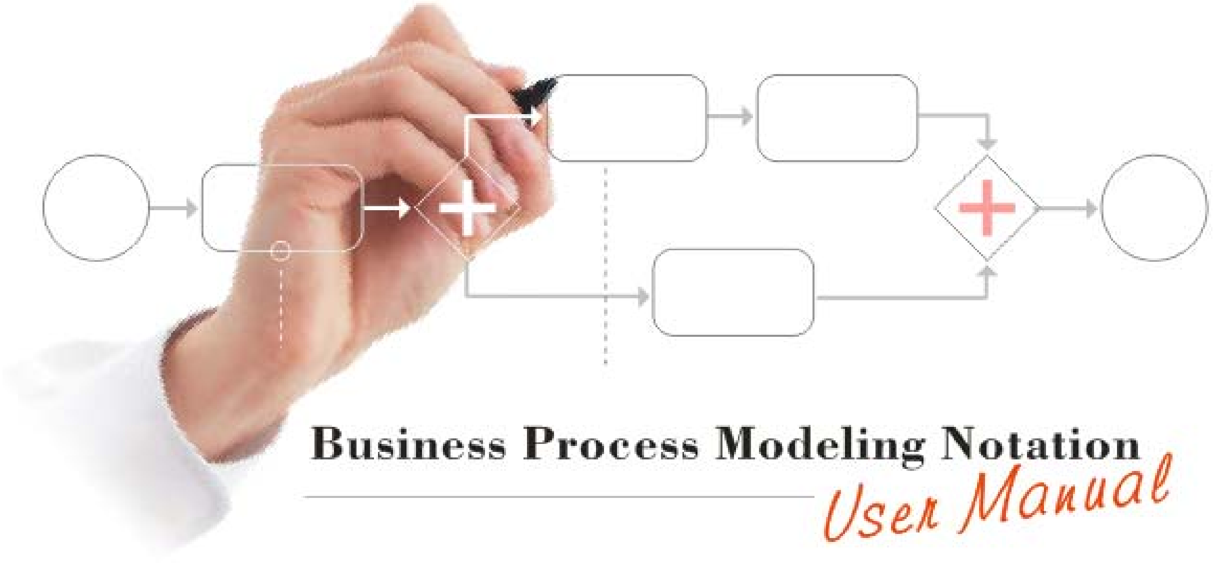


Business Process Modeling Notation

User Manual

Purpose of this Manual

This document is a user guide for BPMN V1.0. It is a reference manual for building applications using Cuecent BPMN.

Intended Readership

This document is for users such as business analysts, application developers and workflow designers, who use the Cuecent BPMN to define and administer workflows. It is addressed to both technical and non-technical users alike.

Structure of this Manual

This document is organized as follows

Chapter 1 Introduction discusses about Cuecent BPMN, feature highlights, and abbreviations and acronyms used in this document

Chapter 2 Getting Started with Cuecent BPMN is a brief overview to get started and familiarized with Cuecent BPMN. It also provides a brief outline and walkthrough of the basic Cuecent BPMN UI.

Chapter 3 BPMN UI in Detail – details features of the BPMN UI in full

Chapter 4 Advanced Features of the BPMN UI: Discusses advanced topics – specifically setting various properties for each BPD element

Other Documents

Cuecent BPMN Release Notes and Installation document gives details specific to a release - including installation and may also include know issues and bugs fixed from previous releases

Table of Contents

1 Introduction

1.1 Background

Bahwan CyberTek Pvt Ltd (BCTPL) is involved in the development of its flagship product – “Cuecent” BPMS (Business Process Modeling System) , and the current targeted release is version 4.0. BPMN 1.0 is part of Cuecent BPMS 4.0 suite and is a typical Business Process Modeling System that conforms to the BPMN and XPDL standards from the OMG (Object Management Group) and WFMC (Workflow Management Coalition) respectively. The version of the BPMN specification that is being targeted is BPMN 1.1 with XDPL version 2.1

The main components of Cuecent BPMS 4.0 are: (i) Cuecent BPMN 1.0 UI (or BPMN 1.0 for short) which is a product that lets one create Business Process Diagrams (BPDs) and convert them into XPDL format (ii) BPMS Engine which allows the XPDL from BPMN to be executed as per an execution model and (iii) BAM or “Business Activity Monitor”

BPMN 1.0 is being developed on the Eclipse 3.3 (Europa) Rich Client Platform (RCP) environment with Java 1.5 and GEF 3.3.2 (Graphical Editor Framework) plug in. This document outlines the BPMN 1.0 UI product as in (i) above. Specifically, it details all the features of the BPMN 1.0 tool and is intended to be a User Manual for Cuecent BPMN1.0.

1.2 Scope

Cuecent BPMN 1.0 implements the BPMN Specification V 1.1 – items not in scope are identified in the BPMN Release note and Installation Guide document.

1.3 Acronyms & Abbreviations

S. No	Acronyms	Description
1	BPMN	Business Process Modeling Notation
2	XPDL	XML Process Definition Language
3	SWT	Standard Widget Toolkit (On which GEF and Draw2d are built)
4	Draw2d	A Lightweight framework built on top of SWT to support graphics
5	GEF	Graphical Editor Framework – a framework built on Draw2d and SWT used to create generalized Graphical Editors
6	UUID	Universally Unique Identifier

1.4 Conventions Used

Bahwan CyberTek online help and PDF manuals provide standards to help you identify notational system used in this manual. The following table lists these conventions.

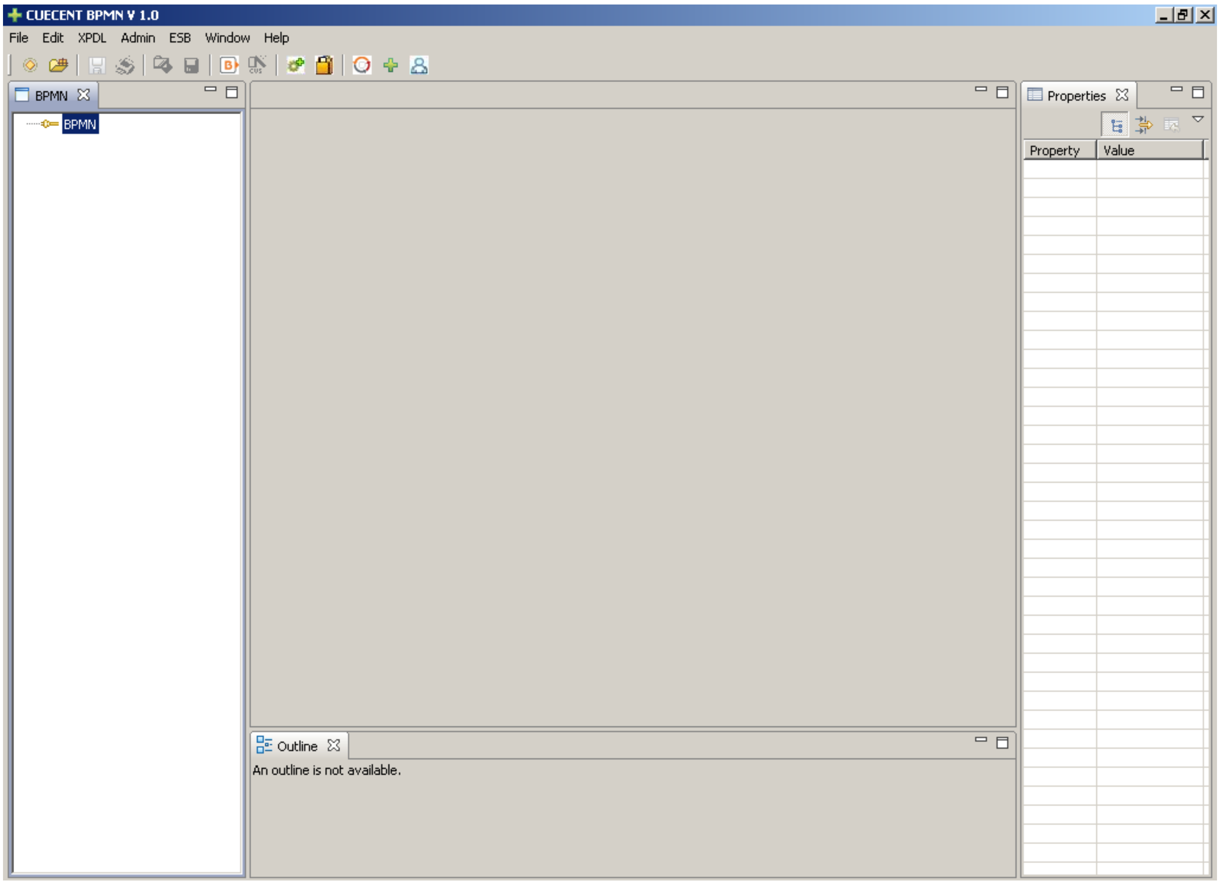
S. No	Type	Indicates
1	**Bold**	Drop down boxes, button names, text fields, check boxes, options, lists, and menus that are the focus of actions or part of a list of such GUI elements and their definitions text to be entered by the user. Example: Click **Edit Project**
2	*Italic*	Terms defined and in glossary, when part of a command syntax, indicates variable, information to be replaced by the user
3	Courier font	Code samples, calculations, registry keys, path and file names, URLs, messages displayed in the screen
4	UPPERCASE	Keyboard command key (ENTER) shortcut key (CTRL+A)
5	📝	Indicates helpful information for a specific situation
6	⚠	Prompts you with crucial information to be read before continuing

1.5 References

S. No	Document Name	Purpose
1	BPMN Specification Version 1.1	Outlines the BPMN specification
2	XPDL Specification Version 2.1	Outlines the XPDL specification
3	BPMN 1.0 Release Note and Installation Document	Release note and Installation Guide specific to a release

2 Getting Started with Cuecent BPMN

This chapter starts off with usage of various elements of the BPMN UI and typical usage. This assumes that BPMN is installed prior to use – please refer to the BPMN Release Note and Installation Guide for information on installation. In the installed package, there should be an executable – bpmn.exe. This is the executable for BPMN – double click on this to start BPMN. This chapter also assumes that you have not created any BPDs yet. So you should see the screen below when you start off. Let us now walk through the basics of the BPMN UI and try and draw a simple BPD.



An outline is not available.

Figure 1 BPMN Initial screen

2.1 First Step: BPMN 1.0 GUI

2.1.1 Drawing your first BPD

The various components of the BPMN UI are graphically depicted in figure 2.

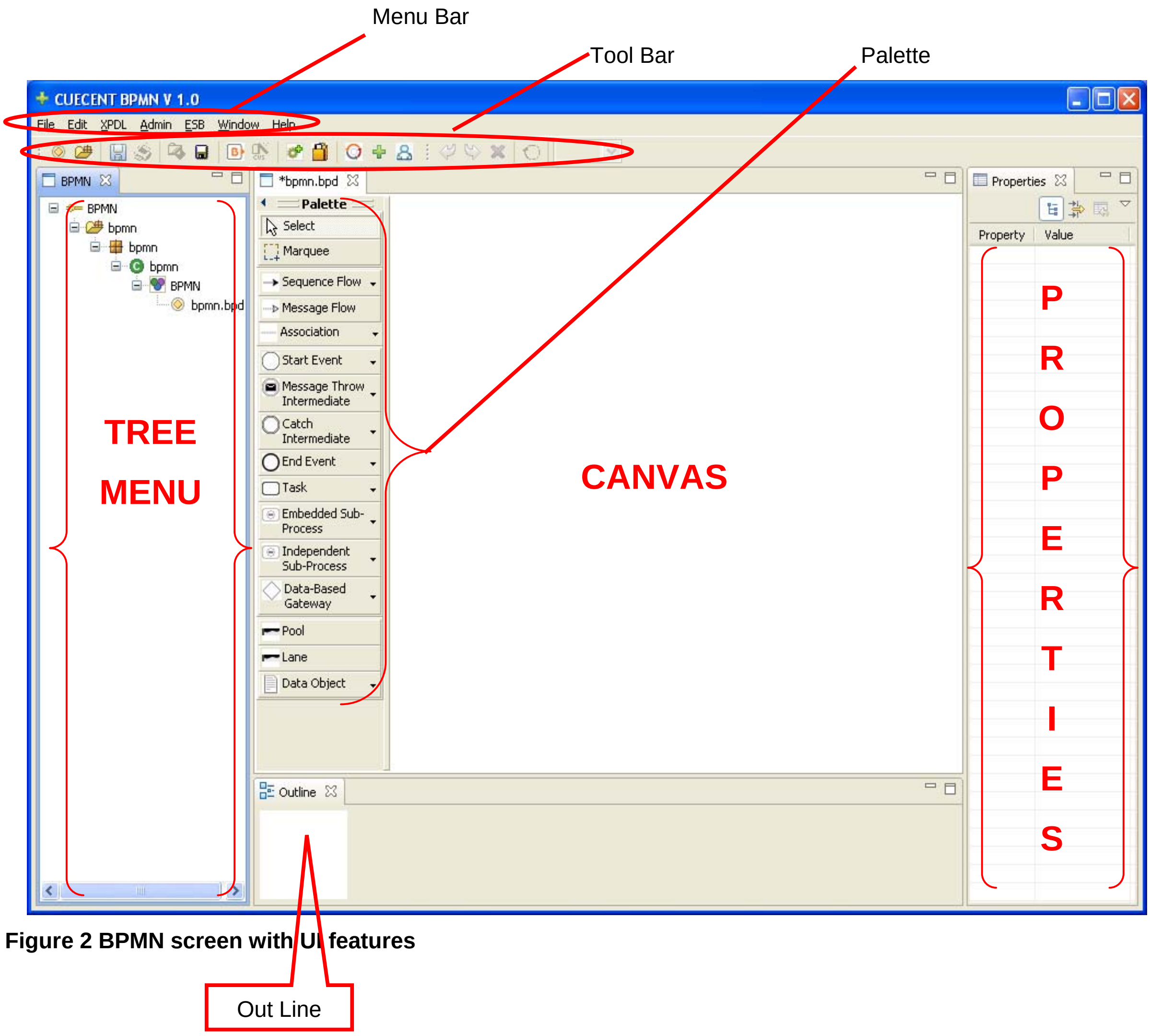


Figure 2 BPMN screen with UI features

The various components in the UI above are explained below:

1. The Tree view lets the user create various entities as explained later in this chapter
2. The Editor area is a container for all graphics and includes a palette and a canvas
3. The menu and toolbar let you implement various actions on BPDs as explained in the next chapter
4. The properties view is explained in the next chapter and constitutes various properties that may be set for various graphical entities in a BPD
5. The outline at the bottom is an “outline view” of the BPD

Any BPD in Cuecent BPMN v1.0 is part of a hierarchy, the root of which is the BPMN element as in the figure above. Under the BPMN root element one can create as many projects as possible. To create a project, simply right click on the BPMN element and enter project name as shown in figure 3 and click OK.

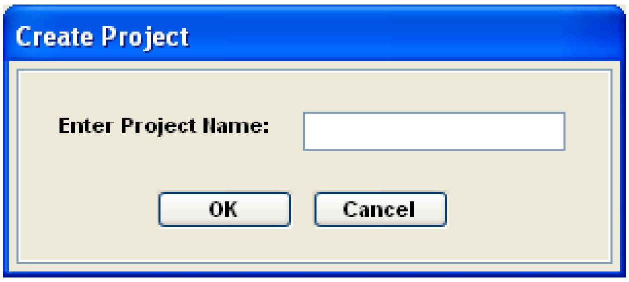


Figure 3 Project Dialog box

The next entity in the hierarchy is the Package. Create a Package by right clicking on the project just created, as shown in figure 4.

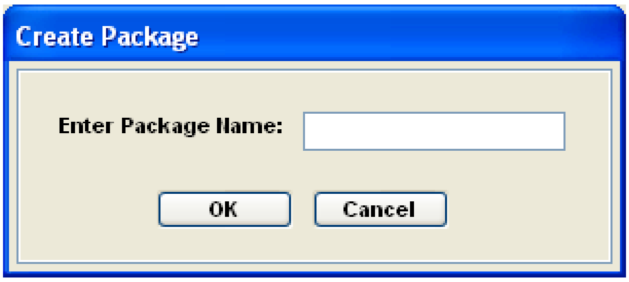


Figure 4 Package Dialog box

After the Package, we have a module. Create a Module by right clicking on the package just created, as shown in figure 5.

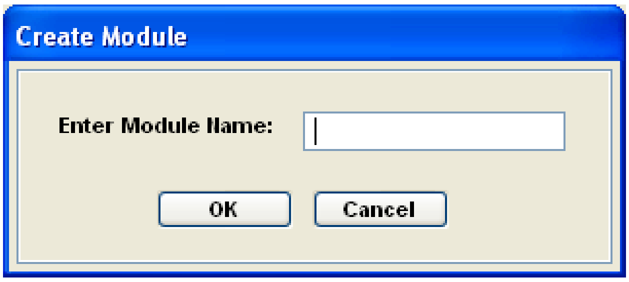


Figure 5 Module Dialog box

After the module entity, we have a builder entity. Currently the builder entity can only be a “BPMN” entity – select the checkbox as below after right clicking on the module just created, as shown in figure 6.

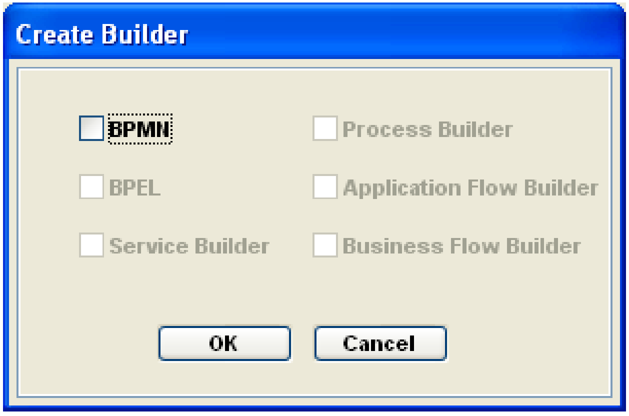


Figure 6 Builder Dialog box

The final (leaf entity) on the hierarchy is the BPD file – create a BPD file by right clicking on the builder as below. Right click the Builder to open a File Name Entry Dialog box. File Name Entry Dialog box appears as shown in figure 7.

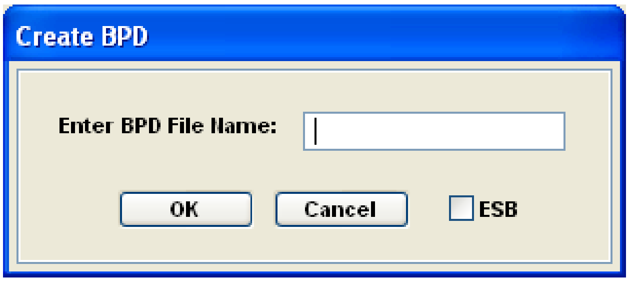
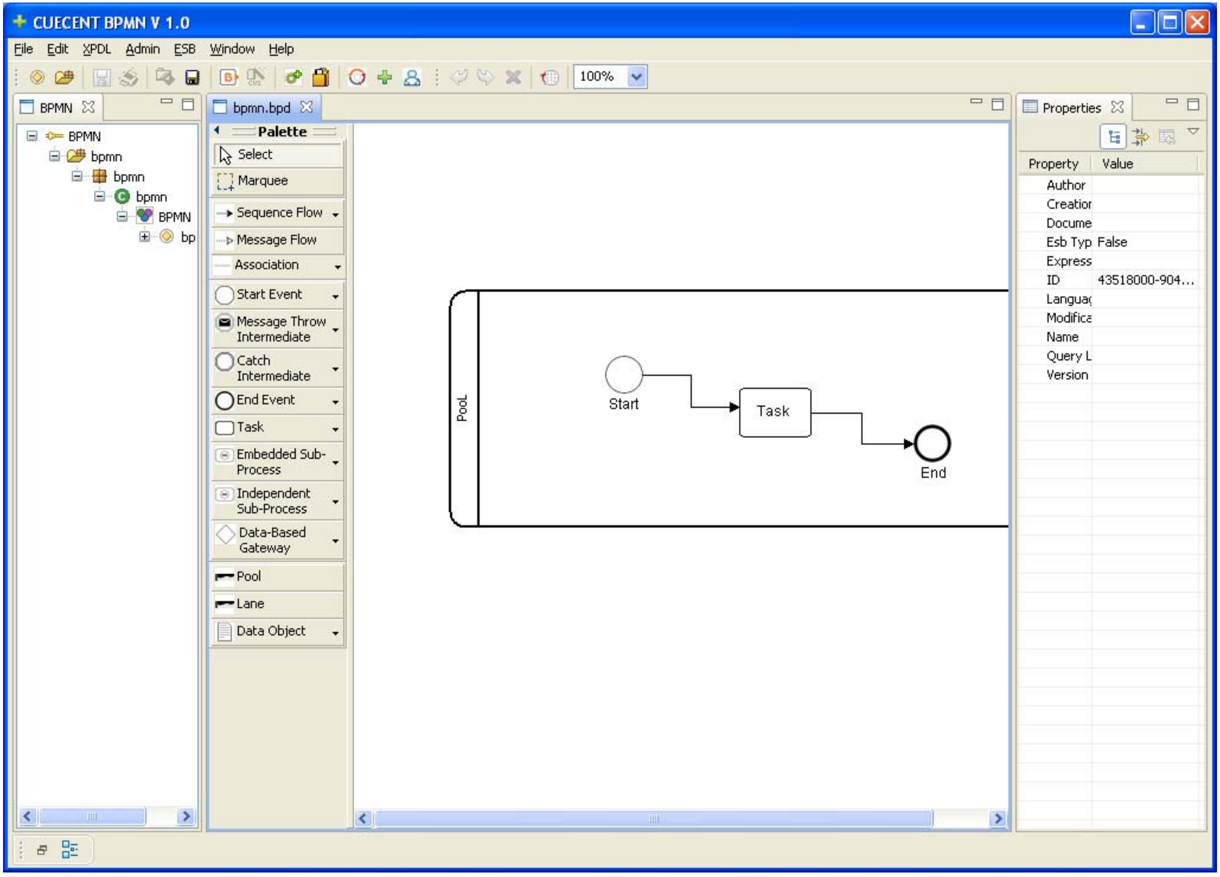


Figure 7 BPD File Name Entry Dialog box

If you click OK, a default BPD will appear as shown in figure 8.



Property	Value
Author	
Creatior	
Docume	
Esb Typ	False
Express	
ID	43518000-904…
Languag	
Modifica	
Name	
Query L	
Version	

Figure 8 BPMN screen with default flow

You should also be able to see the Palette as in the figure. That's it ! You have created a BPD. The hierarchy described can be used to store as many projects as possible and any combination of other elements – there can be any number of BPDs under a BPMN builder, but we suggest you group all related BPDs together.

If you click the "+" next to the BPMN builder, you can also see the file name as displayed next

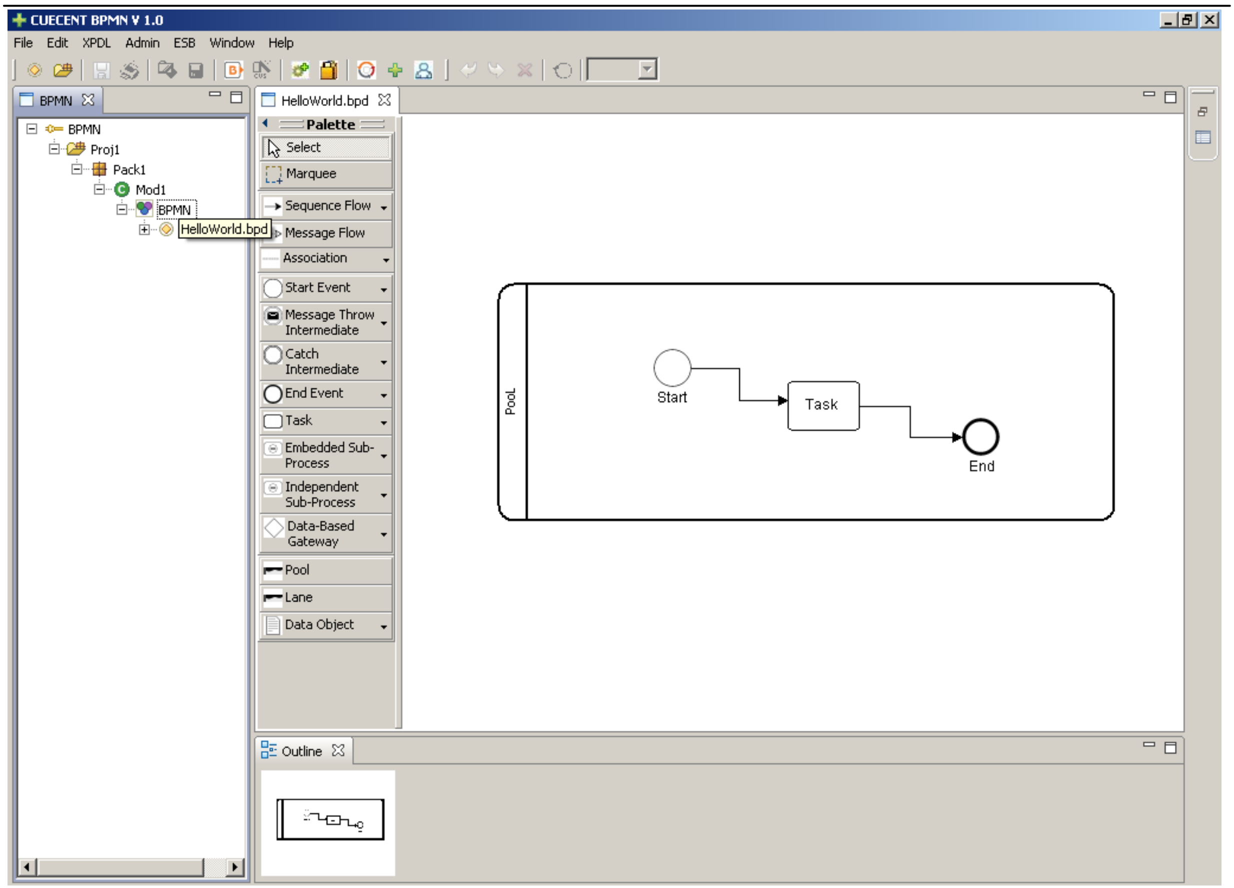


Figure 9 BPMN screen with file name displayed

You can of course draw your own “custom” flows – let’s do this next. Start by deleting the sample: Click on the pool as below – you should see the pool highlighted (we’ll see what exactly a “pool” is a little later):

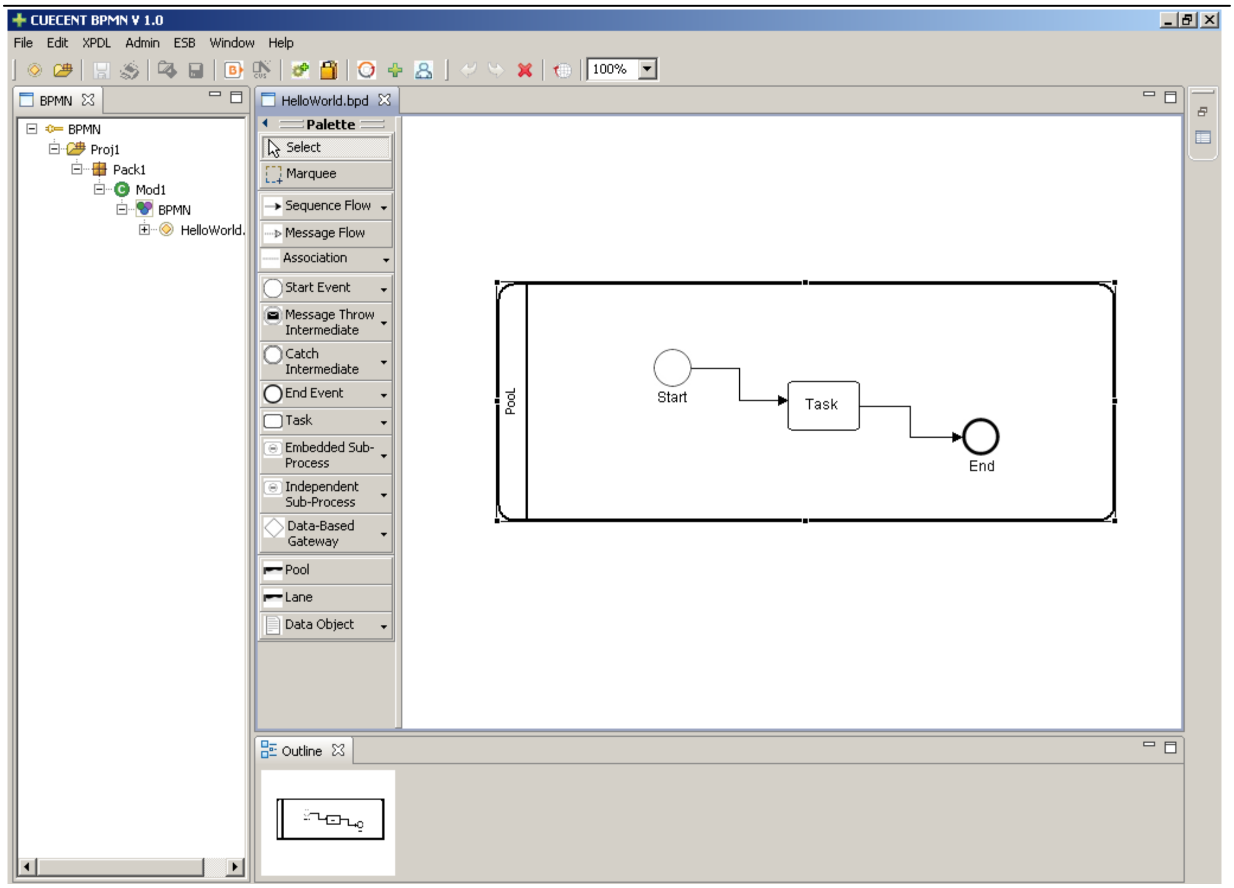


Figure 10 BPMN screen with pool selected

Click on the delete key on the keyboard to delete the existing diagram. Now you should see an empty canvas as shown in figure 11.

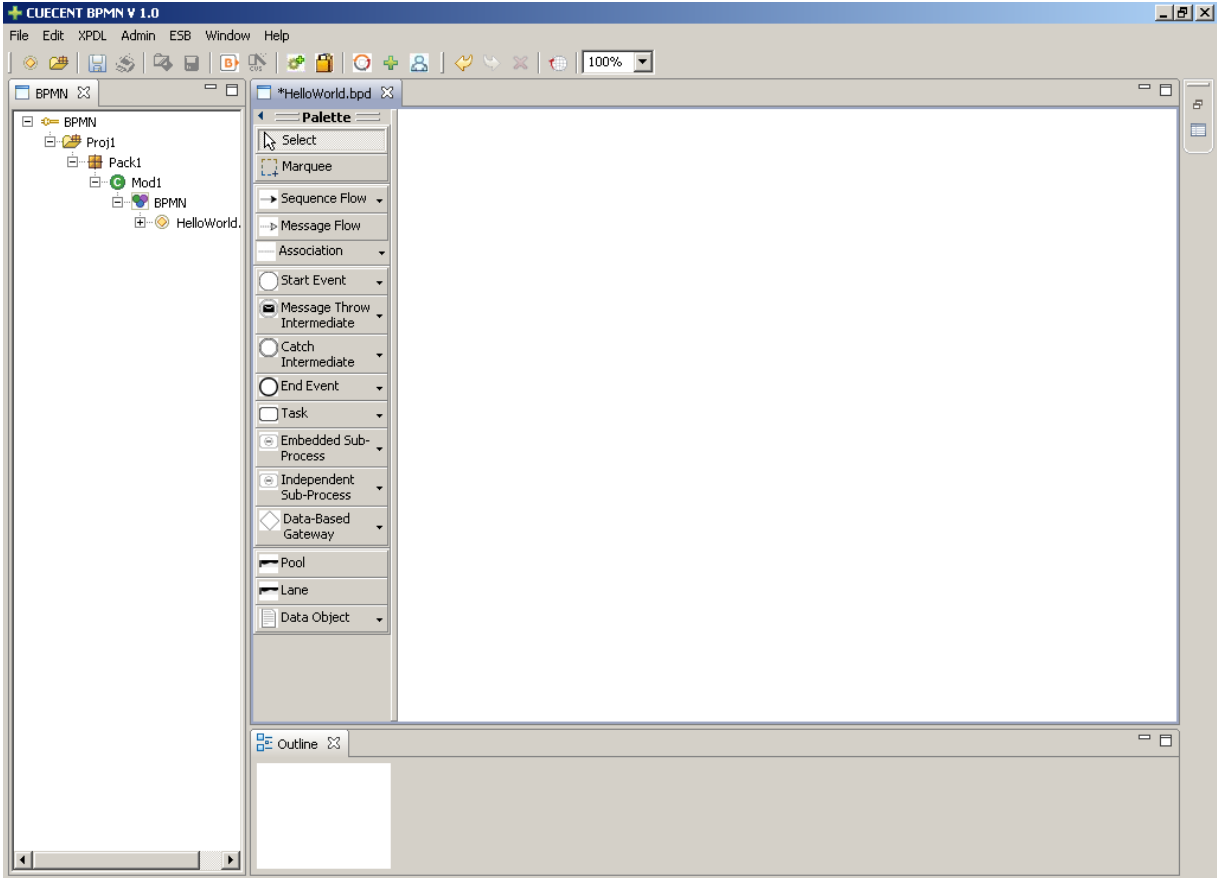


Figure 11 Empty BPMN screen

Now let's draw a simple BPD with a "Start" element, a "Task" and an "End" element and sequence flow between these elements. We'll see what these elements are a little later, but these have intuitively clear meanings as of now. This is a very basic example, but serves to understand the UI to begin with.

Click on the "Start" element on the palette and "drop" on the canvas – this is what you should see as shown in figure 12.

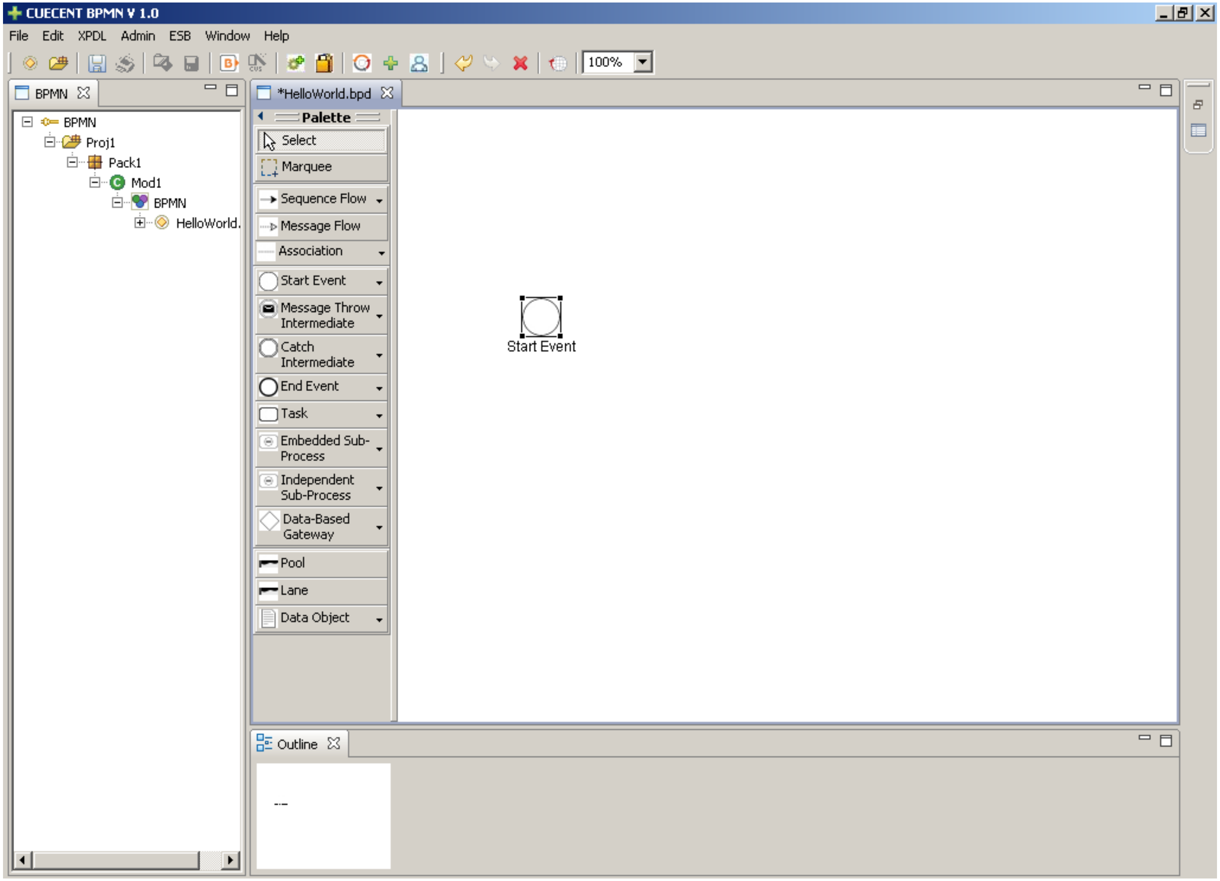


Figure 12 BPMN screen with start figure

Next, click on the “Task” button on the palette and drop on the canvas. This is what you would see as shown in figure 13.

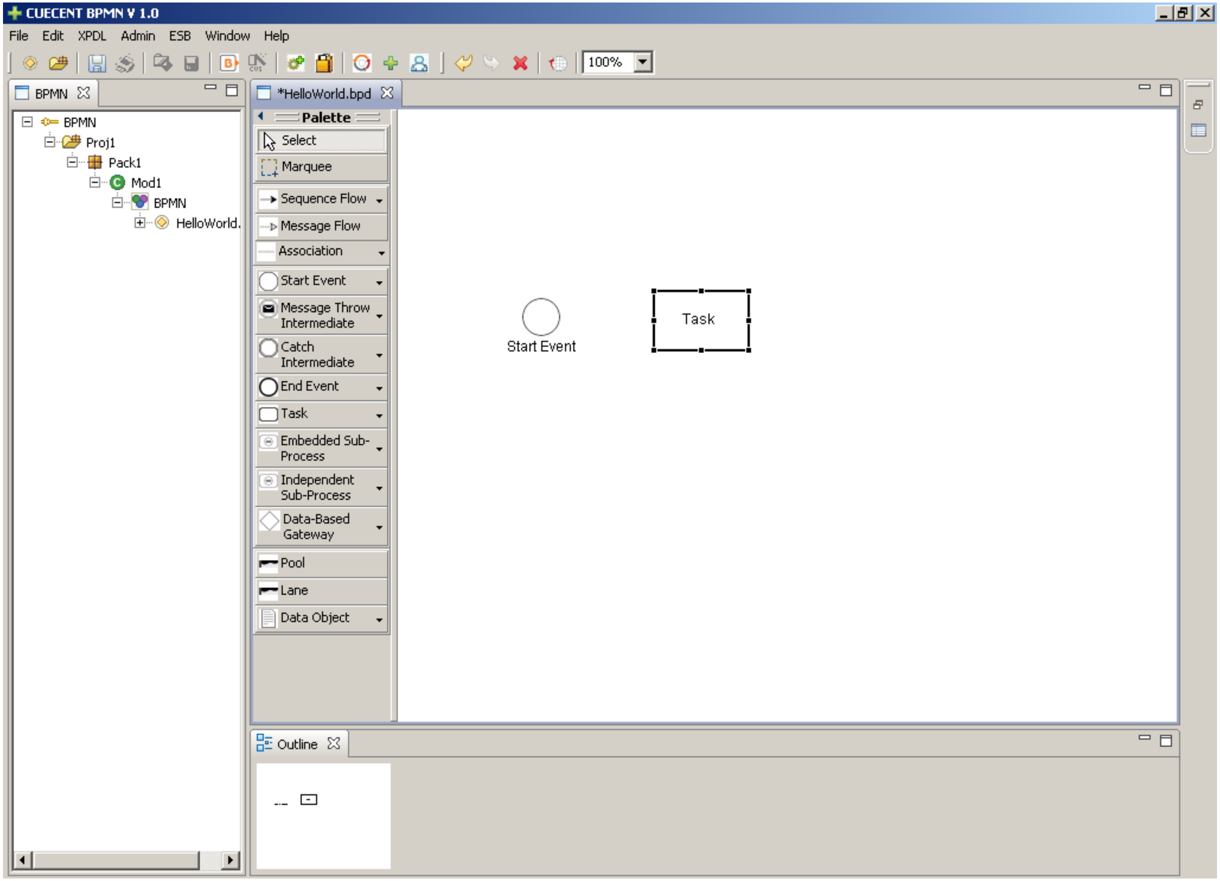


Figure 13 BPMN screen with Task figure also

Next, click on the “End” button on the Palette and drop on the canvas. This is what you should see as shown in figure14.

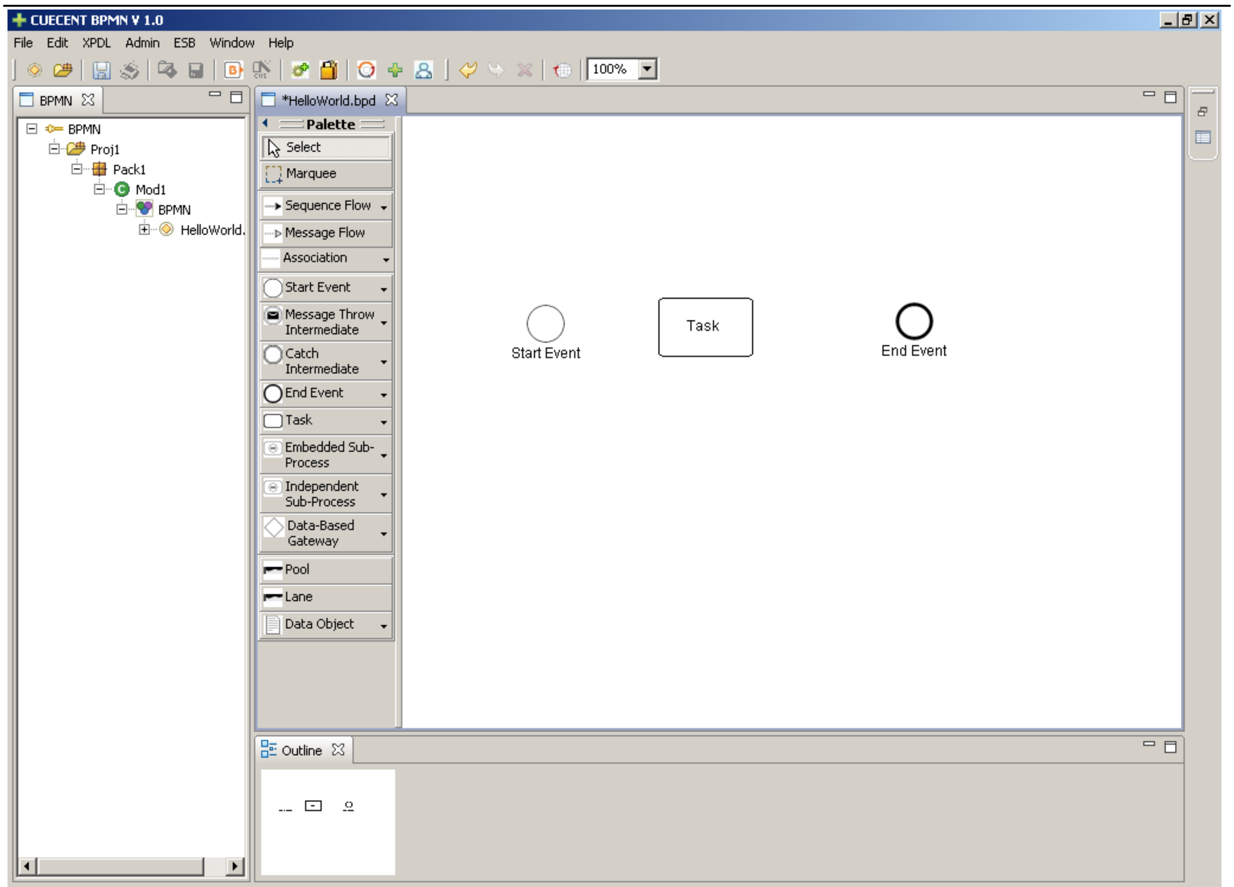


Figure 14 BPMN screen with end node also

Next we'll add sequence flow:

Click the "Sequence Flow" button on the palette and click first on the "Start" element – then drag and drop on the "Task" element. You can continue to draw sequence flows until the "Select" button is clicked. So let's draw a link between the "Task" and the "End" element as well. This is what you'll see as shown in figure 15.

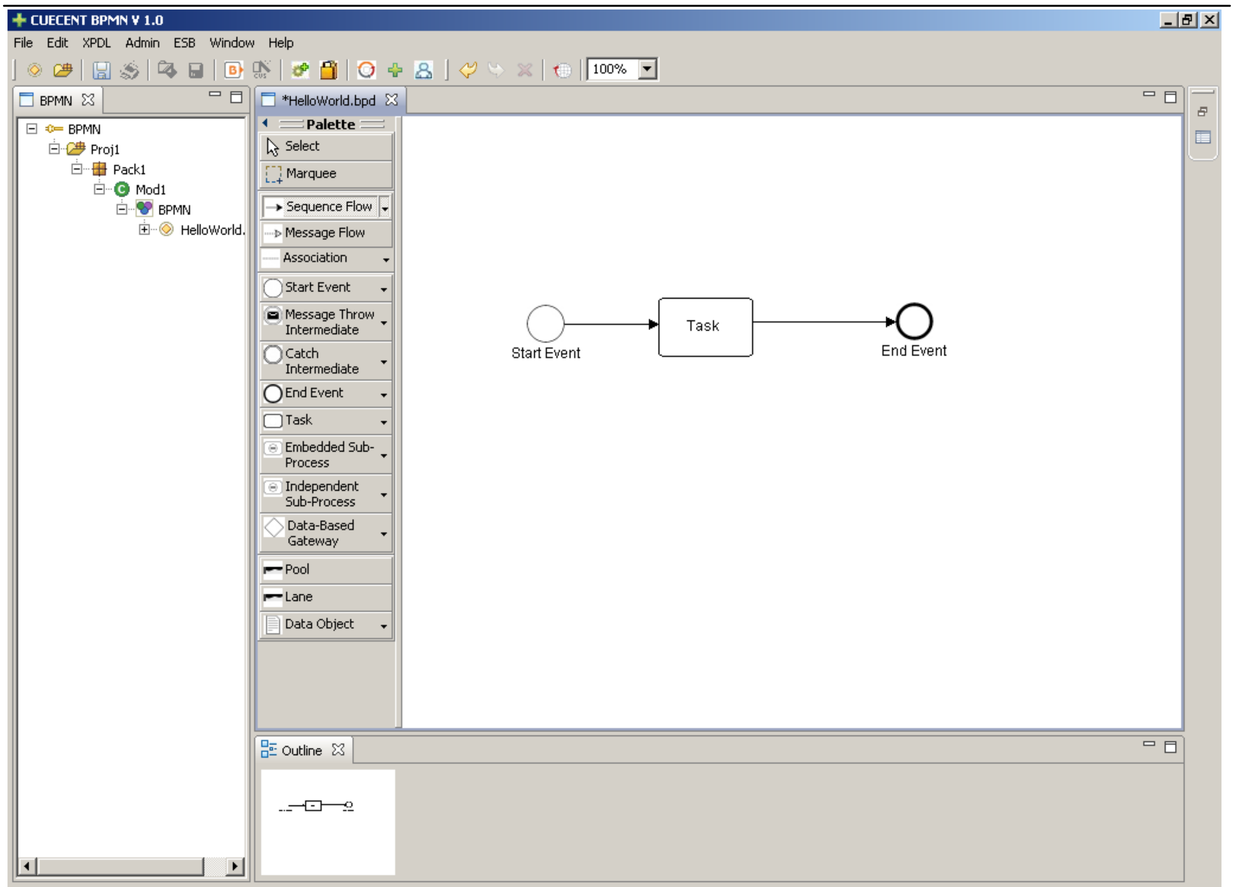


Figure 15 BPMN with sequence flow added

Notice that the cursor has changed. Next click on the “Select” item on the palette – the cursor returns to normal. Notice also the asterisk on top of the tab – this indicates that the file is “dirty” and needs to be saved. You can do this by simply pressing <ctrl-s> - you can also click the Save icon or the Save item under the File menu. Now this is what you should see as shown in figure 16.

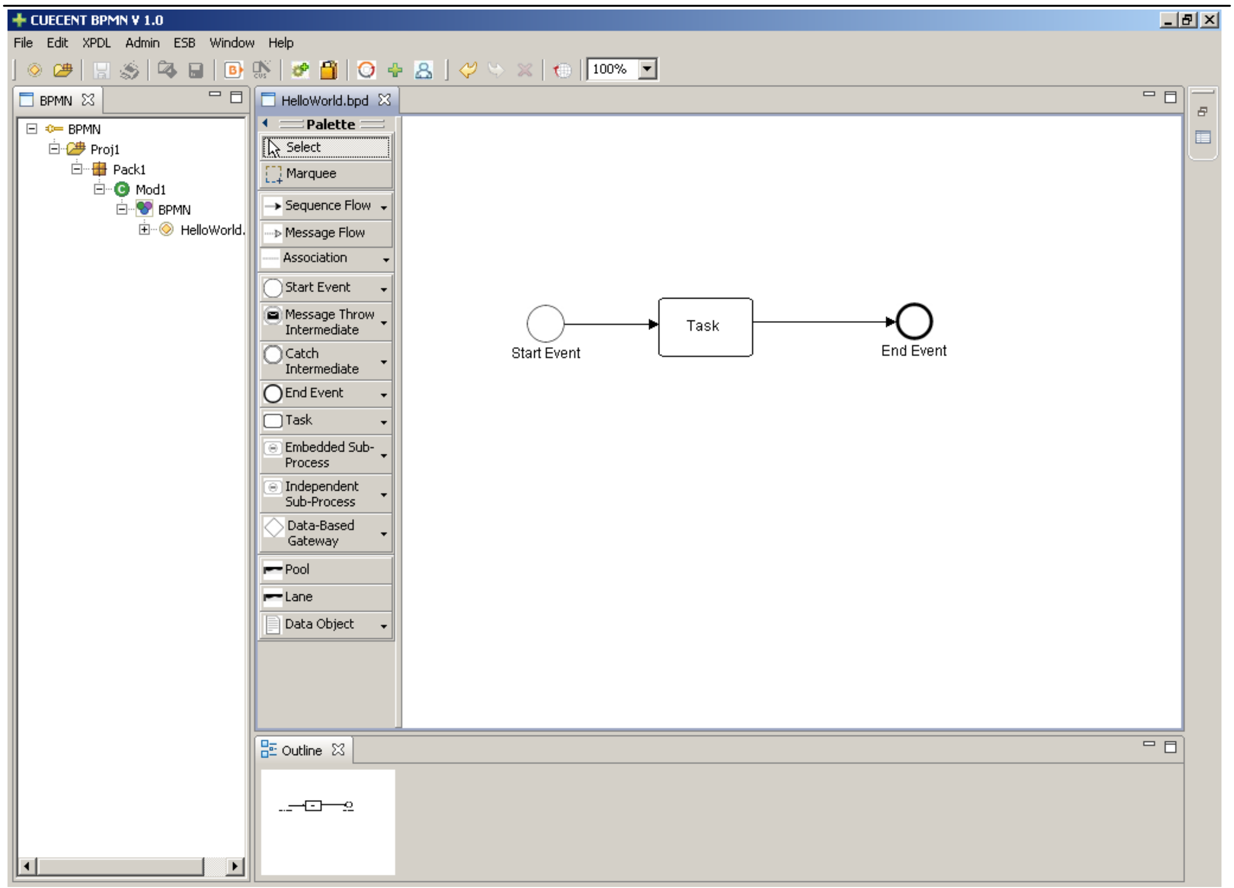


Figure 16 BPMN with BPD file in the saved state

That was a quick tour of the BPMN UI and should give you a “feel” for the UI. The next section discusses finer points of the UI that you need to be aware of. This should give you a broader appreciation of the UI.

2.2 Palette

The palette holds various templates that can be dropped on the canvas. The Select button is clicked to change mode from Drawing to Selection mode. You can click on objects by clicking Select and then the object or link.

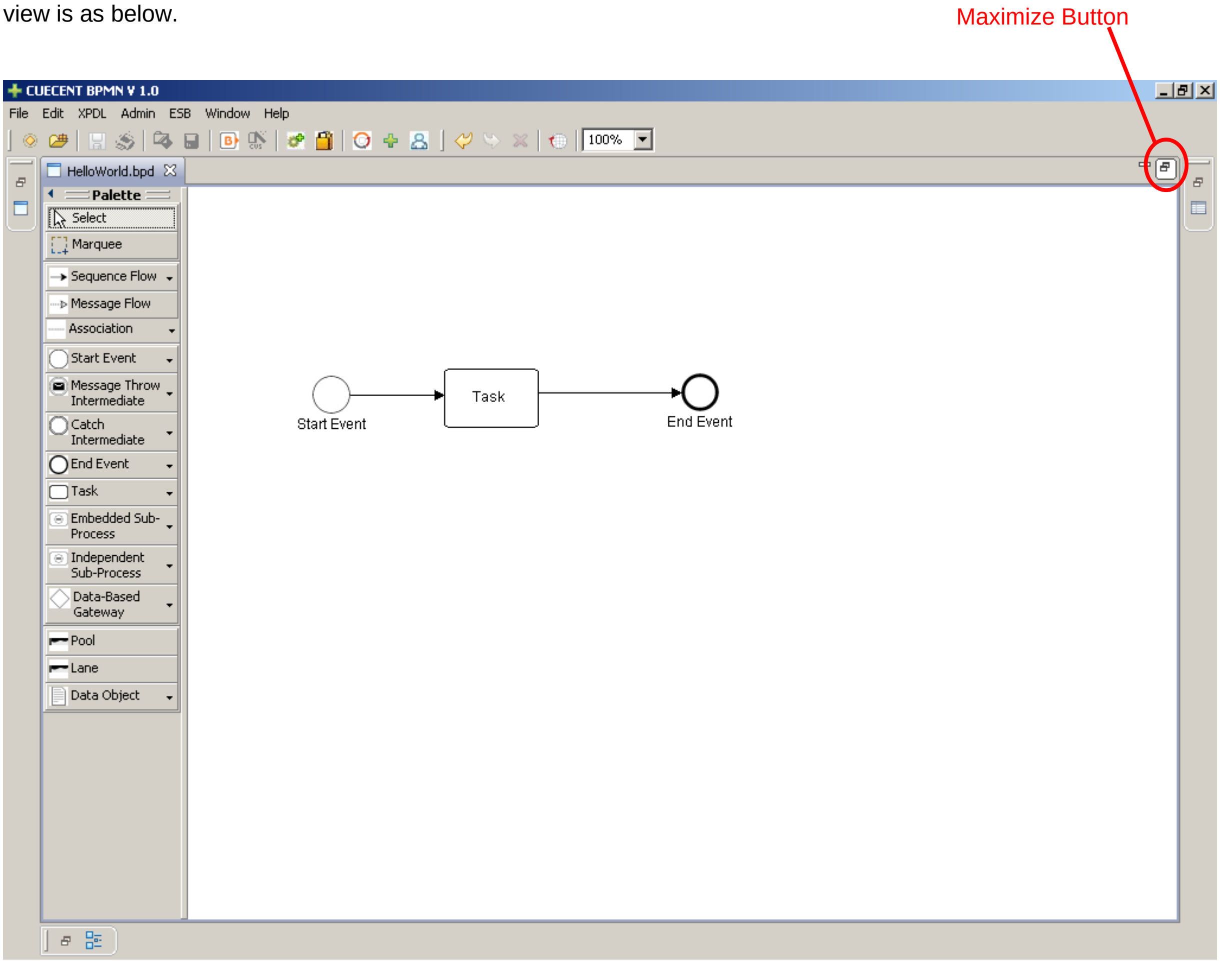
The marquee tool is used to group a selection of objects on the canvas

You can dock the palette by clicking on the small arrow on top of it. You can also drag the palette to either the left or right as per convenience

Some palette items have further drop down elements – these have intuitive meaning. These elements work the same way as earlier. The palette remembers the last choice which is highlighted after each selection.

2.3 Canvas

The canvas may be maximized for convenience by clicking on the “Maximize” rectangle on the top right corner. This view is as below.



Maximize Button

Figure 17 BPD with “Maximize” button highlighted

The maximize rectangle changes to “Restore” now and can be clicked to restore the earlier state.

2.4 Tree View

The Tree View was discussed earlier and has intuitive meaning.

2.5 Properties View

Notice that the Properties view is minimized when the Canvas is maximized. It can be restored by clicking on the button highlighted below.

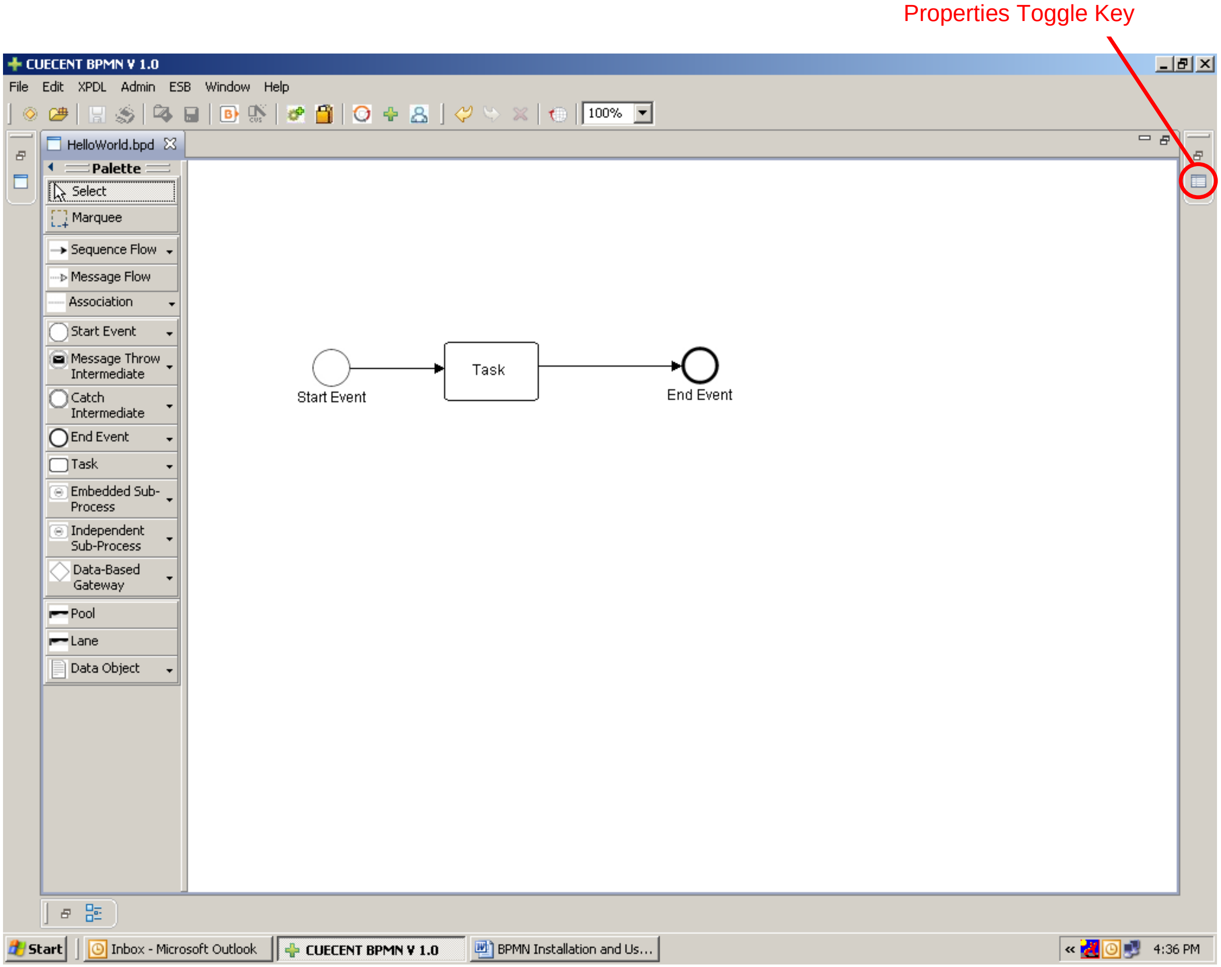
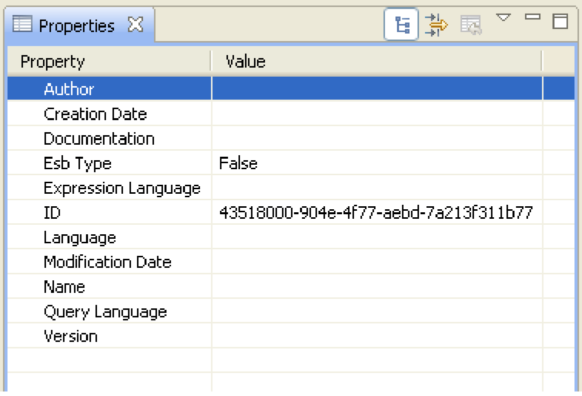


Figure 18 BPD with properties chooser highlighted

This works as a toggle and this is how the screen looks when clicked. You can also use the right click context menu on an object to bring up the properties as also by double clicking the object or using the Properties option on the “Edit” menu.



Properties

Property	Value
Author	
Creation Date	
Documentation	
Esb Type	False
Expression Language	
ID	43518000-904e-4f77-aebd-7a213f311b77
Language	
Modification Date	
Name	
Query Language	
Version	

Figure 19 BPD with properties view displayed

2.6 Snap On Grid

This is another cool feature. This lets you snap objects to a grid if clicked and looks this way when clicked – this is also a toggle.

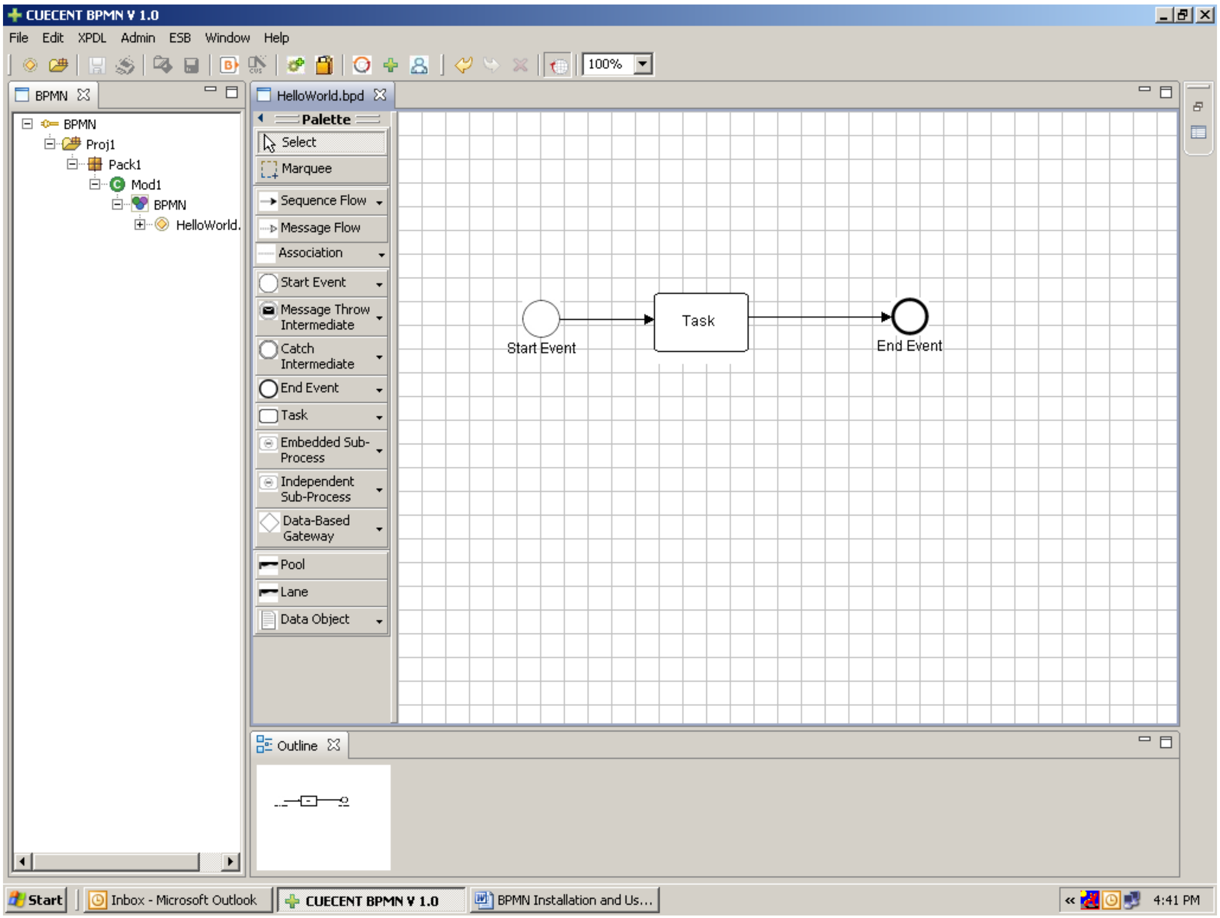


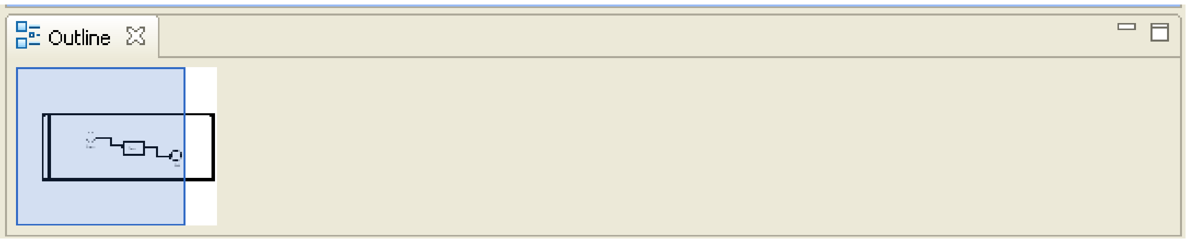
Figure 20 BPD with Snap-On Grid highlighted

2.7 Zoom Tool

This tool facilitates you to zoom-in to the level that you need.

2.8 Outline View

This view provides a miniature view of the BPD and is useful when the BPD is larger than can fit on one screen. The desired portion of the BPD can be navigated to by dragging here.



2.9 How do I Delete, Undo or Redo?

You can delete an element by clicking on it and clicking the delete key. You can also click the delete (Cross) button on the Toolbar or the Delete option on the Edit menu or delete on the right click context menu.
You can undo a previous action by pressing <ctrl-z>. You can also click the undo item on the toolbar or the undo item on the right click context menu. You can redo by pressing the redo item on the toolbar or the redo item on the right click context menu. These are also available on the Edit Menu

2.10 How do I Reset?

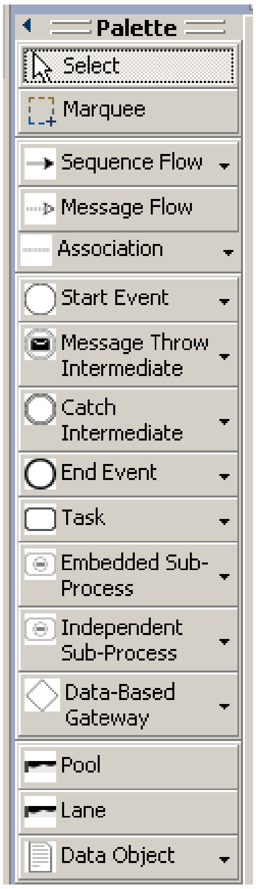
This is useful if you have accidentally pressed a wrong key and would like to restore the window to the original state. This is done by clicking the “Reset Perspective” option under the “Window” menu.

3 BPMN UI in Detail

The design of the BPMN UI follows the Model-View-Controller design paradigm as does the GEF framework. The environment for development, as already discussed, is Eclipse 3.3 with Draw2d and GEF over the native SWT.

3.1 Palette

The following groups of items belong to the Palette.



3.1.1 Group 1

- Selection Tool: The selection tool is used to choose objects in a BPD. Just click the tool and click on the object
- Marquee Tool: This tool is used to select groups of objects

3.1.2 Group 2 (Connection Objects Related)

- Sequence Flow: This tool is used to render various types of connections between objects and has the following sub items:
    - Normal Sequence Flow
    - Conditional Flow

  - Default Flow

- Message Flow: This is used to render message connections between objects as per BPMN syntax. Message flow typically is across pools and cannot be rendered between two objects in the same pool

- Association: An association is used to associate information and artifacts with Flow objects (Flow objects are discussed below)

3.1.3 Group 3 (Start Events)

Start Events are used to start flows and have the following sub types:

- None start event
- Message start event
- Timer start event
- Conditional
- Signal
- Multiple

These are described in detail in Sec 5.2

3.1.4 Group 4 (Throw and Catch Intermediate Events)

Throw and Catch intermediate events are used in between flows and have the following sub types

Throw Intermediate Events

- Message
- Link
- Signal
- Multiple

Catch Intermediate Events

- None
- Message
- Timer
- Conditional
- Link
- Signal
- Multiple

Intermediate Events are described in detail in Sec 5.4

3.1.5 Group 5 (End Events)

End events are used to terminate flows and have the following sub types:

- None
- Message
- Error
- Signal
- Terminate
- Multiple

End events are described in detail in Sec 5.3

3.1.6 Group 6 (Tasks)

Tasks represent typically atomic actions and have the following sub types:

- None Task
- Standard Loop Task
- Multi Instance Loop Task
- Compensation Task

Tasks are described in detail in Sec 5.5

3.1.7 Group 7 (Embedded Sub Process)

The Embedded Sub Process is a compound activity that is defined as a flow of other activities. It can be opened to show the inner details of the sub activity or also be collapsed. It has the following sub types:

- Embedded Sub Process
- Standard Loop Embedded Sub Process
- Multi Instance Embedded Sub Process

Embedded Sub Processes are described in detail in Sec 5.6.

3.1.8 Group 8 (Independent Sub Process)

An Independent Sub Process is similar to an Embedded Sub Process, except that it is a “Reusable” process and the details of the sub process are encapsulated in a separate BPD file and is referenced from the calling process using the Diagram attribute

It has the following sub types:

- Independent Sub Process
- Loop Independent Sub Process
- Multi Instance Independent Sub Process
- Compensation Independent Sub Process

Independent Sub Processes are described in detail in Sec 5.7.

3.1.9 Group 9 (Gateways)

Gateways are elements that are used to control Sequence Flow interaction as these converge and diverge within a Process. These have the following sub types:

- Data Based

- Event Based
- Complex
- Inclusive
- Parallel

Gateways are described in detail in Sec 5.8

3.1.10 Group 10 (Pools and Lanes)

A pool is a container of a flow and is described in detail in Sec 5.9.

3.1.11 Group 11 (Artifact Objects)

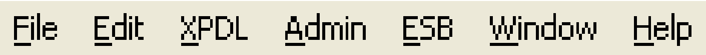
An Artifact object is used to provide information about a process that is not directly related to the Sequence or Message Flow of the process and includes the following:

- Data Object
- Group
- Annotation
- Off Page Connector

Artifacts are described in detail in Sec 5.11

3.2 Menu

The menu has certain miscellaneous functions used by the editor: Zoomed view of the menu bar appears as shown below.



3.2.1 File

- New
- Open
- Save
- Print
- Import
- Exit

3.2.2 Edit

- Undo
- Redo
- Delete
- Properties

The above have intuitive meaning as in any GUI application.

3.2.3 XPDL

This menu item has operations related to XPDL and consists of the following choices

- Open XPDL (Currently not implemented)
- Save XPDL

3.2.4 Admin

This menu item has only CVS functionality – basically to check in and out diagrams

3.2.5 ESB

This menu has items related to Enterprise Service Bus Functionality and has the following choices:

- Listener
- Dispatcher

3.2.6 Window Menu

This menu item has the following functionality:

- Reset Perspective
- Hide Editors
- Next Editor
- Previous Editor

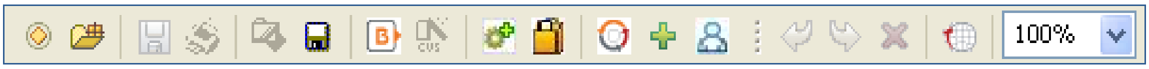
These functions are related to the GUI view

3.2.7 Help/About Menu

This menu has the Help (Currently not implemented) and the usual about menu

3.3 Toolbar

The Toolbar also has certain functionality as below and complements the menu. The entities in the Toolbar are as below:

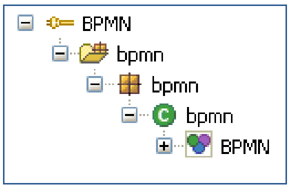


- Create New BPD File
- Open existing BPD

- Save a BPD file
- Print BPD file
- Open XPDL file
- Save as XPDL file
- Properties Icon
- CVS Configuration icon
- Listener and Dispatcher related icons (ESB Stuff)
- Reset Perspective
- Hide/Show Editors
- About Icon
- Undo/Redo/Delete icons
- Grid enabling icon
- Zoom Tool icon

3.4 Tree View

The Tree view is used to create a hierarchical ordering of entities such as: Project, Module, Package and Builder. The Builder has all the BPD files under it. The user interacts with the tree view to create a Project first and then a Package and further a Module and then selects a Builder (BPMN). Then the user right clicks the builder to create a BPD file. The same is now available for drawing BPMN diagrams using the Palette.



3.5 Canvas

The canvas is used to draw the BPD diagrams.

4 Advanced Features of the BPMN UI

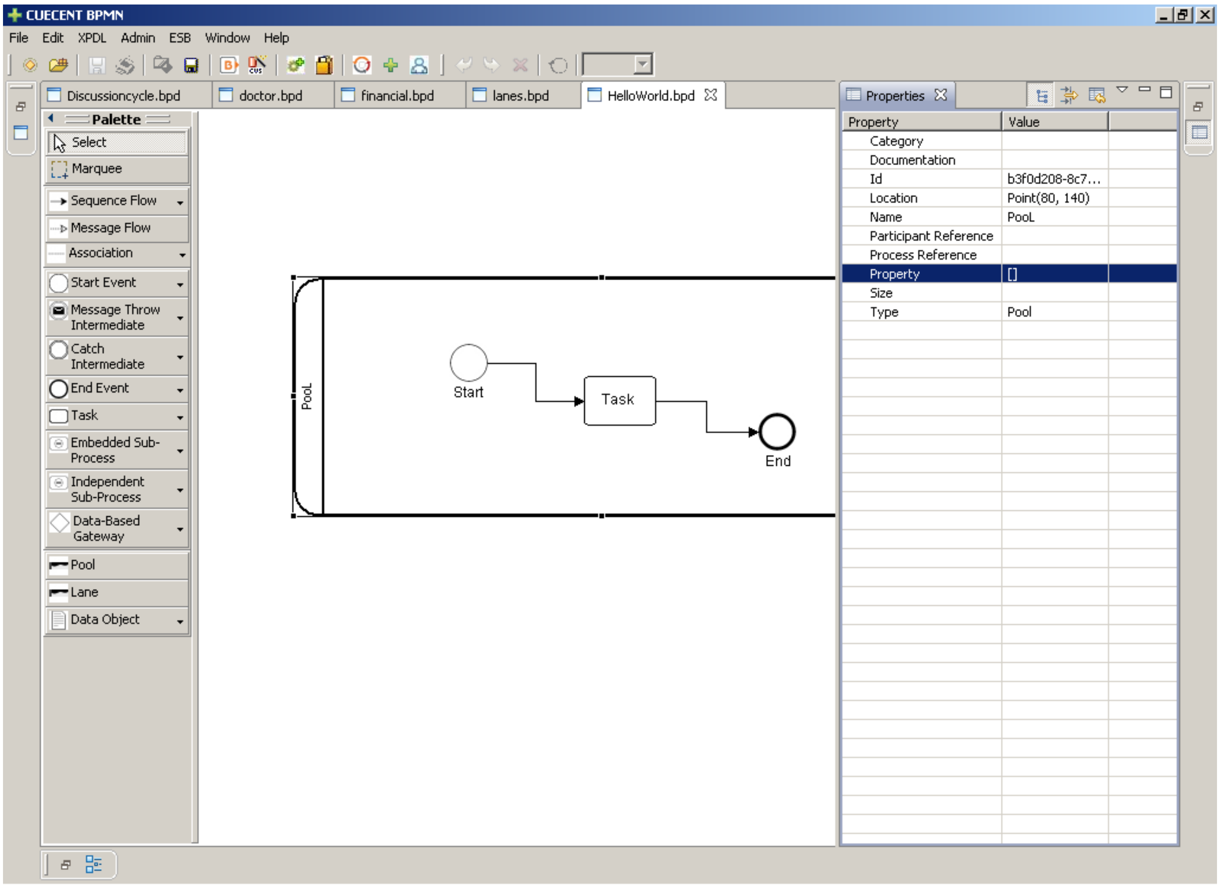
This section has a detailed explanation of all BPMN elements including properties attached to them if any. For complete details the reader is referred to the BPMN 1.1 specification.

4.1 Summary of Attributes Common to all Property Views

This section explains attributes and entities common to properties views. First of all, all elements in the UI have specific properties and these are visible and settable from the properties view.

Properties at Pool Level

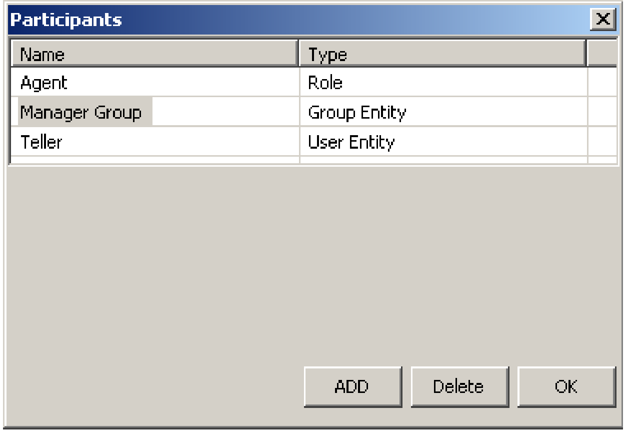
The properties view at the pool level is displayed below and has the following elements as below:



Property	Value
Category	
Documentation	
Id	b3f0d208-8c7...
Location	Point(80, 140)
Name	PooL
Participant Reference	
Process Reference	
Property	[]
Size	
Type	Pool

Figure 21 BPD with POOL level properties

- Category: This is used during grouping
- Documentation: Notes related
- Id: Unique ID of the object (Pool)
- Participant Reference: This is a property that lets the user define Users/Roles/Groups as below:

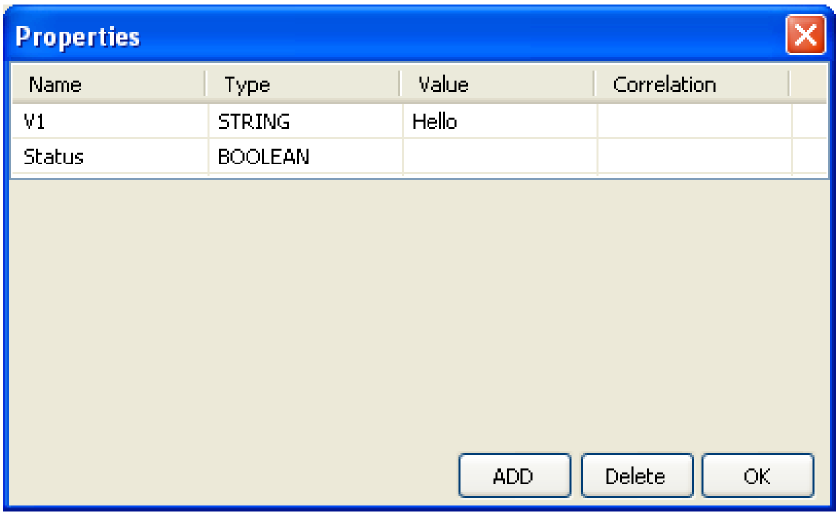


Name	Type
Agent	Role
Manager Group	Group Entity
Teller	User Entity

Figure 22 Participants dialog

- Property

This element is displayed below:



Name	Type	Value	Correlation
V1	STRING	Hello	
Status	BOOLEAN		

Figure 23 Properties Dialog box

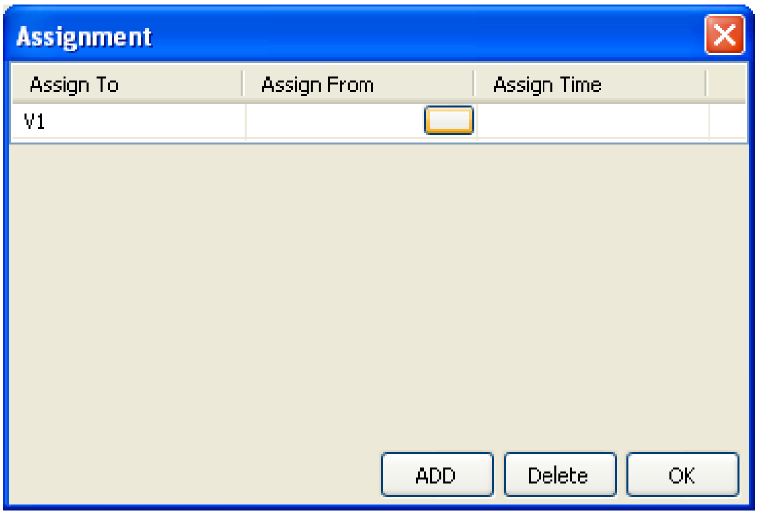
These are actually variables that may be created on the fly. These are assigned values at runtime in different UI elements. E.g. one may create variables “V1” of type integer and “status” of type Boolean and these may be used inside of a task inside of the pool. This is pictured above

- Size: This is related to the physical size of the Pool
- Type: Item type (“Pool”)

Other Common Properties:

Assignments

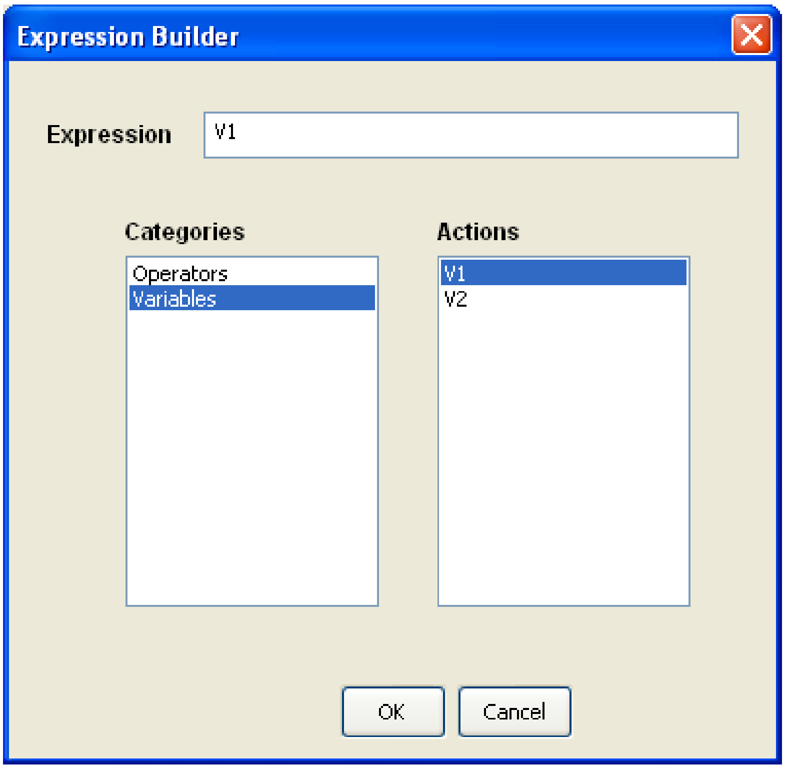
Assignments are common across all UI elements and are pictured below:



Assign To	Assign From	Assign Time
V1		

Figure 24 Assignment dialog

The sources for assignment are the Pool level variables and the Right hand side of the assignment is created using an Expression builder (also displayed below)

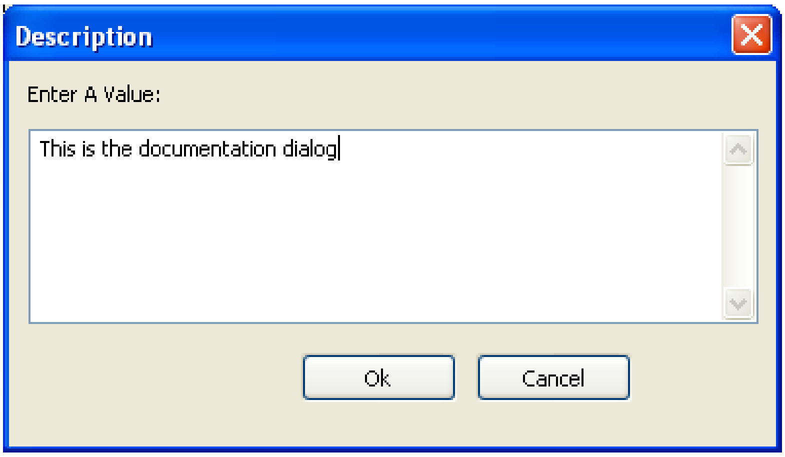


Expression: V1

Figure 25 Expression builder dialog

Documentation

This is a UI as below and is used to add documentation to UI elements:



This is the documentation dialog

Figure 26 Documentation dialog

Category

This is used to group elements

IDs and UUIDs

All UI elements have IDs and the UUID scheme is used to ensure these are unique.

4.2 Start Events

The list of Start Events is as below:

- None Start Event
- Message Start Event
- Timer Start Event
- Conditional Start Event
- Signal Start Event
- Multiple Start Event

The Start Event is commonly used to start the BPMN diagram. Properties for each type of event are as below:

None Start Event

Please refer to section 5.1 for None Start Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “Start” and cannot be changed.
ID: The UUID of the element
Name: The logical name of the event

Message Start Event

Please refer to section 5.1 for Message Start Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “Start” and cannot be changed

ID: The UUID of the element

Name: The Logical name of the element

Trigger: Event Details Type: Set to “Message” and cannot be changed

- Implementation: Combo with choices – Web Service, Unspecified, and Other
- Message Reference
  - From: The element from which the message flow originates
  - Name: Logical name of the element
  - Properties: Formal and Actual parameters
  - To: The element to which the message is targeted

Web Service: GUI that pops up to collect Web Service details if Implementation is “Web Service”

Timer Start Event

Please refer to section 5.1 for Time Start Event properties like Assignment, Category, and Documentation.

Event Type: Set to “Start” and cannot be changed

ID: UUID of the element

Name: Logical name of the element

Trigger: Event Details Type: Set to “TIMER” and cannot be changed

- Time Cycle: Value of the Timer element (E.g. 2 seconds, or 1 hour etc)
- Time Date: Date and Time of the Timer
- Time Unit: Units of the Timer element (Seconds, Minutes, Hours, Days, Months, Years)

Conditional Start Event

Please refer to section 5.1 for Conditional Start Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “Start” and cannot be changed

ID: The UUID of the element

Name: Logical name of the element

Trigger

- Conditional Reference: The conditional reference
- Event Details Type: This is set to “CONDITIONAL” and cannot be changed

Signal Start Event

Please refer to section 5.1 for Signal Start Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “START” and cannot be changed

ID: The UUID of the element

Name: The logical name of the element

Trigger

- Event Details Type: This is set to “SIGNAL” and cannot be changed

Signal Reference: Name: Logical name for Signal Reference

Properties: Actual to Formal Parameters association GUI

Multiple Start Event

Please refer to section 5.1 for Multiple Start Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “Start” and cannot be changed

ID: The UUID of the element

Name: The logical name of the element

4.3 End Events

The list of End Events is as below:

- None End Event
- Message End Event
- Error End Event
- Cancel End Event
- Signal End Event
- Terminate End Event
- Multiple End Event

The End Event is commonly used to end a BPMN flow – there can be more than one End Event in a BPMN diagram.

The Properties for each type of End Event are as below:

None End Event:

Please refer to section 5.1 for None End Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “End” and cannot be changed

ID: The UUID of the element

Name: Logical name of the element

Message End Event:

Please refer to section 5.1 for Message End Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “End” and cannot be changed

ID: The UUID of the element

Name: The logical name of the element

Trigger Details:

Event Details Type: This is set to “MESSAGE” and cannot be changed

Implementation: Can be “Web Service”, “Unspecified” or “Other”

Message Reference

Name: Logical name of the element

From: Message Source from

To: Message Target to

Properties: Pl refer to Sec 5.1

Web service: Dialog that pops up in case the Implementation is “Web Service”. Asks for URL, URI and Operation name.

Error End Event

Please refer to section 5.1 for Error End Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “End” and cannot be changed

ID: The UUID of the element

Name: The logical name of the element

Trigger Details:

Error Code: Error code details

Event Details Type: This is set to ERROR and cannot be changed

Web Service: Web Service details – dialog that asks for URL, URI and Operation Name

Signal End Event

Please refer to section 5.1 for Signal End Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “End” and cannot be changed

ID: The UUID for the element

Name: The logical name for the element

Trigger

Event Details Type: This is set to “SIGNAL” and cannot be changed

Signal Reference

Name: Name of the Signal

Properties: Properties (Pl refer to Sec 5.1)

Web Service: Dialog that asks for URL, URI and Operation name

Terminate End Event

Please refer to section 5.1 for Terminate End Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “End” and cannot be changed

ID: UUID of the element

Name: Logical name of the element

Multiple End Event

Please refer to section 5.1 for Multiple End Event properties like Assignment, Category, and Documentation.

Event Type: Set to “End” and cannot be changed

ID: UUID of the element

Name: Logical name of the element

4.4 Intermediate Events

Intermediate Events are further classified as “Throw” and “Catch” events. The list of Intermediate Events is as below:

Intermediate Throw Events:

- Message Throw Intermediate Event
- Link Throw Intermediate Event
- Signal Throw Intermediate Event
- Multiple Throw Intermediate Event

Intermediate Catch Events:

- None Intermediate Catch Event
- Message Intermediate Catch Event
- Timer Intermediate Catch Event
- Error Intermediate Catch Event
- Cancel Intermediate Catch Event
- Conditional Intermediate Catch Event
- Link Intermediate Catch Event
- Signal Intermediate Catch Event
- Multiple Intermediate Catch Event

The Properties for each type of Throw Intermediate Event are as below:

Message Throw Intermediate Event

Please refer to section 5.1 for properties like Assignment, Category, and Documentation.

Event Type: This is set to “Intermediate” and cannot be changed

Extended Attribute:

  Group: This is an extended attribute

ID: The UUID of the element

Name: The logical name of the element

Target: The Target ID

Trigger:

  Event Details Type: This is set to “MESSAGE” and cannot be changed

  Implementation: Can be one of Web Service/Unspecified/Other

  Message Reference:

    From: Message Source ID

Name: Name of Message
Properties: Refer Sec 5.1
To: Message Target ID

Web Service: This field is applicable only if implementation is “Web Service” and requires the following fields: WSDL UTL, Name space URI and Operation Name

Link Throw Intermediate

This is used as a connecting element across page boundaries

Please refer to section 5.1 for Link Throw Intermediate Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “Intermediate” and cannot be changed

Extended Attribute:

Group: This is an extended attribute

ID: The UUID of the element

Name: The logical name of the element

Target: Target ID

Trigger:

Event Details Type: This is set to “LINK” and cannot be changed
Name: The logical name of the trigger

Signal Throw Intermediate

Please refer to section 5.1 for Signal Throw Intermediate Event properties like Assignment, Category, Documentation, Properties.

Event Type: This is set to “Intermediate” and cannot be changed

Extended Attribute:

Group: This is an extended attribute

ID: The UUID of the element

Name: The logical name of the element

Target: (Not currently implemented)

Trigger:

Event Details Type: This is set to “SIGNAL” and cannot be changed
Signal Reference:
Name: The logical name of the signal

Multiple Throw Intermediate

Please refer to section 5.1 for Multiple Throw Intermediate Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “Intermediate” and cannot be changed

ID: The UUID of the element

Name: The Logical name of the element

Target: (Not currently implemented)

The Properties for each type of Catch Intermediate Event are as below:

Catch Intermediate None Event

Please refer to section 5.1 for Catch Intermediate None Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “Intermediate” and cannot be changed
ID: The UUID of the element
Name: The Logical name of the element
Target: (Not currently implemented)

Message Catch Intermediate Event

Please refer to section 5.1 for Message Catch Intermediate Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “Intermediate” and cannot be changed
Extended Attribute:
    Group: This is an extended attribute
ID: The UUID of this element
Name: The Logical name of the element
Target: (Not currently implemented)
Trigger:
    Event Details Type: This is set to “MESSAGE” and cannot be changed
    Implementation: This is one of “Web Service”, “Unspecified” and “Other”
    Message Reference:
        From: Message from
        To: Message To
        Name: The Logical name of the message
        Properties: Pl refer to Sec 5.1
Web Service: This appears on the UI if Implementation is “Web Service” and requires the following fields: WSDL URL: Namespace URI, Operation Name.

Timer Catch Intermediate Event

Please refer to section 5.1 for Timer Catch Intermediate Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “Intermediate” and cannot be changed
Extended Attribute:
Group: This is an extended attribute
ID: The UUID of the element
Name: The Logical name of the element
Target: (Not currently implemented)
Trigger:
    Event Details Type: This is set to “TIMER” and cannot be changed
    Time Cycle: The value of the Time attribute

Time Date: Date attribute

Time Unit: Can be “Second”, “Minute”, “Hours”, ”Days”, “Months” or “Years”

Error Catch Intermediate Event

Please refer to section 5.1 for Error Catch Intermediate Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “Intermediate” and cannot be changed

Extended Attribute:

Group: This is an extended attribute

ID: The UUID of the element

Name: The Logical name of the element

Target: (Currently not implemented)

Trigger:

Error Code: The error code

Event Details Type: This is set to “ERROR” and cannot be changed

Conditional Catch Intermediate Event

Please refer to section 5.1 for Error Catch Intermediate Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “Intermediate” and cannot be changed

Extended Attribute:

Group: This is an extended attribute

ID: The UUID of the element

Name: The Logical name of the element

Target: (Not currently implemented)

Trigger:

Conditional Reference: The conditional reference

Event Details Type: This is set to “CONDITIONAL” and cannot be changed

Link Catch Intermediate Event

Please refer to section 5.1 for Link Catch Intermediate Event properties like Assignment, Category, and Documentation.

This is used as a connector across pages

Event Type: This is set to “Intermediate” and cannot be changed

Extended Attribute:

Group: This is an extended attribute

ID: The UUID of the element

Name: The Logical name of the element

Target: Not currently implemented

Trigger:

Event Details Type: This is set to “LINK” and cannot be changed

Name: Logical name of the trigger

Signal Catch Intermediate Event

Please refer to section 5.1 for Signal Catch Intermediate Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “Intermediate” and cannot be changed

Extended Attribute:

Group: This is an extended attribute

ID: The UUID of the element

Name: The Logical name of the element

Target: (Not currently implemented)

Trigger:

- Event Details Type: This is set to “SIGNAL” and cannot be changed
- Signal Reference:
  - Name: Logical name of the Signal
  - Properties: Pl refer to Sec 5.1

Multiple Catch Intermediate Event

Please refer to section 5.1 for Multiple Catch Intermediate Event properties like Assignment, Category, and Documentation.

Event Type: This is set to “Intermediate” and cannot be changed

ID: The UUID of the element

Name: The Logical name of the element

Target: (Not currently implemented)

4.5 Tasks

The different Tasks are as below:

- None Task
- Loop Task
- Parallel Multi Instance Loop Task
- Sequential Multi Instance Loop Task

In addition task types may be one of:

- None
- Send
- Receive
- User
- Manual
- Service Task

The properties attached to each type of Task are as below:

None Task

Please refer to section 5.1 for None Task properties like Assignment, Category, Documentation and Property.

Activity Type: This is set to “Task” and cannot be changed

Completion Quantity: (Not currently implemented)

ID: The UUID of the element

Input Set: (Not currently implemented)

IO Rules: (Not currently implemented)

Loop Type: This is set to “None” and cannot be changed

Name: The Logical name of the element

Output Set: (Not currently implemented)

Performers: (Not currently implemented)

Start Quantity: (Not currently implemented)

Status: (Status is set by engine at runtime)

Task Type: can be one of: “None”, “Send”, “Receive”, “User”, “Manual” or “Service”

Standard Loop Task

Please refer to section 5.1 for Standard Loop Task properties like Assignment, Category, Documentation and Property.

Activity Type: This is set to “Task” and cannot be changed

Completion Quantity: (Not currently implemented)

ID: The UUID of the element

Input Set: (Not currently implemented)

IO Rules: (Not currently implemented)

Loop Condition: The logical loop condition expression (see Sec 5.1 for expression builder)

Loop Counter: This is set at runtime by the engine

Loop Maximum: This is set at runtime by the engine

Loop Type: This is set to “Standard Loop” and cannot be changed

Name: The Logical name of the element

Output Set: (Not currently implemented)

Performers: (Not currently implemented)

Start Quantity: (Not currently implemented)

Status: This is set by the engine at run time

Task Type: Can be one of “None”, “Send”, “Receive”, “Service” or “Manual” or “User”

Test Time: One of “Before” or “After” depending on when loop condition is tested

Multi Instance Loop Task

Please refer to section 5.1 for Multi Instance Loop Task properties like Assignment, Category, Documentation and Property.

Activity Type: This is set to “Task” and cannot be changed

Completion Quantity: (Not currently implemented)

Documentation: Pl refer to Sec 5.1

ID: The UUID of the element

Input Set: (Not currently implemented)

IO Rules: (Not currently implemented)

Loop Type: This is set to “Multi Instance Loop” and cannot be changed

MI Condition: Multi Instance Loop conditional expression (See Sec 5.1 for expression builder)

MI Ordering: This can be “Sequential” or “Parallel”

Name: The Logical name of the element

Output Set: (Not currently implemented)

Performers: (Not currently implemented)

Start Quantity: Not currently implemented

Status: This is set by the engine at run time

Task Type: One of “Send”, “Receive”, “Service”, “Manual”, “User” or “None”

Compensation Task

Currently Not Implemented

4.6 Embedded Sub Process

The different types of Embedded Sub Processes are as below:

- Embedded Sub Process
- Loop Embedded Sub Process
- Multiple Instance Embedded Sub Process

The Properties for each type of Sub Process are as below:

Embedded Sub Process

Refer to section 5.1 for properties like Assignment, Category, Documentation and Property.

Activity Type: This is set to “Sub Process” and cannot be changed

Completion Quantity: (Not currently implemented)

ID: The UUID of the element

Input Set: (Not currently implemented)

IO Rules: (Not currently implemented)

Is Transaction: This is set to “false” and cannot be changed

Loop Type: This is set to “None” and cannot be changed

Name: This is the logical name of the Sub process and can be set by user

Output Set: (Not currently implemented)

Performers: (Not currently implemented)

Start Quantity: Not currently implemented

State: Collapsed or Expanded

Status: This is set by the engine at run time

Sub Process Type: This is set to “Embedded Sub process” and cannot be changed

Loop Embedded Sub Process

Please refer to section 5.1 for properties like Assignment, Category, Documentation and Property.

Activity Type: This is set to “Sub Process” and cannot be changed

Completion Quantity: (Not currently implemented)

ID: The UUID of the element

Input Set: (Not currently implemented)

IO Rules: (Not currently implemented)

Is Transaction: This is set to “false” and cannot be changed

Loop Condition: The logical loop condition expression (see Sec 5.1 for expression builder)

Loop Counter: This is set at runtime by the engine

Loop Maximum: This is set at runtime by the engine

Loop Type: This is set to “Standard Loop” and cannot be changed

Name: This is the logical name of the Sub process and can be set by user

Output Set: (Not currently implemented)

Performers: (Not currently implemented)

Start Quantity: Not currently implemented

State: Collapsed or Expanded

Status: This is set by the engine at run time

Sub Process Type: This is set to “Embedded Sub process” and cannot be changed

Test Time: Can be “Before” or “After” depending on when the loop condition is evaluated

Multi Instance Embedded Sub Process

Please refer to section 5.1 for properties like Assignment, Category, Documentation and Property.

Activity Type: This is set to “Sub Process” and cannot be changed

Completion Quantity: (Not currently implemented)

Documentation: Pl refer to Sec 5.1

ID: The UUID of the element

Input Set: (Not currently implemented)

IO Rules: (Not currently implemented)

Loop Type: This is set to “Multi Instance Loop” and cannot be changed

MI Condition: Multi Instance Loop conditional expression (See Sec 5.1 for expression builder)

MI Ordering: This can be “Sequential” or “Parallel”

Name: The Logical name of the element

Output Set: (Not currently implemented)

Performers: (Not currently implemented)

Start Quantity: Not currently implemented

State: “Collapsed” or “Expanded”

Status: This is set by the engine at run time

Sub Process Type: This is set to “Embedded Sub process” and cannot be changed

Transaction Reference: Currently not implemented

4.7 Re-usable Sub Process

The different types of Re-usable Sub Processes are as below:

- Independent Sub Process
- Loop Independent Sub Process
- Multiple Instance Independent Sub Process
- Compensation Independent Sub Process

The properties for each type of Re-usable Sub Process are as below:

Independent Sub Process

Please refer to section 5.1 for properties like Assignment, Category, Documentation and Property.

Activity Type: This is set to “Independent Sub Process” and cannot be changed

Completion Quantity: (Not currently implemented)

Diagram Reference: Refers to the diagram name corresponding to this Sub Process

Extended Attribute: A Place holder for Extended attributes

ID: The UUID of the element

Input Map: This refers to the input Data Mapper UI

Input Set: (Not currently implemented)

IO Rules: (Not currently implemented)

Is Transaction: This is set to “false” and cannot be changed

Loop Type: This is set to “None” and cannot be changed

Name: This is the logical name of the Sub process and can be set by user

Output Maps: This refers to the output Data Mapper UI

Output Set: (Not currently implemented)

Performers: (Not currently implemented)

Process Reference: This refers to the Pool name referred to by the Sub Process in the appropriate diagram

Start Quantity: Not currently implemented

State: Collapsed or Expanded

Status: This is set by the engine at run time

Sub Process Type: This is set to “Reusable Sub process” and cannot be changed

Transaction Reference: Not currently implemented

Loop Independent Sub Process

Please refer to section 5.1 for properties like Assignment, Category, Documentation and Property.

Activity Type: This is set to “Independent Sub Process” and cannot be changed

Completion Quantity: (Not currently implemented)

Diagram Reference: Refers to the diagram name corresponding to this Sub Process

Extended Attribute: A Place holder for Extended attributes

ID: The UUID of the element

Input Map: This refers to the input Data Mapper UI

Input Set: (Not currently implemented)

IO Rules: (Not currently implemented)

Is Transaction: This is set to “false” and cannot be changed

Loop Condition: The logical loop condition expression (see Sec 5.1 for expression builder)

Loop Counter: This is set at runtime by the engine

Loop Maximum: This is set at runtime by the engine

Loop Type: This is set to “Standard Loop” and cannot be changed

Name: This is the logical name of the Sub process and can be set by user

Output Maps: This refers to the output Data Mapper UI

Output Set: (Not currently implemented)

Performers: (Not currently implemented)

Process Reference: This refers to the Pool name referred to by the Sub Process in the appropriate diagram

Start Quantity: Not currently implemented

State: Collapsed or Expanded

Status: This is set by the engine at run time

Sub Process Type: This is set to “Reusable Sub process” and cannot be changed

Test Time: Can be set to “Before” or “After” depending on when the loop condition is evaluated

Transaction Reference: Not currently implemented

Multiple Instance Independent Sub Process

Please refer to section 5.1 for properties like Assignment, Category, Documentation and Property.

Activity Type: This is set to “Independent Sub Process” and cannot be changed

Completion Quantity: (Not currently implemented)

Diagram Reference: Refers to the diagram name corresponding to this Sub Process

Extended Attribute: A Place holder for Extended attributes

ID: The UUID of the element

Input Map: This refers to the input Data Mapper UI

Input Set: (Not currently implemented)

IO Rules: (Not currently implemented)

Is Transaction: This is set to “false” and cannot be changed

Loop Type: This is set to “Multi Instance Loop” and cannot be changed

MI Condition: Multi Instance Loop conditional expression (See Sec 5.1 for expression builder)

MI Ordering: This can be “Sequential” or “Parallel”

Name: This is the logical name of the Sub process and can be set by user

Output Maps: This refers to the output Data Mapper UI

Output Set: (Not currently implemented)

Performers: (Not currently implemented)

Process Reference: This refers to the Pool name referred to by the Sub Process in the appropriate diagram

Start Quantity: Not currently implemented

State: Collapsed or Expanded

Status: This is set by the engine at run time

Sub Process Type: This is set to “Reusable Sub process” and cannot be changed

Test Time: Can be set to “Before” or “After” depending on when the loop condition is evaluated

Transaction Reference: Not currently implemented

Compensation Independent Sub Process

This is currently not implemented

4.8 Gateways

The different types of Gateway figures are as below:

- Normal Gateway
- Parallel Gateway
- Inclusive Gateway
- Event Based Gateway
- Complex Gateway

The Properties for each type of Gateway are as below:

Data Based Gateway

Please refer to section 5.1 for properties like Assignment, Category, and Documentation.

Default gate: ID of the default gate figure
Exclusive Type: This is set to “Data Based” an cannot be changed
Gates: A UI that has the Name, Condition and Logical Expression of each gate
Gateway Type: This is set to “Exclusive” and cannot be changed
ID: The UUID of the element
Name: Logical name of the element

Event Based Gateway

Please refer to section 5.1 for properties like Assignment, Category, and Documentation.

Exclusive Type: This is set to “Event Based” and cannot be changed
Gates: A UI that has the Name, Condition and Logical Expression of each gate
Gateway Type: This is set to “Exclusive” and cannot be changed
ID: The UUID of the element
Instantiate: Not currently supported
Name: The Logical name of the element

Complex Gateway

Please refer to section 5.1 for properties like Assignment, Category, and Documentation.

Complex Incoming Condition: This is a UI that needs the number of Incoming flows and the Compulsory Fork ID
Gates: A UI that has the Name, Condition and Logical Expression of each gate
Gateway Type: This is set to “Complex” and cannot be changed

ID: The UUID of the element

Incoming Condition: (Currently not implemented)

Name: Logical name of the element

Outgoing Condition: (Currently not implemented)

Inclusive Gateway

Please refer to section 5.1 for properties like Assignment, Category, and Documentation.

Default Gate: The ID of the default gate

Extended Attribute: This brings up a “Variable Mapping” UI

Gates: A UI that has the Name, Condition and Logical Expression of each gate

Gateway Type: This is set to “Inclusive” and cannot be changed

ID: UUID of the element

Name: The Logical name of the element

Parallel Gateway

Please refer to section 5.1 for properties like Assignment, Category, and Documentation.

Extended Attribute: This brings up a “Variable Mapping” UI

Gates: A UI that has the Name, Condition and Logical Expression of each gate

Gateway Type: This is set to “Parallel”

ID: The UUID of the element

Name: The Logical name of the element

4.9 Pools and Lanes

Pools and Lanes are used as “Containers” for other objects – e.g. “Business Entities” and/or “Actors”

Please refer to section 5.1 for properties like Category, Documentation, and Property.

Lane Properties are as below:

Name: Logical name of the entity that the lane represents

4.10 Connection Objects

The list of connection objects is as below:

Sequence Flow

- Sequence Flow
- Conditional Flow
- Default Flow
- Message Flow

- Association

The properties for each type of connection are as below:

Sequence Flow

Condition Type: This is set to “None” and cannot be changed

Documentation: (Pl refer to Sec 5.1)

ID: The UUID of the sequence flow object

Name: The Logical name of the sequence flow object

Source Reference: Source Reference of the Sequence Flow

Target Reference: Target Reference of the Sequence Flow

Conditional Flow

Conditional Expression: This shows an Expression Builder UI to set the conditional expression (Sec 5.1)

Condition Type: This is set to “Expression” and cannot be changed

Documentation: (Pl refer to Sec 5.1)

ID: The UUID of the connection object

Name: The Logical name of the connection object

Source Reference: Source ID of the connection

Target Reference: Target ID of the connection

Default Flow

Condition Type: This is set to “Default” and cannot be changed

Documentation: (Pl refer to Sec 5.1)

ID: The UUID of the connection object

Name: The Logical name of the connection object

Source Reference: Source ID of the connection

Target Reference: Target ID of the connection

Message Flow

Documentation: (Sec 5.1)

ID: The UUID of the connection object

Message Reference: (Currently not supported)

Name: Logical name of the message connection

Source Reference: Source object ID

Target Reference: Target object ID

Association

Direction: (Not currently set)

ID: UUID of the connection object

Line Style: Set to “Association”, “Sequence” or “Message Flow”

Name: Logical name of the association connection object

Quantity: (Not currently used)

Source Reference: Source object ID

Target Reference: Target object ID

4.11 Data Objects and Artifacts

The list of Data Objects and Artifacts are as below:

- Data Object
- Text Annotation
- Group
- Label
- Off Page Connector

The properties for each type of Data Object/Artifact are as below:

Common Artifact Properties:

Type: This is one of “Data Object”, “Group” or “Annotation”

Data Object:

Name: The name of the Data Object

State: An optional attribute indicating state of the object

Properties: Please refer to Sec 5.1

Text Annotation:

Text: An attribute used to communicate any information to the reader of the model

Group:

Category Ref: The category that the group represents

Graphical elements: Identifies all elements within the Group

Label:

Label Text: An optional attribute used to label parts of the diagram

Off page connector:

Used when the diagram is larger than a page and used to continue a diagram across pages

Please refer to BPMN release notes for information on deployment

Top