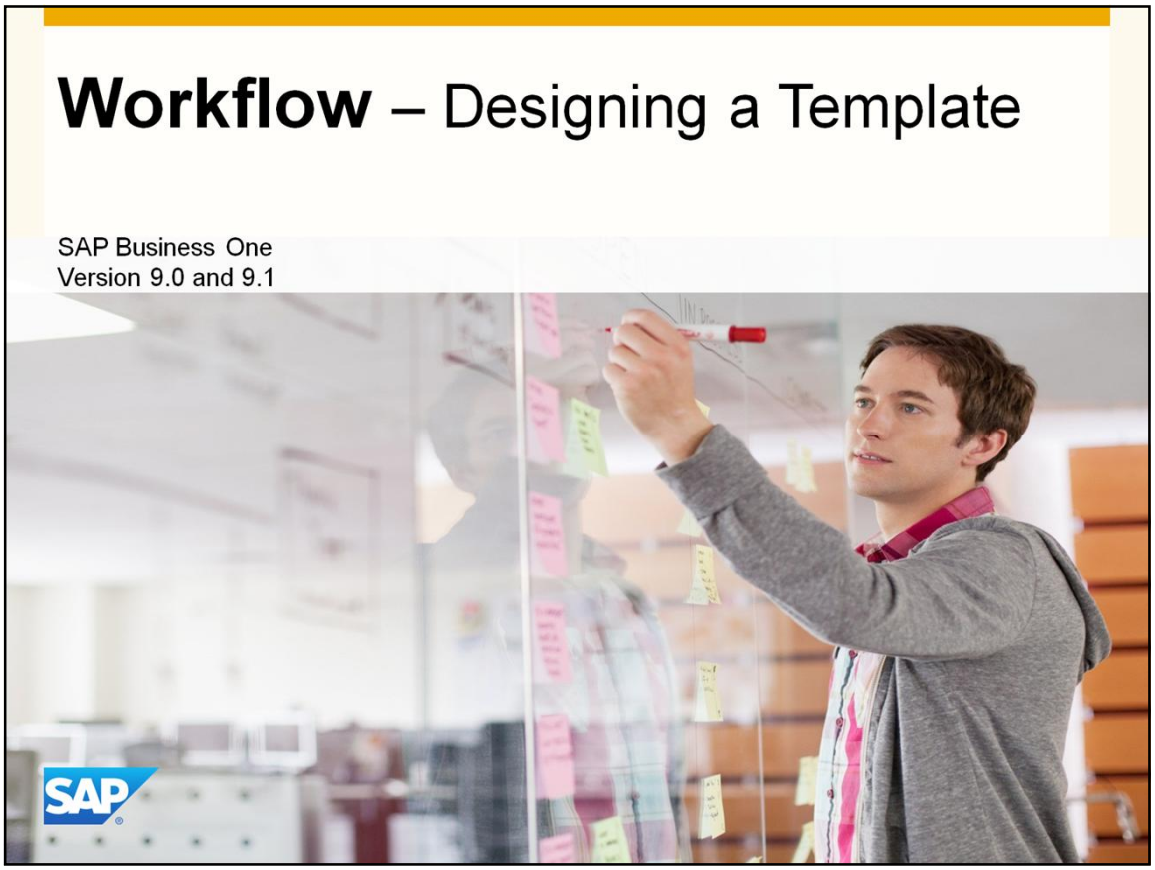
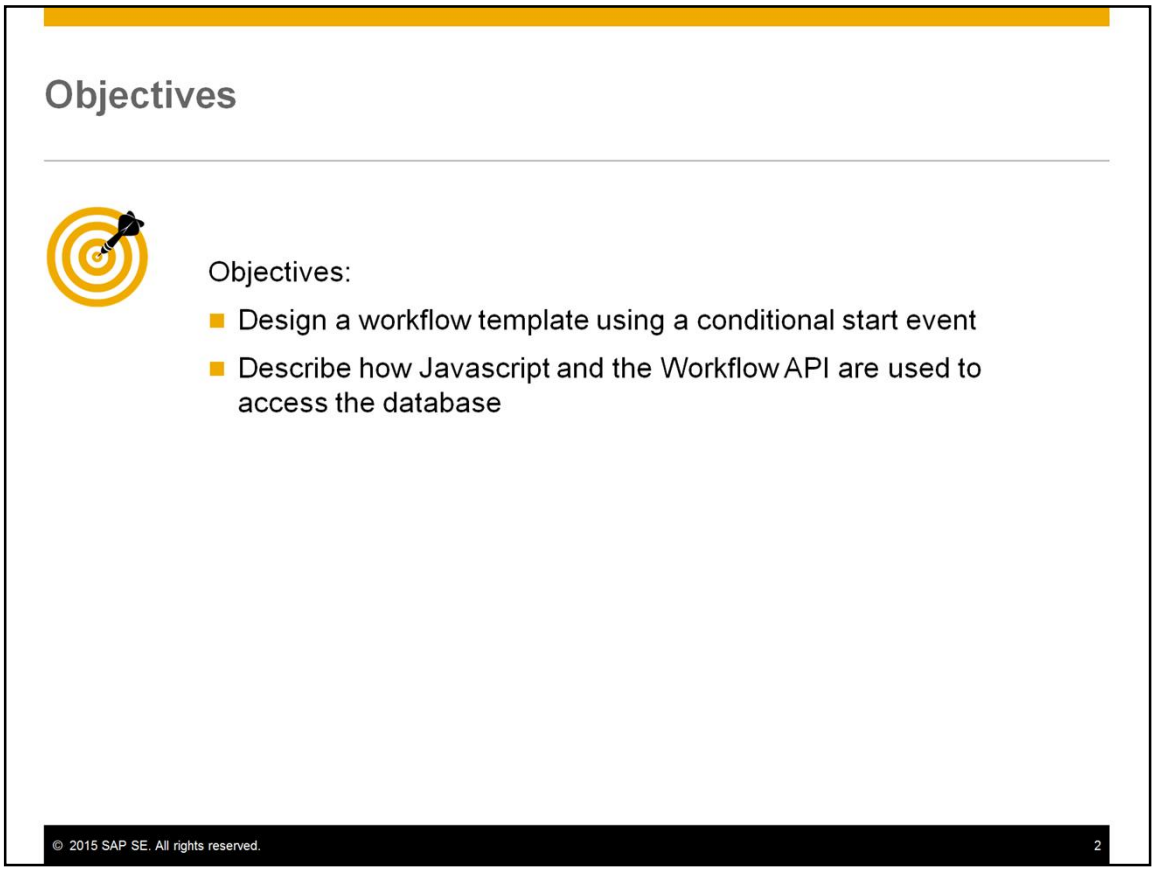


This course focuses on how to build a workflow template that uses a conditional start event.
An assumption is made that you have a basic understanding of workflow in SAP Business One.
To learn about the basics of workflow, see the companion course **Workflow – Introduction**.
This course is valid for release 9.0 and also includes some workflow features that are available in release 9.1 and up.



On completion of this course, you will be able to:

- Design a workflow template using conditional start event
- Describe how Javascript and the Workflow API are used to access the database

Note that this course covers Javascript and the Workflow API at a high-level, but does not provide in-depth training in these topics. If you are a business consultant, we recommend that you work together with a technical developer who is experienced in the SDK DI API.

Case Study - Business Example

OEC Computers wants to streamline the ordering process for built-to-order, high availability servers. Each sales quotation needs to be verified by a technician before it is sent to the customer. If the quotation fails the technical check, it must be changed and resubmitted for approval.

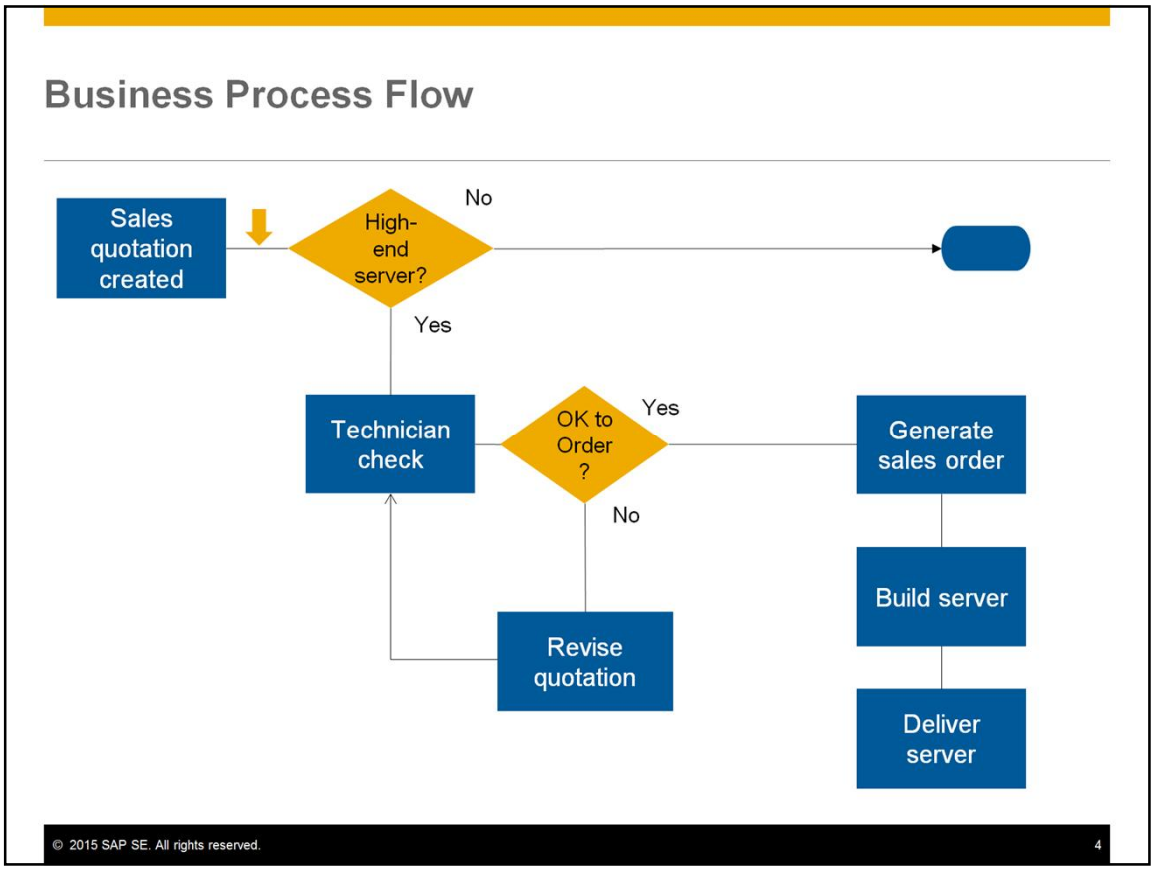
OEC wants to automate this process, and shorten the time it takes from quotation to sales order. The technician check only applies to the high availability server product line, not to other products.

Solution: Design a workflow template using a conditional start event.

OEC Computers wants to streamline the ordering process for built-to-order, high availability servers.

There is a requirement to have the sales quotation verified by one of the technicians before it is sent out to the customer. OEC wants to automate this process, and shorten the overall time it takes from quotation to sales order. The technician check only applies to the high availability server product line, not to other products.

In this course, we will design a workflow template for this process.

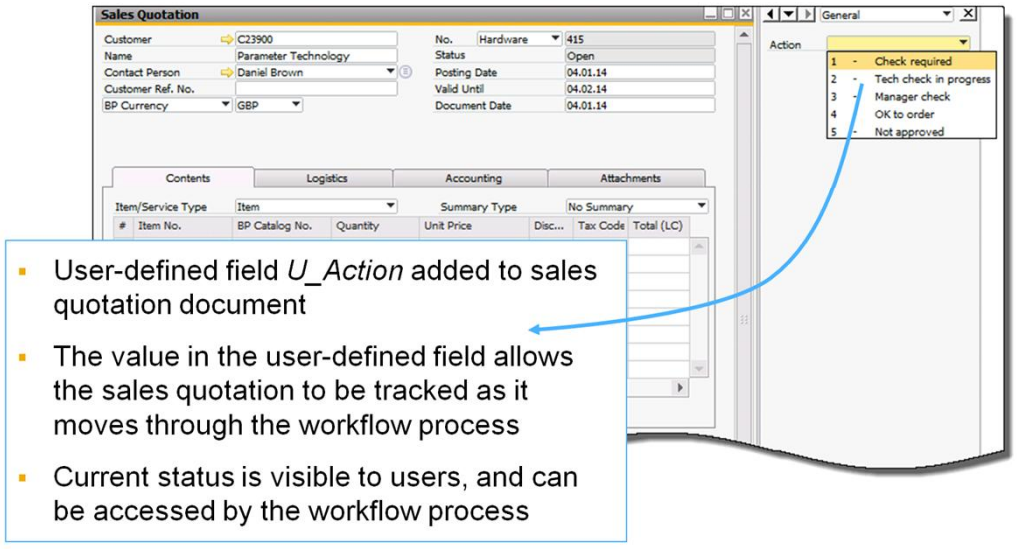


Here is a simplified representation of the OEC Computers' business process as a flow chart. Having this information available, we can start to plan the workflow template.

- The workflow process will be triggered when a sales quotation is created. The actual creation of the quotation is not part of the workflow process that will be modeled.
- Only sales quotations for specific high-end server products are to be processed by the workflow. Therefore the workflow template needs to identify these quotations and ignore quotations for other products.
- The technician checks the sales quotation, and if the configuration is technically accurate the quotation is ready to be ordered.
- Any changes to the sales quotation must go back for approval.
- After the sales order is generated from the quotation, the server can be built and delivered. For the purposes of this training course, the workflow process does not cover the production step.

User-Defined Fields and Workflow

- How should the workflow process identify sales quotations that require a check?



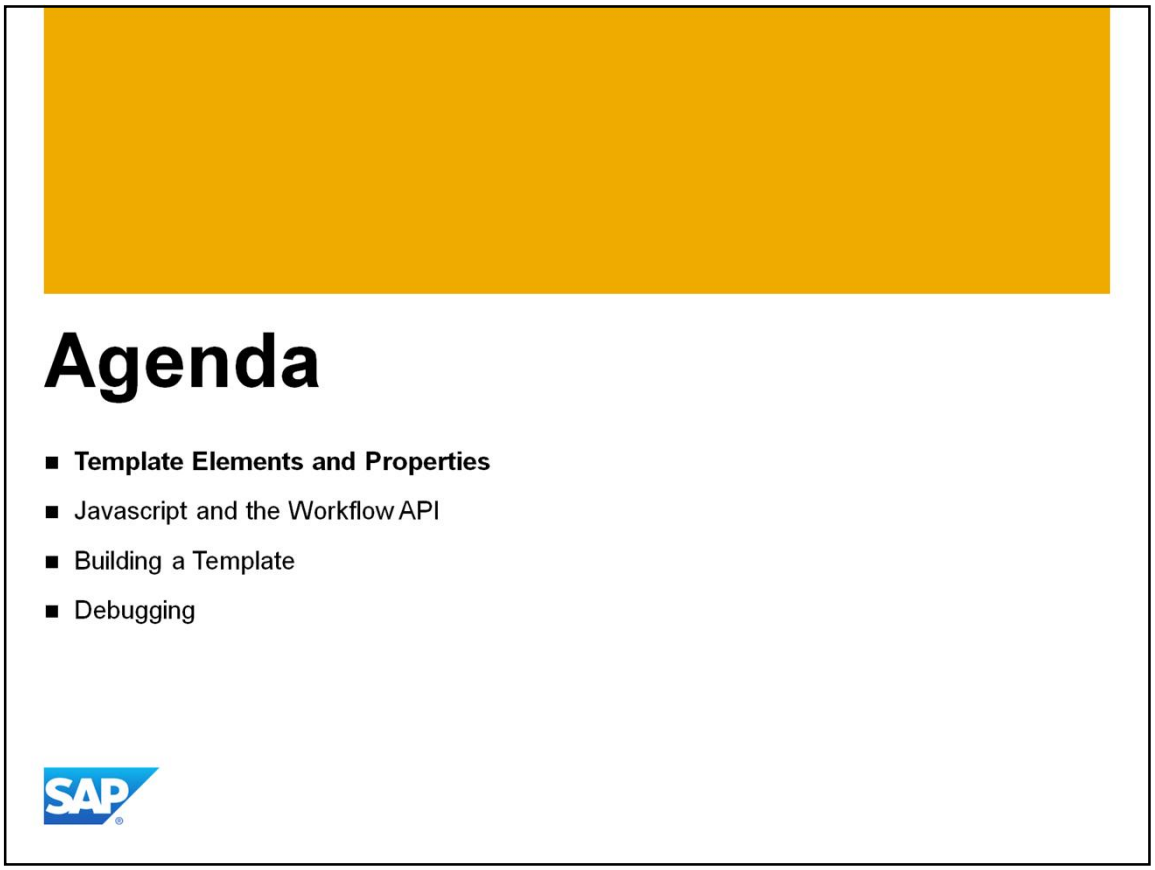
- User-defined field *U_Action* added to sales quotation document
- The value in the user-defined field allows the sales quotation to be tracked as it moves through the workflow process
- Current status is visible to users, and can be accessed by the workflow process

You may be wondering how the workflow process will identify sales quotations that require a check. There are several ways to implement this. For the case study, we added a user-defined field to the header of the sales quotation document.

This will allow the sales quotation to be tracked as it moves through the steps in the workflow process. The current status in the user-defined field will be visible to users, and can also be accessed by the workflow process.

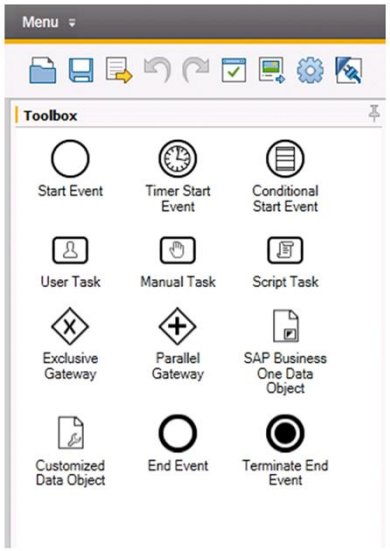
This user-defined field, which we call *U_Action* has the following values:

- If a technical check is required, the user selects the value 1 when the sales quotation is created
- A value of 2 means that the sales quotation is currently being checked
- A value of 3 signifies that the sales quotation needs to be changed and resubmitted for approval
- A value of 4 means that the check is completed and the sales quotation is ready to be converted to a sales order
- A value of 5 indicates a manager has determined that the sales quotation cannot be approved



The first part of this course reviews the key elements of a workflow template.

SAP Business One Toolbox Elements



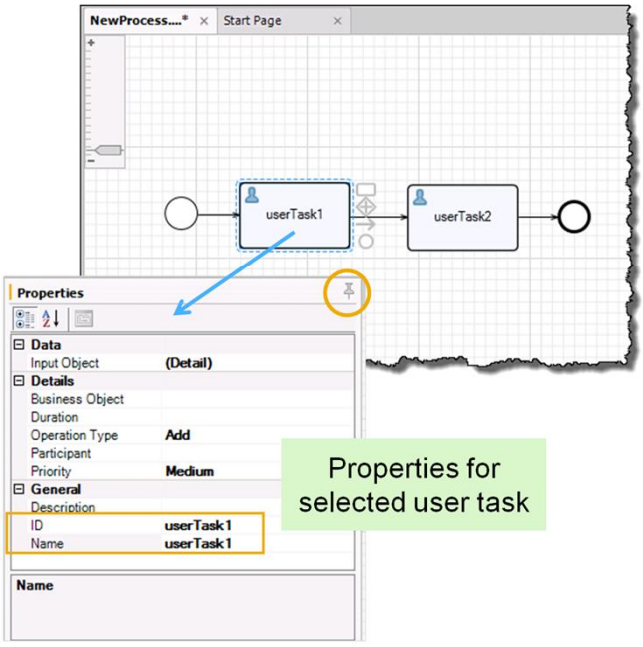
- The SAP Business One Studio toolbox provides the building blocks for the workflow template
- You need to evaluate which elements can be used to model the business process

The SAP Business One Studio toolbox provides the building blocks for the workflow template.

When you design a workflow template, you need to evaluate which elements (icons) you will use to most accurately model the business process.

Properties Window

- Each element has a unique set of properties
- *Properties* window will open when you select element in working pane
- Properties window is moveable and can be docked by selecting pin
- Every element has a pre-allocated ID and Name
- You can change the ID and name, but the new ID must be unique within the template



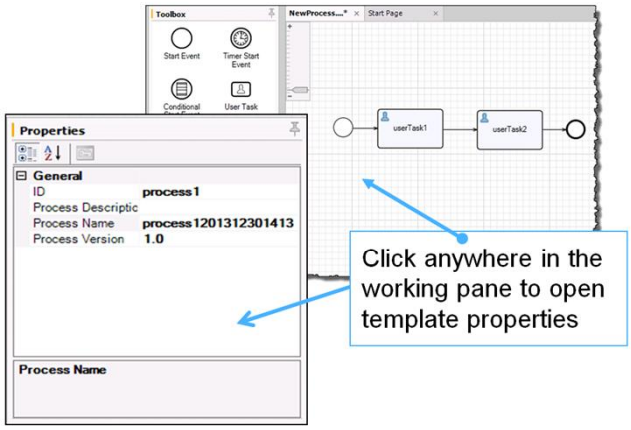
Each element in the workflow template has a unique set of *Properties*. The properties window will open when you select an element in the working pane with your mouse. In the example shown here, when you select the task *userTask1*, the properties window is opened.

Note that the properties window is moveable and can be docked anywhere in the working pane by selecting the pin in the top right corner.

Every element has a pre-allocated *ID* and *Name* in the properties. For example, the user task shown above has been allocated the ID and Name “userTask1”. You can change both the ID and the name of an element, but the new ID must be unique within the workflow template.

Properties for Workflow Template

- The workflow template itself has a set of properties
- Click anywhere in working pane area to open template properties
- Template properties include pre-allocated ID, Process Name and Version – these can be changed



- Process Version initially set as 1.0
- Template name is combination of ID and Process Version. If you change an existing template in the SAP Business One Studio, you must change the ID or version before you re-import

The workflow template itself has a set of properties. Click anywhere in the working pane area to open the template properties. Make sure you do not select an element or line connector.

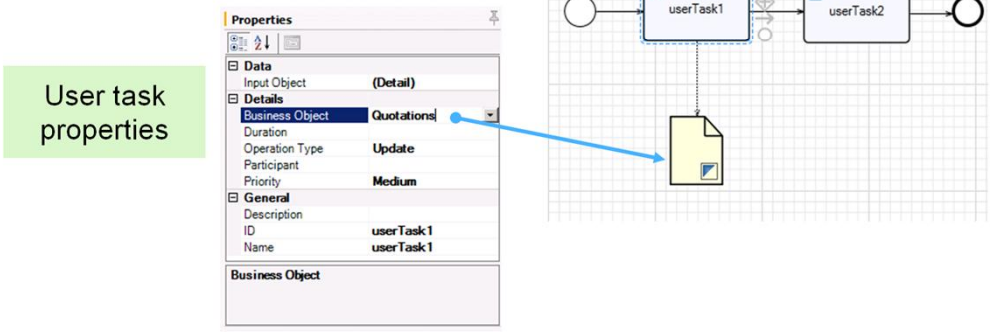
The template properties include a pre-allocated ID, Process Name and Process Version. You can change any of these. You can optionally enter a process description. The process version for a new template is initially set as 1.0.

The template name is a combination of the ID and Process Version. If you make a change to an existing template in the SAP Business One Studio, you must change the ID or version in the properties, since you cannot re-import a template into SAP Business One with the same ID and version as an already imported template.

Defining Output Data Objects

- After you make the selection from the dropdown list, the output data object is automatically connected with an output arrow from the task

User task properties

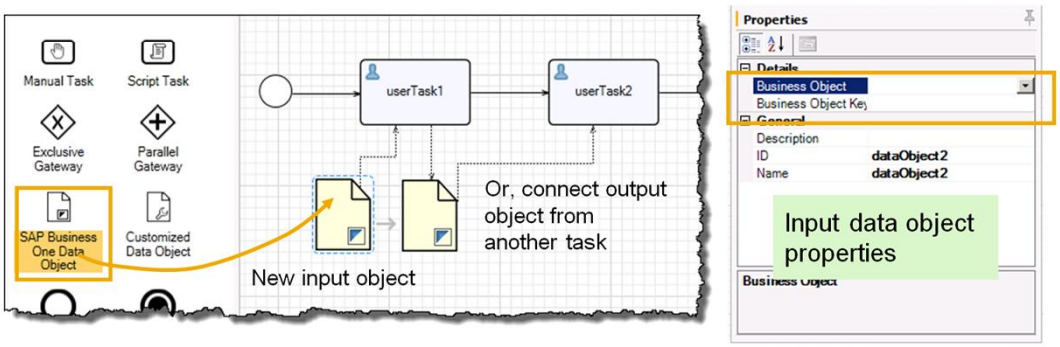


A user task always has an output object. After you select the business object in the task properties, an output data object is automatically added to the working pane and connected with an output arrow from the task, as shown here.

If the operation type is *Add*, the user will be presented with a blank document when he or she picks and processes the task. If the operation type is *Update*, the user is presented with an existing document for update.

Defining Input Data Objects

- To define an input object for a task, either:
  - Drag *SAP Business One Data Object* element on to working pane and connect to user task, *or*
  - Use the output object from another user task as input



To define an *input data object* for a user task, you have two options:

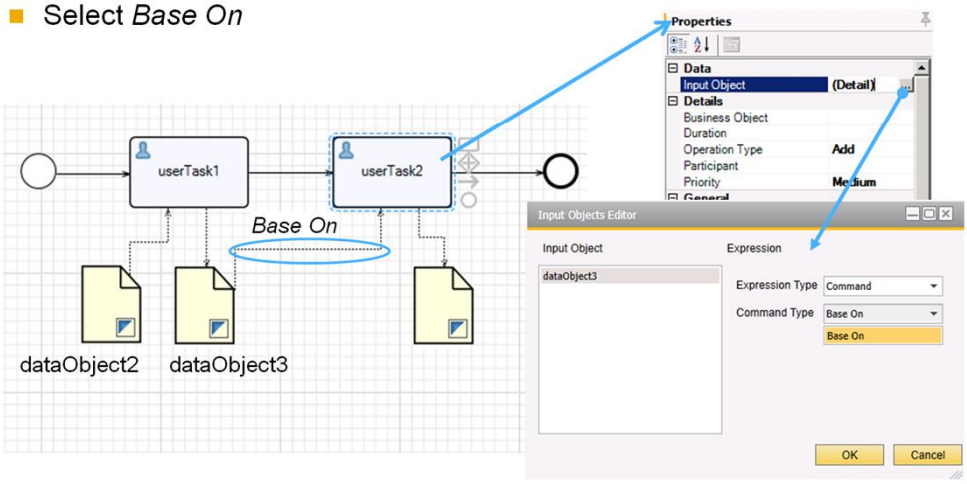
- Drag an SAP Business One Data Object element onto the workflow pane and connect with a line into the user task, or
- Use the output object from another user task by connecting it as input to a task.

When you use a new input object from the toolbox, you need to select the business object in the data object properties window.

Notice the input object has a pre-allocated ID and Name in the properties.

Input Object Properties

- If the input data object is based on the output object from another task:
  - In task properties, select the *Input Object* property
  - Select input object ID from list in
  - Select *Base On*



If the input data object for a task is based on the output data object of another task, you need to specify the *Base On* expression.

To do this, open the properties window for the task, select the *Input Object* property and select the input object ID from the list shown. Then select *Base On* as the command type. As a result of this, when the user processes the task, the input document will be the base document.

In the example shown, the output object for *usertask1* is *dataObject3* and this is also the input object to *userTask2*. When the user processes task 2, the base document will open.

If you do not specify the relationship, the two data objects are assumed to be unrelated.

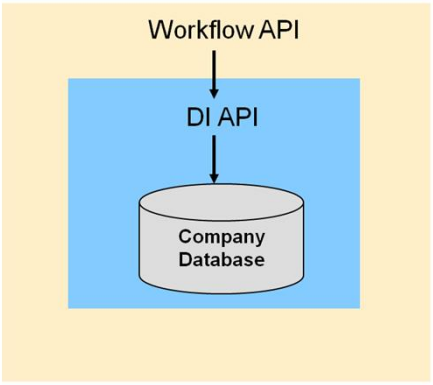
Agenda

- Template Elements and Properties
- **Workflow API**
- Building a Template
- Debugging

In this section, we look at how the Workflow API is used in a workflow template to define a conditional start event.

Workflow API

- In the case study, we want the workflow process to start automatically whenever there are eligible sales quotations to be checked
- We use a *conditional start* that queries the database for new sales quotations
- In the conditional start event, we use the Workflow API to access the database



Ref: Workflow API Reference

In the case study, we want the workflow process to start automatically whenever there are sales quotations to be checked. Therefore we will use a conditional start event. The conditional start event runs a SQL query to detect if there are any eligible sales quotations, and if the query returns a true result, the condition event triggers a workflow instance. If the query returns false, the workflow engine will monitor the condition again at the pre-configured interval (default is 10 minutes).

The code for the conditional start event uses the Workflow API to access the database. You can also use the Workflow API in script tasks.

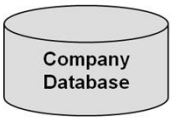
The Workflow API is a subset of the DI API and is documented in the **Workflow API Reference** guide shipped with the product.

Workflow API – Running a Database Query

- *Company* object represents an SAP Business One company database (methods *getRecordsetParams* and *getRecordset*)
- You can pass SQL queries and select records from the database using the *getRecordsetParams* and *getRecordset* methods

Example:

```
var param = company.getRecordsetParams();
var query = "SELECT TOP 1 DocNum FROM OQUT where U_Action = \'1\'";
param.setQuery(query);
var recordset=company.getRecordset();
recordset.doQuery(param);
// check for any results
var success = recordset.read();
if (success)
{
      var field = recordset.getField(0);
      var docnumber = field.getColumnValue();
}
```



The *Company* object represents an SAP Business One company database. The methods *getRecordsetParams* and *getRecordset* are defined in the Workflow API as methods for the object *company*. You can pass SQL queries and select records from the database using these functions. Note that, unlike the DI API, you do not have to explicitly set up a connection to the company database. This is done for you when you activate workflow for a company.

In the example you can see a highlighted statement. Javascript variables are used to hold parameters and store data from the results.

A SQL SELECT query string has been stored in a Javascript variable called *query* .

The *getRecordsetParams* method passes the query string as a parameter using the *setQuery* method. The *Recordset* object is used hold the database results. The *doQuery* method for this object enables you to run the query.

The read() function sets the cursor to the first valid record in the results. It works similarly to the EOF function in the DI API.

The *getColumnValue* method gets the selected field value from the query results.

The document number selected by the SQL query is stored into a local Javascript variable called *docnumber.*

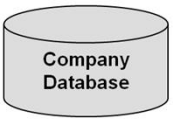
Note that Javascript variables are declared as they are used. Because quote marks are a special character in Javascript, you must insert a backslash character “\” in front of each quotation mark in the SQL query string.

Workflow API – Updating a User-Defined Field

- To add or update a business object, use the *getBusinessService* and *getBusinessObject* methods
- Object types defined in SDK Help (for example: sales quotation is 23)

Example:

```
var docservice = company.getBusinessService("23");
docobject = company.getBusinessObject("23", docnumber);
docobject.getUserFields().put("U_Action", "2");
docservice.update(docobject);
```



To add or update a business object in the company, use the *getBusinessService* followed by the *getBusinessObject* methods. The object type is passed as a parameter to the *company.getBusinessService* method. The object type for a sales quotation is 23.

In the example, when we find a sales quotation for processing, we want to change the value of the user-defined field from 1 to 2.

In the *getBusinessObject* method, we again use the object type and also the Javascript variable *docnumber*. If you remember, this contains the saved document number of the selected sales quotation.

You can find the list of object types by searching in the SDK Help Center using the keyword “BoObjectTypes Enumeration”.

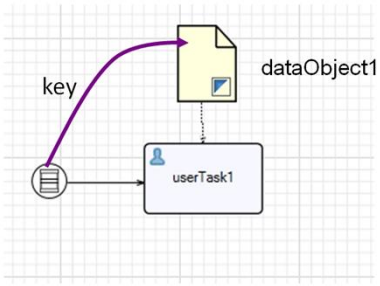
Be aware that you cannot update the record using SQL Update. You must use the Workflow API methods.

Workflow API – Updating a Data Object in a Template

- *StartService* service used in conditional start event to pass parameter to data object in workflow instance
- Typically the key of a record retrieved from the database:

Example:

```
obj = StartService.getStartDataObject("dataObject1");
obj.putItem("Key", docnumber);
obj.putItem("ObjType","23");
```



From release 9.1 onwards, you have the ability to pass a parameter from the start event to an object in the workflow template.

Typically this parameter might be the document number of the input data object for the first task.

Use the *StartService* function to access the data object, using the unique object ID. In the example we use *dataObject1* which is the ID assigned to the input data object for *userTask1*.

We then insert the document number in the object as the key. The document number was stored in a Javascript variable called *docnumber.*

Note that the StartService function does not start the workflow instance, it just passes a parameter to a data object when the workflow instance runs. The workflow instance starts when the conditional start event returns a positive value.

Agenda

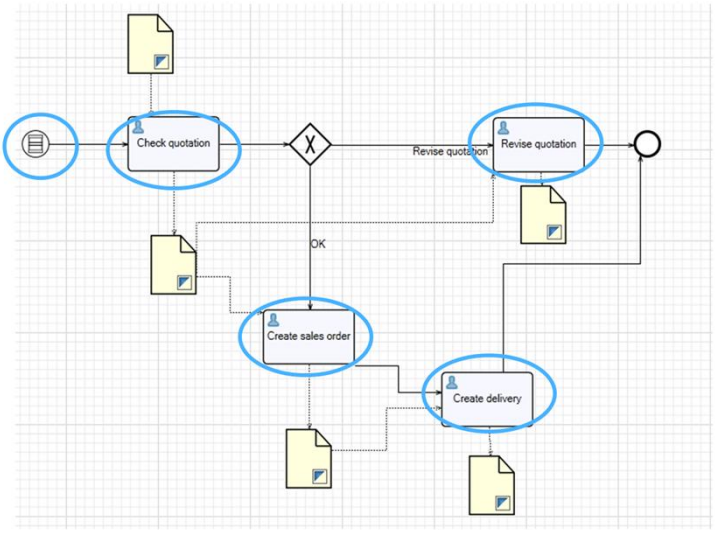
- Template Elements and Properties
- Javascript and the Workflow API
- **Building a Template**
- Debugging

The next part of this course shows you how to build a template step by step for the OEC Computers case study.

Completed Template

Main elements:

- Conditional start event
- User task Check Quotation
- User task Create Sales Order
- User task Create Delivery
- User task Revise Quotation

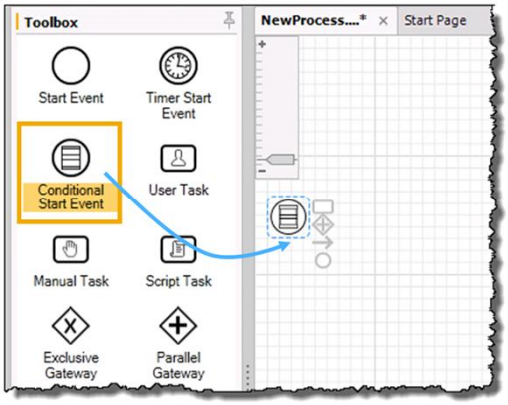


Here is what the completed template looks like. We will work through this step-by-step in this topic. The main elements of the template are:

- A conditional start event
- User task Check Quotation, in which the technician validates the new sales quotation
- User task Create Sales Order, in which the validated sales quotation is converted into a sales order
- User task Create Delivery, in which the ordered product is shipped to the customer
- User task Revise Quotation, in which the sales quotation is revised so that it can be validated again.

Building a Workflow Template - 1

Start > All Programs > SAP Business One > SAP Business One Studio



- In the SAP Business One Studio, create a new workflow project
- Drag a conditional start event onto the working pane
- A setting in the workflow configuration file determines how often the workflow server monitors the start event

We open the SAP Business One Studio and create a new workflow project.

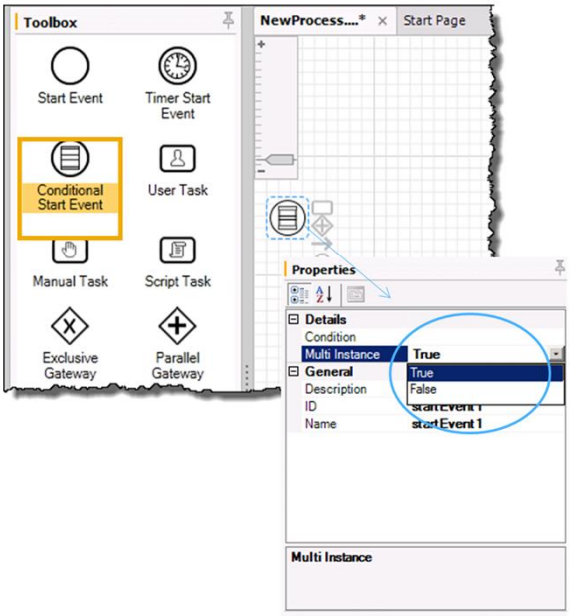
We start by dragging a conditional start event onto the working pane.

The *Conditional Start Event* will check for new sales quotations and trigger the workflow process if records are found.

A setting in the workflow configuration file determines how often the workflow server monitors the start event condition.

When you are testing a workflow template, to avoid having to wait for the time interval, you can manually start the instance in SAP Business One using the *Workflow Instance* menu.

Building a Workflow Template - 2



- A workflow that starts using a conditional or timer event can be set as multi-instance or single instance
- Choose true or false in the start event properties

A workflow that starts using a conditional or timer event can be set as *multi-instance* or *single instance*.

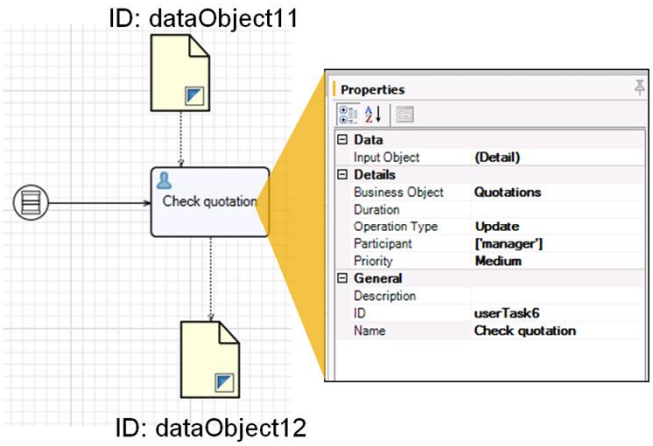
To set this, choose true or false in the start event properties:

- If multi-instance is false, the workflow engine will not start another instance until the previous instance has completed. Each instance will be processed serially. Not using multi-instance could be problematic if an instance is delayed, for example, because a user cannot complete a task because they are out of the office or there is a manual task that takes longer than planned.
- If set to true, multiple instances of the workflow can overlap. This is useful when a workflow process is performed by multiple users.

For the case study, we select multi-instance.

Building a Workflow Template - 3

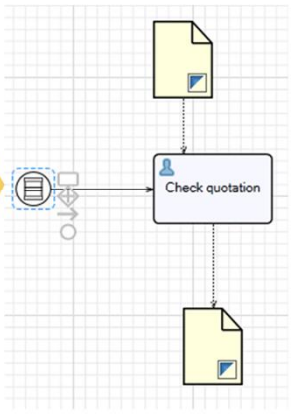
- Drag a user task onto the working pane and connect to the start event
- Drag an SAP Business One Data object as the input object for the task
- In the task properties, select Quotations as the output object – this will automatically be added and connected to the user task



- Next, drag a user task onto the working pane and connect to the start event. Set the Operation Type to *update*, and select a participant. This is the person who will validate the new sales quotation.
- Then drag an SAP Business One Data object as the input object for the user task. We select the business object *Quotations*. We need to make a note of the ID of this data object, since the start event logic will store the document number as the key for this object. Here the ID is *dataObject11*.
- Back in the task properties, select *Quotations* as the output object – the object will automatically be added and connected to the user task.

Building a Workflow Template - 4

```
1  // set and run query
   var param = company.getRecordsetParams();
   var query = "SELECT TOP 1 DocNum FROM OQUT where U_Action = \'1\'";
   param.setQuery(query);
   var recordset=company.getRecordset();
   recordset.doQuery(param);

2  // check for results
   var success = recordset.read();
   if (success) {
   // get document number
   var field = recordset.getField(0);
   var docnumber = field.getColumnValue();
   print("\r\n Quotation document number =" + docnumber + "\r\n");

3  // update UDF value in database
   var docService = company.getBusinessService("23");
   doc = company.getBusinessObject("23",docnumber);
   doc.getUserFields().put("U_Action", "2");
   docService.update(doc);

4  // insert document number in input quotation
   var obj = StartService.getStartDataObject("dataObject11");
   obj.putItem("Key", docnumber);
   obj.putItem("ObjType","23");

   }

   success > 0;
```



Code for conditional event

Now we know the ID of the data object representing the input sales quotation, we can complete the code for the conditional start event.

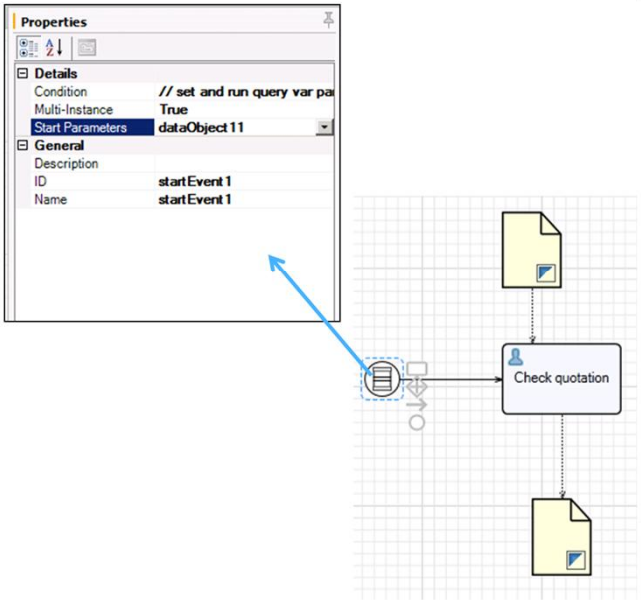
We select the start event icon and open the script editor window in the properties. We type or paste the code into the script editor.

The details of the Workflow API were discussed in the previous topic of this course. Basically the code:

1. Assembles the SQL query and runs the query
2. Checks for any results and retrieves the document number from the results
3. Updates the value in the user-defined field of the sales quotation
4. Inserts the document number as a key for *dataObject11* in the workflow instance

Building a Workflow Template - 5

- In release 9.1 and on, you can specify a start parameter for the workflow instance.
- Parameter is entered in properties window for the start event icon
- For a conditional start event, parameter is typically the document number
- For a manual start event you can use the document number or a date range

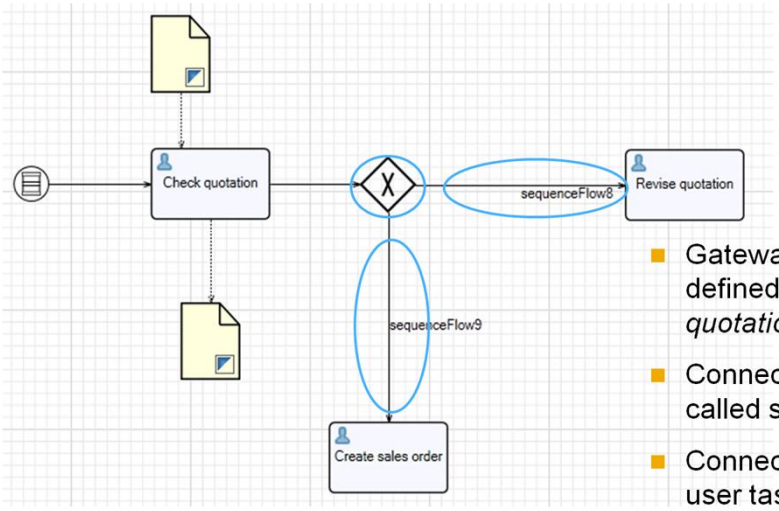


In release 9.1 and on, you can specify a parameter for the workflow instance.

We define the start parameters in the properties for the conditional start event. In this case, we select the ID of the input data object, which is the sales quotation retrieved from the database.

For manual start events, you can define a variable in the template and use as a parameter. When the workflow starts, you get a prompt and can enter a value, for example, a specific document number or a date range. For example, you might design a workflow template that only processes certain documents on the last day of the month.

Building a Workflow Template - 6



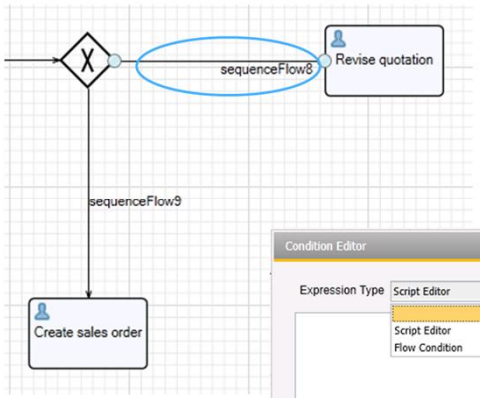
- Gateway checks value of user-defined field set by *Check quotation* task
- Connector lines from gateway called sequence flows
- Connect sequence flows to user tasks

In the workflow template, we next drag an *Exclusive Gateway* element onto the working pane. Using this gateway we can test the value of the user-defined field set by the task “Check quotation”.

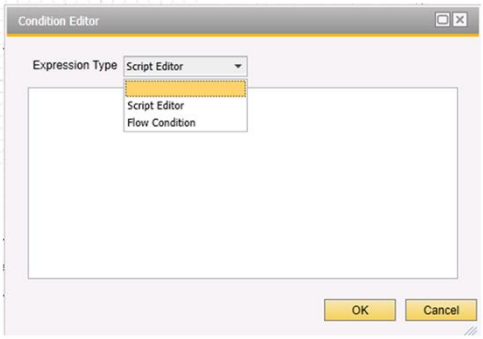
The connector lines from an exclusive gateway are called *sequence flows*. In order to define the sequence flow conditions, we need to drag other elements into the working pane.

We connect each sequence flow to a user task.

Building a Workflow Template - 7



- Condition is set on each sequence flow from gateway
- Two ways to define condition:
  1. Type Javascript directly
  2. Build condition step-by-step using *Flow Condition*



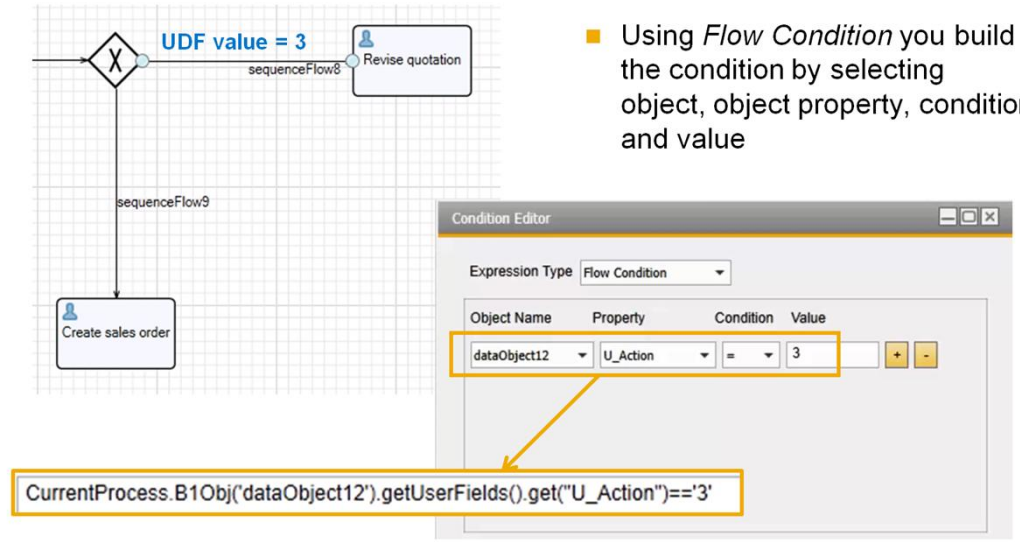
The gateway condition is not set on the gateway properties, but on each of the sequence flows branching from the gateway.

Select the connector line to open the properties and choose the browse button to open the *Condition Editor* window.

The condition is entered using Javascript and there are two ways to define Javascript for a sequence flow:

- Choose *Script Editor* to directly type in the Javascript, or
- Choose *Flow Condition* editor to automatically build the condition step-by-step

Building a Workflow Template - 8



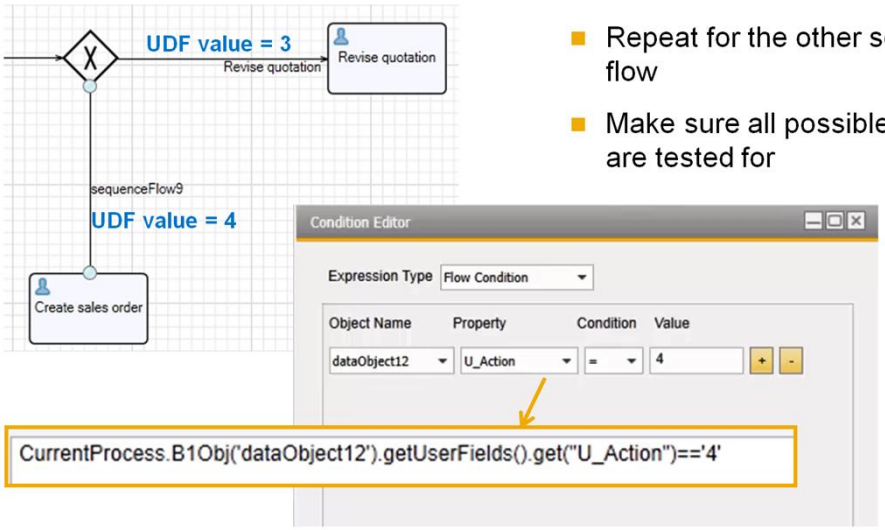
- Using *Flow Condition* you build the condition by selecting object, object property, condition and value

The *Flow Condition* editor is easy to use and lets you build the condition by selecting:

- The object ID from the template
- The object property
- The condition
- The value for the condition

The result is shown here. We test if the value of the user-defined field is 3. This indicates that the validation of the sales quotation has failed.

Building a Workflow Template - 9

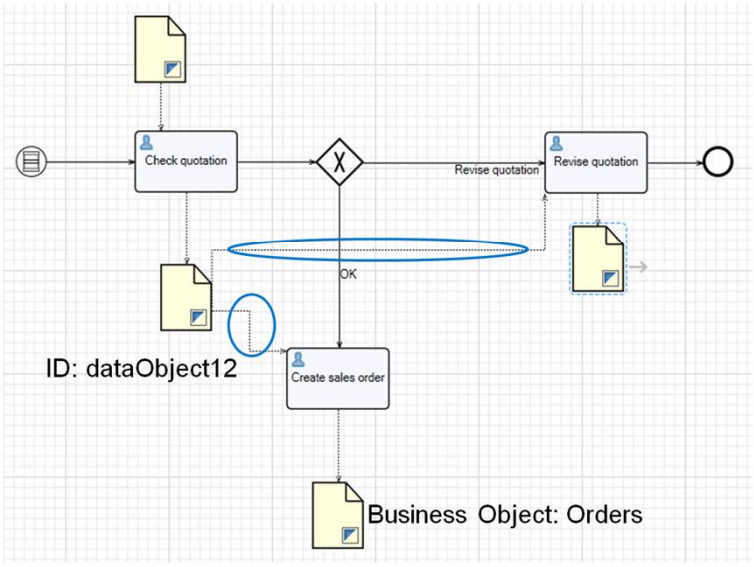


- Repeat for the other sequence flow
- Make sure all possible values are tested for

Similarly we define the condition for the other sequence flow. We test if the value of the user-defined field is 4. This indicates that the sales quotation is technically valid.

Note that when you use a gateway to check a field that can have multiple values, you should ensure that all possible values are tested for. In this example, we could add a third sequence flow that is taken when the value is NOT 3 or 4.

Building a Workflow Template - 10



- Connect the updated sales quotation from the *Check quotation* task as the input data object to the *Revise quotation* and *Create sales order* tasks
- Use *Base On* expression
- Add end event

Now we set the input and output data objects for the remaining user tasks.

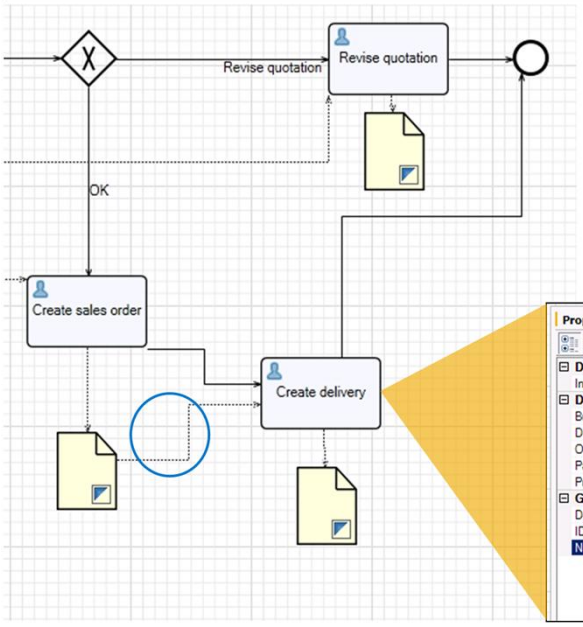
We connect the updated sales quotation from the *Check quotation* task as the input data object to the *Revise quotation* and *Create sales order* tasks.

We select the business object Orders as the output object from the *Create sales order* task. Because this is based on the sales quotation, we use the *Base On* expression in the input object properties for the task.

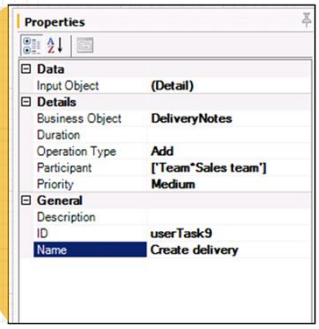
We select the business object Quotations as the output object from the *Revise quotation* task. Because this is based on the incoming sales quotation, we use the *Base On* expression in the input object properties for the task.

We add an end element to the workflow template and connect the *Revise quotation* task, since this is the final task in the workflow process.

Building a Workflow Template - 11



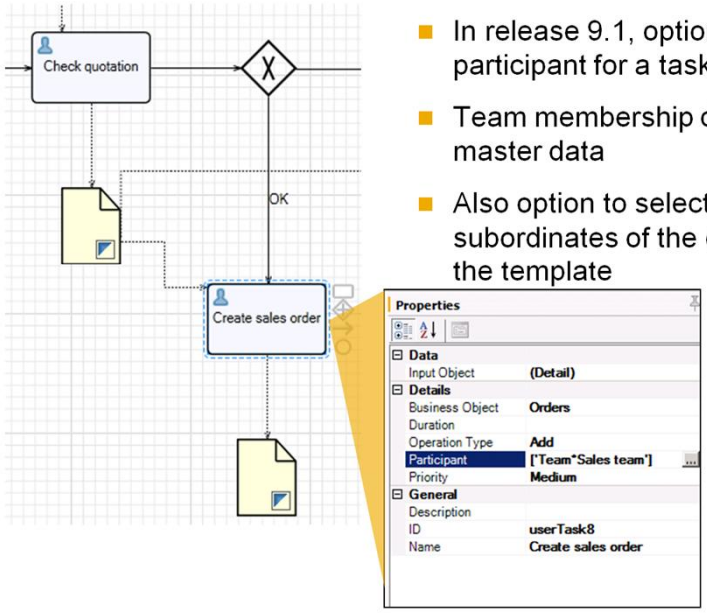
- We add the last user task to the template
- We connect the output sales order from the *Create sales order* task as input to this task, and select *DeliveryNotes* as the output business object



Finally we add the last task to the template. We add the *Create Delivery* task and connect the output from the *Create sales order* task as an input to this task. This task has a *DeliveryNotes* object as the output.

We also connect this task to the end event.

Specifying a Team as a Participant

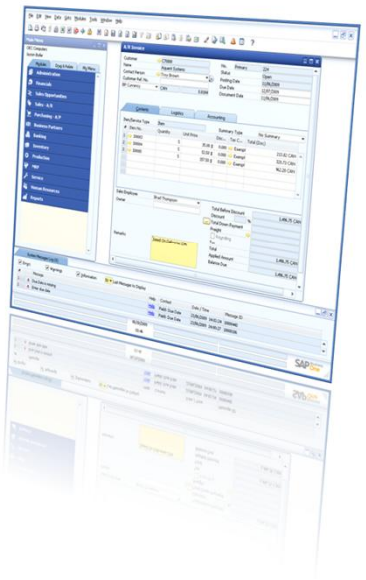


- In release 9.1, option to specify a team as participant for a task
- Team membership defined in employee master data
- Also option to select the manager or subordinates of the owner of another task in the template

In release 9.1 you have more flexibility for defining the users who will receive the task for pick up. You can use the name of a team as a participant. All users who belong to the team will receive the task (only the first one can pick up). The team membership is defined in the employee master data.

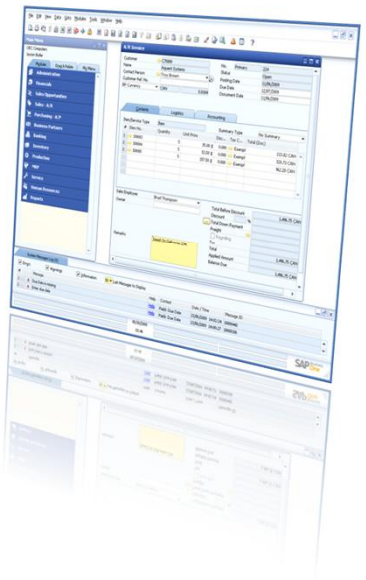
You can also pecify the manager or subordinates of the owner of another task in the template. This is useful when a task need to be approved by a user's manager, or when a manager creates a new object, such as a customer master record, that needs to be completed by a subordinate.

Demo: Design a Workflow Template



In this demo, you will see how to design a workflow template.

Demo: Import and Run a Workflow Template

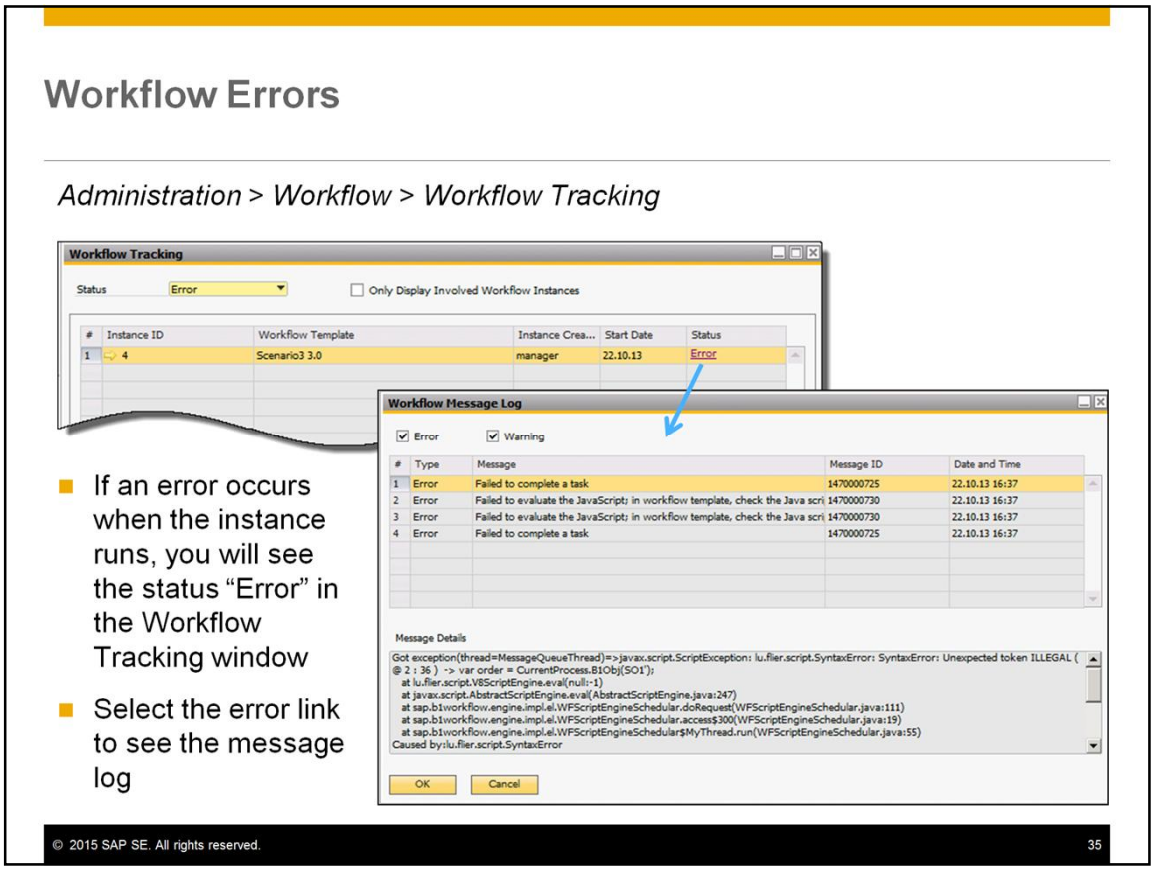


In this demo, you will see how to import and run a workflow template.

Agenda

- Template Elements and Properties
- Javascript and the Workflow API
- Building a Template
- **Debugging**

The last part of this topic provides some guidelines for debugging workflow templates.



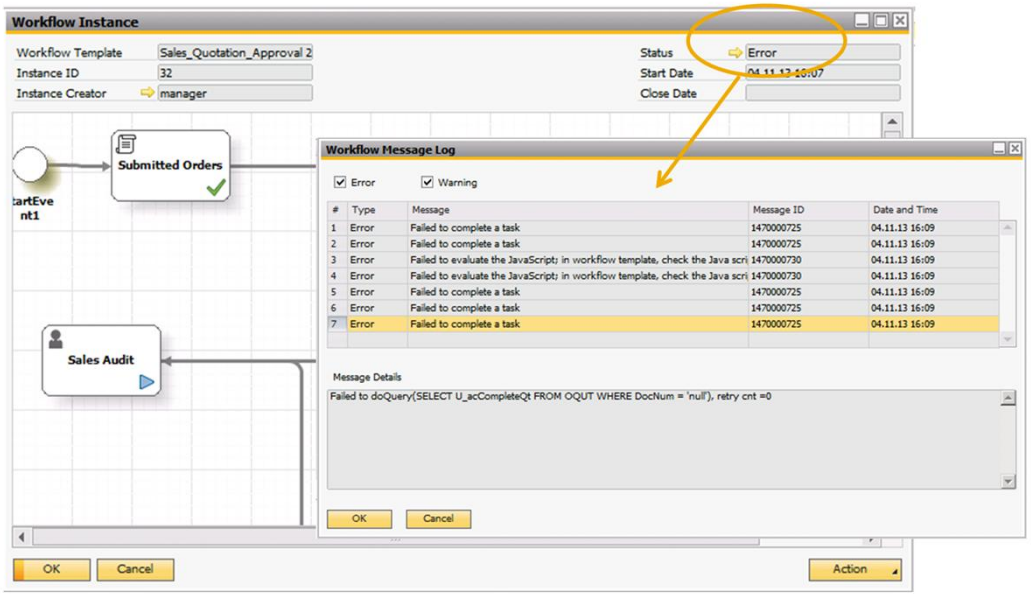
If an error occurs when the workflow instance runs in SAP Business One, you will see the status “Error” in the *Workflow Tracking* window in SAP Business One. When you select the error link, you will see detailed error messages in the message log.

Common problems are Javascript syntax errors, such as missing quotation marks, or using the wrong name for a data object.

If you encounter an error when testing a new workflow template, remember to change the version number for the template after correcting the error in the SAP Business One Studio in order to re-import the template into SAP Business One.

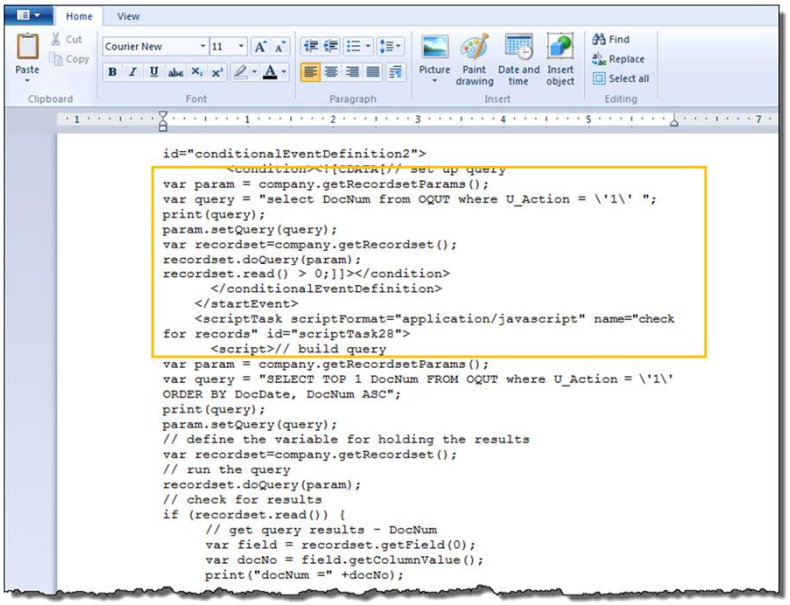
Workflow Message Log

Administration > Workflow > Workflow Instance



You can also access the same message log from the *Workflow Instance* window if there is an error. Here you can also see a graphical view of the tasks in the workflow that have completed – these are marked with a green check mark.

Workflow Template File



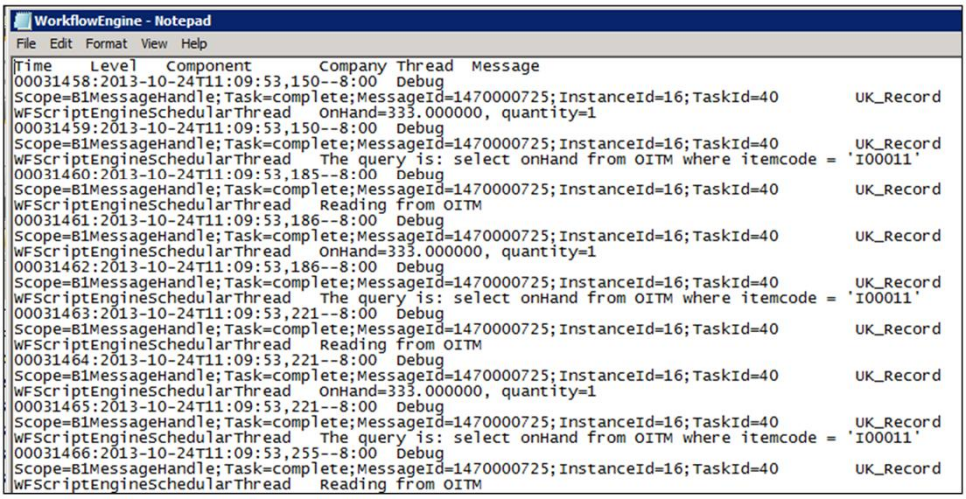
- When you save the workflow template, the xbml file contains the entire code for the template, including conditional event and the script tasks

When you save the workflow template, the xbml file contains the entire code for the template, including the Javascript in the conditional event and script tasks.

Examining the script within the <script> tags can sometimes be useful for identifying errors in the scripts.

Workflow Log

C:\Program Files\SAP\SAP Business One Server Tools\Workflow\Log



The workflow log is produced by the workflow service. You can access the log file by navigating to the *SAP Business One Server Tools* directory on the server. This folder contains the Workflow folder, which contains the log.

It is a good practice to include the *print()* function when you first test the script tasks in your template.

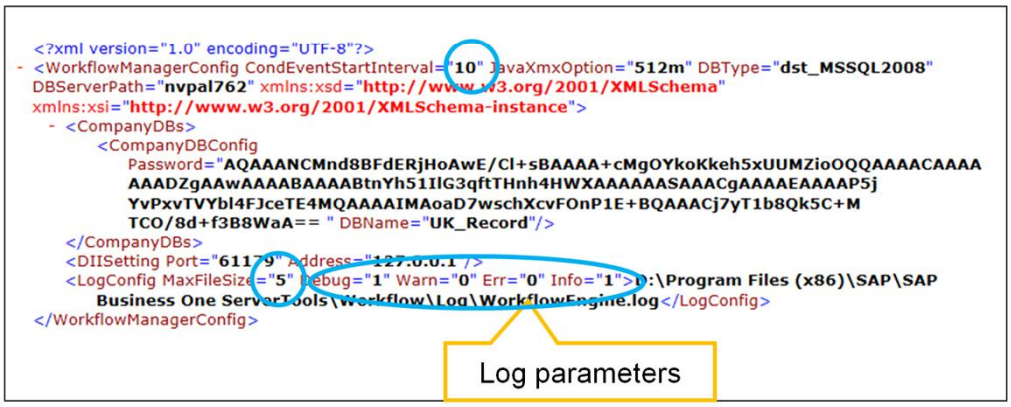
The print output appears in the workflow log. Include a unique text string in the print function to make it easier to search the log.

The newest entries in the workflow log are at the bottom of the file.

Workflow Configuration File

Workflow configuration file *b1-workflow-manager.xml*

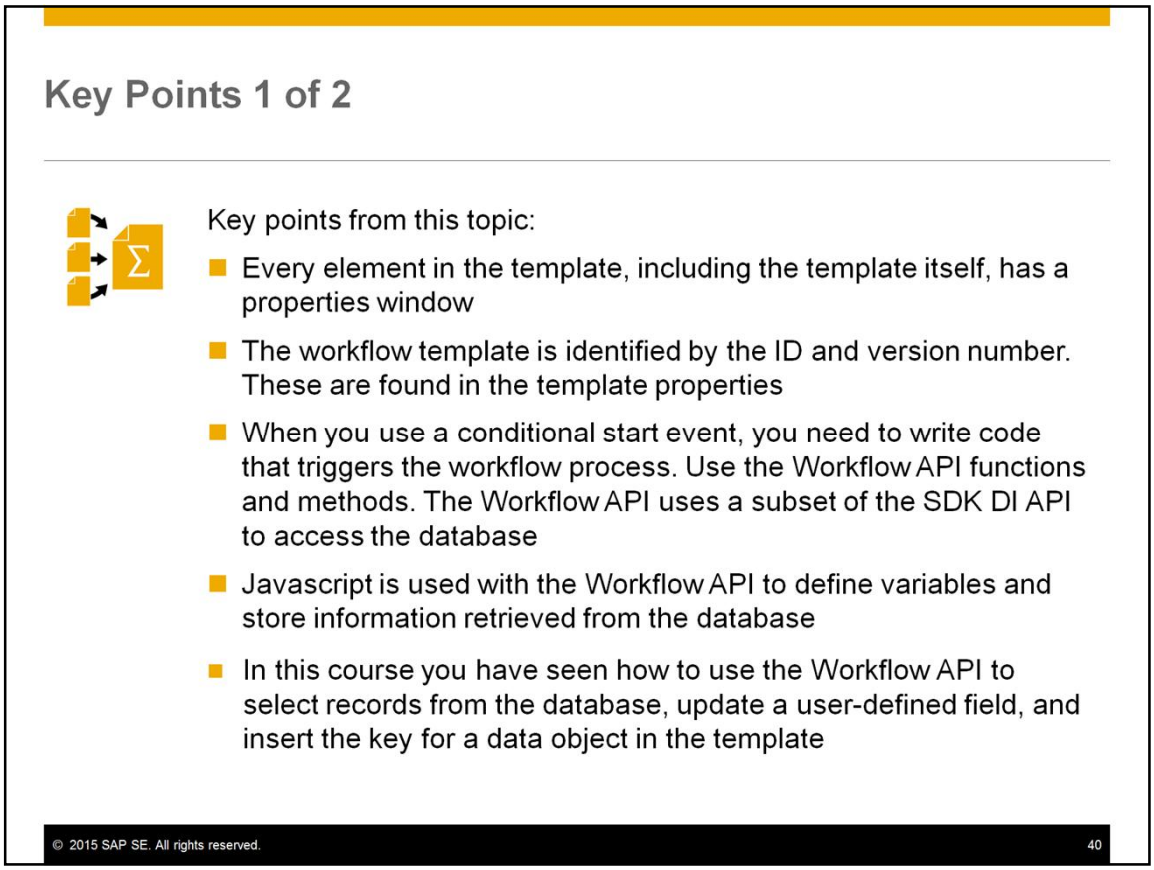
File path: ***Program Files\SAP\SAP Business One Server Tools\Workflow***



The workflow configuration file is an xml file located in the Workflow folder under the SAP Business One Server Tools directory.

You can edit this file to change various settings. Some of these settings are described here.

- The monitor interval for conditional events is by default setting 10 minutes. You can change this to a shorter or longer interval, depending on business requirements.
- The log file size is set to 5 (5 megabytes). When this size is reached, the system will start a new log file.
- The log parameters control the type of messages appearing in the log:
  - If the *Debug* parameter is set to 1, debugging messages will appear in the log.
  - If the *Warn* parameter is set to 1, the log will include all warning messages from Java. This can make the log very long.
  - If the *Err* parameter is set to 1 the log will include error messages.
  - If the *Info* parameter is set to 1, the output from the Javascript *print()* function will appear in the log.



Here are some key points to take away from this session.

- Every element in the template, including the template itself, has a properties window
- The workflow template is identified by the ID and version number. These are found in the template properties
- When you use a conditional start event, you need to write code that triggers the workflow process. Use the Workflow API functions and methods. The Workflow API uses a subset of the SDK DI API to access the database
- Javascript is used with the Workflow API to define variables and store information retrieved from the database
- In this course you have seen how to use the Workflow API methods to select records from the database, update a user-defined field, and insert the key for a data object in the template. Be sure to use the Workflow API Reference guide for information about methods not covered in this course.

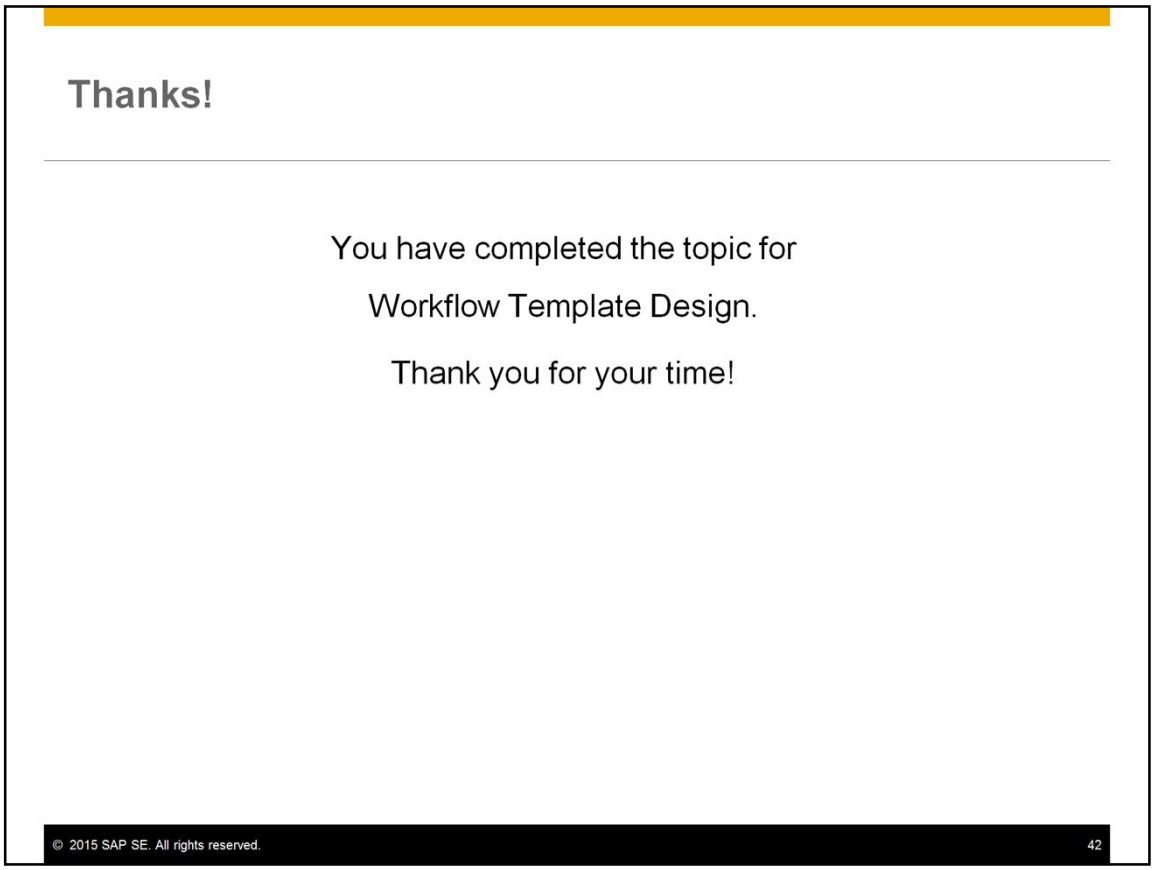
Key Points 2 of 2

Key points from this topic:

- If the input data object for a task is based on the output object from another task, select *Base On* in the task properties
- You can easily build the condition for a gateway sequence flow using the *Flow Condition* editor and selecting object, object property, condition and value
- In release 9.1, you have additional options:
  - You can specify a team as participant for a task. Team membership defined in the employee master data
  - You can select the manager or subordinates of the owner of another task in the template. This is useful for adding approval sequences in a workflow process
  - You can define a start parameter for the workflow instance

- If the input data object for a task is based on the output object from another task, select *Base On* in the task properties
- You can easily build the condition for a gateway sequence flow using the *Flow Condition* editor and selecting object, object property, condition and value
- In release 9.1, you have additional options:
  - You can specify a team as participant for a task. Team membership defined in the employee master data
  - You can select the manager or subordinates of the owner of another task in the template. This is useful for adding approval sequences in a workflow process

  - You can define a start parameter for the workflow instance. For a conditional template the parameter could be the key to a record from the database, or for a manual template you can define a variable that prompts for a document number or date.



You have completed the topic for Workflow Template Design. Thank you for your time!

© 2015 SAP SE. All rights reserved.