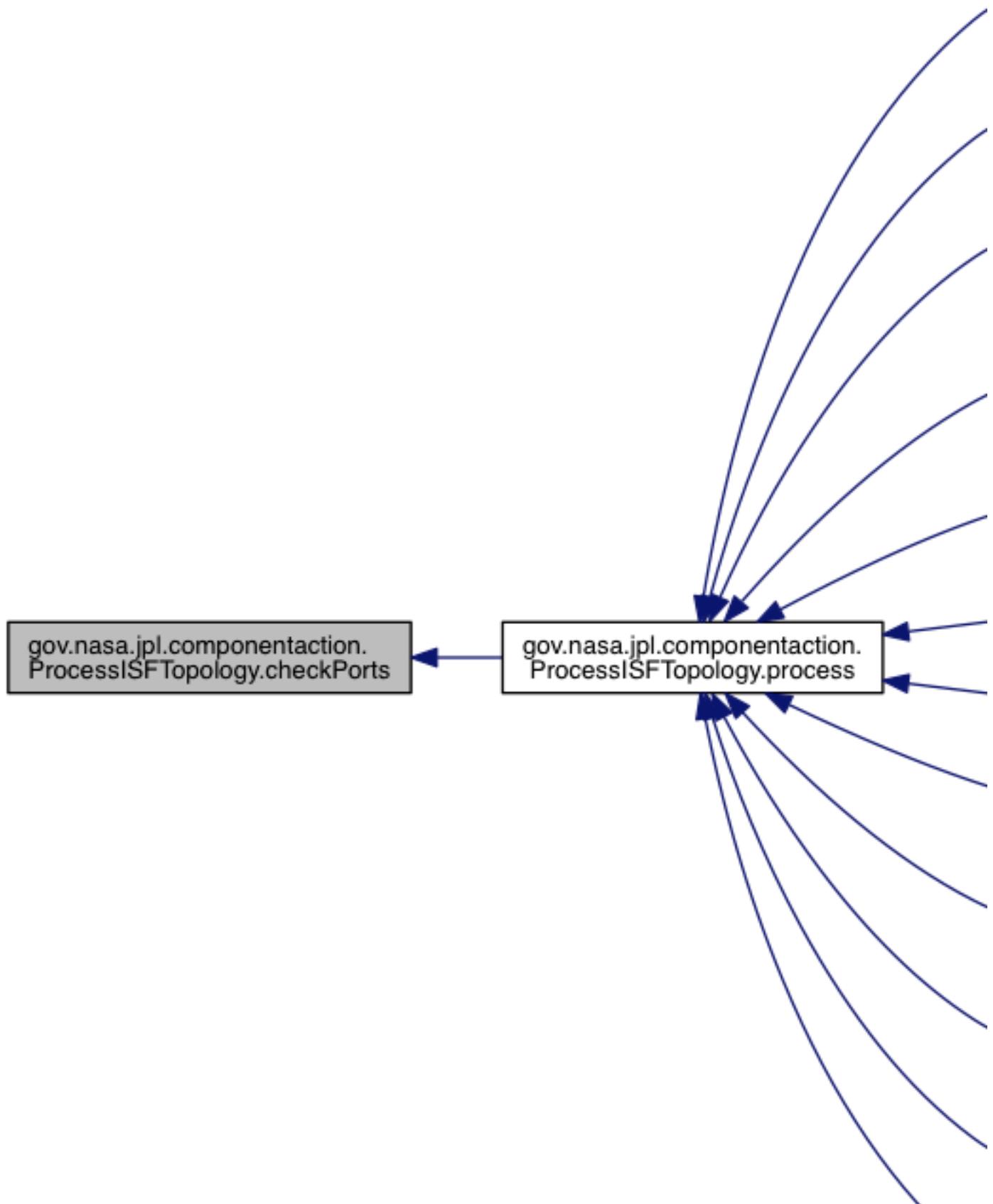
Exceptions:

ConnectorException	
---------------------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



```
static void gov.nasa.jpl.componentaction.ProcessISFTopology.createConnections () throws
ConnectorException [static], [private]
```

PHASE TWO FUNCTION

Iterates through the list of source ports and tries to find the final destination, while checking for errors.

For each source port, the program traces down its path by accessing the subsystemMap with the port's ID. All connections associated with the port are put onto a list, the multiplicity is reevaluated, and the program moves onto processing the next port on the list. Every time a new connector is accessed, the target is checked to see if it is a leaf target with the same type as the source. If it is, an **ISFSubsystem** physicallyConnectionType object is put onto a list.

Additionally, to support multiple instances of ibds, is a source instance and target instance name is created along side every connection. This name is the root modules name followed by each subsystem which the connection's path goes through. This name is used if the program detect that target or source subsystem is used more than once.

Exceptions are thrown in the instances where

- Source multiplicity is repeated by the same port onto multiple connectors.
- A source on a singular branch of travel (no other sources share the same path) has a target port at the end with a different type as the source port.

Exceptions:

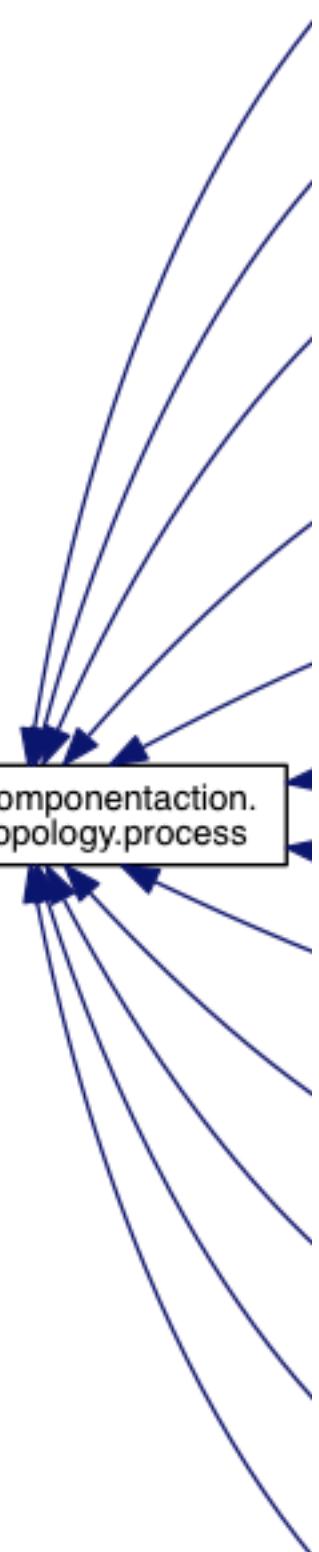
ConnectorException	
on	

Here is the call graph for this function:

Here is the caller graph for this function:

gov.nasa.jpl.componentaction.
ProcessISFTopology.createConnections

gov.nasa.jpl.componentaction.
ProcessISFTopology.process



```
static int gov.nasa.jpl.componentaction.ProcessISFTopology.generateNewMultiplicity (Connector
e, int currentMultiplicity, int previousConnectorTargetMult) throws
ConnectorException [static]
```

PHASE TWO FUNCTION

`generateMultiplicity` is used within `createConnections` to find the multiplicity of the next connector.

The function takes in three arguments: the current Connector, the multiplicity of the source of the connector (calculated from this function in the previous iteration of the loop) and a '`previousConnectorTargetMult`', which is an integer which indicates the 'inorder' position of the target of the Connector by keeping a track of how many connectors have accessed the same end.

Parameters:

<code>e</code>	Connector element
<code>currentMultiplicity</code>	The current multiplicity value of the connector
<code>previousConnectorTargetMult</code>	Multiplicity value based on the position of the connector in the port.

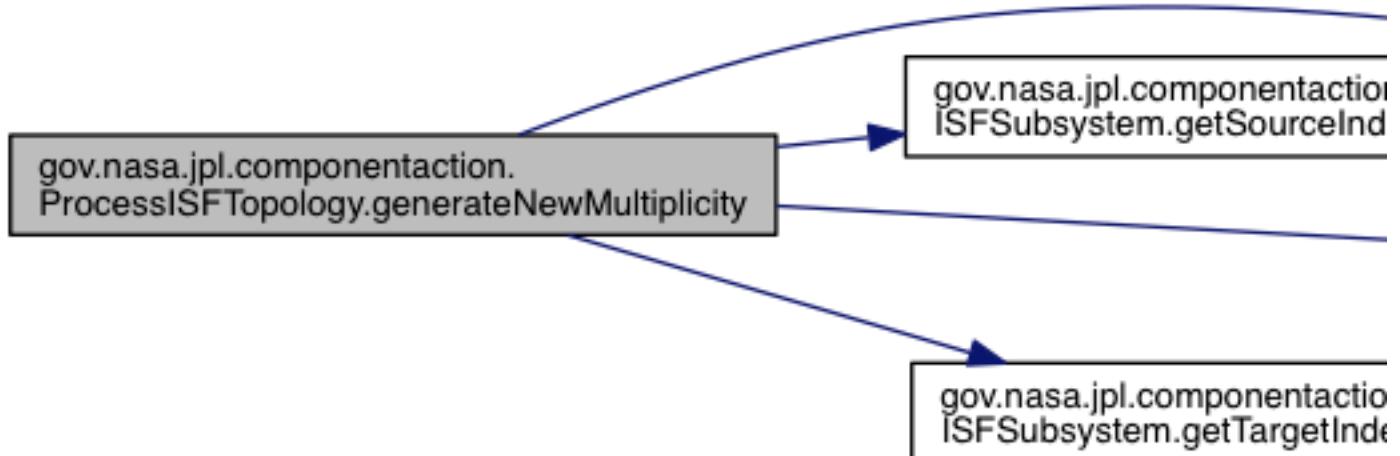
Returns:

New multiplicity value to be used in next iteration

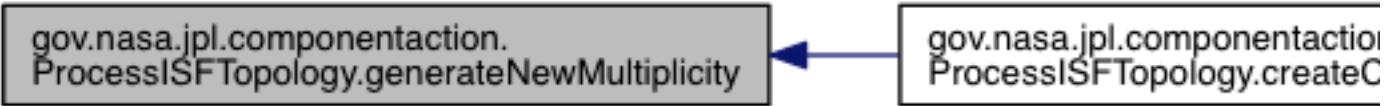
Exceptions:

<code>ConnectorException</code>	Only <code>ISFSubsystem.getConnEnd()</code> will raise the exception in this function.
---------------------------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



```

static ArrayList<ISFS subsystem.physicalConnectionType>
gov.nasa.jpl.componentaction.ProcessISFTopology.indexChecking (HashSet<
ISFS subsystem.physicalConnectionType > pctSet) throws ConnectorException [static],
[private]

```

PHASE THREE FUNCTION

This function serves two purposes. The first is to convert the physicalConnectionSet to a physicalConnecitonList. The second is to perform index checking and correction on ports of the cmdReg/cmd or com/cmdResponse type.

The index checking/correction is done so multiplicites do not need to be specified for cmd/cmdReg or com/cmdResponse (with port names of seqCmdStatus and seqCmdBuff) blocks. The latter pair has the additinal port name constraint because there are multiple other ports with the same cmdResponse type. Some of these port are not to be auto-indexed. This makes it easier for the user to add and remove these ports without needing to worry about if the indexes of the pairings match.

The program redefines the indexes in numerical order. Even if initial indexes are given, for the origin of Cmd and the target of CmdReg, they are overwritten. They target index of Cmd is preserved and also is written to the source of CmdReg.

METHODOLOGY

- Iterate through all items of the physicalConnectionType and add those who have types of Cmd/CmdReg or those who have type of CmdResponse/Com and names of seqCmdStatus/seqCmdBuff to a list in a map where the index is the instance name of the associated cmdDispatch.
- Iterate through all keys in the dictionary.
 - Iterate through each item in the list and find another item that has the opposite target and source components but is off the same pair.
 - If the associated item does not exist and if the search object is of the type of Cmd or Cmd reg, throw an error.
 - If the item is of the type Com/CmdResponse or there is an associated pair, add the item(s) to the output list.
 - Return a new list of physicalConnectionType objects with auto indexed IDS.
 - pctSet Set of Physical Connection Types
- **Returns:**
 - List of Physical Connection Types

Here is the call graph for this function:

Here is the caller graph for this function:

```
static boolean gov.nasa.jpl.componentaction.ProcessISFTopology.isTopSubsystemNode (String  
key) [static], [private]
```

Checks the subsystemInstances map (which is a key string mapped to a value of how many times this class is instantiated. Will return true if it is a singular system, false if the system isn't instantiated or the instances equal one.

Parameters:

key	
-----	--

Returns:

Here is the caller graph for this function:

```
static void gov.nasa.jpl.componentaction.ProcessISFTopology.process (Project proj, File pluginDir) throws IOException [static]
```

process handles all the phases of this methodology.

Parameters:

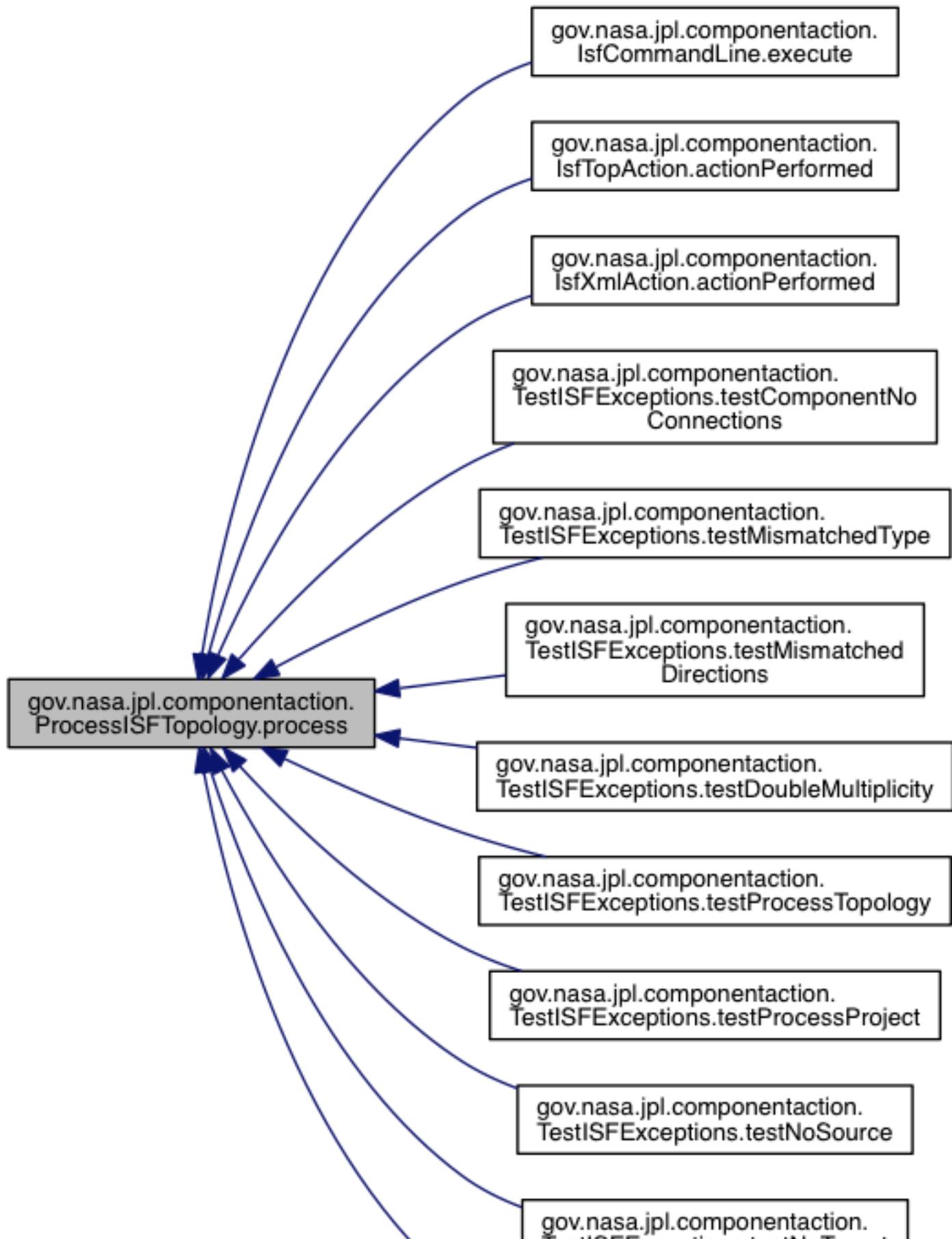
<i>proj</i>	Project object
<i>pluginDir</i>	Directory to generate documents into

Exceptions:

<i>IOException</i>	
--------------------	--

Here is the call graph for this function:

Here is the caller graph for this function:



```
static List<Element> gov.nasa.jpl.componentaction.ProcessISFTopology.processModel (Element root) throws ConnectorException [static], [private]
```

PHASE ONE FUNCTION

processModel has two main functions: to create a list of subsystems and to create a componentMap.

All the elements in the module are iterated through using the Element root.

If the element type is Subsystem, the element is added to a list to be returned.

If the element is an instance of a subsystem, the subsystemOwners map is updated, where each key (name of subsystem) is associated with an list of two string arrays. The first String corresponds to the instance name and the second corresponds to the name of the owner of the subsystem.

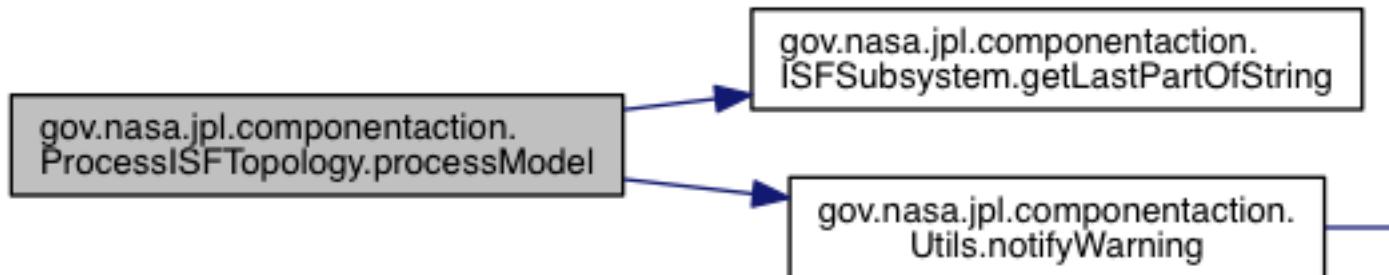
Parameters:

<i>root</i>	Root of the system
-------------	--------------------

Returns:

List of subsystem elements

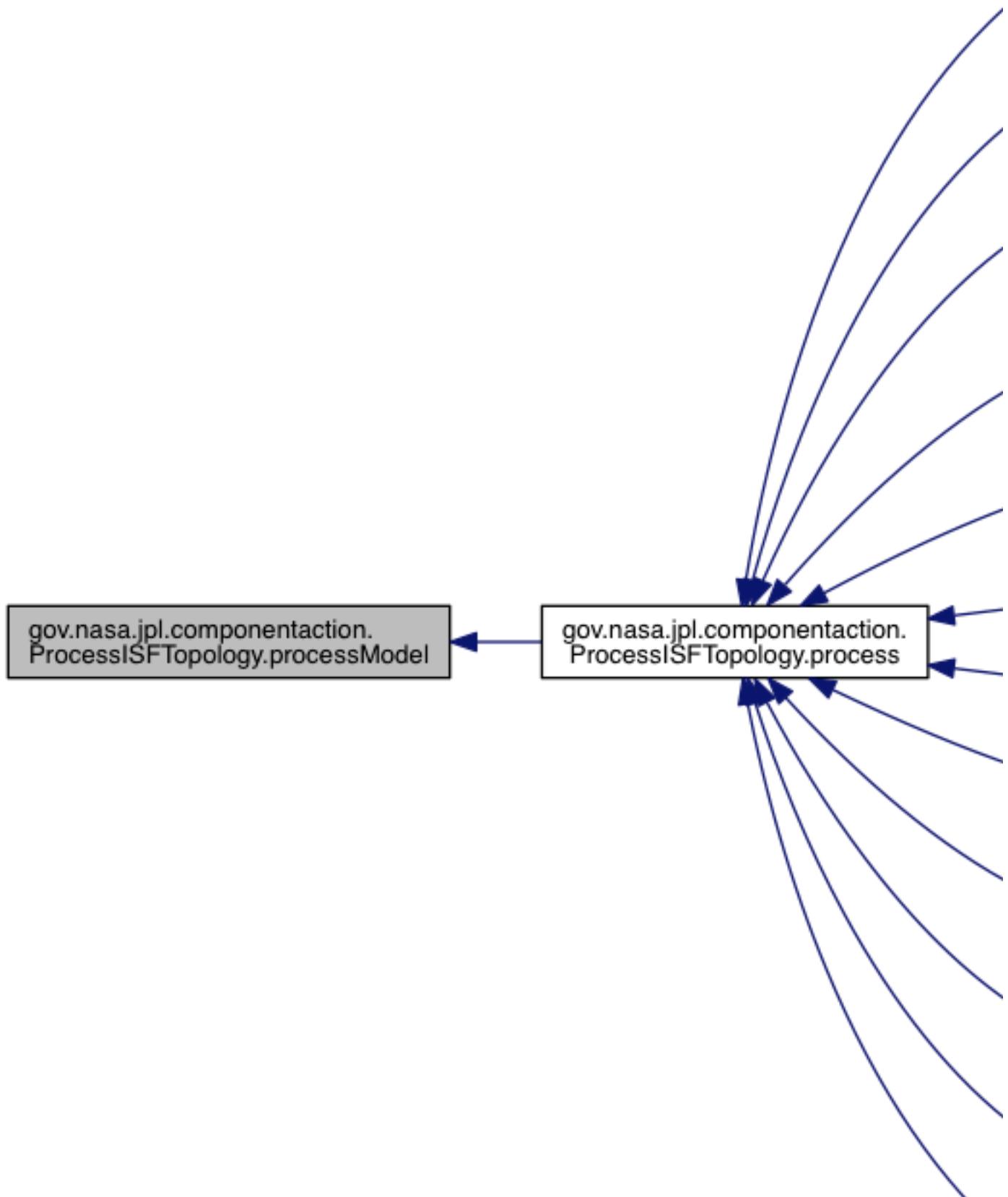
Here is the call graph for this function:



Here is the caller graph for this function:

gov.nasa.jpl.componentaction.
ProcessISFTopology.processModel

gov.nasa.jpl.componentaction.
ProcessISFTopology.process



```
static void gov.nasa.jpl.componentaction.ProcessISFTopology.processSubsystem (Element subsystem) throws ConnectorException [static], [private]
```

PHASE ONE FUNCTION

processSubsystem populates both the subsystemMap as well as the sourcePorts/targetPorts list.

First, the function's name is used with the subsystemOwners map to create a list tree. This list tree is then used to generate all the different names needed to clarify

The function iterates through all Connectors associated with the subsystem. The subsystemMap, sourcePorts, and targetPorts are populated through this iteration.

The subsystemMap is a Map where the key is the ID of each port with an associated list of input and output connectors.

ConnectorEnds of leaf components are also pushed onto sourcePorts and targetPorts lists, respective of what direction they accept data.

Using the list of all instances, each internal module (bbd) is added as an **ISFSubsystem** componentType to the component map, with the new name of the module

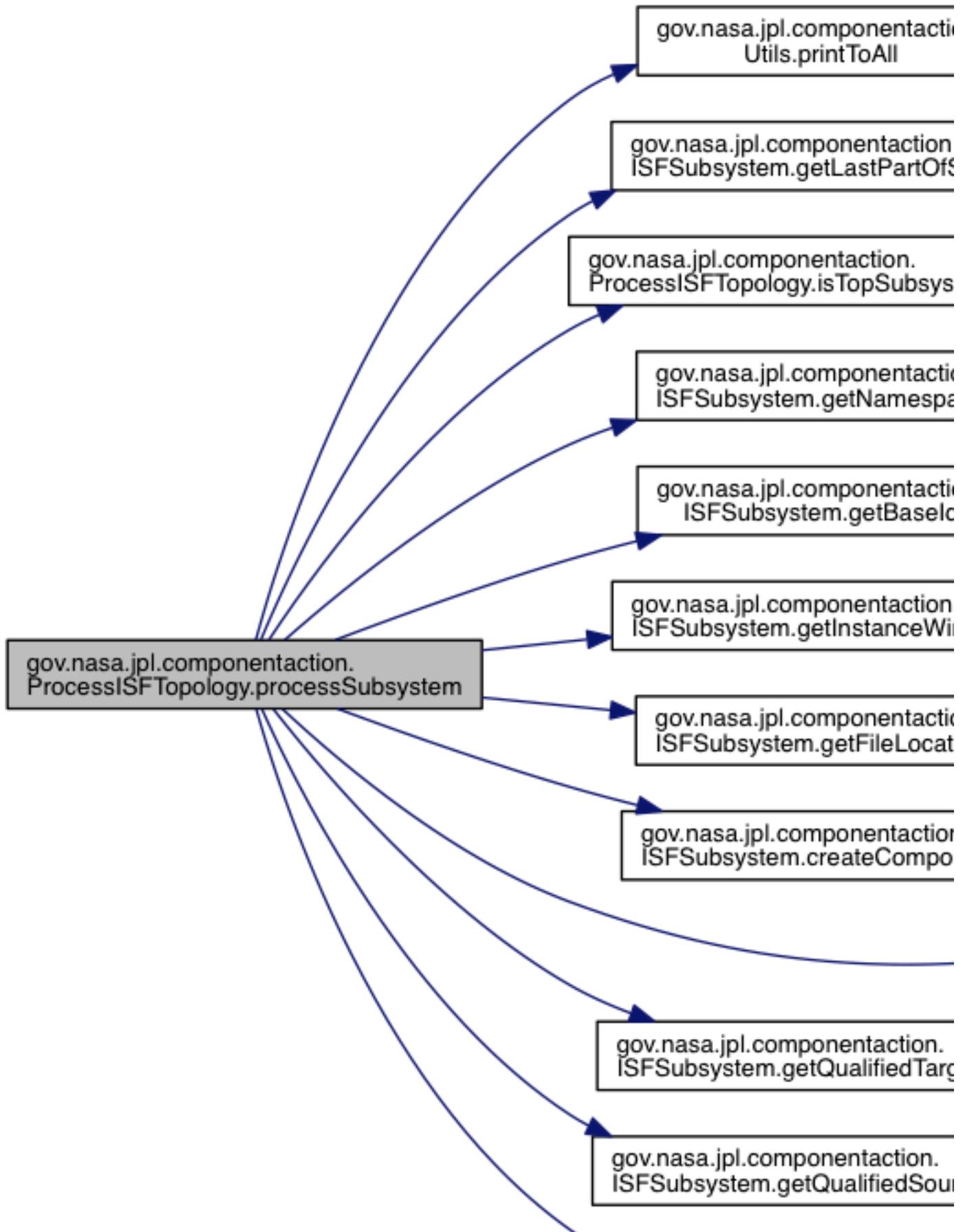
Parameters:

<i>subsystem</i>	Subsystem Element
------------------	-------------------

Exceptions:

<i>ConnectorException</i>	
---------------------------	--

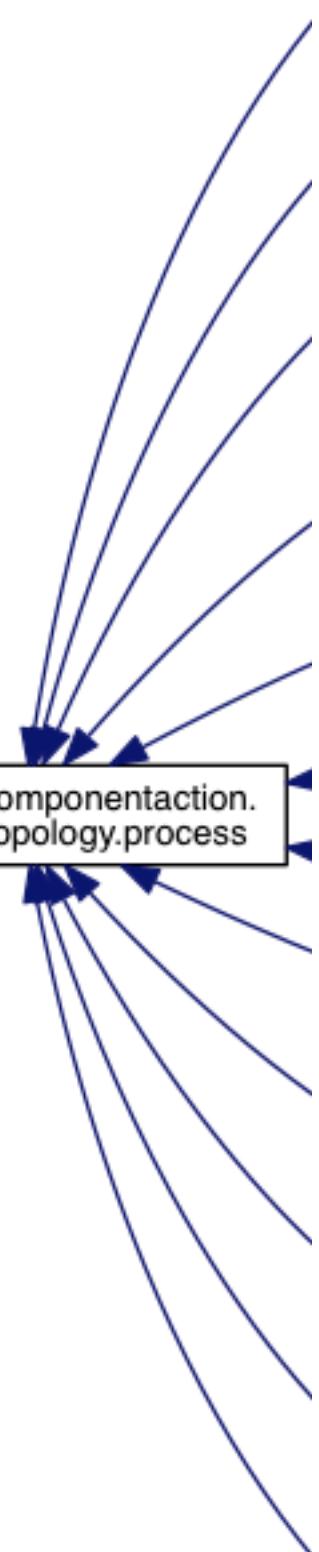
Here is the call graph for this function:



Here is the caller graph for this function:

gov.nasa.jpl.componentaction.
ProcessISFTopology.processSubsystem

gov.nasa.jpl.componentaction.
ProcessISFTopology.process



Member Data Documentation

HashMap<String , ISFSubsystem.componentType>
gov.nasa.jpl.componentaction.ProcessISFTopology.componentMap [static]

ArrayList<ISFSubsystem.physicalConnectionType>
gov.nasa.jpl.componentaction.ProcessISFTopology.physicalConnectionList [static]

HashSet<ISFSubsystem.physicalConnectionType>
gov.nasa.jpl.componentaction.ProcessISFTopology.physicalConnectionSet [static]

Project gov.nasa.jpl.componentaction.ProcessISFTopology.project [static], [private]

Set<ConnectorEnd> gov.nasa.jpl.componentaction.ProcessISFTopology.sourcePorts [static],
[private]

Map<String , Integer>
gov.nasa.jpl.componentaction.ProcessISFTopology.subsystemInstances [static]

HashMap<String , List<ArrayList<Connector> >>
gov.nasa.jpl.componentaction.ProcessISFTopology.subsystemMap [static], [private]

Set<String> gov.nasa.jpl.componentaction.ProcessISFTopology.subsystemNameSet [static],
[private]

Map<String , ArrayList<String[]> >
gov.nasa.jpl.componentaction.ProcessISFTopology.subsystemOwners [static]

Set<ConnectorEnd> gov.nasa.jpl.componentaction.ProcessISFTopology.targetPorts [static],
[private]

Map<String ,ArrayList<String> >
gov.nasa.jpl.componentaction.ProcessISFTopology.uniqueSubsystemInstanceNames [static]

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

- src/gov/nasa/jpl/componentaction/ProcessISFTopology.java

gov.nasa.jpl.componentaction.ISFComponent.referencePort Class Reference

Public Member Functions

- String **getReference ()**

Package Functions

- **referencePort ()**

Package Attributes

- String **Reference**

Detailed Description

A base class for all Ports that do not specify a data type but have a reference to some existing xml data structure

Constructor & Destructor Documentation

gov.nasa.jpl.componentaction.ISFComponent.referencePort () [package]

Member Function Documentation

String gov.nasa.jpl.componentaction.ISFComponent.referencePort.getReference ()

Member Data Documentation

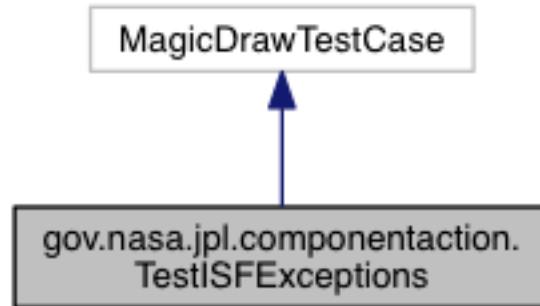
String gov.nasa.jpl.componentaction.ISFComponent.referencePort.Reference [package]

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

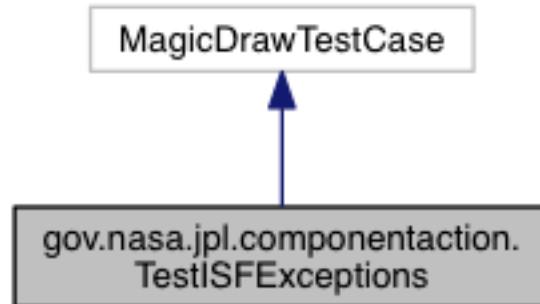
- src/gov/nasa/jpl/componentaction/**ISFComponent.java**

gov.nasa.jpl.componentaction.TestISFExceptions Class Reference

Inheritance diagram for gov.nasa.jpl.componentaction.TestISFExceptions:



Collaboration diagram for gov.nasa.jpl.componentaction.TestISFExceptions:



Public Member Functions

- **TestISFExceptions** (String testMethodToRun, String testName)
- **TestISFExceptions** (String testMethodToRun)
- void **testComponentNoConnections** () throws IOException
- void **testPortException** () throws IOException
- void **testMismatchedType** () throws IOException
- void **testMismatchedDirections** () throws IOException
- void **testDoubleMultiplicity** () throws IOException
- void **testPassiveComponent** () throws IOException
- ISFComponent **getCompForTesting** (String projectName, String compType)
- void **testActiveComponent** ()
- void **testQueuedComponentAsync** () throws IOException
- void **testQueuedComponentSync** () throws IOException
- void **testProcessTopology** () throws IOException
- void **testAbout** ()
- void **testProcessProject** () throws IOException
- void **testNoSource** () throws IOException
- void **testNoTarget** () throws IOException
- void **testMultipleSubsystems** () throws IOException
- void **testCommandLine** ()

Static Public Member Functions

- static void **main** (String[] args)

- static Test suite ()

Protected Member Functions

- void setUpTest () throws Exception
- void tearDownTest () throws Exception

Static Private Member Functions

- static List< Element > getElementsOfStereotype (String stereotype, Project project)

Private Attributes

- String pluginDir
-

Detailed Description

JUnit test suite to check that exception handling works.

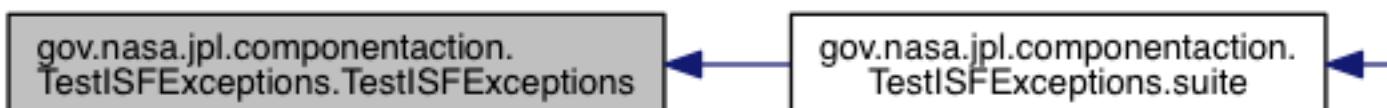
Most tests involve three parts:

- Load the Magic Draw file.
 - Run the plugin function that will be tested (IE, process from **ProcessISFTopology** will only test the processing of the topology models, and so on).
 - Check if an exception is thrown with the correct error message, or if a pass test is being run, check if the appropriate files have been generated.
 -
 -
-

Constructor & Destructor Documentation

gov.nasa.jpl.componentaction.TestISFExceptions.TestISFExceptions (String testMethodToRun, String testName)

Here is the caller graph for this function:



gov.nasa.jpl.componentaction.TestISFExceptions.TestISFExceptions (String testMethodToRun)

Member Function Documentation

ISFComponent gov.nasa.jpl.componentaction.TestISFExceptions.getCompForTesting (String projectName, String compType)

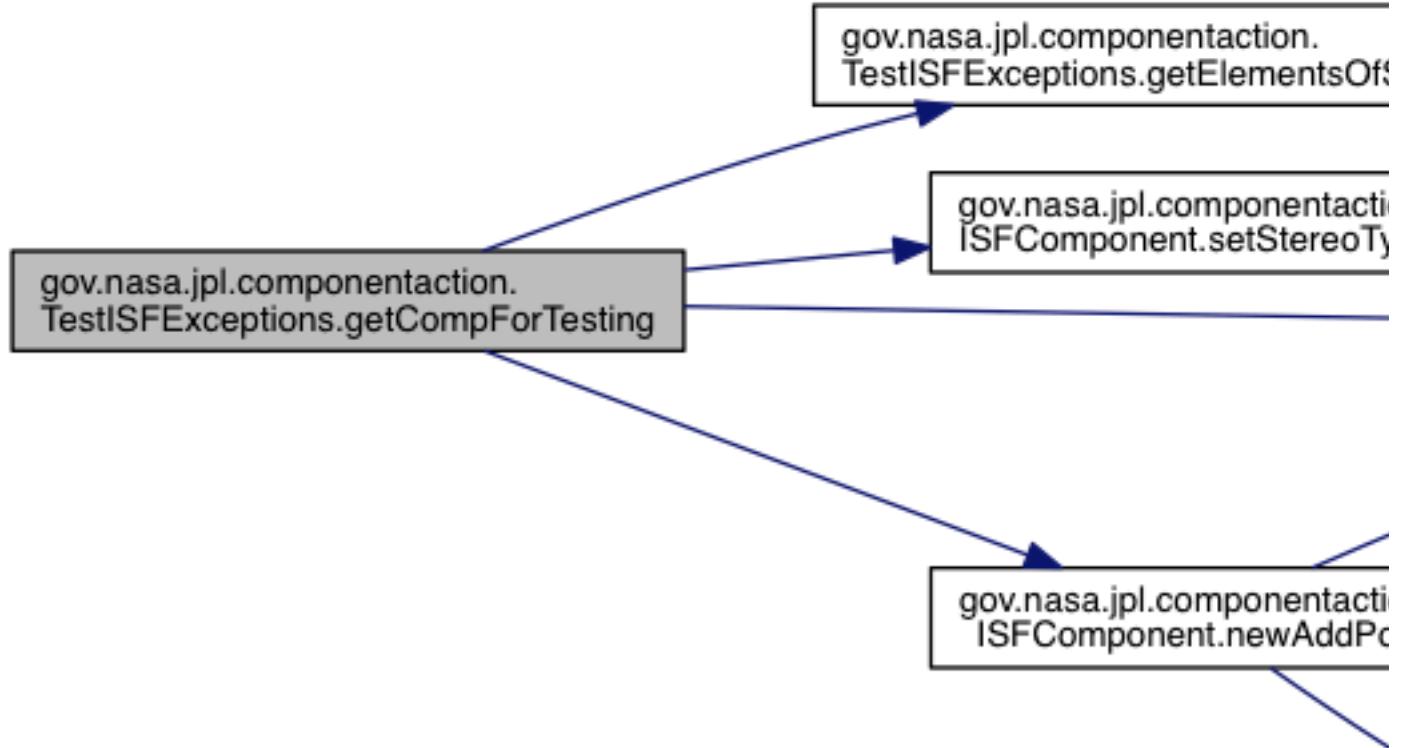
assumes that there is exactly one component in the project

Parameters:

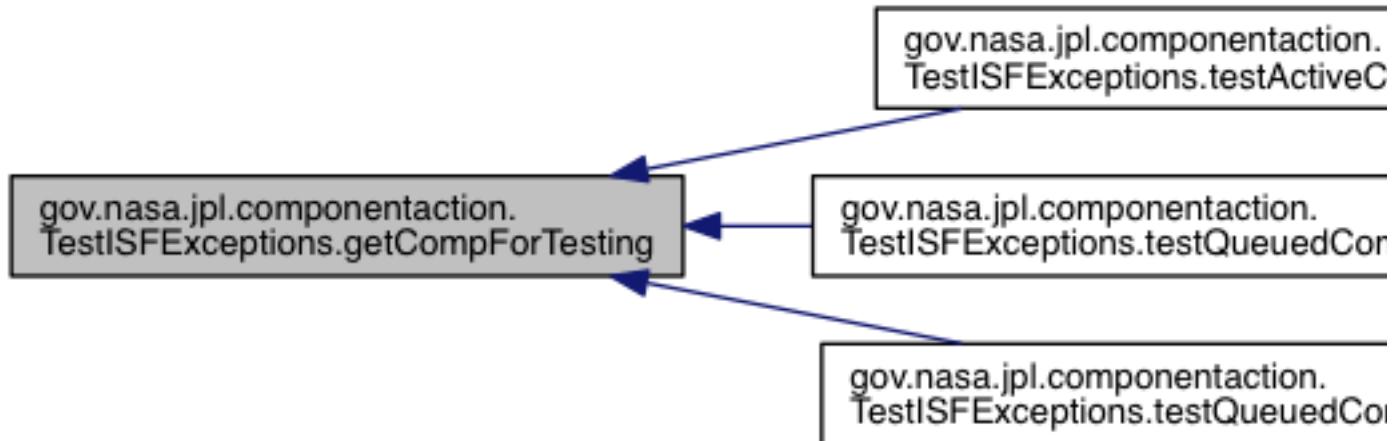
<i>projectName</i>	
<i>compType</i>	

Returns:

Here is the call graph for this function:

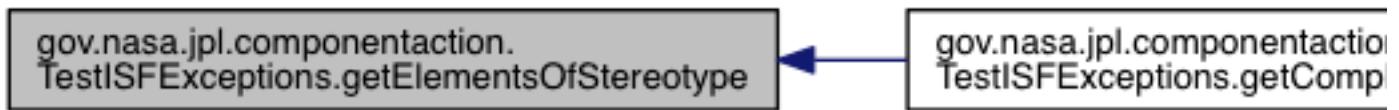


Here is the caller graph for this function:



```
static List<Element> gov.nasa.jpl.componentaction.TestISFExceptions.getElementsOfStereotype
(String stereotype, Project project) [static], [private]
```

Here is the caller graph for this function:



```
static void gov.nasa.jpl.componentaction.TestISFExceptions.main (String[] args) [static]
```

Here is the call graph for this function:



```
void gov.nasa.jpl.componentaction.TestISFExceptions.setUpTest () throws
Exception [protected]
```

```
static Test gov.nasa.jpl.componentaction.TestISFExceptions.suite () [static]
```

Organizes the order in which tests run.

Returns:

Here is the call graph for this function:



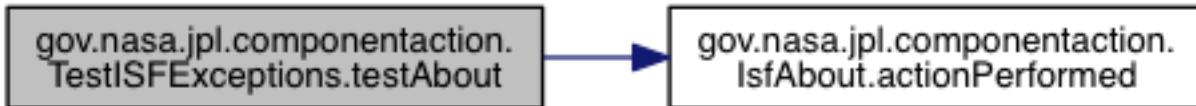
Here is the caller graph for this function:



void gov.nasa.jpl.componentaction.TestISFExceptions.tearDownTest () throws Exception [protected]

void gov.nasa.jpl.componentaction.TestISFExceptions.testAbout ()

Here is the call graph for this function:



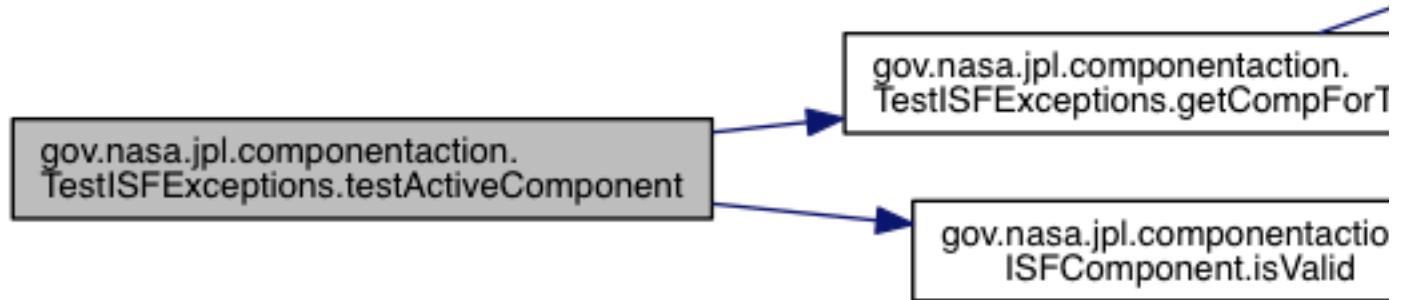
void gov.nasa.jpl.componentaction.TestISFExceptions.testActiveComponent ()

Checks if active component without async ports gives warnings

Exceptions:

<i>IOException</i>

Here is the call graph for this function:



```
void gov.nasa.jpl.componentaction.TestISFExceptions.testCommandLine ()
```

```
void gov.nasa.jpl.componentaction.TestISFExceptions.testComponentNoConnections () throws  
IOException
```

Here is the call graph for this function:


```
void gov.nasa.jpl.componentaction.TestISFExceptions.testDoubleMultiplicity () throws  
IOException
```

Checks if connections originating from the same output port throws exceptions

Exceptions:

<i>IOException</i>	
--------------------	--

Here is the call graph for this function:


```
void gov.nasa.jpl.componentaction.TestISFExceptions.testMismatchedDirections () throws  
IOException
```

Checks if connections with directions that do not match throw exceptions

Exceptions:

<i>IOException</i>	
--------------------	--

Here is the call graph for this function:


```
void gov.nasa.jpl.componentaction.TestISFExceptions.testMismatchedType () throws IOException
```

Checks if connections with types that do not match throw exceptions

Exceptions:

<i>IOException</i>	
--------------------	--

Here is the call graph for this function:


```
void gov.nasa.jpl.componentaction.TestISFExceptions.testMultipleSubsystems () throws  
IOException
```

Tests both subsystems within subsystems, as well as multiple instances of one subsystem.

Note: The input file fails parsing for ports and components.

Exceptions:

<i>IOException</i>	
--------------------	--

Here is the call graph for this function:

```
void gov.nasa.jpl.componentaction.TestISFExceptions.testNoSource () throws IOException
```

Here is the call graph for this function:


```
void gov.nasa.jpl.componentaction.TestISFExceptions.testNoTarget () throws IOException
```

Here is the call graph for this function:


```
void gov.nasa.jpl.componentaction.TestISFExceptions.testPassiveComponent () throws  
IOException
```

Checks if passive components with async inputs throw exceptions

Exceptions:

<i>IOException</i>	
--------------------	--

Here is the call graph for this function:

```
void gov.nasa.jpl.componentaction.TestISFExceptions.testPortException () throws IOException
Checks if ports without types throw exceptions
```

Exceptions:

<i>IOException</i>	
--------------------	--

Here is the call graph for this function:

```
void gov.nasa.jpl.componentaction.TestISFExceptions.testProcessProject () throws IOException
```

Here is the call graph for this function:

```
void gov.nasa.jpl.componentaction.TestISFExceptions.testProcessTopology () throws IOException
```

Checks if topology is generated correctly

Exceptions:

<i>IOException</i>	
--------------------	--

Here is the call graph for this function:


```
void gov.nasa.jpl.componentaction.TestISFExceptions.testQueuedComponentAsync () throws  
IOException
```

Checks if queued component without async ports gives warnings

Exceptions:

<i>IOException</i>	
--------------------	--

Here is the call graph for this function:



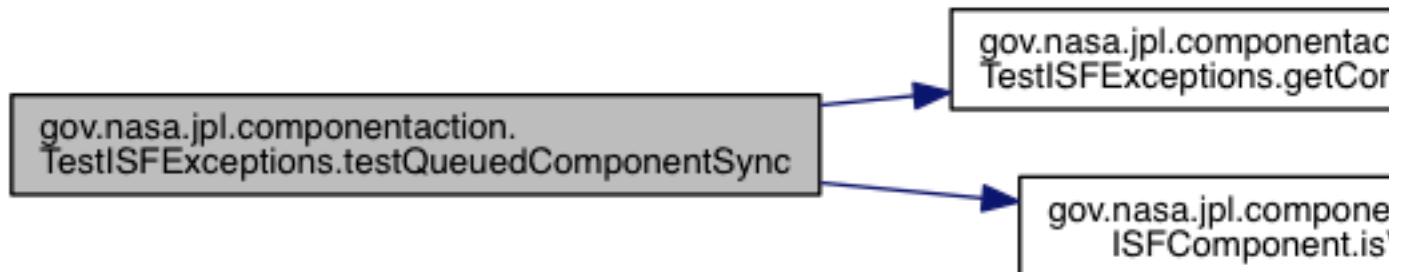
```
void gov.nasa.jpl.componentaction.TestISFExceptions.testQueuedComponentSync () throws  
IOException
```

Checks if queued component without sync ports gives warnings

Exceptions:

<i>IOException</i>	
--------------------	--

Here is the call graph for this function:



Member Data Documentation

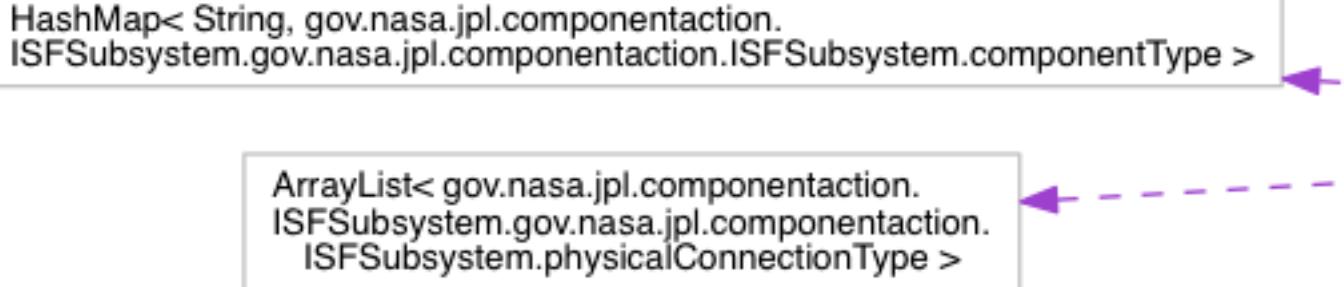
`String gov.nasa.jpl.componentaction.TestISFExceptions.pluginDir [private]`

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

- `src/gov/nasa/jpl/componentaction/TestISFExceptions.java`

gov.nasa.jpl.componentaction.ISFSubsystem.topologyModel Class Reference

Collaboration diagram for gov.nasa.jpl.componentaction.ISFSubsystem.topologyModel:



Public Member Functions

- **topologyModel** (HashMap< String, ISFSubsystem.componentType > componentItemsMap, ArrayList< ISFSubsystem.physicalConnectionType > physicalConnList)
- HashMap< String, ISFSubsystem.componentType > **getComponentMap** ()
- ArrayList< ISFSubsystem.physicalConnectionType > **getPhysicalConnectionList** ()
- String **getBaseID** ()
- String **getInstanceWindow** ()
- String **getName** ()

Package Attributes

- HashMap< String, ISFSubsystem.componentType > **componentMap**
- ArrayList< ISFSubsystem.physicalConnectionType > **physicalConnectionList**
- String **baseID**
- String **instanceWindow**
- String **name**

Detailed Description

Used as a object to store the componentMap, physicalConnectionList, name, baseID, and instaneWindow. An object of this type is then sent to IsfSubXmlWriter and is used to write to the XML output file.

Constructor & Destructor Documentation

```
gov.nasa.jpl.componentaction.ISFSubsystem.topologyModel.topologyModel (HashMap< String, ISFSubsystem.componentType > componentItemsMap, ArrayList< ISFSubsystem.physicalConnectionType > physicalConnList)
```

Member Function Documentation

`String gov.nasa.jpl.componentaction.ISFSubsystem.topologyModel.getBaseID ()`

`HashMap<String , ISFSubsystem.componentType>`
`gov.nasa.jpl.componentaction.ISFSubsystem.topologyModel.getComponentMap ()`

`String gov.nasa.jpl.componentaction.ISFSubsystem.topologyModel.getInstanceWindow ()`

`String gov.nasa.jpl.componentaction.ISFSubsystem.topologyModel.getName ()`

`ArrayList<ISFSubsystem.physicalConnectionType>`
`gov.nasa.jpl.componentaction.ISFSubsystem.topologyModel.getPhysicalConnectionList ()`

Member Data Documentation

`String gov.nasa.jpl.componentaction.ISFSubsystem.topologyModel.baseID [package]`

`HashMap<String , ISFSubsystem.componentType>`
`gov.nasa.jpl.componentaction.ISFSubsystem.topologyModel.componentMap [package]`

`String gov.nasa.jpl.componentaction.ISFSubsystem.topologyModel.instanceWindow [package]`

`String gov.nasa.jpl.componentaction.ISFSubsystem.topologyModel.name [package]`

`ArrayList<ISFSubsystem.physicalConnectionType>`
`gov.nasa.jpl.componentaction.ISFSubsystem.topologyModel.physicalConnectionList [package]`

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

- `src/gov/nasa/jpl/componentaction/ISFSubsystem.java`

gov.nasa.jpl.componentaction.Utils Class Reference

Classes

- class **Pair**

Static Public Member Functions

- static void **throwConnectorException** (String input) throws ConnectorException
- static void **notifyWarning** (String input) throws ConnectorException
- static String **getExtension** (File f)
- static void **printToAll** (String s)

Static Public Attributes

- static final String **aadl** = "aadl"

Detailed Description

This class is used by the JFileChooser to just list files with the *.aadl extension

Author:

kevensen

Member Function Documentation

static String gov.nasa.jpl.componentaction.Utils.getExtension (File *f*)**[static]**

Simple utility method to get the file extension of a file.

Parameters:

<i>f</i>	File to be checked
----------	--------------------

Returns:

String value extension of the file

static void gov.nasa.jpl.componentaction.Utils.notifyWarning (String *input*) throws ConnectorException**[static]**

Appends "====> Warning: " to the input then prints the input to standard out and the MagicDraw console

Parameters:

<i>input</i>	Input String
--------------	--------------

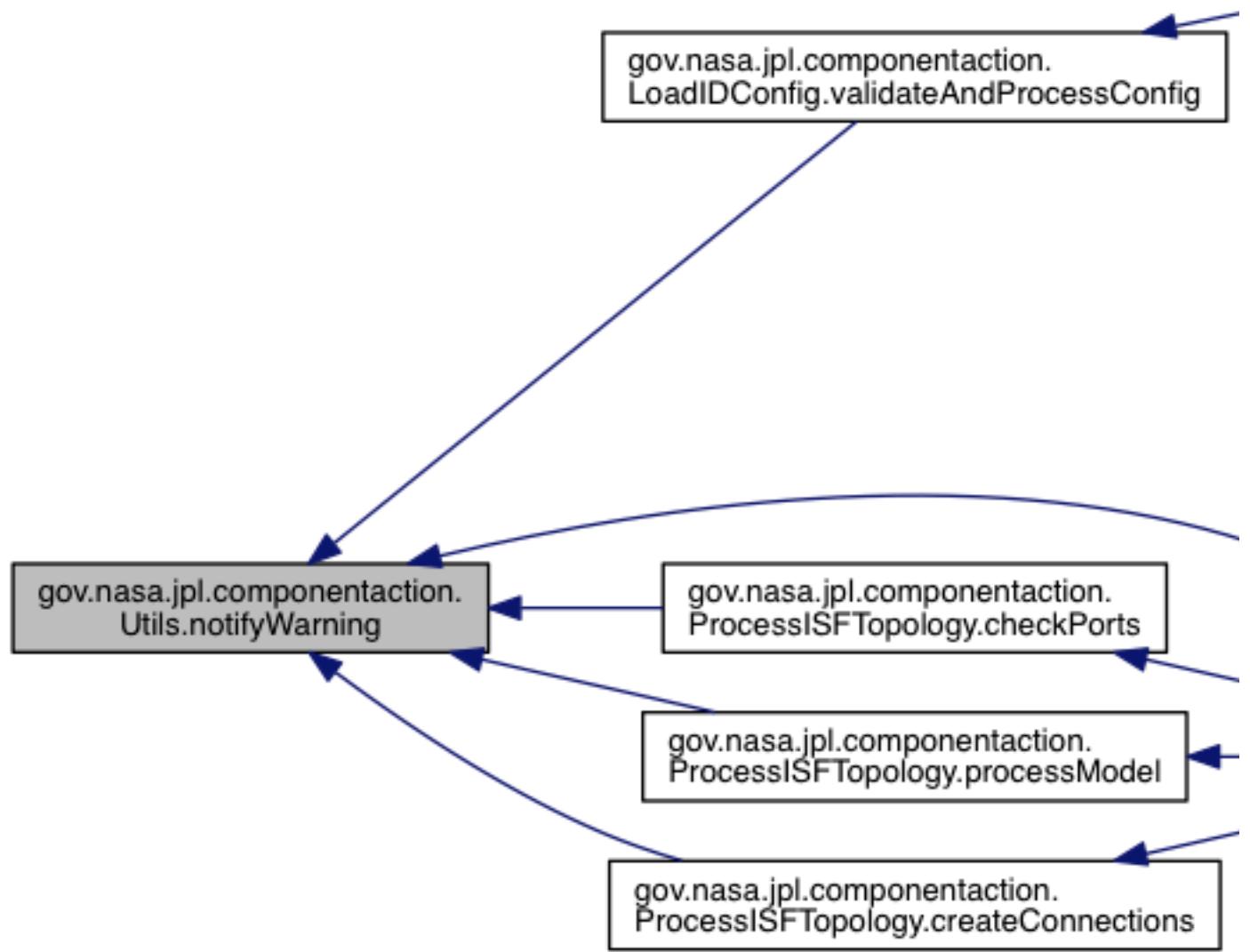
Exceptions:

<i>ConnectorException</i>	
---------------------------	--

Here is the call graph for this function:



Here is the caller graph for this function:



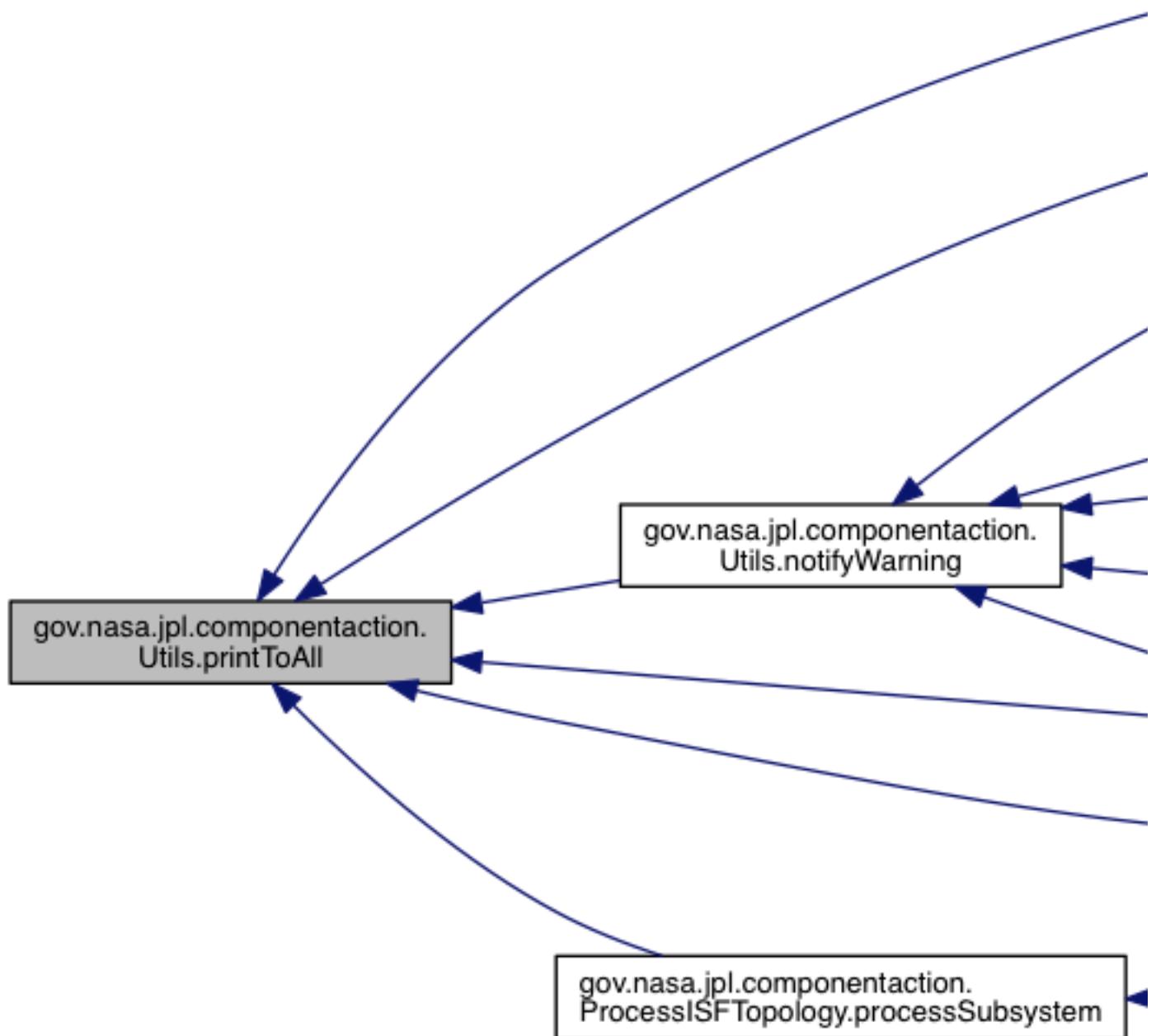
static void gov.nasa.jpl.componentaction.Utils.printToAll (String s)[static]

Combines printing to console out on the Java IDE and MagicDraw

Parameters:

<i>s</i>	String to be printed
----------	----------------------

Here is the caller graph for this function:



```
static void gov.nasa.jpl.componentaction.Utils.throwConnectorException (String input) throws  
ConnectorException [static]
```

Appends to front "====> FATAL Exception: " to the input then prints and throws Connector Exception

Parameters:

<i>input</i>	Input String
--------------	--------------

Exceptions:

<i>ConnectorException</i>	
---------------------------	--

Here is the caller graph for this function:

Member Data Documentation

```
final String gov.nasa.jpl.componentaction.Utils.aadl = "aadl" [static]
```

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

- [src/gov/nasa/jpl/componentaction/Utils.java](#)

File Documentation

src/gov/nasa/jpl/componentaction/ComponentException.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.ComponentException**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/ConnectorException.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.ConnectorException**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/IsfAbout.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.IsfAbout**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/IvfCommandLine.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.IvfCommandLine**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/ISFComponent.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.ISFComponent**
- class **gov.nasa.jpl.componentaction.ISFComponent.portObject**
- class **gov.nasa.jpl.componentaction.ISFComponent.referencePort**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/IsfComponentAction.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.IsfComponentAction**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/ISFPort.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.ISFPort**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/ISFS subsystem.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.ISFS subsystem**
- class **gov.nasa.jpl.componentaction.ISFS subsystem.topologyModel**
- class **gov.nasa.jpl.componentaction.ISFS subsystem.componentType**
- class **gov.nasa.jpl.componentaction.ISFS subsystem.physicalConnectionType**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/IsfTopAction.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.IsfTopAction**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/IsfXmlAction.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.IsfXmlAction**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/IssfXmlAutocoderAction.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.IssfXmlAutocoderAction**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/isfxmlwriter/IsfCompXmlWriter.java

File Reference

Classes

- class **gov.nasa.jpl.componentaction.isfxmlwriter.IsfCompXmlWriter**

Packages

- package **gov.nasa.jpl.componentaction.isfxmlwriter**

src/gov/nasa/jpl/componentaction/isfxmlwriter/IsfPortXmlWriter.java

File Reference

Classes

- class **gov.nasa.jpl.componentaction.isfxmlwriter.IsfPortXmlWriter**

Packages

- package **gov.nasa.jpl.componentaction.isfxmlwriter**

src/gov/nasa/jpl/componentaction/isfxmlwriter/IsfSubXmlWriter.java

File Reference

Classes

- class **gov.nasa.jpl.componentaction.isfxmlwriter.IsfSubXmlWriter**

Packages

- package **gov.nasa.jpl.componentaction.isfxmlwriter**

src/gov/nasa/jpl/componentaction/LoadIDConfig.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.LoadIDConfig**
- class **gov.nasa.jpl.componentaction.LoadIDConfig.configItem**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/LoadIDConfigAction.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.LoadIDConfigAction**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/LoadIDEException.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.LoadIDEException**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/MainMenuConfigurator.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.MainMenuConfigurator**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/MpmcsErrorStream.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.MpmcsErrorStream**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/Partition.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.Partition**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/PortException.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.PortException**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/ProcessISFProject.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.ProcessISFProject**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/ProcessISFTopology.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.ProcessISFTopology**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/TestISFExceptions.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.TestISFExceptions**

Packages

- package **gov.nasa.jpl.componentaction**

src/gov/nasa/jpl/componentaction/Utils.java File Reference

Classes

- class **gov.nasa.jpl.componentaction.Utils**
- class **gov.nasa.jpl.componentaction.Utils.Pair< A, B >**

Packages

- package **gov.nasa.jpl.componentaction**

Index

INDEX