Class Activator

```
java.lang.Object
    |
    +--org.eclipse.core.runtime.Plugin
        |
        +--org.eclipse.ui.plugin.AbstractUIPlugin
            |
            +--pl.eiti.bpelcg.Activator
```

All Implemented Interfaces:

org.osgi.framework.BundleActivator

---

public class **Activator**
extends org.eclipse.ui.plugin.AbstractUIPlugin

The activator class controls the plug-in life cycle.

Fields

PLUGIN_ID

`public static final java.lang.String` **`PLUGIN_ID`**

Constructors

Activator

`public` **`Activator`**`()`

The constructor.

Methods

getDefault

`public static pl.eiti.bpelcg.Activator` **`getDefault`**`()`

Returns the shared instance

Returns:

the shared instance

---

getImageDescriptor

```
public static org.eclipse.jface.resource.ImageDescriptor
getImageDescriptor(java.lang.String path)
```

Returns an image descriptor for the image file at the given plug-in relative path

Parameters:

path - the path

Returns:

the image descriptor

---

start

```
public void start(org.osgi.framework.BundleContext context)
```

Overrides:

start in class org.eclipse.ui.plugin.AbstractUIPlugin

---

stop

```
public void stop(org.osgi.framework.BundleContext context)
```

Overrides:

stop in class org.eclipse.ui.plugin.AbstractUIPlugin

Class CopyGenerateAction

```
java.lang.Object
    |
    +--pl.eiti.bpelcg.actions.CopyGenerateAction
```

All Implemented Interfaces:

org.eclipse.ui.IWorkbenchWindowActionDelegate

---

public class **CopyGenerateAction**
extends java.lang.Object
implements org.eclipse.ui.IWorkbenchWindowActionDelegate

Action implements workbench action delegate. The action proxy will be created by the workbench and shown in the UI. When the user tries to use the action, this delegate will be created and execution will be delegated to it.

IWorkbenchWindowActionDelegate

Constructors

CopyGenerateAction

```
public  CopyGenerateAction()
```

The constructor.

Methods

dispose

```
public void dispose()
```

We can use this method to dispose of any system resources we previously allocated.

---

init

```
public void init(org.eclipse.ui.IWorkbenchWindow window)
```

We will cache window object in order to be able to provide parent shell for the message dialog.

---

run

```
public void run(org.eclipse.jface.action.IAction action)
```

The action has been activated. The argument of the method represents the 'real' action sitting in the workbench UI.

---

selectionChanged

```
public void selectionChanged(org.eclipse.jface.action.IAction action,
                             org.eclipse.jface.viewers.ISelection selection)
```

Selection in the workbench has been changed. We can change the state of the 'real' action here if we want, but this can only happen after the delegate has been created.

Class UndoAction

```
java.lang.Object
    |
    +--pl.eiti.bpelcg.actions.UndoAction
```

All Implemented Interfaces:
org.eclipse.ui.IWorkbenchWindowActionDelegate

---

public class **UndoAction**
extends java.lang.Object
implements org.eclipse.ui.IWorkbenchWindowActionDelegate

Action implements workbench action delegate. The action proxy will be created by the workbench and shown in the UI. When the user tries to use the action, this delegate will be created and execution will be delegated to it.

IWorkbenchWindowActionDelegate

Constructors

UndoAction

```
public  UndoAction()
```

Methods

dispose

```
public void dispose()
```

---

init

```
public void init(org.eclipse.ui.IWorkbenchWindow arg0)
```

---

run

```
public void run(org.eclipse.jface.action.IAction arg0)
```

---

selectionChanged

```
public void selectionChanged(org.eclipse.jface.action.IAction arg0,
                             org.eclipse.jface.viewers.ISelection arg1)
```

Interface IAnalysisResult

---

public interface **IAnalysisResult**

Generic analysis result structure interface.

K key type.

V value type as List.

Methods

get

```
public java.util.List get(java.lang.Object key)
```

Gets value from map for key given as parameter.

Parameters:

key - key to read from map.

Returns:

value from map mapped by a given key.

---

put

```
public java.util.List put(java.lang.Object key,
                          java.util.List value)
```

Puts a value into a mapped by a key given as parameter.

Parameters:

key - key element from map.
value - value to put in the map.

Returns:

value put into the map.

Interface IAnalyzer

public interface **IAnalyzer**

Behavior of analyzer for model created from BPEL process.

Methods

analyze

`public pl.eiti.bpelcg.analyzer.IAnalysisResult` **`analyze`**`()`

Analyze existing loaded model and give results of the analyze back.

Returns:

results of the model analyze

---

createCopy

`public org.eclipse.bpel.model.Copy` **`createCopy`**`()`

---

getAssignActivities

`public java.util.List` **`getAssignActivities`**`()`

Gets all assign blocks from BPEL process.

Returns:

list of assign block of BPEL process

---

getProcessVariables

`public java.util.List` **`getProcessVariables`**`()`

Gets all variables of analyzed BPEL process.

Returns:

list of variables of BPEL process

---

init

```
public void init(java.lang.String pathToBPEL)
```

Initial method for analyzer to prepare necessary things (load BPEL, WSDLs) to analyze.

Parameters:

pathToBPEL - BPEL process file location

Class AnalysisResult

```
java.lang.Object
    |
    +--java.util.AbstractMap
        |
        +--java.util.HashMap
            |
            +--pl.eiti.bpelcg.analyzer.impl.AnalysisResult
```

All Implemented Interfaces:

java.io.Serializable, java.lang.Cloneable, java.util.Map, pl.eiti.bpelcg.analyzer.IAnalysisResult

---

public class **AnalysisResult**
extends java.util.HashMap
implements pl.eiti.bpelcg.analyzer.IAnalysisResult

Implementation of analysis result created by Analyzer element. Structure represented by map with assign elements as a keys and list of copy instructions as values mapped by keys.

Constructors

AnalysisResult

```
public  AnalysisResult()
```

Methods

get

```
public java.util.List get(org.eclipse.bpel.model.Assign key)
```

---

put

```
public java.util.List put(org.eclipse.bpel.model.Assign key,
                          java.util.List value)
```

Overrides:

put in class java.util.HashMap

Class Analyzer

```
java.lang.Object
    |
    +--pl.eiti.bpelcg.util.Settings
        |
        +--pl.eiti.bpelcg.analyzer.impl.Analyzer
```

All Implemented Interfaces:
pl.eiti.bpelcg.analyzer.IAnalyzer

---

public class **Analyzer**
extends pl.eiti.bpelcg.util.Settings
implements pl.eiti.bpelcg.analyzer.IAnalyzer

BPEL graph model analyzer class. Holds references to DAO object - access to BPEL and WSDL files. Analyzes graph model representing BPEL process and finds matches between variables used in BPEL process.

Constructors

Analyzer

```
public  Analyzer()
```

Default analyzer constructor.

Methods

analyze

```
public pl.eiti.bpelcg.analyzer.IAnalysisResult analyze()
```

---

createCopy

`public org.eclipse.bpel.model.Copy **createCopy**()`

---

getAssignActivities

`public java.util.List **getAssignActivities**()`

---

getBPELProcess

`public org.eclipse.bpel.model.Process **getBPELProcess**()`

BPEL process getter.

Returns:

BPEL process as BPEL Designer EMF object.

---

getModel

`public pl.eiti.bpelcg.model.impl.GraphModel **getModel**()`

Graph model of BPEL process getter.

Returns:

BPEL process graph representation.

---

getProcessVariables

`public java.util.List **getProcessVariables**()`

---

init

`public void **init**(java.lang.String pathToBPEL)`

---

saveProcess

`public void **saveProcess**()`

Delegates BPEL process save.

Class BPELDAO

```
java.lang.Object
    |
    +--pl.eiti.bpelcg.dao.BPELDAO
```

public class **BPELDAO**
extends java.lang.Object

Reader class using to load BPEL process file to object model based on org.eclipse.bpel.model package classes.

Constructors

BPELDAO

`public` **`BPELDAO`**`()`

Default constructor.

BPELDAO

`public` **`BPELDAO`**`(java.lang.String newBPELFileLocation)`

File location set constructor.

Parameters:

newBPELFileLocation - file location

Methods

getAllAssignBlocks

`public java.util.List` **`getAllAssignBlocks`**`()`

BPEL process all assign instructions getter.

Returns:

list of all assign instruction from BPEL process.

getAllReceives

`public java.util.List` **`getAllReceives`**`()`

Gets receive elements from BPEL process.

Returns:

list of receives from BPEL process.

---

getAllVariables

`public java.util.List` **`getAllVariables`**`()`

Gets all variables from BPEL process.

Returns:

list of BPEL process variables.

---

getBPELProcess

`public org.eclipse.bpel.model.Process` **`getBPELProcess`**`()`

Gets BPEL process reference.

Returns:

BPEL process reference.

---

getFactory

`public org.eclipse.emf.ecore.resource.Resource.Factory` **`getFactory`**`()`

Factory getter.

Returns:

factory object.

---

loadProcess

`public void` **`loadProcess`**`()`

BPEL process load method.

---

savePocess

`public java.lang.String` **`savePocess`**`()`

BPEL process save method.

Returns:

saved process name

---

setBPELFileLocation

`public void` **`setBPELFileLocation`**`(java.lang.String newBPELFileLocation)`

BPEL process file location setter.

Parameters:

newBPELFileLocation - location of BPEL process file.

Class WSDLDAO

```
java.lang.Object
    |
    +--pl.eiti.bpelcg.dao.WSDLDAO
```

---

public class **WSDLDAO**
extends java.lang.Object

WSDL DAO (Data Access Object) closes the functionality of loading WSDL files for retrieving messages elements.

Methods

getInstance

`public static pl.eiti.bpelcg.dao.WSDLDAO` **`getInstance`**`()`

Gets singleton instance of WSDLDAO object reference.

Returns:

reference to WSDL DAO object.

---

getMessage

```
public org.eclipse.wst.wsdl.Message getMessage(javax.xml.namespace.QName
qName)
```

Gets message with given qName element.

Parameters:

qName - query name.

Returns:

WSDL message element.

---

getMessages

```
public java.util.List getMessages()
```

Gets list of all messages loaded.

Returns:

list of messages loaded by DAO object.

---

load

```
public void load(java.util.List importLocations,
                 java.lang.String absolutePath)
```

WSDL load method.

Parameters:

importLocations - WSDL files path list
absolutePath - absolute path to BPEL process file

Interface IMatcher

---

public interface **IMatcher**

Behavior of element used to finding matches between BPEL process elements for generation of copy instructions in the process.

Methods

createCopyForMatchedVariables

```
public java.util.List createCopyForMatchedVariables(java.util.List
settedVariables,
                                                     java.util.List
followingInvokes)
```

Finds all variables from given list that match (with name and type) to any input variables of invokes from given list.

Parameters:

settedVariables - list of variables that can be used as invoke call parameters
followingInvokes - list of invoke activities following currently processed assign activity

---

Class DefaultMatcher

```
java.lang.Object
    |
    +--pl.eiti.bpelcg.util.Settings
        |
        +--pl.eiti.bpelcg.matcher.impl.DefaultMatcher
```

All Implemented Interfaces:
pl.eiti.bpelcg.matcher.IMatcher

---

public class **DefaultMatcher**
extends pl.eiti.bpelcg.util.Settings
implements pl.eiti.bpelcg.matcher.IMatcher

Default plugin matcher using simple rule for searching matches between variables which checks equality of names and types.

Constructors

DefaultMatcher

```
public  DefaultMatcher()
```

Default constructor used for get instances of singleton WSDL DAO and resolver elements.

Methods

createCopyForMatchedVariables

```
public java.util.List createCopyForMatchedVariables(java.util.List
settedVariables,
                                                     java.util.List
followingInvokes)
```

Interface IModel

---

public interface **IModel**

Model behavior interface.

Class Graph

```
java.lang.Object
    |
    +--pl.eiti.bpelcg.model.graph.Graph
```

Direct Known Subclasses:
pl.eiti.bpelcg.model.impl.GraphModel

---

public class **Graph**
extends java.lang.Object

Graph implementation with root node selected.

T graph nodes type

Constructors

Graph

```
public  Graph()
```

Graph default constructor.

Graph

`public` **`Graph`**`(java.lang.Object data)`

Constructor creating new node from given data.

Parameters:

data - node data element

Graph

`public` **`Graph`**`(pl.eiti.bpelcg.model.graph.GraphNode rootNode)`

Constructor setting given graph node as root.

Parameters:

rootNode - node element to set as root element

Methods

getRoot

`public pl.eiti.bpelcg.model.graph.GraphNode` **`getRoot`**`()`

Graph root element getter.

Returns:

root node

---

setRoot

`public void` **`setRoot`**`(pl.eiti.bpelcg.model.graph.GraphNode rootNode)`

Graph root element setter.

Parameters:

rootNode - node to set as graph root

Class GraphNode

```
java.lang.Object
    |
    +--pl.eiti.bpelcg.model.graph.GraphNode
```

---

public class **GraphNode**

extends java.lang.Object

Graph node element implementation.

T type of node data

Constructors

GraphNode

`public` **`GraphNode`**`(java.lang.Object newData)`

Graph node data set constructor.

Parameters:

newData - graph node data element to set

Methods

addNextNode

`public void` **`addNextNode`**`(pl.eiti.bpelcg.model.graph.GraphNode nextNode)`

Adds next node to list of next nodes.

Parameters:

nextNode - node to add.

---

addPreviousNode

`public void` **`addPreviousNode`**`(pl.eiti.bpelcg.model.graph.GraphNode previousNode)`

Adds previous node to list of previous nodes.

Parameters:

previousNode - node to add.

---

getData

`public java.lang.Object` **`getData`**`()`

Gets data of node.

Returns:

node data element.

---

getNextNodes

`public java.util.List` **`getNextNodes`**`()`

Gets next nodes list.

Returns:

list of nodes.

---

getPreviousNodes

`public java.util.List` **`getPreviousNodes`**`()`

Gets previous nodes list.

Returns:

list of nodes.

---

getState

`public pl.eiti.bpelcg.model.graph.GraphNode.State` **`getState`**`()`

Gets state of node.

Returns:

enumerated state.

---

hasNext

`public java.lang.Boolean` **`hasNext`**`()`

Next element existing check.

Returns:

next element existing bool value

---

hasPrevious

`public java.lang.Boolean` **`hasPrevious`**`()`

Previous element existing check.

Returns:

previous element existing bool value

---

isBranched

`public java.lang.Boolean` **`isBranched`**`()`

Multiple next elements check.

Returns:

many next elements existing bool value

---

isProcessing

`public java.lang.Boolean` **`isProcessing`**`()`

Processed check.

Returns:

if is processing

---

isUnvisited

`public java.lang.Boolean` **`isUnvisited`**`()`

Unvisited check.

Returns:

if was not visited

---

isVisited

`public java.lang.Boolean` **`isVisited`**`()`

Visited check.

Returns:

if was visited

---

setNextNodes

`public void` **`setNextNodes`**`(java.util.List nextNodes)`

Sets next nodes list.

Parameters:

nextNodes - list of nodes.

---

setPreviousNodes

`public void` **`setPreviousNodes`**`(java.util.List previousNodes)`

Sets previous nodes list.

Parameters:

previousNodes - list of nodes.

---

setProcessing

`public void` **`setProcessing`**`()`

Processed state setter.

---

setUnvisited

`public void` **`setUnvisited`**`()`

Unvisited state setter.

---

setVisited

`public void` **`setVisited`**`()`

Visited state setter.

Class GraphModel

```
java.lang.Object
    |
    +--pl.eiti.bpelcg.model.graph.Graph
        |
        +--pl.eiti.bpelcg.model.impl.GraphModel
```

All Implemented Interfaces:

pl.eiti.bpelcg.model.IModel

---

public class **GraphModel**
extends pl.eiti.bpelcg.model.graph.Graph
implements pl.eiti.bpelcg.model.IModel

Graph model with nodes org.eclipse.bpel.model.Activity type.

Constructors

GraphModel

`public` **`GraphModel`**`()`

Default graph model constructor.

Class WSDLResolver

```
java.lang.Object
    |
    +--pl.eiti.bpelcg.util.Settings
        |
        +--pl.eiti.bpelcg.resolver.WSDLResolver
```

---

public class **WSDLResolver**
extends pl.eiti.bpelcg.util.Settings

WSDL resolver used to retrieving information about complex and simple types defined in WSDL and used to communication with service which interface describes WSDL document.

Methods

getInstance

`public static pl.eiti.bpelcg.resolver.WSDLResolver` **`getInstance`**`()`

---

resolveMessageType

`public java.util.Map` **`resolveMessageType`**`(org.eclipse.wst.wsdl.Message complexMessageType)`

Resolves message type to map part element to elements included in part element.

Parameters:

complexType - complex type message

Returns:

map of message part elements to list of elements within selected part element

---

retrieveElements

```
public java.util.List retrieveElements(org.eclipse.wst.wsdl.Message
complexType)
```

Retrieves all elements of given complex type message and create concatenated information about them delimited by M_LIMITER defined in Settings class (MessagePartName/delimiter/xsdElementTypeName/delimiter/xsdElementName).

Parameters:

complexType - defined in WSDL complex type message

Returns:

list of concatenated information about elements

Interface IProcessTransformer

---

public interface **IProcessTransformer**

Factory transformer of BPEL Process to model and model to BPEL Process behavior.

Methods

processToModel

```
public pl.eiti.bpelcg.model.IModel
processToModel(org.eclipse.bpel.model.Process process)
```

BPEL process to model declaration.

Parameters:

process - process element to transform

Returns:

model created from process

Class GraphTransformer

```
java.lang.Object
    |
    +--pl.eiti.bpelcg.transformer.impl.GraphTransformer
```

All Implemented Interfaces:

pl.eiti.bpelcg.transformer.IProcessTransformer

---

public class **GraphTransformer**

extends java.lang.Object
implements pl.eiti.bpelcg.transformer.IProcessTransformer

Singleton concrete factory class for producing Graph model from BPEL process, and update BPEL process from Graph model.

Methods

getInstance

```
public static pl.eiti.bpelcg.transformer.impl.GraphTransformer getInstance()
```

Graph transformer instance getter.

Returns:

graph transformer instance reference

---

processToModel

```
public pl.eiti.bpelcg.model.IModel
processToModel(org.eclipse.bpel.model.Process process)
```

Interface IUpdater

public interface **IUpdater**

Behavior of updater BPEL process elements using elements from analysis result.

Methods

createNewFrom

```
public org.eclipse.bpel.model.From createNewFrom()
```

Creates a new From element - source of value copying in copy instruction from assign BPEL element.

Returns:

instance of From element.

---

createNewQuery

```
public org.eclipse.bpel.model.Query createNewQuery()
```

Creates a query used to retrieve value of field of complex type variable in BPEL copy instruction.

Returns:

instance of Query element.

---

createNewTo

```
public org.eclipse.bpel.model.To createNewTo()
```

Creates a new To element - destination of value copying in copy instruction from assign BPEL element.

Returns:

instance of To element.

---

update

```
public void update(org.eclipse.bpel.model.Process BPELprocess,
                   pl.eiti.bpelcg.analyzer.IAnalysisResult analysis)
```

Updates BPEL process using analysis result elements.

Parameters:

BPELprocess - BPEL process in memory.
analysis - result of process analyze.

Class Updater

```
java.lang.Object
    |
    +--pl.eiti.bpelcg.updater.impl.Updater
```

All Implemented Interfaces:
pl.eiti.bpelcg.updater.IUpdater

---

public class **Updater**
extends java.lang.Object
implements pl.eiti.bpelcg.updater.IUpdater

Generator updates all assign activities of given process from analysis result map - if copy elements generated exists for given assign activity.

Constructors

Updater

public **Updater**()

Methods

createNewFrom

public org.eclipse.bpel.model.From **createNewFrom**()

---

createNewQuery

public org.eclipse.bpel.model.Query **createNewQuery**()

---

createNewTo

public org.eclipse.bpel.model.To **createNewTo**()

---

update

public void **update**(org.eclipse.bpel.model.Process BPELprocess,
pl.eiti.bpelcg.analyzer.IAnalysisResult analysis)