Queensland Government Enterprise Architecture

BPMN 2.0 guideline

Recommended Business Process Model and Notation 2.0 Elements

Final

January 2016

V3.0.0

PUBLIC

Document details

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. **Security classification** | 1. PUBLIC | | | | | |
| 1. **Date of review of security classification** | 1. January 2016 | | | | | |
| 1. **Authority** | 1. Queensland Government Chief Information Officer | | | | | |
| 1. **Author** | 1. Queensland Government BPMN Working Group 2. Department of Science, Information Technology and Innovation (chair) | | | | | |
| 1. **Documentation status** |  | 1. Working draft |  | 1. Consultation release | 1. 🗹 | 1. Final version |

Contact for enquiries and proposed changes

All enquiries regarding this document should be directed in the first instance to:  
  
Chief Information Office  
Department of Science, Information Technology and Innovation

[margaret.turner@dsiti.qld.gov.au](mailto:margaret.turner@dsiti.qld.gov.au)

Acknowledgements

This version of the *Business Process Model and Notation (BPMN 2.0) Guideline* was developed by a Queensland Government working group comprising staff from

* Department of Science, Information Technology and Innovation (chair)
* Department of Communities, Child Safety and Disability Services
* Department of Transport and Main Roads
* Queensland Government Chief Information Office
* Queensland Shared Services
* Smart Service Queensland.

Copyright

*Business Process Model and Notation (BPMN 2.0) guideline*

Copyright © The State of Queensland (Department of Science, Information Technology and Innovation) 2015

Licence



*Business Process Model and Notation (BPMN 2.0) guideline* is licensed under a Creative Commons Attribution 4.0 International licence. To view the terms of this licence, visit <http://creativecommons.org/licenses/by/4.0/>. Permissions may be available beyond the scope of this licence.

Information security

This document has been security classified using the Queensland Government Information Security Classification Framework (QGISCF) as PUBLIC and will be managed according to the requirements of the QGISCF.

Contents

[1 Introduction 4](#_Toc439766485)

[1.1 Purpose 4](#_Toc439766486)

[1.2 What is BPMN? 4](#_Toc439766487)

[1.3 Audience 4](#_Toc439766488)

[1.4 Scope 4](#_Toc439766489)

[1.5 Benefits 5](#_Toc439766490)

[2 Background 5](#_Toc439766491)

[2.1 Uses of BPMN models 5](#_Toc439766492)

[2.2 Business and operational views of processes 6](#_Toc439766493)

[2.3 Terminology 6](#_Toc439766494)

[3 BPMN recommendations 7](#_Toc439766495)

[4 BPMN elements 7](#_Toc439766496)

[5 Queensland Government business process model conventions 16](#_Toc439766497)

[5.1 Process descriptions 16](#_Toc439766498)

[5.2 Business processes and sub-processes 16](#_Toc439766499)

[5.3 Tasks 18](#_Toc439766500)

[5.4 Words to avoid in the business view of a process 19](#_Toc439766501)

[5.5 Events 20](#_Toc439766502)

[5.6 Pools and swim lanes 20](#_Toc439766503)

[5.7 Connecting objects 21](#_Toc439766504)

[5.8 Data objects 21](#_Toc439766505)

[5.9 Diagram layout 21](#_Toc439766506)

[6 References 22](#_Toc439766507)

[6.1 Document history 23](#_Toc439766508)

[Appendix A Table of event symbols 24](#_Toc439766509)

[Appendix B The case to be used for element names 25](#_Toc439766510)

[Appendix C Gateways and conditional flows 26](#_Toc439766511)

[Appendix D Business process description 28](#_Toc439766512)

# Introduction

## Purpose

This guideline provides information and advice to Queensland Government agencies on the recommended practices for documenting business processes using the [Business Process Model and Notation (BPMN[[1]](#footnote-2)) Specification](http://www.omg.org/spec/BPMN/2.0/) Version 2.0 released January 2011. This guideline is for information only and agencies are not required to comply. However, alignment with the guideline will simplify the sharing and reuse of BPMN diagrams and documentation between agencies.

## What is BPMN?

BPMN is a graphical notation for drawing business processes. It is an industry standard first developed in 2002 as the Business Process Modelling Notation and now managed as the **Business Process Model and Notation 2.0** [2] by the international, open membership, not-for-profit, technology standards consortium, the Object Management Group (OMG). BPMN is a flowcharting technique very similar to activity diagrams from the Unified Modelling Language (UML).

1. The objective of BPMN is to

* assist communication about business processes
* support business process management
* provide a mapping between the notation and computing execution languages such as the Business Process Execution Language (BPEL) [1].

## Audience

This document is primarily intended for:

* business process analysts
* business analysts
* business architects.

It is assumed that the reader has:

* some knowledge of business process modelling
* a basic understanding of the BPMN Specification v1.2.

It should be noted that this is ***not*** a BPMN training document.

## Scope

### In scope

This guideline is available to all Queensland Government departments and agencies.

This guideline relates to the business layer of the Queensland Government Enterprise Architecture (QGEA).

The focus of the guideline is on the recommended practices for Queensland Government use of BPMN 2.0 such as the recommended elements, naming conventions, diagram layout and process descriptions.

### Out of scope

* Defining organisational models and resources such as:

strategy models

functional models

data and information models

business rules models

value chain.

* Operational simulation, monitoring and deployment of business processes.

## Benefits

A variety of modelling notations is available to document business processes. The use of different process models incurs a high overhead across government in maintaining the various knowledge and skills. It can also be problematic to share models between systems which use different notations.

The benefits of establishing a common, standard and royalty-free notation for business process modelling across Queensland Government include:

* Improved communication and understanding amongst all users, from the business analysts that create the initial drafts of the processes, to the technical staff responsible for implementing the technology that will support the processes, through to the business people who will own, undertake and monitor the processes [2]
* Decreased training costs due to rationalisation of skillsets based on an industry standard
* Increased potential for comparing, sharing and reusing models amongst different business units and agencies. This will simplify analysis and enable an architectural approach to be applied to identify synergies across business areas and agencies.

# Background

## Uses of BPMN models

Business processes are underlying mechanisms that support the delivery of every organisation's day-to-day services. A business process encompasses participants, tasks and supporting systems that work together to produce an end result that is of value to the organisation. Organisations are becoming increasingly aware of the importance of adopting a process-centric view of their business activities to better coordinate implementation activities.

Business process modelling is a technique that promotes the visual representation of an organisation's business processes. The use of BPMN process models allows an organisation to obtain a view of its business activities so that it can then:

* identify interactions and inter-dependencies between various components of a process
* analyse a process for improvements
* allocate ownership of a process
* identify risk in a process
* apply metrics to a process to measure the impact of any process change
* report the metrics and key performance indicators for use in business performance management
* determine the alignment of a process with business strategy and objectives
* identify the information, applications and technologies associated with a process
* simulate changes to a process and determine the outcome.

## Business and operational views of processes

Process models are developed to provide a certain level of detail or granularity about processes. It is important to ensure that the granularity level of any model is appropriate for the audience and the use of the model.

This document refers to two views of processes which are modelled using BPMN:

1. Business view – this is the view of the straightforward process that deals with the standard scenario. Exceptions are generally not modelled. The level of detail is appropriate for communicating with business staff. The variety of symbols used is minimal to assist with communication.
2. Operational view – this view may include more details of the process such as computer systems or exceptions. It is suitable for conveying work instructions or for testing software requirements. A greater variety of symbols is used in order to provide the detail required.

Agencies may determine a series of levels or views for their business processes – in effect a hierarchy. This is outside the scope of this guideline. However, a common government hierarchy may be considered in the future.

**Operational view**

* For operational staff
* Detailed process
* Exceptions included
* Variety of symbols

More detail

**Business view**

* For business staff
* Standard process
* No exceptions
* Minimal symbols

## Terminology

|  |  |
| --- | --- |
| **Activity** | Work that an organisation performs using processes. An activity can be atomic (such as a task) or non-atomic (Process or Sub-Process). |
| **Process** | A defined sequence of activities in an organisation that represent the steps required to carry out work and achieve a business objective. It includes the flow and use of information and resources. In BPMN, a Process is depicted as a graph of Flow Elements, which are a set of Activities, Events, Gateways, and Sequence Flows. |
| **Sub-Process** | A Process that is included within another Process. The Sub-Process can be in a collapsed view that hides its details. |
| **Task** | An atomic activity that is included within a Process. A Task is used when the work in the Process is not broken down to a finer level of Process Model detail. Generally, an end-user, an application, or both will perform the Task. |

For further definitions, please refer to the glossary on page 499 of the OMG specification [2] at <http://www.omg.org/spec/BPMN/2.0/PDF>.

# BPMN recommendations

This guideline supports the use of:

* collaboration models
* data object references
* loop sub-processes
* some specialist event types such as:

link, message, and timer events for the business view of a process and

conditional, error, signal, and terminate events for the operational view of a process

* minimal colour – in order to support colour perception issues and black and white printing, symbols should use a means other than colour to differentiate themselves.

This guideline ***does not*** recommend the use of the elements listed below. Please note that the elements listed below ***are*** part of the BPMN 2.0 specification [2] but simply have ***not*** been recommended in this guideline in order to improve readability of BPMN diagrams and to simplify learning and communicating the commonly-used BPMN element set. Refer to the OMG specification for more information on these elements.

* conversations
* choreographies
* some specialist event types such as escalation, cancel, compensation, multiple and parallel multiple events
* non-interrupting start and intermediate events
* event sub-processes, transaction sub-processes, ad-hoc sub-processes, multi-instance sub-processes and call activities
* multi-instance loop sub-processes
* compensation and hence compensation events, compensation event sub-processes, and compensation activities
* event-based, parallel event-based or complex gateways
* multi-instance pools
* message symbols on message flows as this requires each message to be classified as either the first (initiating) message or a return (non-initiating) message, and the symbol to be coloured (clear for initiating, light fill for non-initiating) making the model very busy
* numbering of processes, sub-processes, activities and tasks.

# BPMN elements

The following table provides the recommended subset of elements to use. It is based on the tables in the specification [2] pages 29-41. All page references are to the specification unless otherwise noted. The recommended subset covers the following elements:

1. Flow Objects

* Event
* Activity
* Gateway

1. Data

* Data Object
* Data Object Reference
* Data Store

1. Connecting Objects

* Sequence Flow
* Message Flow
* Data Association

1. Swimlanes

* Pool
* Lane

1. Artefacts

* Group
* Text Annotation

| Category / element | Description | Notation |
| --- | --- | --- |
| * 1. **Flow objects** | **Define the behaviour of a business process** | |
| Event | An Event is something that ‘happens’ during the course of a Process (see page 233[[2]](#footnote-3)). Thus Events affect the flow of the process. There are three types of Events, based on when they affect the flow: Start, Intermediate, and End. Examples of events include the start of an Activity, the end of an Activity, the change of state of a document, or the arrival of a Message. The Start Event and some Intermediate Events have ‘triggers’ that define the cause for the Event. End Events may define a ‘result’ that is a consequence of a Sequence Flow path ending. Start Events can only react to (‘catch’) a trigger. End Events can only create (‘throw’) a result. Intermediate Events can catch or throw triggers. Events are shown as circles with open centres to allow internal markers to differentiate different triggers or results. For the triggers that catch, the event markers are unfilled, and for triggers and results that throw, the markers are filled. See Table of Event Symbols in Appendix A on page 24 of this guideline. | |
| 1. **Start event** | The Start Event indicates where a particular Process will start (see [2] page 238). It does not have any incoming Sequence Flows. It must have at least one outgoing Sequence Flow. A Start Event may have zero or more incoming Message Flows; each one is a trigger for the Process. Only one of the triggers is required to start the Process. A Start Event can***not*** have outgoing Message Flows. Use of a Start Event is recommended. Use of multiple Start Events is ***not*** recommended as it impedes understanding. A Start Event is a circle drawn with a single thin line. It has an open centre so that markers can be placed within the circle to indicate variations of the Event. | |
| None | The None Start Event does not have a defined trigger. It is displayed without a marker. |  |
| Conditional | This type of event is triggered when a condition such as ‘Project expenditure to date exceeds tolerance’, or ‘Temperature above 300C’ becomes true. The Condition Expression for the Event must become false and then true before the Event can be triggered again. The Condition Expression of a Conditional Start Event must not refer to the data context or instance attribute of the Process (as the Process instance has not yet been created). Instead, it may refer to static Process attributes and states of entities in the environment. It is displayed with a lined paper marker. |  |
| Message | The Message Start Event indicates that a Message arrives from a Participant and triggers the start of the Process. It is displayed with an envelope marker. |  |
| Signal | Signals are triggers generated in the Pool in which they are published. They are typically used for broadcast communication within and across Processes, across Pools, and between Process diagrams. A Signal that has been broadcast from another Process arrives and triggers the start of the Process. Note that the Signal is ***not*** a Message, which has a specific target for the Message. Multiple Processes can have Start Events that are triggered from the same broadcasted Signal. The Signal Event is displayed with a triangle marker. |  |
| Timer | A specific time-date or a specific cycle (e.g. every Monday at 9am) can be set that will trigger the start of the Process. It is displayed with a clock marker. |  |
| 1. **End Event** | The End Event indicates where a path of the Process will end (see [2] page 246). It must have at least one incoming Sequence Flow but does ***not*** have any outgoing Sequence Flows. An End Event must ***not*** have any incoming Message Flows. It can have zero or more outgoing Message Flows; each one will have a Message sent when the Event is triggered. There may be multiple End Events within a single level of a Process. Use of an End Event is recommended. An End Event is a circle drawn with a single thick line. It has an open centre so that markers can be placed within the circle to indicate variations of the Event. | |
| None | The None End Event does not have a defined result. It is displayed without a marker. |  |
| Error | The Error End Event indicates that a named Error should be generated. All currently active threads in the particular Sub-Process are terminated as a result. The Error will be caught by a Catch Error Intermediate Event. The behaviour of the Process is unspecified if no Activity in the hierarchy has such an Error Intermediate Event. The system executing the process can define additional Error handling in this case, a common one being termination of the Process instance. The Error Event uses a filled lightning marker. |  |
| Message | The Message End Event indicates that a Message is sent to a Participant at the conclusion of the Process. It uses a filled envelope marker. |  |
| Signal | This type of End Event indicates that a Signal will be broadcast when the End has been reached. Note that the Signal, which is broadcast to any Process that can receive the Signal, can be sent across Process levels or Pools, but is ***not*** a Message that has a specific source and target. It uses a filled triangle marker. |  |
| Terminate | This type of End Event indicates that all Activities in the Process should be immediately ended. It uses a filled circle marker. |  |
| 1. **Intermediate Event** | Intermediate Events occur between a Start Event and an End Event (see [2] page 249). They will affect the flow of the Process but will not start or (directly) terminate the Process. | |
| None | The None Intermediate Event is only valid in normal flow, i.e. it may not be used on the boundary of an Activity. Although there is no specific trigger for this Event, it is defined as a throw Event. It is used for modelling methodologies that use Events to indicate some change of state in the Process. |  |
| Conditional  (catching) | This type of Event is triggered when a condition becomes true. A condition is a type of Expression, which can be in natural language text or a formal expression language. It is displayed with a lined paper marker |  |
| Error  (catching) | A catch Intermediate Error Event can only be attached to the boundary of an Activity, i.e. it may not be used in normal flow. If used in this context, it reacts to (or catches) a named Error, or to any Error if a name is not specified. It uses an unfilled lightning marker. |  |
| Link  (catching)  (throwing) | A Link is a mechanism for connecting two sections of a Process. It can be used to create looping situations or to avoid long Sequence Flow lines. It is particularly useful for splitting a process into several screens or pages for viewing. See section 5.9.1 Fragmentation on page 22 of this guideline. Link Event uses are limited to a single Process level (i.e. they cannot link a parent Process with a Sub-Process). They can also be used as generic ‘Go To’ objects within the Process level. There can be multiple source Link Events, but there can only be one target Link Event. Link Events use a block arrow marker. When used to ‘catch’ from a source Link, the block arrow is unfilled. When used to ‘throw’ to a target Link, the block arrow is filled. Link Intermediate Events are only valid in normal flow, i.e. they may not be used on the boundary of an Activity. |  |
| Message  (catching)  (throwing) | A Message (catching) Intermediate Event is used to receive a Message. This causes the Process to continue if it was waiting for the Message. The envelope marker is unfilled. A Message (throwing) Intermediate Event is used to send a Message. The envelope marker is filled. |  |
| Signal  (catching)  (throwing) | A Signal is a trigger generated in the Pool in which it is published and is used for general broadcast communication within and across Process levels, across Pools, and between Business Process Diagrams. It is similar to a signal flare that has been shot into the sky for anyone who might be interested to notice and then react. Thus, there is a source of the Signal, but no specific intended target. This type of Intermediate Event can send (throw) or receive (catch) a Signal if the Event is part of a normal flow. When attached to the boundary of an Activity, the Event can only receive a Signal. The Signal Event differs from an Error Event in that the Signal defines a more general, non-error condition for interrupting Activities (such as the successful completion of another Activity) as well as having a larger scope than Error Events. When used to ‘catch’ the Signal, the triangle marker is unfilled. When used to ‘throw’ the Signal, the triangle marker is filled. |  |
| Timer  (catching) | Timer triggers are implicitly thrown. When they are activated they wait for a time based condition to trigger the catch Event. In normal flow the Timer Intermediate Event acts as a delay mechanism based on a specific time-date or a specific cycle (e.g., every Monday at 9am) which can be set and trigger the Event. This Event is displayed with a clock marker. |  |
| Activity | An Activity is a generic term for work that an organisation performs in a Process (see [2] page 151). An Activity can be an atomic Task or a non-atomic (compound) Sub-Process. An Activity can have multiple incoming and multiple outgoing Sequence Flows. An Activity can have zero or more incoming and zero or more outgoing Message Flows. | |
| 1. **Task** | A Task is an atomic Activity in a Process (see [2] page 156). A Task is used when the work in the Process cannot be broken down to a finer level of detail. Generally, an end-user or an application is used to perform the Task when it is executed. Tasks are drawn as a rounded rectangle with the task name in the centre.  Tasks can loop. Loops are marked with a small line with an arrowhead that curls back upon itself in the lower-centre of the rounded rectangle.  In an operational view of a process, the Type of the Task (see [2] page 158) may be indicated through the use of graphical markers in the upper left corner of the task shape. If there is no marker the Task is called an Abstract Task (this was referred to as the ‘None Task’ in BPMN 1.2). It is also possible to combine an intermediate event symbol with a task (see [2] page 47) or a sub-process (see [2] page 179) as a type of shorthand. |  |
| 1. **Collapsed Sub-Process** | A Sub-Process is a compound Activity that is included within a Process (see page 173). It is compound in that it can be broken down into a finer level of detail (a Process) through a set of sub-Activities. In a Collapsed Sub-Process, the details of the Sub-Process are not visible in the diagram. This guideline recommends that Sub-Processes are presented in their collapsed form in the original Process diagram, unless they are very simple (such as a two or three task sub-process). The expanded view of the Sub-Process can be provided separately. Sub-Processes can be used to indicate a group of Activities in a less-cluttered, more compact way. A ‘plus’ sign in a square in the lower-centre of the rounded rectangle indicates that the Activity is a Sub-Process (not a Task) and has a lower level of detail. For examples of expanded Sub-Processes, please see section 5.2.1 Examples of Expanded Sub-Processes in this document on page 18. |  |
| Gateway | A Gateway is used to control the divergence and convergence of Sequence Flows in a Process (see [2] page 287). The term ‘Gateway’ implies that there is a gating mechanism that either allows or disallows passage through the Gateway. Thus, it will determine branching, forking, merging, and joining of paths. It is recommended that a single Gateway has either multiple input or multiple output flows but ***not*** both. Thus, it would take two sequential Gateways to first converge and then to diverge the Sequence Flows. It is recommended that sequence flows connect to the corners of the diamond where possible. Gateways do not represent ‘work’ being done and they are considered to have zero effect on the operational measures of the Process being executed (cost, time, etc.). A Gateway is a diamond that may have internal markers to indicate the type of behaviour control. | |
| 1. **Exclusive** | Exclusive gateways perform ‘either/or’ decisions (diverging) and merging (see page 290). A diverging Exclusive Gateway (Decision) is used to create alternative paths within a Process flow. This is basically the ‘diversion point in the road’ for a Process. Only one of the paths is taken. A Decision can be thought of as a question that is asked at a particular point in the Process. The question has a defined set of alternative answers. Each answer is associated with a condition Expression that is associated with the Gateway’s outgoing Sequence Flows. A default path can optionally be identified, to be taken in the event that none of the conditional Expressions evaluate to true. However, this guideline does not recommend the use of explicit default sequence flows. If a default path is not specified and the Process is executed such that none of the conditional Expressions evaluates to true, a runtime exception occurs. An Exclusive Gateway can be shown with or without an ‘X’ marker. However, diagrams must be consistent in their use of the ‘X marker. This guideline recommends that if exclusive gateways are the only type of gateway used in a collection of models then ***no*** ‘X’ marker is used throughout the collection. |  |
| 1. **Inclusive** | Inclusive Gateways perform ‘or’ decisions and merging where more than one path can be taken (see [2] page 292). A diverging Inclusive Gateway (Inclusive Decision) can be used to create alternative but also parallel paths within a  Process flow. Unlike the Exclusive Gateway, all condition Expressions are evaluated. The true evaluation of one  condition Expression does not exclude the evaluation of other condition Expressions. All Sequence Flows with  a true evaluation will be traversed. Since each path is considered to be independent, all combinations of the  paths MAY be taken, from zero to all. However, the gateway should be designed so that at least one path is taken. A default path can optionally be identified, to be taken in the event that none of the conditional Expressions evaluate to true. However, this guideline does not recommend the use of explicit default sequence flows. If a default path is not specified and the Process is executed such that none of the conditional Expressions evaluates to true, a runtime exception occurs. A converging Inclusive Gateway is used to merge a combination of alternative and parallel paths. The Inclusive Gateway MUST use a marker in the shape of a circle or an ‘O’ placed within the Gateway diamond to distinguish it from other Gateways. |  |
| 1. **Parallel** | A Parallel Gateway is used to model ‘and’ forking and joining, i.e. to create parallel flows and to synchronize (combine) parallel flows (see [2] page 293). A Parallel Gateway creates parallel paths without checking any conditions; each path is taken. For incoming flows, the Parallel Gateway will wait for all incoming flows before triggering the flow through its outgoing Sequence Flow. The Parallel Gateway MUST use a marker in the shape of a plus sign placed within the Gateway diamond to distinguish it from other Gateways. |  |
| * 1. Data | **Models the physical or information items that are created, manipulated, and used during the execution of a process** | |
| 1. **Data Object** | Data Objects provide the information requirements or information products of an Activity (see [2] page 205). Data Objects model data within the Process flow and must be contained within a Process or Sub-Process element. A Data Object can represent a single instance of a data structure or, with the addition of a set of three vertical lines in the lower-centre of the document shape, it can represent a Collection of data structures. A Data Object can be represented in a Process multiple times, each time referencing the same Data Object instance. |  |
| 1. **Data Input** | Data Input and Data Output provide information about the data required and data produced by Processes (see [2] page 211). They are represented by an arrow in the top left corner of the data shape – unfilled arrow for Data Input, filled arrow for Data Output. There may be multiple Data Inputs and Data Outputs related to a Task or in a Process. |  |
| 1. **Data Output** |
| 1. **Data Object Reference** | Data Object References can be used to specify different states of the same data object at different points in a process (see [2] page 205). They are represented as <Data Object Name> [ <Data Object Reference State> ]  For example: ICT Strategic Plan [Draft], ICT Strategic Plan [Approved] |  |
| 1. **Data Store** | A Data Store provides a mechanism for Activities to retrieve or update stored information that will persist beyond the  scope of the Process (see [2] page 208). The same Data Store can be visualised, through a Data Store Reference, in one or more places in the Process. The Data Store Reference can be used as the source or target for a Data Association. When data flows into or out of a Data Store Reference, it is effectively flowing into or out of the Data Store that is being referenced. This guideline recommends that Data Stores are only used in operational views of business processes. |  |
| * 1. Connecting Object | **Connects Flow Objects to each other or other information** | |
| 1. **Sequence Flow** | A Sequence Flow is used to show the order in which Activities will be performed in a Process i.e. the order of the flow (see [2] page 97). Each Sequence Flow has only one source and only one target. The source and target MUST be from the set of the following Flow Elements: Events (Start, Intermediate, and End), Activities (Task and Sub-Process; for Processes), and Gateways. Two kinds of Sequence Flow are recommended for use:   1. Standard – has no additional markings. It is a solid line with a solid arrowhead. 2. Conditional – A Sequence Flow can optionally define a condition Expression, indicating the path taken if the Expression evaluates to true. If a conditional Sequence Flow is used from a source Activity, then there MUST be at least one other outgoing Sequence Flow from that same Activity. A conditional outgoing Sequence Flow from an Activity MUST be drawn with a mini-diamond marker at the beginning of the line, and a solid arrowhead at the end. In effect, the gateway diamond shape is being replaced by the mini-diamond marker. See Gateways and Conditional Flows on page 26 of this guideline. |  |
| 1. **Message Flow** | A Message Flow is used to show the flow of Messages between two Participants that are prepared to send and receive them (see [2] page 43 and 120). The Participants MUST be in two separate Pools. The Message Flow connects either to the Pool boundary or to Flow Objects within the Pool. They MUST ***not*** connect two objects within the same Pool. A Message Flow is a dashed line with an open circle line start and an open arrowhead line end. It is recommended to avoid using Messages (envelope shape). If messages are shown then they need to be coloured (clear for initiating message, dark for non-initiating message – see [2] page 411). This can make a diagram very busy, so the Message shape is not recommended. |  |
| 1. **Data Association** | Data Associations are used to move data between Data Objects and inputs and outputs of Activities and Processes (see [2] page 221). They have no direct effect on the flow of a Process. A Data Association has sources, a target, and an optional transformation. When a data association is ‘executed,’ data is copied to the target. What is copied depends if there is a transformation defined or not. If there is no transformation defined or referenced, then only one source MUST be defined, and the contents of this source will be copied into the target. If there is a transformation defined or referenced, then this transformation Expression will be evaluated and the result of the evaluation is copied into the target. There can be zero to many sources defined in this case, but there is no requirement that these sources are used inside the Expression. In any case, sources are used to define if the data association can be ‘executed,’ if any of the sources is in the state of ‘unavailable,’ then the data association cannot be executed, and the Activity or Event where the data association is defined MUST wait until this condition is met. Data Associations are always contained within another element that defines when these data associations are going to be executed. Activities define two sets of data associations, while Events define only one. For Events, there is only one set, but they are used differently for catch or throw Events. For a catch Event, data associations are used to push data from the Message received into Data Objects and properties. For a throw Event, data associations are used to fill the Message that is being thrown. An alternative to using a Data Association is a visual shortcut which directly associates a Data Object with a Sequence Flow connector by using an Association. A Data Association is a dashed line with an arrowhead which indicates the direction of the data flow. |  |
| 4. Swimlane | **Groups elements** | |
| 1. **Pool** | A Pool is the graphical representation of a Participant in a Collaboration (see [2] page 112). It also acts as a ‘swimlane’ and a graphical container for partitioning a set of Activities and their Sequence Flows from other Pools, usually in the context of business-to-business situations. The Sequence Flows can cross the boundaries between Lanes of a Pool but cannot cross the boundaries of a Pool. That is, a Process is fully contained within the Pool. The interaction between Pools is shown through Message Flows. Another aspect of Pools is whether or not there is any Activity detailed within the Pool. Thus, a given Pool MAY be shown as a ‘White Box,’ with all details (e.g. a Process) exposed, or as a ‘Black Box,’ with all details hidden (see [2] page 45). No Sequence Flows are associated with a ‘Black Box’ Pool, but Message Flows can attach to its boundaries. A Pool is a square-cornered rectangle with a label name separated from the contents of the Pool by a single line. It is recommended that all pools have a boundary. To help with the clarity of the Diagram, a Pool extends horizontally the entire length of the Diagram. However, there is no specific restriction to the size and/or positioning of a Pool. |  |
| 1. **Lane** | A Lane is a sub-partition within a Process, sometimes within a Pool, and will extend horizontally the entire length of the Process (see [2] page 305). Lanes are used to organise and categorise Activities. Lanes are often used for such things as internal roles (e.g. Manager, Contact Centre Staff), systems (e.g. an application), or internal business units (e.g. Finance). A Lane is a square-cornered rectangle with a solid single line. This guideline does ***not*** recommend the use of nested lanes. |  |
| 5. Artefact | **Provides additional information about the Process** | |
| 1. **Group** | A Group is a grouping of graphical elements that are within the same Category (see [2] page 68). It provides a visual mechanism to group elements of a diagram informally. This type of grouping does not affect the Sequence Flows within the Group. The Category name appears on the diagram as the group label. Categories can be used for documentation or analysis purposes. Groups are one way in which Categories of objects can be visually displayed on the diagram. As an Artefact, a Group is not an Activity or a Flow Object, and, therefore, cannot connect to Sequence Flows or Message Flows. In addition, Groups are not constrained by restrictions of Pools and Lanes. This means that a Group can stretch across the boundaries of a Pool to surround Diagram elements, often to identify Activities that exist within a distributed business-to-business transaction. Groups are often used to highlight certain sections of a Diagram without adding additional constraints for performance as a Sub-Process would. The highlighted (grouped) section of the Diagram can be separated for reporting and analysis purposes. A Group is a rounded corner rectangle with a dashed line. |  |
| 1. **Text Annotation** | Text Annotations (attached with an Association) are a mechanism for a modeller to provide additional text information for the reader of a BPMN Diagram (see [2] page 71). It does not affect the flow of the Process. A Text Annotation is an open rectangle with a solid line ‘bracketing’ the text. |  |

# Queensland Government business process model conventions

## Process descriptions

Providing a textual description of the business process is essential to ensure understanding and maintenance of process models.

It is strongly recommended that the business process is fully described as per the information and data elements detailed in Business process description in Appendix D on page 28 of this guideline.

## Business processes and sub-processes

When naming business processes and sub-processes:

* Specify ***'what'*** is being done – the objective or outcome of the process.
* An acronym may be used in a process name, however, it should be fully spelled out the first time it is used and also in the accompanying process description. If there are several acronyms, consider associating a text annotation or including a legend in the model diagram.
* Use title case ― each word begins with a capital letter followed by lower case letters for the rest of the word. See The case to be used for element names on page 25 of this guideline.
* Avoid the use of the business area or organisational unit in the name of the process. If it seems difficult to name the process, carefully consider the objective of the process. Note that the business area name may be needed in order to differentiate two similar processes in different business areas.
* To distinguish between the current, future and archived process models use version numbers for the process models – for example, use version numbers when there is more than one to-be state, to test and accommodate when to-be becomes the current state, for evidence of when the to-be process became the current process, or for performance management comparisons between different versions of the process. The advantage of version numbering over labelling models with ‘as-is’ and ‘to-be’ is that the labels do not need to be updated when the to-be model is implemented and becomes the as-is process.
* Examples: ‘Travel Booking’, ‘Incident Management’ [6] and ‘Smart Service Queensland (SSQ) Policy Development v1.0’.
* The following diagrams are examples of expanded sub-processes.

### Examples of Expanded Sub-Processes



Figure 1: Expanded sub-process (standard)

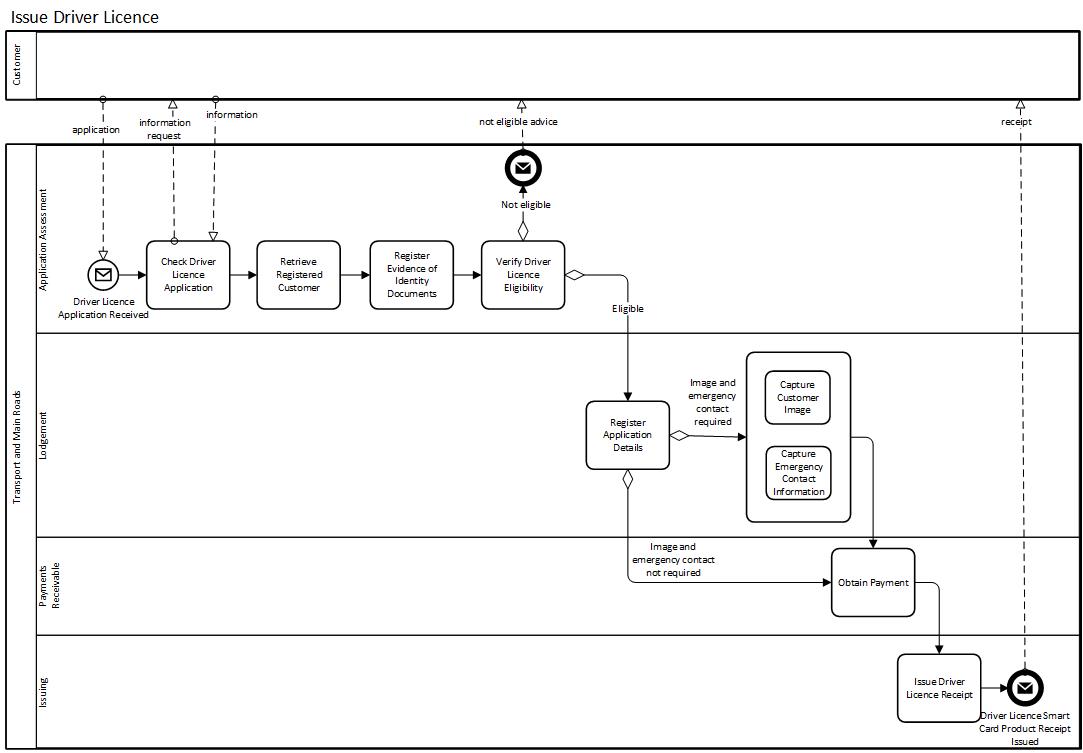


Figure 2: Expanded sub-process (parallel)

## Tasks

When naming tasks:

* Specify ***'what'*** is being done – the objective or outcome of the task. Avoid names that indicate who, when, where or how something is done. For example, Review Patient Chart, Order Diagnostic Test.
* For the business view of a process, tasks should reflect the business actions, not an information system or a system activity. System activities are modelled as the operational view of a process.
* Begin with a verb followed by a business object name, which can be qualified with an adjective if necessary. For example, Endorse Project Proposal, Renew Seniors Card, Obtain Payment.
* Do not start with generic verbs such as Process, Handle, Record, Manage, or Maintain, etc. Use words such as Issue, Reissue, Renew, Register or Assess that indicate the objective or outcome of the process. Refer to the following section 5.4 Words to Avoid in the Business View of a Process.
* Do not include the words *to, and, for, from*, etc. in the names. Refer to the following section 5.4 Words to Avoid in the Business View of a Process
* Be precise and concise. The name should ideally be three words and never more than five words unless there is a very good need for it. The longer the name the more cluttered the diagram, which reduces readability.
* Use title case ― each word begins with a capital letter followed by lower case letters for the rest of the word.
* An acronym may be used in a task name, however the acronym should be fully spelt out the first time it is used. It also needs to be consistently used throughout the process and in the accompanying process description. If there are several acronyms, consider associating a text annotation or including a legend in the model diagram. For example, ‘Amend Smart Service Queensland (SSQ) Information Asset Register (IAR)’.
* Colours, prefixes or suffixes may be used to differentiate between aspects of a task such as ownership, and version control. However, this guideline supports the use of black and white printing, so tasks should use a means other than colour to convey meaning.
* Task directives (policies, procedures, work instructions, standards, legislation, etc.) may be captured in the task's description in the process documentation.
* Task name examples: Surrender Marine Licence, Lodge Industry Authority Application.

## Words to avoid in the business view of a process

The table below lists **verbs** which should **not** be used in the name of a process, sub-process or task in a business view of a process.

Choose a replacement verb carefully so that it reflects the objective of the process.

|  |  |  |
| --- | --- | --- |
| Verb | Do not use because… | Recommended Replacement |
| 1. Update | 1. Too technical / system focussed. | 1. Amend, Change |
| 1. Create | 1. Too technical / system focussed. | 1. Generate, Produce |
| 1. Read | 1. Too technical / system focussed. | 1. Retrieve, Obtain |
| 1. Delete | 1. Too technical / system focussed. | 1. Remove |
| 1. Record | 1. Too technical / system focussed. | 1. Capture, Log, Register |
| 1. Download | 1. Too technical / system focussed. | 1. Retrieve |
| 1. Transmit | 1. Too technical / system focussed. | 1. Send |
| 1. Process 2. Handle 3. Manage | 1. Too generic, would not reflect the specific objectives of the process. | 1. Depends on the objective of the process but could be 2. Assess, Coordinate, Review |
| 1. Maintain | 1. Usually indicates a Create, Read, Update, Delete (CRUD) transaction that is ICT-specific not business. Additionally, if it does indicate CRUD then there would be separate business events, so one process would be incorrect. | 1. Depends on the objective of the process |
| 1. Receive | 1. The recipient must do more than just receive some information. It should be what the receiver does after receiving the information. | 1. Depends on the objective of the process |
| 1. Input | 1. Implies data entry. Why are we entering the information? | 1. Depends on the objective of the process |

The table below lists **words** which should **not** be used in process, sub-process or task names.

|  |  |
| --- | --- |
| Word | Do not use because… |
| 1. and, or | 1. Indicates more than one process. |
| 1. to, for, by | 1. Implied by the lane it is flowing to. |
| 1. data | 1. Too technical. Use information or details. |
| 1. from, of, when | 1. Should be detailed in the process description documentation. |

## Events

If the event type is known (e.g. if it is a message or timer) then the appropriate trigger type should be displayed for the event.

All events are to be named. An event name should:

* Clearly reflect the type of event (start, end, timer, etc.), for example:

Start event: Invoice Received, Application Received (suggests that it is a message)

End event: Urgent Order Sent, Application Rejection Advice Sent (suggests that it is a message)

Timer event: Monthly, Weekly, Invoice Due, 9am on 1st of month, etc. (suggests that it is a timed event)

* Be in title case ― each word begins with a capital letter followed by lower case letters for the rest of the word (as shown in the examples above), and
* Not include the role, document, system or application name in the business view of the process.

1. Start and end event names (not timer events) should:

* Begin with either a noun or a noun prefixed by a qualifier and end with a past tense verb. For example, Invoice Received, Urgent Order Sent, File Received, Driver Licence Application Received.

The naming of throw and catch link intermediate events used to fragment large models is described in section 5.9.1 Fragmentation on page 22.

## Pools and swim lanes

When naming pools:

* use title case ― each word begins with a capital letter followed by lower case letters for the rest of the word.
* organisational unit names are permitted within pools.
* examples of pools are Queensland State Archives (QSA), Information Technology Branch (ITB).

When naming swim lanes:

* use title case ― each word begins with a capital letter followed by lower case letters for the rest of the word.
* use functional area names rather than organisational unit names whenever possible.
* system or application names are permitted within swim lanes.
* examples of swim lanes are Order Processing, Customer Service Centre, Finance System, Risk Management System.

Note that:

* when modelling an external organisation as a black box, there are no lanes (or any other details) in the pool.
* white boxes have at least one lane.
* best to avoid but if you have to – Pools and lanes may be repeated in the one diagram provided the name of the repeated pool or lane is *italicised*.

## Connecting objects

For ease of readability, sequence and message flows should be drawn from left to right and top to bottom.

### Message Flows

* Use a noun and consider using an adjective, for example, valid sales order, invoice details, etc., but do not use a verb, e.g. ‘valid sales order sent’ is incorrect; ‘valid sales order’ is correct.
* A message flow should be named in a way that describes what is being sent or received and not the action itself. For example, product request, invoice details, payment receipt, etc.
* A message flow name should be in lower case.
* This guideline recommends placing the message flow name within the 'white' space between pools.

### Sequence flows

* Sequence flows to/from a task should be positioned as follows:

Input flows come in from the top or the left

Output flows go out at the bottom or the right.

* Only name a sequence flow if it is a conditional flow. A conditional flow name should contain the name of the condition being applied. For example, ‘Application approved’ could be a condition for one flow and ‘Application refused’ for the other.
* Conditional flows are in sentence case – the first word begins with a capital letter followed by lower case letters for the rest of the word; each subsequent word is in lower case. See The Case to be used for Element Names on page 25 of this guideline.
* The associated business rule which defines the condition should be detailed in the process documentation for the task where the conditional flow originates.

## Data objects

When naming data objects

* Use title case and a noun. The data object name may contain an adjective, for example, Customer Contact Details, Purchase Order, Change Request, but it cannot contain a verb.
* The name of the data object may be qualified by its state (called Data Object Reference), for example, End Project Report [Draft], ICT Strategic Plan [Approved].
* The use of data objects and data object references requires careful consideration – judicious use adds valuable information, but too many and the diagram becomes unreadable.

## Diagram layout

Aim to keep the process models as simple and uncluttered as possible. The level of detail required depends on the objective of the modelling task. Note that the ability to apply the following conventions may be limited by the software tool used.

* For readability, it is recommended that a business process model can have as many tasks that comfortably fit on a single page (preferably A4). If the business process model has several tasks and reduced readability, re-assess the level of detail in the diagram or fragment the end-to-end process (refer to the following section on Fragmentation – Link Intermediate Events).
* All business process models should read left to right.
* Order pools and lanes by the way the tasks flow. That is: top-down and left to right.
* Best to avoid but if you have to – Pools and lanes may be repeated in the one diagram provided the name of the repeated pool or lane is *italicised*.
* Keep tasks and sub processes the same size where possible. If the task name does not fit into the task, either decrease the font size or expand all the tasks and process boxes so that they are the same size. Closely examine the name of the task – is it comprehensible?
* Align all tasks horizontally and vertically as much as possible.

### Fragmentation

#### Link intermediate events

Fragmentation is used to make a process more presentable and readable. Fragmentation breaks a large process at logical points so that the fragments can easily fit onto a page or a screen. The points at which the process is fragmented depend on the process itself but the key guidelines are:

* Use the intermediate event type ‘Link’ to show fragmentation.
* For the fragmented process:

the first fragment starts with a Start event and has an intermediate Link Throwing Event at its end

subsequent fragments (except the last) have an intermediate Link Catching Event at the start and an intermediate Link Throwing Event at the end

the last fragment has an intermediate Link Catching Event at the start and an End event at the end.

* Ensure each link event has the name of the process that it links to or from. Name the throwing event ‘To Process\_Name’ and the catching event ‘From Process\_Name’ for example, To Issue New Licence, From Change Licence Details.
* Many throwing links can go to a single catching link.
* Each throwing link can go to only one catching link.

#### Sub-processes

If a process could be reused, then typically it is a sub-process rather than a process.

# References

1. Wikipedia, Business Process Model and Notation, accessed 13 March 2013

<http://en.wikipedia.org/wiki/BPMN>

1. OMG, Business Process Model and Notation Version 2.0, January 2011

<http://www.omg.org/spec/BPMN/2.0/PDF>

1. OMG, Business Process Model and Notation home page, accessed 13 March 2013

<http://www.bpmn.org/>

1. OMG, Semantics of Business Vocabulary and Rules (SBVR), accessed 13 March 2013

<http://www.omg.org/spec/SBVR/index.htm>

1. Queensland Government Business Process Discover Report template v1.0.0 January 2011

<https://portal.qgcio.qld.gov.au/products/qg-methodologies/570-workflow/3302-business-process-improvement>

1. BPMN v2.0 Examples, 2 June 2010

<http://www.omg.org/spec/BPMN/20100601>

## Document history

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Date | Status | Key changes made | Author/s | |
| 1.0.0 | 28/05/2009 | Final | Published on QGCIO Methodologies website | | QGCIO | |
| 1.0.2 | 23/12/2011 | Draft | Revised using QGCIO guideline template | | QGCIO | |
| 1.0.3 | 8/04/2013 | Draft | Revised to align with BPMN 2.0, provided to QG BPMN WG 10 April Meeting | | DSITIA | |
| 1.0.4 | 30/04/2013 | Draft | Revised based on comments at 10 April meeting, provided to QG BPMN WG for 30 April meeting. | | DSITIA | |
| 1.0.5 | 13/05/2013 | Draft | Revised based on comments at 1 May meeting, provided to QG BPMN WG for 15 May meeting. | | DSITIA | |
| 1.0.6 | 27/05/2013 | Draft | Revised based on comments at 15 May meeting, provided to QG BPMN WG for 29 May meeting | | DSITIA | |
| 1.0.7 | 11/06/2013 | Draft | Revised based on comments at 29 May meeting, provided to QG BPMN WG for 13 June meeting | | DSITIA | |
| 1.0.8 | 23/06/2013 | Draft | Revised based on comments at 13 June meeting, provided to QG BPMN WG for 26 June meeting | | DSITIA | |
| 1.0.9 | 26/06/2013 | Draft | Revised based on comments at 26 June meeting. | | DSITIA | |
| 1.0.10 | 6/05/2014 | Draft | Version provided to records management staff. | | DSITIA | |
| 1.0.11 | 7/07/2014 | Draft | Content placed in latest QGEA template. | | DSITIA | |
| 1.0.12 | 12/02/2015 | Draft | Version provided to WG for comments. | | DSITIA | |
| 1.0.13 | 16/03/2015 | Draft | Version provided to QGEA Reference Group for comments. | | DSITI | |
| 1.0.14 | 23/12/2015 | Final | Updated to newer template.  Provided to QGCIO for publication. | | DSITI | |

1. Table of event symbols

Use of event symbols is recommended as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **Start** | **Intermediate** | | **End** |
|  | Catching | | Throwing | |
| **For business and operational views of processes** | | | | |
| None |  |  |  |  |
| Link |  |  |  |  |
| Message |  |  |  |  |
| Timer |  |  |  |  |
| **In addition to the above, for operational views of processes** | | | | |
| Conditional |  |  |  |  |
| Error |  |  |  |  |
| Signal |  |  |  |  |
| Terminate |  |  |  |  |

1. The case to be used for element names

|  |  |
| --- | --- |
| Element | Case |
| Conditional sequence flow | Sentence case |
| Data Object | Title Case |
| Data Object Reference | Title Case |
| Data Store | Title Case |
| Event | Title Case |
| Gateway | Title Case |
| Group | Title Case |
| message flow | lower case |
| Pool | Title Case |
| Process | Title Case |
| Sub-Process | Title Case |
| Swim Lane | Title Case |
| Task | Title Case |
| Text annotation | Sentence case |

1. Gateways and conditional flows

The following information may be useful from a business view of BPMN.

A **Gateway** is used to control how a Process flows along Sequence Flows as they converge and diverge within a Process. If the flow does not need to be controlled, then a Gateway is not needed. The term ‘gateway’ implies that there is a gating mechanism that either allows or disallows passage through the Gateway. Thus, the Gateway will determine branching, forking, merging, and joining of paths [2].

A **Conditional Flow** is a Sequence Flow which has a conditional Expression which is evaluated at runtime to determine whether or not the Sequence Flow will be used [2]. A Conditional Flow which is outgoing from an Activity such as a Task, has a mini-diamond at the beginning of the connector.

Thus, both Gateways and Conditional Flows can be used to model decision points and each has its own advantages. So how do you decide which element to use?

Gateways:

* Tend to be more universally understood by business users as the diamond shape has a history of use in workflow diagrams.
* Can flow to a second gateway, whereas it is not possible to join two sequential flows. However, joining gateways is not encouraged.
* Must be used when no condition exists, as a conditional flow would not be possible. For example, a parallel gateway would be used when several tasks are to be performed in parallel.
* Can be used to replace simple tasks that result in a decision point. The gateway is then labelled with a summary question.

Conditional Flows:

* Are useful in saving space when the model contains many conditions.

This guideline does not recommend one method over the other. However, it does recommend taking into account:

* Organisational culture – use the method that best suits the organisation and the audience.
* Consistency – where possible, use all conditional flows or all gateways in the one diagram. However, there are occasions where using both elements in the one diagram is advantageous.
* Simplicity – gateways are exclusive by default so for exclusive gateways, always omit the ‘X’ marker.
* Simplicity – describe the business rules associated with the decision point in the process description document. Do not include any details of the business rules in the process model.
* Simplicity – for the business view of a process the exception-free scenario is depicted. This often requires minimal use of decision points (i.e. gateways or conditional flows). Operational views of processes which must include the specific steps, decisions and actions are more likely to include several decision points.
* Responsibility for the decision – a gateway can be placed in any swim lane, whereas a conditional flow is in the lane of the task from which it is linked.
* Point – do gateways need to be preceded by the task which makes the decision? This would avoid having a model with 2 sequential gateways. One solution is that the business view of a simple decision task could have gateways (with a question), whereas operational views are best modelled with an explicit task before the gateway.

|  |  |
| --- | --- |
| **Option 1: Decisions are represented by Conditional Flows and associated business rules.** | **Option 2: Decisions are represented by Gateways and associated business rules.** |
| Associated business rule in the process description document:  IF (seller is authenticated AND item-is-verified-by-trading-authority) OR (item-has-been-auctioned-before-by-our-organisation) THEN  Item Allowed for Sale | |

1. Business process description

This description was previously captured in the Current state section of the Queensland Government [Business Process Discover Report](http://methodologies.govnet.qld.gov.au/Methodology/BusinessProcessManagement/BPI/Documents/Business%20Process%20Discover%20Report.doc) v2.0 (no longer available).

**Business area functions**

*Insert a brief description of the functions of the business area.*

**Consultation list**

*Insert a brief description of the stakeholders consulted.*

**Current state**

**<Process 1>**

*Replicate this complete section for each process in scope.*

*Replace <Process 1> with Business Process name.*

*Name must start with a verb.*

### Business process model

*Insert current state process model diagram.*

### Business process description

*High level description of the end to end process.*

### Business process detail

|  |
| --- |
| **Process owner:** *Position and name of nominated process owner.* |
| **Process directives:** *List of legislation, policies, standards, procedures, work instructions that support and/or constrain this process.* |
| **Scope of use (internal):** *The number and type of participants involved in and/or impacted by this process. Include a brief description of any business areas not already covered in the ‘Business area function’ above.* |
| **Scope of use (external):** *The number and type of participants involved in and/or impacted by this process. Include a brief description of participants.* |
| **Frequency of use:** *State how often this process is performed.* |
| **Constraints:** *Limitations to the possible changes to a process.* |
| **Assumptions:** *Any assumptions made during performing this process or documenting this process.* |
| **Information capture:** *The mechanisms used to capture the information to support the process (e.g. paper or electronic forms, information manually entered into an application).* |
| **Associated application:** *The associated application/s name.* |
| **Upstream process:** *The process which occurs before this.* |
| **Downstream process:** *The process which occurs after this.* |
| **Issues:** *Short description of problems with this process.* |
| **Metrics:** *Metrics used for this process e.g. Percent outstanding requests more than 5 days old.* |
| **Baseline measures:** *Current values e.g. 25% outstanding requests more than 5 days old.* |
| **Target measures:** *Target values e.g. Less than 3% outstanding requests more than 5 days old.* |
| **Reporting:** *Describe any reporting associated with this process.* |
| **Notes:** *Any additional notes captured.* |

**Task descriptions**

*Insert/add additional tasks as required.*

|  |  |
| --- | --- |
| **Start event** | *Name of start event* |
| **Description** | *Description of start event (if necessary), specifically triggers that cause the process to commence* |

|  |  |
| --- | --- |
| **Start event** | *It is rare but a process can potentially be triggered by multiple start events* |
| **Description** | *Description of start event (if necessary), specifically triggers that cause the process to commence* |

|  |  |
| --- | --- |
| **Task name** | *Name of the first task* |
| **Description** | *Brief description of how and why this task is performed, including actions or decisions involved* |
| **Roles** | *If pools or lanes already present enough information, leave blank. Otherwise, provide roles of people involved in enacting the task* |
| **Supporting ICT** | *Any ICT used to support the task* |
| **Business rules** | *The business rules that support or constrain the execution of this task* |
| **Input** | *Optional. Info flows should already have been documented in the process model. Could include names of data objects used as input to support the execution of this task* |
| **Output** | *Optional. Info flows should already have been documented in the process model. Could include names of data objects produced as a result of executing this task* |

|  |  |
| --- | --- |
| **Task name** | *Name of the second task* |
| **Description** | *Brief description of how and why this task is performed, including actions or decisions involved* |
| **Roles** | *If pools or lanes already present enough information, leave blank. Otherwise, provide roles of people involved in enacting the task* |
| **Supporting ICT** | *Any ICT used to support the task* |
| **Business rules** | *The business rules that support or constrain the execution of this task* |
| **Input** | *Info, materials, products, services, input frequency and volume* |
| **Output** | *Info, materials, products, services, output frequency and volume* |

|  |  |
| --- | --- |
| **End event** | *Name of end event* |
| **Description** | *Description of end event (if necessary) – including any links to other processes that might follow from the completion of this process* |

**Process problems**

*Insert any problems currently experienced with the process.*

A process problem is something that *has already happened* during a process. For example, mail arrives late and therefore is distributed too late in the day for directors to action before close of business.

**Risk exposure**

*Insert any risk exposure associated with the current state.*

A risk is something which *could happen* during a process and has a consequence. For example, a major fire in the building is relatively unlikely to happen, but would have enormous impact on Business Continuity. Conversely, occasional car breakdowns are fairly likely to happen, but would not usually have a major impact on the organisation.

**Assumptions**

*Insert any assumptions made in observing the current state.*

An assumption is something a process modeller has assumed during the documentation of a process. For example, it has been assumed the second approval in this process is only required if the first approval is rejected.

**Queries**

*Insert any unresolved questions about process uncertainties or information gaps.*

A query is something a process modeller has raised a question about during the documentation of a process. For example, does the director complete this process if the senior manager is unavailable or away?

**Exceptions**

*Insert any exceptions to the process.*

An exception is something the business has made clear during the collection of information on a process. For example, this process is not completed if the minister formally requests the grant to be approved regardless of restrictions.

1. Please note that since BPMN 2.0 the abbreviation BPMN refers to ‘Business Process Model and Notation’ and not “Business Process Modelling Notation”. [↑](#footnote-ref-2)
2. All page references are to the specification [2] unless otherwise noted. [↑](#footnote-ref-3)