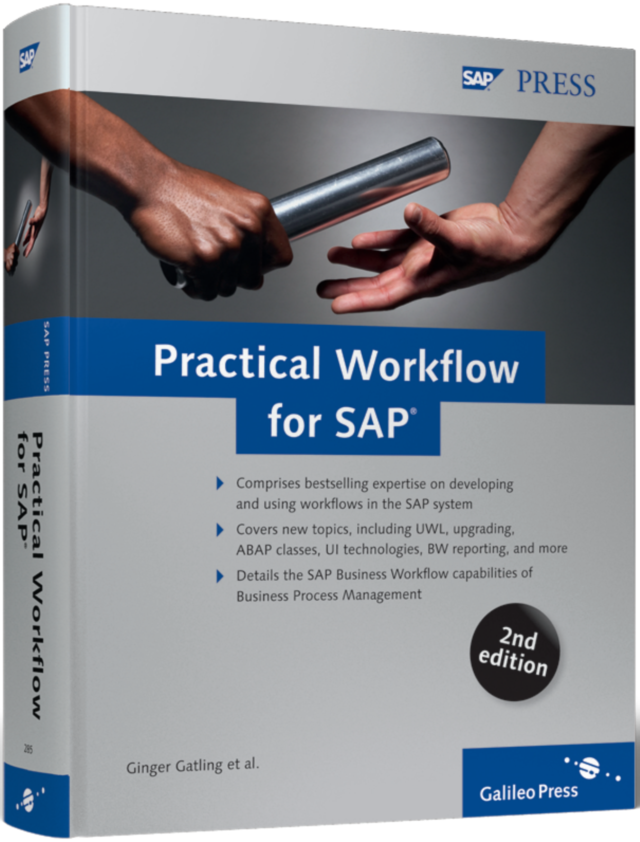
Jocelyn Dart, Erik Dick, Ginger Gatling, Oliver Hilss,
Somya Kapoor, Silvana Kempf, Susan Keohan, Thomas Kosog,
Paul Médaille, Mike Pokraka, Alan Rickayzen,
Shalini Sabnani, Jörn Sedlmayr, Ted Sohn

Practical Workflow for SAP®



- Comprises bestselling expertise on developing and using workflows in the SAP system
- Covers new topics, including UWL, upgrading, ABAP classes, UI technologies, BW reporting, and more
- Details the SAP Business Workflow capabilities of Business Process Management

Galileo Press®

Bonn • Boston

Contents at a Glance

PART I Using SAP Business Workflow

PART II Developing Your Own Workflow

PART III User Interface Technologies and SAP Business Workflow

PART IV Using SAP Business Workflow in SAP Business Suite Applications

Appendices

Contents

PART I: Using SAP Business Workflow

PART II: Developing Your Own Workflow

15 Service-Enabling Workflows 501

16 Advanced Diagnostics 535

PART III: User Interface Technologies and SAP Business Workflow

PART IV: Using SAP Business Workflow in SAP Business Suite Applications

Whether you are building a new workflow from scratch or using an SAP supplied workflow, it is important that you understand the Workflow Builder tool. This chapter gets you started by enabling you to create your first workflow, enhance your workflow, understand data flow, understand types of steps you can use when building a workflow, and get a workflow to production.

8 Creating a Workflow

The central tool for creating, displaying, and processing a workflow is the Workflow Builder. The Workflow Builder is accessed through Transaction SWDD. Alternatively, you can use the menu path TOOLS • BUSINESS WORKFLOW • DEVELOPMENT • DEFINITION TOOLS • WORKFLOW BUILDER • WORKFLOW BUILDER. Within the Workflow Builder, you can create all components of a workflow, including the process flow, the details for each step, and the data flow between steps.

This chapter describes the main features of the Workflow Builder, giving a good foundation to start your own development. To introduce the Workflow Builder, we will explain how to create a simple workflow. We will build upon the simple workflow, enhancing it to show additional features. However, this is not an exhaustive description of all workflow features. When you need more information, the SAP Help Portal (*http://help.sap.com*) describes all of the Workflow Builder features in detail.

Most workflows are started by an *event* (e.g., when a sales order is created, when a quote is entered, when an email arrives, when an error occurs, when a document is printed) that happens in a business application. You define which data from this event needs to be passed to the workflow via *binding*. Events are described in Chapter 13, Using Events and Other Business Interfaces, and Chapter 14, Custom Programs. You can also start any workflow directly, for example, through a transaction code, user interface, or a concept called generic object services (refer to Chapter 13 as well). Because events are a major topic on their own and to keep the focus on the basics of the Workflow Builder, this chapter starts the workflow directly using test tools.

The *workflow container* is used to hold all of the data needed by the workflow. Each workflow has a number of *steps* that execute activities or control the workflow. Data may be passed from one step to another. The activities are handled within *tasks*. You can use the same task in several steps of a workflow (or even in the steps of several different workflows) if you wish. A task has a *task container* that holds all of the data necessary for that task. As described previously, *binding* is the term used to pass data from the workflow container to the task container or from the task container back to the workflow container. Every step has one or more possible *outcomes* depending on the step type, the task, and what the step is doing. For example, for an approval step, possible outcomes might be APPROVE or REJECT.

Expressions are variables used in the workflow to control the workflow (e.g., branches) or to deliver a result (e.g., the agent ID for executing a step). Examples of expressions are simple container elements or the attributes of objects (objects are discussed in Chapter 10, Business Objects, and Chapter 11, ABAP Classes). *Basic data* controls global aspects of the workflow, such as constructor and destructor methods and defaults for the workflow steps. One part of this basic data is version-dependent; the other part applies to all versions. Lastly, the workflow will have one *end point.* There are no hidden exit points.

8.1 Workflow Builder Basics

This section helps you get familiar with the Workflow Builder tool.

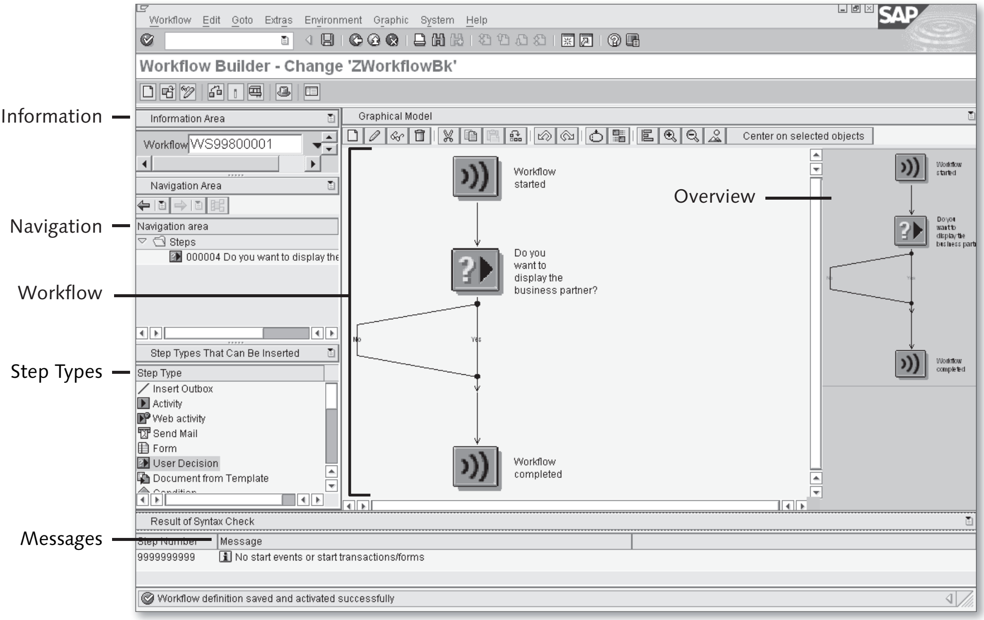
8.1.1 Look and Feel of the Workflow Builder

The Workflow Builder provides a graphical view of the workflow definition. The Workflow Builder screen is divided into the following frames (see Figure 8.1 as well); each of the frames can be resized:

- **Workflow**
  Here you can insert new steps into the workflow definition and process existing ones. Double-clicking on a step displays the associated step definition.
- **Overview**
  The overview graphic shows all steps in a workflow. The part of the workflow graphic displayed in the Workflow frame is marked with a green rectangle.

Changing the size or position of the rectangle changes the display in the Workflow frame.

- **Step types**
  Step Type is the default view when you enter a workflow. It lists all of the types of steps you can insert into your process. To insert a new step into the workflow, drag the desired step to the workflow panel, and drop it on the location where you want the step. When dragging in new step types, you will see a plus icon (+) in the appropriate locations to add steps. In Figure 8.1, you can also see a limited list of step options. By resizing the frame, you can see more step options as shown in Figure 8.2.

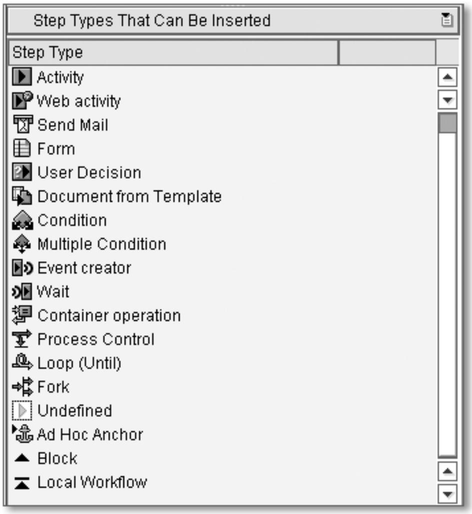


9999999999 No start events or start transactions/forms

Workflow definition saved and activated successfully

Figure 8.1 Major Elements of the Workflow Builder

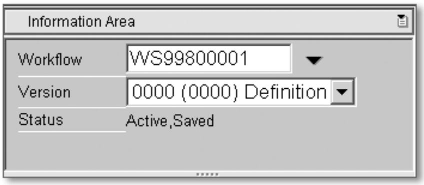
- **Information**
  The information frame (see Figure 8.3) displays which workflow is loaded, the status of the workflow, and the version number of the workflow in the original system. To load a different version, select the version. To load a different workflow, enter the workflow number in the format `WS<number>` and press ↵. If you do not know the workflow number, select the arrow and you can search for the workflow.



Step Types That Can Be Inserted

Step Type
Activity
Web activity
Send Mail
Form
User Decision
Document from Template
Condition
Multiple Condition
Event creator
Wait
Container operation
Process Control
Loop (Until)
Fork
Undefined
Ad Hoc Anchor
Block
Local Workflow

Figure 8.2 More Step Options



Information Area

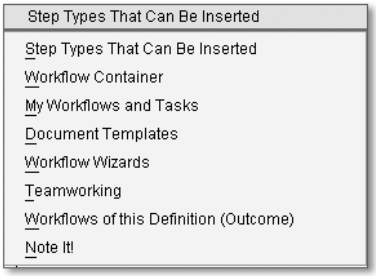
Workflow	WS99800001
Version	0000 (0000) Definition
Status	Active,Saved

Figure 8.3 Information Area

- **Navigation**
  The navigation frame contains a list of all of the steps in the workflow. You can jump directly to the relevant step definition from the list. As with all of the frames in the Workflow Builder, you can resize this frame to display the amount of information that you require. The step number corresponds to the number in the workflow technical log (logs are discussed in Chapter 6, Workflow Administration).

- **Messages**
  This area contains messages, including general messages and results from where-used lists, syntax checks, and searches.

- **Optional information to display**
  In addition to the frames you see in Figure 8.1, you can optionally switch the Step Types to one of the items in Figure 8.4.



Step Types That Can Be Inserted

- Step Types That Can Be Inserted
- Workflow Container
- My Workflows and Tasks
- Document Templates
- Workflow Wizards
- Teamworking
- Workflows of this Definition (Outcome)
- Note It!

Figure 8.4 Optional Information to Display

- **Workflow Container**
  The workflow container holds all elements required during the life of the workflow. Container elements will be created automatically, but you can also create your own workflow container elements.
- **My Workflows and Tasks**
  This frame enables you to quickly see all workflows you have edited. Additionally, you can search for tasks to add to the list. Your choice also determines what is displayed in the Workflow Explorer, which is a separate transaction (SWDM).
- **Document Templates**
  Document templates add digital documents to a workflow. All of the workflow's document templates that can be used in the step type DOCUMENT FROM TEMPLATE are displayed here. You can generate a where-used list to find the steps in which a document template is used.
- **Workflow Wizards**
  All Workflow wizards available for the definition of your workflow are displayed here.
- **Teamworking**
  Here you can search for steps by selected criteria such as who last edited the step in the definition or which steps are grouped together. The result is displayed graphically in the Workflow frame.
- **Workflows of this Definition (Outcome)**
  Your workflow outbox is displayed here, which displays all currently running workflows for this definition.
- **Note It!**
  You can create notes and documentation about the workflow in this space.

8.1.2 Building Your First Workflow

In this section, you will build a simple workflow, adding to it step by step. In the end, you will have a workflow with different types of workflow steps:

- The first user interaction step asks the user to make a decision: "Do you want to display the business partner?"
- If the answer is yes, the business partner is displayed.
- If the answer is no, an email is sent to the user, telling the user that he did not choose to display a business partner.

This is a simple scenario that should demonstrate how easy it is to build and execute your first workflow process. Eventually, we will link this workflow to a business partner creation in the application that triggers the workflow, so someone will create a business partner, the workflow will notify that a new business partner has been created, and give the user the option to review the business partner. In this chapter, we focus on creating the basic workflow.

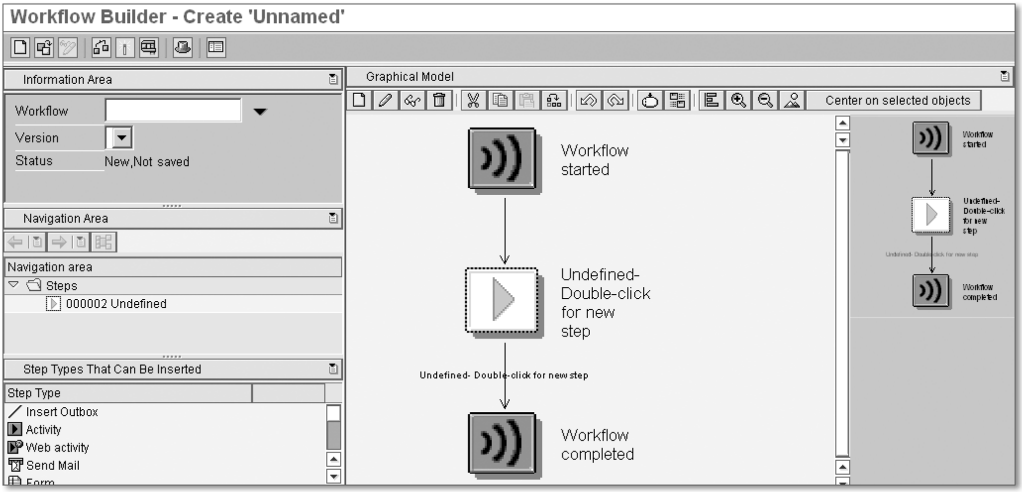
We start by creating the workflow and creating a decision step. To keep it simple, you will be the agent. If you have a test system, you may want to build this process yourself. By following a simple example that becomes more sophisticated as the chapter progresses, you will get a good idea of what workflow can achieve.

Starting the Workflow Builder

When the Workflow Builder is called for the first time or you opt to create a new workflow, a newly created initial workflow definition appears (see Figure 8.5).

This initial workflow has the following parts:

- The start of the workflow definition is indicated by Workflow started (◙).
- The end of the workflow definition is indicated by Workflow completed (◙).
- The area in which you insert the new workflow definition is indicated by an undefined step with one outcome (▷). Steps are represented by symbols. The name of the outcome is displayed on the arrow leading to the next step in the standard view.


Workflow Builder - Create 'Unnamed'
Information Area
Navigation Area
Step Types That Can Be Inserted
Graphical Model
Center on selected objects
Workflow started
Undefined- Double-click for new step
Undefined- Double-click for new step
Workflow completed

Workflow:
Version:
Status: New,Not saved

Navigation area
Steps
000002 Undefined

Step Type
Insert Outbox
Activity
Web activity
Send Mail
Form

Figure 8.5 Initial Workflow

Create and Save Your First Workflow

Now you are ready to create your first workflow:

1. If you haven't done so yet, navigate to the Workflow Builder by entering Transaction SWDD or using the menu path Tools • Business Workflow • Development • Definition Tools • Workflow Builder • Workflow Builder.
2. If you do not see an initial workflow similar to Figure 8.5, select Workflow • New or click on the Create New Workflow icon (□).
3. The workflow now has the title "Unnamed" and has a status, New, Not saved.
4. Save your workflow by providing an abbreviation and a name. This is discussed in more detail in Section 8.1.3, Saving, Activating, and Testing. For now, enter a name you will remember later, such as "zFirstWorkfl". You will also need to provide transport information. Assuming this work is being done on a sandbox, select Local object.
5. Congratulations, you have just created and saved your first workflow.

Your first step illustrates how a user decision works. *User decisions* have their own step type and symbol that looks like this: ▶. When a user decision executes, a question together with a predefined list of answers (the possible outcomes) is displayed to the recipients. User decisions are useful in the following situations:

- Only *one of several* possible alternatives should be processed in the workflow at execution time. An agent needs to make a business decision about which alternative is appropriate.
- An *instruction* (e.g., a user decision with only one outcome) is required to continue the workflow.
- For *approval*, *release*, or *status change* steps, the recipient needs to choose whether to approve or reject.

On the Decision tab, you can make all of the entries required to define an executable user decision. At runtime, the user will see the Decision title as the Subject text of the work item in his inbox.

Add a User Decision Step to Your Workflow

In the next step of our example, you create a workflow that request a decision about displaying a business partner. The decision will let the user respond "Yes" or "No."

1. In the Workflow Builder, locate the User Decision step type in the Steps That Can Be Inserted frame.
2. Drag the User decision icon (?▶) to the Undefined step. The step definition of the user decision is now displayed.
3. Enter the title "Do you want to display the business partner?"
4. Enter the decision texts "Yes" and "No." The outcome names default to the decision texts, but you can specify your own names if you prefer.
5. Select Expression in the Agents area of the screen. You will see a dropdown box. Set the agent to the expression to Workflow Initiator.

   The workflow initiator is always the person who started the workflow. There are many other options when determining an agent, as discussed in Chapter 5, Agents. However, to keep the first workflow simple, we will route all steps to the workflow initiator.
6. Complete the step by choosing the Transfer to graphic button (✓).

You have now defined a workflow with a user decision step. However, it must be saved and activated before you can use it.

8.1.3 Saving, Activating, and Testing

When you choose Save (💾) for a new workflow, you must enter an abbreviation and a name for the workflow. You can change both at any time in the basic data of the workflow. You also have to choose a package for transporting the workflow to other systems. If you are in your sandbox environment, then you can select Local Object, which denotes it will not be moved to any other system. The status in the title bar of the Workflow Builder is always visible. After you save your workflow, notice your workflow has a `WS` name and a number. Workflow templates are saved on the database with a `WS` and a number. The number range is from the settings you made in Chapter 3, Configuring the System.

To execute your workflow, activate it by choosing Activate ( ), which compiles the definition so that the SAP Business Workflow Engine can interpret the workflow for execution. Before activating, the workflow definition is subjected to a syntax check. If you only want to check the syntax of the workflow definition, you can choose Syntax Check ( ). All recognized problems are classified as errors or warnings and are output in the message area, together with other useful information. You can process the step in which the error occurs by clicking on the message.

The workflow will only be activated if no syntax errors are found. The status of the workflow is now Active, Saved. You can now test your workflow by choosing Test ( ).

Tip

When you choose Test, the workflow is automatically saved, checked, and activated if it is in the *inactive* state and you are in change mode of the Workflow Builder. There is no need to check and activate separately.

Test One

In this example, you will save, activate, and test your workflow:

1. Save your new workflow.
2. Activate the workflow.
3. Test the workflow by choosing Test ( ).
4. In the test environment, choose Execute ( ) to start the workflow.

5. Execute the user decision. Notice the text "Do you want to display the business partner?" and the choices available. Choose Yes or No.
6. Now return to the Workflow Builder (by using the Back arrow in the Test screen). Toggle the Steps That Can Be Inserted to the Workflows of this definition frame.
7. You can double-click on the new entry to see the matching workflow log. Notice the status of the workflow is Completed, and the result of the decision step is displayed in the Result column.

Test Two

In the previous test, you executed the workflow immediately, without navigating to the inbox. In this example, you will test the workflow again but also use the inbox:

1. Test the workflow again, selecting Test and then Execute as you did in the previous test. The workflow executes immediately.
2. When the decision appears, notice you have three options: Yes, No, and Cancel and keep work item in inbox.
3. Select the option Cancel and keep work item in inbox.
4. Select Business Workplace.
5. Select Inbox • Workflow. You will see your work item. Double-click the work item, and select either Yes or No.
6. Use the Back arrow to return from the inbox to the workflow definition.
7. Notice both times when you execute the workflow, there is only one step, the step to make a decision. Later in this chapter, you will add the step to display a business partner if the choice is Yes and to send an email if the choice is No.

If you tried this example in a test system, you may be surprised that when testing the workflow, you are presented with the decision straight away without having to look in your workflow inbox first. This is because the step is configured as part of the synchronous dialog chain by default (described in Chapter 4, Work Item Delivery). Because the person starting the workflow (you) is identical to the person assigned to perform the first step in the workflow (you), you are presented with the task straight away. To change this behavior, follow these steps:

1. Double-click on the User Decision step in your workflow.
2. Select the Details tab.
3. Deselect the checkbox Advance with dialog.
4. Return to your workflow, and Activate and Test it.
5. This time when you test, you receive a message at the bottom of your screen that says *Task started under work item ID ##### (current status: In Process)*. The work item number you receive is the process id. Your workflow has the status In Process and is in the inbox.
6. To execute the work item, select Business Workplace.

Congratulations on executing your very first workflow!

8.2 Enhancing Your Workflow

Now that you have created your first workflow, it is time to enhance it. This section covers many topics needed to create workflows. You will add a simple deadline to ensure work is performed on time according to process regulations. You will also add the step to display a business partner, which requires you to create new tasks and understand how objects are used in workflow, including how data is accessed and managed. You will also learn about ad-hoc activities for workflows and the use of review workflows so key or sensitive processes can be closely monitored.

8.2.1 Deadline Monitoring

A major advantage of workflow is the ability to monitor workflow steps according to a predefined schedule. This can be very useful if you want to monitor service level agreements or other process controls that ensure timeframes are enforced in the process. You can monitor a number of different date/time deadlines against each workflow step: requested start, latest start, requested end, and latest end:

- If a *requested start* deadline is active for a work item, then the work item only becomes visible to the recipients after the date/time specified. Background work items are started (executed) when the start deadline is reached.
- If a *latest start*, *requested end,* or *latest end* deadline is active, then the workflow reacts to the deadline when the specified date/time is reached.

The standard reaction of the workflow system is to send an escalation email. However, you can perform more complex escalation procedures by specifying a deadline outcome name. This lets you add steps to your workflow, which are executed after the deadline fails. This is called a *modeled deadline*.

You define deadlines with respect to a *reference date/time*. The system offers the following reference date/times:

- **The creation date/time of the work item**
  For example, assume a workflow has 10 steps. Step 6 must be executed within three hours of its start time. The 3-hour clock starts when Step 6 is initiated.
- **The creation date/time of the workflow to which the monitored work item belongs**
  In this example, assume Step 6 of the 10-step workflow must be completed within 2 days of the workflow starting. The clock for the deadline starts from the moment the workflow was initiated, not from when Step 6 was initiated.
- **A date in the form of an expression, which is derived from the context of the application during execution of the workflow**
  In this example, assume the step must be completed according to a specific business guideline. Perhaps you have 2 days for a priority B service complaint but only 1 day for a priority A service complaint. Another example would be within 3 days of an invoice posting date. The work item must read the invoice posting date and start the deadline based on the date.

To see the deadline options, double-click on the User Decision task in your workflow, and notice the tabs: Latest End, Requested Start, Latest Start, and Requested End. You can activate monitoring of the relevant deadline by selecting one of the deadline tabs, selecting the reference date/time for the deadline, and providing the time details. Activated deadlines are marked with a ringing bell icon (🔔) in the tab index.

If you choose Expression, you must define the reference date/time by specifying expressions for the date or time. Use the value help (F4) for entering expressions. In the example mentioned previously, of a deadline within three days of a posting date, you need to have the posting date in the workflow container. You then use Expression to select the posting date variable from the container, and select three days for the time. We will discuss more about how to get the posting date (and other fields) in the container in Section 8.3, Basics of Containers and Bindings.

Tip

The value referenced using the expression must be of data type `D` for the date and data type `T` for the time. If you specify a date but *no time*, the system sets the time to `00:00:01` (requested and latest start) or `23:59:59` (requested and latest end).

Specify the deadline by entering duration and an appropriate time unit (e.g., minutes, hours, days). Negative durations can only be used if you define the reference date/time via an expression.

When specifying the type of deadline, the date/time threshold, you can also specify who to notify and what text to send. The text is stored in the details of the task being monitored. For example, if the deadline is on a step to approve purchase requisitions, the task to approve the purchase requisitions holds the text that will be used in case of a deadline. Each task can have its own deadline text.

Tip

With the standard deadline reaction, the status of the monitored work item is unchanged. The work item still has to be executed by one of its recipients before the workflow can continue. If the monitored work item is to be aborted when the deadline is exceeded, you need to use the modeled deadline reaction. Refer to Chapter 9, Advanced Workflow Design Techniques, and to Appendix A, Tips and Tricks, for more details.

Add a Deadline to Your Process

In the following example, you add a deadline to your user decision step and test the deadline:

1. Return to your workflow definition in the Workflow Builder.
2. Double-click on the User Decision step. In your user decision step, choose the Latest End tab.
3. For the reference date and time, select Work Item Creation.
4. For the Time field, select Minutes and enter "2". This means the user will have two minutes from the moment the work item is created to complete the work item.
5. For the Recipient of message when latest end missed, select Expression and then select Workflow Initiator from the dropdown menu.

6. Test your changed workflow (remember, saving and activating is performed automatically when you choose the Test option from the Workflow Builder). This time, do not execute the decision step (cancel out of it if you have not removed the Advance with dialog checkbox).
7. Navigate to the Business Workplace.
8. Wait for the deadline to be exceeded, and you will receive a deadline message in the Business Workplace: The Deadline messages folder contains a message that the deadline was missed. The Overdue entries folder displays all work items that have an overdue deadline.

Tip

The background job for deadline monitoring must be scheduled so that the Workflow Engine can monitor and escalate deadlines. This job is explained in Chapter 3, Configuring the System. When the deadline job runs, all exceeded deadlines are escalated. If you are running this job periodically, then the actual time of escalation is delayed until the job next executes. Use Transaction SWWB to have the job run immediately.

8.2.2 How to Create and Use Tasks

In this section, you add more tasks to our workflow. You will learn how to create a task to display a business partner and how to create a task to send an email. Most steps in your workflow will be tied to business functions: updating a business partner, posting an invoice, approving a purchasing document, updating employee data, and so on. To execute business functions, you use the Activity steps (▶).

Activity steps are related to tasks, which start with TS. Workflows are created with WS and a number. The number range is determined from the settings you made in Chapter 3, Configuring the System. Those same number ranges are used for TS steps within your workflow.

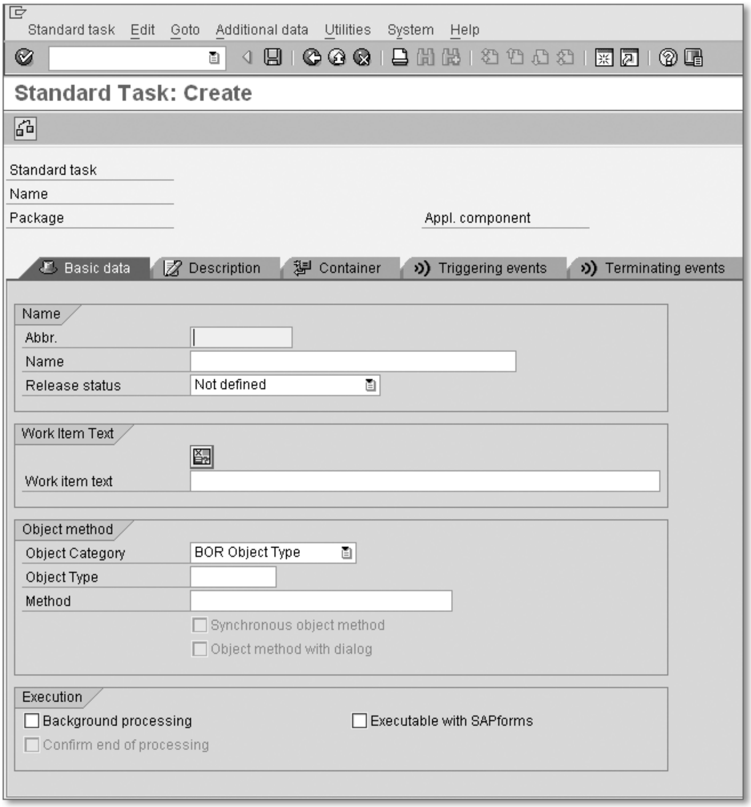
The user decision step you used earlier is based on a generic decision task (TS00008267) as standard. If you double-click on the user decision step in your workflow, select the Control tab, and you will see the task number. After a TS task is created, it can be reused in multiple workflows.

In this section, you create tasks from within the workflow. However, you can also create tasks independent of the Workflow Builder using Transaction PFTC.

Regardless of how you call the task definition, the same screen for editing the task definition is displayed.

Explanation of Standard Task (TS): Create Screen

Before creating a task to display a business partner, a discussion of the options available when creating a task is needed. Figure 8.6 shows the fields available when creating a new task.



Standard Task: Create

Standard task
Name
Package
Appl. component

Name
Abbr.
Name
Release status: Not defined

Work Item Text
Work item text

Object method
Object Category: BOR Object Type
Object Type
Method
☐ Synchronous object method
☐ Object method with dialog

Execution
☐ Background processing
☐ Executable with SAPforms
☐ Confirm end of processing

Figure 8.6 Create a New Standard Task (TS Task)

Every task must answer two major questions:

- *What* should the task do (*display* a business partner, *update* a business partner, *approve* employee leave)?
- *Who* can do the task?

As the options in Figure 8.6 are discussed, keep in mind that the task must always be able to address the "what" and "who" questions:

- Abbr. is the abbreviated name of the task. You use this name when searching for the task. You should have a naming convention for creating both TS tasks and WS tasks.

Tip

It is a good idea to decide on a naming convention for task abbreviations to make tasks easier to find, such as a specific prefix for all tasks in the project. Customers normally define their own naming conventions. Some customers may start all of their workflows with Z. Others may use the first two letters as functional area, then an underscore, and nine characters of text. An example might be HR_DisplPern for display personnel workflow that is part of the HR area, or ZHR_DispPern if you want to start all workflows with a Z to indicate they are custom built.

- Name is the full name of the task.
- Work Item Text is the description that will appear in the inbox at runtime. The work item text is very important because it is the first thing the user will see, and it should describe the task for the user. You can pass variables (such as business partner number, invoice amount, etc.) into the work item text to give the item more meaning. Keep in mind that users may have hundreds of work items in their inbox, so the text should be meaningful. During your design phase, you should work with the business users to determine brief but meaningful text to use.
- Object Category describes how you link this task to actual business data. The options available are Business Object Repository (BOR) objects and ABAP classes. BOR objects are discussed in detail in Chapter 10, Business Objects; ABAP Classes are discussed in Chapter 11.
- Object Type is where you enter the actual object name. The trick to this field is you need to know which object to use. Over time, you will become familiar with the objects provided by SAP, and you will become very familiar with the ones you create yourself. Common BOR objects include: BUS1006 (Business Partner), BUS2032 (Sales Order), and PERSDATA (Employee Personal Data).
- Method is the action you want to execute for the task. Examples of methods include create, display, update, block, remove block, approve, and release.

The combination of the Object and Method fields answers the "what" question. You need both an object type and method to know what the task can do. When you insert the method, the system takes the following from the definition of the object method, as applicable:

- Synchronous or asynchronous object method
- Object method with or without dialog

You cannot change these. If method parameters are defined for the object method, the system gives you the option of creating matching container elements automatically in the task container. The names of these container elements are then identical in the task container and the method container.

Tip

Methods that you execute with a task may be synchronous or asynchronous methods. Section 8.2.3, Using Asynchronous Tasks, explains the differences and their effect on modeling.

- To answer the "who" question, from Figure 8.6, follow the menu path Additional Data • Agent assignment • Maintain. You are assigning all of the agents who could ever possibly do this task. (Agents are discussed in detail in Chapter 5, Agents, and Chapter 12, Agent Determination Rules.) For example, if the task is displaying a business partner, who are all of the people who would ever need to display a business partner, or approve a purchase requisition, or enter an expense report. The "who" assigned here is who in the broadest sense of the term. In our examples, we will normally make the task a General task, which means everyone is a possible agent. To make a task a general task, select Attributes • General task.

In addition to the fields in Figure 8.6, also notice the following tabs:

- The Description tab enables you to add a longer task description. This *task description* appears in the users' inbox at runtime. The *work item text* is the one liner that appears in the inbox, and the task description is the long description the user will see after selecting the work item. The *task description* can also have variables to better describe to the user what the task is and what is required for the task.

  The Description tab also enables you to add texts for the deadlines. This is the text the user will see when a deadline has passed. For example, if a user has two days to update a business partner, after the deadline has passed, a note is sent to the manager that the deadline has passed. The note sent to the manager contains the text entered in the deadline task description. There is text for each type of deadline: Latest end text, Requested end text, and Latest start text. Additionally, there is also Completion text, which is used for notifications (discussed in Section 8.2.5, Notifications).

- The Container tab contains data in the task container. The task container holds all required runtime data. The container always contains what object is used and who is executing the task.
- The Triggering events and Terminating events tabs contain events that can be used to stop and start this specific task. This topic is a bit more advanced and is discussed in Chapter 13, Using Events and Other Business Interfaces.
- The Default rules tab is used when the task will execute outside of the workflow template (WS task).

  An example is a task that starts due to an inbound IDoc (Intermediate Document). Inbound IDocs normally execute a single TS task, not a full WS workflow. In that case, you must know who should get this task at runtime. For example, if an inbound sales order IDoc fails, you may want to route it to the sales area manager for the sales organization. You would use a rule to find the correct sales area manager as discussed in Chapter 12, Agent Determination Rules.

Creating a Task to Display the Business Partner

Now you will create the task to display the business partner. In this example, you add a step to display a business partner if the result from the user decision step is Yes.

1. Return to your workflow in the Workflow Builder.
2. Drag an Activity step (▶) to the Yes branch of your user decision.
3. Select the Create task option from the button list on the button next to Task as shown in Figure 8.7.
4. Enter appropriate texts for the abbreviation and name. For example:
   - Abbr: zbp_display
   - Name: Display business partner
5. Enter the following for the business object fields:

- Object Category: BOR Object type
- Object Type: `BUS1006`
- Method: Display

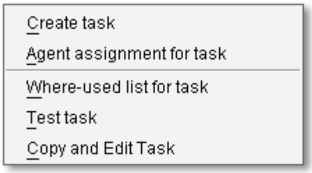


Figure 8.7 Dropdown to Create a Task from the Workflow Builder

> **Tip**
>
> When entering the method, you can see a list of all methods for the object by selecting the dropdown.

6. If you see a message *Transfer missing elements from the object method?*, answer Yes (the system is helping you set up the task container; more on this later in Section 8.3, Basics of Containers and Bindings).
7. Provide a work item text. Remember, this is the text the user will see at runtime. To give the text more meaning, we will pass in variables from the business object. Enter the text “Review business partner”.
8. Select the Insert variables icon (▦). Select `BUS1006` • Business Partner Number. Notice your work item text now has the following variable: `&_WI_OBJECT_ID.BUSINESSPARTNER&`. Place your cursor at the end of the text, and select the description for the business partner by inserting the variable `BUS1006` • Description.
9. Your work item text should now read *Review business partner &_WI_OBJECT_ID.BUSINESSPARTNER& &_WI_OBJECT_ID.DESCRIPTION&.*
10. Save (💾) the task. (You need to select Local object or a development class.)
11. Set up the security for this task by selecting Additional data • Agent assignment • Maintain. Normally you would assign the appropriate security role or organizational objects here, but for simplicity, make this a general task by selecting Attributes • General task.
12. Return to the workflow. The workflow suggests a binding between the workflow and the task. Accept the binding by choosing Enter (✓). Set the agent of

the activity step to the workflow initiator expression using the dropdown help.

13. When you return to your workflow, you should see your new task as part of the YES branch from the user decision task. Now when the workflow executes, you can display a business partner.
14. For testing purposes, you need to provide this business partner number when the workflow starts. To do this, toggle the STEPS THAT CAN BE INSERTED frame to WORKFLOW CONTAINER, and double-click on the workflow container element `BUS1006`.
15. Select the PROPERTIES tab, and select the IMPORT parameter setting. This means the business partner can be imported when the workflow starts. Normally, this is set so that the application can pass the business partner to the workflow container. If the flag is not set, then the business partner is solely contained in the context of the workflow.
16. Test your workflow. To do so, you must enter an object instance of your business partner object type. Select `BUS1006`. At the bottom of the screen, you see the OBJECT TYPE and a KEY. Select the dropdown in the KEY field, and enter a valid business partner number. (If you do not have a business partner and are on a sandbox system, you can use Transaction BP to create a business partner. When creating a business partner, it is easiest if you create a person and just provide first name and last name.)
17. After you enter a valid business partner number, select SAVE in the TEST DATA area of the screen. This enables you to select LOAD in subsequent tests to use the same business partner number, instead of entering the business partner number for each test.
18. After you start your workflow, you should execute two tests. In the first test, select YES to the user decision. When you select YES, a business partner should display. In the other test, select NO to the user decision, and the workflow should end.
19. Remember, you can see each execution in the WORKFLOWS OF THIS DEFINITION (OUTCOME) frame.

Hint

When testing the workflow, the first step may or may not start automatically depending on the setting of the ADVANCE WITH DIALOG flag in the DETAILS tab of the task.

Tip

The basic data is used to identify a task. Each task is identified by the object identifier `TS` (for standard task) and an eight-digit task number automatically allocated by the system during creation.

You must specify a package when saving the task in the same way that you specified a package when saving the workflow definition. The transport object in this case is `PDTS`. If you use Local Object, this means the workflow cannot be transported to another system.

Tip

If you also assign your tasks and workflows to a task group, you can use the task group as a logical package for the complete scenario. This is not only useful for documenting which workflows are the active valid workflows (as opposed to test workflows or scrapped versions) but also makes navigation in the Workflow Builder and Workflow Explorer easier.

Remember the importance of the work item text for your task. The work item text is displayed in the inbox, in the work item selection reports, and in the workflow log. If a work item text is not specified, the name of the task is used instead. You can use expressions within the work item text. They are replaced at execution time with the current values from the task container.

Note that container elements used in the work item text must be filled using a binding. To use expressions in the work item text, position the cursor at the relevant point in the text, and choose Expression ().

Tip

If you are looking at an SAP task or a task imported from another system in display mode, you can redefine the task description (which appears in the inbox as the work item text). Choose the Refind Work Item Text icon (), and enter a new text to replace the original text. Using the Redefine option overlays your version over the original task description as an extension rather than as a modification.

8.2.3 Using Asynchronous Tasks

A full description of the reasons for using asynchronous methods and the consequences is given in Chapter 10, Business Objects, but for the time being, just think of them as methods that are not completed until the application itself confirms completion by sending an event to the workflow. For an example, think of a step that deletes something: An employee requests vacation and then later decides he does not want the vacation. The original request is deleted.

Deleting is a natively asynchronous activity. You do not want the task completed until the deletion is completed, so a delete task is normally an asynchronous task. From the Workflow Builder point of view, an asynchronous task requires the terminating events to be activated as outcomes of your step. When modeling an activity step, you can bind the results of a synchronous method to a workflow when the method is completed. However, the asynchronous method cannot return information to the workflow via the bindings from method to task, task to workflow. Instead, you define at least one terminating event. You can bind data from the terminating event to the task.

The *terminating events* of a task express the possible end statuses of the task. A terminating event always refers to the specific object for which the event must be triggered. For example, if the work item deletes document 123456, then the work item should only be completed when the event "document 123456 is deleted" occurs and not when any other document is deleted.

When the work item is created, the Workflow Engine automatically registers that this specific work item is waiting for a terminating event for the specific object involved. This is called an *event instance linkage* (refer to Chapter 13, Using Events and Other Business Interfaces, for more details). You define the object for which the event must be triggered in the terminating event definition using a container element, referenced to the event's object type, in the task container.

Example

For example, the event `DELETED` of object type `BUS2032` (Sales Order) is defined as a terminating event for a task. At runtime, the sales order `123456` is passed via the task container. The work item is now only terminated if the sales order `123456` is deleted and the event `BUS2032.DELETED` is triggered for that sales order. If another sales order is deleted, the work item is not terminated.

If a work item is terminated by an event, the execution of the workflow is continued by a matching outcome. You must activate this outcome, or the terminating event will be ignored. The method does not necessarily have to have been executed before the event occurs. If a work item is aborted, or undefined processing statuses arise, the work item is not terminated because no event is created. Only when the terminating event is received will the work item finish and the workflow continue.

Within a TS task, you maintain the terminating events on the Terminating events tab. You specify a container element of the task container that at runtime will contain a reference to the relevant object. This is generally the task container element _WI_Object_ID. The fields Object type category and Object type are filled automatically.

You identify the event by specifying its event name. The event must be defined for this object type. The workflow system creates the event instance linkage required at runtime and activates it. To check the properties of the terminating event, choose the Settings icon (▣). The properties of the instance linkage are displayed and can be changed.

8.2.4 How to Access Data and Activities

To access data, activities, and events within your workflows, they need to be defined as parts of an object. The object could be part of the Business Object Repository (BOR) (discussed in Chapter 10, Business Objects) or an ABAP class (discussed in Chapter 11, ABAP Classes). Objects can be used in many different workflows, tasks, and rules. SAP components contain many predefined business object types and ABAP classes. These predefined data, activities, or events can be used as is, or you can create your own:

- Objects describe, for a particular business entity, the data (attributes), functions (methods), and events used in a workflow. In our example, we used business object type BUS1006, representing a business partner.
- Data relating to a business entity needs to be defined as an *attribute* of a business object type before it can be used in a workflow. Attributes are defined as part of the object type to which they are most closely related. For example, the attributes Material name and Material number are defined within the Material object type, but Order number and Order value are defined as part of the Order object type. In other words, the attributes describe data related to the business object.
- Activities to be performed on or by using a business entity are defined as *methods* of a business object type before they can be used in a workflow. Every

object has methods that define activities that can be executed on that object or its data (e.g., "create business partner" or "update business partner"). Within the method, you can call SAP functions, your own functions, or other external applications.

- Events are another important component of an object. The events describe the status changes that an object can undergo (e.g., "business partner deleted" or "business partner changed"). A workflow can be started, canceled, or resumed (wait step) when an event of this kind is raised. Just like attributes and methods, events need to be defined as part of a business object before they can be used in a workflow. Events are discussed in Chapter 13, Using Events and Other Business Interfaces.

If you want to use workflow to implement a business process, this is a rough picture of what is involved in accessing the relevant data and functions:

1. Identify all business entities involved in your business process. You sort out which business functions and events you want to map in your scenario and which data you want to access.
2. Check whether the relevant business object types with their methods, attributes, and events are defined in the BOR or as an ABAP class. The grouping of object types in the application component hierarchy, and the option of searching generically for parts of a name, help when looking for existing object types:
    - If you find an object type whose definition meets your requirements, you can use it without making any modifications.
    - If you find an object type whose definition does not quite meet your requirements, you can extend its definition.
    - If you do not find a suitable object type, you can define your own object type.
3. Use the methods, attributes, and events of the object type in the relevant parts of your workflow.

Further information on how to create your own object types or extend existing object types can be found in Chapter 10, Business Objects, and Chapter 11, ABAP Classes.

Adding an Attribute of a Business Partner Object to a User Decision Step

In this example, you add an attribute of the business partner object BUS1006 to your user decision step:

1. Return to your workflow, and double-click on the user decision step.
2. Add a variable in your user decision title, for example, *Do you want to display business partner &1?*, and set PARAMETER 1 to the description of your business partner (&BUS1006.Description&) using the input help.
3. TEST your workflow. If you saved the data from a previous test, select LOAD. Otherwise, you must enter an object key of your business object type before executing the workflow test. You can reuse your test data for each execution by selecting SAVE in the TEST DATA area after you select a business partner from the dropdown. On subsequent tests, select LOAD from the TEST DATA area to load your business partner number.

Now that you have tested the workflow a couple of times, you know that after you start the test, you can go to the Business Workplace to see the work item. This time, notice the parameters in the work item text for the user decision task.

8.2.5 Notifications

You have the option to notify someone when a step is completed. It is a simple notification that sends a text note (not a work item) to a specified user when a step completes. Normally it is used to inform someone when a critical step has completed.

Adding a Notification to the Business Partner Display Step

Follow these steps to add a notification so someone will be notified when the business partner display step is completed:

1. Return to your workflow in the Workflow Builder.
2. Double-click on your step to display the business partner (in the YES branch of the user decision step).
3. Select the NOTIFICATION tab. Notice in this tab you update *who* to send the notification to and *what* the notification text should say.
4. Currently there is no specific NOTIFICATION (COMPLETION) text. Double-click on the link to add a notification text.

5. Ensure you are in change mode for the task (using the DISPLAY – CHANGE icon to toggle between change and display mode).
6. Select the DESCRIPTION tab. For the TEXT TYPE, select COMPLETION TEXT. Select the CHANGE TEXT icon so you can update the text.
7. After you have added a COMPLETION TEXT, save your task, and use the BACK arrow to return to the workflow.
8. Update the MESSAGE RECIPIENT FOR COMPLETION to be the WORKFLOW INITIATOR.
9. TEST the workflow again. Be sure to follow the path to display the business partner. After the workflow completes, go to the Business Workplace. You will see the notification text in the DOCUMENTS folder of the inbox.

8.3 Basics of Containers and Bindings

Containers and bindings are a bit tricky when first learning workflow, but as you understand the stability, flexibility, and scalability they provide, you will soon come to appreciate the powerful use of binding between containers. Here are a few of the advantages:

- You can reuse elements in your workflow.
- You can make major changes to activities within the process without jeopardizing the process as a whole (or vice versa).
- Even when the applications that trigger the workflows are changed from release to release, your workflow is sheltered from these changes.
- You can use parallel activities within the workflow without worrying about data reconciliation problems or interference between the activities.

Containers and bindings are explained in more detail in Chapter 9, Advanced Workflow Design Techniques. This section provides an introduction to how they are used by first focusing on the task container and then focusing on the workflow container.

All of the data needed to execute the method or to display in the task text must be available in the task container. Container elements for the task container are generated automatically when you enter a method in the task. The container elements needed for the execution are recognized by the workflow system, and the workflow system prompts you to automatically insert these container elements in the task container. In addition to what is automatically provided in the containers, you may want to create your own container elements in the task container and define a binding between the task and the workflow so that these container elements are filled at runtime.

8.3.1 How to Create Containers and Bindings for Tasks

The task container is edited on the CONTAINER tab page (refer to Figure 8.6 as well). To enable the method to process the data, you may (optionally) define a binding from the task container to the method container. The task object itself is automatically bound to the method container. For other method parameters, the system makes a proposal for the binding that can be reviewed on the BASIC DATA tab of the task by selecting the BINDING OBJECT METHOD icon.

However, not defining any binding between task and method is simpler and offers a performance gain, provided the container element names are the same in both containers. In this case, the contents of the task container are matched by element name and automatically copied to the method container (for all elements defined in the method). The same applies to the reverse binding.

Variables used in your work item texts and descriptions are also bound from the task container. In the "Creating a Task to Display the Business Partner" example in Section 8.2.2, How to Create and Use Tasks, you added a task to your workflow and bound the business partner number and description to the work item text. As a reminder, to add variables while editing your description, choose INSERT EXPRESSION to choose a variable from the task container. You can add as many variables to the text as you want (up to the limit of the text field).

8.3.2 Creating Container Elements in the Workflow Container

The work item text of the user decision can display current runtime values from the workflow. You can integrate these values by including variables relevant to the decision directly in the work item text. The variables are replaced at runtime with values from the matching workflow container elements.

Of course, this is just one example of how container elements are used in the workflow, but it is very easy for you to try yourself. (Using the workflow container to link variables for the user decision work item text was done in an example provided in Section 8.2.4, How to Access Data and Activities.)

1. Create container elements by selecting the Workflow container frame and double-clicking on the <Double-click to create> line in the workflow container tray.
2. Enter the technical name of the container element in the Element field.
3. Give each container element a technical name (minimum of two characters) that can be used to identify it uniquely. The technical name is not case sensitive and must begin with a letter, but it can be followed by letters, underscores, or digits. Because the technical name is not translated, it is conventional to use English words in multilanguage environments.
4. Under Texts, maintain the Name and the Description (optional). Both of these can be translated in multilanguage environments.
5. According to the data type reference of the container element, make the following entries on the Data type tab: First check whether your container element is modeled on one of the predefined types. Choose the Selection of Pre-defined Types icon (), and double-click to choose the predefined type. The system carries out the necessary entries for the data type. If you want to create a container element that is not predefined, make the following entries, depending on the data type:
   - **Object type**
     Choose Object type, select an object type category, and enter the name of the object type. Examples include a specific BOR object (such as `BUS1006`) or a specific ABAP class.
   - **ABAP Dictionary Reference**
     Choose Structure and Field. In this case, you enter a table/structure and field that the container data should be based on. This reserves space in the container equivalent to the field you enter. It works as a "like" statement.
   - **ABAP Dictionary Data Type**
     Choose ABAP Dictionary Reference, and enter the table or structure in the field Type name. Use this to provide a data type to describe the field in the workflow container.

Tip

The specification of an object type is not mandatory. If no object type is specified, the container element can be assigned a reference to any object type at runtime. However, binding restrictions may limit its use later in the workflow.

Tip

A common misconception of workflow is that only one business object can be used per workflow. This is *not* the case. Often cross-application workflows use several different business objects, and the flow itself forms the link between them. A simple example of such a scenario is the link between a scanned document (e.g., the object type `IMAGE`) and the invoice record that is posted to the database (e.g., the object type `BUS2081`).

On the PROPERTIES tab, select whether the new element is to be an IMPORT and/or an EXPORT element. Mark an import element as MANDATORY if applicable. Import means this field will be passed from the application to the workflow. For example, a document is created, triggering an event that starts a workflow. For workflow to receive the document information from the application, the receiving element in the workflow container must be marked as IMPORT.

8.3.3 Changing Container Elements

For any workflow container element you add, it is your responsibility to bind data to the workflow element; otherwise, the element will be empty. There are several ways to get data into your custom workflow container elements:

- **By initial values**
  You can assign an initial constant value to a container element. When the workflow is executed, the container element is initially filled with this value. Any changes made to the contents of the container element will overwrite this value.
- **By a container operation step**
  A container operation step lets you fill a container element with a constant or another container element.
- **By bindings in a workflow step**
  From any workflow step that can output data to the workflow (such as activity steps, user decision steps, document from template steps, etc.), you can transfer data from the task container of the workflow step to the workflow container (or vice versa) via container bindings. Think of bindings as the rules for parameter passing within your workflow.
- **By bindings from an event**
  Whenever your workflow responds to an event — for example when it is started by a triggering event — data can be passed from the event container to the workflow container. If you want to pass data from a triggering event to

start your workflow, the workflow container elements to be filled from the event container need the IMPORT flag turned on before the bindings can be defined. Refer to Chapter 13, Using Events and Other Business Interfaces, for more details.

Adding a Custom Workflow Container Element and Binding Data to the Element

In this example, you experiment with adding workflow container elements and manipulating them. This example shows how to add and use your own container elements. In this example, you add a container element to represent a date, add another date to this date, and use the new date in the user decision. (Normally you do not add two dates together, so this is not something you would probably use in production, but it will get you familiar with working with container elements.)

1. Toggle to the WORKFLOW CONTAINER frame, and create a new container element by double-clicking on the <DOUBLE-CLICK TO CREATE> line. Provide the following information:
   - ELEMENT: NewDate
   - NAME: MyNewDate
   - SHORT DESCRIPTION: My first try with containers
2. Select the ABAP DICTIONARY REFERENCE:
   - STRUCTURE: SYST
   - FIELD: DATLO

   You can now see the new field in your workflow container.
3. Switch from the WORKFLOW CONTAINER frame to the STEPS THAT CAN BE INSERTED FRAME. Drag the step type CONTAINER OPERATION to the line before the USER DECISION step, and enter the following information:
   - STEP NAME: AddDates
   - OUTCOME NAME: two dates added
   - RESULT ELEMENT: Use the dropdown to select NEWDATE.
   - EXPRESSION: Use the dropdown to see all of the options in the workflow container. Select the object `BUS1006`, and select CREATEDON.
   - OPERATOR: Use the dropdown to select ADD.
   - EXPRESSION: Use the dropdown to select SYSTEM FIELDS. Then select `TIMLO`.

4. In your Workflow Builder, you now have an AddDates step before the user decision step. You will display the result of the container operation step in the user decision step.
5. Double-click on the user decision step. Set Parameter 2 to your NewDate container element. Use the dropdown help to do this.
6. Use the variable in your work item text by writing &1 in the text where you want the value to appear, for example, *Do you want to display the business partner &1 &2?*

Tip

The &2 declares this to be a reference to parameter 2 of this step definition. It is optional — you can just use &. However, in a multilingual environment, specifying the number is very useful because the variables often appear in a different order in the translation.

7. Save, Activate, and Test your workflow. Notice the date you see in the work item text. It should be a date far in the future. (Keep in mind the only point for this example was to show how to add a workflow container element and use the container operation step type.)

A Word About the Task Description on the User Decision Step

Although the short text in the generic decision is part of the step definition (to make things simpler), the long text is part of the task. To create your own long text, you can copy task TS00008267 to a new task and write a suitable task description for this new task. You may select your own variables and add these to the task container. After you have created your task, substitute it into the step's Control tab in place of task TS00008267. Do not forget that you need to assign possible agents to your new task.

8.4 Steps

As well as the step types User decision, Container operation, and Activity shown earlier, there are other step types available for modeling a workflow. Although Activity is the main step type to link the workflow to the application, there are many other step types needed to control the workflow process.

8.4.1 What Other Step Types Exist

Table 8.1 shows all step types available in SAP NetWeaver 7.0. The step types cover all of the functions you need to control the workflow process, from what business functions to call, to looping, conditions, container manipulation, using multiple branches, and many other functions.

Step Type	Icon	Runtime Function
Activity	▶	Execution of a task or subworkflow. At runtime, data is passed from the task or subworkflow to the workflow container on creation of the matching work item, and vice versa on work item completion.
Ad hoc anchor	⚓	In the definition, you specify workflows that can replace this step. At runtime, an authorized user can select one of the specified workflows. The steps of this workflow then dynamically replace the Ad hoc anchor.
Condition	◆	Depending on the result of the Condition, either the true or the false path is followed. In the condition editor, you can simulate the results of the condition to make the testing of complex conditions easier.
Container operation	⇄	The Container operation is used to perform arithmetic operations or value assignments to workflow container elements using constants and data in the workflow container. This includes operations on multiline container elements, for example, appending to a list.
Document from template	📄	A digital document is created from a document template using variables in the text that are filled during workflow execution using the workflow container elements. The workflow container receives a new container element that contains the document ID.
Event creator	▶)	An event is raised. You fill the event container from the workflow container.
Fork	⇉	A Fork is used for parallel processing. You can define how many parallel branches exist and how many branches must be completed for the fork to terminate and the workflow to continue. Alternatively, simply define an end condition.

Table 8.1 Step Types

Step Type	Icon	Runtime Function
Form	▤	A structure-based container element can be displayed, processed, or approved as a Form. The data is transferred directly from the workflow container and back again.
Loop (UNTIL)	↻	A sequence of steps is processed at least once and then repeatedly until the defined termination condition occurs.
Multiple condition	◆	Based on the value of a workflow container element, one of several branches defined in the workflow definition is processed. Any value not specifically assigned to a branch can be processed in an Other values branch.
Process control	⇥	This can be used to cancel the execution of a work item or workflow or set a work item to obsolete, so that alternative steps can be taken in the Processing obsolete branch.
Send mail	✉	The text entered in this step type is sent as an email. The task required and the necessary bindings are automatically created by the workflow system.
Block	▲	A Block is a modeling construct that enables you to model a group of steps together. The block has a data interface. Additionally, you can add deadlines to a block, ensuring the entire block must be completed in a certain timeframe. Blocks are discussed in more detail in Chapter 9, Advanced Workflow Design Techniques.
Local Workflow	⤒	A Local Workflow is a “free-floating” block that is not connected to the main workflow. Local workflows are triggered by events and enable a design element to be incorporated into the workflow. They may need to execute multiple times during a workflow execution, or may not be executed at all during a workflow execution. Local workflows are discussed in more detail in Chapter 9, Advanced Workflow Design Techniques.
Undefined step	⬚	An Undefined step can be used as a placeholder during development. These steps are ignored at runtime.
User decision	▶	The agent is asked a question and given a predefined list of answers. Each predefined answer is a separate branch in the workflow.

Table 8.1 Step Types (cont.)

Step Type	Icon	Runtime Function
Wait for event	⚙▶	The system waits for a specific event. The work item is only completed if the expected event occurs. Data from the event container can be sent to the workflow container using a binding.
Web activity	▶⚙	The selected container elements are posted using HTTP in an XML or SOAP message. This step can also wait for a message reply.

Table 8.1 Step Types (cont.)

8.4.2 How to Insert New Steps

When inserting new steps, you drag and drop the step type to the location where you want the step. You can insert steps before or after an existing step. Table 8.2 provides an overview of how to insert steps into a workflow.

Where Do You Want to Insert the Step?	What Do You Have to Select?
After a step	Drag and drop on the outcome of the preceding step.
Before a step	Drag and drop on a step to insert before the step.
As a new branch of a fork	Drag and drop on the Fork symbol (⇉) at the start of the fork.

Table 8.2 Insert Steps Into a Workflow

The My workflows and tasks frame provides an efficient way of inserting tasks as activities in your workflow. My workflows and tasks displays tasks and workflows that you have selected or previously edited. The selection is made using a search area that provides diverse selection criteria. If you frequently need a group of tasks to define your workflows, you can put these tasks together in a task group and insert the group into your search area.

Display the contents of the task group in the tray, select the position in your workflow where you want to insert the task, and choose the task by double-clicking on it. An activity step is then automatically created in your workflow that refers to this task.

Insert a Send Mail Step

In this example, you insert a send mail step if the user decides not to display the material:

1. Return to your workflow, and drag the Send Mail step to the No branch of the User Decision step.
2. Enter the following information:
   - Send express: Select the checkbox
   - Subject:
     Part 1: You chose not to display business partner.
     Part 2: Select the Insert expression icon, and select `BUS1006` • Business Partner.
3. In the large text box, enter text for the email.
4. Select the green checkmark (Transfer and to graphic). You are asked for an abbreviation and name, which creates a new task. Enter appropriate values for the name and the description. You also need to provide a development class or have the task be a Local object.

At this point, your process should look like Figure 8.8. The user decision step is followed by two steps: review of the business partner and send email.

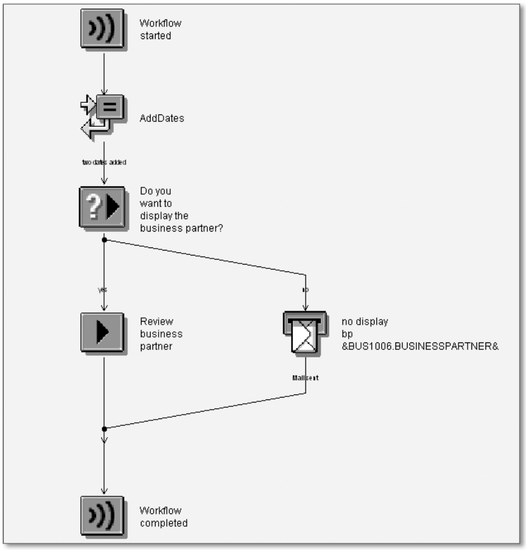


Figure 8.8 Example Process Built in This Section

8.4.3 What Kinds of Outcomes Exist

Outcomes are what the calling step/function can return. Certain outcomes appear in the workflow modeler by default. Outcomes are important because the workflow process branches are based on outcomes. There are different outcomes available according to the step type chosen. To see the possible outcomes of a step, follow these steps:

1. Double-click on a step, and select the OUTCOMES tab.
2. In the workflow from Figure 8.8, if you double-click on the USER DECISION step (*Do you want to display the business partner?*) and select the OUTCOMES tab, you will see three outcomes: YES, NO, and PROCESSING OBSOLETE.
3. If you double-click on the ACTIVITY step (Review business partner) and select the OUTCOMES tab, you see two outcomes: STEP EXECUTED and PROCESSING OBSOLETE.

Notice in the USER DECISION step, the outcome PROCESSING OBSOLETE does not appear in the workflow modeler. Normally, the only outcomes that are displayed are ones that require the workflow to react. Some outcomes are optional, and others are only displayed by the system if they are necessary as a result of specific settings. Table 8.3 shows all possible outcomes.

The Outcome Is ...	The Outcome Exists If ...	Notes and Comments
Event name (terminating event of task)	The task was defined with terminating events.	If the underlying method is an asynchronous method, you must activate at least one event as an outcome.
Value name (possible value of method result)	The synchronous object method is defined with a result for which fixed values are maintained in the ABAP Dictionary.	If you deactivate all values of the results, the system activates the STEP EXECUTED outcome instead.
Exception name (method exception)	The object method is defined with exceptions.	Refer to Chapter 10, Business Objects.
System outcome: DOCUMENT COULD NOT BE CREATED	The step is a document from template step.	This outcome is triggered if document creation fails.

Table 8.3 Step Outcomes

The Outcome Is ...	The Outcome Exists If ...	Notes and Comments
System outcome: TASK EXECUTED SYNCHRONOUSLY	The step is a document from template step.	Normal completion of a document from template step.
System outcome: STEP EXECUTED	▸ The activity refers to a synchronous object method without result. ▸ The activity refers to a synchronous object method with result, but no result is selected.	Normal completion of a step.
System outcome: PROCESSING REJECTED	The indicator PROCESSING CAN BE REJECTED is set.	If processing of the relevant work item is rejected at runtime (e.g., using REJECT EXECUTION in the Business Workplace), the steps defined after this outcome are executed.
System outcome: PROCESSING OBSOLETE	The work item can be set to obsolete using a PROCESS CONTROL step.	▸ The steps defined after this outcome are executed. ▸ This outcome is used to skip steps when modeled deadlines are missed.
System outcome: REQUESTED END, LATEST END, LATEST START	The relevant deadline monitoring is activated and a modeled reaction required. This applies to the workflow wizard MODEL DEADLINE MONITORING.	▸ Within these branches, you model steps to be executed when the deadline is missed. For example, you can model a PROCESS CONTROL step that sets the work item of this step to obsolete. ▸ You cannot deactivate these outcomes.

Table 8.3 Step Outcomes (cont.)

8.4.4 Which Task and Step Attributes Impact Work Item Execution

When setting up a task, there are certain options that impact how the task behaves at runtime. Attributes that influence the execution of work items can be found in both the task definition (when creating the `TS` task) and the step definition (when inserting the task into a workflow). The following settings reside in the task definition (refer to Figure 8.6):

- **Background processing**
  Set this checkbox if you want the workflow system (i.e., user `WF-BATCH`) to execute the work item automatically in the background without user involvement. This flag is only available if the underlying method is non-dialog, meaning that it does not require user involvement. The work item will not appear in any inbox, but you can view it via the work item reports or workflow logs.
- **Confirm end of processing**
  Set this checkbox if you want the user to decide when the work item is complete. As long as this confirmation has not taken place, the relevant work item remains in the inbox of the agent even if the work item has already been executed. The agent can execute the work item again or forward it. You *cannot* assign this indicator for tasks that are to be executed in the background.

Tip

CONFIRM END OF PROCESSING forces the user to indicate that he is done with the task, in addition to completing the work for the task. Only use this flag if you want additional confirmation before completing the work item. This setting also enables the user to add an attachment with the confirmation completion.

The following settings are allowed in the step definition (in the Workflow Builder, double-click on a task, and select the DETAILS tab):

- **Processing can be rejected**
  Set this checkbox if the user can opt to skip this step. You can model alternative steps to be taken against the matching REJECTED outcome.
- **Step not in workflow log**
  Work items for this step do not appear in the standard logs, but they are always displayed in the technical workflow log. The graphical log filters out not only these steps but also the outcomes. If a step with several outcomes is filtered out, all of the outcome branches and the steps included in these branches are

filtered out of the graphical log, until the point is reached where the paths merge together again.

- **Advance with dialog (Synchronous dialog chain)**
  If the agent of the previous step is also an agent of this step, this step is executed immediately on completion of the previous step as described in Chapter 4, Work Item Delivery.

8.5 Documenting, Translating, Transporting, and Team Development

This section covers topics that should be included in your workflow design, including documenting your workflows, translation, moving your workflow from development to production, and options for collaborative team workflow development.

8.5.1 Documenting Workflow Definitions

Your project should have certain guidelines and expectations around workflow documentation. A workflow definition can be documented in several parts. First of all, you can describe the purpose of the workflow, how it is started, and which subworkflows it calls in the description of the workflow definition. This documentation is found in the Workflow Builder by selecting GOTO • BASIC DATA. Then select the DESCRIPTION tab. You can document using the NOTE IT! link. Additionally, at the task level, you can provide detailed task descriptions that will be available at runtime. Objects can be documented at the object level.

Note

The task description for dialog steps is displayed in the work item preview, so for dialog steps, only instructions to the user should be entered here. Only add technical documentation to the task description if the step is to be executed in the background.

Obviously, there is a lot more needed to document a workflow project than simply the workflow definition. Checklists can be downloaded from *http://www.sap-press.com* to help you with this.

8.5.2 Translating a Workflow into Other Languages

You can translate all language-dependent texts that appear in the workflow definition. This applies to:

- The names of steps and outcomes
- Decision texts and titles
- Container element names

A user can use personal settings to specify whether work item texts and the work item preview are to be displayed in the original language of the workflow or in the logon language (if the text is available in this language).

To get a complete translation, you have to translate the steps in the Workflow Builder and also translate all of the tasks used in the workflow in the translation transaction (SE63). Finally use the compare translation function in the Workflow Builder to import the changes into the current workflow version; otherwise, the changes will only be visible when the workflow is imported into the next system downstream.

8.5.3 Transporting New Versions of a Workflow

A new workflow always has one version with the number `0000`. This version is overwritten by default every time you save your workflow. If you do not want the system to do this, you can generate a new version by choosing Workflow • Generate version. The workflow definition is set to status New, Saved.

To avoid increasing the disk space requirements for workflow definitions excessively, only generate a new version under the following circumstances:

- If you have made incompatible changes
- If there are production workflows running that refer to the current version

The system manages several versions of a workflow definition. Only one of the versions of a workflow is the active version. Select the active version by activating the appropriate version.

Tip

The import and export parameters of the workflow container are not subject to any versioning.

The information tray displays the version you are processing and whether an active version is involved. To display an overview of all versions, choose the Basic Data symbol () in the Workflow Builder. The version number is displayed on the Version overview tab in the version-independent basic data. An overview of all versions of the workflow definition can be found on the Versions tab in the version-dependent basic data.

Tip

A running workflow always refers to the version of the workflow active at the time it started. Even if subsequently a new version of the workflow becomes the active version, workflows still running continue to refer to the version active when they were started. If you overwrite this version while there are still active workflows, for example, by making changes directly in the production system, unexpected errors can occur.

If a workflow definition is transported into another system, only the active version is transported. If the workflow definition exists in the target system with the same version number, it is overwritten by the transported version if it has no workflows running. Otherwise, the transported workflow definition is saved with a new, free version number. The transported workflow definition becomes the active workflow definition in the target system.

8.5.4 How Do You Share Workflow Development Within a Team

The Workflow Builder offers teamworking functions that support workflow development by a team of developers. It also offers the option to assign a self-defined grouping characteristic to each step. In the optionally displayable Teamworking frame, you can search for steps that have a particular grouping characteristic or particular change data. The Workflow Builder options can also graphically highlight the steps according to the grouping characteristics or according to the last user to make a change. The last user can be seen on the Change data tab of the step definition. You define the grouping characteristic of each step on the Change data tab page in the step definition.

A grouping name can be assigned to every step. Using this grouping name, you can structure the graphical representation of the workflow. The selection field displays all of the group descriptions previously defined. One of the group descriptions just entered is automatically entered in the selection field and is also available as a selection for all other steps. You can use this to mark steps that need rework or that are to be transferred to a subworkflow, or you can use this to demarcate the different logical parts of a large workflow.

Index