BPMN-js Complete Reference Sheet

A Comprehensive Guide to Business Process Model and Notation Symbols

Author: Manus AI

Date: August 18, 2025

Table of Contents

Introduction to BPMN-js {#introduction}

Business Process Model and Notation (BPMN) 2.0 is the international standard for modeling business processes, providing a graphical notation that is easily understood by all business stakeholders [1]. BPMN-js is a powerful JavaScript library that enables the creation, rendering, and manipulation of BPMN 2.0 diagrams directly in web browsers [2]. This comprehensive reference sheet covers all BPMN symbols supported by bpmn-js, their meanings, appropriate usage contexts, and practical examples.

BPMN serves as a bridge between business process design and technical implementation, offering a standardized visual language that facilitates communication between business analysts, process designers, developers, and stakeholders [3]. The notation uses specific graphical elements to represent different aspects of business processes, from simple tasks to complex orchestrations involving multiple participants.

The power of BPMN lies in its ability to model processes at different levels of detail while maintaining clarity and consistency. Whether you're documenting a simple approval workflow or designing a complex multi-party collaboration, BPMN provides the necessary symbols and constructs to accurately represent your business logic [4].

Flow Objects {#flow-objects}

Flow Objects are the core elements that define the behavior and flow of a business process. They represent the main components that drive process execution and include Events, Activities, and Gateways [5]. These elements work together to create a logical sequence that describes how work flows through an organization.

Events {#events}

Events represent something that happens during the course of a business process. They affect the flow of the model and usually have a cause (trigger) or an impact (result) [6]. Events are depicted as circles with different internal markers to indicate their specific type and behavior.

Event Categories

Events are categorized into three main types based on when they occur in the process flow:

Start Events (Single thin circle): These events trigger the instantiation of a process. They indicate where a particular process will start and define the trigger that initiates the process flow [7].

Intermediate Events (Double thin circles): These events occur between Start and End events. They represent something that happens during the process and can either catch (wait for) or throw (send) triggers [8].

End Events (Single thick circle): These events indicate where a process path will end. They define the result or outcome when a process path is completed [9].

Event Types by Trigger

None Events

- **Symbol**: Circle with no internal marker
- **Meaning**: Generic events without specific triggers

- **When to use**: For simple process starts/ends or intermediate points without specific conditions
- **Example**: A basic "Process Started" or "Process Completed" event

Start Event Example: Use when a process begins automatically without external triggers, such as a daily batch processing routine.

Intermediate Event Example: Use as a milestone marker in a process, such as "Phase 1 Completed."

End Event Example: Use when a process concludes normally without specific outcomes, such as "Request Processed."

Message Events

- **Symbol**: Circle with envelope icon
- **Meaning**: Events triggered by or sending messages between processes or participants
- **When to use**: For inter-process communication, notifications, or when waiting for external input
- **Example**: Receiving a customer order, sending an approval notification

Start Event Example: A customer service process that begins when a support ticket is received via email.

Intermediate Event Example: Waiting for manager approval during an expense reimbursement process.

End Event Example: Sending a confirmation email to a customer after order completion.

Timer Events

- **Symbol**: Circle with clock icon
- **Meaning**: Events triggered by time-based conditions (specific time, duration, or cycle)
- **When to use**: For scheduled processes, timeouts, or recurring activities
- **Example**: Monthly report generation, payment due reminders, session timeouts

Start Event Example: A payroll process that automatically starts on the 1st of each month.

Intermediate Event Example: A 24-hour waiting period before escalating an unresolved issue.

End Event Example: Automatically closing a case after 30 days of inactivity.

Conditional Events

- **Symbol**: Circle with document/list icon

- **Meaning**: Events triggered when specific conditions are met
- **When to use**: For condition-based process initiation or branching
- **Example**: Starting a review process when inventory falls below threshold

Start Event Example: Initiating a reorder process when stock levels drop below minimum quantity.

Intermediate Event Example: Triggering additional verification when transaction amount exceeds $10,000.

Signal Events

- **Symbol**: Circle with triangle icon
- **Meaning**: Events that broadcast or receive signals across multiple processes
- **When to use**: For coordinating multiple processes or broadcasting status changes
- **Example**: Emergency shutdown signal, system maintenance notification

Start Event Example: Multiple processes starting simultaneously when a "new quarter" signal is broadcast.

Intermediate Event Example: All active processes receiving a "system maintenance" signal to pause operations.

End Event Example: Broadcasting a "project completed" signal to notify all stakeholders.

Error Events

- **Symbol**: Circle with lightning bolt icon
- **Meaning**: Events that handle error conditions or exceptions
- **When to use**: For exception handling and error recovery scenarios
- **Example**: Database connection failure, validation errors, system exceptions

Start Event Example: An error handling subprocess that starts when a system error occurs.

Intermediate Event Example: Catching a timeout error during external service calls.

End Event Example: Terminating a process due to a critical system error.

Escalation Events

- **Symbol**: Circle with arrow pointing up
- **Meaning**: Events for escalating issues to higher authority levels
- **When to use**: For non-interrupting escalations that require management attention
- **Example**: Escalating overdue tasks, budget overruns, quality issues

Start Event Example: Starting a management review process when project delays are detected.

Intermediate Event Example: Escalating to senior management when resolution time exceeds SLA.

Termination Events

- **Symbol**: Circle with filled circle (bull's eye)
- **Meaning**: Events that immediately terminate all process instances
- **When to use**: For emergency stops or immediate process cancellation
- **Example**: Emergency shutdown, critical system failure, regulatory compliance violation

End Event Example: Immediately terminating all related processes when a security breach is detected.

Compensation Events

- **Symbol**: Circle with rewind/backward arrows
- **Meaning**: Events for handling compensation or rollback scenarios
- **When to use**: For undoing completed activities or handling transaction rollbacks
- **Example**: Refund processing, inventory adjustment, service credit application

Start Event Example: Starting a refund process when a customer cancels an order.

Intermediate Event Example: Triggering compensation activities when a payment fails after order fulfillment.

End Event Example: Completing a compensation process after successful rollback.

Cancel Events

- **Symbol**: Circle with X mark
- **Meaning**: Events for canceling transactions or processes
- **When to use**: Specifically for transaction boundaries and cancellation scenarios
- **Example**: Transaction cancellation, order cancellation, booking cancellation

Intermediate Event Example: Canceling a reservation when payment authorization fails.

End Event Example: Completing a process with cancellation status.

Link Events

- **Symbol**: Circle with arrow/link icon

- **Meaning**: Events for connecting separate parts of a diagram
- **When to use**: For improving diagram readability by avoiding crossing sequence flows
- **Example**: Connecting distant parts of complex processes

Intermediate Event Example: Using link events to connect process flows across different pages or sections of a large diagram.

Multiple Events

- **Symbol**: Circle with pentagon (five-sided shape)
- **Meaning**: Events that can be triggered by multiple different triggers
- **When to use**: When any one of several triggers can initiate or continue the process
- **Example**: A process that can start via email, phone call, or web form submission

Start Event Example: A customer service process that can begin through multiple channels (email, phone, chat, or web form).

Intermediate Event Example: Waiting for either customer response or timeout to proceed with order processing.

Parallel Multiple Events

- **Symbol**: Circle with plus sign
- **Meaning**: Events that require all specified triggers to occur
- **When to use**: When multiple conditions must be met simultaneously
- **Example**: Process requiring both manager approval AND budget availability

Start Event Example: A project initiation process that requires both client approval and resource availability.

Intermediate Event Example: Proceeding only when both technical review and legal review are completed.

Activities {#activities}

Activities represent work that is performed within a business process. They are the fundamental building blocks that describe what needs to be done and can be either atomic (Tasks) or compound (Subprocesses) [10]. Activities are depicted as rounded rectangles with various internal markers to indicate their specific type and behavior.

Task Types

Tasks are atomic activities that cannot be broken down into finer levels of detail. They represent a single unit of work performed by a person, system, or service [11].

Task (Generic)

- **Symbol**: Rounded rectangle with no internal marker
- **Meaning**: Basic work unit without specific implementation details
- **When to use**: For high-level process modeling or when implementation details are not yet defined
- **Example**: "Review Application," "Process Payment," "Generate Report"

Usage Example: In early process design phases, use generic tasks to outline the main work steps before defining specific implementation approaches.

User Task

- **Symbol**: Rounded rectangle with person icon
- **Meaning**: Tasks performed by human users through software applications
- **When to use**: For activities requiring human judgment, decision-making, or interaction with systems
- **Example**: Data entry, document review, customer service interactions, approval decisions

Practical Example: "Review Loan Application" - A loan officer uses a banking system to evaluate creditworthiness, requiring human expertise to assess risk factors and make approval decisions.

Service Task

- **Symbol**: Rounded rectangle with gear/cog icon
- **Meaning**: Automated tasks performed by software services, web services, or applications
- **When to use**: For system-to-system interactions, API calls, or automated processing
- **Example**: Credit score lookup, payment processing, email notifications, database updates

Practical Example: "Validate Credit Score" - An automated service calls a credit bureau API to retrieve and validate the applicant's credit score without human intervention.

Send Task

- **Symbol**: Rounded rectangle with filled envelope icon

- **Meaning**: Tasks that send messages to external participants or processes
- **When to use**: For one-way communication where no response is expected
- **Example**: Sending notifications, alerts, or status updates

Practical Example: "Send Approval Notification" - Automatically sending an email to inform a customer that their loan application has been approved.

Receive Task

- **Symbol**: Rounded rectangle with empty envelope icon
- **Meaning**: Tasks that wait for messages from external participants or processes
- **When to use**: For receiving information, documents, or signals from external sources
- **Example**: Waiting for customer documentation, receiving system responses, getting approval confirmations

Practical Example: "Receive Supporting Documents" - Waiting for the customer to submit required documentation before proceeding with loan processing.

Manual Task

- **Symbol**: Rounded rectangle with hand icon
- **Meaning**: Tasks performed manually without software assistance
- **When to use**: For physical activities or work done without system support
- **Example**: Physical document filing, manual inspections, offline activities

Practical Example: "File Physical Documents" - Manually organizing and storing hard copies of loan documents in physical filing cabinets.

Business Rule Task

- **Symbol**: Rounded rectangle with table/grid icon
- **Meaning**: Tasks that execute business rules or decision logic through rule engines
- **When to use**: For complex decision-making based on predefined business rules
- **Example**: Risk assessment, pricing calculations, eligibility determination

Practical Example: "Calculate Loan Interest Rate" - Using a business rules engine to determine the appropriate interest rate based on credit score, loan amount, term, and current market conditions.

Script Task

- **Symbol**: Rounded rectangle with script/code icon

- **Meaning**: Tasks executed through scripts or code by the process engine
- **When to use**: For simple calculations, data transformations, or lightweight automation
- **Example**: Data formatting, simple calculations, variable assignments

Practical Example: "Calculate Monthly Payment" - Executing a script to compute monthly payment amounts based on loan principal, interest rate, and term length.

Subprocess Types

Subprocesses are compound activities that contain other flow objects and represent a collection of related activities [12].

Subprocess (Collapsed)

- **Symbol**: Rounded rectangle with plus sign at bottom
- **Meaning**: A subprocess that contains other activities but is shown in collapsed form
- **When to use**: To hide complexity while maintaining process overview
- **Example**: "Customer Onboarding," "Order Fulfillment," "Quality Assurance"

Usage Example: "Complete KYC Process" - A collapsed subprocess containing multiple verification steps (identity check, address verification, document validation) shown as a single activity in the main process flow.

Subprocess (Expanded)

- **Symbol**: Rounded rectangle containing other BPMN elements
- **Meaning**: A subprocess showing its internal flow and activities
- **When to use**: When you need to show the detailed steps within a subprocess
- **Example**: Detailed view of onboarding steps, payment processing workflow

Usage Example: An expanded "Loan Approval Process" showing the complete flow of credit checks, document verification, risk assessment, and decision-making activities.

Event Subprocess

- **Symbol**: Rounded rectangle with dotted border
- **Meaning**: A subprocess triggered by events and runs in parallel with the main process
- **When to use**: For exception handling, monitoring, or parallel activities triggered by events
- **Example**: Error handling, escalation procedures, audit logging

Usage Example: An "Escalation Handler" event subprocess that monitors loan processing time and automatically escalates to management if processing exceeds SLA thresholds.

Call Activity

- **Symbol**: Rounded rectangle with thick border and plus sign
- **Meaning**: References a reusable process or global task defined elsewhere
- **When to use**: For reusing common processes across multiple workflows
- **Example**: Standard verification procedures, common approval workflows

Usage Example: "Perform Credit Check" - A call activity that references a standardized credit verification process used across multiple loan products.

Transaction

- **Symbol**: Rounded rectangle with double border
- **Meaning**: A specialized subprocess with transaction semantics (all-or-nothing execution)
- **When to use**: For activities that must complete entirely or be rolled back completely
- **Example**: Financial transactions, multi-step operations requiring atomicity

Usage Example: "Process Payment Transaction" - Ensuring that both account debiting and credit posting complete successfully, or both are rolled back if either fails.

Activity Markers

Activity markers are small icons added to activities to indicate special behaviors or characteristics [13].

Loop Marker

- **Symbol**: Circular arrow icon at bottom of activity
- **Meaning**: The activity repeats sequentially until a condition is met
- **When to use**: For iterative processes or retry mechanisms
- **Example**: Retrying failed service calls, processing items in a queue

Usage Example: "Attempt Payment Processing" with loop marker - Retry payment processing up to three times if initial attempts fail due to temporary network issues.

Parallel Multi-Instance Marker

- **Symbol**: Three parallel vertical lines at bottom of activity

- **Meaning**: Multiple instances of the activity execute simultaneously
- **When to use**: For parallel processing of similar tasks
- **Example**: Processing multiple documents simultaneously, parallel approvals

Usage Example: "Review Applications" with parallel multi-instance marker - Multiple loan officers simultaneously reviewing different applications to improve processing speed.

Sequential Multi-Instance Marker

- **Symbol**: Three horizontal lines at bottom of activity
- **Meaning**: Multiple instances of the activity execute one after another
- **When to use**: For sequential processing where order matters
- **Example**: Sequential approval levels, ordered processing steps

Usage Example: "Obtain Approvals" with sequential multi-instance marker - Sequential approval from team lead, department manager, and senior management in that specific order.

Ad Hoc Marker

- **Symbol**: Tilde (~) symbol at bottom of activity
- **Meaning**: Activities within can be performed in any order or repeated as needed
- **When to use**: For flexible processes where sequence is not predetermined
- **Example**: Research activities, creative processes, problem-solving sessions

Usage Example: "Gather Additional Information" with ad hoc marker - Loan officers can collect various supporting documents in any order based on case-specific requirements.

Compensation Marker

- **Symbol**: Rewind/backward arrow at bottom of activity
- **Meaning**: The activity is used for compensation or rollback purposes
- **When to use**: For activities that undo or compensate for previous actions
- **Example**: Refund processing, inventory adjustments, service credits

Usage Example: "Reverse Transaction" with compensation marker - Automatically triggered to reverse a payment if subsequent validation steps fail.

Gateways {#gateways}

Gateways are decision points that control how sequence flows interact as they converge and diverge within a process. They determine the branching, forking, merging, and joining of paths based on various conditions [14]. Gateways are depicted as diamond shapes with different internal markers to indicate their specific behavior.

Gateway Behaviors

Gateways can operate in two modes:

- **Split (Diverging)**: One incoming sequence flow is divided into multiple outgoing flows
- **Join (Converging)**: Multiple incoming sequence flows are merged into one outgoing flow

Gateway Types

Exclusive Gateway (XOR)

- **Symbol**: Diamond with X or empty
- **Meaning**: Exactly one path is chosen based on conditions (data-based) or events
- **When to use**: For mutually exclusive decisions where only one option can be selected
- **Example**: Approve/Reject decisions, routing based on customer type, priority-based routing

Split Example: "Check Credit Score" - Route to "Approve Loan" if score > 700, "Request Additional Info" if score 600-700, or "Reject Application" if score < 600. Only one path is taken based on the credit score value.

Join Example: After separate approval and rejection paths, both converge at an exclusive gateway before proceeding to "Send Notification" - only one of the paths will have been active.

Practical Usage: Use exclusive gateways for binary decisions (yes/no), categorization (high/medium/low priority), or any scenario where exactly one option must be chosen from multiple alternatives.

Parallel Gateway (AND)

- **Symbol**: Diamond with plus sign (+)
- **Meaning**: All paths are executed simultaneously (split) or all paths must complete before continuing (join)
- **When to use**: For parallel processing or when multiple conditions must all be satisfied
- **Example**: Parallel document reviews, simultaneous system updates, concurrent approval processes

Split Example: "Begin Loan Processing" - Simultaneously start "Credit Check," "Employment Verification," and "Document Review" processes. All three activities begin at the same time.

Join Example: Wait for completion of "Credit Check," "Employment Verification," and "Document Review" before proceeding to "Make Loan Decision." The process continues only when all three parallel activities are finished.

Practical Usage: Use parallel gateways to improve process efficiency by executing independent activities simultaneously, or when multiple requirements must all be met before proceeding.

Inclusive Gateway (OR)

- **Symbol**: Diamond with circle (O)
- **Meaning**: One or more paths are chosen based on conditions (split) or waits for all active paths to complete (join)
- **When to use**: For scenarios where multiple options can be selected simultaneously
- **Example**: Multiple notification channels, optional additional services, multi-criteria evaluations

Split Example: "Send Notifications" - Based on customer preferences, send notifications via email (if email opted-in), SMS (if phone provided), and/or postal mail (if address on file). Multiple paths can be active simultaneously.

Join Example: After sending notifications through various channels, wait for all active notification processes to complete before proceeding to "Update Customer Record."

Practical Usage: Use inclusive gateways when multiple conditions can be true simultaneously, for optional parallel processes, or when you need flexible routing based on multiple criteria.

Event-Based Gateway

- **Symbol**: Diamond with circle containing pentagon or double circle
- **Meaning**: The gateway waits for one of several possible events to occur, then follows the path of the first event that happens
- **When to use**: For event-driven decisions where the process waits for external triggers
- **Example**: Waiting for customer response, timeout scenarios, first-come-first-served processing

Usage Example: "Wait for Customer Response" - After sending a loan approval offer, wait for either "Customer Accepts" (via signed documents), "Customer Rejects" (via decline notification), or "Timeout" (after 30 days). The first event to occur determines which path the process follows.

Practical Usage: Use event-based gateways when the process must wait for external events, when implementing timeout mechanisms, or when the next step depends on which of several possible events occurs first.

Complex Gateway

- **Symbol**: Diamond with asterisk (*)
- **Meaning**: Uses complex expressions to determine routing behavior that cannot be captured by other gateway types
- **When to use**: For sophisticated routing logic involving multiple variables and complex conditions
- **Example**: Advanced business rules, multi-factor decision making, complex mathematical conditions

Usage Example: "Determine Loan Terms" - Route based on a complex formula considering credit score, debt-to-income ratio, loan amount, collateral value, and market conditions. The routing expression might be: "Route to Premium Terms if (credit_score > 750 AND debt_ratio < 0.3) OR (collateral_value > loan_amount * 1.5)."

Practical Usage: Use complex gateways sparingly, only when the routing logic cannot be adequately expressed using combinations of other gateway types. Document the complex expressions clearly for maintainability.

Gateway Usage Patterns

Decision Points

Gateways are commonly used to represent decision points in business processes where the flow must branch based on data, conditions, or business rules.

Example Pattern: Application Processing

1. Receive Application → Exclusive Gateway (Check Completeness)
2. If Complete → Continue Processing
3. If Incomplete → Request Additional Information → Return to Gateway

Parallel Execution

Use parallel gateways to model activities that can happen simultaneously, improving process efficiency.

Example Pattern: Employee Onboarding

1. New Employee Starts → Parallel Gateway (Split)
2. Parallel Paths: Setup IT Equipment + Prepare Workspace + Schedule Orientation
3. Parallel Gateway (Join) → Begin First Day Activities

Conditional Processing

Inclusive gateways enable flexible processing where multiple conditions can be true simultaneously.

Example Pattern: Marketing Campaign

1. Launch Campaign → Inclusive Gateway
2. Conditional Paths: Send Email (if email available) + Send SMS (if mobile available) + Post Social Media (if social accounts linked)
3. Inclusive Gateway (Join) → Measure Campaign Results

Event-Driven Flows

Event-based gateways handle scenarios where the process must wait for and react to external events.

Example Pattern: Customer Service

1. Send Response to Customer → Event-Based Gateway
2. Wait for: Customer Satisfied (close case) OR Customer Escalates (transfer to supervisor) OR Timeout (follow up)
3. Process continues based on which event occurs first

Best Practices for Gateway Usage

Clarity and Simplicity: Choose the simplest gateway type that accurately represents your business logic. Avoid complex gateways unless absolutely necessary.

Condition Documentation: Clearly document the conditions or rules that determine gateway routing. Use descriptive labels on sequence flows leaving gateways.

Balanced Flows: Ensure that every split gateway has a corresponding join gateway where appropriate to maintain process flow integrity.

Default Paths: For exclusive gateways, always provide a default path to handle unexpected conditions or edge cases.

Testing Scenarios: Consider all possible combinations of conditions when using inclusive gateways to ensure proper process behavior in all scenarios.

Connecting Objects {#connecting-objects}

Connecting Objects link Flow Objects together to create the basic skeletal structure of a business process. They define the order and conditions under which activities are performed [15].

Sequence Flow

- **Symbol**: Solid arrow line
- **Meaning**: Shows the order in which activities are performed within a process
- **When to use**: To connect activities, events, and gateways within the same pool
- **Example**: Task A → Task B → End Event

Usage Guidelines: Sequence flows can only connect elements within the same pool or lane. They represent the normal flow of the process and can have conditions when leaving gateways.

Message Flow

- **Symbol**: Dashed arrow line with circle at start and arrowhead at end
- **Meaning**: Shows the flow of messages between separate participants (pools)
- **When to use**: For communication between different organizations, systems, or process participants
- **Example**: Customer sends order → Company receives order

Usage Guidelines: Message flows cross pool boundaries and represent communication between separate participants. They cannot be used within a single pool.

Association

- **Symbol**: Dotted line
- **Meaning**: Links artifacts (data objects, text annotations) to flow objects
- **When to use**: To show relationships between activities and supporting information
- **Example**: Task associated with required document or explanatory note

Usage Guidelines: Associations are used to connect artifacts to flow elements and do not affect the process flow. They provide additional context and information.

Data Association

- **Symbol**: Dashed arrow line

- **Meaning**: Shows data flow between activities and data objects
- **When to use**: To indicate data input/output relationships
- **Example**: Activity reads from or writes to data store

Usage Guidelines: Data associations show how activities interact with data objects, indicating data dependencies and transformations.

Swimlanes {#swimlanes}

Swimlanes provide a way to organize and categorize activities within a process based on function or responsibility [16].

Pool

- **Symbol**: Large rectangle containing process elements
- **Meaning**: Represents a participant in the process (organization, role, or system)
- **When to use**: To separate different organizations or major participants
- **Example**: Customer pool, Bank pool, Insurance Company pool

Usage Guidelines: Pools represent separate participants and contain their respective processes. Message flows connect different pools, while sequence flows operate within pools.

Practical Example: In a loan application process, create separate pools for "Customer," "Bank," and "Credit Bureau" to clearly show which activities each participant performs.

Lane

- **Symbol**: Subdivision within a pool
- **Meaning**: Represents a role, department, or system within a participant
- **When to use**: To assign responsibility for activities within an organization
- **Example**: Sales Department, IT Department, Manager, System

Usage Guidelines: Lanes subdivide pools to show internal organization and responsibility assignment. Activities placed in lanes indicate who performs them.

Practical Example: Within a "Bank" pool, create lanes for "Loan Officer," "Credit Analyst," "Branch Manager," and "Loan Processing System" to show role-based responsibilities.

Artifacts {#artifacts}

Artifacts provide additional information about the process but do not directly affect the sequence flow [17].

Text Annotation

- **Symbol**: Square bracket with text
- **Meaning**: Provides additional descriptive information
- **When to use**: To add explanations, comments, or clarifications
- **Example**: Business rules, assumptions, constraints

Usage Example: Add a text annotation to explain complex business rules: "Credit score must be above 650 for automatic approval."

Group

- **Symbol**: Dashed rounded rectangle
- **Meaning**: Visual grouping of elements for documentation purposes
- **When to use**: To highlight related activities or create visual organization
- **Example**: Grouping all approval-related activities

Usage Guidelines: Groups have no semantic meaning and don't affect process execution. They're purely for visual organization and documentation.

Data Objects {#data-objects}

Data Objects represent information that flows through the process and show how activities interact with data [18].

Data Object

- **Symbol**: Rectangle with folded corner
- **Meaning**: Information created, manipulated, or used by activities
- **When to use**: To show data inputs, outputs, and transformations
- **Example**: Customer Application, Credit Report, Approval Decision

Usage Example: "Customer Application" data object flows from "Receive Application" to "Review Application" activity, showing data movement through the process.

Data Store

- **Symbol**: Open rectangle (cylinder-like shape)

- **Meaning**: Persistent data storage that exists beyond the process instance
- **When to use**: For databases, file systems, or repositories
- **Example**: Customer Database, Document Repository, Transaction Log

Usage Example: "Customer Database" data store is accessed by multiple activities throughout the loan process to read customer information and update records.

Data Input

- **Symbol**: Data object with arrow pointing into it
- **Meaning**: Data required as input for the process
- **When to use**: To show external data requirements
- **Example**: Market rates, regulatory requirements, customer preferences

Data Output

- **Symbol**: Data object with arrow pointing out of it
- **Meaning**: Data produced by the process
- **When to use**: To show process results and outputs
- **Example**: Loan decision, generated reports, updated customer records

Conversations {#conversations}

Conversations provide a high-level view of message exchanges between participants [19].

Conversation

- **Symbol**: Hexagon
- **Meaning**: Represents a logical grouping of message exchanges
- **When to use**: For high-level process modeling focusing on communication
- **Example**: Loan Application Conversation, Customer Service Conversation

Call Conversation

- **Symbol**: Hexagon with thick border
- **Meaning**: References a global conversation defined elsewhere
- **When to use**: For reusable conversation patterns
- **Example**: Standard inquiry process, common approval workflow

Sub-Conversation

- **Symbol**: Hexagon with plus sign
- **Meaning**: Contains other conversations or can be expanded to show details
- **When to use**: For complex communication patterns with multiple sub-conversations
- **Example**: Complete customer onboarding conversation containing multiple sub-conversations

Quick Reference Tables {#quick-reference}

Event Types Quick Reference

Event Type	Start	Intermediate	End	Common Use Cases
None	○	◎	●	Simple process points, milestones
Message	✉○	✉◎	✉●	Email notifications, system messages
Timer	⏰○	⏰◎	⏰●	Scheduled processes, timeouts, deadlines
Conditional	📋○	📋◎	-	Condition-based triggers, thresholds
Signal	△○	△◎	△●	Broadcast notifications, coordination
Error	⚡○	⚡◎	⚡●	Exception handling, error recovery
Escalation	↗○	↗◎	↗●	Management escalation, SLA breaches
Termination	-	-	⊙●	Emergency stops, immediate termination
Compensation	↶○	↶◎	↶●	Rollback procedures, refunds
Cancel	✖○	✖◎	✖●	Transaction cancellation
Link	🔗○	🔗◎	-	Diagram connectivity, page connectors
Multiple	⬟○	⬟◎	⬟●	Any of several triggers

Parallel Multiple	⊕◯	⊕◎	-	All triggers must occur

Task Types Quick Reference

Task Type	Symbol	Performer	Automation Level	Common Examples
Generic Task	⬜	Unspecified	Varies	High-level activities
User Task	👤⬜	Human + System	Manual with system support	Data entry, reviews, approvals
Service Task	⚙️⬜	System	Fully automated	API calls, calculations, validations
Send Task	📤⬜	System	Automated	Email sending, notifications
Receive Task	📥⬜	System	Automated	Message waiting, document receipt
Manual Task	✋⬜	Human	Fully manual	Physical tasks, offline work
Business Rule Task	📊⬜	Rule Engine	Automated	Decision making, rule evaluation
Script Task	📜⬜	Process Engine	Automated	Simple calculations, data manipulation

Gateway Types Quick Reference

Gateway Type	Symbol	Split Behavior	Join Behavior	Use When
Exclusive (XOR)	◊ or ✖◊	One path only	Wait for one path	Mutually exclusive decisions
Parallel (AND)	⊕◊	All paths	Wait for all paths	Simultaneous activities

Inclusive (OR)	○ ◊	One or more paths	Wait for active paths	Multiple optional conditions
Event-Based	⬟ ◊	First event wins	N/A	Event-driven decisions
Complex	✱ ◊	Complex expression	Complex expression	Sophisticated routing logic

Connecting Objects Quick Reference

Connection Type	Symbol	Purpose	Connects	Crosses Pools
Sequence Flow	→——	Process flow	Flow objects within pool	No
Message Flow	○---→	Communication	Pools/lanes	Yes
Association	·········	Information link	Artifacts to flow objects	No
Data Association	○--→	Data flow	Data objects to activities	No

Swimlane Quick Reference

Element	Symbol	Purpose	Contains	Usage
Pool	Large rectangle	Participant/Organization	Complete process	Separate organizations
Lane	Pool subdivision	Role/Department	Activities and events	Internal organization

Best Practices {#best-practices}

Process Modeling Guidelines

Start Simple: Begin with high-level processes using basic elements (tasks, gateways, events) before adding complexity. This approach ensures clarity and makes the model easier to understand and validate with stakeholders.

Use Meaningful Names: Activity and event names should clearly describe what happens using verb-noun combinations. For example, use "Review Loan Application" instead of just

"Review" or "Application."

Maintain Consistent Granularity: Keep activities at a similar level of detail within the same process diagram. Don't mix high-level strategic activities with detailed operational steps in the same view.

Follow the Happy Path First: Model the normal, successful flow of the process before adding exception handling and error scenarios. This provides a clear foundation for understanding the core process logic.

Gateway Usage Best Practices

Document Decision Criteria: Always label the sequence flows leaving exclusive and inclusive gateways with clear conditions. Use business-friendly language that stakeholders can understand.

Provide Default Paths: Include a default or "else" path from exclusive gateways to handle unexpected conditions. This prevents process instances from getting stuck.

Balance Split and Join: Every gateway that splits the flow should have a corresponding gateway that joins the flow back together, unless the paths lead to different end events.

Avoid Gateway Chains: Don't connect gateways directly to each other without intervening activities. This creates confusion and may indicate that the process logic needs simplification.

Event Modeling Best Practices

Use Start Events Consistently: Every process should have at least one start event. Use different start event types to clearly indicate how the process can be triggered.

Handle Exceptions Gracefully: Use error events and escalation events to model exception handling. Don't ignore potential error conditions in your process design.

Consider Timing: Use timer events for time-sensitive processes, deadlines, and timeout scenarios. This makes temporal aspects of your process explicit.

Plan for Interruptions: Use boundary events on activities to handle interruptions, cancellations, and exceptional situations that might occur during activity execution.

Data Modeling Best Practices

Show Key Data Flows: Include data objects for information that is critical to understanding the process, but don't model every piece of data to avoid cluttering the diagram.

Distinguish Data Types: Use different data object types (input, output, collection) to clarify how activities interact with information.

Connect Data Appropriately: Use data associations to show which activities read from or write to data objects, making data dependencies explicit.

Swimlane Organization Best Practices

Align with Organization: Structure pools and lanes to reflect your actual organizational structure or system architecture for maximum relevance and clarity.

Minimize Crossings: Arrange lanes to minimize the number of sequence flows that cross lane boundaries, improving diagram readability.

Group Related Activities: Place related activities in the same lane when they're performed by the same role or system, creating logical groupings.

Collaboration Modeling Best Practices

Use Pools for External Parties: Create separate pools for external organizations, customers, or systems that participate in the process but are outside your direct control.

Show Message Exchanges: Use message flows to explicitly show communication between different participants, making collaboration points clear.

Keep Pool Processes Simple: Don't overcomplicate the processes within external pools unless you have detailed knowledge of their internal operations.

Diagram Layout and Readability

Follow Reading Patterns: Arrange process flows to follow natural reading patterns (left-to-right, top-to-bottom) to improve comprehension.

Use White Space Effectively: Provide adequate spacing between elements to avoid cluttered diagrams that are difficult to read and understand.

Group Related Elements: Use visual grouping and alignment to show relationships between related process elements.

Consistent Sizing: Use consistent sizes for similar elements to create visual harmony and professional appearance.

Validation and Review Guidelines

Walkthrough Scenarios: Test your process model by walking through different scenarios, including normal flow, exception cases, and edge conditions.

Stakeholder Review: Have business stakeholders review and validate the process model to ensure it accurately represents their understanding of the business process.

Check Completeness: Ensure every path through the process leads to an appropriate end event and that there are no "hanging" elements without proper connections.

Verify Gateway Logic: Double-check that gateway conditions are mutually exclusive for XOR gateways and properly inclusive for OR gateways.

Tool-Specific Best Practices for bpmn-js

Leverage Validation: Use bpmn-js built-in validation features to check for modeling errors and inconsistencies in your diagrams.

Utilize Properties Panel: Take advantage of the properties panel to add detailed information, documentation, and technical configuration to process elements.

Export Considerations: When exporting diagrams, ensure that all stakeholders have access to appropriate viewing tools or export to formats that maintain visual fidelity.

Version Control: Implement proper version control for your BPMN models, especially when multiple people are collaborating on process design.

References {#references}

[1] Object Management Group. "Business Process Model and Notation (BPMN) Version 2.0." https://www.omg.org/bpmn/

[2] bpmn.io. "bpmn-js: BPMN 2.0 rendering toolkit and web modeler." https://bpmn.io/toolkit/bpmn-js/

[3] Camunda. "BPMN 2.0 Symbols - A complete guide with examples." https://camunda.com/bpmn/reference/

[4] Allweyer, T. (2016). "BPMN 2.0: introduction to the standard for business process modeling." Books on Demand.

[5] Creately. "The Complete List of BPMN Symbols and Their Meanings." https://creately.com/guides/bpmn-symbols/

[6] Visual Paradigm. "BPMN Activity Types Explained." https://www.visual-paradigm.com/guide/bpmn/bpmn-activity-types-explained/

[7] Lucidchart. "BPMN Diagram Symbols & Notation." https://www.lucidchart.com/pages/tutorial/bpmn-symbols-explained

[8] Gliffy. "BPMN Symbols & Notation Guide." https://www.gliffy.com/blog/guide-to-bpmn-symbols

[9] InRule Technology. "Common BPMN 2.0 Symbols." https://support.inrule.com/hc/en-us/articles/12660774454413-Common-BPMN-2-0-Symbols

[10] ProcessMaker. "Understanding BPMN Diagrams and Symbols."

https://www.processmaker.com/blog/bpmn-diagram-and-symbols/

[11] GBTEC. "BPMN 2.0 modeling, rules and symbols simply explained." https://www.gbtec.com/wiki/process-management/bpmn/

[12] Signavio. "Business Process Model & Notation - BPMN Introductory Guide." https://www.signavio.com/bpmn-introductory-guide/

[13] Bizagi. "A Quick Guide to BPMN: Business Process Model and Notation." https://www.bizagi.com/en/blog/guide-to-bpmn

[14] Visual Paradigm. "Types of Gateway in BPMN." https://www.visual-paradigm.com/guide/bpmn/bpmn-gateway-types/

[15] Lucidchart. "BPMN Gateway Types." https://www.lucidchart.com/pages/tutorial/bpmn-gateway-types

[16] HEFLO. "BPMN Gateways: Understanding Types and Best Practices." https://www.heflo.com/blog/bpmn-gateways

[17] draw.io. "Navigating BPMN Gateways: Making Sense of 69 Options." https://drawio-app.com/blog/navigating-bpmn-gateways-making-sense-of-69-options/

[18] Camunda Academy. "BPMN - Task Types." https://academy.camunda.com/bpmn-task-types

[19] UiPath Documentation. "Tasks in BPMN modeling - Maestro." https://docs.uipath.com/maestro/automation-cloud/latest/user-guide/tasks-in-bpmn-modeling

---

This reference sheet was created by Manus AI to provide comprehensive guidance for BPMN-js users. For the most current information, please refer to the official BPMN 2.0 specification and bpmn-js documentation.

Document Version: 1.0

Last Updated: August 18, 2025

Total Symbols Covered: 50+ BPMN elements

Compatibility: BPMN 2.0 Standard, bpmn-js library