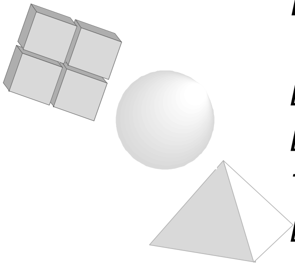
Chapter 2 Business Process Automation Development



Business Automation Workspace, BPA editors
Business Organization Model
The Business Process Model
Expression steps
Process flow elements
Cyclic flows
Consistency checks
Integration with other workspaces
Test mode

2 Business Process Automation Development

The Business Process Automation (BPA) is one of the workspaces of the UNIFACE eight Development Environment (UDE). Here you can define a workflow management system in UNIFACE. A workflow management system manages a business process that executes your business. For each business process, a number of tasks must be performed. The order in which the tasks are performed is managed by the workflow management system.

To execute a task, one or more application components can be used. For example when the user enters an order, he needs some application components to enter data for this order. Another option is that the use does not need an application component. An example of this kind of tasks is ‘call customer’. It depends on the type of task wether or not an action of a user is needed. A batch task, such as for example ‘credit check’, does not need end-user interaction. Although for a batch task some application components are needed.

During BPA development you enter all the definitions needed to build this workflow management system into UNIFACE and store definition data in a development workflow repository.

2.1 Business Automation Workspace, BPA editors

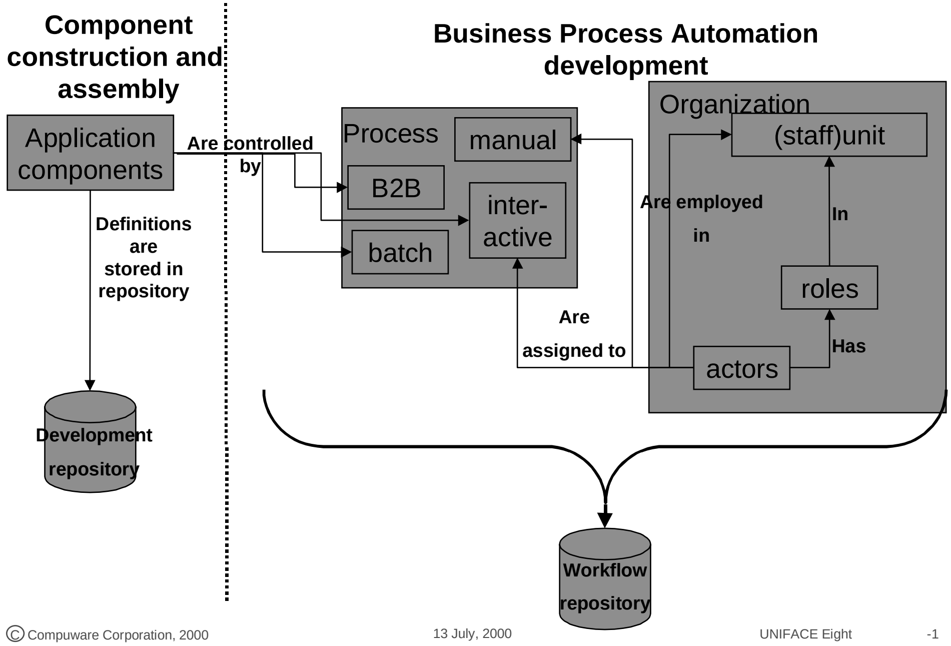
One of the development workspaces of UNIFACE eight is the Business Automation Workspace. From this workspace you can start the Business Engineer Workbench (BEW). A number of BPA editors are available in this workbench to define all objects needed for your models.

The following models are defined:

- Organization model: A representation of the organization.
- Process model: A representation of the task sequence.

One organization can have multiple business processes. The integration of one business process with an organization model defines the whole BPA process.

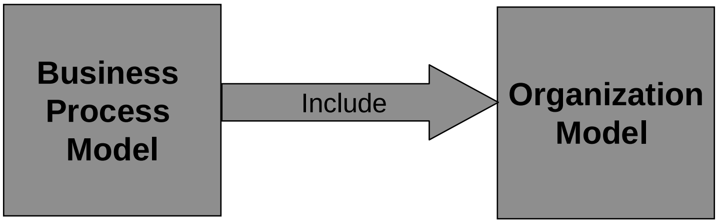
UNIFACE V8



© Compuware Corporation, 2000 13 July, 2000 UNIFACE Eight -1

UNIFACE V8

Business Process Automation Development



© Compuware Corporation, 2000 13 July, 2000 UNIFACE Eight -2

2.2 Business Organization Model

The organization model is the organization chart of the organization. The model holds the logical grouping of units, including the organization roles and actors who fill those roles.

The container for the organization model is a non-versioned package. This package includes units, staff units, roles, actors and business constants. The package of units is represented in graphical way in the modeler of the Business Engineer Workbench (BEW).

You can define packages that contain groups of similar objects.

The tool bar of the graphical modeler contains a number of buttons that allow you to create comments, units and unit structures for an organization model.

2.2.1 Non-versioned Package properties

The term ‘non-versioned’ comes from the fact that during deployment there can be only one package that represents the organization.

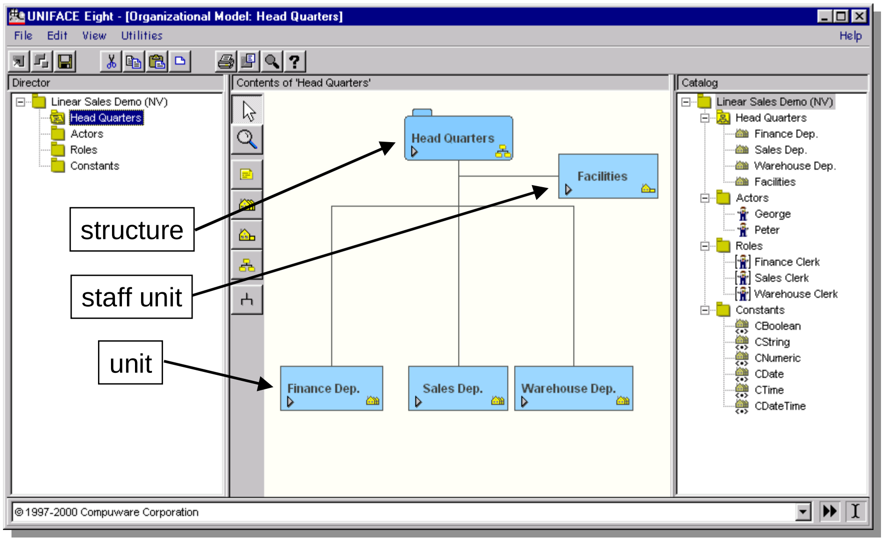
The properties to be defined, for the non-versioned package, are related to naming, description and comments. A non-versioned package can be included by a versioned package. A versioned package describes the business process. The inclusion of an non-versioned package by a versioned package accomplishes the integration of business model and organization model.

2.2.2 Unit and structure properties

Within the organization structure units can be defined as staff-unit, for supportive activities, or as units in the hierarchy of the organization.

Properties to be defined for an organization unit are related to naming, hierarchy and roles. On the roles tab of the properties form you select one or more roles. For each role you must assign one or more actors.

Model the Organization



UNIFACE Eight - [Organizational Model: Head Quarters]

Director

- Linear Sales Demo (NV)
  - Head Quarters
  - Actors
  - Roles
  - Constants

Contents of 'Head Quarters'

- Head Quarters
  - Facilities
  - Finance Dep.
  - Sales Dep.
  - Warehouse Dep.

Catalog

- Linear Sales Demo (NV)
  - Head Quarters
    - Finance Dep.
    - Sales Dep.
    - Warehouse Dep.
    - Facilities
  - Actors
    - George
    - Peter
  - Roles
    - Finance Clerk
    - Sales Clerk
    - Warehouse Clerk
  - Constants
    - CBoolean
    - CString
    - CNumeric
    - CDate
    - CTime
    - CDateTime

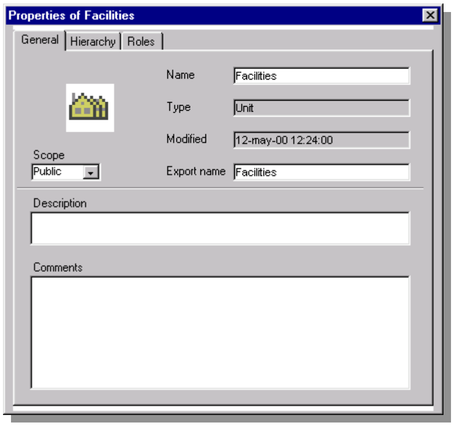
©1997-2000 Compuware Corporation

structure

staff unit

unit

Structure and (staff) unit properties



Properties of Facilities

Name: Facilities

Type: Unit

Modified: 12-may-00 12:24:00

Scope: Public

Export name: Facilities

Description

Comments

2.2.3 Actors and roles

An actor is a person that performs the work represented by a task. This work is presented as one or more tasks, assigned to the actor via the tasklist. During deployment the tasks for each actor are displayed in this tasklist, the Task Desktop. Each actor has his own Task Desktop.

Actors have roles in units.

A role represent a function that supports the business process within the organization

An actor is assigned a task by explicitly associating the actor with an existing role. Assigning a task can be done through the unit properties form, but also through the actor properties form.

Actors and roles are defined separately and each actor can have a number of roles. An actor, assigned a certain role, can be assigned to perform a task within the business process.

Actors and roles are part of the organization. Therefore they are defined within the context of a non-versioned package.

To have large sets of actors or roles conveniently arranged, a package can be defined for roles and actors.

2.2.4 Actor and role properties

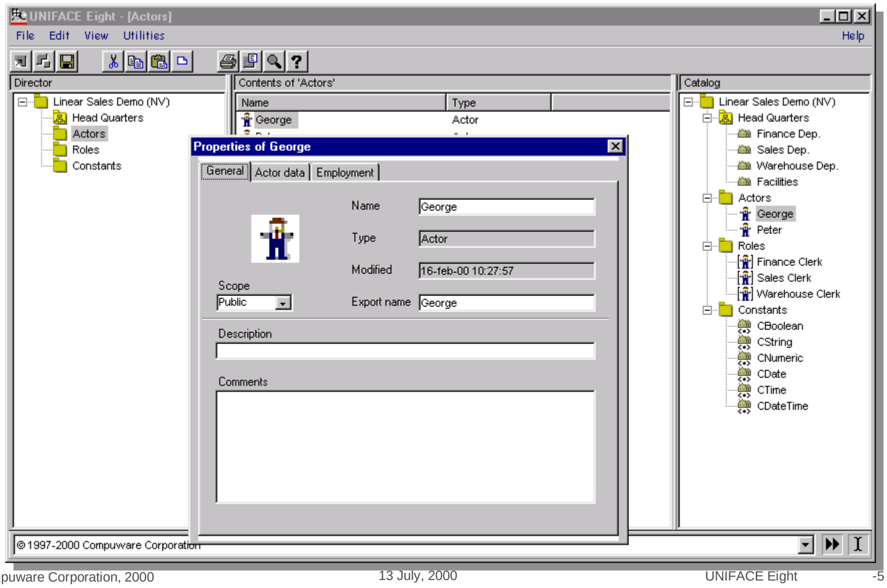
For actors and roles a number of properties can be defined. Roles, for example, can be ‘manager’, ‘sales clerk’. The main property for a role is the name of the role.

For an actor you must define a set of properties:

- General, where you can define a proper name for the actor.
- Data related with the actor, for example a password can be set for an actor. When an actor logs on during deployment he has to use the proper actor Name and Password combination.
- Employment property, you can assign one or more organization units to an actor. You also assign one or more roles to the actor as an implementation of the actor's responsibility in the business he is supporting.

UNIFACE V8

Actors



Director
- Linear Sales Demo (NV)
  - Head Quarters
  - Actors
  - Roles
  - Constants

Contents of 'Actors'

Name	Type
George	Actor

Properties of George

General | Actor data | Employment

Name: George

Type: Actor

Modified: 16-feb-00 10:27:57

Scope: Public

Export name: George

Description

Comments

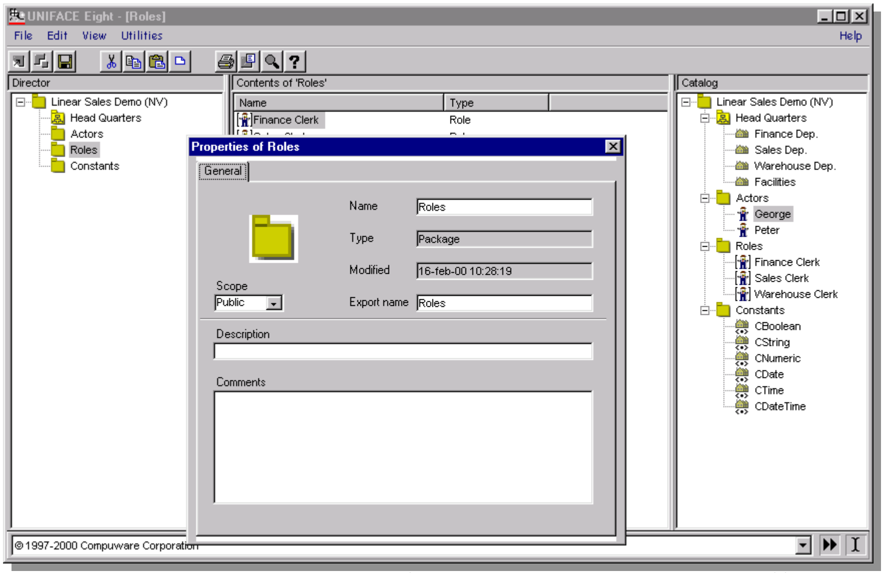
Catalog
- Linear Sales Demo (NV)
  - Head Quarters
    - Finance Dep.
    - Sales Dep.
    - Warehouse Dep.
    - Facilities
  - Actors
    - George
    - Peter
  - Roles
    - Finance Clerk
    - Sales Clerk
    - Warehouse Clerk
  - Constants
    - CBoolean
    - CString
    - CNumeric
    - CDate
    - CTime
    - CDateTime

©1997-2000 Compuware Corporation

© Compuware Corporation, 2000 13 July, 2000 UNIFACE Eight -5

UNIFACE V8

Roles



Director
- Linear Sales Demo (NV)
  - Head Quarters
  - Actors
  - Roles
  - Constants

Contents of 'Roles'

Name	Type
Finance Clerk	Role

Properties of Roles

General

Name: Roles

Type: Package

Modified: 16-feb-00 10:28:19

Scope: Public

Export name: Roles

Description

Comments

Catalog
- Linear Sales Demo (NV)
  - Head Quarters
    - Finance Dep.
    - Sales Dep.
    - Warehouse Dep.
    - Facilities
  - Actors
    - George
    - Peter
  - Roles
    - Finance Clerk
    - Sales Clerk
    - Warehouse Clerk
  - Constants
    - CBoolean
    - CString
    - CNumeric
    - CDate
    - CTime
    - CDateTime

©1997-2000 Compuware Corporation

© Compuware Corporation, 2000 13 July, 2000 UNIFACE Eight -6

2.2.5 Business constants

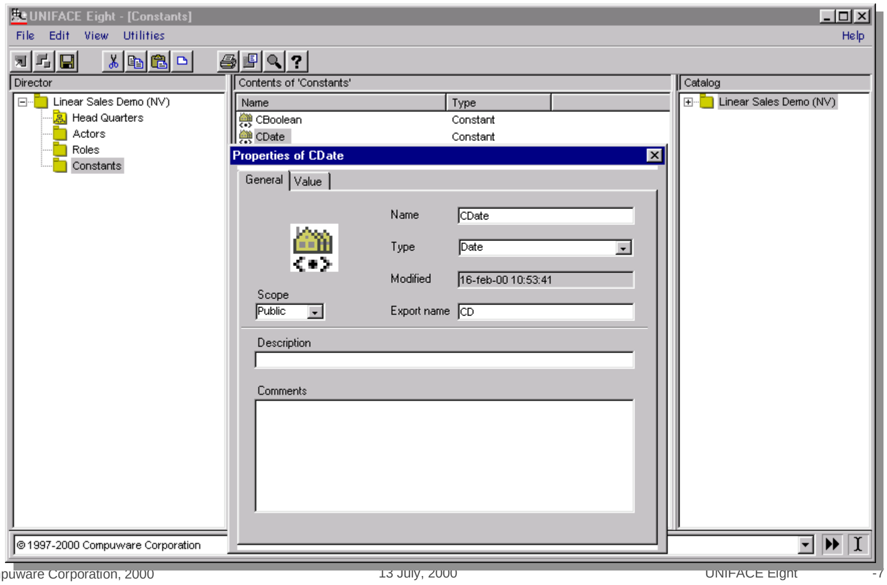
A business constant is a definition of a value that can be used in expressions. Expressions are explained in the section about the business process model.

Business constants have values that do not change very often. However, when changed, these values need to be changed at once for all occurrences throughout a process. For example, a business constant could be the percentage of VAT that applies to certain kinds of transactions and needs to be changed only rarely.

Business constants are non-versioned model objects, so if the value of a business constant is changed, it applies to all versions of the process model in which it is used.

UNIFACE V8

Business constants



Director:

- Linear Sales Demo (NV)
  - Head Quarters
  - Actors
  - Roles
  - Constants

Contents of 'Constants'

Name	Type
CBoolean	Constant
CDate	Constant

Properties of CDate

Name: CDate

Type: Date

Modified: 16-feb-00 10:53:41

Scope: Public

Export name: CD

Description

Comments

Catalog: Linear Sales Demo (NV)

©1997-2000 Compuware Corporation

© Compuware Corporation, 2000 13 July, 2000 UNIFACE Eight -7

2.3 The Business Process Model

A business process is a sequence of tasks that support a business to achieve a business goal. The model to represent this business process is called the Business process model or the ‘process model’.

For each process model a number of tasks must be defined. The tasks must be placed in a certain sequence to give a representation of the Business Process. In the deployment phase the process flow can be rearranged by modifying the task sequence of the process model.

Each well-defined process model helps a business run in an optimized way. The process model is used to define the run time execution of the workflow system. UNIFACE does not have any limitations on the number of processes that can be stored in the workflow repository.

2.3.1 Model the business process

Within a versioned package a process can be defined. A process is similar to a folder with a number of tasks. Properties for this ‘Process’, like the implementation of naming conventions, can be set.

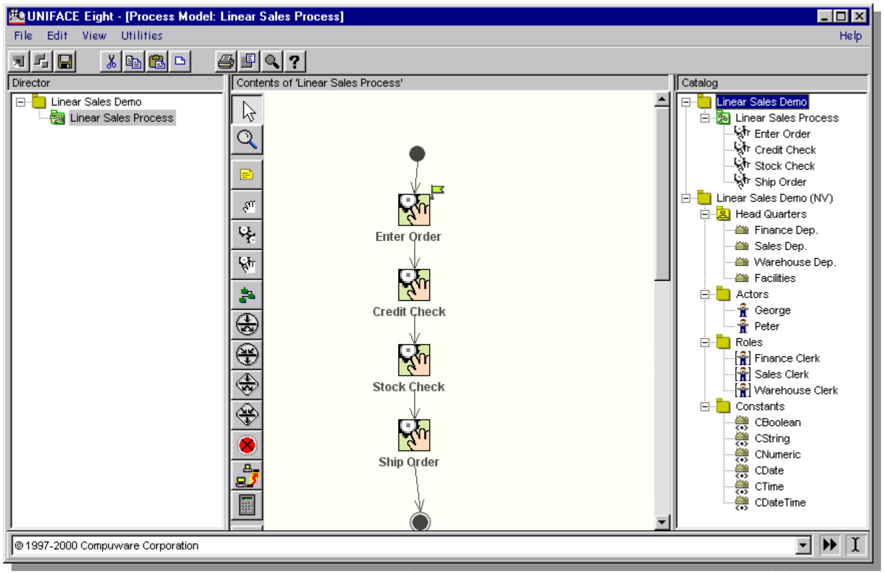
When a new Process is created in the graphical modeler the start and end symbol of the process are automatically displayed. There can be one process start and one formal end for each Process. To end a process in a different way, you can add an abort step.

An export / import functionality allows the development workflow repository to be copied to the deployment workflow repository.

During deployment there can be more than one version of the process model. Defining a process model starts with the definition of a versioned package. This package is used as the container for the process model.

UNIFACE V8

Model the business process



© Compuware Corporation, 2000 13 July, 2000 UNIFACE Eight -9

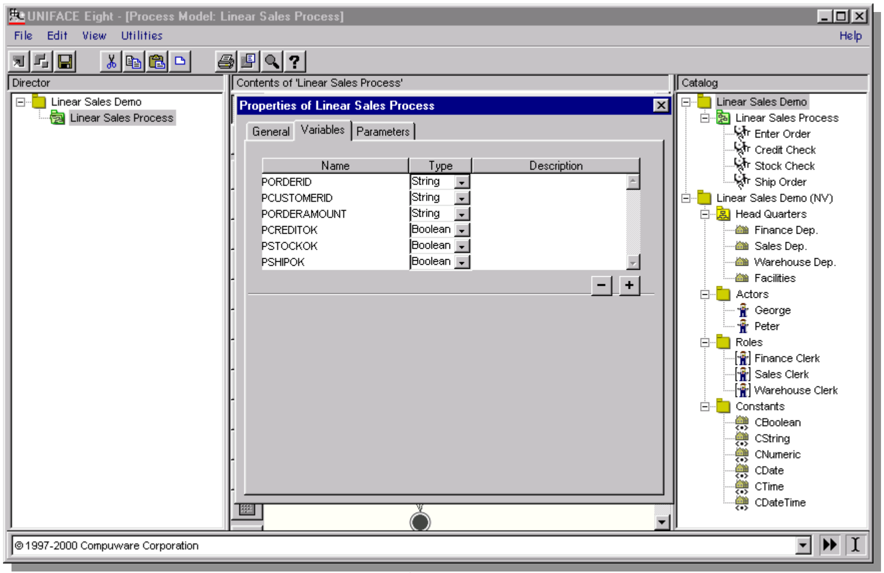
2.3.2 Process properties

On the properties form parameters and variables can be defined for the process. Process variables are needed to support the whole business process and enable passing of data from one task to another. Data that is stored in process variables is available to the process and to all tasks in the process (via mapping of parameter values to business variables).

A process that is embedded within another process does not know about variables from its parent, and the parent process does not know about the variables of its children. Process variables are restricted to the process in which they are declared. Process variables that are mapped to process parameters enable communication to and from another process. Parameters can be string, numeric or boolean data types.

Data that belongs to a parent process and are needed by a child process must be passed to the child process using input parameters. Similarly, data stored in variables of the child process, needed by the parent process must be passed using output parameters of the process.

Process properties


UNIFACE Eight - [Process Model: Linear Sales Process]
File
Edit
View
Utilities
Help
Director
Contents of 'Linear Sales Process'
Catalog
General
Variables
Parameters
−
+

Director:
- Linear Sales Demo
  - Linear Sales Process

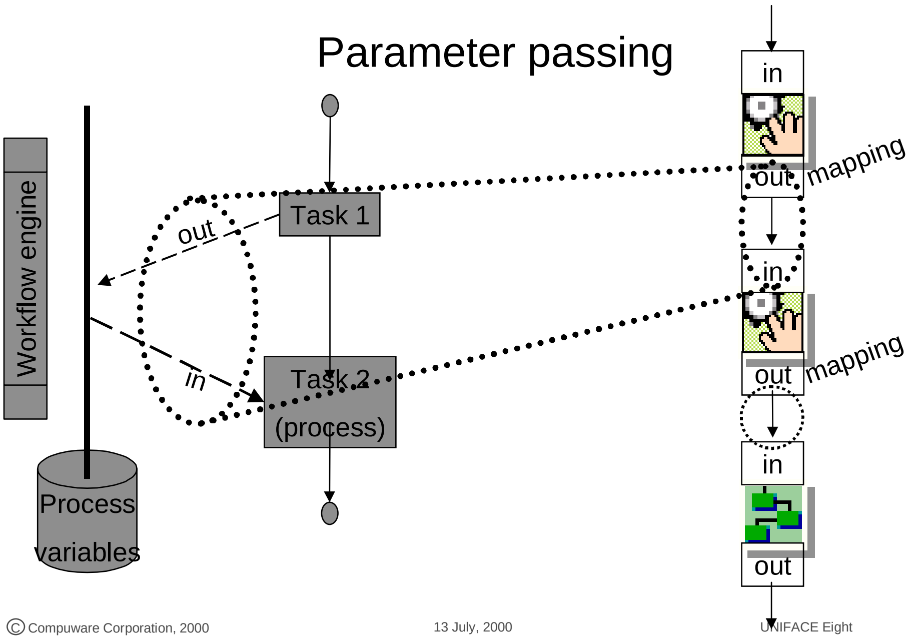
Properties of Linear Sales Process

Name	Type	Description
PORDERID	String	
PCUSTOMERID	String	
PORDERAMOUNT	String	
PCREDITOK	Boolean	
PSTOCKOK	Boolean	
PSHIPOK	Boolean	

Catalog:
- Linear Sales Demo
  - Linear Sales Process
    - Enter Order
    - Credit Check
    - Stock Check
    - Ship Order
- Linear Sales Demo (NV)
  - Head Quarters
    - Finance Dep.
    - Sales Dep.
    - Warehouse Dep.
    - Facilities
  - Actors
    - George
    - Peter
  - Roles
    - Finance Clerk
    - Sales Clerk
    - Warehouse Clerk
  - Constants
    - CBoolean
    - CString
    - CNumeric
    - CDate
    - CTime
    - CDateTime

©1997-2000 Compuware Corporation

Parameter passing



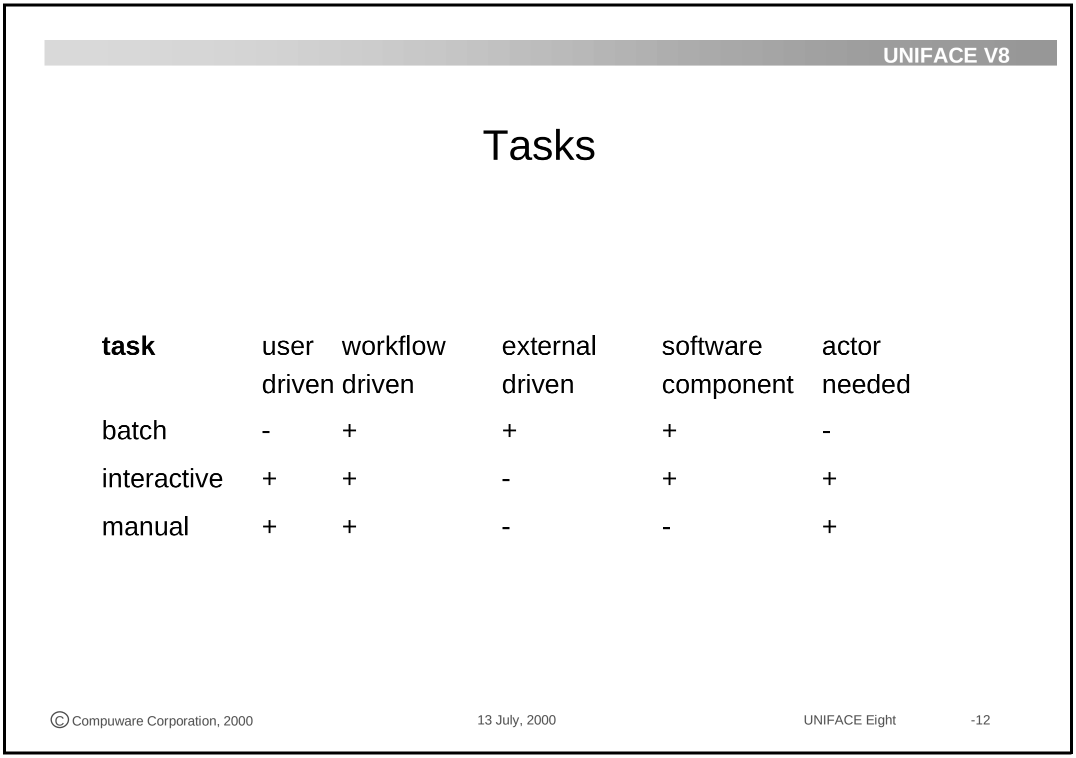
2.3.3 Tasks

A task is a piece of work that forms one logical step within a business process. A task is the lowest granularity of the business process and they are defined on the operational level of the business process. For example, using a bottom up approach, the application component used to execute a task can be used to define the granularity of a task.

Fulfilling a task can be a manual exercise, in which case user intervention is needed. A task can be like a batch job, where no user intervention is necessary.

Tasks can be:

- User-driven: starting this task is determined by outside event.
- Workflow-driven: the decision to start this task is determined by the Workflow Engine.
- External-driven: the decision to start a task is determined by an external event.



UNIFACE V8

Tasks

task	user driven	workflow driven	external driven	software component	actor needed
batch	-	+	+	+	-
interactive	+	+	-	+	+
manual	+	+	-	-	+

© Compuware Corporation, 2000 13 July, 2000 UNIFACE Eight -12

2.3.4 Task types

A business process consists out of a number of tasks in a certain sequence. For example the task ‘Enter Order’ can be the first step in a business process. The next step can be a ‘Credit Check’ followed by a ‘Stock Check’ and ‘Shipment’ of the order. Some tasks need a software implementation, some tasks do not. To enter an order for example you need a software component that allows you to enter all data needed for an order. On the other hand a credit check can be done by calling the bank and ask for the solvency of the customer. You do not need a software component for that task.

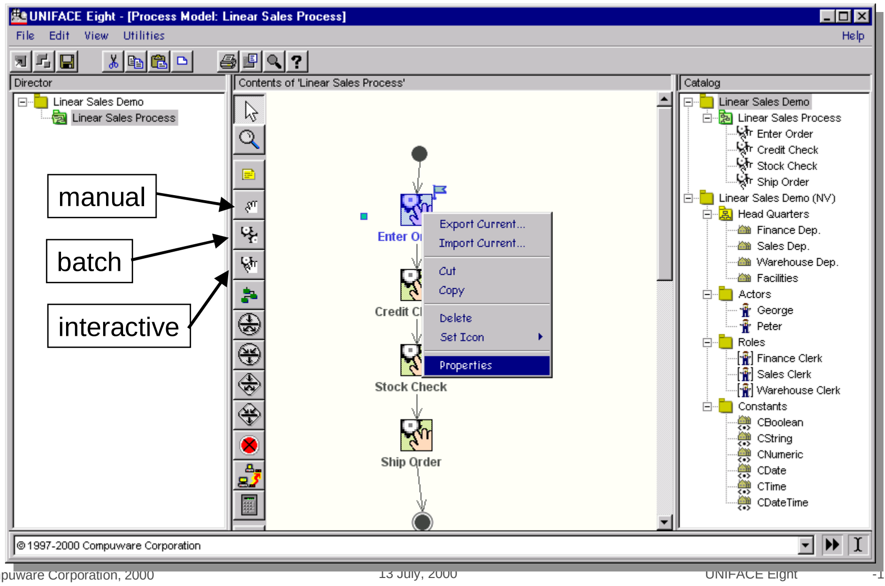
The following tasks can be used:

- A manual task has no software implementation other than its interface. A manual task is used to update the process variables. A typical manual task could be 'Call customer'.
- An interactive task is implemented as a component or a set of components, which are executed by a component engine. An actor is needed to start the application for the execution of the task. An example of this type of task is 'Enter sales order'.
- A batch task is implemented by a (set of) component(s) without user interaction. It is started by a previous task or by an external task. A ‘Check stock’ can be a batch task.

Use the tool bar of the graphical modeling window to add tasks to the compound process.

A set of tasks can be put in a new compound process to make it more manageable.

Task types

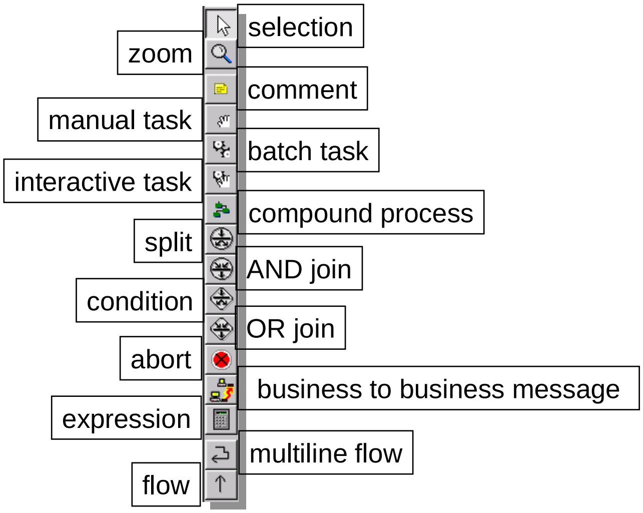


manual

batch

interactive

Tools for modeling the business process



selection

zoom

comment

manual task

batch task

interactive task

compound process

split

AND join

condition

OR join

abort

business to business message

expression

multiline flow

flow

2.3.5 Task properties

Task assignment

Before doing a task assignment, the non-versioned package must be made known to the versioned package. This is done via the include facility of the business process properties. There you can select a non-versioned package.

A task must be assigned according the assignment rules.

A task assignment is a property of each manual or interactive task in a process. A task can be assigned to:

- a particular actor
- all actors in a unit
- actors in a given role
- an actor that performed another task earlier in the process, and so on.
- an actor(2) with Role in Hierarchy of Actor(1) who performed a Task.

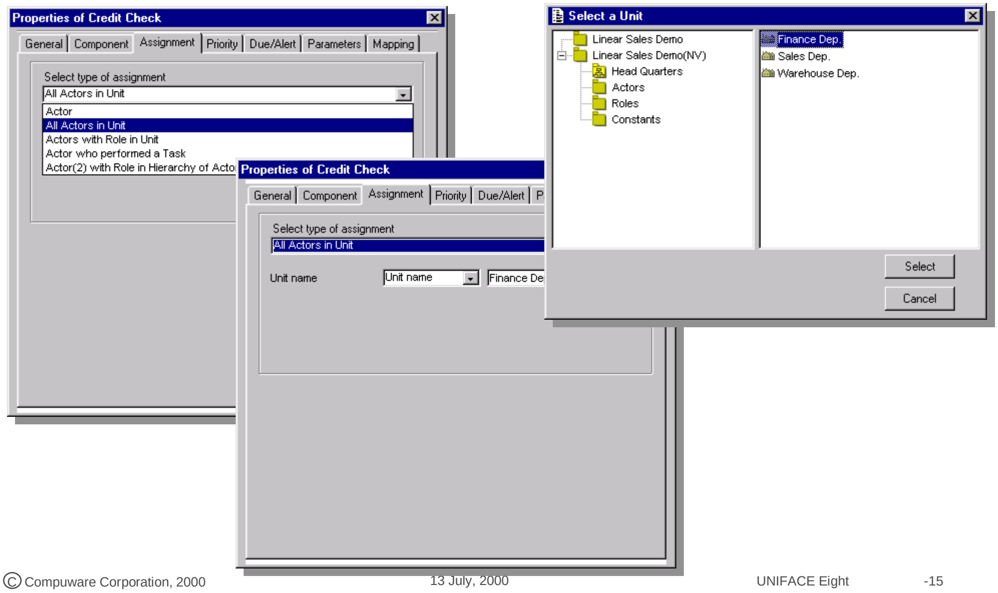
All assignments can be replaced by process variables that represents the assignments. During deployment these process variables will have the value needed to assign the task.

For a batch task, there is no task assignment needed, because there is no person involved in the execution.

At run time, the task assignment determines to which actor the Workflow Engine should assign the task. For user-driven tasks, the assignment determines who can initiate the task from their Start New Task menu or drop-down list in the Task Desktop.

UNIFACE V8

Task properties, assignment (1)



Properties of Credit Check

Select type of assignment: All Actors in Unit

- Actor
- All Actors in Unit
- Actors with Role in Unit
- Actor who performed a Task
- Actor(2) with Role in Hierarchy of Acto

Properties of Credit Check

Select type of assignment: All Actors in Unit

Unit name: Unit name | Finance De

Select a Unit

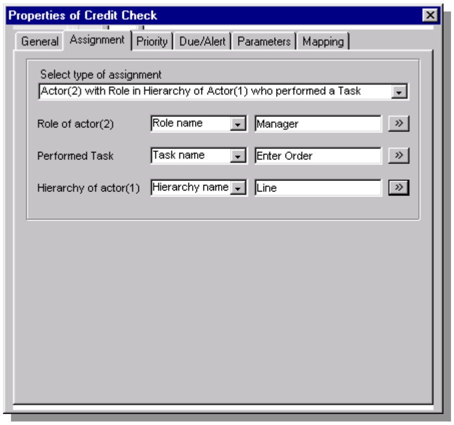
- Linear Sales Demo
- Linear Sales Demo(NV)
  - Head Quarters
  - Actors
  - Roles
  - Constants

- Finance Dep.
- Sales Dep.
- Warehouse Dep.

© Compuware Corporation, 2000 13 July, 2000 UNIFACE Eight -15

UNIFACE V8

Task properties, assignment (2)



Properties of Credit Check

Select type of assignment: Actor(2) with Role in Hierarchy of Actor(1) who performed a Task

Role of actor(2)	Role name	Manager
Performed Task	Task name	Enter Order
Hierarchy of actor(1)	Hierarchy name	Line

© Compuware Corporation, 2000 13 July, 2000 UNIFACE Eight -16

Parameters, mapping

Parameters contain the data that is needed or returned by the component that implements the task.

For each task a number of parameters and mappings must be defined.

To pass business variable values to and from a task, you have to assign the process variables to input and output parameters of the tasks.

Specify only that particular parameter to be paired with or mapped to one of the variables of the parent process. A parameter can also be mapped to a combination of process variables, system variables and operators.

This means that, at deployment, the Workflow Engine fills an input or output parameter with the value of the variable or expression to which the parameter has been mapped.

This association of a parameter with a (parent) process variable is done declaratively by using the Mapping property form for the task or sub-process.

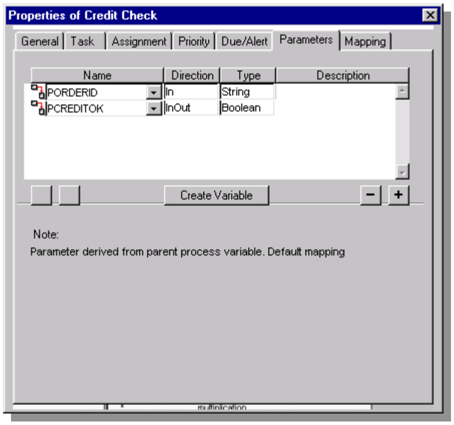
Matching

A user-driven task that continues or initiates the flow of a process must be matched with the correct instance of the process.

It does this by matching the value of a user-driven task parameter against an expression or an entered value of a process variable before process instance can continue.

For example, a Payment Entry task has the requirement that the invoice number and customer ID from a received payment should match the invoice number and customer ID of the original invoice issued to the customer.

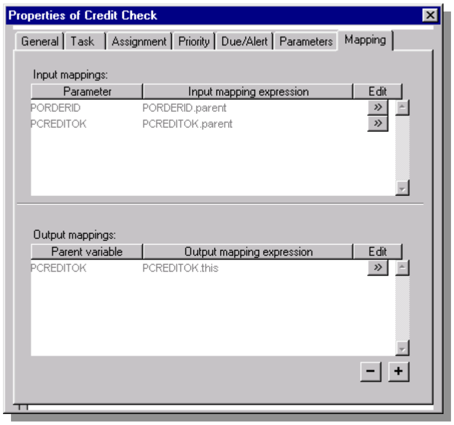
Task properties, mapping



Properties of Credit Check

Name	Direction	Type	Description
PORDERID	In	String	
PCREDITOK	InOut	Boolean	

Note:
Parameter derived from parent process variable. Default mapping



Properties of Credit Check

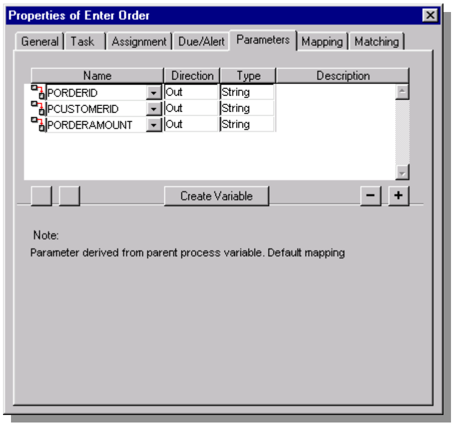
Input mappings:

Parameter	Input mapping expression	Edit
PORDERID	PORDERID.parent	»
PCREDITOK	PCREDITOK.parent	»

Output mappings:

Parent variable	Output mapping expression	Edit
PCREDITOK	PCREDITOK.this	»

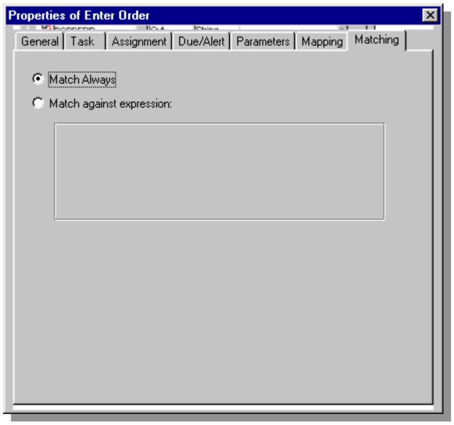
Task properties, matching



Properties of Enter Order

Name	Direction	Type	Description
PORDERID	Out	String	
PCUSTOMERID	Out	String	
PORDERAMOUNT	Out	String	

Note:
Parameter derived from parent process variable. Default mapping



Properties of Enter Order

● Match Always
○ Match against expression:

2.3.6 Business to Business messages

A Business to Business (B2B) message gives you the opportunity to integrate the workflow management system with the outside world.

A B2B message can be considered a special case of a batch task, because they behave very similarly - a sending B2B message behaves like a workflow-driven batch task, and a receiving B2B message behaves like a user-driven task

Information can be sent to and received from customers or suppliers. For example you can send an order confirmation message to your customers.

The integration is done by XML based messages. A generic DTD is supplied, which packs any series of parameters that is entered on the parameter tab page, the 'DTD of Envelope'. A specific DTD is used to explain the user data, the 'DTD of Body'.

A B2B message must be set as 'Receive' or 'Send'. B2B messages should contain the name of a DTD that is both available for the sender and the receiver. The DTD for the Envelope can have the value 'Biztalk' or a user defined name.

When a business message is sent the firstly found process instance that has a matching step for it is bound to it.

HTTP and HTTPS are the supported protocols for B2B messaging.

Send message

Send messages can be defined for a local destination or a remote destination. They are implemented as workflow driven tasks.

For a send message the Target Address must always be entered. If the message has a local destination the Message Identifier must be available. The identifiers is used by the Receive message to identify the message. The message identifier must be unique. Local messages can be used for communication between different business processes in the same organization.

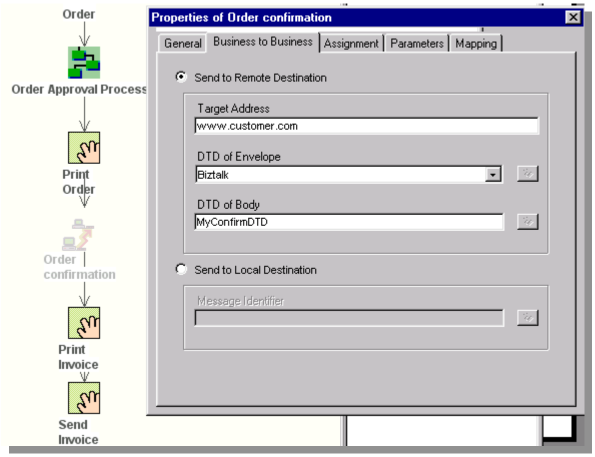
Send message uses IN parameters.

Receive message

For the receive message the target address is disabled. A receive message has been implemented as a completion driven task.

The Receive message only has OUT parameters.

Business to Business messages



Properties of Order confirmation

General | Business to Business | Assignment | Parameters | Mapping

● Send to Remote Destination

Target Address: www.customer.com

DTD of Envelope: Biztalk

DTD of Body: MyConfirmDTD

○ Send to Local Destination

Message Identifier:

2.3.7 Expression editor

The BPA expression editor is available in the BPA Workbench. In process modeling it enables you to define expressions and conditions to be used for mapping, matching, assignment rules, conditions, and expression steps.

The BPA expression editor allows you to:

- Choose from a list of process variables, task parameters, constants and functions to create an expression that can be used for mapping and expression steps.
- Build an expression with relational operators to evaluate a condition, used in a condition step and in matching of initiation and continuation tasks.

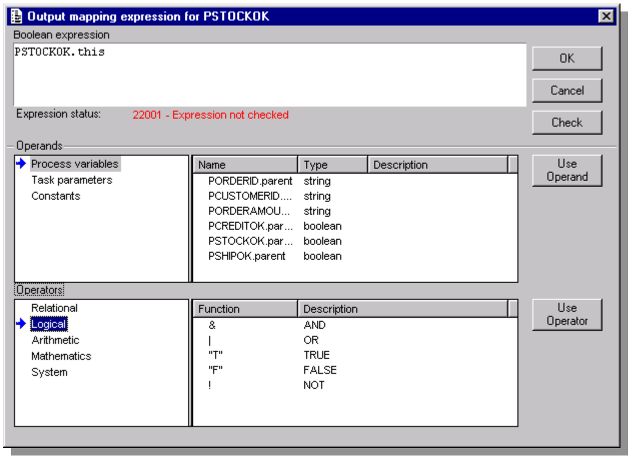
The BPA expression editor form displays the following areas:

- An expression area, where the expression is built, through a combination of your direct entry and your usage of the lists of operands and operators
- An operand area, that lists the available types of operands and, on right-hand side, the list of operands for the chosen operand type.
- An operator area, that lists the available types of operators and, on right-hand side, the list of operators for the chosen operator type.

The expression editor, for example, can be used to calculate the selling-price including VAT.

UNIFACE V8

Expression editor



Output mapping expression for PSTOCKOK

Boolean expression

```
PSTOCKOK.this
```

Expression status: 22001 - Expression not checked

Operands

- Process variables
- Task parameters
- Constants

Name	Type	Description
PORDERID.parent	string	
PCUSTOMERID....	string	
PORDERAMOU...	string	
PCREDITOK.par...	boolean	
PSTOCKOK.par...	boolean	
PSHIPOK.parent	boolean	

Operators

- Relational
- Logical
- Arithmetic
- Mathematics
- System

Function	Description
&	AND
|	OR
"T"	TRUE
"F"	FALSE
!	NOT

© Compuware Corporation, 2000 13 July, 2000 UNIFACE Eight -20

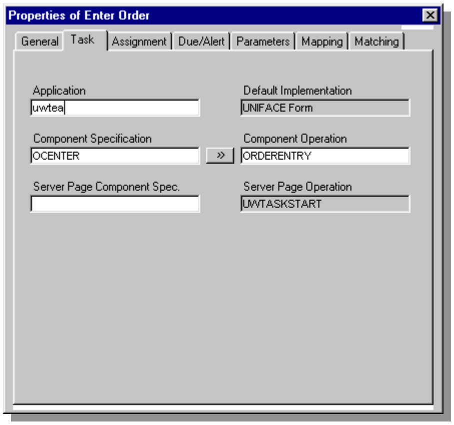
2.3.8 Software implementation

For batch and interactive tasks, application components specifications must be defined as one of the properties. Operation, component and application name that are associated with the execution of the task must be defined on the properties form of these tasks. For example when the task ‘Enter Order’ is executed via the operation ‘ORDERENTRY’ and startup shell ‘uwtea’ activates this operation on component ‘ocenter’; this information must be defined on the properties form of task ‘Enter Order’.

The operation ‘ORDERENTRY’ on the component ‘ocenter’ will be activated via the Task Desktop of an actor who is responsible for the execution of this task.

UNIFACE V8

Batch and interactive tasks, software implementation



Application: uwtea

Default Implementation: UNIFACE Form

Component Specification: OCENTER

Component Operation: ORDERENTRY

Server Page Component Spec.:

Server Page Operation: UWTASKSTART

© Compuware Corporation, 2000 13 July, 2000 UNIFACE Eight -21

2.3.9 Priority

A workflow-driven task can be assigned a priority, from 1 (highest) to 5 (lowest), that controls where a task instance is placed in the task list of the person or persons assigned to perform the task. The declaration of the priority property is made on the Task Properties form. It can be changed dynamically, that is, at deployment time, via the evaluation of a priority expression or priority condition, or by the task list manager explicit reassignment of the task priority.

For example you can set a higher priority for the ‘credit check’ if the order amount exceeds $100,000.

2.3.10 Due date and alert

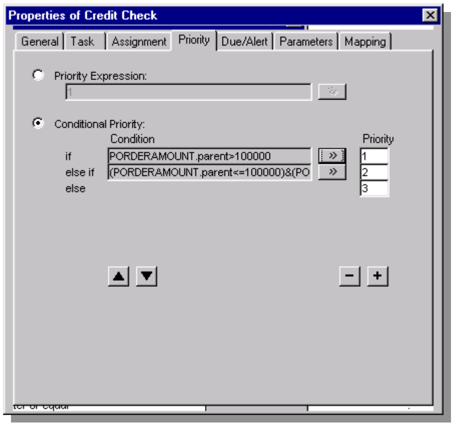
Due dates and alerts are separate, optional properties of an interactive or manual task that are used by the Workflow Engine to calculate when a task is due and when an alert should be raised.

If enabled, the due and alert dates are used to calculate the Due Time and Alert Time, and to raise the flags (icons) that are displayed on the actor's and manager's Task Desktops.

If an alert is enabled, the task can also be assigned to other actors (All actors in Unit, Actors with Role in Unit, and so on), in addition to the primary actor assigned to the task.

In this way the task is escalated to the responsible manager. He can than take appropriate action.

Priority

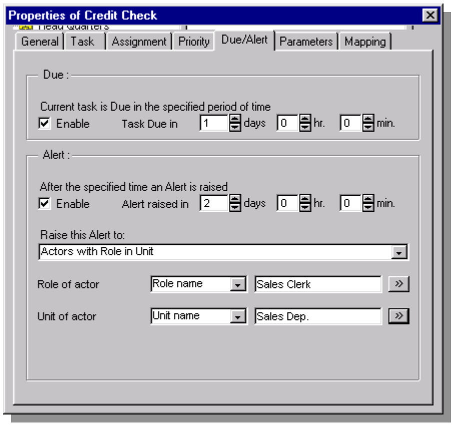


○ Priority Expression:
1

● Conditional Priority:

	Condition		Priority
if	PORDERAMOUNT.parent>100000	»	1
else if	(PORDERAMOUNT.parent<=100000)&(PO	»	2
else			3

Due date and alert



Due :

Current task is Due in the specified period of time

☑ Enable Task Due in 1 days 0 hr. 0 min.

Alert :

After the specified time an Alert is raised

☑ Enable Alert raised in 2 days 0 hr. 0 min.

Raise this Alert to:
Actors with Role in Unit

Role of actor Role name Sales Clerk »

Unit of actor Unit name Sales Dep. »

2.4 Expression steps

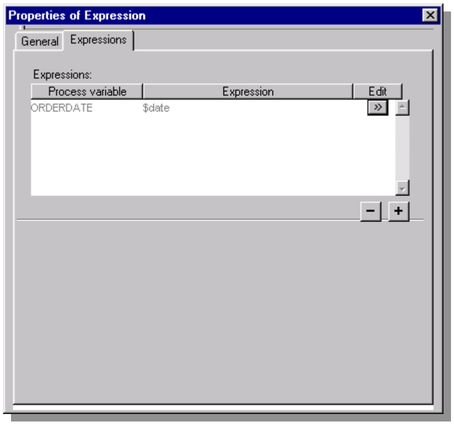
A expression step can be inserted in the flow of a process model to set the value of one or more process variables or process parameters.

You can define expressions to assign values to process variables and process parameters, exactly as you define the out mappings of a task or child process.

For example you can use expression steps to implement a counter in a cyclic process or to set an order date.

UNIFACE V8

Expression step



Expressions:

Process variable	Expression	Edit
ORDERDATE	$date	»

13 July, 2000 UNIFACE Eight -25

2.5 Process flow elements

Flow elements interconnect ‘steps’ within a process. (A step is a collective term used when discussing all parts of a process in a non-specific way, split, AND-join, condition OR-join, abort)

Simple flow elements connect directly from one task to another, depicted in the model diagram by a uni-directional arrow. For more complex flows a set of logic flow elements is available.

2.5.1 Splits

AND-split

An AND-split is a process graph element with one incoming and two or more outgoing flows. As soon as the incoming flow is enabled, both outgoing flows are enabled. The single flow before the AND-split becomes a parallel flow after the AND-split.

An AND split will always be followed by an AND join.

Condition

Conditions are specific constraints that a business has decided to make when faced with particular circumstances. For example, a condition can be that the business does not issue a purchase order if the value of that order does not exceed a specific monetary amount. This would be implemented in the process model using a condition that checks the value of the order to make sure it is above the set level. If it passes the check, the order process continues.

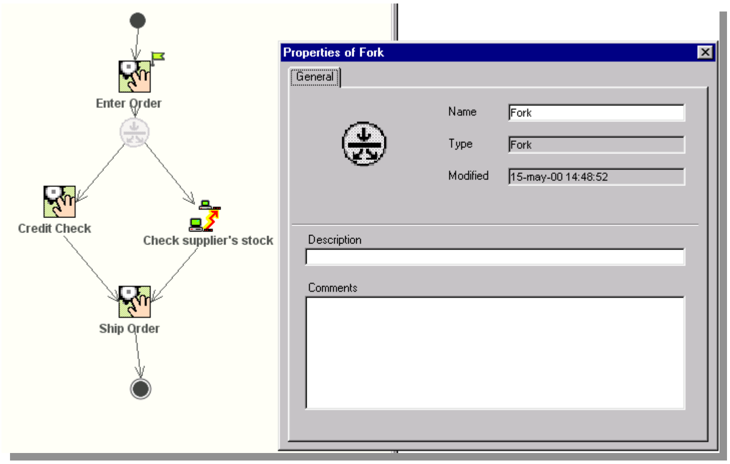
Conditions add flexibility to the process flow based on specific process data. Condition can be modified during deployment. For example, a large order can require an extra check.

A condition is modeled within the process model as a decision point where a conditional expression, a property of the condition, is evaluated. The expression used in the condition is created in the Expression Editor using process variables, business variables, and a choice of operators. The direction of flow after the condition is either along the True flow or along the False flow that originate from the condition.

A condition will always be followed by an OR join.

UNIFACE V8

AND-Split



Properties of Fork

General

Name: Fork

Type: Fork

Modified: 15-may-00 14:48:52

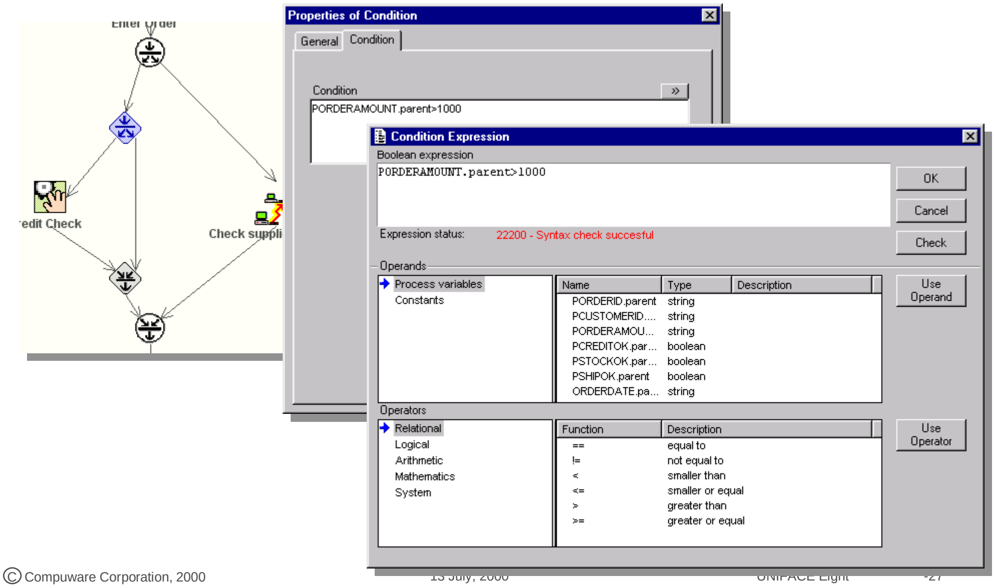
Description

Comments

© Compuware Corporation, 2000   13 July, 2000   UNIFACE Eight   -26

UNIFACE V8

Condition



Properties of Condition

General | Condition

Condition: PORDERAMOUNT.parent>1000

Condition Expression

Boolean expression

```
PORDERAMOUNT.parent>1000
```

Expression status: 22200 - Syntax check succesful

OK | Cancel | Check

Operands

- Process variables
- Constants

Name	Type	Description
PORDERID.parent	string	
PCUSTOMERID....	string	
PORDERAMOU...	string	
PCREDITOK.par...	boolean	
PSTOCKOK.par...	boolean	
PSHIPOK.parent	boolean	
ORDERDATE.pa...	string	

Use Operand

Operators

- Relational
- Logical
- Arithmetic
- Mathematics
- System

Function	Description
==	equal to
!=	not equal to
<	smaller than
<=	smaller or equal
>	greater than
>=	greater or equal

Use Operator

© Compuware Corporation, 2000   13 July, 2000   UNIFACE Eight   -27

2.5.2 Join

A join is a process graph element with two incoming and one outgoing flow. Joins are used to synchronize processes after parallel execution. There are the following types of joins in a process flow:

AND-join

AND-join is a point in the workflow, after an AND-split, where two parallel executing activities converge into a single common thread or control

OR-join

An OR-join is a point in the workflow, after a condition, where the alternative activity workflow branches converge into a single common activity as the next step within the workflow.

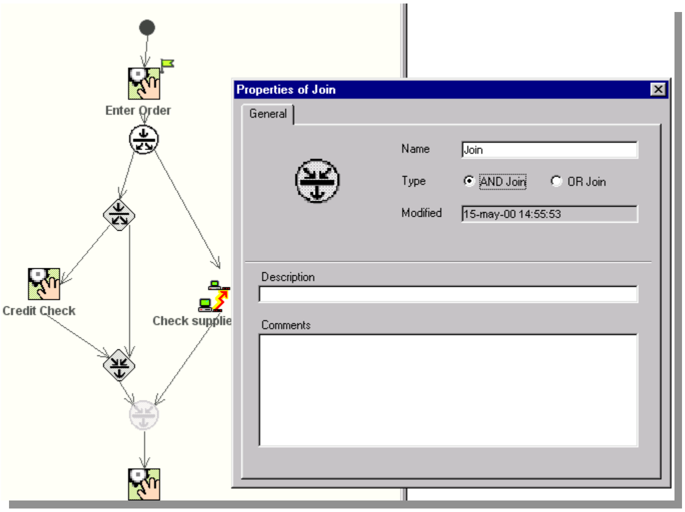
2.5.3 Abort

An abort step can be inserted in a process flow to abort the process.

All scheduled tasks will be aborted, including parent processes.

For example a process instance can be aborted if the credit check for a customer fails.

Join



Properties of Join

General

Name: Join

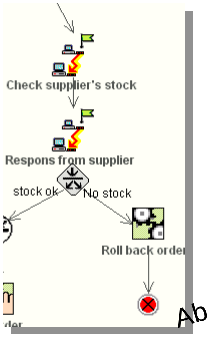
Type: ● AND Join ○ OR Join

Modified: 15-may-00 14:55:53

Description

Comments

Abort



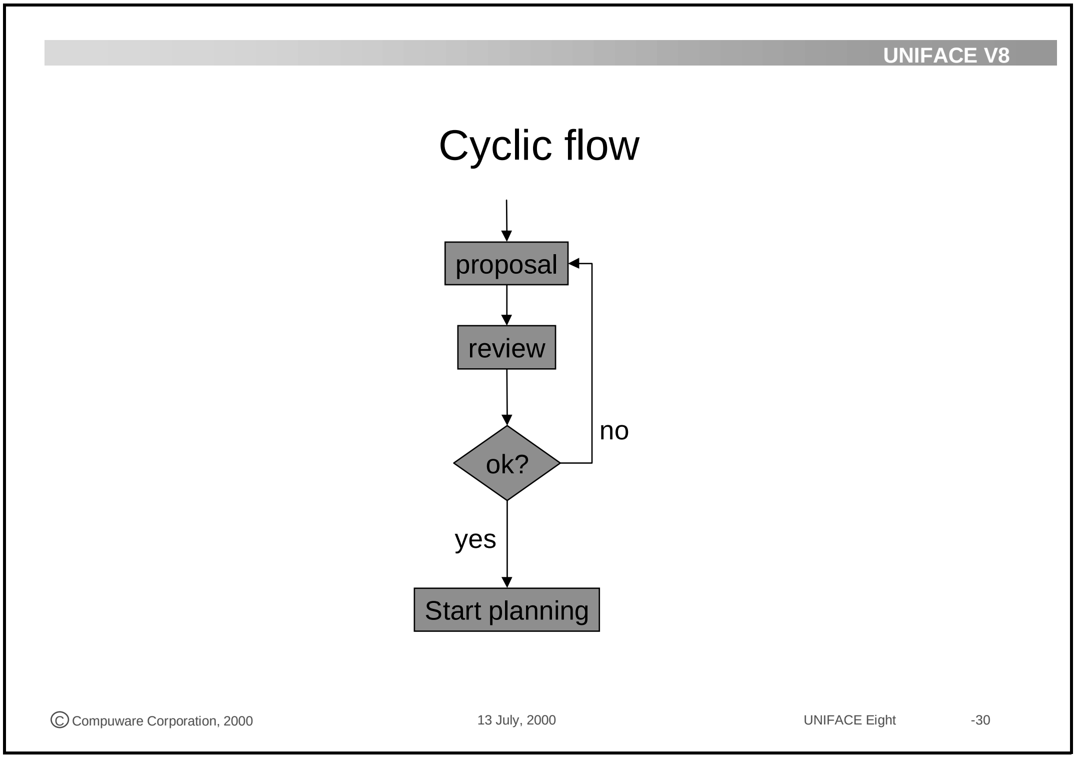
Aborts the process

2.6 Cyclic flows

A cyclic flow occurs in a process model when one of the flows leads back to a task that occurs before. A cyclic flow is used to enable the looping of a sequence of tasks until the evaluation of the expression in the condition element allows the process flow to break out of the loop.

A condition element must be present to break or finish this loop, otherwise the cyclic flow continues indefinitely.

An example of a cyclic flow is the processing of a review process. The process starts with a proposal, followed by a review. If the review is approved, the planning can start. If the review is not approved, process flow returns to the proposal stage. This return to the proposal starts a cyclic flow that continues until the review is approved.



UNIFACE V8

Cyclic flow

proposal

review

ok?

no

yes

Start planning

ⓒ Compuware Corporation, 2000 13 July, 2000 UNIFACE Eight -30

2.7 Consistency checks

Consistency checking is a feature of the BPA Workbench that examines processes and their underlying tasks for completeness and consistency. Consistency checking must be done before the models are exported to the deployment database.

When you choose Edit->Check Consistency, a Consistency Check Report is produced on-screen that shows warnings and errors concerning your process model, for example:

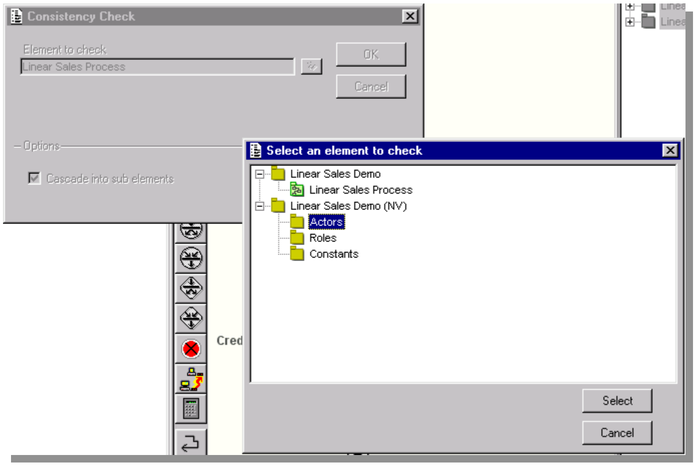
- An interactive or batch task that does not have complete reference information about the component that is to implement the task
- A task that does not contain an input expression to an input parameter
- A parameter that has no in mapping and no default value, or has an in mapping and a default value
- A task that is not correctly included in the process flow, that is, one that is missing an incoming or outgoing flow element
- A task that has no assignment rule specified
- A missing flow elements for splits, joins, and conditions

Overview

There are a number of guidelines to help you model processes. Some of them are intuitive or obvious. In the following list, the word ‘step’ is used just in a general way meaning all parts of a process in a non specific way - start, end, task, join, split, condition, and so forth:

- Every process contains a start step, the beginning of a process flow
- Every process contains an end step, the final step of a process flow
- A process can contain any number of abort steps
- Steps are connected to one another by a flow
- A flow can only be drawn once the steps it is connecting are drawn.
- A task within a process has one incoming and one outgoing flow
- An AND-split step has one incoming and two outgoing flows
- A join step has two incoming flows and one outgoing flow
- A condition has one incoming flow and two outgoing flows
- Properties must be defined for all processes, tasks, and conditions
- Flow properties must be defined for flows leaving a condition to determine which flow to follow if the condition is true or is false.

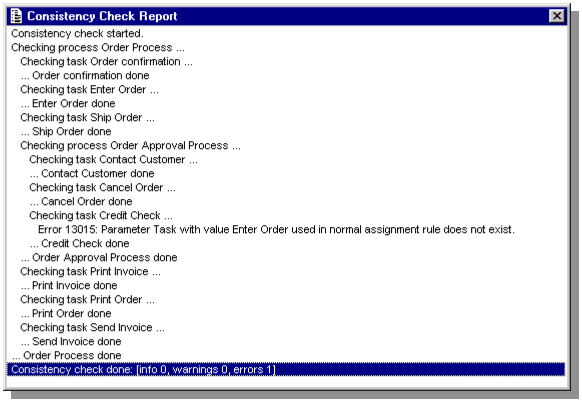
Consistency check (1)



Element to check: Linear Sales Process

☑ Cascade into sub elements

Consistency check (2)



```
Consistency check started.
Checking process Order Process ...
  Checking task Order confirmation ...
  ... Order confirmation done
  Checking task Enter Order ...
  ... Enter Order done
  Checking task Ship Order ...
  ... Ship Order done
  Checking process Order Approval Process ...
    Checking task Contact Customer ...
    ... Contact Customer done
    Checking task Cancel Order ...
    ... Cancel Order done
    Checking task Credit Check ...
      Error 13015: Parameter Task with value Enter Order used in normal assignment rule does not exist.
    ... Credit Check done
  ... Order Approval Process done
  Checking task Print Invoice ...
  ... Print Invoice done
  Checking task Print Order ...
  ... Print Order done
  Checking task Send Invoice ...
  ... Send Invoice done
... Order Process done
Consistency check done: [info 0, warnings 0, errors 1]
```

2.8 Integration with other workspaces

BPA automates a set of procedures and activities. For the execution of interactive and batch tasks a software implementation is needed. A reference to the task execution application must be entered on the properties form of the task.

2.8.1 Task execution application

A task execution application (tea) is an application that is invoked by the workflow management system to execute a task component. Both a tasklist and a Workflow Engine can invoke a task execution. The Workflow Engine can invoke the execution of a batch task using the Batch Task Server.

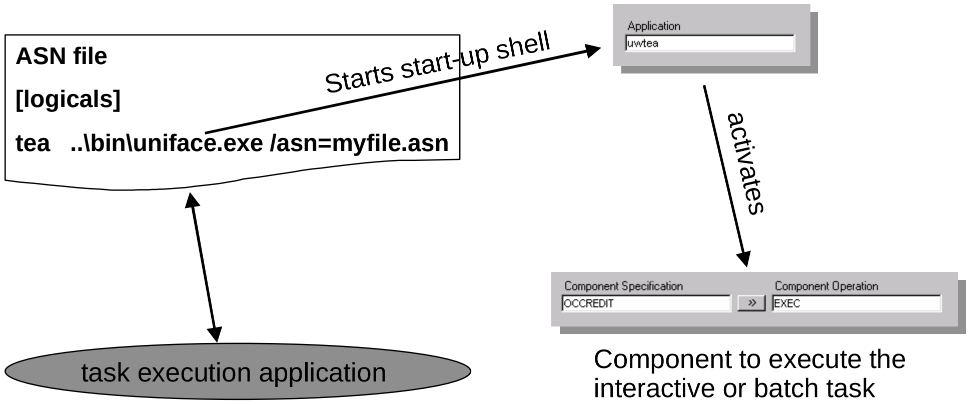
For example, when a user starts an order entry task, the Task Desktop starts the application that includes the order entry form.

The task execution application is installed in the task execution path.

In the logicals section of the assignment file, map this task execution path to the logical tea.

UNIFACE V8

Task execution application



ASN file

[logicals]

tea ..\bin\uniface.exe /asn=myfile.asn

Application: uwtea

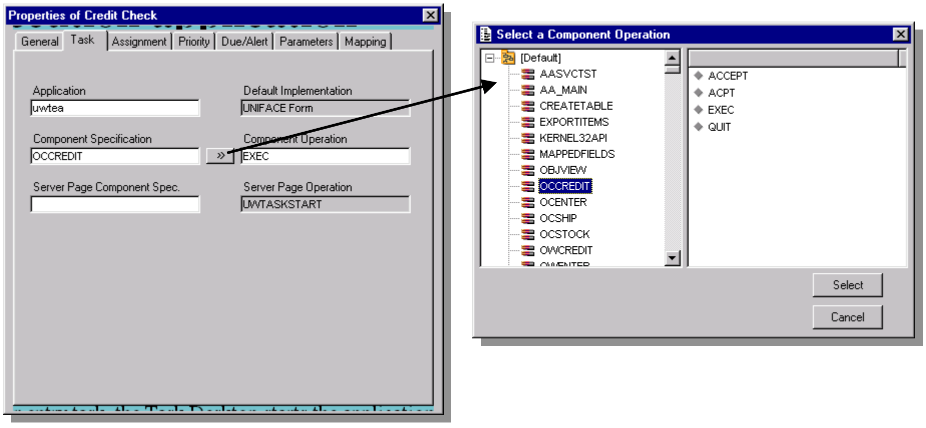
Component Specification: OCCREDIT

Component Operation: EXEC

Component to execute the interactive or batch task

UNIFACE V8

Integration with the other workspaces



Properties of Credit Check

Application: uwtea

Default Implementation: UNIFACE Form

Component Specification: OCCREDIT

Component Operation: EXEC

Server Page Component Spec.:

Server Page Operation: UWTASKSTART

Select a Component Operation

[Default]
- AASVCTST
- AA_MAIN
- CREATETABLE
- EXPORTITEMS
- KERNEL32API
- MAPPEDFIELDS
- OBJVIEW
- OCCREDIT
- OCENTER
- OCSHIP
- OCSTOCK
- OWCREDIT

- ACCEPT
- ACPT
- EXEC
- QUIT

2.8.2 Task component construction

There is no difference in constructing components to execute tasks from the Task Desktop in contrast with constructing components not managed by the workflow system.

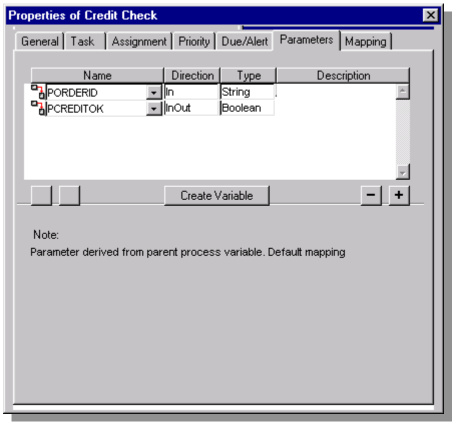
What has to be taken into account is the parameter passing. Parameters are used to pass information from one task to another but also to pass information from one component to another.

You must define the parameters that are used by the software component in same order and as the same data type as you have defined the parameters for the task. You do not need to use the same names for the parameters.

Because one task execution component does not immediately activates another task execution component the value of the parameters is assigned to a process variable and stored in the workflow repository by the process of matching. When the next task is executed, the process variable is retrieved from the repository and assigned to the parameter.

UNIFACE V8

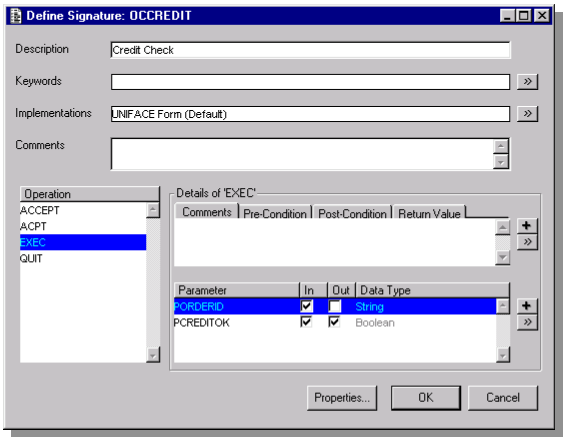
Task component construction



Properties of Credit Check

Name	Direction	Type	Description
PORDERID	In	String	
PCREDITOK	InOut	Boolean	

Note:
Parameter derived from parent process variable. Default mapping



Define Signature: OCCREDIT

Description: Credit Check
Keywords:
Implementations: UNIFACE Form (Default)
Comments:

Parameter	In	Out	Data Type
PORDERID	☑	☐	String
PCREDITOK	☑	☑	Boolean

© Compuware Corporation, 2000 13 July, 2000 UNIFACE Eight -35

2.9 Test mode

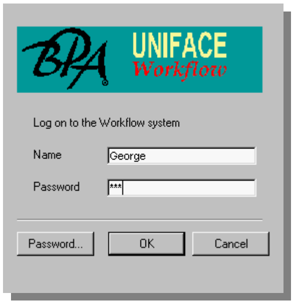
For both the Task Desktop and the Workflow Engine there is a test mode that can be used during the development phase. In the test mode the development workflow repository is used as if it was deployed. The functionality offered is the same as during deployment.

After logging on to the Workflow Engine, using the proper user account a Task Desktop is displayed. The same goes for the Workflow Engine to monitor the whole workflow process.

The next lesson provides more information on testing facilities.

UNIFACE V8

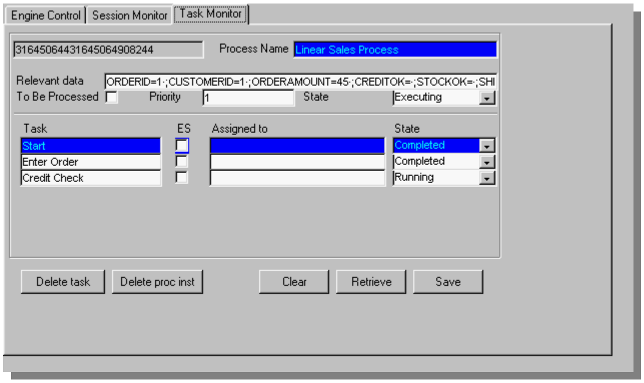
Test modes



Log on to the Workflow system

Name: George

Password: ***



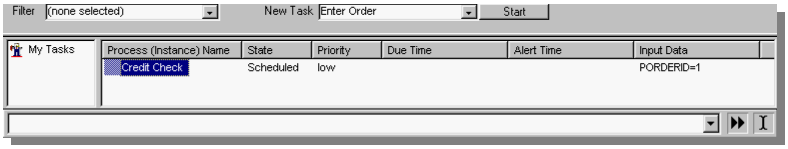
31645064431645064908244

Process Name: Linear Sales Process

Relevant data: ORDERID=1·;CUSTOMERID=1·;ORDERAMOUNT=45·;CREDITOK=·;STOCKOK=·;SHI

To Be Processed □ Priority: 1 State: Executing

Task	ES	Assigned to	State
Start	□		Completed
Enter Order	□		Completed
Credit Check	□		Running



Filter: (none selected)

New Task: Enter Order

Process (Instance) Name	State	Priority	Due Time	Alert Time	Input Data
Credit Check	Scheduled	low			PORDERID=1

Exercise 2.1. Create an non versioned package

Every business process is executed within an organization. The organization has a structure consisting of organizational units and roles. The roles are performed by actors. You will now examine the Linear Sales Organization and then create your own organization.

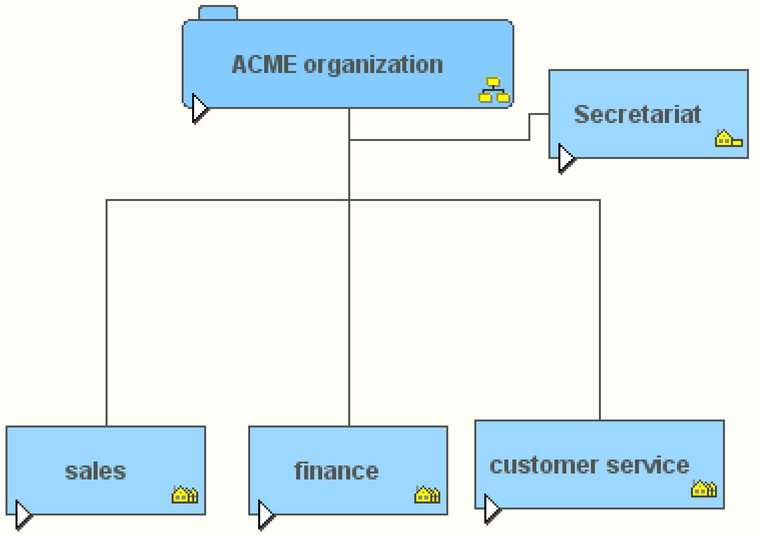
- Start the UNIFACE 8 development environment and go to the Business Automation workspace. Select the Business Engineer Workbench.
- Select the Linear Sales Organization and open it in the Director tree.
- In the Director, expand the tree and then click on Head Quarters. In the graphical modeling area you should now see the organizational structure. In that structure, click right on the Finance department and invoke the properties inspector. Examine the Roles tab to see who does what in Finance.
- Now click in the Director tree on Actors and notice how the property inspector shifts focus. In the graphical modeling area select the actor George and examine the Employment tab to see which roles George currently performs.

Now create a new organization called the ACME Organization.

- Select File—>New—>Non Versioned Package and call it My Organization Package.

A package is a container for the objects that are needed to define a organization model or a business process model. Packages can be defined on several levels and can be compared with a directory structure.

- Select File—>New—>Organization to create the new organization. In the Director, expand the tree and use the General tab of the Property inspector to name the organization: ACME Organization.
- In the Director tree click on the ACME Organization. In the graphical modeling area you should now see the organizational structure.
- The ACME Organization has 4 departments, Sales, Finance, Customer Service and one staff department Secretariat. The ACME organization has one Secretary as Staff department. Use the Unit Tool to draw these the departments. Use the staff unit tool to draw the staff department.
- With the Relation tool you can draw the complete organizational diagram:



Now create roles for the new ACME Organization

- First create a package to contain all role definitions. In the director tree, select the top level package My Organization package and then select File—>New—>Package. Rename this package to Roles.
- Select this new package Roles and select File—>New—>Role. Call this role Clerk. Create also the roles Manager, Secretary and Receptionist

These roles can now be allocated to the departments of the ACME Organization.

- Go to the property inspector of the Finance department and select the Roles tab. Use the (+) button to select the role Clerk. Do the same for the Sales department. Allocate the role manager to the ACME Organization. Assign the role of Secretary to the Secretariat and Receptionist to customer service.

The organizational infrastructure is now completed, you only still need to populate this structure with actors.

- In the director tree, select the top level package: ‘My Organization Package’, then select File—>New—> Package and rename this package Actors. Select this package Actors and choose File—>New—>Actor. Do this 4 times. Rename the actors to Boris, David, John and Suzie.

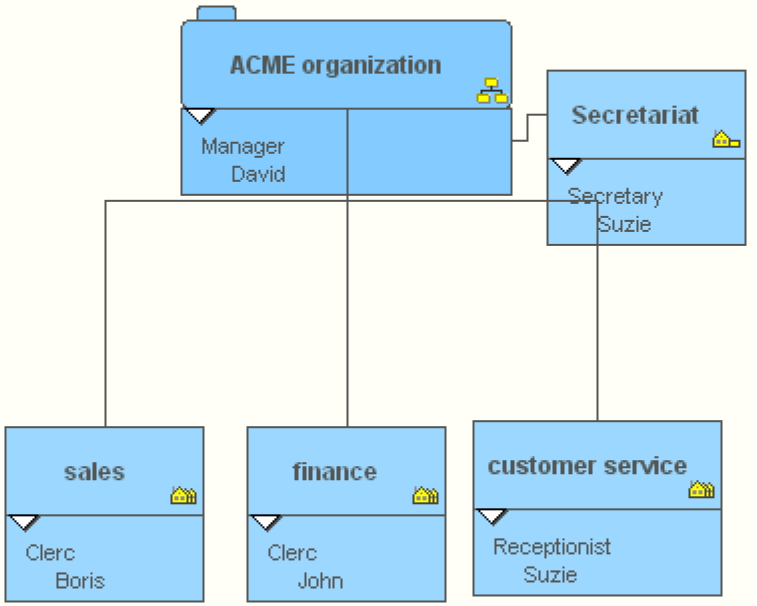
- Use the Actor Data tab of each actor to define a password. The initial password is the same as the name of the actor. Use the actors name (all lowercase) as password.

Allocate the actors to roles and organization units, using the Employment tab of each actor:

Boris	Clerk in the Sales department
David.	Manager in the ACME Organization
John	Clerk in Finance department
Suzie	Secretary in the Secretariat
	Receptionist in the Customer service department

- Click on the ACME organization in the Director pane and expand all the units.

You will see an overview of the ACME organization with actor and assigned roles for the actors.



- Save the non versioned package.

Exercise 2.2. Create a versioned package

In this exercise you will create a new versioned package to store the business process model definitions. This business process model will be enriched during subsequent exercises.

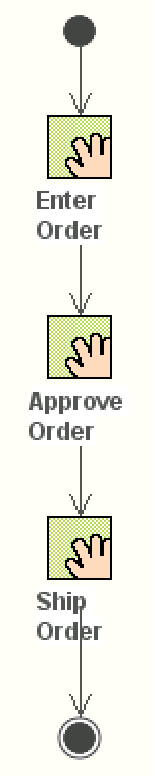
- The first step is to create a new top package. Select File—>New—>Versioned Package and rename this package to Order Process Package.

The Order Process will be executed by the ACME Organization. To make the organization known to the process, you must include the non versioned organization package into the process package.

- Invoke the property inspector of the Order Process Package and go to the Imported tab and import the non versioned organization package you've created in the previous exercise.
- In the catalog pane you can expand the tree to keep an overview of all the available objects from the organization for the business process.

In the catalog pane of the Business Engineer Workbench you can browse through the imported organizational package.

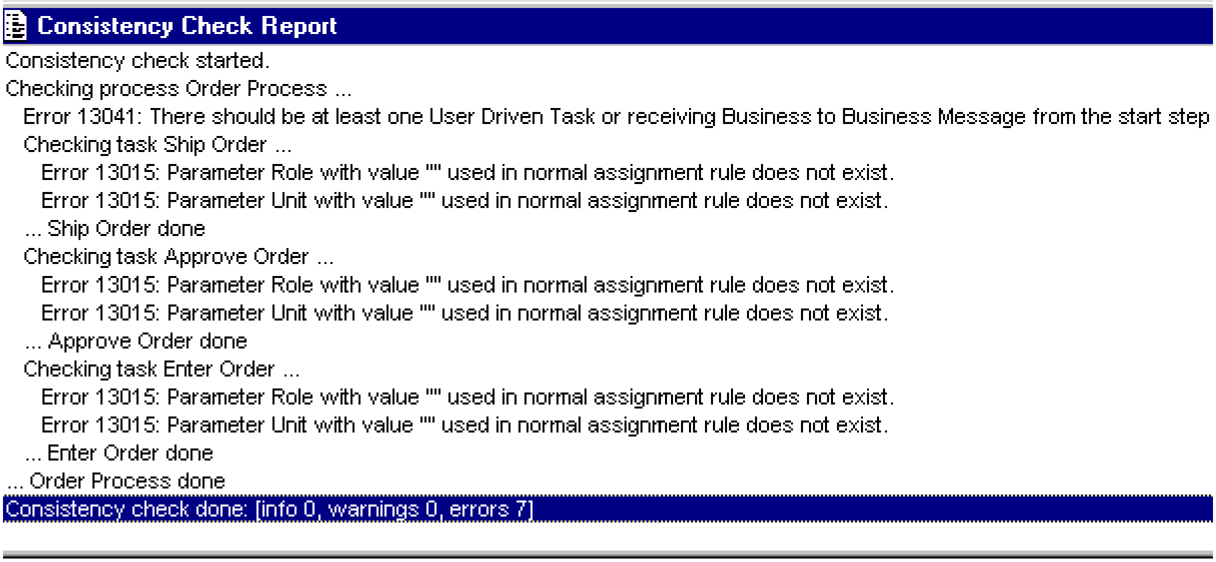
- Now create the process. Select File—>New—>Compound Process and rename this to Order Process.
- Highlight the 'Order Process' in the Director pane. In the graphical modeling area you now see the start and ending points of the process to be defined. Begin by creating a manual task Enter Order.
- Add two more manual tasks, Approve Order and Ship Order.
- To combine these 3 tasks into a process you need to define the process flow. Use the flow tool to create the following process flow:



To verify the completeness of this initial process definition you can check the process model.

- In the Director, navigate to the process, then select Edit—>Check Consistency.

You receive errors because the tasks are not yet assigned to actors and each business process always starts with User driven task or a Business to Business message. These are the remaining step to complete the process definition.

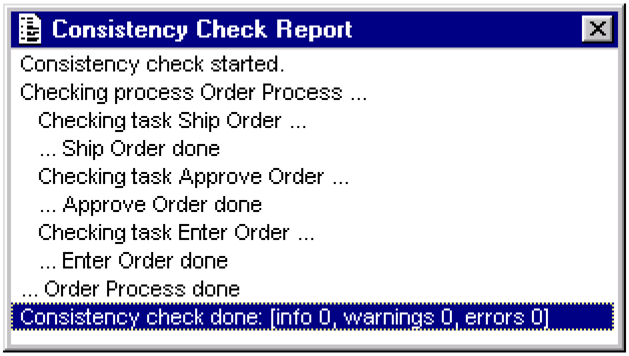


Consistency Check Report

```
Consistency check started.
Checking process Order Process ...
  Error 13041: There should be at least one User Driven Task or receiving Business to Business Message from the start step
  Checking task Ship Order ...
    Error 13015: Parameter Role with value "" used in normal assignment rule does not exist.
    Error 13015: Parameter Unit with value "" used in normal assignment rule does not exist.
  ... Ship Order done
  Checking task Approve Order ...
    Error 13015: Parameter Role with value "" used in normal assignment rule does not exist.
    Error 13015: Parameter Unit with value "" used in normal assignment rule does not exist.
  ... Approve Order done
  Checking task Enter Order ...
    Error 13015: Parameter Role with value "" used in normal assignment rule does not exist.
    Error 13015: Parameter Unit with value "" used in normal assignment rule does not exist.
  ... Enter Order done
... Order Process done
Consistency check done: [info 0, warnings 0, errors 7]
```

- Invoke the property inspector of the Enter Order task and go to the Assignment tab. Choose as the assignment type for the Enter Order task “All Actors in Unit”, and select the unit Sales.

  On the General tab define the task as ‘Driven by User’.
- For the Approve Order task, choose as the assignment type “Actors with Role in Unit”. Select the Manager role the ACME organization unit.
- For the Ship Order task, choose the assignment type “Actor” and select Boris.
- Now check again the consistency of your modeled process by selecting Edit—>Check Consistency.



Consistency Check Report

```
Consistency check started.
Checking process Order Process ...
  Checking task Ship Order ...
  ... Ship Order done
  Checking task Approve Order ...
  ... Approve Order done
  Checking task Enter Order ...
  ... Enter Order done
... Order Process done
Consistency check done: [info 0, warnings 0, errors 0]
```

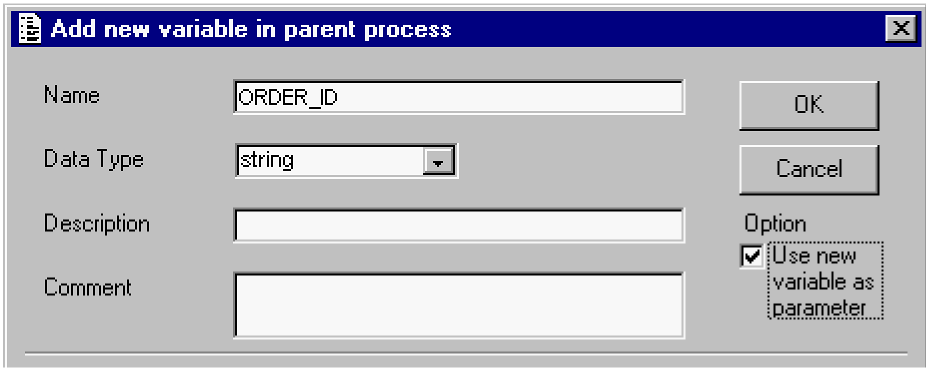
Exercise 2.3. Define Task types

All tasks in the Order Process are initially implemented without any specifications. In this exercise the task definitions are being refined. To pass data and state you need to define interfaces for the tasks.

Enter Order is the first task of the process. The process is started by the appropriate event, in this case a customer who places an order. Because the execution of this first task is triggered by an event, it must be defined as a user driven task. In the General tab of the properties inspector for Enter Order, you have set the Driven By property to user. Notice that Enter Order now is flagged in the graphical modeling area.

The parameters of Enter Order must be mapped to process variables, to make their value available for subsequent tasks. This mapping can be automated by directly generating the parameters from process variables.

- Go to the parameters tab of Enter Order. Use the button Create Variables and create ORDER_ID as a string variable. Check the option: Use new variable as parameter



Add new variable in parent process

Name: ORDER_ID

Data Type: string

Description:

Comment:

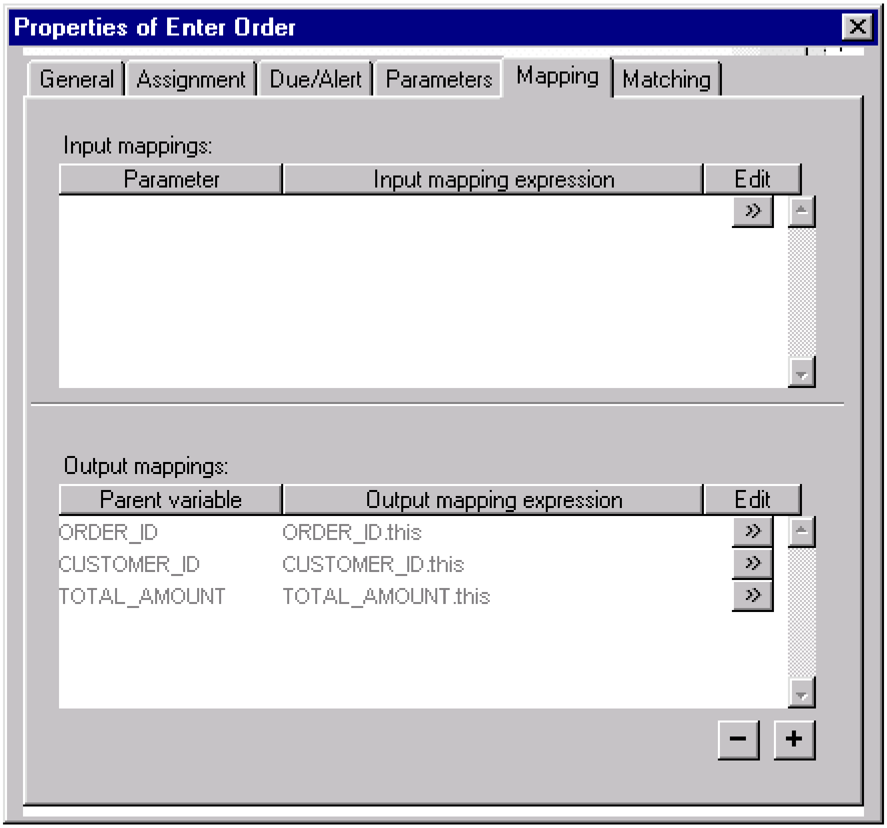
Option

☑ Use new variable as parameter

- Create in the same manner the variables CUSTOMER_ID (String) and TOTAL_AMOUNT (Numeric) also.
- The Enter Order task now has three parameters. Be sure that all parameters have been defined as OUT parameters

UNIFACE has now automatically mapped the task parameters to their corresponding process variables.

- Go to the mapping tab of Enter Order to inspect the generated mappings:



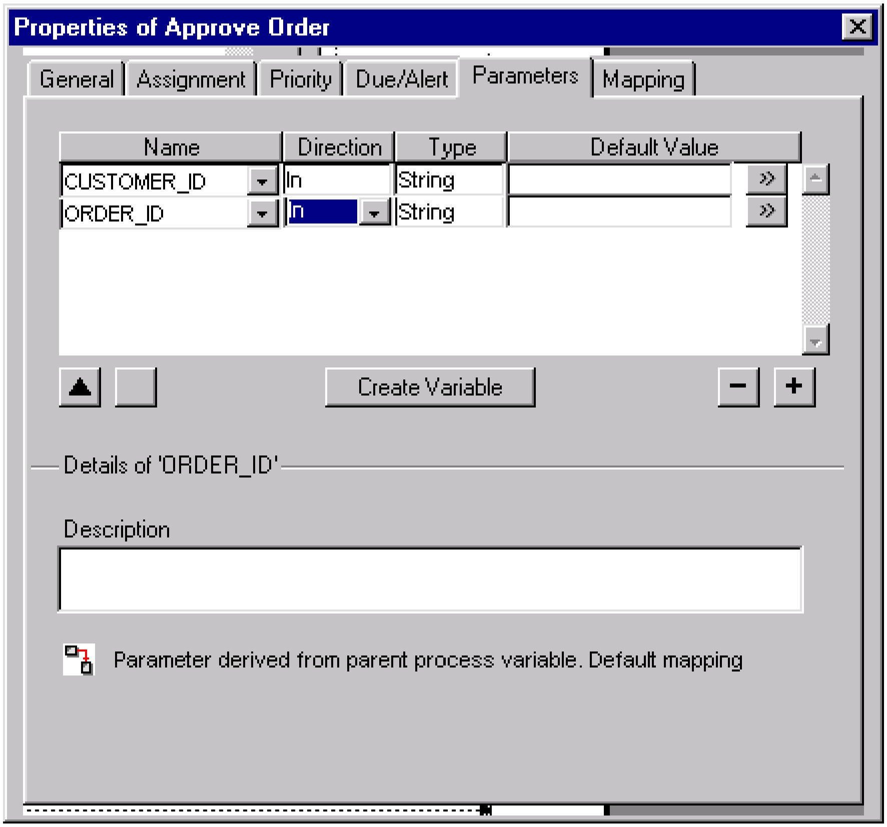
Input mappings:

Parameter	Input mapping expression	Edit

Output mappings:

Parent variable	Output mapping expression	Edit
ORDER_ID	ORDER_ID.this	»
CUSTOMER_ID	CUSTOMER_ID.this	»
TOTAL_AMOUNT	TOTAL_AMOUNT.this	»

- To define the parameters for the task Approve Order, go to the parameters tab and click Add (+). In the name field of the parameter a drop-down list has appeared that allows you to choose from existing process variables. Choose CUSTOMER_ID and ORDER_ID and specify both to be IN parameters.



Name	Direction	Type	Default Value
CUSTOMER_ID	In	String	
ORDER_ID	In	String	

Details of 'ORDER_ID'

Description

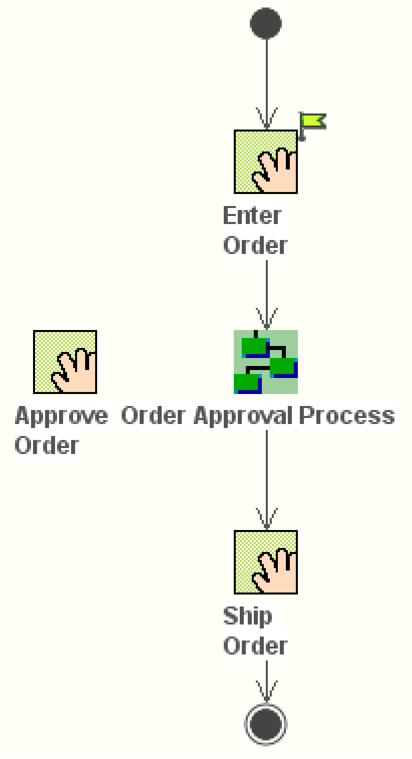
Parameter derived from parent process variable. Default mapping

- Examine the mapping of the parameters of Approve Order to the corresponding process variables.
- Now define the parameters for the task Ship Order ORDER_ID and SHIPPED_OK. ORDER_ID is derived from a process variable and is an IN parameter. SHIPPED_OK (Boolean) is a new value and needs to be created as process variable (Using the create variables button on the parameters tab). Define SHIPPED_OK as an OUT parameter.
- Examine the mapping of the parameters of Ship Order to the corresponding process variables.
- Check the consistency of your model.

Exercise 2.4. Create a compound process

In this exercise you create a compound process and embed that within its parent process. The task Order Approval is going to be refined into 3 subtasks, Credit Check, Contact Customer and Cancel order. Together these three tasks form the compound process Order Approval. If the credit check fails the customer is contacted and the order is cancelled. This aborts the process.

- Navigate to the graphical modeling area of the Order Process. Use the Compound Process Tool to add a nested (compound) process Order Approval Process.
- Re-arrange the process flow to include the compound process Order Approval Process, instead of the manual task Approve Order. After re-arranging the flow you can delete the manual task Approve Order.



The compound process Order Approval Process can be seen as a task in its parent process, the Order Process. The parent and child process need to exchange data, using parameters and variables, just like tasks within a single process need to exchange data. Therefore you need to define parameters for Order Approval Process. These process parameters of the child process are mapped to process variables of the parent process.

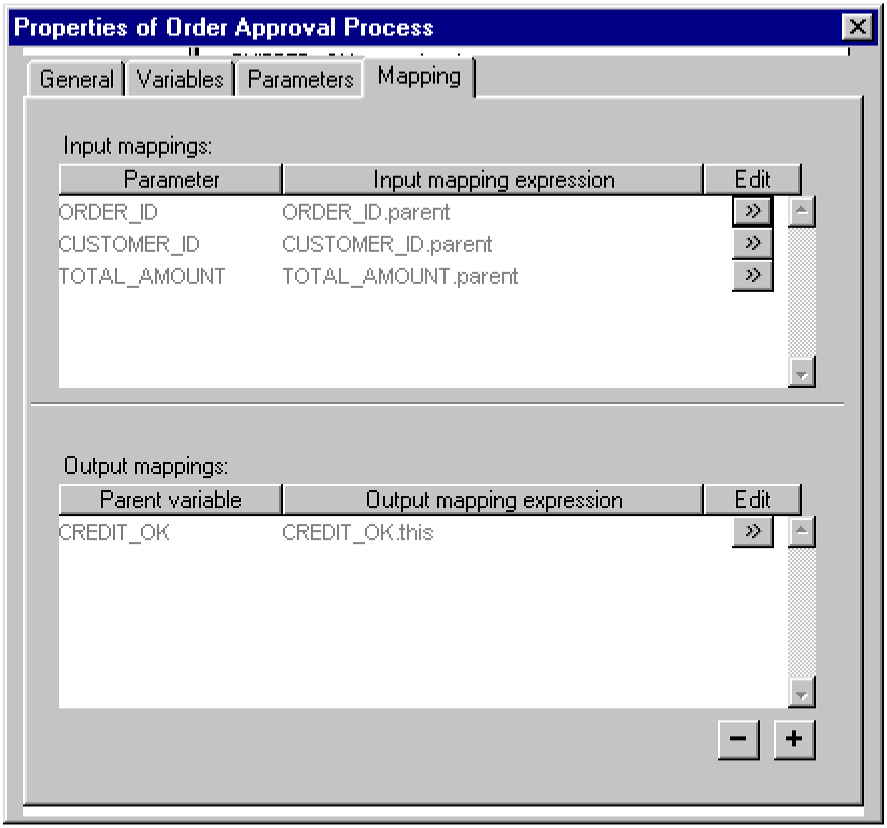
- For the Order Approval Process create the following 4parameters.

  The 3 IN parameters can be created from the existing process variables, using the Add (+) button on the parameters tab of the property inspector for the Order Approval Process.

  The 1 OUT parameters first need to be created as process variables.

ORDER_ID	String	IN
CUSTOMER_ID	String	IN
TOTAL_AMOUNT	Numeric	IN
CREDIT_OK	Boolean	OUT

The following mappings are generated for these process parameters:



Input mappings:

Parameter	Input mapping expression
ORDER_ID	ORDER_ID.parent
CUSTOMER_ID	CUSTOMER_ID.parent
TOTAL_AMOUNT	TOTAL_AMOUNT.parent

Output mappings:

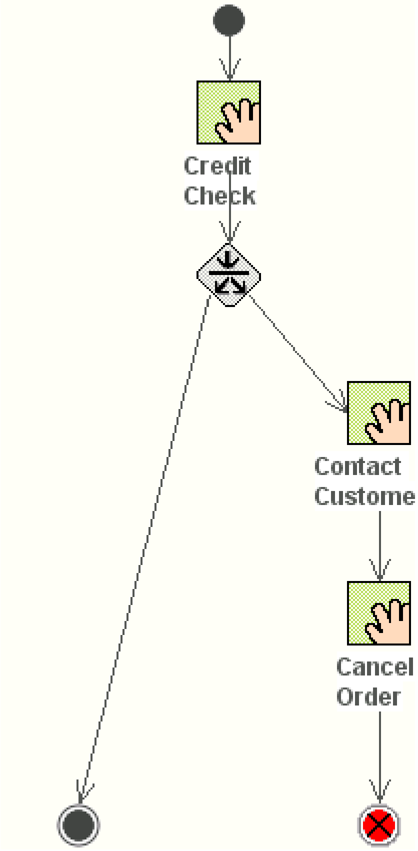
Parent variable	Output mapping expression
CREDIT_OK	CREDIT_OK.this

Now the process to process communication has been arranged you can define the tasks within the Order Approval Process.

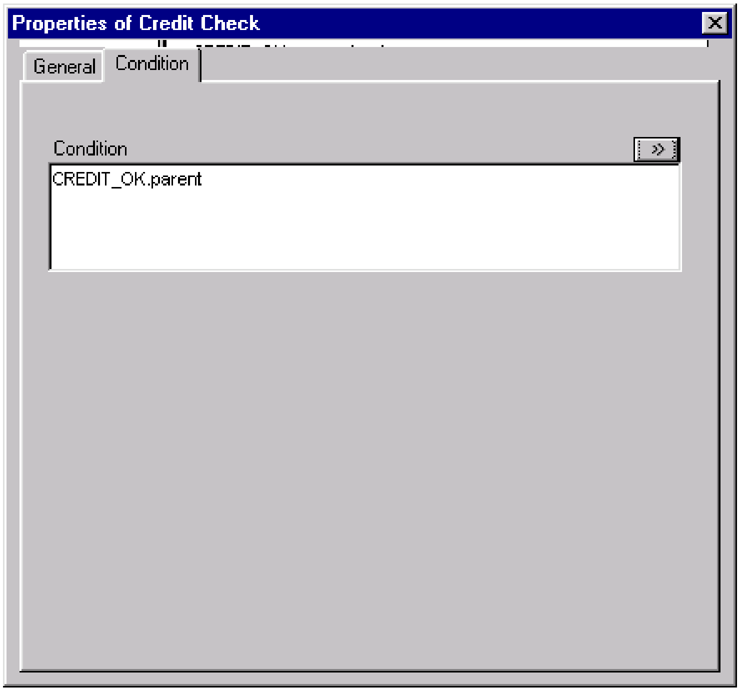
- In the Director, select the Order Approval Process and open the diagram.
- Create 1 condition and 3 new manual tasks in this Diagram:

  The condition checks if the Credit Check was successful.

  Contact Customer and Cancel Order are defined as manual tasks.
- Add an abort step, for if the credit checks fails the order process is cancelled.



- For the task Credit Check, define the following parameters:
  CUSTOMER_ID, STRING, IN
  TOTAL_AMOUNT, NUMERIC, IN
  CREDIT_OK, BOOLEAN, OUT
- For the Condition Credit Check, define the following condition:
  CREDIT_OK.parent



CREDIT_OK.parent

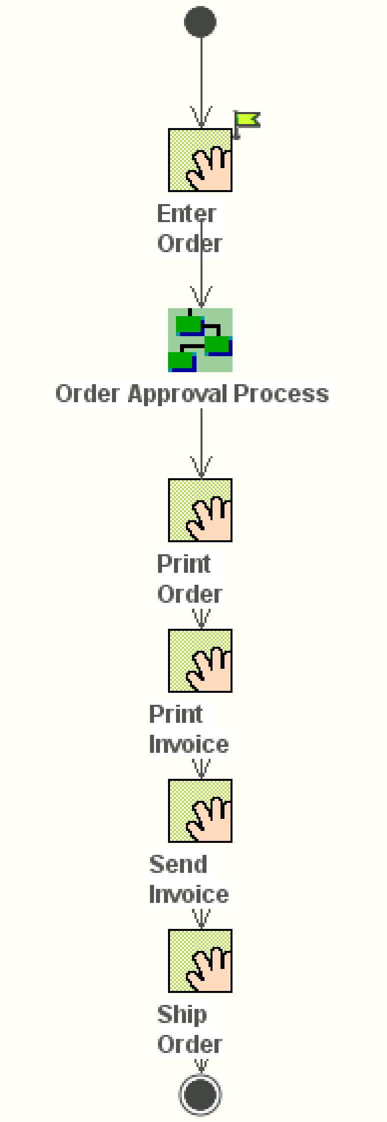
- The outgoing arrow to the ‘Contact Customer’ must be set for the condition to False. The other arrow in case the condition is True. Use the flow tab page of the Flow properties.
- For both ‘Contact Customer’ and ‘Cancel Order’ use ORDER_ID and CUSTOMER_ID as IN parameters.
- Assign all actors of the Customer Service department to ‘Contact Customer’ and ‘Cancel Order’. The ‘Credit Check’ must be assigned to ‘manager’ of the ‘ACME organization
- Run a consistency check for the ‘Order Approval’ compound process.
- Save the Business Process Model.

You have now completed the definition of the compound process Order Approval. It is ready to be tested.

Exercise 2.5. Add tasks

To complete the order handling the Order Process needs to be extended with 3 more tasks: Print Order, Print Invoice and Send Invoice

- Create the manual tasks: Print Order, Print Invoice and Send Invoice as follows:



- Define the following parameters for these tasks (using the available process variables):

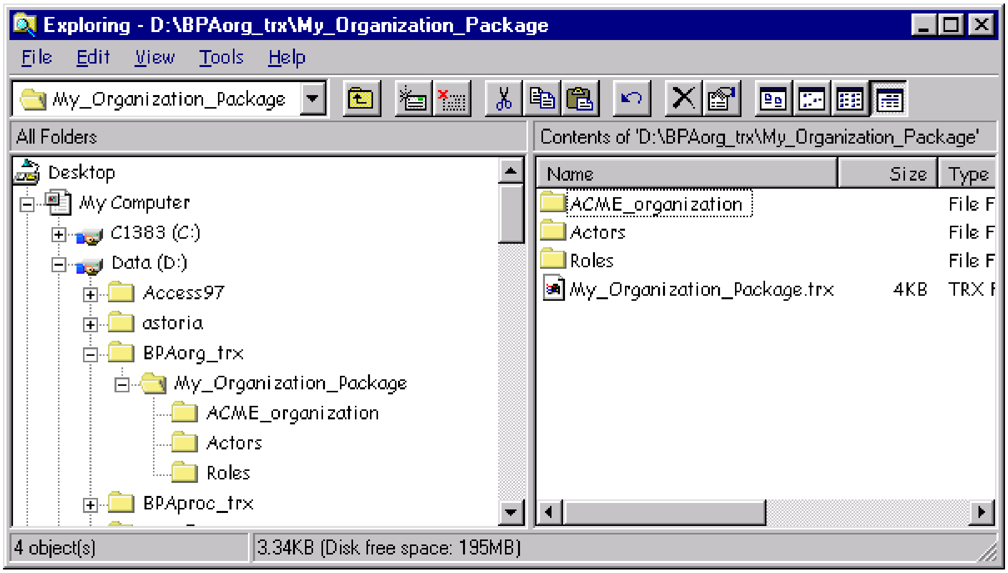
Use ORDER_ID and CUSTOMER_ID as IN parameters for all the tasks defined.

- For each added task actors must be assigned to execute the task: The finance department is responsible for invoicing and the sales department is responsible for order administration. Assign actors to the tasks according the responsibilities of the departments.
- Save and run a consistency check.

2.6. Prepare for deployment

Before you will deploy your modeled business process you will build two exports. One export must be created from the organizational model and the other one from the business process model. Both model must be exported because the organizational model is use as reference for the process model.

- Create from your Windows explorer two folders: BPAorg_trx and BPAproc_trx. The first will contain the export of your organization the second folder will be used to store the exported business process.
- Load ‘My Organization Package’ into your development environment.
- Go to Utilities -> Export and browse to your BPAorg_trx folder.
- Take ‘My Organization Package’ as element to be exported.
- Browse to your BPAorg_trx folder as Export Folder.
- Export Format must be checked as Tree and the Export File Type is trx.
- Click ‘OK’ to start the export process.



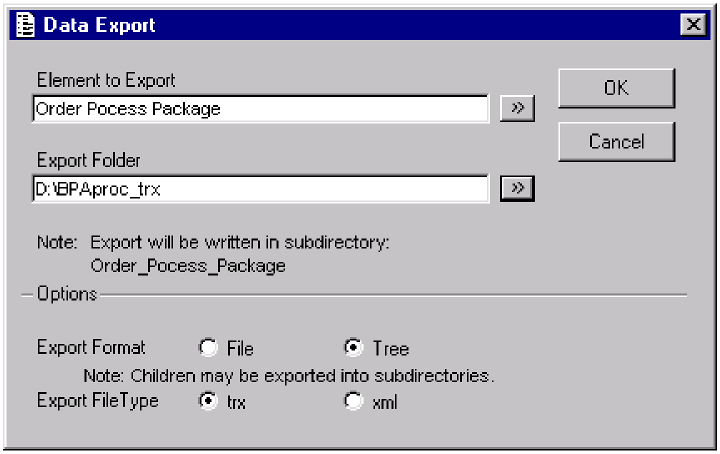
Contents of 'D:\BPAorg_trx\My_Organization_Package'

Name	Size	Type
ACME_organization		File F
Actors		File F
Roles		File F
My_Organization_Package.trx	4KB	TRX F

4 object(s)   3.34KB (Disk free space: 195MB)

- Check in your windows explorer the directory structure that has been created during the export.
- Repeat this procedure for the Business Process. Load the ‘Order Process Package’
- Go to Utilities --> Export

- Select ‘Order Process Package’ as element to be exported.
- Browse to BPAproc_trx folder.
- Export Format as Tree and Export File Type as trx.



Element to Export: Order Pocess Package

Export Folder: D:\BPAproc_trx

Note: Export will be written in subdirectory:
Order_Pocess_Package

Options

Export Format ○ File ● Tree

Note: Children may be exported into subdirectories.

Export FileType ● trx ○ xml

- Click ‘OK’ to start the export process.

Both exports will be used as a starting point for deployment.