BPMN Export

SME / Author: Jean-Philippe HENG

Ist Edition (April 2014)

mega

CONTENTS

BPMN EXPORT OVERVIEW

The Business Process Modeling and Notation (BPMN) is a graphical representation for specifying processes. This design can be serialized into an xml file that supports the BPMN objects and diagrams. Each process is serialized into a bpmn file.

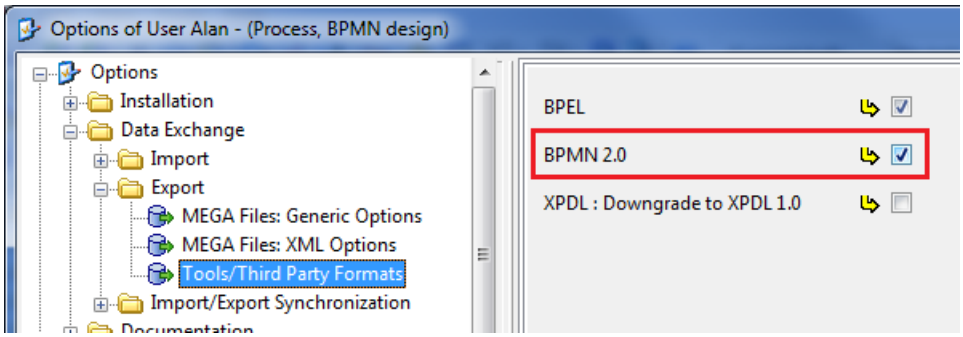
The MEGA BPMN export project aims at exporting workflow definitions as BPMN files so that workflows modeled in MEGA can be used by other BPMN modeling tools or workflow engines.

Prerequisites

The BPMN export feature is available with **MEGA Process BPMN and MEGA System BluePrint**, and supports BPMN version 2.0.

Activate / Deactivate BPMN export

The BPMN export feature can be activated or deactivated for a user, a profile, an environment, or a site. This can be done using the corresponding option:



Scope of BPMN export

The purpose of BPMN export is to export a MEGA Process into a BPMN file.

The tool handles translation of the following processes:

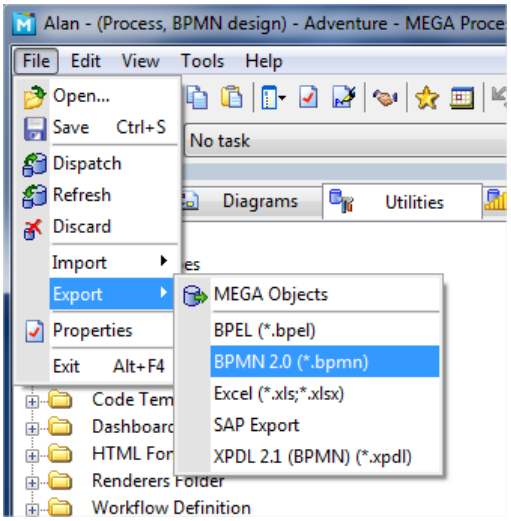
- System Processes
- Organizational Processes
- Business Processes
- Functional Processes

The list of supported mappings is detailed below. See Object correspondences.

Exporting BPMN Files

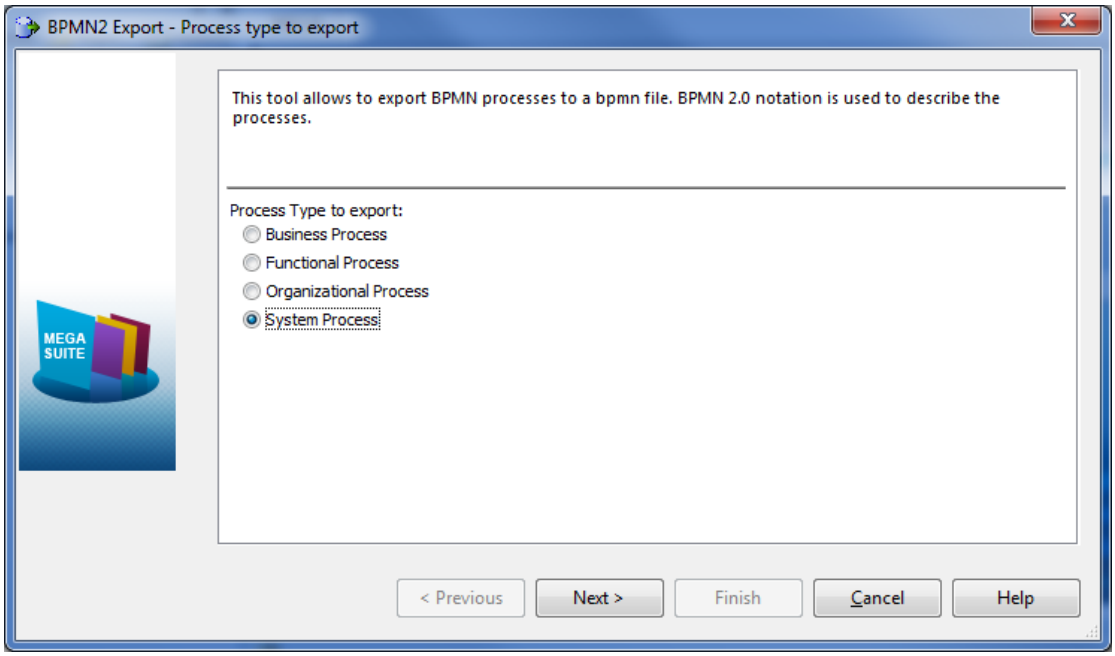
To export a BPMN file from MEGA:

1. In MEGA, select **File > Export > BPMN 2.0 (*.bpmn).**

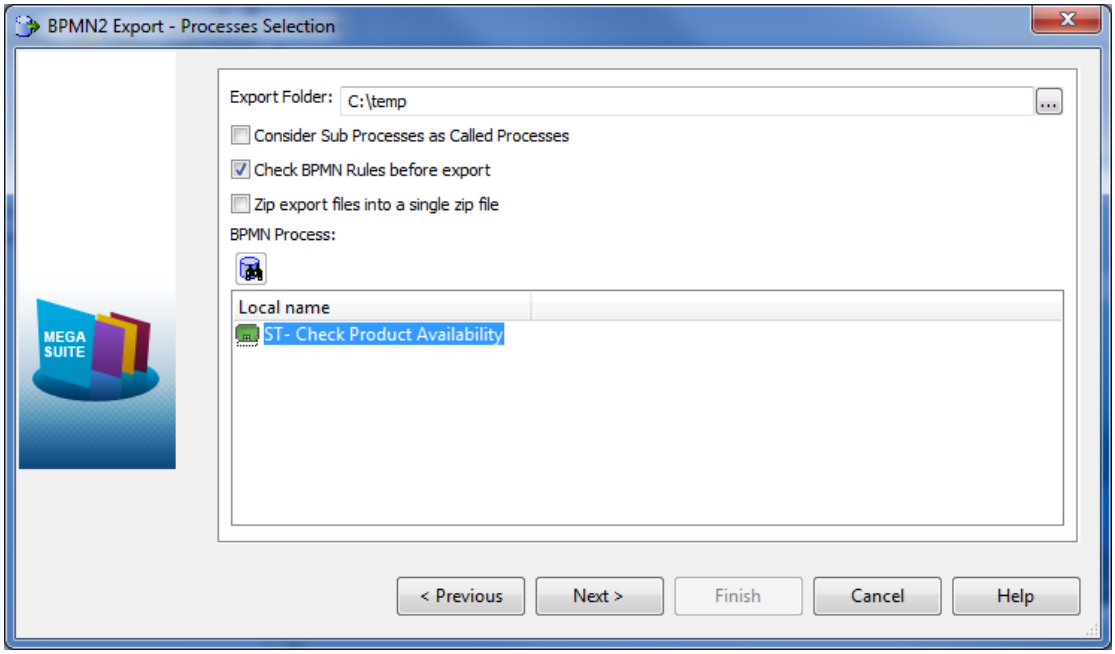


The export dialog box appears.

2. Select the type of object you want to export, for example System Process, and click **Next**.



3. Specify the path of the files to be created.

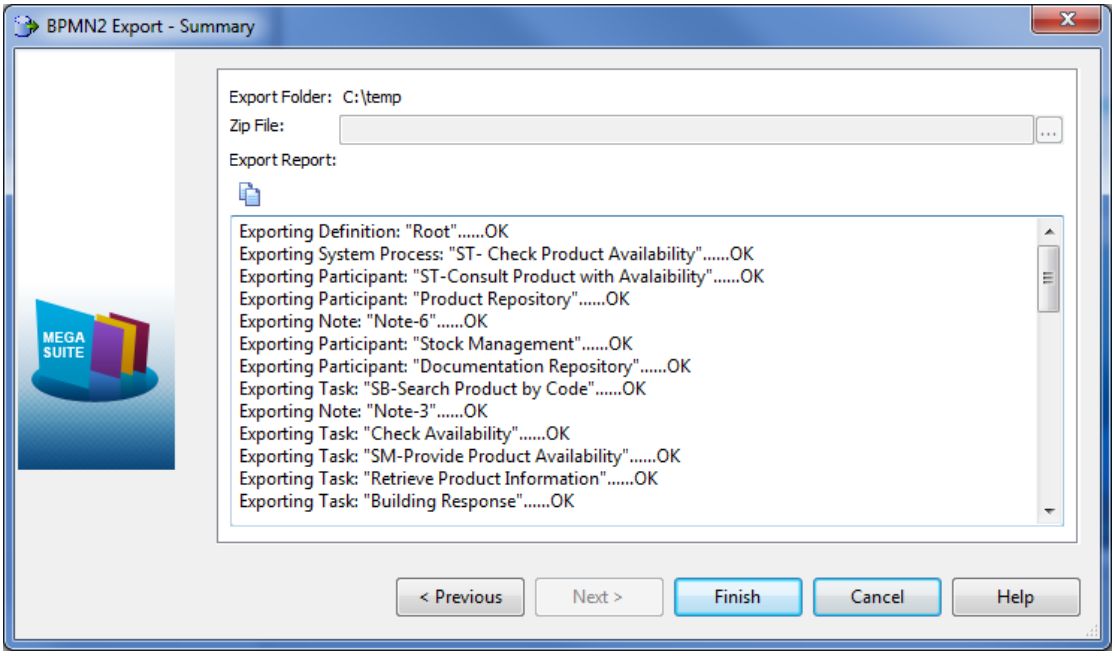


Three options are available:

- **Consider Sub Processes as Called Processes**: In BPMN a sub process is embedded into the parent process, so when exporting the parent process the sub processes will be exported in the same file as embedded processes. This option allows to consider these sub processes as call process tasks.
- **Check BPMN Rules before export**: this option checks whether processes you want to export conform to BPMN modeling rules. If not, a message asks if you want to continue with the export. Click **Yes** to continue or **No** to stop the export.
- **Zip export files into a single zip file**: this option enables to add all exported bpmn files into a single archive file (zip file). Some BPMN modeling tools accept only zip files as import BPMN file.

4. Select one or several processes to be exported.
5. Click **Next**.

   A window shows progression and report of the export.



6. Click **Finish**.

MEGA / BPMN OBJECT CORRESPONDENCE

The BPMN export feature translates a BPMN element specification in MEGA into a BPMN-compliant output file. It is based on a set of mappings between the MEGA BPMN Process metamodel (including diagrams) and BPMN version 2.0.

Conventions

MEGA concept names start with uppercase. Example: Task.

BPMN elements are delimited by <..>.

BPMN attributes are prefixed by ::.

BPMN concepts relate to the following OMG Specification document: “Business Process Model and Notation (BPMN), formal/2011-01-03”.

Object correspondences

The following table indicates concepts managed by the export tool:

MEGA Concepts	BPMN Concepts
Organizational Process	**<process>**
Short Name	name
_Hexaidabs	id
Comment	<documentation>
Owned Operation	<task>
Owned Element	
Gateway	<gateway>
Event	<event>
Sequence Flow	<sequenceFlow>
Data Object	<dataObject> and <dataObjectReference>
Message Flow	<messageFlow>
Participant	<participant> or <lane>
Internal Participant	<laneSet> and <lane>
Owned Organizational Process	<callActivity> or <subProcess>
Note	<textAnnotation> and <text>
Description	<BPMNDiagram>
Operation	**<task>** (*See Table 1)*
Short Name	name
_Hexaidabs	id
Comment	<documentation>
Compensation	isForCompensation

MEGA Concepts	BPMN Concepts
Loop	<standardLoopCharacteristics>
Loop Condition	<loopCondition>
Multiple	<multiInstanceLoopCharacteristics>
Called Organizational Process	calledElement
Data Object Used	<dataInput> and <dataInputAssociation>
Produced Data Object	<dataOutput> and <dataOutputAssociation>
Note	<textAnnotation> and <text>
Gateway	**<gateway>** (*See Table 2)*
Short Name	name
_Hexaidabs	id
Comment	<documentation>
Note	<textAnnotation> and <text>
Event	**<event>** (*See Table 3)*
Short Name	name
_Hexaidabs	id
Comment	<documentation>
Interrupting	isInterrupting or cancelActivity
Bounded Activity	attachedToRef
Note	<textAnnotation> and <text>
Sequence Flow	**<SequenceFlow>**
_Hexaidabs	id
Comment	<documentation>
Predicate	<conditionExpression>
Predecessor	sourceRef
Successor	targetRef
Data Object	**<dataObject> and <dataObjectReference>**
Short Name	name
_Hexaidabs	id
Collection	isCollection
Comment	<dataObjectReference> and <documentation>
Data Object Status	<dataObjectReference::name>
Data Store	**<dataStore> and <dataStoreReference>**
Short Name	name
_Hexaidabs	id
Message Flow	**<messageFlow>**

MEGA Concepts	BPMN Concepts
_Hexaidabs	id
Comment	<documentation>
Source Element	sourceRef
Target Element	targetRef
Internal Participant	**<lane>**
Short Name	name
_Hexaidabs	id
Assignment	name
BPMN Element	<flowNodeRef>
External Participant	**<participant>**
Short Name	name
_Hexaidabs	id
Assignment	name
Note	<textAnnotation> and <text>
Note	**<textAnnotation> and <association>**
_Hexaidabs	id
Comment	<text>
System Process	**<process>**
Short Name	name
_Hexaidabs	id
Comment	<documentation>
Owned Task	<task>
Owned Element	
Gateway	<gateway>
Event	<event>
Sequence Flow	<sequenceFlow>
Data Object	<dataObject> and <dataObjectReference>
Message Flow	<messageFlow>
Participant	<participant> or <lane>
Internal Participant	<laneSet> and <lane>
Owned System Process	<callActivity> or <subProcess>
Note	<textAnnotation> and <text>
Description	<BPMNDiagram>
Task	**<task>** (*See Table 1)*
Short Name	name

MEGA Concepts	BPMN Concepts
_Hexaidabs	id
Comment	<documentation>
Compensation	isForCompensation
Loop	<standardLoopCharacteristics>
Loop Condition	<loopCondition>
Multiple	<multiInstanceLoopCharacteristics>
Called Organizational Process	calledElement
Data Object Used	<dataInput> and <dataInputAssociation>
Produced Data Object	<dataOutput> and <dataOutputAssociation>
Note	<textAnnotation> and <text>
Functional Process	**<process>**
Short Name	name
_Hexaidabs	id
Comment	<documentation>
Owned Functional Activity	<task>
Owned Element	
Gateway	<gateway>
Event	<event>
Sequence Flow	<sequenceFlow>
Data Object	<dataObject> and <dataObjectReference>
Message Flow	<messageFlow>
Participant	<participant> or <lane>
Internal Participant	<laneSet> and <lane>
Owned Functional Process	<callActivity> or <subProcess>
Note	<textAnnotation> and <text>
Description	<BPMNDiagram>
Functional Activity	**<task>** (*See Table 1)*
Short Name	name
_Hexaidabs	id
Comment	<documentation>
Compensation	isForCompensation
Loop	<standardLoopCharacteristics>
Loop Condition	<loopCondition>
Multiple	<multiInstanceLoopCharacteristics>
Called Organizational Process	calledElement
Data Object Used	<dataInput> and <dataInputAssociation>

MEGA Concepts	BPMN Concepts
Produced Data Object	<dataOutput> and <dataOutputAssociation>
Note	<textAnnotation> and <text>
Business Process	**<process>**
Short Name	name
_Hexaidabs	id
Comment	<documentation>
Owned Functional Activity	<task>
Owned Element	
Gateway	<gateway>
Event	<event>
Sequence Flow	<sequenceFlow>
Data Object	<dataObject> and <dataObjectReference>
Message Flow	<messageFlow>
Participant	<participant> or <lane>
Internal Participant	<laneSet> and <lane>
Owned Functional Process	<callActivity> or <subProcess>
Note	<textAnnotation> and <text>
Description	<BPMNDiagram>

Table 1:

Task Type	BPMN
Task Type	**<task>**
User	<userTask>
IT-Service Call	<serviceTask>
System Process Call	<callActivity>
Script	<scriptTask>
Send	<sendTask>
Receive	<receiveTask>
Manual	<manualTask>
Business Rule	<businessRuleTask>

Table 2:

Gateway Type	BPMN
Gateway Type	**<gateway>**
Inclusive	<inclusiveGateway>
Complex	<complexGateway>
Parallel	<parallelGateway>
Exclusive (Data)	<exclusiveGateway>
Exclusive (Event)	<eventBasedGateway> and ::eventGatewayType=Exclusive and ::instantiate=false
Exclusive (Instantiate)	<eventBasedGateway> and ::eventGatewayType=Exclusive and ::instantiate=true
Parallel (Instantiate)	<eventBasedGateway> and ::eventGatewayType=Parallel and ::instantiate=true

Table 3:

		Event Nature				Bounded Activity
		Start	End	Throw	Catch	
		<startEvent>	*<endEvent>*	*<intermediate ThrowEvent>*	*<intermediate CatchEvent>*	*<boundary Event>*
Event Type	Generic					
	Message	<messageEventDefinition>	<messageEventDefinition>	<messageEventDefinition>	<messageEventDefinition>	<messageEventDefinition>
	Timer	<timerEventDefinition>	*	*	<timerEventDefinition>	<timerEventDefinition>
	Error	<errorEventDefinition>	<errorEventDefinition>	*	*	<errorEventDefinition>
	Cancel	*	<cancelEventDefinition>	*	*	<cancelEventDefinition>
	Compensation	<compensateEventDefinition>	<compensateEventDefinition> and ::waitForCompletion=true	<compensateEventDefinition> and ::waitForCompletion=true	*	<compensateEventDefinition> and ::waitForCompletion=true
	Conditional	<conditionalEventDefinition> and <condition>	*	*	<conditionalEventDefinition> and <condition>	<conditionalEventDefinition> and <condition>
	Link	*	*	<linkEventDefinition>	<linkEventDefinition>	*
	Signal	<signalEventDefinition>	<signalEventDefinition>	<signalEventDefinition>	<signalEventDefinition>	<signalEventDefinition>
	Terminate	*	<terminateEventDefinition>	*	*	*
	Escalation	<escalationEventDefinition>	<escalationEventDefinition>	<escalationEventDefinition>	*	<escalationEventDefinition>
	Parallel Multiple	::parallelMultiple=true	*	*	::parallelMultiple=true	::parallelMultiple=true
	Multiple					

* = Combinaison not possible