Unit4 Business World

Reference manual

Workflow

7.5

UNIT4

Last updated 2019-09-30

The documentation is designed to support Unit4 Business World.

This document is intended for Unit4 Business World Consultants and customer Super Users, and thus assumes in-depth knowledge of existing Unit4 Business World functionality.

Every effort has been made to supply complete and accurate information; however, the information in this document is subject to change without notice. UNIT4 N.V. and its group companies assume no responsibility or liability for any errors that may occur in the documentation.

Unit4 welcomes your comments as part of the process of continuous development and improvement of the documentation. Please contact Unit4 Customer Support if you have any questions.

Table of contents

Read this first

Purpose

This reference manual gives a detailed description of the Workflow functionality in Unit4 Business World, and is intended for people responsible for design and implementation of workflow processes. We assume that the reader - you - belong to this group of people.

Participation in workflow processes and actual processing of workflow tasks - performed by workflow users - is not described here.

Read on-line

The manual is published as a PDF document with hyperlinks, intended to be read on-line. You don't find any page references.

Hyperlinks are displayed like this: Read this first.

Content

We have divided the manual into a series of sections, starting with a general introduction:

- Workflow processing - an introduction. Introduces the tools and objects used to define typical workflow processes. This chapter explains the workflow model, the commonly used activity types and routing mechanisms. Since Unit4 Business World Workflow is very complex, and probably has more exceptions than rules(!), you may find the introduction a bit difficult when you first read it.
  In the final chapter in this section, Workflow terminology, we explain (define) the common, as well as Business World specific, concepts used in a workflow context. This chapter can also be used as an introduction.
- A set of sections describing the various tools and elements used in workflow process definition:
  - Users
  - Alerts
  - Actions (workflow actions)
  - Element types
  - Distribution and routing of tasks
- Process definition, where all elements are merged into a functional workflow process.

Business World Workflow - an introduction

Business process as workflow process

The Business World Workflow module allows you to model your business processes as automated workflow processes, and provides you with a set of tools for workflow process design.

Business objects as element types: Business World Workflow may be used for most business processes, but not for all. An important condition is that the objects intended for workflow distribution are properly defined in the system as so-called element types.

The Business World Element type

An element type is a tailored object type in Business World, intended for workflow distribution and processing. The element type

- provides a set of properties that makes it recognizable for the workflow engine, and
- behaves as a wrapper around the business objects normally found in the Business World system.

Most business object types relevant for workflow processing are already wrapped in predefined element types. But you can also add new ones - based on an attribute.

Process definition

A workflow process is defined for (is valid for) one or more element types, and consists of one or more activities or steps, as well as information about participating users (also called task recipients). A running (instantiated) process will always handle one – and only one – instance of one element type, referred to as the work item.

The work item will be routed from one activity to another, according to the defined *routing* and *distribution rules*. It can either be handled by a participant or processed automatically. In Business World, we refer to an activity handled by a participant as a *User step*, while a fully automated activity is referred to as a *System step*.

Process instantiation

A process definition can be seen as a dormant process - a process template - which needs to be activated - instantiated - before anything happens. This can happen in two ways:

- **Conditional – dependent of how the first activity is defined in the process:** Every time someone (a user, a server process) tries to save changes to element type objects which are included in an alive, but dormant, workflow process definition, the *workflow* *engine* will check if the start-up conditions for the process are fulfilled. If so, it will create a new instance of the process, grab the object, transform it to a proper work item, and execute the first workflow step.

- **As sub process:** When a workflow process is included as a sub-process step in another process definition, it will be instantiated if the work item of the parent process is routed through the sub-process.

Work lists (Task lists in Business World)

The various workflow participants will always get access to pending tasks via a *task list*. The task list will show all tasks assigned to the participant, and they may originate from one or more process instances.

In order to provide a clearer view of your tasks, you are able to group the task list according to various grouping criteria, such as the workflow process area and client. You can also set if the column is sorted in ascending or descending order and see all groups (within a selected column context) in a collapsed or expanded way.The task list is dynamically updated based on your grouping selection.

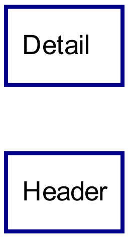
The task grouping is retained and used every time you use the task list until you reconfigure the grouping.

> **Note:** Additional options for a customisation of a task list are available in the Task Management Experience pack.

An important note about transactional element types

Most element types intended for transactional data (requisitions, invoices, sales order etc.) represents a header - detail structure (i.e. one invoice may consist of many lines). In Business World, these element types are defined with the "details" on top.

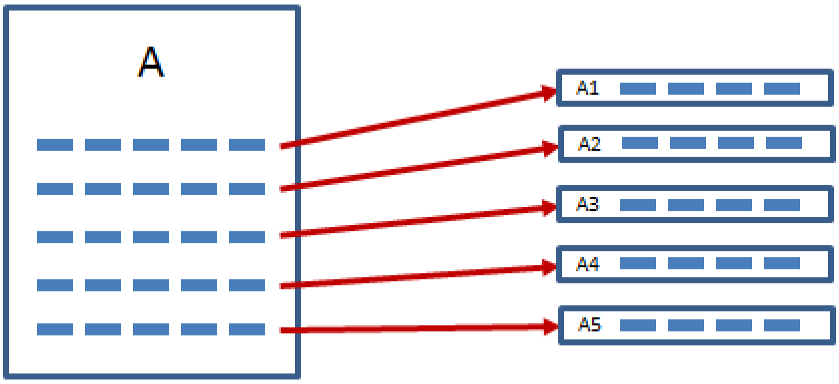
The diagram below illustrates the inverted header - detail structure:



Process instantiation: When you register and submit a new invoice, intended for workflow, the workflow engine will create *one process instance per detail row*, and not one for the invoice as a totality. The actual work item, per instance, will be the *detail* row.

It is thus - as intended - very likely that the various detail lines will be passed to different recipients, due to - for example - accounting rules, responsibility, etc. related to the posting information on the detail lines.

Illustration: You will work with the whole object (A). When you save your work, however, every single line (A1 to A5) will instantiate a new workflow process instance.



Implementing a workflow process - an example

The case

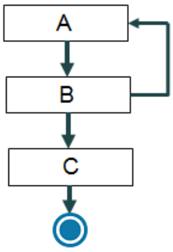
We will use what we think is a typical example to illustrate some main points in the workflow system.

The case is simple: When an invoice - normally with several detail lines - is registered in the system, it must be automatically passed to the relevant department manager(s) for approval - before posting. If a manager cannot approve the invoice line (probably a registration error), it must be sent back to the first user for correction. When all invoice lines are found ok, the complete invoice will be automatically posted.

Note that an invoice with several lines will lead to the instantiation of a *workflow process for each line*. The line details will be the *work item* for the instance.

Informal process diagram

The diagram below illustrates the general workflow:



Explanation: When a user registers and submits a new invoice (**A**), the workflow engine will start *one process instance per invoice line*. Each invoice line will, in workflow terminology, be the *work item* in the various process instances.

On basis of the COSTC value on each work item, it is passed to the correct department manager (**B**). If the manager rejects the work item, it is sent back to the user who originally initiated the workflow instance, with information about the error. The user must then correct the invoice line, and submit it again.

If the work item is OK, it is passed to the automatic posting routine (**C**). The workflow engine will ensure that **C** is not processed before all invoice lines are approved (this is part of the processing rules for invoices; invoice posting cannot take place before all invoice lines are approved).

When all detail lines are approved, **C** will be executed. The invoice is posted and the process stops.

Tasks in the task list

Each line in an invoice will create a workflow process instance, and several lines will probably be routed to the same department manager - as several workflow tasks.

To avoid a large amount of related tasks, Business World allows you to group related tasks in many ways. Default grouping is by process and task.

So, although a department manager may have 50 single invoice lines awaiting approval, they are grouped into one task line in the task list, under the task name (*Invoice approval* for example), and with a number indicating how many invoices these tasks are divided between. See Grouping of tasks for more details.

The formal process definition

Preparations

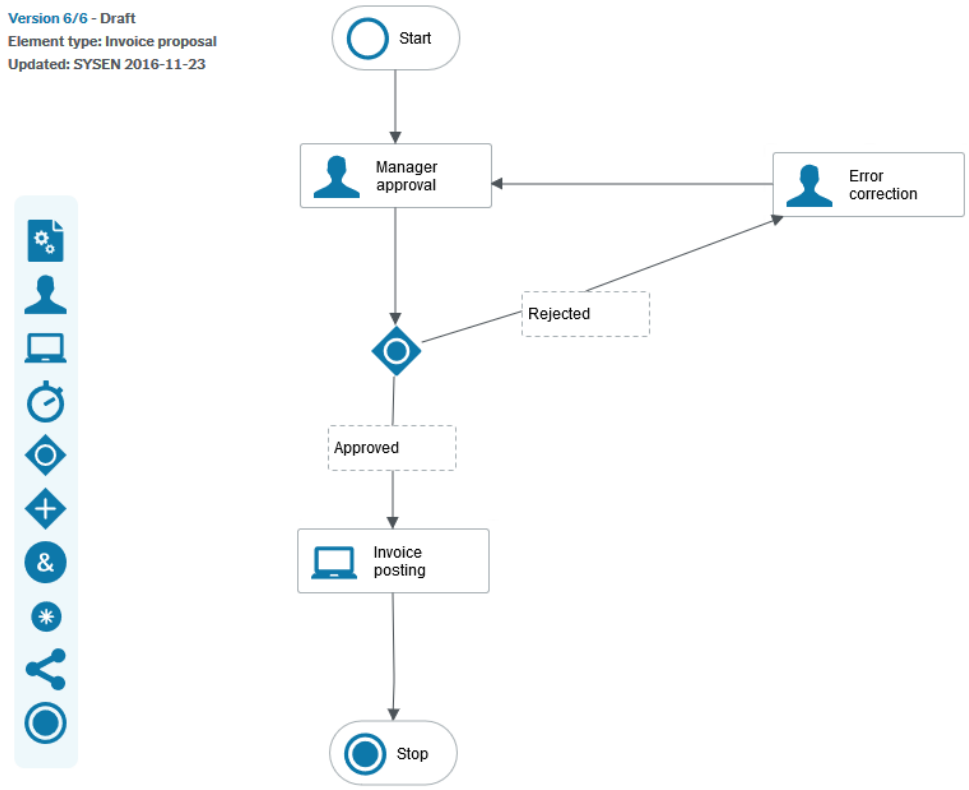
To transform the un-formal process diagram (previous page) into a working process definition in Business World, we need to make some preparations. The main considerations are related to the following:

- Element type for the invoice. Business World workflow processes are tailored for one or more element types. We need to find out which element type we must use in the process definitions.
- Finding the department manager. The various work items must be passed to the correct department manager for approval. Thereby we need to create a so-called *distribution rule* in Business World which allows us to do exactly that.
- Handle the alternative routes (for approval / rejection). We must ensure that the manager can select between approval and rejection, and that the process behaves according to intention.
- Screens and functions. We need to decide how the manager will "see" the work items, and which screen the user needs to make corrections. This is not handled automatically. In addition, we must tell the process how the invoice will be approved.

This doesn't stop us from drawing the process diagram, however,

The process definition diagram

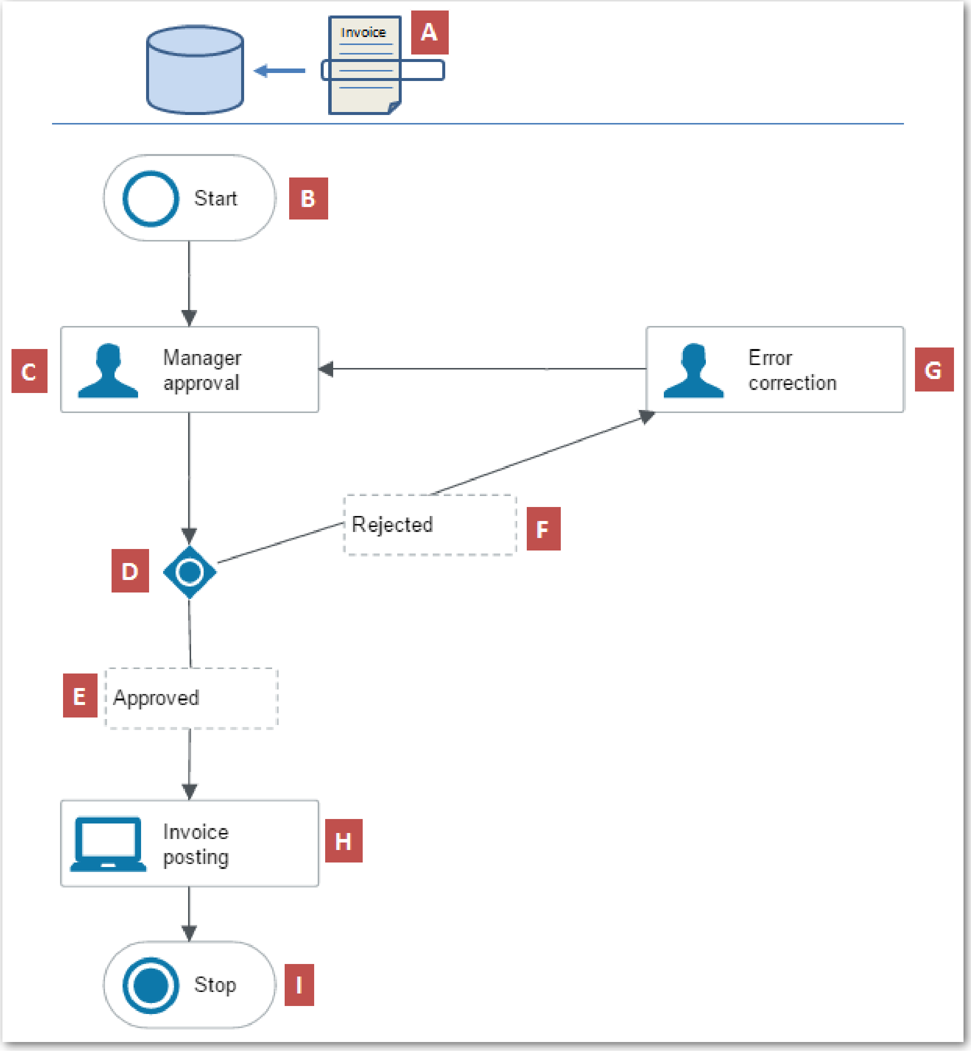
When we use the tools available in the **Process definition** window (under Common > Workflow), we can draw the following process diagram:



As you see, there is no task in the diagram where the user submits the invoice. A process is automatically instantiated when the workflow engine detects that there are un-processed workflow items. See the definition of Process instantiation. Therefore, we also needed to add the *Error correction* task as a separate user step.

Explaining the diagram

Below, we have added letters for each task in the process diagram. We have also included the actual invoice registration (A), to illustrate how a process will be instantiated.



The various tasks are explained below:

Task	Task type	Description
A	Initial invoice registration	When the invoice is saved, the workflow engine will create one process instance per invoice line.
B	Process start	The workflow engine grabs the invoice line, transforms it into a work item and starts executing the workflow process.
C	User step	The correct manager will get the work item, due to the rules found in the step's *distribution rule*. Before the work item is passed to C, the workflow engine will use the distribution logic to find the manager.
D	OR-split	This is not a task, but a mechanism that allows us to design alternative routes, based on a decision in the user step (C).
E / F	Branches	An OR-split branch represents an alternative route, based on certain

Task	Task type	Description
		criteria.
G	User step	The original user - the workflow initiator - will get the work item back for corrections. When the task is completed, the same distribution logic will pass the work item back to the manager.
H	System step	The server job will wait for all work items related to the same invoice to be approved, and then set a status flag on the invoice. A standard Business World server process (not part of the workflow) will later perform the actual posting.
I	Process stops	The current process instances - one per invoice line - are finished.

Details

Making it work

To make the process definition executable, we need to specify some important details:

- There are several element types for invoices, and we need to select one.
- All user steps must be connected to a distribution rule group, and we will take a brief look at how you create them.
- Business World workflow uses *actions* to complete a user step. We must ensure that the correct actions are available for the department manager and the workflow initiator.
- We must determine the screens to use for the two user steps (i.e. screens where the recipient will see the invoice details).
- We must explicitly link the correct server job to the system step.

The element type

Business WorldWorkflow can only handle work items defined as element types. Everyelement type has its characteristics and setup, and must be therefore studied before use(!).

There are several element types for invoices, and in our example, two may be of interest:

- Incoming invoice - a general invoice type, covering contractual invoices, supplier invoices etc.
- Supplier invoice - a specific element type for one invoice type only.

Treatment code for invoices: To be able to select the correct element type, we need to know how the invoice is registered, and more specific: the *treatment code* used for the invoice. The treatment code is - indirectly - selected when the user selects the *posting cycle* for the registration (a posting cycle is linked to a transaction type, which in turn is linked to a treatment code).

In our example, we know that the treatment code is **1**, and therefore we will use *Incoming invoice*.

> **Note:** You find the predefined treatment codes when you select the *TC* column in the **Transaction types** window and do a table lookup (F9 / F7). (Common >> Fixed registers >> Transaction types). You will also see the treatment code - if exists - when you look at the element type definition in the Element types window (also the *TC* column).

Distribution logic

To determine the actual recipient(s) of a workflow task, the workflow engine will use the connected distribution rule group. All user steps must be connected to a rule group, which will contain the necessary logic for locating recipients.

Since we don't have any rules so far, we need to create the ones we need, using the **Distribution rules** window in Common >> Workflow.

Identifying the manager: In order to identify the manager of a department, we will use the RESNO relation to the COSTC attribute (we know that a specific RESNO value for a given COSTC value represents the manager of that COSTC).

In the *Routing* table (in the Distribution rules window), we need the following line (only relevant columns are shown here):

---- Routing ----

	...	Find recipient	...	Attribute	Via relation	Recipient	Sequence	...
1		Relation on attribute		COSTC		RESNO	1	

This is the only rule we need for the manager, and we can save it in a new rule group.

Re-routing back to the workflow initiator or to the last owner: In the case where the manager rejects a work item, it can be sent back to the workflow initiator or to the previous owner of the task. To achieve that, we need no specific rule, since the Business World process definition allows us to send a work item directly to the Workflow initiator or to the last owner - via an OR-split branch (take a look at Process definition / OR-split setup. The options become available when you select an OR-split branch.)

Still, the user step *must* be connected to a rule group. For this purpose (and for all other situations where we really don't need a rule), we create a new rule group - No rules - and saves it without setting up any rules.

User step actions

A user step must always be set up with an action.

The manager will need two options - or actions: one for approval and for rejection. When we select the user step symbol in the process diagram, we find that these actions are selectable as predefined actions on the **Actions** tab.

The workflow initiator, on the other hand, needs to tell the workflow engine when the corrections are complete. For that purpose, we will just select the *Approve* action for the Error correction user step. When the user has corrected the error, he or she will use this button to pass the work item back to the manager.

Note: **Note 1:** *Approve* is a predefined action, and probably not the ideal action for this user step. Ideally, we must therefore create our own, custom action and name it *Corrections complete* - for example. But this is outside the scope of this example.

Note: **Note 2:** In Business World, an invoice line may be opened in VP10 (**Registration of incoming invoices**) for corrections. VP10 has a lot of hidden functionality, also related to workflow. If VP10 is used in our example, the user will not need an action button at all (but the system will still require that we define at least one action for the user step).

Screens used in user steps

All user steps in a process must have an assigned screen (otherwise it would not be possible for the user to view or edit the work item). Business World has a series of predefined screens - or windows - tailored for the various element types, and *Incoming invoice*, the most general invoice type, has several (when you select the user step in the diagram, you find available screens under Step setup / Usage):

- Supplier invoice approval
- Purchase invoice approval
- Contract invoice approval
- Registration of incoming invoices (VP10)
- Missing goods receipt

All these screens display the relevant data for viewing or editing the work item details. In our case, we can use **Supplier invoice approval** for the department manager and **VP10** for the workflow initiator. When the work item is presented to the manager, **Supplier invoice approval** will open with two buttons (**Approve** / **Reject**). When the work item is passed back to the workflow initiator, the invoice line will be ready for corrections in **VP10**, but there will be no visible **Approve** button. This is due to special functionality in **VP10**, where the **Save** command automatically trigger a so-called positive action - corresponding to a click on Approve.

Server job for the system step

When selecting the system step Invoice posting, we get access to all available server jobs for the current element type (the Function column in the Usage section). As you will see, there is currently only one pre-defined function: **Set invoice ready for auto-post**. We will use that.

Note: Business World uses various server processes to process transactional data. One such server process, **EI03 Posting of electronic invoices**, runs at fixed intervals and looks for transactions ready for posting. If there are any, the server process will perform the necessary operations, and perform the actual posting. Our system step, therefore, does not perform the actual posting, but it makes the work items ready for the EI03 server process.

Ready for testing

With all the above in place, the process definition is ready for testing. The **Validate** button (in the **Process definition** window) will check the definition, and inform you about any formal errors. When the validation is ok, you can activate the process definition (using the button in the action toolbar) and make it ready for testing.

Terminology

Workflow Management Coalition and Business World workflow

The terminology used in the Business World workflow system does not fully correspond with the terminology used by the Workflow Management Coalition (WfMC) in their “Terminology and Glossary” document (1994-1999). Sometimes, we use accepted synonyms (Activity = Task, for example), but we have also introduced a few, Business World specific terms.

General workflow concepts

Workflow management system

A Workflow management system defines, creates and manages the execution of workflow processes, and is based on one or more workflow engines.

In Business World, the Workflow management system consists of set of functionality for process definition, available from the Business World clients, while process execution and monitoring is taken care of by the Workflow Processing Server (the engine), running on the Business World server.

> **Note:** Note that Business World also has other kinds of automated processing, often based on IntellAgent. This *event based* processing, not executed by the workflow engine, is not part of the workflow management system.

Workflow engine

A Workflow engine is a software service that provides the runtime execution environment for a workflow process (instance).

In Business World, the workflow engine can be identified as the *Workflow Processing Server*, running on the Business World server, and managed via the Management Console.

Workflow process

A Workflow process is the automation of a business process, where information and tasks are automatically passed from one workflow participant to another, for action.

Sub-process

A sub-process is a workflow process which forms part of a main process.

In Business World Workflow, the following apply:

- You can define process definitions to be used only as sub-processes.
- A sub-process can only be instantiated from a main process instance.

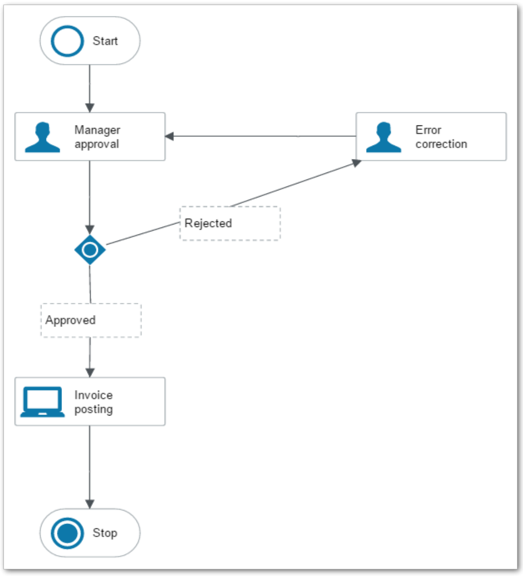
Concepts for process definition

Process definition

A process definition is a formalised model of a business process as a workflow process. A process definition identifies:

- The *work item type* subject to distribution and processing. A process definition is always tailored for one or more object types, determining what kind of business objects the process can handle.
- The *activities* (tasks) to be carried out.
- The *participants* in the various activities.

Example: A process definition in Business World is normally viewed as a diagram, with symbols for start and finish, the various activities and how they are connected:



As sub-process: You can define a process definition as a sub-process. A sub-process must always be instantiated through a main process.

Work item type = Element type in Business World

A work item type generally identifies the object type that is passed around in a process. In Business World, an object type that will be used for workflow processing, must have been defined as an element type in Business World.

See *Element type* below.

Element type

The Element type in Business World is a specialised object type, tailored to be handled by the workflow engine.

An element type behaves as a “wrapper” around a business object type, and presents a standardised interface to the workflow process. An element type wraps one designated object type.

Most of the object types relevant for workflow processing are already defined as element types in Business World.

Element types are described in detail below. See Element types.

Activity

An activity is usually referred to as *task* or *step*in theBusiness World Workflow. See below.

Task

A task is a piece of work that forms one logical *step* within a process definition. In a workflow process, a task will consist of a work item (an element type object) and some processing requirements.

We distinguish between three task types:

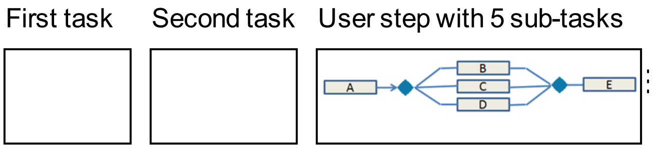
- Tasks for participants, i.e. where a user is defined as responsible for the task. In Business World, these are called *user steps*.
- Automated tasks, performed by the system (i.e. the Business World system). In Business World, these are referred to as *system steps*.
- Manual tasks, i.e. a task that cannot be handled by the workflow process, but will require some manual action (e.g. put a printout in a folder in a cabinet). If a manual task must be included in the process definition, it will be defined as a user step, with some instructions for the recipient regarding the manual operations.

Tasks within user steps: You can use distribution logic to split a task into several (sub-) tasks. See Sub-task below.

Sub-task

A sub-task (normally just called a task) occurs when the distribution logic for a user step divides the processing among several recipients. The recipients will all process the same work item, either in sequence or in parallel - or as a combination.

A sub-task is not visible in the process diagram, but can be identified by investigating the Distribution rule group. For example like this:



User step (Business World specific)

A *user step* is a workflow task where the task recipient (a workflow user) must carry out some actions on an element type object. The actions can be of two types:

- Formal and required action, selectable from a set of available, pre-defined actions. See Formal action below.
- Optional action – i.e. to modify the properties of the attached object.

Manual user step

A manual user step is like any other user step, but (also) intended to be used as an ad-hoc action (or single task) from an Action overview panel. It is therefore required that the step's element type(s) has support for an Action overview panel.

There are two types of manual user steps:

- An *independent user step* created solely to be used for ad-hoc or single tasks. You create an independent user step by selecting *Add user step* from the context menu of the **Manual user steps** folder in the Process tree (**Process definition** window).
- A user step which is part of a process definition and selected for manual use. In the **Step setup** tab (Shape details section) of the corresponding user step, select the *Step available for manual use* option. Manual user steps of this type will be available only when the process definition it belongs to is activated.

See also Action management in workflow.

System step (Business World specific)

A system step is a fully automated task, initiated by the workflow engine and performed by some standard Business World functionality, normally a server process.

Participant

A participant is a workflow user, responsible for a task. In Business World, a workflow participant is usually referred to as a *recipient* (the recipient of a workflow task, or a work item).

Recipient

A recipient is a workflow user responsible for a special task (or sub-task) in the process.

Synonyms: Task owner, task responsible.

Formal action

A formal action - or just action - is a function which can be linked to a user step and made available to the task owner via a button in the relevant screen.

We have two types of formal actions:

- System defined actions - a set of predefined functionality available for all user steps.
- Custom actions (referred to as User defined actions in Business World) - i.e. actions defined by your organisation and valid for one - or all - element types.

While a custom action is basically used to produce a *positive* or *negative* outcome as the result of a user step, some of the system actions are very sophisticated.

Normally, the task responsible will need to select one from a set of available formal actions to complete the user step.

Concepts for process in action

Process instance

A process instance is a running process, according to the process definition. It will have a very specific work item (e.g. invoice number 1234), specific activity instances and specific users.

While the definition can be seen as a “dormant” process model – or process template - a process instance is the process in action.

Process state

The current state of a process instance, expressed as a status value.

Workflow map

A workflow map is an Business World term for a graphical representation of an active or completed process instance, displaying the actual routing of the work item to specific recipients etc.

All recipients will have access to the workflow map via the screens they work with.

Activity instance = Task instance

A run-time activity. See Task instance below.

Task instance

A Task instance is a specific task, with a specific element type instance (work item) and processing requirements, that occurs within a process instance.

User step instance

A User step instance is a specific user step, with a specific element type instance and processing requirements, that occurs within a process instance.

Work item = Item (or Element type instance)

See Item instance below.

Item

An *item* is the object to be processed during an Task instance. In Business World, an item is always an element type instance.

Work list

See Task list below.

Task list

A list of task instances, associated with a given workflow participant.

Concepts for workflow process and distribution

Introduction

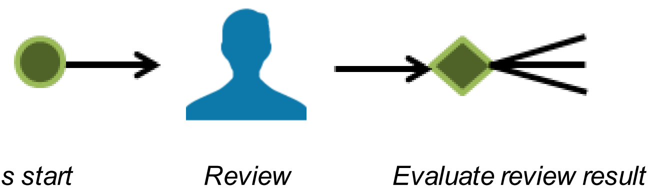
In Business World, the process is determined by two main factors:

- The *routing model*, which determines whether the workflow must continue directly with the next task), be split into several parallel tasks, or select a path based on a condition.
- The *distribution logic* defined for the receiving (next) task, determining the actual recipients as well as processing rules. Distribution logic is only relevant for user steps, and can either be handled automatically, using distribution rules, or manually.

In a running process (process instance), the routing from one task to another (or from one task to several others) is always according to the definition, as shown in the process definition diagram . When it comes to *user step instantiation*, however, the workflow may be much more complex, and dependent on the logic to use when the actual recipients must be identified.

Routing and distribution example

Assume the following process sequence:



Routing (or main workflow): A work item is always routed to the *Review* task (a user step).

When Review is completed, further routing will be determined on basis of a condition, based on the situation after Review.

This basic routing will always be followed, and is pretty straightforward.

Distribution: When this process runs, it will be the distribution logic set up for *Review* that determines the actual recipients of this task, and how the task will be processed. Although the diagram suggests a simple review task processed by a single user, a specific task instance may be very complex, and involve a series of recipients and sub-tasks.

This complexity is further explained in the Distribution of tasks section.

Distribution rule

A distribution rule holds the necessary logic for identifying recipients of a task. The purpose is to *find recipients*. If more than one recipient is identified for a task instance, the task will be distributed to all of them.

A rule will always belong to a rule group (see below).

Distribution rule group

A user step must always be linked a so-called *Distribution rule group* (or simply rule group), holding one or more distribution rules (see above).

When a task instance is about to be processed, the workflow engine will find the rule group it belongs to, and, for each rule on the group, compute the specific recipients.

The empty group: A rule group may be empty, i.e. have no rules. If you connect a user step to an empty rule group, you may identify the recipient(s) via two other methods:

- You can let the *responsible for the previous user step*, select recipients from a pre-defined list.
- You can assign the task to a recipient responsible for any previous task.

See Distribution of tasks for more details.

Distribution logic

The logic determining the recipients of a user step. In Business World the distribution logic is mainly expressed via a set of distribution rules, found in the distribution rule group attached to the step.

Manual distribution

Manual distribution means that the responsible for a user step must select the *next task*'s recipient(s). The next task must be a user step. Business World supports manual distribution in three cases only:

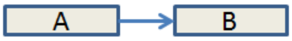
- After invoice registration in VP10 or TVP002.
- As part of a **Forward** action.
- As part of an **Inform** action.

Sequential routing

Sequential routing occurs when a process definition describes that a task must be succeeded by one next task.

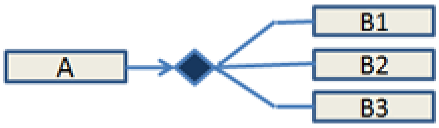
The routing can either be absolute, or made conditional, as illustrated below:

Absolute



B is always next task.

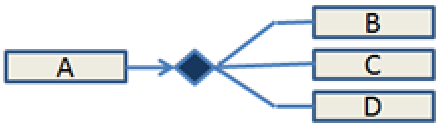
Conditional



Next task is either B1, B2 or B3 B2

Parallel routing

Parallel routing means that a task is succeeded by two or more tasks, processed (more or less) simultaneously. The diagram gives an illustration:

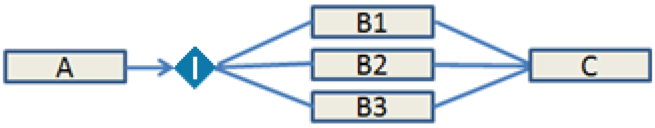


Tasks B, C and D are processed in parallel

OR-Split

An OR-split is a point within the workflow where the process will select one of a set of alternative tasks, dependent on a condition.

The diagram below means that the process will select one of the alternative routes (B1, B2, or B3), but always continue with C afterwards:

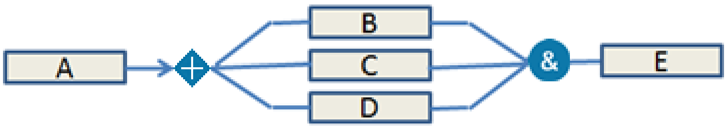


OR-split

Note: In most workflow systems, a so-called *OR-join* is required to assemble the alternative threads, before progressing to task C. In Business World, this OR-join is made automatically by the workflow engine and is not visible for the process designer.
(It is, however, no requirement that an OR-split is joined later in the process.)

AND-split and AND-join

An And-split is a point in the workflow where the workflow process is split into two or more parallel threads. In order to continue with sequential processing, we need an *AND-join* to assemble the threads (if that is what we want). This is illustrated below:



AND-split AND-join

Iteration

An iteration is a repetition of one or more tasks until a condition is met.

An iteration is usually designed via an OR-split branch, which takes the workflow back to a previous task. See also Selecting recipient in next step.

Escalation

Escalation refers to processing of user steps, where the original recipient is not found by the workflow engine, and the task is - via distribution rules - automatically forwarded (escalated) to somebody else.

You can use escalation to ensure that a user step are processed, and not sent to a supervisor, or not handled at all, when a recipient is not found.

Concepts used for Business World specific workflow control

Introduction

In Business World we use two specialised mechanisms for workflow control:

- A *Delay* is a system step that will not be completed or processed, before some pre-defined conditions are met.
- A *Synchronisation step* is a system step which behaves like a logical AND-split, but based on Business World specific functionality put into an element type.

Delay

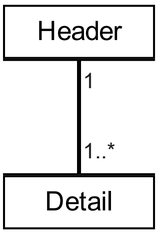
A Delay is a system step that monitors the item attached to the process, and delays execution until specific conditions are met.

The workflow engine will, at certain intervals, check the status of database objects identified by delay conditions in all process definitions and active process instances. When it finds that conditions are met for a process definition where the Delay is the first process task, it will start a new process instance. If it finds that the conditions are met for an active process instance, it will pass execution to the next task.

Example: A process for payment control is set up for Incoming invoices (the element type), but where the first task is a Delay node: The process will not start until 5 days before due date.

Synchronisation step

A synchronisation step is a system step which behaves like an AND-join, but on the basis of inherent functionality in the *transactional* element types, where the business object type is composed of at least two object types; a header object, and one or more detail objects:



Example: You have an invoice with a header object type and with one or more detail objects (invoice lines), which have to be approved by different recipients, and you want to wait until all the tasks are completed before the workflow continues. In this case, you will use a synchronisation step.

To fulfil general processing requirements, Business World has created specialised functionality for element types with this structure.

How it works: When one of these element types is used in a workflow process, the workflow engine will create one process instance per detail line.

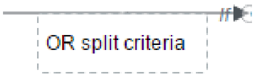
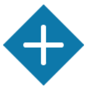
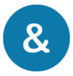
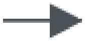
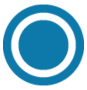
To assemble these process instances - for example before everything goes to a recipient for final approval - the process must use a synchronisation step before the workflow continues. As for a standard AND-join, the synchronisation step will not be completed until all the previous (parallel) tasks are finished.

Symbols

Symbols used in Process definition

Below, we give a short description of the main symbols used in Process definition. The links in the *Name* column go to the definitions:

Symbol	Name	Description
○	Process start	Indicates the start of a process definition.
👤	User step	A task which must be processed by one or more recipients. A user step will always be linked to a distribution rule group which will locate the actual recipients during a process instance.
💻	System step	A task which will be automatically processed by a server report or process in Unit4 Business World. There are no recipients involved.
✳	Synchronisation step	Step used to join a set of parallel process instances, after they all have been completed.
<	Publish step	Step used to send a message when an item will reach this step in a workflow process in Unit4 Business World. You are also able to include the object of the domain model in the message as well as specific properties.
⏱	Delay	This step pauses the process until a condition is met. A delay can be used as first task in the process, to filter out objects (work items) that fulfil certain criteria.
⚙	Sub-process	A process that can only be executed as part of a main process. The sub process must support the same element type(s) as the main process.
◈	OR-split	A point where the workflow process may take one of several routes depending on certain criteria. An OR-split will always be followed by two or more alternative paths.

Symbol	Name	Description
	OR-split criteria	Criteria -which can be of several types- attached to an OR-split branch, inside a label. The label is editable by the user, and is intended to have text that gives information of the criteria set for the branch.
	AND-split	A point where the workflow process will be split into two or more parallel threads. An AND-split has no special setup.
	AND-join	A point where a set of parallel threads - defined by an AND-split - will converge when all the parallel tasks are completed.
	Line	It is connects shapes in the diagram. Represents the direction where the workflow process goes. You can set the colour of the line. In the diagram, a line looks like an arrow.
	Stop	Stops the process instance. The Stop symbol can be used at various points in a process definition.

Symbols used in the workflow enquiry map

Following, a brief description of the main symbols used in Workflow enquiry. Note that some of them are also used in Process design.

Symbol	Name	Description
▶	Active	Step is currently active.
✋	Active parked	Step is currently parked, but active.
🚫	Aborted	Step has been aborted or cancelled.
🔔	Time out	For steps that develop within a time frame, if said time has expired, the system performs a pre-configured action.
⏱	Active delay	There is a current delay in the step.
⧉	Hide	Successive steps remain unseen for a given user, according to their configuration.
❗	Missing	Missing data, e.g.: missing distribution or missing map.
»	Waiting	Waiting for processing or forwarded.
⏫	Escalated	Escalated for further action.
✓	Finished	Processed without changes.
✓⚠	Finished with changes	Task finished with changes.

Symbol	Name	Description
⛔	Rejected	Step indicating rejection.
👥	Shared	The step has been shared with another user.
✳	Synchronisation	Step used to join a set of parallel process instances, after they all have been completed.
↻	Redistribution	Full redistribution of actions.

Colours indicate the status of the map elements. This is what they mean:

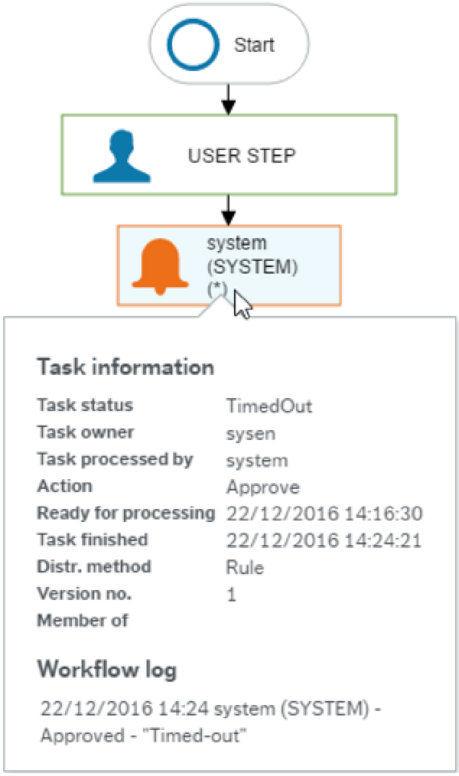
Colour	Meaning
Yellow	In progress
Green	Finished
Dark green	Automatically approved
Red	Negative action
Blue	Forwarded or escalated
Grey	Waiting for processing
Orange	Parked
Purple	Shared

Moreover, the line style also reveals the status of the map elements. Two line styles are used:

Line style	Meaning
Solid	Depending on the colour, it can indicate either Active, Finished, Waiting or Undefined steps.
Dotted	It indicates a deactivated status.

The next figure portrays an example of an workflow enquiry. In this case, the User Step is finished because the rectangle indicating the action is green. Then, the system step is in orange -which means parked-, with a bell symbol -denoting a time out-.

The tooltip provides more information about the status of the workflow. Once there is a time out, the system has performed automatically its pre-configured action, which in this example is approving.



Task information

Task status	TimedOut
Task owner	sysen
Task processed by	system
Action	Approve
Ready for processing	22/12/2016 14:16:30
Task finished	22/12/2016 14:24:21
Distr. method	Rule
Version no.	1
Member of	

Workflow log

22/12/2016 14:24 system (SYSTEM) - Approved - "Timed-out"

Workflow users

All users can be recipients

All registered Business World users can take part in a workflow process - per default.

Although you can choose to not to select the workflow checkbox for a certain user (the *Usage* section in **User master file**), this will not have any effect if the user is identified by a distribution rule.

When it comes to manual distribution, however, where a task owner explicitly assigns the task to a certain user, only those who are "workflow-enabled" will be available for selection.

This section

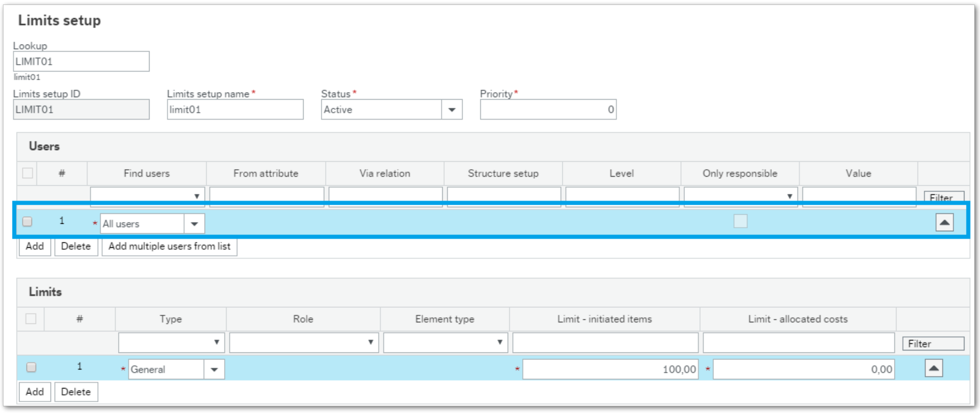
In this section we will describe how you can

- set up approval limits for work items where the
  - distribution is based on an amount and
  - the task recipient is "involved" in the amount to be approved, either as the one who initiated the process instance, or identified by the posting attributes for a transaction.
- assign potential substitutes for a workflow user, for example to be used when the user is on holiday.

Finding recipients on Limits, Supervisor and Substitute setup

It is possible to define limits, supervisors and substitutes for more than one user at once. There are 6 variants of defining users to whom Limits/substitutes/supervisors will be assigned to.

1. For **All users**.



Limits setup

Lookup: LIMIT01 (limit01)

Limits setup ID: LIMIT01

Limits setup name *: limit01

Status *: Active

Priority *: 0

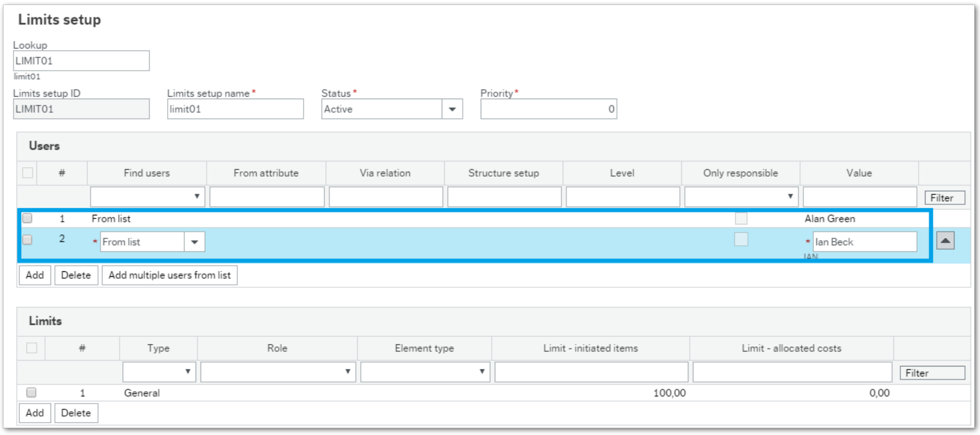
Users

□	#	Find users	From attribute	Via relation	Structure setup	Level	Only responsible	Value
□	1	* All users					□	

Limits

□	#	Type	Role	Element type	Limit - initiated items	Limit - allocated costs
□	1	* General			* 100,00	* 0,00

2. **From list** - you can define limits, supervisors and substitutes for a single user's **From list**.



Limits setup

Lookup: LIMIT01 (limit01)

Limits setup ID: LIMIT01

Limits setup name *: limit01

Status *: Active

Priority *: 0

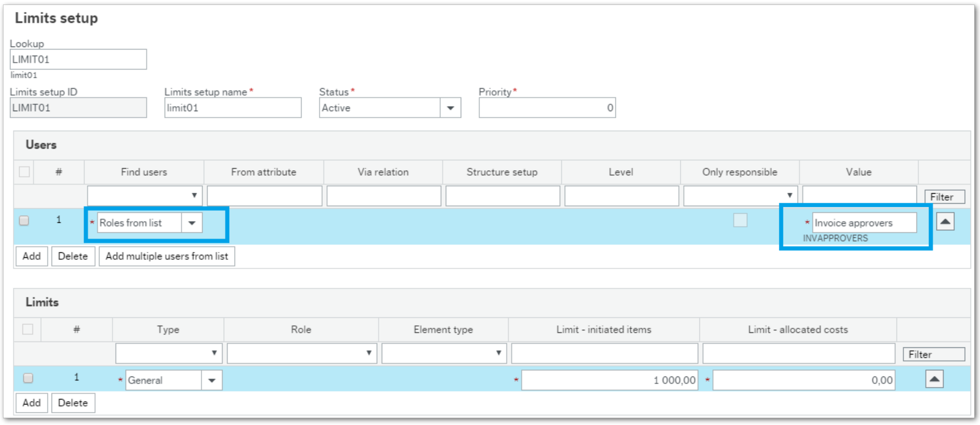
Users

□	#	Find users	From attribute	Via relation	Structure setup	Level	Only responsible	Value
□	1	From list					□	Alan Green
□	2	* From list					□	* Ian Beck IAN

Limits

□	#	Type	Role	Element type	Limit - initiated items	Limit - allocated costs
□	1	General			100,00	0,00

3. **Roles from list** - for all users assigned to the selected role.



Limits setup

Lookup: LIMIT01
limit01

Limits setup ID: LIMIT01
Limits setup name*: limit01
Status*: Active
Priority*: 0

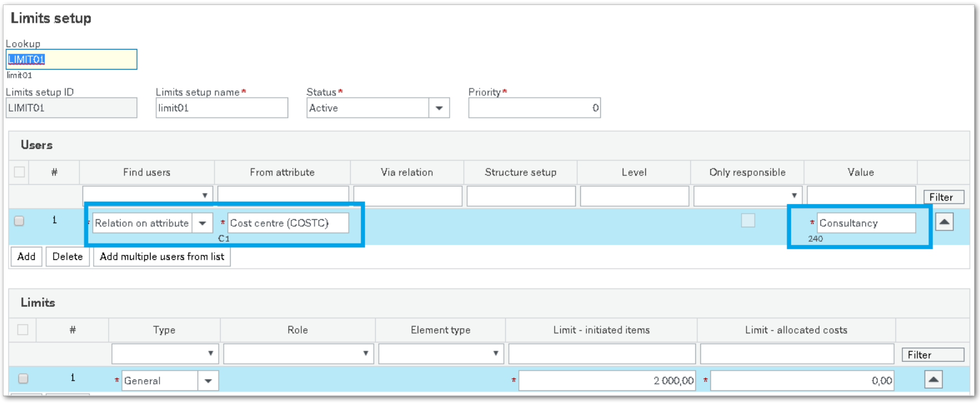
Users

☐	#	Find users	From attribute	Via relation	Structure setup	Level	Only responsible	Value	
									Filter
☐	1	* Roles from list					☐	* Invoice approvers INVAPPROVERS	

Limits

☐	#	Type	Role	Element type	Limit - initiated items	Limit - allocated costs	
							Filter
☐	1	* General			* 1 000,00	* 0,00	

4. **Relation on attribute** - you can use specific relations to identify a group of users, for instance, a limit for all resources from Cost centre 240 can be defined at once.



Limits setup

Lookup: LIMIT01
limit01

Limits setup ID: LIMIT01
Limits setup name*: limit01
Status*: Active
Priority*: 0

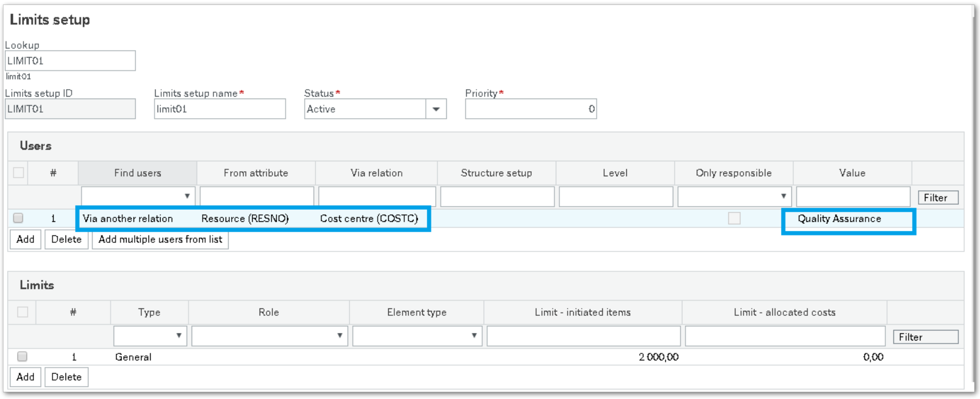
Users

☐	#	Find users	From attribute	Via relation	Structure setup	Level	Only responsible	Value	
									Filter
☐	1	* Relation on attribute	* Cost centre (COSTC) C1				☐	* Consultancy 240	

Limits

☐	#	Type	Role	Element type	Limit - initiated items	Limit - allocated costs	
							Filter
☐	1	* General			* 2 000,00	* 0,00	

5. **Via another relation** - you can use specific relations via another relation to identify a group of users.



Limits setup

Lookup: LIMIT01
limit01

Limits setup ID: LIMIT01
Limits setup name*: limit01
Status*: Active
Priority*: 0

Users

☐	#	Find users	From attribute	Via relation	Structure setup	Level	Only responsible	Value	
									Filter
☐	1	Via another relation	Resource (RESNO)	Cost centre (COSTC)			☐	Quality Assurance	

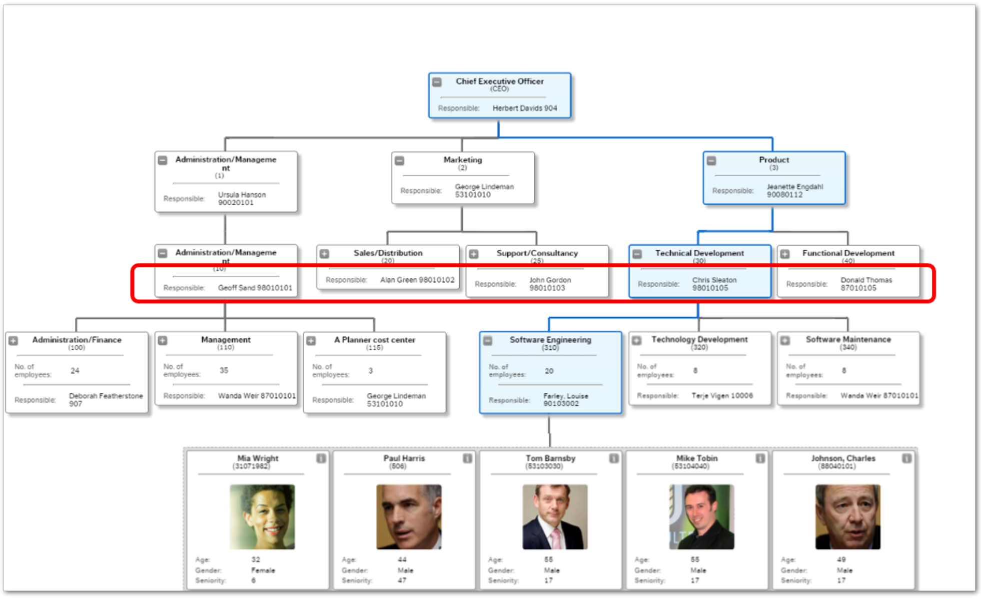
Limits

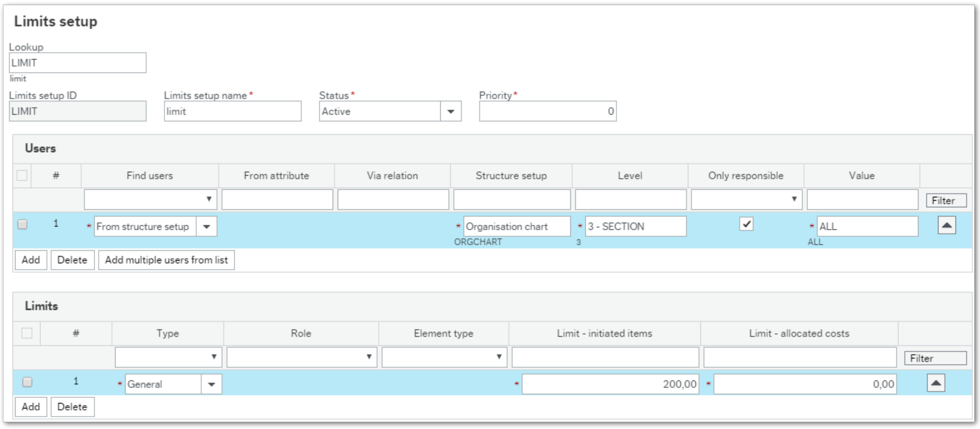
☐	#	Type	Role	Element type	Limit - initiated items	Limit - allocated costs	
							Filter
☐	1	General			2 000,00	0,00	

6. **From structure setup** - you can define the limits setup, supervisor setup or substitutes for:

   - Fixed level, for instance, the manager of the whole section.

> **Note:** Note that the **Only responsible** check box must be selected.


Chief Executive Officer (CEO)
Responsible: Herbert Davids 904
Administration/Management (1)
Responsible: Ursula Hanson 90020101
Marketing (2)
Responsible: George Lindeman 53101010
Product (3)
Responsible: Jeanette Engdahl 90080112
Administration/Management (10)
Responsible: Geoff Sand 98010101
Sales/Distribution (20)
Responsible: Alan Green 98010102
Support/Consultancy (25)
Responsible: John Gordon 98010103
Technical Development (30)
Responsible: Chris Sleaton 98010105
Functional Development (40)
Responsible: Donald Thomas 87010105
Administration/Finance (100)
No. of employees: 24
Responsible: Deborah Featherstone 907
Management (110)
No. of employees: 35
Responsible: Wanda Weir 87010101
A Planner cost center (115)
No. of employees: 3
Responsible: George Lindeman 53101010
Software Engineering (310)
No. of employees: 20
Responsible: Farley, Louise 90103002
Technology Development (320)
No. of employees: 8
Responsible: Terje Vigen 10006
Software Maintenance (340)
No. of employees: 8
Responsible: Wanda Weir 87010101
Mia Wright (31071982)
Age: 32
Gender: Female
Seniority: 6
Paul Harris (506)
Age: 44
Gender: Male
Seniority: 47
Tom Barnsby (53103030)
Age: 55
Gender: Male
Seniority: 17
Mike Tobin (53104040)
Age: 55
Gender: Male
Seniority: 17
Johnson, Charles (88040101)
Age: 49
Gender: Male
Seniority: 17



Limits setup

Lookup: LIMIT
limit

Limits setup ID: LIMIT

Limits setup name*: limit

Status*: Active

Priority*: 0

Users

□	#	Find users	From attribute	Via relation	Structure setup	Level	Only responsible	Value	
									Filter
□	1	* From structure setup			* Organisation chart ORGCHART	* 3 - SECTION 3	☑	* ALL ALL	

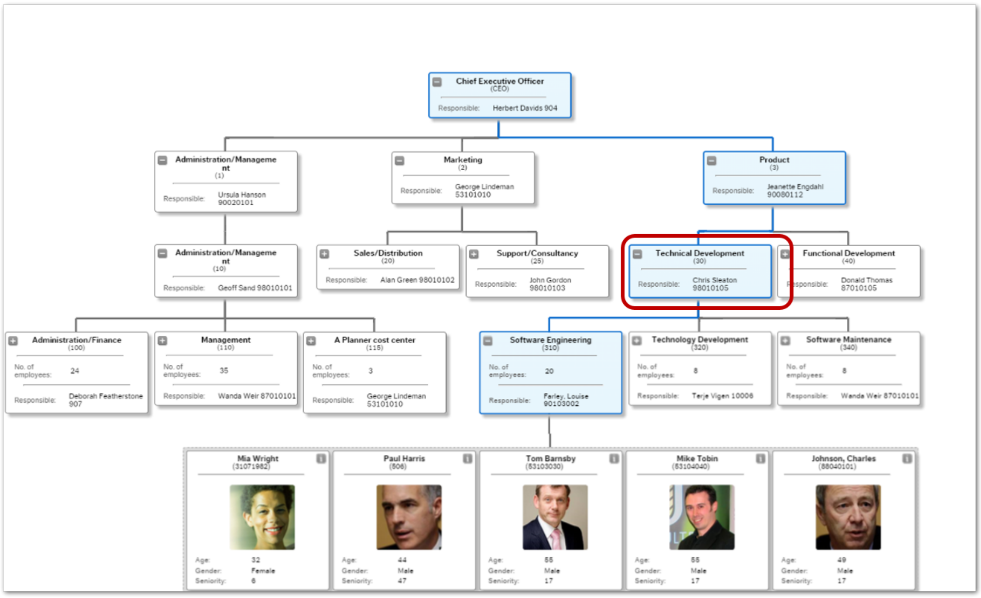
Add | Delete | Add multiple users from list

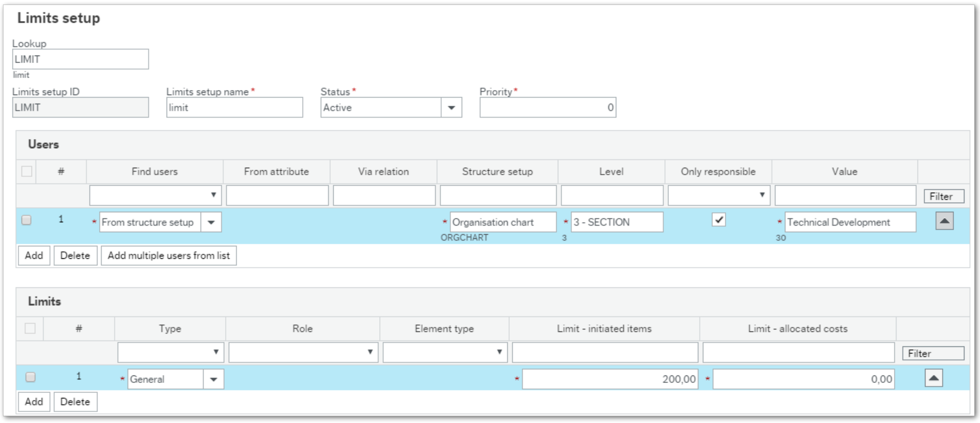
Limits

□	#	Type	Role	Element type	Limit - initiated items	Limit - allocated costs	
							Filter
□	1	* General			* 200,00	* 0,00	

Add | Delete

- Only for the responsible person from the selected section.





Limits setup

Lookup
LIMIT
limit

Limits setup ID: LIMIT
Limits setup name *: limit
Status *: Active
Priority *: 0

Users

□	#	Find users	From attribute	Via relation	Structure setup	Level	Only responsible	Value	
		▾					▾		Filter
□	1	* From structure setup ▾			* Organisation chart ORGCHART	* 3 - SECTION 3	☑	* Technical Development 30	▲

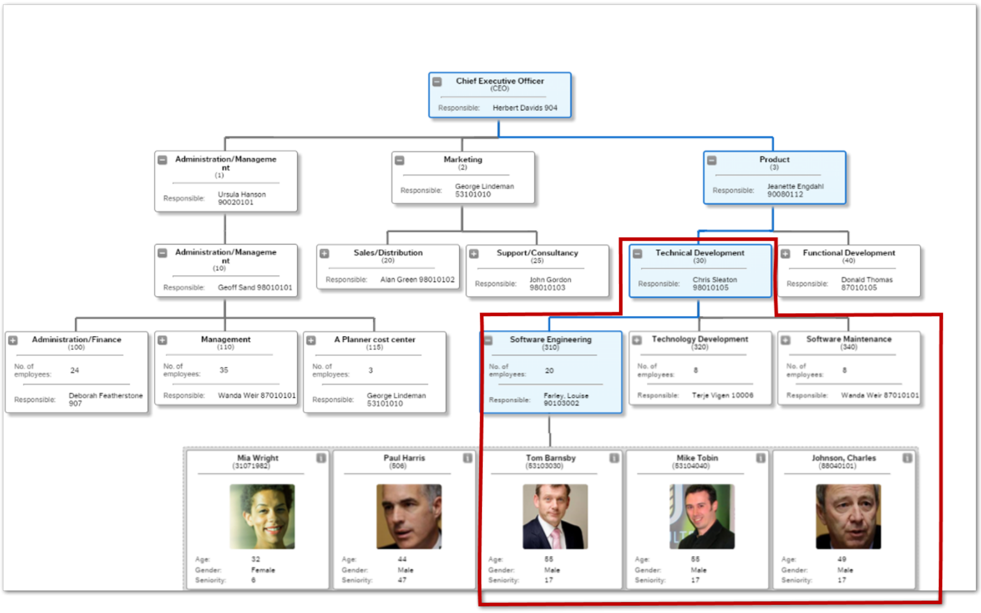
Add | Delete | Add multiple users from list

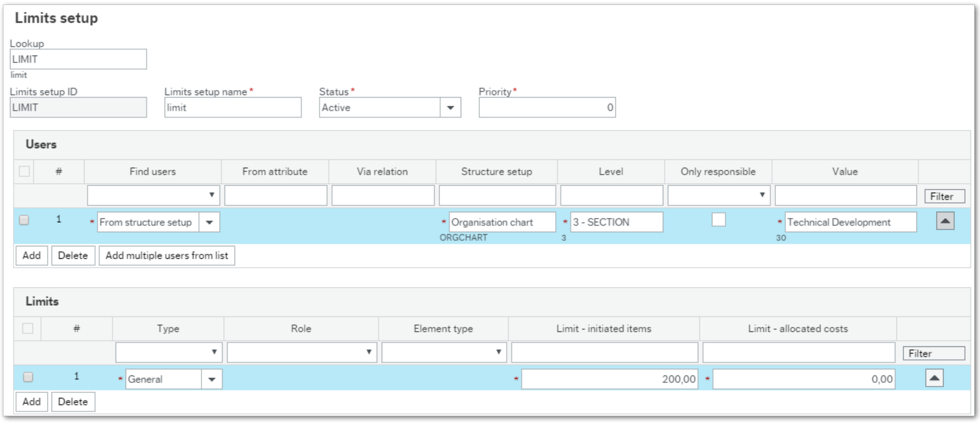
Limits

□	#	Type	Role	Element type	Limit - initiated items	Limit - allocated costs	
		▾	▾	▾			Filter
□	1	* General ▾			* 200,00	* 0,00	▲

Add | Delete

- For the selected section and everything below e.g. Technical Development (40) and everything below (resources). The responsible person is excluded.





Limits setup

Lookup: LIMIT
limit

Limits setup ID: LIMIT

Limits setup name*: limit

Status*: Active

Priority*: 0

Users

☐	#	Find users	From attribute	Via relation	Structure setup	Level	Only responsible	Value	
		▾					▾		Filter
☐	1	* From structure setup			* Organisation chart ORGCHART	* 3 - SECTION 3	☐	* Technical Development 30	▲

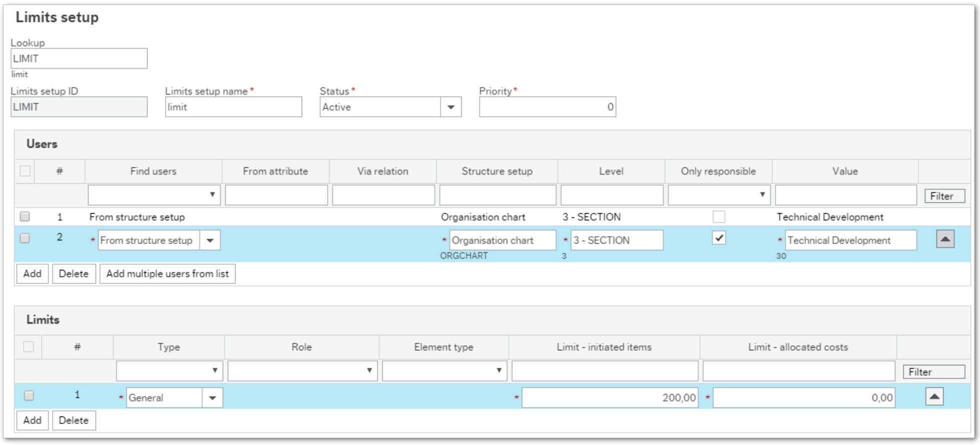
Add | Delete | Add multiple users from list

Limits

☐	#	Type	Role	Element type	Limit - initiated items	Limit - allocated costs	
		▾	▾	▾			Filter
☐	1	* General			* 200,00	* 0,00	▲

Add | Delete

- For resources from the section and the person responsible for it. Two rows have to be added: *for selected section* and *for responsible person*.



Limits setup

Lookup	LIMIT limit
Limits setup ID	LIMIT
Limits setup name*	limit
Status*	Active
Priority*	0

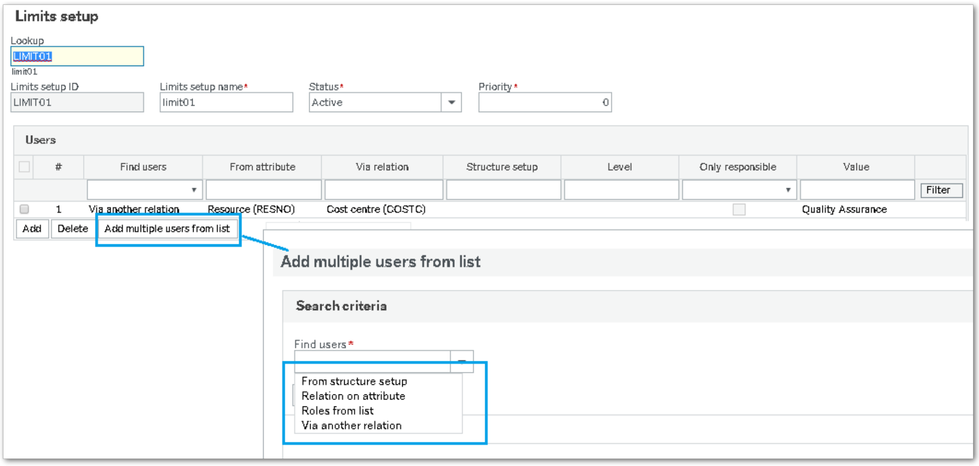
Users

□	#	Find users	From attribute	Via relation	Structure setup	Level	Only responsible	Value
□	1	From structure setup			Organisation chart	3 - SECTION	□	Technical Development
□	2	* From structure setup			* Organisation chart ORGCHART	* 3 - SECTION 3	☑	* Technical Development 30

Limits

□	#	Type	Role	Element type	Limit - initiated items	Limit - allocated costs
□	1	* General			* 200,00	* 0,00

You can also add multiple users from the list by using a dedicated action. All setups described above, excluding **All** and **From list**, are available in the **Add multiple users from list** window:



Limits setup

Lookup	LIMIT01 limit01
Limits setup ID	LIMIT01
Limits setup name*	limit01
Status*	Active
Priority*	0

Users

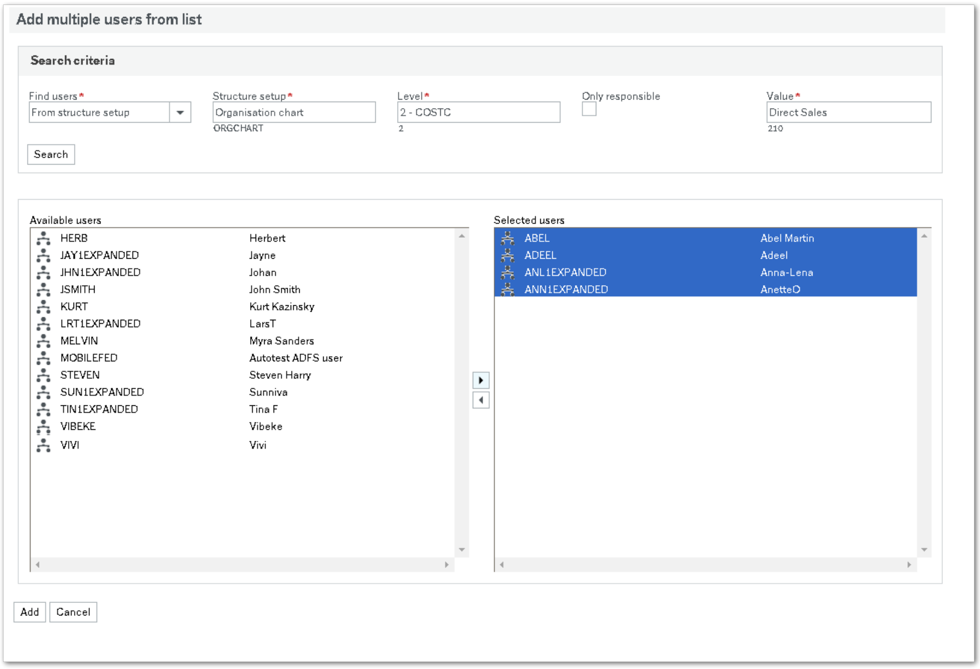
□	#	Find users	From attribute	Via relation	Structure setup	Level	Only responsible	Value
□	1	Via another relation	Resource (RESNO)	Cost centre (COSTC)			□	Quality Assurance

Add multiple users from list

Search criteria

Find users*

The selected users are displayed on the grid as the **From list** users.



Add multiple users from list

Search criteria

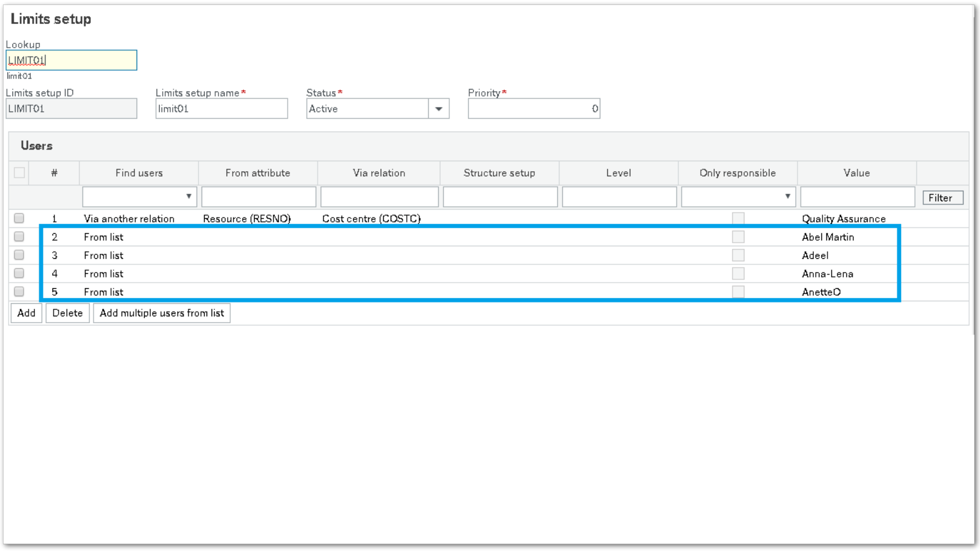
Find users*	Structure setup*	Level*	Only responsible	Value*
From structure setup	Organisation chart ORGCHART	2 - COSTC 2	□	Direct Sales 210

Available users

HERB	Herbert
JAY1EXPANDED	Jayne
JHN1EXPANDED	Johan
JSMITH	John Smith
KURT	Kurt Kazinsky
LRT1EXPANDED	LarsT
MELVIN	Myra Sanders
MOBILEFED	Autotest ADFS user
STEVEN	Steven Harry
SUN1EXPANDED	Sunniva
TIN1EXPANDED	Tina F
VIBEKE	Vibeke
VIVI	Vivi

Selected users

ABEL	Abel Martin
ADEEL	Adeel
ANL1EXPANDED	Anna-Lena
ANN1EXPANDED	AnetteO



Limits setup

Lookup: LIMIT01
limit01

Limits setup ID	Limits setup name*	Status*	Priority*
LIMIT01	limit01	Active	0

Users

□	#	Find users	From attribute	Via relation	Structure setup	Level	Only responsible	Value
□	1	Via another relation	Resource (RESNO)	Cost centre (COSTC)			□	Quality Assurance
□	2	From list					□	Abel Martin
□	3	From list					□	Adeel
□	4	From list					□	Anna-Lena
□	5	From list					□	AnetteO

Limits and supervisor setup

General description

You use **Limits and supervisor setup** to define the *amounts* a user will be authorised to approve during a workflow process instance, and, if the system finds out that authorisation is not in order, who will receive the work item as supervisor.

> **Note:** Note: The limits are in principle general, and can be used for all numeric properties you can base a distribution rule on. The intended use, however, is for amount properties.

Amounts and limits

The setup only takes effect when an identified recipient of a task instance is directly related to an amount value of a work item, and where the distribution rule is based on this amount type. The work item may either:

- be initiated (i.e. saved to the database) by this user (a new purchase order for example), or
- contain an amount which must be *paid to*, or *paid on behalf of*, the user. In this case the workflow engine will find the user via a RESNO value (or USER value) which *is included in the posting attributes* on the work item (a phone bill for the employee, for example). See description of Limits.

Supervisors

A supervisor will receive work items the identified recipient is not authorised to handle, as well as work items the recipient fails - or forgets - to handle (the work item remains un-processed until a deadline).

If there are no defined supervisors, and no defined escalation logic (see below), the process instance will abort, and the work item will be marked as not handled (and transferred to Items to follow up).

Automatic escalation

When defining distribution rules, you can both set up alternative routing on basis of amount values, as well as automatic escalation to specific recipients if tasks are not handled. In many cases, the use of an escalation sequence can be a flexible alternative to supervisors.

In cases where you have defined both a supervisor and an escalation sequence, the escalation logic in the distribution rule will be executed first.

Limits

Validity of limits

When you define a set of limits (one new row), you must first select the basis for the limit, in the Type column. The following options are available:

- Element type: the row will be used when the user receives a work item of the selected element type - unless the distribution rule located the recipient as member of a role.
- Role: will be used if the distribution rule located the user as part of a role.
- General: will always be valid.

Precedence

You can specify various limits (many rows), of different types. If more than one row is valid, only the row with the highest precedence will be used:

- If the user received the task as member of an identified role, the limits will be taken from the matching *Role* row, regardless of element type.
- If the user receives a work item belonging to an identified element type, and the distribution is not based on role membership, or there are no rows matching the role membership, the limits will be taken from the relevant *Element type* row.
- If neither Role or Element type is applicable, the limits will be taken from the *General* row.

The zero (0) limit

If a user is set up with no limits, or if a row contains **0** in a limit column, the user will not be authorised to handle work items of this kind (where distribution is based on an amount value).

Supervisors

The user type

The Type column in the supervisor table refers to the task recipient and the current work item, as described for Limits. See Validity of limits.

You can therefore, assign different supervisors to the same user, dependent on - and matching - the basis of the defined limits.

Types

You can define supervisors of three types, here described in prioritised sequence:

- Role - valid if the user received the task as member of a defined role. You can set up one supervisor per role.
- Element type - valid if the work item is of an identified element type (and the task is not received due to a defined role membership). You can set up one supervisor per element type.
- General - valid if there are no other matching supervisors.

Example: The diagram below gives an illustration:

	Type	Role	Element type	Find supervisor
1	Role	Responsible		...
2	Role	SalesPerson		...
3	Element type		Incoming invoice	...
4	General			...

Locating supervisors

Regardless of the *Type*, you use the following columns to locate a specific supervisor:

...	Find supervisor	Attribute	Via relation	Supervisor	Date from	Date to
...						

Validity period

A supervisor has a limited validity period, defined by the *Date from* and *Date to* columns. If you want to keep track of historical changes, you can set up different supervisors for different time periods - for the same user and *Type*.

The Find supervisor column

The *Find supervisor* column gives you three options to locate a supervisor:

Option	Description
From list	You select the supervisor directly in the *Supervisor* column.
Relation on attribute	Currently, the only selectable attribute is RESNO - which per definition will identify the original recipient. Currently, you can only select RESNO in the Supervisor column also, meaning that
Via another relation	The only selectable attribute is RESNO - the original recipient. You select a related (parent) attribute to RESNO in *Via relation*, for example representing a superior level in the organisation, and then the RESNO attribute in *Supervisor*, identifying the manager of the selected organisational unit.

Substitutes for task recipients

Substitutes

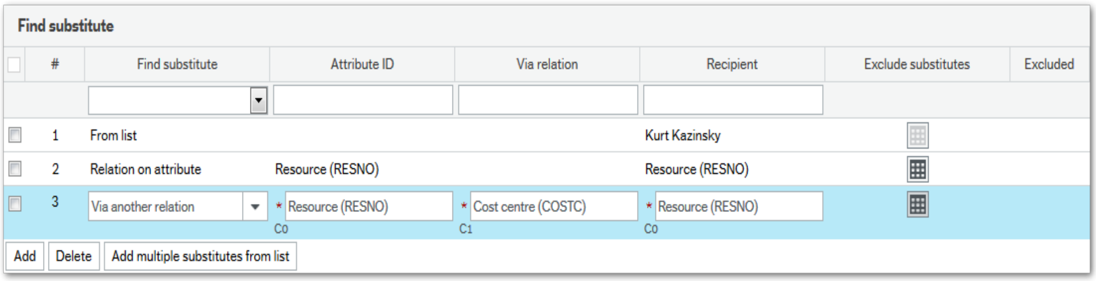
A *substitute* is a Business World user who will receive workflow tasks on behalf of the original recipient - as defined in the distribution rules for workflow process.

Substitutes are used to avoid hang-ups in running workflow processes, and a given user can have several potential substitutes. Substitutes are set up as follows:

1. An administrator (you) will first define a number of users that can be used as substitutes for a list of given workflow users. This is done in the **Administer workflow substitutes** window.
2. When a workflow user goes for holiday - or in any other way knows that he or she will be unavailable for a period of time - the workflow user can select specific substitutes in the **Activate your substitutes** window (the available substitutes will be those previously defined).
3. If a workflow user becomes sick, or has forgotten to set up substitutes before a period of absence, *another authorised user* can use the **Assign substitutes** window to set up substitutes on the workflow user's behalf.

The Find substitute section

You use the *Find substitute* section to pick the substitutes. The table columns are shown below, with the various Find options filled in:



Find substitute

□	#	Find substitute	Attribute ID	Via relation	Recipient	Exclude substitutes	Excluded
□	1	From list			Kurt Kazinsky	▦	
□	2	Relation on attribute	Resource (RESNO)		Resource (RESNO)	▦	
□	3	Via another relation	* Resource (RESNO) C0	* Cost centre (COSTC) C1	* Resource (RESNO) C0	▦	

The Find logic

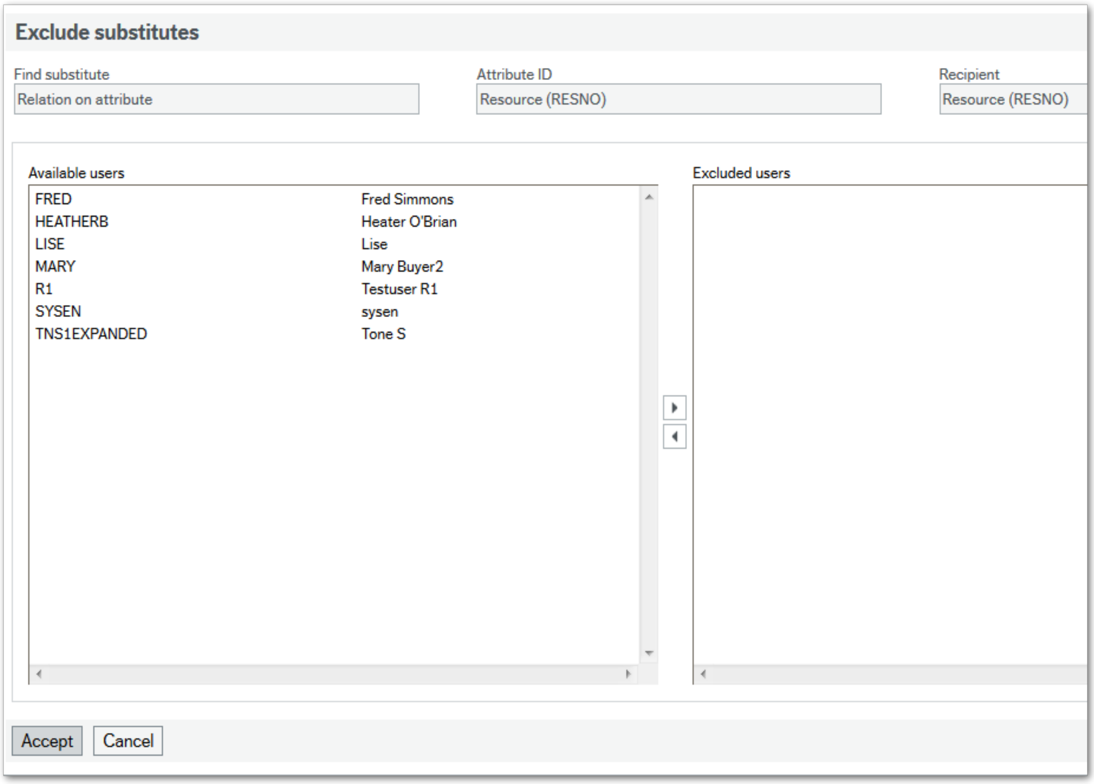
To select potential substitutes for workflow users, you have several options:

- You can choose among all defined workflow users (**From list**). This is straightforward, and you simply select the substitute you want in the *Recipient* column.

- You can let the system find another user with a defined relation to the workflow user (**Relation on attribute**). In the example above, we use the RESNO relation on the current user as a starting point (*Attribute ID*), and then identifies the potential substitute via the RESNO relation on the identified resource (this requires a convention, for example that the RESNO relation on RESNO identifies the immediate superior).
- You can let the system find other users via a related attribute (**Via another relation**) . Above, we use the RESNO relation on USER to get the resource record for the current user (*Attribute ID*). Then we use the COSTC relation on RESNO to identify the department (by convention, COSTC = department). The RESNO in *Recipient* will then identify the department manager as the potential substitute (also a convention).
- You can add multiple rows with **From list** by using the **Add multiple substitutes from list** button. It works in the same way as the **Add multiple user from list** button.

> **Note:** The **Add multiple user from list** button is located in the **Users** section. This section is available in the following windows: **Limits setup**, **Supervisor setup** and **Administer workflow substitutes** windows.

How the button in Exclude substitutes works



Exclude substitutes

Find substitute: Relation on attribute

Attribute ID: Resource (RESNO)

Recipient: Resource (RESNO)

Available users

FRED	Fred Simmons
HEATHERB	Heater O'Brian
LISE	Lise
MARY	Mary Buyer2
R1	Testuser R1
SYSEN	sysen
TNS1EXPANDED	Tone S

Excluded users

1. The button in the *Exclude substitutes* column is available for **Relation on Attribute** and **Via another relation** buttons.
2. Clicking the button from the *Exclude substitutes* column opens the **Exclude substitutes** window.
3. **Available users** contains the sum of all available substitutes for all users from the **Users** section.
4. Moving the selected user into **Excluded users** and accepting the changes results in displaying an icon in the *Excluded* column and this particular user will no longer be available as a substitute in the **Activate your substitutes** window.

Alerts in workflow

Reminders and notifications

A recipient will always get access to pending workflow tasks via the Task list.

Sometimes you may want to stress the importance of a task, and notify the recipient via other means. This can be relevant when the deadline for a task is approaching, or if a task has been forwarded from the original user to somebody else.

Generally, Business World uses **IntellAgent** to handle notifications (alerts) based on the occurrence of a certain event. Workflow events, however, are not available via the standard IntellAgent functionality (System administration > IntellAgent). The **Alert setup** window (under Common > Workflow > Alerts) is the only place where you can define alerts for Workflow initiated events.

Alert types for Workflow events

There are two major event types which can trigger an IntellAgent alert:

- When a recipient receives a task. This can be the original recipient, or the final responsible for the task if it has been transferred to somebody else.
- When you have defined a deadline condition where you want notify the recipient (original or alternative) or a supervisor about the situation.

Alert validity: An alert can either be valid for one element type, or it can be general, i.e. valid for all element types. You can - for each workflow event - define different alerts for different element types, and have one general as well. The latter will be used when the current element type has no tailored alert.

Alert templates

An Alert template identifies the alert content (message) and how it must be presented (the layout). You can design different layout, using different content, for the various alert media (Email, SMS/Text message, Alert list). All alerts must be linked to a template.

Template content

When you create an alert template for a special element type, you can use up to 10 element type properties as part of the alert message. The properties must have been selected for this purpose in theElement type detailstab (in the**Element types**window).

Further, the selected properties must have been explicitly linked (mapped) to generic tags (named **Data1** - **Data10**), which you can use as fields in the template editor. See Mapping of generic tags for details.

Default tags

In addition to mapped tags, you can use the following default tags - available for all element types (the tag brackets are not shown):

Tag	Represents
Alert recipient	Full name of the target user.
Client	The Business World company where the workflow event occurred.
Distr. method	How the task was distributed: • according to a rule (Rule), • manually distributed (Manual), • forwarded (Forward), • escalated (Escalation).
Element type	Name of element type for current work item.
Event name	Name of the Workflow event (incident) that triggered the alert.
Instance number	The number of times this alert has been triggered for the current client (can be a very high number).
Link	A hyperlink to the relevant page for task processing.
Original user	The first task owner (may also be the current task owner).
Step	Name of the current user step - as set up in the process definition.
Sub-process	Name of the process definition.
Task owner	Current task owner.
Today	Today's date.

Alert setup

IntellAgent for Workflow events

The Alert setup window is used to manage **IntellAgent** responses to workflow initiated events - not possible to set up using standard IntellAgent functionality (System administration > IntellAgent).

Alternative notification of pending workflow tasks

A Workflow task will always be available in the recipients task list. Using alerts, you can notify the original recipient - or other interested parties - via Email , Text messages, or the Alert list.

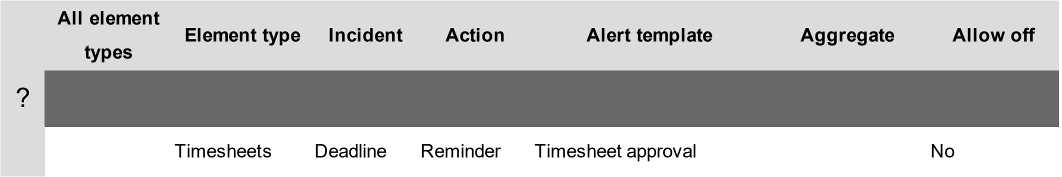
Workflow events

Currently, you can set up alerts for the following workflow events (called *incidents* in **Alert setup**):

- When a new task is available in the task list for the original recipient. The recipient receives the alert.
- When a task is transferred to a substitute. The substitute receives the alert.
- When a system defined action - Forward or Escalate - is carried out. The new recipient will receive the alert.
- When a deadline is reached and you want to notify the original recipient, or the new recipient , when the task has been escalated.

Window description

Alert setup mainly consists of a table, where one row represents one alert for a certain event (incident).



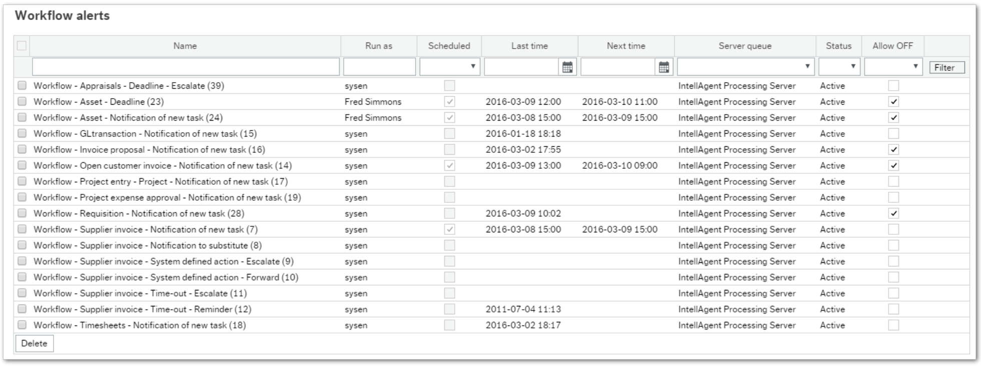
	All element types	Element type	Incident	Action	Alert template	Aggregate	Allow off
?							
		Timesheets	Deadline	Reminder	Timesheet approval		No

When you have saved a new row, you cannot modify the settings, but you can delete it and create a new one.

Workflow alerts TIN005 - Web client

Window usage

This window is used to enquire about the alerts, namely IntellAgent events not initiated by workflow processes. You can see when they were last run and in case of scheduled alerts - when they will be run next. It is possible to change the status of an alert, the server queue and also choose whether the recipient is allowed to switch off all alerts. In addition, you can delete selected alerts and all the alert history of their occurrences. The **Workflow alerts** window is displayed below:



Workflow alerts

□	Name	Run as	Scheduled	Last time	Next time	Server queue	Status	Allow OFF	
			▾			▾	▾	▾	Filter
□	Workflow - Appraisals - Deadline - Escalate (39)	sysen	□			IntellAgent Processing Server	Active	□	
□	Workflow - Asset - Deadline (23)	Fred Simmons	☑	2016-03-09 12:00	2016-03-10 11:00	IntellAgent Processing Server	Active	☑	
□	Workflow - Asset - Notification of new task (24)	Fred Simmons	☑	2016-03-08 15:00	2016-03-09 15:00	IntellAgent Processing Server	Active	☑	
□	Workflow - GLtransaction - Notification of new task (15)	sysen	□	2016-01-18 18:18		IntellAgent Processing Server	Active	□	
□	Workflow - Invoice proposal - Notification of new task (16)	sysen	□	2016-03-02 17:55		IntellAgent Processing Server	Active	☑	
□	Workflow - Open customer invoice - Notification of new task (14)	sysen	☑	2016-03-09 13:00	2016-03-10 09:00	IntellAgent Processing Server	Active	☑	
□	Workflow - Project entry - Project - Notification of new task (17)	sysen	□			IntellAgent Processing Server	Active	□	
□	Workflow - Project expense approval - Notification of new task (19)	sysen	□			IntellAgent Processing Server	Active	□	
□	Workflow - Requisition - Notification of new task (28)	sysen	□	2016-03-09 10:02		IntellAgent Processing Server	Active	☑	
□	Workflow - Supplier invoice - Notification of new task (7)	sysen	☑	2016-03-08 15:00	2016-03-09 15:00	IntellAgent Processing Server	Active	□	
□	Workflow - Supplier invoice - Notification to substitute (8)	sysen	□			IntellAgent Processing Server	Active	□	
□	Workflow - Supplier invoice - System defined action - Escalate (9)	sysen	□			IntellAgent Processing Server	Active	□	
□	Workflow - Supplier invoice - System defined action - Forward (10)	sysen	□			IntellAgent Processing Server	Active	□	
□	Workflow - Supplier invoice - Time-out - Escalate (11)	sysen	□			IntellAgent Processing Server	Active	□	
□	Workflow - Supplier invoice - Time-out - Reminder (12)	sysen	□	2011-07-04 11:13		IntellAgent Processing Server	Active	□	
□	Workflow - Timesheets - Notification of new task (18)	sysen	□	2016-03-02 18:17		IntellAgent Processing Server	Active	□	

Amend alerts

1. You can change the status of an alert to: **Active**, **Parked** or **Closed** by changing the value in the *Status* field.
2. To delete one or more selected alerts, click the **Delete** button. Alerts from the **Alert setup** window and all the alert history of their occurrences from the **Workflow alert history** window will be also removed.
3. To change the server queue for a particular alert, select another value (if more than one is available) in the *Server queue* field.
4. You can also change whether or not recipients of a particular alert must be able to switch off all alerts by selecting or clearing the **Allow OFF** check box.

Explanation of fields

Name

The alert (IntellAgent event) name.

Run as

The user who owns the event.

Scheduled

Information only: indicates whether the alert is scheduled (in active use) or not.

Last time

The date and time the event was last run.

Next time

The date and time the event will be run next. This data is available only for scheduled events.

Server queue

The server queue defined for this event.

Status

The current status of the event.

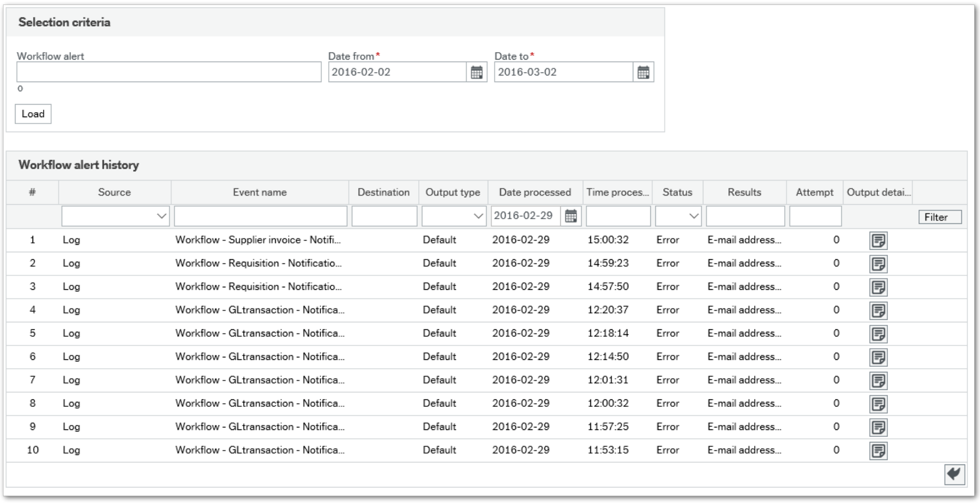
Allow OFF

If selected, the recipient can switch off all alerts generated by the event in the **Personal alert setup** tab available in the **Your profile** window.

Workflow alert history TIN006 - Web client

Window usage

You can use this window to enquire about the history of IntellAgent output generation initiated by workflow processes. This window is in read-only mode. For each output, you can see the output generation time, type, status, destination and other information. You can open additional details for each output.



Selection criteria

Workflow alert:

Date from*: 2016-02-02

Date to*: 2016-03-02

0

Workflow alert history

#	Source	Event name	Destination	Output type	Date processed	Time proces...	Status	Results	Attempt	Output detai...
					2016-02-29					
1	Log	Workflow - Supplier invoice - Notifi...		Default	2016-02-29	15:00:32	Error	E-mail address...	0	
2	Log	Workflow - Requisition - Notificatio...		Default	2016-02-29	14:59:23	Error	E-mail address...	0	
3	Log	Workflow - Requisition - Notificatio...		Default	2016-02-29	14:57:50	Error	E-mail address...	0	
4	Log	Workflow - GLtransaction - Notifica...		Default	2016-02-29	12:20:37	Error	E-mail address...	0	
5	Log	Workflow - GLtransaction - Notifica...		Default	2016-02-29	12:18:14	Error	E-mail address...	0	
6	Log	Workflow - GLtransaction - Notifica...		Default	2016-02-29	12:14:50	Error	E-mail address...	0	
7	Log	Workflow - GLtransaction - Notifica...		Default	2016-02-29	12:01:31	Error	E-mail address...	0	
8	Log	Workflow - GLtransaction - Notifica...		Default	2016-02-29	12:00:32	Error	E-mail address...	0	
9	Log	Workflow - GLtransaction - Notifica...		Default	2016-02-29	11:57:25	Error	E-mail address...	0	
10	Log	Workflow - GLtransaction - Notifica...		Default	2016-02-29	11:53:15	Error	E-mail address...	0	

Enquire about workflow alert history

1. Select an event type from the Workflow alert list in the **Selection criteria** section.
2. Select *Date from* and *Date to*. These dates form a time range for your enquiry.

   Tip: you can press **t** or **d** to enter the current date.
3. Click **Load**. The **Workflow alert history** grid is updated.

Explanation of fields

Source

Indicates whether the search is made against the output queue or the output log.

Event name

The name of the event generating the output.

Destination

The destination of the output.

Output type

The type of the output. The following types are available:

- **Alert list**
- **Business connector**
- **Business World server process**
- **Default**
- **E-mail**
- **External executable/command**
- **File handling**
- **Query definition**
- **SMS**
- **Workflow**
- **Workflow delay point**

Date processed

The date of the event output generation.

Time processed

The time of the event output generation.

Status

The status of the output generation. The following values are available:

- **Active** - the output has not been generated yet.
- **Completed** - the output has been successfully generated.
- **Error** - the output has not been generated.

Results

A short description of the result of the generation of the IntellAgent output.

Attempt

Indicates how many times the system has attempted to generate the output.

Output details

Opens a window displaying more details on the generation of the IntellAgent output.

Mapping of generic tags

Tags in alert messages

When you create an alert template, you will normally convey a message to the target user, explaining the reason for the alert (an empty email wouldn't make much sense).

If you intend to use the template for a certain element type, you may want to include some details from the current work item in the message. You can do this by mapping some generally available tags (available in the template editor) with properties of the element type, and then use the tags directly in the message.

> **Note:** Note that you must explicitly identify the properties that can be mapped. Only properties (i.e. columns) selected in the *E-mail* column in the **Details** tab (in the **Element types** window) will be available for mapping.

Alert example

According to company policy, timesheets will be approved (or rejected) no later than two days after the department manager has been assigned the (approval) task via workflow.

As soon as an employee submits a timesheet, a workflow process (instance) will pass it to the department manager. After two days of passivity, however, a deadline action will be performed, where the manager (as task owner) receives an alert, informing about the un-processed task.

The alert message identifies the timesheet period and the employee (i.e. resource) who submitted the timesheet. This information is taken directly from the current work item (timesheet) and put into the message via *tags*.

Mapping tags and element type properties

When you have selected the element type for mapping, you can map up to 10 properties with the generic tags **Data1** - **Data10**.

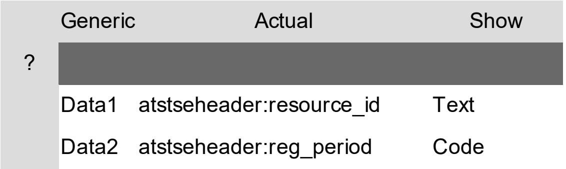
In some cases, you will also have the option to display either the code or the description part of a property - if a description exists.

Default tags

When you create an alert message, you will always have access to some default tags, representing the target user (Alert recipient), current date (Today) etc. See Alerts in workflow.

Mapping example and template

Mapping: Assume the following mapping for the *Timesheet* element type:



	Generic	Actual	Show
?			
	Data1	atstseheader:resource_id	Text
	Data2	atstseheader:reg_period	Code

Alert template message: The template message, for example for Email, could then look like this (the tags are displayed in brackets):

[Today]

Dear [Alert recipient]

Please approve the timesheet for resource [Data1] for the period [Data2] immediately.

Formal actions

System actions and user defined actions

A formal action is a function which can be linked to a user step and displayed as a button in the screen used to handle the current work item.

There are two types of actions:

- System actions - a set of predefined actions.
- User defined (custom) actions - any actions defined by you.

System defined actions

Availability

System defined actions are available for all element types, although some of them will only behave properly for some. We have split the system actions into two groups:

- Actions completing the current task.
- Internal task actions.

An action can be added to a user step, and will then appear as a button in the relevant window.

Actions completing the current task

Two system defined actions will complete the current task, and let the workflow process continue (just like custom - or user defined - actions):

Action	Description
Approve	Completes the task with a positive result. **Note:** The workflow process will always use the positive action(s) to predict the path the transaction will take on workflow.
Reject	Completes the task with a negative result. A negative action often leads to some means of correction. A comment has to be entered when rejecting a task.

In addition, we have one special action, which not only completes the task, but stops the entire process:

Action	Description
Abort	Stops the current process instance. Must be used with care(!). The item is moved toItems to follow up.

Internal task actions

The actions described below will not complete the task - as defined in the process definition, but may transfer responsibility to another workflow user, or just halt the process for later completion.

Action	Description
Escalate	The task is transferred to the supervisor of the current task owner defined in **Workflow user information**. If there is no supervisor defined for the user, the

Action	Description
	item goes to Items to follow up.
Forward	Allows the task owner to manually transfer the task responsibility to another workflow user.
Inform	Allows the task owner to create a sub task - containing the current work item - to inform other users about the task you are handling. The current task owner must select another action to complete the task.
Distribute	Allows users to add new recipients to share the responsibility of the current task. The task is not completed.
Park	The task is set on hold. A comment has to be entered when parking a task.
Save draft	The current work item is saved, but the task is not completed.
Split row	Relevant for element types holding GL transactions. Allows the recipient to correct the current posting setup, by splitting the transaction into several new ones (one invoice line is split into several new lines). There will be one new process instance for each split, starting according to the setup of the current user step. In order to use split row, you need to select the **Users allowed to change data** checkbox on the **Options** tab.

See Choose recipients for action to learn how to select recipients for the following actions: forward, inform and distribute.

User defined (custom) actions

Screen location

The screen where you define user defined actions is found under *System administration>System setup>Workflow*.

Defining custom workflow actions

A user defined or custom action is a formal action defined by yourself, and made available for a user via a button in the relevant window. When the task owner selects a custom action - or any other action available - the task will (normally) be completed, and the workflow process will continue.

You will normally need a custom action when none of the corresponding system actions (Approve / Reject etc.) is sufficient to present the action alternatives in the relevant screen.

Validity: An action can either be defined as a *general* custom action, valid for all element types, or restricted to *one element type only*.

Action button: A custom action will be displayed in the relevant screen as a button - with a symbol (see below) and a descriptive text. When a custom action button is pressed, the task is completed, and further process flow will be according to the process definition.

Setup properties (columns)

You define a custom action by adding a new row to the User defined actions table.

Below, we explain the relevant properties in detail (a few columns are omitted):

Column name	Description
Action code	The unique identifier. Must be 3- 6 characters long.
Description	The text that will be displayed on the action button.
Text	If you enter any Text, this will be displayed - as a receipt - after the action is performed.
Behaviour	The action result. Determines how the button will be displayed in the relevant window: • Positive ✓ + Description • Negative ⛔ + Description **Note:** Note that the workflow process- for example OR-split branches - must take care of further processing. The selected *Behaviour* does not

Column name	Description
	affect the processing per se.
Comment	If selected, the task owner will be prompted to add a comment when the action is selected. If it is not selected, the task owner will not have the option to add a comment.
Automatic action	Relevant for deadline setup only. If selected, the action description will appear as a possible action when the deadline is reached (the *Perform* column in deadline setup).

Action management in workflow

Actions and workflow tasks

In Business World, there is a tight integration between action management (i.e. actions handled in the **Action overview** panel) and tasks.

This integration is expressed via some options in the **Process definition** window, related to the setup of a *user step*:

- You can tell the system that this task must also generate an action, which then will be displayed in the **Action overview** panel connected to the window used to handle the work item.
  As an option, you can also choose to display the action when working with a "parent" to the current work item: if the work item is a customer invoice, you may choose to display the action when working with the customer record also - in **Customer master file**.
- You can tell the system that the user step you are working with also can be used as a *manual user step*. When a process with a manual user step is activated, the step will be available as an action - or an ad hoc task - in windows used to handle objects of the process' element type.

Setup requirements

Element type specific windows with Action overview pane: A manual user step is defined for one or more element types and accompanying windows. The selected window *must have an Action overview pane*.

Link to a defined Action type and action window: The manual user step must be of a valid Action type (defined in **Action type master file**), and thereby linked to a special action window (Task, Phone etc.). Like any other user steps definitions, a manual user step must be set up with at least one workflow action (e.g. Approve, Task completed etc.).

Distribution logic: The manual user step must also have its own distribution logic, and be connected to a distribution rule group.

Creating an ad-hoc process instance

When a user with the correct access rights selects a record in a window targeted for a manual user step, the Action overview pane will display an action button with the name of the user step, allowing the user to create a process instance on the fly.

As soon as a new action is saved, the workflow engine will use the inherent distribution logic to locate the recipient(s), and distribute the new task accordingly. The selected record will be the task's work item.

Completing the task

The recipient may now handle the task in two ways:

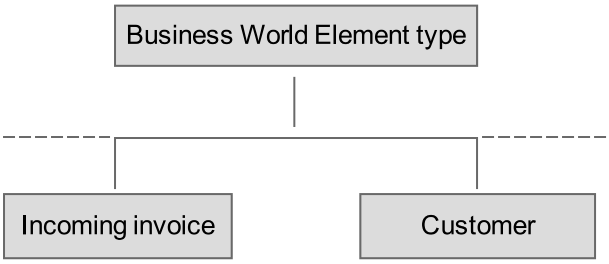
- **Via the task list:** The task list will show the new task - as any other tasks. When the recipient opens it, however, the window selected for the user step will open with the work item details. The new task will be available as a new action in the Action overview pane. To process the task, the user must open the action item, and select one of the defined buttons to complete the task.
- **Via Action overview pane directly:** If the recipient already works with the record selected for the ad-hoc task instance, the Action overview pane will show the new task as an action item - generated by Workflow. The recipient can open the action item directly, and complete the task without accessing the task list.

Element types

Parent object type for Workflow types

An element type is a specialised "meta object" type in Business World, originally created to be processed by the workflow engine. In Business World, all important business objects are "wrapped" in corresponding element types, and thus workflow enabled.

Common set of properties: Formally, the various element types all *inherits from* the general Business World element type, giving them all a common, basic interface (the properties inherited from the parent):



Element types are valid for the installation

All element types are defined for the complete installation, and available for all companies. The **Element types** screen is found under System administration > System setup > Workflow > Element types.

Tables and columns

An element type is constructed on basis of a set of tables, holding the data for the underlying business object type.

When defining or modifying an element type, you have access to all columns from the underlying tables, and you can select the relevant columns to be displayed (exposed) in various contexts (columns for approval, for routing, for distribution etc.).

Table for Work items: In Business WorldWorkflow, it is always the table listed on top that will constitute the basis for awork item. You will select columns from other tables - if present - to supply contextual information about the work item.

Element type variants

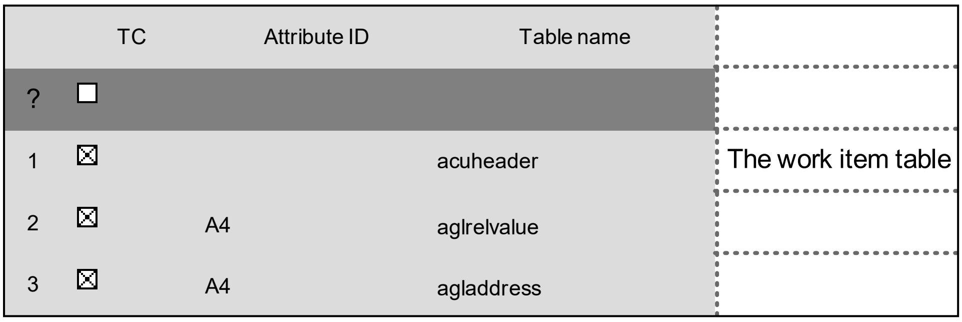
When an element type is constructed on basis of more than one table, it will often be a one-to-many relationship between table rows. An invoice, for example, consists of a header (an invoice as a unit) and many detail rows (a Customer with several addresses).

The predefined element types are constructed in such a way that the *natural work item table* is defined *on top*. For a transactional data type, like an invoice, it is the *single line* which will be the work item, while a typical master file element type (Customer, for example) will use the main record as work item.

We will look at two element type definitions below.

Customer example

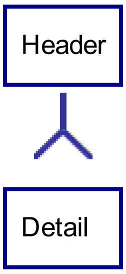
When we take a look at the Customer element type we get the following:



	TC	Attribute ID	Table name
?	☐		
1	☒		acuheader
2	☒	A4	aglrelvalue
3	☒	A4	agladdress

The work item table

Logical structure: Customer is a constructed as a normal header-details (one-to-many) object type:

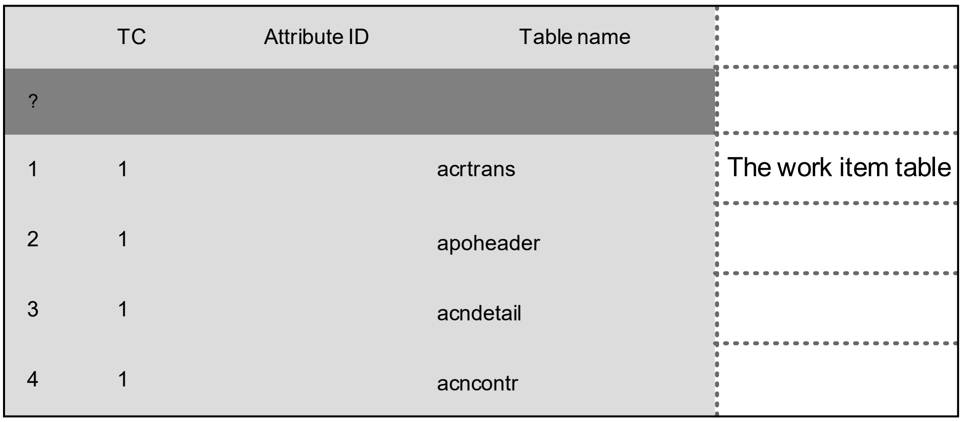


Description: The *acuheader* table, with Id, Name, Bank account etc, is on top. This means, by convention, that it is a row from this table that will be the main work item when sent on workflow.

By selecting various columns from *acuheader*, and any of the other tables, you can configure how a customer record must be presented in a workflow context.

Incoming invoice example - transactional element type

The Incoming invoice element type is defined like this:

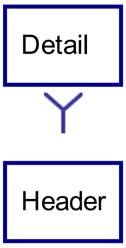


	TC	Attribute ID	Table name
?			
1	1		acrtrans
2	1		apoheader
3	1		acndetail
4	1		acncontr

The work item table

5	1	apoarrord	

Logical structure: As all transactional element types, Incoming invoice is logically constructed as an inverted header-details (one-to-many) object type:



Description: The element type is constructed on basis of five tables, with *acrtrans* on top. It will therefore be a row from *acrtrans* that will be the main work item when an Incoming invoice is sent on workflow (i.e. selected columns from this table, supplied with some columns from the other tables, according to configuration).

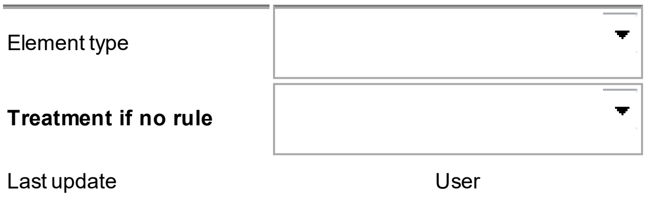
In practice, this means that when an invoice - with several *acrtrans* rows - is saved and sent on workflow, it is actually many *acrtrans* rows that will be sent on workflow; each line will trigger a new process instance.

Still, it is only one invoice, and Business World workflow therefore has a few mechanisms allowing us to treat the invoice as one.

See also Grouping of tasks.

When a work item is not handled

As part of the element type definition, you must specify what to do with un-processed work items, shown as *Treatment if no rule*:



Element type	
Treatment if no rule	
Last update	User

Treatment if no rule: This setting defines what to do with a work item when the distribution failed, i.e. when the distribution logic couldn't find a recipient. You have two alternatives:

- **Move to items to follow up**. An un-handled work item will be taken out of the process routing and only made available for the administrator. See Items to follow up.

- **No workflow**. The work item will taken out of the process and treated as there was no workflow defined for it.

Options

When you have opened a specific element type, you may define detailed features for the element type:

- Element type details(the**Details**tab) allows you to select columns, from all tables, to be used or displayed in various contexts.
- The Aggregate button (in the action toolbar) is intended for transactional element types, where you want the single work item amounts - for example belonging to the same invoice - to be aggregated before evaluation. This is relevant if a recipient must approve "personal amounts": Although the amount on every single work item may pass the approval limit, this may not be the case for the aggregated amount.
- Importance(the**Importance**tab) is a feature that lets you set criteria for*work item importance* (from 5 on top to 1). The importance can then be used to tailor the presentation of work items in the users task list.

Element type details

Properties for various purposes

You use the **Details** tab (in the **Element types** window) to set up detailed rules for how an element type object (a work item) will be displayed and handled in various situations.

Since an element type normally is based on several tables, often with a large set of columns per table, you can select relevant columns (properties) for either

- functional purposes, or for
- display only.

Date columns and relational properties: In some cases (i.e. for some element types) you may need to identify a specific date property to use when retrieving relational values for work items of that type, instead of using the default *today's date*. For this purpose you must use the system setup value WF_REL_VALUE_DATE_COL, described below (see Dates on relations and WF_REL_VALUE_DATE_COL).

Columns used for functional purposes

The following columns in the details table have a bearing on workflow related functionality:

Column	Description
New distribution / Operator	If you select any columns (element type properties) for *New distribution*, any changes to these properties during a process instance, will restart the process for this work. You can use the *Operator* column to specify whether a change to a greater or smaller value must trigger process restart. If *Operator* is left blank, all changes will lead to a restart.
Delay / deadline setup	If selected, you can use the property in a delay condition, or as a condition in deadline setup. In the latter case, the property must be a *Date* property.
Rules / Or-split setup	If selected, the property will be available in the Data specifications section in the **Distribution rules** window, when *Rule type* = **Element type**. It will also be available in the Routing section , when you will find a recipient **From column**. Finally, you may use the column - i.e. the actual property value on the work item - as a condition for selecting an OR-split branch.

Column	Description
E-mail	Indicates whether or not you can use the column as a tag in an E-mail template. E-mail templates are handled by *IntellAgent*, and can also be used in a workflow context.
Document	The columns you select for *Document* (i.e. document type properties) will be available when you define indexes and mapping for a document type - valid for the current element type.

Columns for display

The following columns are used to expose (display) column values in various contexts:

Column	Description
Task list / Order	If *Task list* is selected for a table column, the column will be displayed as a Task property in the user's task list. It will also appear as a property for an *action item*. Note that if you make any changes, the existing task list and action item properties will be updated immediately. All tasks with identical values for the selected properties, will be grouped into one line in the task list. You can select maximum ten columns for this purpose. You must set an *Order* number for each. **Note: Note 1:** You can use the WF_TASKLIST system setup value to override (hide) columns selected in *Task list*. This affects how the tasks are grouped and displayed. **Note: Note 2:** If the recipient is supposed to make changes to the work item as part of a task, you must have selected the correct identifier property. See Task properties for editing below.
Item details / Order	Selected columns will be displayed in Items to follow up and in **WF03** - available from Invoice registration (VP10). You can select as many columns as you like. If you don't use the Order column, the selected properties will be randomly displayed.
Action / Order	You can select up to three columns that will be displayed as task properties when you connect an action to a task via the **Action overview panel** (the columns selected in *Task list* (above) are displayed on the Task's front page).
Master file workflow	Relevant for master files element types only. The selected columns will be included in the work item when appearing in a **Master file approval** process.
Log value	Selected columns will be logged per task and displayed as part of the workflow map, for example when you zoom from an approval window.
Web services	Selected columns will be available for display by the Workflow web ser-

Column	Description
	vice. Application programmers using the Web service must always know the selected columns before trying to display their values.

Task properties for editing

If you create workflow processes where a recipient needs to modify or correct the work item details in a specific screen as part of task completion, the element type must be set up with certain identifier properties in *Task list*, as required by the data entry screen.

The table below lists required properties (columns) for a few, common element types - all transactional. The columns belong to the topmost table (the workflow table) in the element type definition:

Element type	Required properties (columns)
Requisition (REQ)	Order_no
Purchase order (PO)	Order_no
Incoming invoice (IIN)	Voucher_no
Timesheets (TS)	Resource_id, reg_period

Non-transactional element types will normally require that you identify the unique Id column from the topmost table.

Reference: The entries in the *asysvalues* table with the name WF_GROUPING shows the required properties (columns) used for most *transactional* element types. Since entry screens for these element types usually works with the composite "header" objects, you will find that the grouping properties often represents foreign keys into the corresponding header table.

Using aggregated amounts

Relevance

You can use the value of an amount property (of an element type) as a basis for OR-split branching, as part of the process definition diagram, or as a condition for distribution - in a distribution rule.

Transactional element types: All element types defined for financial transactions, are based on two or more tables: one table identifies the composite object - as a header, and one (or more) tables holds the transactions (details).

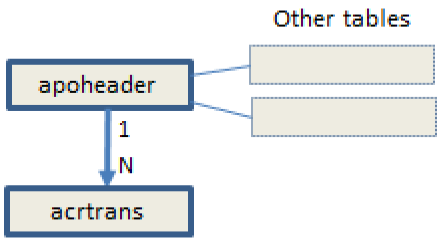
When the workflow engine detects that an element type object with this structure must be sent on workflow, it will create *one process instance per detail row*.

All amount values sent on workflow will, by default, only represent the amount for one detail line in the composite object. If we want the workflow engine to distribute tasks according to the total sum (for all detail rows), we must turn on aggregation.

OR-Split and aggregation: When defining the conditions for an OR-split branch, you have the option to use Aggregate or not (check-box). In order to use it with any effect, you must already have defined aggregation on the element type.

Incoming invoice example

If we take a look at the *Incoming invoice* element type, we find this structure (*acrtrans* is the topmost table in the element type definition):



When a new invoice is saved to the database, and we have an active process definition for the *Incoming invoice* element type, the workflow engine will create one process instance for each new row in *acrtrans*. And, in *acrtrans* we find, among several other properties, *Amount*, *Amount 3*, *Amount 4* etc., i.e. amounts a recipient may be authorised to handle - or not.

The workflow engine knows that the process instances are related, but unless you explicitly say so, it will not use this knowledge during the workflow process.

> **Note:** We repeat: it will always be a row from the first table in the element type definition (table on top) that will be sent on workflow - along with selected property values from other tables.

Aggregation per attribute

Aggregation is always based on an attribute - which must be a property of the element type.

If you for example set up aggregation based on COSTC, all amount values - from different detail rows with the same COSTC value - will be summarised before further processing. In order to do this, however, you must know that COSTC is represented by one of the posting attributes on the detail object.

Aggregation sequence

You can define several aggregation attributes, in which case the workflow engine will compute the aggregated amount for all single items with the same value on the aggregation attribute properties.

	Sequence No	Attribute
?		
1	1	COSTC
2	2	PROJECT

In this example, all amount properties for the single work items belonging to the same header, will be aggregated per COSTC and PROJECT.

Example: Amounts for COSTC 240 and Project 1000 will be aggregated, as will amounts for COSTC 240 and project 2000 - when appearing on different lines within the same header entity.

Reference

Header entities are identified as an entry in the *asysvalues* table with name WF_ GROUPING. See Grouping of tasks.

Master file approval

Functionality

Master file approval is a special feature for important business objects (master file records), allowing you to define workflow processes to validate changes to the objects before they are saved to the database.

It requires, however, that you

- have defined the process as a **Master file approval** process (Checkbox in **Process definition** / **General** tab), and
- selected at least one column for *Master file workflow* (**Details** tab, in the **Element types** window).

> **Note:** Unless both the above conditions are in place, master file records may be passed around in a process, just like any other work items: Unless the process is defined as a master file approval process, selected columns for Master file workflow will be ignored.

Approval screens: Before master file approval can be used, you must also define the various screens (in User step setup) that can be selected for the master file element type.

Approval screens are set up and configured in the Master file approval setup window, available under Common>>Workflow>>Fixed registers.

Columns for validation and editing

You can set up the various approval windows - for different master file types - according to your liking, by selecting contextual properties to display along with selected details from the current record.

The selected details, however, which will be the subject for approval or rejection, will be editable, and *must be selected* from the *Master file workflow* column.

New or changed master file records

A master file approval process will be triggered when a user makes changes to any of the properties (columns) selected for *Master file approval* (a new record will per definition include changes).

When the new or changed record is submitted, changes to the selected properties are not saved to the database, but kept in temporary storage awaiting formal approval. Other users, outside the workflow process, who will access a changed record, will see the old values for these properties.

Master file approval setup

Defining and configuring screens for approval

You use **Master file approval setup** to define the screens to use for master file approval tasks. A master file approval screen is always defined per element type, and identified by a *Function name*.

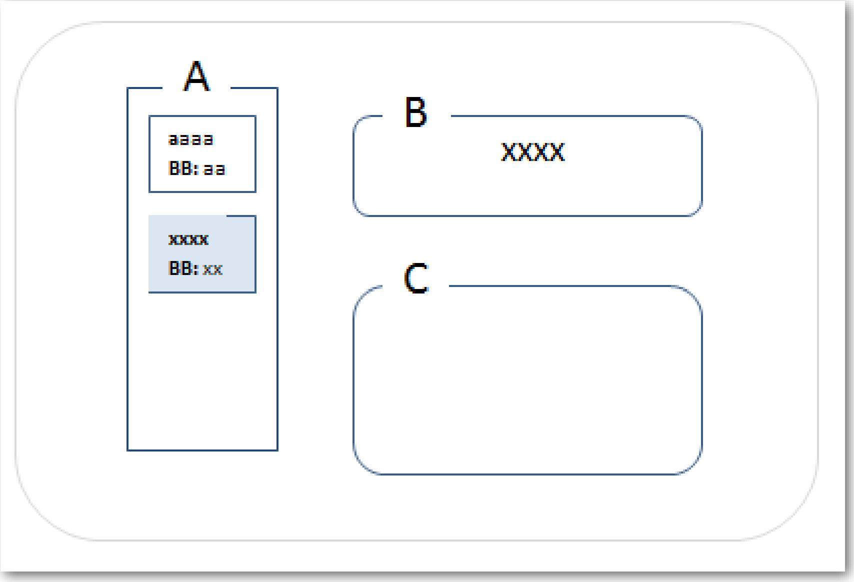
Process definition: When you set up a user step in a master file approval process, the defined screen(s) will be available for selection in the *Screen* field (**Step setup** tab, *Usage* section).

General layout of approval windows for master files

All approval windows for master files use three different elements to display the approval tasks:

- One *pane* (A) for listing the records awaiting approval.
- Two sections for displaying the selected record:
  - The *header* section (B), identifying some unique record properties.
  - The *fields for approval*section (C) - displaying the properties and values selected for approval in the**Details**(in the**Element types**window).

Layout: The diagram below illustrates the general layout:



Comments: The record **xxxx** is selected from the *pane* (A). The *header* section (B) displays some basic properties for the selected record, while the *fields for approval*section (C) displays the fields selected for approval in theMaster file approvalcolumn.

Selecting properties for the records pane and header section

Master file approval setup lets you select the properties to display as *Pane information* and *Header information* respectively.

When you have selected the element type, you can enter a new *Function name* to create a new setup.

Note the following:

- You can select as many properties as you need. There are no limits in the system.
- A selected property will get a default display *order* number which you can overwrite before you save.

Importance

Workflow task importance

Work items are made available for recipients via the *task list*. Business World clients can make the prioritisation of tasks easier for recipients by setting up importance criteria.

An importance criterion is set up with a weight between 5 and 1, where 5 is the highest.

For Task management Experience pack only

Currently, the importance feature is only displayed in the **Task list** (in the Task management Experience pack).

All tasks (i.e. work items) set up with importance criteria, will be displayed with an *importance bar*, stretching from 1 to 5 in length, and with a colour code. A task with work items without importance setup will have no importance bar. If all tasks on the task list have the importance equal to 0, the information about the importance will not be displayed there at all.

Importance criteria

Importance criteria are based on property values on the work item. You can in principle use any column available for the element type as basis for importance settings, and you can use many criteria (properties) if required.

For example: If you always want to give top priority to *Incoming invoices* from a certain supplier (e.g. supplier id = 12345), you will select the *apar_id* column from the *apoheader* table, and assign maximum importance to value **12345**.

You will also differentiate between invoice amounts, and add a criteria for the *amount* column as well (*acrtrans* table), and assign weights to different amount values.

When a work item appears in the recipients task list, the importance will be calculated on bases of the total weight of all the importance criteria, divided by the number of criteria used. See Example below for more details.

The importance setting section

When you open Importance for a certain element type, you define the importance criteria in the *Importance setting* section. First you select the relevant table and column:

-------- Importance setting ---------------

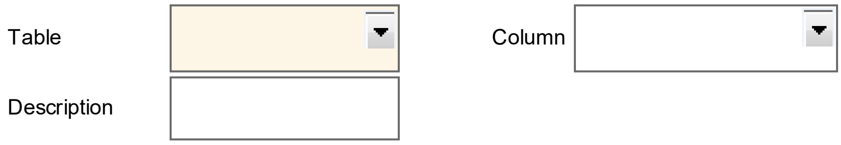


Table		Column	
Description			

Next, depending on the data type of the selected column, you will get a table where you set the actual weighting for relevant property values.

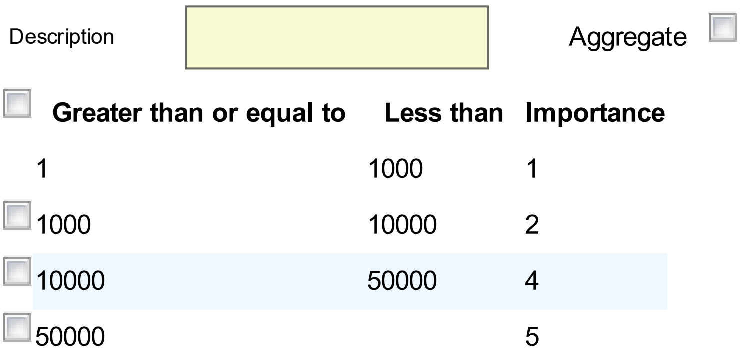
Example

The following example gives an illustration of two possible settings for *Incoming invoice*

- **Criterion 1:** Supplier importance is set to 5 for supplier with Id = 12345. No other suppliers are weighted (Table: *apoheader*, Column: *apar_id*):

□	Value	Importance
□	12345	5

- **Criterion 2:** Invoice amounts can be weighted as follows (Table: *acrtrans*, Column: *amount*):



Description □ Aggregate □

□	Greater than or equal to	Less than	Importance
	1	1000	1
□	1000	10000	2
□	10000	50000	4
□	50000		5

So, when a task recipient receives a work item from supplier 12345, with the amount 15000, the importance is calculated to 4,5 = (5+4)/2.

When the same task recipient receives a work item from another supplier, with the same amount (15000), the task importance is calculated to 2 = (0 + 4)/2.

Amounts and aggregation: When setting up an importance criterion based on amount (i.e. any numeric value) for a transactional element type - typically an invoice - you can choose whether to use the current work item amount, or the total, aggregated, amount for the invoice.

> **Note:** Note that *Aggregate* in this context has no influence on the general Aggregate settings for the element type. See Using aggregated amounts.

Importance criteria for date properties

You can define importance criteria on basis of date properties on the task's work item, where the value of the date property is evaluated in relation to the current date (the date when the user opens the task list).

The criterion must be entered as a number (decimals allowed), where **0** means "when the property value is the same as today", **1** is the day after, etc.

To refer back in time, you use negative numbers (**-1** will be **true** the day before the property value day).

Example: Assume you want to use the following setup for an invoice, based on *due_date*:

- When it is three days or less until due date, show a reminder (weight 3).
- If the task is un-processed at due date - or later - show an alert (weight 5).

This will lead to the following setup (Table: *acrtrans*, Column: *due_date*):

□	Greater than or equal to	Less than	Importance
□	-3	-1	3
□	0		5

> **Note:** Note that you cannot use date constants (e.g. 12.6.2013) when setting up date based criteria. You can only use relative, numeric, values. If desired, decimals can be used.

Dates on relations

Dates on relations

All element types with relational properties may in principle be set up with *Dates on relations*, meaning that a specific relational value for an element type object always will have a *validity period* (start date and end date). A given object may therefore have several values for the same relational property, with different validity periods (only one is valid at a given date).

Relational value for a work item

By default, Business WorldWorkflow will use the current date when it retrieves the correct relational value for a work item, i.e. the value as it is*today*.

In some cases, however, the relevant date may be represented by the value of a *date property on the work item*, and you are interested in the relational value for that date.

For an employee (Personnel - RES - element type), this can for example be the value when the employment started (or *will start* if this is a future date), represented by the *date_started* column in the *ahsresources* table.

Exceptions

The two element types, Timesheets (TS) and Project expense approval (PX) do not use today's date as default relation date. Instead, they use the following property values:

Element type	Date column	Table
Timesheets (TS)	trans_date	atstsehourdet
Project expense approval PX)	voucher_date	atsexpenses

asysvalues reference: These properties are defined by Business World and stored in the *asysvalues* table, under the name WF_REL_VALUE_DATE_COL(!).

These values can be overriden for defined contexts by entering new WF_REL_VALUE_DATE_COL system setup values for TS and PX (see below).

WF_REL_VALUE_DATE_COL

You can use the system setup value WF_REL_VALUE_DATE_COL to identify the date property to use when retrieving relational values for a given element type, in a given context.

WF_REL_VALUE_DATE_COL

Identifying date column for relational values

You can use the system setup value WF_REL_VALUE_DATE_COL to identify date properties to use when the workflow engine must retrieve relational values for a work item (belonging to a certain element type).

Every entered value must be explicitly linked to an existing *distribution rule* (defined outside a specific workflow process) or an existing *delay node* (part of one specific process definition).

Default: By default - if no values are defined for an element type - the system will use the relational value for the current date, i.e. the date when the workflow step is executed.

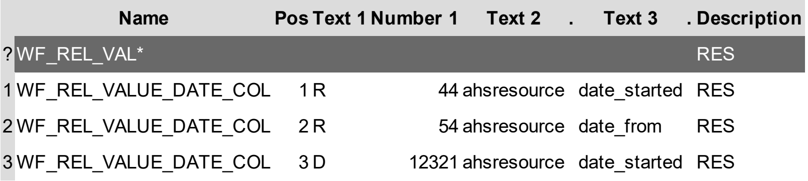
Value conventions

The columns in the **System setup values** screen are used as follows:

Column	Description
Name	Identifies the value (set). Must always be **WF_REL_VALUE_DATE_COL**.
Pos	Sequence number. Identifies a unique row along with *Name*.
Text 1	Either **R** (to be used in a distribution rule) or **D** (delay node).
Number 1	Identifies either the correct rue or delay node. Dependent on the value in Text 1: • If **R**: The rule identifier (the number you find outside a selected rule in the Distribution rules window). • If **D**: The delay node Id. To find the node identifier, you must query the database. See Finding the delay node id below.
Text 2	Name of element type *table* holding the date column.
Text 3	Name of date *column* to use.
Description	Name (code) of element type.

Example

Below, we have retrieved the three rows valid for the Personnel (RES) element type. Two are used in distribution rules, one in a delay node:



	Name	Pos	Text 1	Number 1	Text 2	.	Text 3	.	Description
?	WF_REL_VAL*								RES
1	WF_REL_VALUE_DATE_COL	1	R	44	ahsresource		date_started		RES
2	WF_REL_VALUE_DATE_COL	2	R	54	ahsresource		date_from		RES
3	WF_REL_VALUE_DATE_COL	3	D	12321	ahsresource		date_started		RES

Finding the delay node id

There is no simple way to retrieve a delay node id for a certain node in a certain process. Below, we list a query that will show all delay nodes for all active process definitions for the RES element type in the EN client.

You can copy the query text and make your modifications.

```
SELECT n.node_id as "ID", n.description AS "Node name", p.description as "Process name"

FROM awfprocess n, awfprocess p, awfprocelemtype e, awfversion v

WHERE e.client = 'EN'

AND e.element_type = 'RES'

AND e.client = p.client

AND e.node_id = p.node_id

AND e.version_no = p.version_no

AND p.status = 'N'

AND e.client = v.client

AND e.node_id = v.node_id

AND e.version_no = v.version_no

AND v.active_flag = 1

AND n.client = p.client

AND n.parent_id = p.node_id

AND n.version_no = p.version_no

AND n.node_type = 5
```

Comments: The query result is three columns identifying the node id, the name of the node, and the name of the process where the node appears. As you see, we will only list nodes for active processes.

Distribution of tasks

User steps in the process

The workflow process definition holds all details about process tasks and how the work item is routed between tasks.

In this chapter, we will explore the various options for task distribution. This is only relevant for user steps, and referred to as the distribution logic defined for the step.

The distribution logic for a task is not directly related to the workflow process, as expressed in a process definition diagram, but to how recipients of user steps must be identified (located), and, optionally, related to the conditions that must be met before a task is considered completed. These conditions may be very complex.

Reminder: Element types with a header and details

An important feature with the Business World Workflow, is that all processes designed for element types constructed with a header object and detail objects, will be always processed *per detail.*

When - for example - a new requisition is saved to the database, and we have an active process definition for the *Requisition* element type, the workflow engine will create *one process instance per requisition line* . We can use various grouping mechanisms for related tasks (originating from the same requisition) before they are presented to the recipient.

> **Note:** For a composite element type, it is always data from the *first (topmost) table* in the element type definition that will trigger new process instances. For the pre-defined element types, we have defined the table with the most detailed properties on top.
> **Example:** For new requisitions, the workflow engine will create a new process instance for each new row in the *acrtransdetail* table.

Distribution logic

When you define a process, in the **Process definition** window, you have basically two options when determining the recipients of a user step instance:

- **By rule group:** Before the user step is instantiated, the workflow engine will process the rules found in the connected distribution rule group, and then identify the correct task recipients.
  Rules and groups are defined in the Distribution rules window in the Desktop or Web

client.

- **Manually:** If the rule group is empty (contains no rules), the responsible for the previous task instance must manually select the recipients. Per default, all defined workflow users and roles are selectable, but you can limit the possible task recipients as part of the user step definition.

Special circumstances

In addition to the two basic alternatives, we have two special circumstances which may affect task distribution:

- workflow iteration via an OR-split.
- data entry in tailored windows.

Workflow iteration via an OR-split branch: If the user step comes at the end of an OR-split branch, and the OR-split is set up to rollback to the workflow initiator or to the last owner, any of the *previous recipients* in the process can also be selected as recipient for the user step. In this case, the distribution is computed on basis of the OR-split branch definition and previous user step instances, and not on basis of any rules defined for the user step.

Tailored Business World windows: **VP10 – Registration of invoices** has tailored functionality for distribution of an invoice to the next – normally the first – task instance in a running workflow process: If the system parameter WF_MANUAL_DISTRUBUTION is activated, and a new invoice is saved, the user can select task recipients directly.

The selectable recipients can either be found via a rule group or - if the group is empty - be all or a limited set of workflow users.

Managing distribution to a user step

A distribution rule group is used to identify the recipients of a user step instance, and may contain zero or more rules. If a user step is linked to an *empty rule group*, the actual recipients must be identified by other means.

Several rules in a group

The main purpose of a distribution rule is to identify the recipients of the task. If a group holds several rules, the workflow system will process the rules according to priority (lowest number first). If a rule is considered valid - according to the Data specifications - rules with a higher priority number will be ignored.

You can also define rules with the same priority, meaning that all will be processed in parallel. This may lead to a very sophisticated processing inside the step, and not visible in the general process definition diagram.

Components of a distribution rule

A distribution rule consists of two component types:

- *Data specifications*, which is used to limit the rule's validity to one or more element types, with certain values on important properties. You can either identify the element types directly (by name), or indirectly, by identifying (element type) attributes - which must be available properties for relevant element types.
  In addition, you can specify that a rule must be based on a specific (numeric or date time) property on the element type - and further specified by the *Routing* rules.
- *Routing* specifications, used to explicitly set up the logic for how the recipients will be found.

> **Note:** You will always need to set up routing specifications, while the data specifications are optional (if empty, the rule can in principle be used for all element types).

General and specific rules

Some user steps may be of a general nature and used in many processes. An example is that a work item must be passed on to the department head before further processing. The head of department is typically identified via a relation on the COSTC attribute, which is a property of many element types. We can thus create a simple distribution rule to be used by several processes and user steps.

Very often, however, we find that processes require tailored distribution logic, and that each user step in the process needs specific distribution rules. In such cases, we will create a rule group, probably with only one rule, to be used by one user step in one process.

> **Note:** **Keep it simple:** We recommend that you keep the distribution logic simple, whether it is for a general purpose, or for specific use. It is much easier to create a new rule (and rule group) which you understand, than to find and correct unwanted distribution based on a complex rule or rule group.

The Distribution rules window

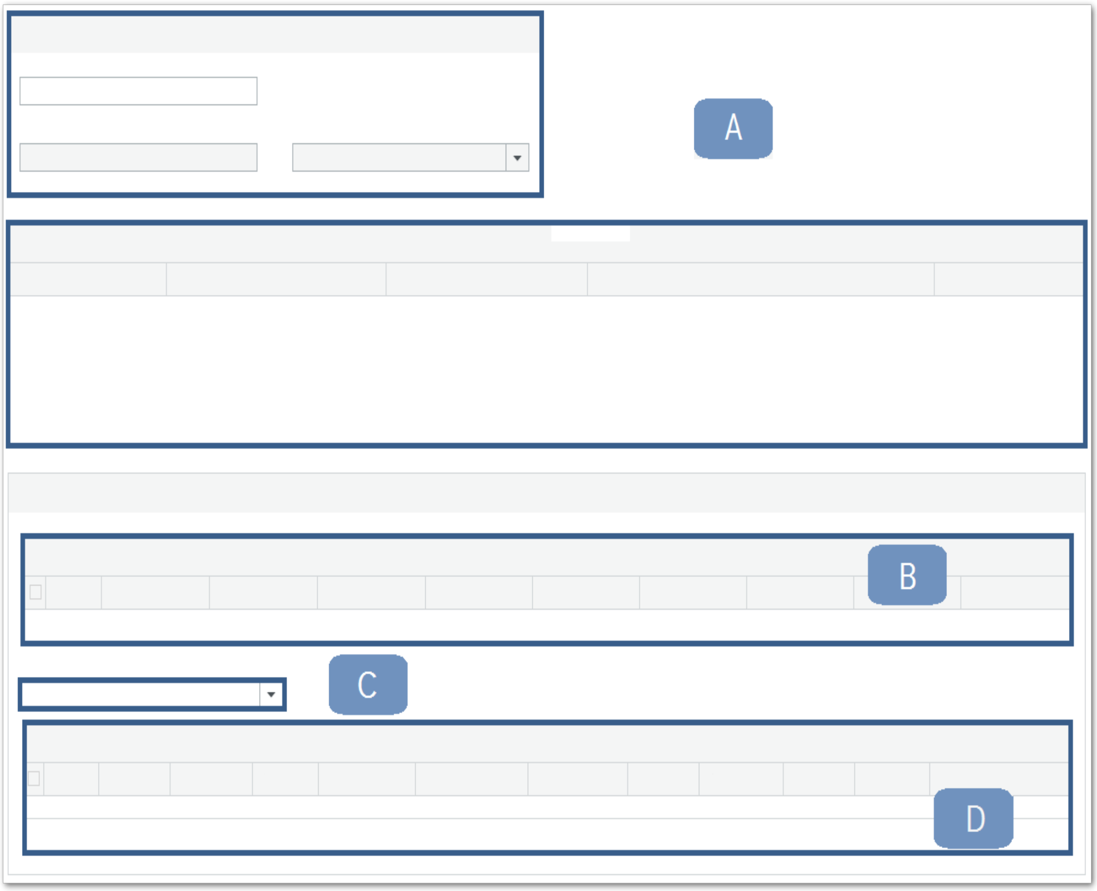
Window structure

The Distribution rules window consists of four main elements,

- General information about rule groups and rule - identified with an **A** below.
- The Data specifications section - **B**.
- A drop-down allowing you to select a specific property that the rule will be based on - **C**.
- The Routing section - **D**.

Overview diagram

The diagram below illustrates the window structure:



The main elements

A: You use the upper two sections in the screen to enter general properties for rule groups and rules. Other sections of the screen are used to handle one rule. See Routing: Rules and rule groups.

B: The Data specifications for a rule are used to limit the rule's validity to one or more element types, based on the values of one or more attribute properties (on the current work item). You can enter several Data specification lines, meaning that you can make the rule valid for a wide range of element types and property values (not recommended, though). See Distribution rules: Data specifications for details.

C and D: The Routing section (D) may be linked to a selected property (C), or it may not. In the routing table (D), you will select a method for finding recipients, and set up parameters according to selected method. See Distribution rules: Routing.

Rules and rule groups

General rule properties

The upper two sections of the **Distribution rules** window holds general properties for distribution rule groups and distribution rules respectively.

Distribution rule group

You always start in the Rules groups section, where you can select an existing, or create a new rule group. The rule group name is unique.

Distribution rule

You can use the window to define multiple rules. Selecting a rule will cause the display of rule definition in the **Rule details** section. A rule name must be unique within a rule group.

A rule has a validity period (Date from - Date to) and a priority. The rule with the lowest priority number will be validated first.

When a rule is considered valid (the work item properties matches the Data specifications), it will be used, and other rules in the group will be ignored.

Rule priority and data specifications

When you have more than one rule in a group, the workflow engine starts with the rule(s) with lowest priority number. Rules with the same priority and valid for the current work item, are executed in parallel.

When a workflow item is to be distributed, the workflow engine will check the rule(s) according to Data specifications. If the workflow engine confirms that the Data specifications are valid for the current work item, rules with a higher priority number are skipped (even if the actual routing will not find any recipients).

Example: Assume that we have three rules in a group, all of the Attribute/Relation rule type and based on values of the COSTC attribute:

Rule	Priority	Data specifications	Comment
Rule 1	1	COSTC between **100;499**	Top priority. Will always be *evaluated*, but used only if current work item is related to a COSTC value between 100 and 499.
Rule 2	1	COSTC like **5***	Top priority. Will always be evaluated, but only used when the COSTC value starts with 5.

Rule	Priority	Data specifications	Comment
Rule 3	20	COSTC like * (i.e. all values)	Will be evaluated (and used) if none of the above specifications are valid.

Distribution rules: Data specifications

Restricting rule validity

You use the Data specifications section to limit the validity of a rule. If you want no restrictions at all, you can keep the section empty.

When the rule is connected to a user step, the workflow engine will first check whether the rule is relevant for the current work item or not. This is done by evaluating the work item against the various parameters (column values) in the *Data specifications* table (can be more than one row). If the work item values correspond to the acceptable values, the rule will be used to locate recipients (using the logic in the Routing section).

Normally, you create a rule group for a specific user step in a specific process definition. It is not recommended to create general rule groups, intended for several processes and element types (unless it is very simple).

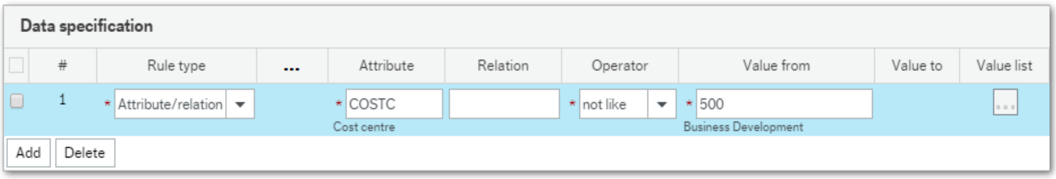
More than one rule

The typical use of Data specifications is when the actual routing of work items must be very different, depending on specific property values on the work item. In such cases, you need more than one rule in the rule group.

Example: Assume that all invoices must be routed to the COSTC manager, with the exception of invoices related to COSTC = 500 which must go to the Division manager.

In this case we need two rules, for example with the following Data specifications:

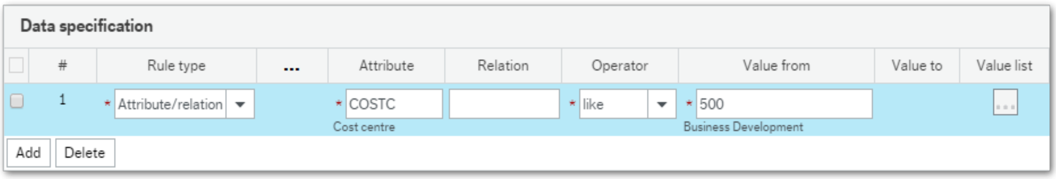
Rule 1: The most used rule is given priority 0:



Data specification

□	#	Rule type	...	Attribute	Relation	Operator	Value from	Value to	Value list
□	1	* Attribute/relation ▾		* COSTC Cost centre		* not like ▾	* 500 Business Development		...

Rule 2: The special rule will have priority 1:



Data specification

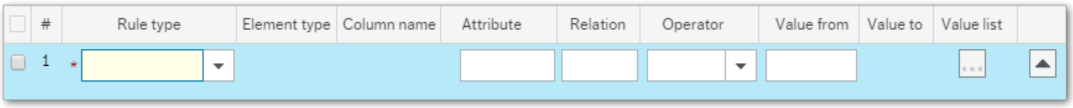
□	#	Rule type	...	Attribute	Relation	Operator	Value from	Value to	Value list
□	1	* Attribute/relation ▾		* COSTC Cost centre		* like ▾	* 500 Business Development		...

When a work item is about to be distributed according to this rule group, Rule 1 will be evaluated first (top priority). If the work item has a COSTC property with a value not like 500, this rule will be used. If the COSTC value is 500, however, the rule is skipped, and Rule 2 will be evaluated - and it will match the COSTC value on the work item.

Data specifications and Routing: In many cases (and our example is no exception), you can instead use advanced setup in the Routing section to ensure alternative distribution. It is all a matter of preferences.

Table description

The Data specifications table are shown below. You may enter zero to many rows in order to add restrictions on the current rule's validity. If you enter more than one row, all rows must be evaluated to TRUE in order to make the rule valid.



□	#	Rule type	Element type	Column name	Attribute	Relation	Operator	Value from	Value to	Value list	
□	1	*								...	▲

The columns are explained below:

Column	Description
Rule type	Can either be **Element type** or **Attribute/Relation**. • Element type: Restricts the rule to a specific element type. • Attribute/relation: Restricts the rule to element types where a selected attribute (in the *Attribute* column) must be present as a property.
Element type	Relevant if *Rule type* is **Element type**.
Column name	Relevant if *Rule type* is **Element type**. You need to select a specific column from the available element type columns, and then specify valid values for this column (*Operator* and *Value* columns).
Attribute	Relevant if *Rule type* is **Attribute**. The selected attribute must be a property on the work item (i.e. on the element type).
Relation	Relevant if *Rule type* is **Attribute**. Can be used to indirectly limit validity. If used, *Operator* and *Value* refers to the *Relation* value, not the *Attribute* value.
Operator	Required. Defines the comparison operator for values.
Value from	Required. Wildcards (e.g.'*') are allowed.
Value to	Required. Wildcards (e.g.'*') are allowed. Can be edited when two value operators are used. The value operators are: between, not between.
Value list	Opens a pop-up window where you can create a list of values. Available for multi-value operators, such as: in list, not in list.

Distribution rules: Routing

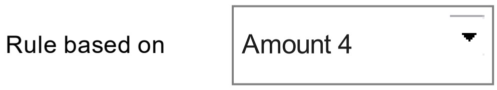
Identify recipients

In the Routing section of Distribution rules, you will mainly set up rules for recipient identification. In addition, you may also set the routing conditions, by using different routing for different property values.

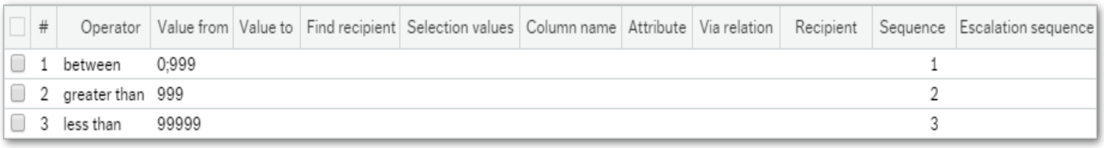
Rule based on

If you want to base the rule on a specific property of the element type, you will define the actual routing on basis of the *value* (on the current work item) of this property. Note that some of the property options (hours, due date) are limited to very few element types.

The two columns, *Operator* and *Value* in the Routing table, are linked to this property, as illustrated below (Amount 4 = Company currency):



Rule based on: Amount 4



□	#	Operator	Value from	Value to	Find recipient	Selection values	Column name	Attribute	Via relation	Recipient	Sequence	Escalation sequence
□	1	between	0;999								1	
□	2	greater than	999								2	
□	3	less than	99999								3	

Comment: In this example, we have set up routing on basis of an amount value. When the current work item holds this property, the property *value* will be used to identify the relevant line(s) in the routing table.

Conditional recipients: When you base a rule on a property, you will typically have several lines in the Routing table, where each line identifies the correct recipients on basis of the value of the selected property. If the values overlap, some tasks will be distributed to several recipients, according to the value in the *Sequence* column.

Amounts and self-approval

When you base a rule on one of the available amount types, and locates a recipient based on the location method in the Routing table, the workflow engine will always check a special condition:

Does the current work item hold a RESNO value as part of the posting attributes and which is the same as the recipient's RESNO? (Or: Is the current work item saved by the recipient?)

- If no, just continue.
- If yes: Is the RESNO - as workflow user - authorised to handle an amount of current size?
  - If no, skip this line.
  - If yes, use this line to located recipient(s).

By default, the amount will refer to the amount on one detail line (on original work item), but - if you have defined aggregation on the element type, the amount will refer to the aggregated amount.
Amount limits are described as part of Limits and supervisor setup.

Finding recipients

When a rule is evaluated at run-time, it is only the data specifications that decide whether a rule is valid or not.

If the current work item does not match the data specifications, the rule will be invalid - and ignored.

Valid lines - i.e. lines that match any *Based on* conditions - in the Routing table will be executed according to the numbering in the *Sequence* column (lowest number first):

- If two or more *valid lines* use the same *Sequence* number, the task will be distributed to several recipients in parallel.
- When two lines have different *Sequence* number, the line with the highest number will not be executed until the recipients identified by the first line have completed the task.

> **Note:** A line can be *not valid* if the routing is based on a property - like Amount - and the actual value on the current work item does not match the condition expressed by Operator and Value.

Routing: Find recipients

The Routing table

For each valid row in the Routing table, the workflow engine will use the criteria set up in the row to find recipients.

The main columns are shown below:

Find recipient	Selection values	Column name	Attribute	Via relation	Recipient	Seq	Escalation sequence

Seq and Escalation sequence: The two columns *Seq* and *Escalation sequence* are described at the end of this chapter.

Methods for finding recipients

The *Find recipient* column gives you the available methods for locating recipients:

- Direct methods - where you select a specific user or role from available workflow users or workflow roles.
- Property based methods, i.e. methods using property values on the current work item to identify recipients. The property must always be an attribute, which either identifies the recipient directly (RESNO for example) or via a relation.
- Organisation based methods, where you use the hierarchy in an organisational chart to identify the recipient.
- Special methods, based on pre-defined functions.

Direct methods

There are two direct methods:

- **From list** - which allows you to select one workflow user, which then will be the sole recipient (according to the current line).
- **Roles from list** - which allows you to select a workflow role as recipient. All members of this role will then receive the task, either simultaneously (in parallel) or one after another. The distribution details, when a *role* is the task recipient, is set up in the role definition.

Property based methods

There are five localization methods based on a property (value) on the current work item:

Method name (Find)	Description
From attribute	Will find the recipient based on the value of an attribute. You select the attribute from the *Recipient* column, which will list attributes which can be used to identify a user.
Relation on attribute	Will find the recipient on basis of an attribute which is related to an attribute on the work item. You will identify the work item attribute in the *Attribute* column and the related attribute in the *Recipient* column.
Via another relation	Will find the recipient on basis of a relation to a related attribute(!). You identify the work item attribute in the *Attribute* column, the related attribute in the *Via relation* column, and the indirectly related attribute in the *Recipient* column.
From Flexi-field group	You must select the Flexi-field group from the *Selection values* column, and the attribute to which the group is attached, in *Attribute*. In the *Recipient* column you select a Flexi-field property which represents a user or resource.
From column	(Column in an element type - exposed for Rule / OR-split). Only relevant if you have a general ("free") element type property which holds an attribute value, i.e. Business World does not know that the property (column) represents an attribute value. 1. You select the element type in *Selection values* and the relevant attribute column in *Column name*. 2. You identify the attribute corresponding to the column in *Attribute*, and - if relevant, any relation to be used to identify the recipient in *Via relation*. The *Recipient* column is used to identify the recipient, either as a property of *Attribute* - or of the related attribute.

Organisation based methods

There is one organisation based method, **From Structure setup**, where distribution is carried out on basis of an attribute hierarchy (also called Structure setup) - normally an organisation chart.

> **Note:** You need to be familiar with the chart definition, the levels and responsible (Business World user) per level, in order to set up correct distribution. For example: a typical chart is based on 1. RESNO (resource/employee) - 2. COSTC (department) - 3. SECTION - 4. DIVISION, where the three highest levels (2-4) all have an identified *level responsible*.

Selecting level attribute: You select the chart in the *Selection values* column (can be more than one available), and then the attribute representing a specific level in the hierarchy - in the *Attribute* column (note that the levels are not necessarily listed in the correct order). The selected level attribute *must be available on the element type*.

Finding recipient: You set up the distribution rule in the *Recipient* column, which always will be the responsible for a specific level in the hierarchy:

- **0 levels above** - means that distribution goes to the responsible for the level identified by *Attribute*. For example: If the level attribute = **COSTC** (department), we assume that the work item will hold a COSTC property with a specific value, identifying the department. Distribution goes to the responsible for this department.
- **1 Levels above** - **N Levels above** - where N represents the number of levels minus one - means that distribution goes to the responsible for a higher level. For example: If COSTC is the level attribute, and you select 2 levels above, distribution will go to the DIVISION manager.
  Note: *Levels above* is only relevant when the chart allows *Alternative level connections*, for example that a COSTC can be directly related to a DIVISION, but without a SECTION relation. Unless you have alternative level connections, you can use *Fixed level X* (see below).
- **Fixed level X** - where X represents the level number (first level is always 1). A fixed level = **3** will always distribute the task the SECTION manager.

Special method

There is one special method, called **From function**, which utilises special functionality related to a few roles mainly used in the Procurement module.

In order to set up correct distribution using **From function**, you must be familiar with the procurement roles and relevant element types.

Available functions: You will select the function in the *Selection values* column:

The function...	is valid for...	and supports distribution to...
General	the following element types: • Contract invoice • Contract master file • GL transaction • Incoming invoice • Posted contact invoice • Posted GL transaction • Posted purchase invoice • Posted supplier invoice • Posting incoming invoice • Purchase invoice • Supplier invoice	the *Responsible* role only.
Procurement	the following element types: • Contract invoice • Contract master file • Goods receipt • Incoming invoice • Internal order • Missing goods receipt • Order confirmation • Posted contact invoice • Posted purchase invoice • Posting incoming invoice • Purchase invoice • Purchase order • Requisition	one of the following roles: • Accountable. • Requested by. • Responsible.
Sales	the *Sales order* element type only.	one of the following roles: • Responsible. • Salesman.

Routing: Seq and Escalation sequence (tasks within a user step)

Relevance and use

Seq and *Escalation sequence* are only relevant if you set up more than one row in the Routing table, where each row (probably) will identify different recipients. When you set up more than one row, all recipients (for each row) must process the task, before the current user step is considered completed.

Only valid rows (i.e. if you base the rule on a property) will normally be processed, but you can, via the escalation mechanism, transfer the routing logic to any row - even if it is not valid.

Escalation and deadlines: Escalation sequence is restricted to deadlines, and the column is only relevant when

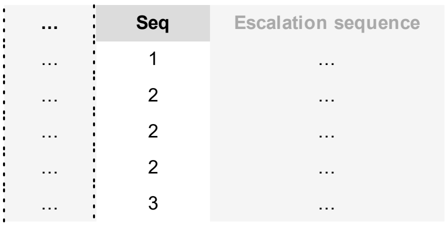
- the current user step is set up with a deadline where
  a. the action is **Reminder** or **Escalate**, and
  b. the target user will be located by a **Rule**,
  and
- the time limit defined for the deadline is reached.

Splitting a user step into several tasks

By using advanced routing options, you can actually create a new workflow process - within a single user step.

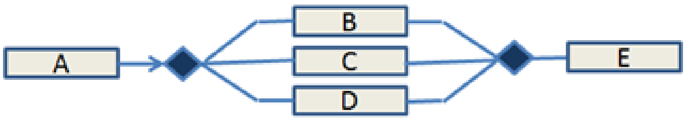
- When you assign the same *Seq* number to several rows, the workflow engine will create (at least) one task per row - with the same work item - and try to process them all in parallel. In practice, you set up an AND-split inside the step.
- If you have rows with a higher sequence number, these will not be executed until all the previous tasks have been completed.

Example: Assume five rows, with the following *Seq* numbers:



...	Seq	Escalation sequence
...	1	...
...	2	...
...	2	...
...	2	...
...	3	...

If all rows are valid, this setup will create the following workflow task within the current user step:



If one of the tasks (A -E) fails, the failed task will be marked as un-handled (and only available for processing in the Items to follow up window.) This may cause delays.

Tasks and user step: In general workflow terminology, a user step is normally one type of workflow task. When we discuss workflow within a user step, the original task (user step) is split into several identical (sub-) tasks, where each recipient will handle the same work item.

The Seq column

Seq indicates the execution or distribution order, and rows with the lowest number are executed first. If more than one row shares the same *Seq* number, these are executed in parallel. The row with the next *Seq* number will not be executed before the previous task is complete.

If the task is not processed within a defined time limit (deadline), one of the following will happen (in this sequence):

a) If no recipient is found - the work item is marked as not handled and only available via Items to follow up.

b) If the user step supports escalation based on distribution rule, *and* there is an alternative row (escalation sequence), use this row.

c) If b) applies, but does not produce a recipient, check if there is a supervisor. If no supervisor is found, the work item is marked as not handled and made available via Items to follow up.

The Escalation sequence column

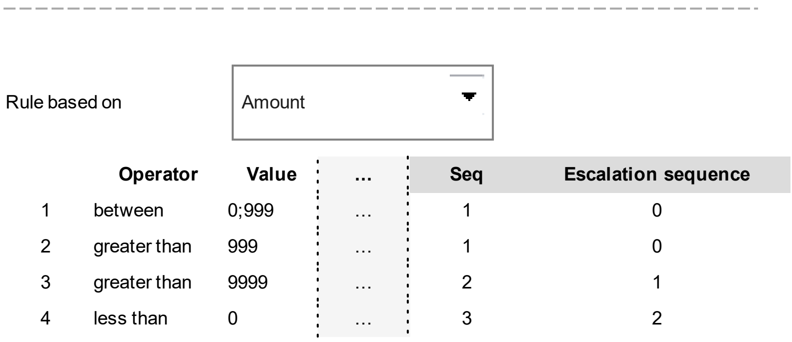
Escalation sequence indicates an *alternative row* to be executed - if the original task fails. This can happen if current row is not processed by a recipient within a defined time limit (deadline) - or if the recipient chooses to manually Escalate the task (requires that an *Escalate* action is set up for the user step). The purpose is mainly to avoid that a process instance is aborted due to unhandled tasks.

The alternative escalation row does not have to be a valid row (i.e. if routing is based on a property value on the work item - see Example below).

> **Note:** Note that an escalation will take place before the workflow engine will look for a possible supervisor.

Example

Assume the following setup:



Rule based on: Amount

	Operator	Value	...	Seq	Escalation sequence
1	between	0;999	...	1	0
2	greater than	999	...	1	0
3	greater than	9999	...	2	1
4	less than	0	...	3	2

Generally, the distribution will be as follows:

- An amount under 1000 will be handled by the first row, while higher amounts are handled by row 2.
- An amount higher than 10000 will first be handled by row 2, then by row 3.

When a deadline is reached, the following may happen:

- If the recipient(s) from rows 1 or 2 does not process the task, it will be escalated to row 3.
- If the recipient(s) from row 3 does not process the task, it will be escalated to row 4.

Distribution of grouped tasks

If required, the WF_CONTINUE_DISTR system setup value can be created for a specific element type to group all the transactions in a workflow task and distribute them at once, instead of one by one.

In case you need more than one element type, create one system setup value for each element type.

> **Note:** It is not recommended to use this system setup values with element types used in complex workflow processes containing sub-processes.

Process definition

Where everything is assembled

A process definition is where all the rules, actions and functions are assembled and made into an executable workflow process.

In practice, you will often start with a draft definition, and make frequent visits to the other windows (Element types, Distribution rules, etc.), to make sure that the process will behave as expected.

When the definition is complete, validate, and activated, it will become an executable process, ready for testing.

Process instantiation

Typically, you will reserve a process definition for one element type only, but you may need to define more than one process for the same element type. If so, you will also need to separate the processes by defining different *conditions for instantiation*.

How a process is instantiated: When you attempt to save a record of an element type with a defined workflow process to the database, the workflow engine will proceed as follows:

It will check for any start up conditions for all the active process definitions for the current element type. For every process definition with a valid startup condition, the workflow engine will create a new workflow process instance for the work item, and will transfer it to the first defined task.

Transactional element types: Note that if the element type is a transactional type (header and many details), and the current work item consists of many detail lines, the workflow engine will create a new process instance for each detail line.

Overview

We will go through most of the options you may need in a process definition:

- How you work with the Process definition window.
- How you define a user step.
- How you define a system step.
- How you set up an OR-split.
- How you configure a Delay node.

There are a few setup elements for which we only give a short description below:

- AND-splits and AND-joins
- Synchronisation steps
- Sub-process

AND-splits and AND-joins

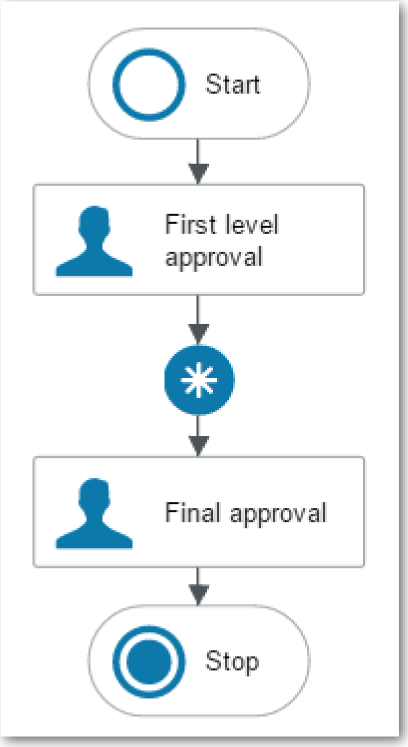
AND-split is used to divide the general workflow into several parallel branches and tasks.

AND-join is used to converge different branches and to avoid the workflow moving to the next task until all the parallel tasks are completed.
An AND-join is not required after an AND-split. Each parallel branch can have its own workflow and can end at any time.

Synchronisation steps

You use a Synchronisation step to ensure that all the detail tasks belonging to the same header object are completed before the process continues.

Example: A common scenario would be when a transaction with multiple lines has to be authorised, but before the task owner completes the final authorisation, each line of the transaction has to be approved by different recipients (determined by the distribution rule). In this case, you will use a synchronisation step to avoid the task owner getting tasks in stages for each and every line and, instead, to receive all tasks at once.



> **Note:** In this example, multiple 'First level approval' tasks are generated and assigned to different users due to the distribution rule.

Sub-processes

A sub-process step is defined and validated as any other process, but reserved for sub-process use only. It can only be executed as part of a main process.

Note the following:

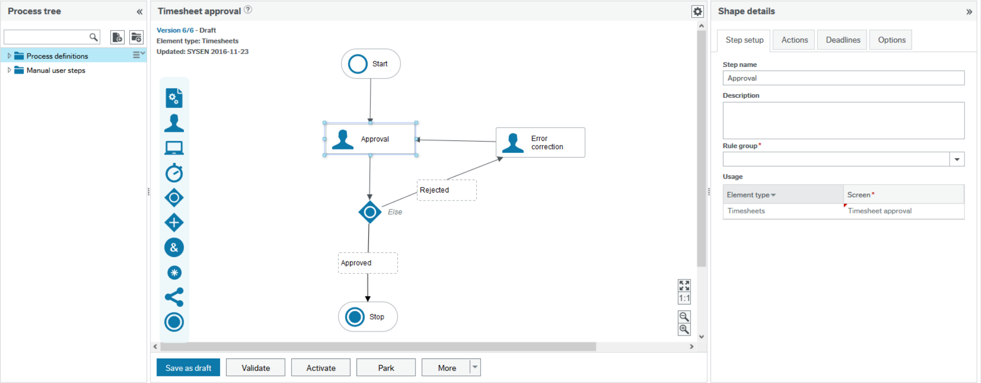
- A process must be activated before it can be used in a main process as sub-process.
- When you have added a sub-process symbol to the main process diagram, you must select the sub-process you want to use. Only sub-processes supporting the same element type as your main process will be available for selection.
- The work item going into the sub process, may be modified before it is returned to the main process (after the sub-process has finished). The main process will still be able to detect *the last recipient in the sub-process* who handled the work item. You can therefore - in the main process - avoid that the same user gets the same work item twice (read about Two step approval).

> **Note:** In the **General configuration** window, you can indicate that a process must be use only as a sub-process.

Working with the process definition window

Window layout

The Process definition window consists of three main areas: Process tree, the modelling area, and Shape details.



Version 6/6 - Draft
Element type: Timesheets
Updated: SYSEN 2016-11-23

Step name: Approval

Description:

Rule group *:

Usage

Element type	Screen *
Timesheets	Timesheet approval

Process tree area

In this tree structure you have a list with folders and process definitions, under the *Process definitions* folder, and with all the tasks that can be triggered directly from the Action overview pane, under the *Manual user steps* folder. Normally, all folders will initially be empty. You would normally create new folders (under *Process definitions*) for related element types.

> **Note:** Use the drag and drop feature to move folders and processes through the tree and the search function to look for processes.

📄	Draft process
⚙	Active process
📄	Parked process
👤⚙	Manual user step being used in a process
👤	Independent manual user step

📁	Folder
≡	Context menu for folders, processes and manual user steps.

> **Note:** Processes with the historical status are only accessible through the **History** window.

Modelling area

The modelling area is used to set up the workflow process. You can build a process diagram by adding *shapes* from the *palette*. You can drag any symbol from the palette directly into the diagram.

Start	Stop	User step	System step	Delay	Or-split	AND-split	AND-join	Sync step	Sub-process	Publish
○	◉	👤	💻	⏱	◈	✚	&	✱	⚙	⋖

Elements inside the modelling area:

- The palette: includes the workflow symbols you can use to build your process diagram.
- A label with the process version information and a link to the History window.
- The action toolbar: lists the actions available for the current version of the process.
- The zoom options: contains several zoom-related buttons. In addition to zoom in and zoom out, you can restore the original zoom and fit the diagram.
- The General configuration ⚙ button.

Options available:

- It is also possible to connect shapes: by using the lines coming out from the ports of the shapes and by dragging and dropping a shape into another shape or into a line.
- For lines, you can set a line colour (using the Shape details section) and create, drag and delete pivots inside the line.

Shapes details

When you select a shape with details in a process diagram, any properties related to the underlying workflow element are displayed in the shapes details area. Here you can set the configuration of Or-split branches, User, System and manual user steps, Delay and Sync steps. You can even set the colour of the lines of the diagram.

Options available (from the menu in the tree and action toolbar)

You will find that most of the actions appear in context-sensitive menus and in the action toolbar:

Element	Options
Process options	These options appear when you click the context menu of a process folder or a process in the Process tree under *Process definitions*. • Folder options: add folder or a process and rename, delete and move the folder. From the *Process definitions* menu you can copy all the processes included under this folder. • Process options: add folder or a process, create a new version, view history, rename, copy, delete and move the process.
Version options	They appear for the current version in the Action toolbar. Options available: save as draft, validate, activate, park, delete version, print, export and import.
Shape options	These options appear when you right-click a shape, a line or any place inside the modelling area.
Manual steps options	From the context menu of a folder or step under *Manual steps:* • Folder option: add a user step. • Step options: add a user step, delete and rename. **Note:** Additional buttons appear when opening a manual user step in *Shape details*: save and validate.

Creating a new process

Typically, you will begin your process definition by naming your process and by choosing which element type(s) you will define for it. You can specify these settings in the **Add process** window. This window appears automatically the first time you add a new process.

> **Note:** It is mandatory to name the process but you can select the element type(s) later.

In this window you also have the following options available:

□*Use as sub-process only*

By selecting this checkbox, we are indicating that this process is going to be used only as a sub-process. In other words, there will be no elements entering the workflow for that process unless this process is included as part of another process.

□ *Leave workflow status unchanged*

This is normally used to avoid an invalid workflow status when you have multiple active processes.

The availability of this option depends on the selected element type.

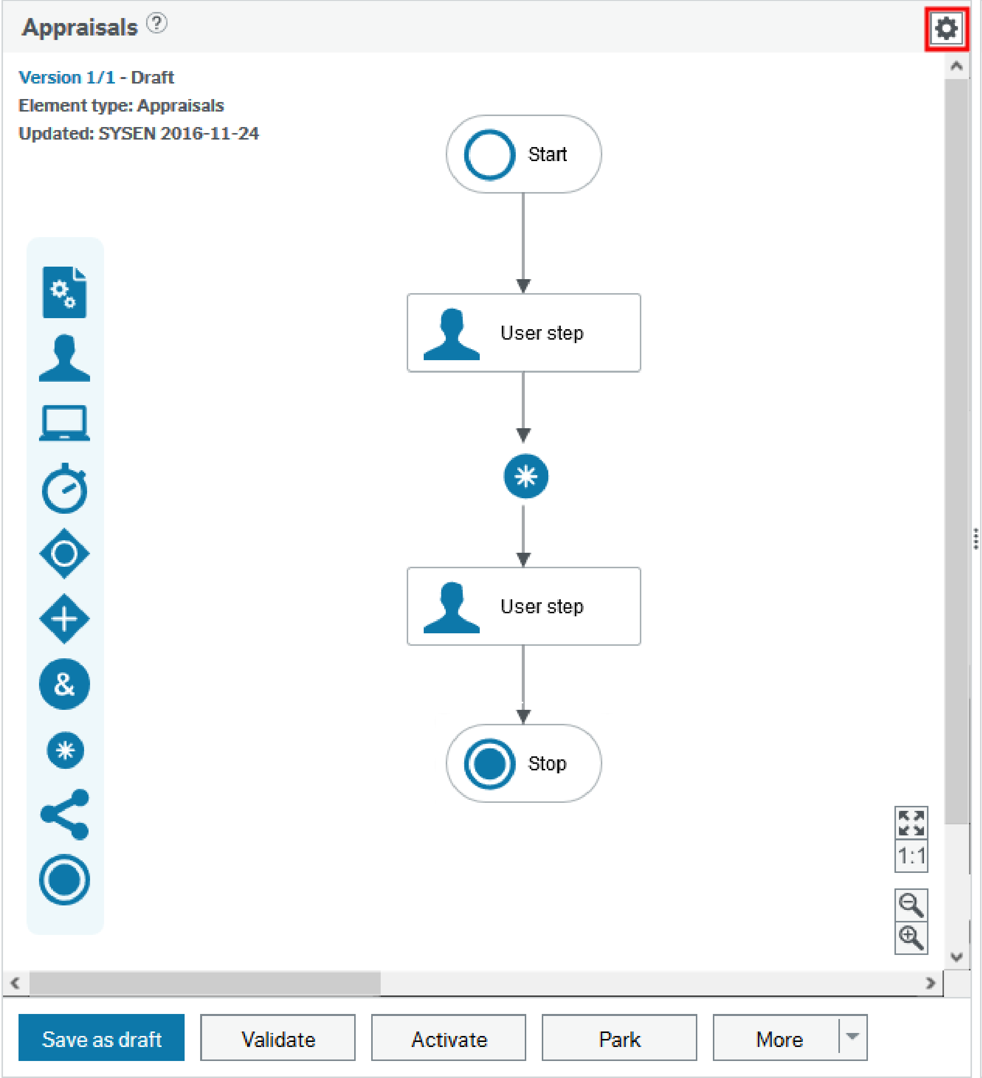
□ *Master file approval*

You select this checkbox when master file data has to be sent for approval for the chosen element type.

The availability of this option depends on the selected element type.

> **Note:** Setting this option implies that you will only be able to choose approval screens in user steps.

⚙ To go back to these settings use the **General configuration** button. Note that once you have activated or parked your version you won’t be able to change the element type(s) or to edit the checkboxes.



Appraisals

Version 1/1 - Draft

Element type: Appraisals

Updated: SYSEN 2016-11-24

Validating and activating diagram

You can always check whether a draft diagram is valid or not, by clicking the **Validate** button. Any validation errors are explained in detail.

Activating the process definition: When the diagram is complete and validated, you must select **Activate** or **Park** (under the action toolbar) to make the process definition active. As soon as you have activated a version, the process will be taken care of by the workflow system, which will then start looking for relevant work items and startup conditions.

Creating a new version

When you want to make changes to an existing version of a process, you have to create a new version using the corresponding menu option. To make a transaction enter this new version of the process, you must activate the version again. This will change the status from **Draft** to **Active** and will create the next version of this process.

Previous versions of the process definition can be viewed in the **History** window.

A transaction can only be in one version at a time. You can move a transaction into another version via Items to follow up.

Deleting a version or a process

To delete the current version of your process use the corresponding button from the action toolbar. You can also delete all versions of your process using the **Delete process** option from the context menu of your process in the tree.

Printing, importing and exporting a process version

In the action toolbar of a process version you have further options available:

- **Print**: it is possible to print the diagram of any process version regardless of the status of the process (draft, active, parked or historical). Information about the version, the element type and the last time the version has been updated will appear next to the diagram.
- **Export** and **import**: you can export and import process versions as **awp** files. When you import a process, only the diagram is imported, therefore you will have to set the different properties: rules, actions, recipients etc.

> **Note:** As long as there are no validation errors, you can export any process version regardless of the status, but you can only import **awp** files to draft versions.

Copying processes

You use the **Copy process** window to copy workflow processes to other clients. This function is available using the menu option for the corresponding process in the tree (for copying various processes, use the option available in the context menu of the *Process definition* folder). Users can only copy processes to the companies they are assigned to in the **User master file** window.

The window usage is straightforward; you simply select the process versions of your interest—one version at a time—and the clients they must be copied to. Then, you just have to save. As a result, the selected processes, as well as any parallel sub-processes, will be copied to the new client(s).

> **Note:** Note that the availability of this functionality depends on the rights given to the client in the **Menu access** window.

Prerequisite: Objects must exist.

There are some objects that may be used by the workflow process which must exist in both clients:

- Attributes
- Attribute values
- Roles
- Users
- Resources
- Flexi-field connections.

Workflow rules are copied

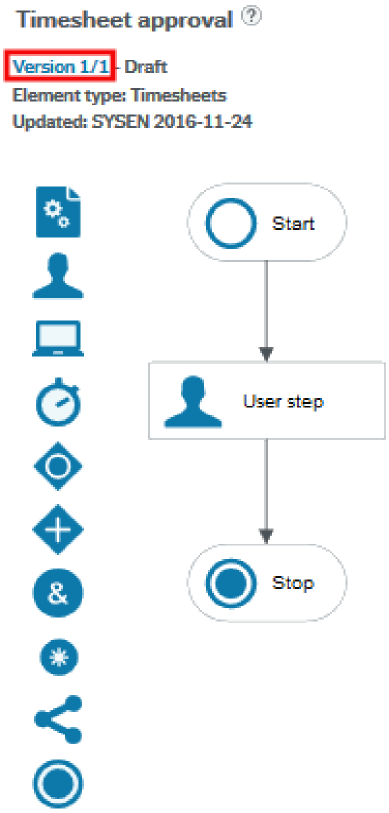
When a process is copied to the new client, all the relevant distribution rules are copied as well. If you copy the same process for a second time, previously copied rules will be overwritten. The full process details and the folder structure are also copied.

Additional tasks for master file approval processes

If you copy a process for master file approval, you must manually create company specific instances for all the functions used in the process. Use the Master file approval setup window for this purpose.

Viewing the history of a workflow process definition

You can view the different versions of your workflow process definition in the **History** window. You can access this window by using the corresponding menu option (**View history**) for each process in the tree, or by clicking the link which appears in the upper left side of the modelling area where the information of the current version is shown in a label.


Timesheet approval

Version 1/1 - Draft

Element type: Timesheets

Updated: SYSEN 2016-11-24

In the **History** window you can find a list with all the versions of the process with the following details: version number, status of the version, date of the last update, User ID, comments and a button to open the required version.

> **Note:** Note that you can open any version of a process but you can’t edit active, parked or historical versions. In those cases, to edit a version of your process you must create a new version.

Defining a system step

System step

A system step is a workflow task that will be processed automatically, either by a pre-defined Business World function or a custom component made available with Business World Customisation Tools.

When the system step is finished, the workflow engine will pass the work item to the next task in the process.

System component — or function

Apart from giving the system step a name and an optional description, you need to identify the *system component* (function) that will process the task. You can also select pre-defined variants for those functions.

Available components for the current element type are listed under *Function*s in the *Usage* section. See System step functions, below.

Custom components: You may use Business World Customisation Tools to create specialised functions for various element types.

Following steps in workflow map

You can use the **workflow map** to see a complete routing of the current work item. Selecting *Hide subsequent steps from workflow map*, the recipient will not be able to see in the workflow map any steps following the current one (will see a clock icon instead), until the current task is completed.

Specific properties

When **Open customer invoice** element type is selected, some specific fields appear in system steps configuration. Those fields allow to include pre-defined texts in the reports generated by the function selected in the system step. For example, it is possible to include a reminder in the report generated by the *Statements of account* function.

You can choose from a list of predefined text types and variants according to the text you want to generate. These texts are defined in **Texts** window (CR10). You have additional fields available, where you can enter possible fees and where you can set reminder levels for the corresponding text.

System step functions

Functions

A system step function will run a component (a .dll) and perform some actions on basis of the current work item. Often, the function will start a server process, and use the work item properties as parameter values. System step function can also trigger a SOAP web service or a RESTful API call. To find out more about service call function, see Service call function setup.

Only a few element types offer pre-defined functions.

Technical reference: Predefined functions for system steps are found in the tables *awfelemtypemenu* and *asyselemtypemenu*, where *s_usage*='**WFR**'.

Functions for Open customer invoices (CRED)

All functions for the CRED element type will start a corresponding server process, using the properties for the current work item (invoice line) as input parameters.

Function name	Description
Collection proposal	Starts the Collection proposal (**CU11**) server process.
Copy invoice	Starts the Copy invoice (**SO07**) server process.
Reminder proposal	Starts the Reminder proposal (**CU09**) server process.
Statement of accounts	Starts the Statement of accounts (**CU04**) server process.
Write-off proposal	Starts the Write-off proposal (**CU16**) server process.

Functions for Invoice proposal (INV)

The table below describes the available functions for the INV element type:

Function name	Description
Invoice	Sets the work item status (invoice line status) to **I** (Invoice).
Invoicing	Starts the Generate Invoice (**TS02**) server process.
Move from invoice proposal to invoice base	Will remove the current work item from the process (the process stops).
Not to be invoiced	Sets the work item status to **C** (Not to be invoiced).
Park	Sets the work item status to **P** (Parked).
Postponed invoicing	Sets the work item status to **N** (Postponed).

Functions for Salary review (SRP)

All functions for the SRP element type will start a corresponding server process, using the properties for the current work item (an *apntransaction* row) as input parameters.

Function name	Description
Salary review letter	Starts the Salary review letter (**PS06**) server process.
Salary review printout	Starts the Salary review printout (**PN12**) server process.
Salary review update	Starts the Salary review update (**PN10**) server process.
Standard letter	Starts the Standard letter (**PS06**) server process.

Functions for Travel expenses (TIN)

The table below describes the available functions for the TIN element type:

Function name	Description
Posting travel invoices	Posts travel expense using the default payment processing method selected for the processed travel type.

Incoming invoice (IIN)

The table below describes the available functions for the IIN element type:

Function name	Description
Set invoice ready for auto-post	Will update the *po_flag* value on the work item and thus make it acceptable by the Posting of authorised invoices (**EI03**) server process. The work item will be posted next time **EI03** runs. **Note:** The system parameter EI03_AUTO_ VARIANT must identify the EI03 variant to
Distribute evenly between all lines Distribute proportionally between all lines Distribute to highest value Add to extra line	These four functions define how an invoice discrepancy amount can be: - Evenly distributed on the matched order lines - Proportionally distributed on the matched order lines - Put on the highest matched order line - Added to an extra line

Miscellaneous functions

The following element types offer one function. Most of these functions will just start a corresponding server process on the basis of the current work item:

Element type	Function	Description
Assignment (BOOK)	Calendar item	Will create an Outlook appointment on basis of the assignment - for the resource and time period identified by the current work item.
Structure setup draft (STDR)	Update relations based on Modeler	Will run the server process with the same name, and thus implement the structure changes from the draft.
Service order (OS)	Work order printout	Will run the Work order printout (**TS21**) server process.
Purchase invoice (PIN)	Posting of authorised invoices	Will run Posting of authorised invoices (**EI03**).
Personnel (RES)	Standard letter	Will run Standard letter **(PS06**).
Timesheet (TS)	Close timesheets	Will run Close timesheets (**TS52**).

User step setup

Tab purpose

In addition to giving the user step a meaningful name and (an optional) description, you must always connect the step to a distribution rule group and select a window (Usage section) and at least one action (Actions tab) for task completion.

If required, you can also make the task directly available as an action.

Connect to rule group

You must always connect a user step to a rule group, even if you don't need any rules to find a recipient. In the latter case, you will probably use an empty rule group, just to satisfy the formal requirement.

See Distribution of tasks for details.

Adding actions to tasks

You can add actions to items in workflow processes such as Toolbar buttons (**Action** tab) and, when the relevant conditions are met, as rows on the list of actions for an item (**Action overview** screen). Clicking on such button or action row, the relevant action window opens with the corresponding details on the workflow action.

Action overview panel

Action overview is the main panel used for action management in Business World. Many element types have support for action management. The Action overview panel appears as an extension to the windows we use to handle objects of these types (an object - or a record - is called a work item when it is included in a task).

Work items with parents or owners: A few element types, for example a customer invoice, belong logically to another element type, almost like "child" to a "parent" relation; a customer invoice always belongs to a customer.

You can add actions to both levels, but only if two conditions are in place:

- the current work item has an owner (parent element type),
- the screens used to handle the child and parent objects respectively, both have support for the Action overview panel.

Action overview options

The following options are only available for element types supporting action overview:

□ *Action overview in parent element type*

Only relevant when the current work item has a parent or an owner. The action linked to the current work item will appear on the list of actions in the Action overview panel of the screen used to handle the "parent" object.

□ *Step available for manual use*

If selected, the step can be used to create an ad-hoc action, for any record of the current element type. A new manual step corresponding to the current user step will then appear in the Process tree under the *Manual user steps* folder, and as an action in the action overview tab of the corresponding element type. See Manual user step.

Usage

In the *Usage* section you can define the screen the recipient will use to handle the task. The type of screens available depend on the selected element type.

The columns are explained below:

Column	Description
Element type	The corresponding element type.
Screen	The screen or window name where the task's work item will be displayed. **Note:** Note that some *screens* will not display workflow action buttons for some element types. The solution can then be to use an Action type and Action overview screen (below) to display the action buttons.
Action type	If the screen has support for the Action overview panel, here you can define the action type to be used.
Action overview screen	The action window opened from the Action overview panel. It is automatically defaulted when selecting an action type.

> **Note:** Action type and Action overview will be only visible for element types with the **Action** column (property) selected and with the *Master file approval* checkbox in the **General configuration** window deselected.

User step actions and recipients

Workflow actions and general actions (using the Action overview panel)

Sometimes Business World does not provide a suitable screen with the ability to host action buttons. In these cases, you will typically select an *Action type* and an *Action overview screen* for task completion - if the main screen has support for the Action overview panel.

The action buttons will then appear in the selected action window (labelled *Action overview screen*) when the recipient opens the action, and one of them must be used for task completion.

See also Action management in workflow.

Workflow actions for user steps

It is mandatory to select at least one action in a user step, otherwise the user who receives the task will not be able to perform any action on it (complete the task). You can choose from a list of actions in the **Actions** tab. The actions selected will be available as buttons in the window(s) where the user will process the task.

Actions tab in User steps

The list on the left in the Actions tab shows all available system and user defined actions. You select the actions the recipient can use to complete the task by adding them to the list on the right. For further explanation of the actions and their corresponding functionality see predefined system actions and user defined (custom) actions.

Choose recipients for action

The recipients of a user step are normally identified by a distribution rule group (exceptions: when the routing may go back to a previous task owner - or to the process initiator, as part of an or-split).

The Inform, Forward and Distribute actions allow the task owner to involve new recipients for a task. Unless you explicitly select specific roles or users, the task owner can select new recipients from all the roles and users allowed to take part in workflow processes.

The action ...	lets the task owner select users who will ...
Inform	Be just informed about the task.
Forward	Take over responsibility for completing the task. The task owner is no longer responsible for the task.
Distribute	Share responsibility (with the task owner) for completing the task.

In order to select specific roles or users, in the *Choose recipients for action* dropdown list select the corresponding action; then choose from the *Available* list the recipients you will enable for the task owner. You can filter by users and roles.

User step deadlines

Actions

When a user step is distributed, you can define certain actions to be performed based on different conditions and including different targets:

- Approve automatically a task at different points in time.
- Notify users when a task is pending or taking too long.
- Escalate a task.
- Show a task as due date.

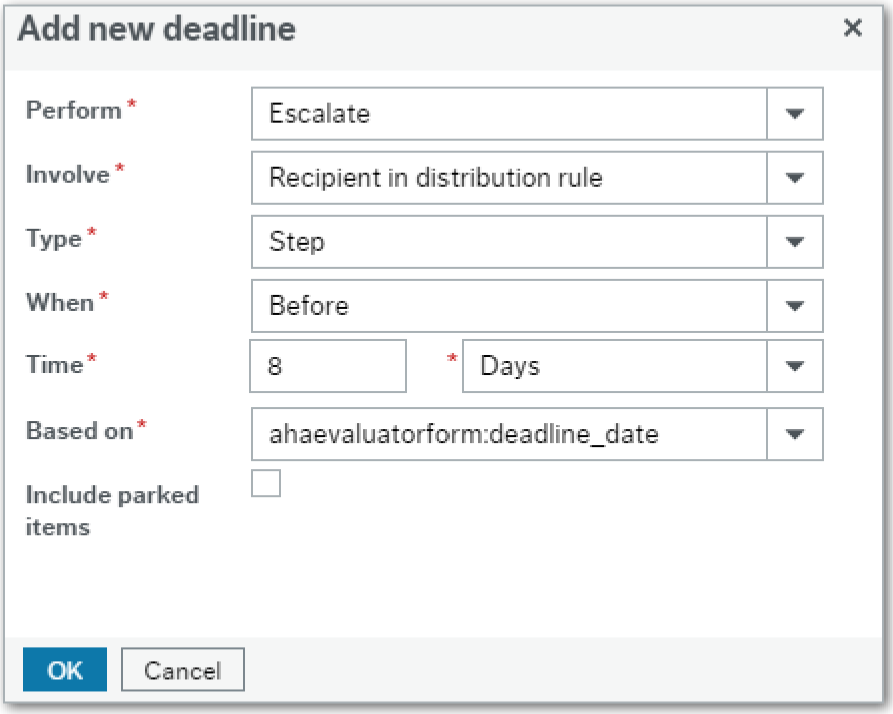
Actions and alerts: When you select an action, you will also need to specify who the target user (recipient of the action) must be.

When you just want to send a reminder to the original task recipient, you must also specify *how* the reminder must go to the target user (options are Email, SMS/Text and Alert list), as well as, to specify the *content* of the reminder. This will require that you have defined a custom *Alert* which will be triggered when the deadline is reached.

An Alert is handled by **IntellAgent**, and you use the Alert setup window (Common > Workflow > Alerts > Alert setup) to define alerts for workflow related events.

The Deadline setup window

You define the deadline rules in a pop-up window. Here you have an example:



Add new deadline

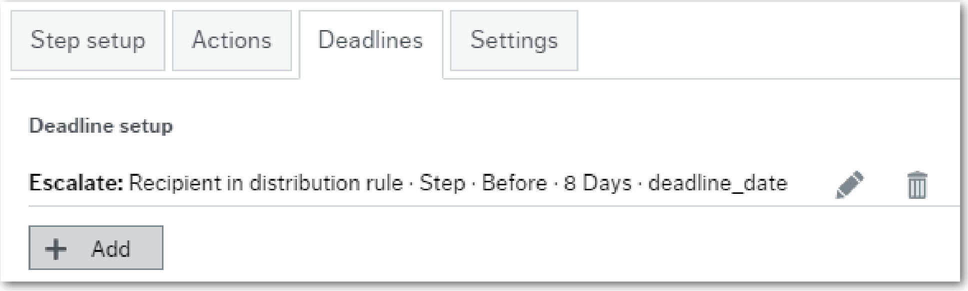
Perform *	Escalate
Involve *	Recipient in distribution rule
Type *	Step
When *	Before
Time *	8 * Days
Based on *	ahaevaluatorform:deadline_date
Include parked items	□

The options available depend on the selected values.

Perform	Involve	Type	When	Time	Based on	Include parked items
Reminder						
Escalate	User		After	Days		
Approve	Supervisor	Step	Before	Hours		
Show as due date	Recipient in distribution rule	Task	Scheduled	Minutes		
User defined actions			Running	Weeks		

Once you have set up your deadline actions, the values are displayed in rows. The action to perform appears at the head of the row followed by the rest of the values in the same order as the fields in the deadline window.

Set up schedules will also appear here to indicate the next scheduled event. See Schedule setup.



Deadline setup

Escalate: Recipient in distribution rule · Step · Before · 8 Days · deadline_date

Actions (Perform)

There are a few predefined action types available, where two of them require that you identify the target user for the action in the *Involve* field.

In addition, Custom actions (user defined actions) which are valid for the current element type, and marked as *Automatic action*, will be available.

The various action types are described below:

Action type	Description
Reminder	Notifies the original recipient or supervisor about the task. Note that a Reminder action requires that you have defined an *alert*, valid for reminders and the current element type. See description of Alert setup.
Escalate	Forwards the task according to the selected value in the *Involve* field: • To the supervisor, if defined. • The recipient is allocated according to the current distribution rule. Note that you can also send a notification - via a tailored alert - if you select the **Escalate** action.
Approve	Will approve the current work item, and thereby complete the task automatically.
Show as due date	The deadline will appear as due date in the task list.
<Custom actions>	The deadline will automatically trigger the custom action, meaning that the task result will be either positive or negative. Further processing will occur according to the action result, and the task will be removed from the original recipient's task list.

Target users (Involve)

You identify the target user in the *Involve* field. The only actions that require a target user are **Reminder** and **Escalate**:

- Reminder - allows you to send an alert to the original recipient (**User**), or to defined **Supervisors**.
  You can also choose that the recipient will be identified according to the distribution rule (**Recipient in distribution rule**).
  A reminder does not affect the actual task processing, and must always have a valid

alert - defined in Alert setup.

- Escalate - transfers the task ownership to identified **Supervisors**, or according to the escalation sequence defined for the current distribution rule (**Recipient in distribution rule**). In addition, you can also send an alert (an additional notification) to the new recipient - if such alert is defined for the element type.

Deadline types

Deadline setup allows you to define several rules, where each rule is valid either for a *Step* or a *Task*:

- The *Step* type refers to the task (the user step with possible sub-tasks) as a total. A deadline action for a Step will be performed once. If the user step has many sub-tasks, the action will only be performed for the first sub-task that matches the deadline condition.
- The *Task* type refers in principle to all the tasks (sub-tasks) inside a user step. If a distribution rule defines complex routing, the action will be performed for all the sub-tasks (that match the deadline condition).

When is the deadline happening

The *When* field offers different options, used in conjunction with other fields:

Value	Description
After Before	Uses an element type column which you identify in *Based on*. The actual deadline is set up in the *Time* fields (e.g. **3 Days**). This option is only available for element types with at least one date property set with a *Delay / Deadline setup* column (you select columns for various purposes when you define the element type details).
Running	Refers to the moment when a user step becomes active: when a task has been received by the recipient. After the step is active, you can set a period of time for the recipient to perform the task (using the actions selected in the**Action** tab). After this period of time, if the task hasn't been performed, the system will automatically complete the action selected in*Perform*. The actual time deadline is set up in the*Time* fields. **Task or Step:**

Value	Description
	- If the type is **Task**, and the user step is set up with complex routing, a certain sub-task may not become active (be ready to be performed by the recipient) until several previous sub-tasks are processed. - With a **Step** deadline, however, the user step will become active when the first sub-task enters the step. This is useful in cases where you just want the step to be active for a period of time, regardless of when the rest of the sub-tasks arrive.
Scheduled	The deadline action will be performed according to a scheduled setup. You can select a basic frequency period (per day, week etc.), and specify a exact time for the action.

Schedule setup

By selecting **Schedule** in *When*, you can set the following options:

Event	Recurrence	Occurrence
Daily	- Every selected number of days. - Every weekday (from Monday to Friday).	- Occurs at a set time. - Occurs in an interval of time (expressed in hours or minutes) defined by a start and end time.
Weekly	- Recurs every selected number of week(s) on the selected day(s).	- Occurs at a set time. - Occurs in an interval of time (expressed in hours or minutes) defined by a start and end time.
Monthly	- Recurs a set day per month and every selected number of month(s). - Recurs every first, second, third, fourth or last weekday, of every selected number of months.	- Occurs at a set time. - Occurs in an interval of time (expressed in hours or minutes) defined by a start and end time.
Yearly	- Recurs on a set date - Recurs every first, second,	- Occurs at a set time. - Occurs in an interval of time

Event	Recurrence	Occurrence
	third, fourth or last weekday, of every selected month.	(expressed in hours or minutes) defined by a start and end time.

The Time fields

You use *Time* to define when the deadline action will be performed. When selecting **Running**, these fields refers to a moment *after* the task became active.

The Based on field

The *Based on* field is only relevant if in the *When* field you have selected **Before** or **After**.

Based on will allow you to select a date property (table column for the element type, representing a date) which must have been set with the *Delay / deadline setup* column in the **Details** tab (in the **Element types** window). Unless you have at least one Delay / deadline setup column, *Before* and *After* cannot be used.

Parked items

Selecting this checkbox you indicate that the deadline action applies also for parked tasks.

User step options

The Options tab

You can use the Options tab to define some additional rules for the user step, not covered by another setup. The various options are described below.

Specific properties

There are some specific setups which are only available for some element types:

Element type	Specific options
Incoming invoice **Requisition** **Supplier invoice**	- A new option is shown to decide whether or not to validate the step according to the account rule.
Travel expenses	- The *Process control* field is shown. Possible values for this field are those available for the **C18 PROCCTRL** attribute. The chosen value will set the workflow routing. - This field can be used to decide whether or not a step is a requisition on a travel. **Note:** Process control is linked to the setup of the **Travel rules** window (TTT017).
Salary review	- Two available fields: *Salary review status* and *Group type*. - The possible values for the *Salary review status* field are those available for the **C17 STSTATUS** attribute. - These fields can be used to set the status of a transaction when it reaches a certain step in the process (based on the group type).

Users allowed to change data

When a user is allowed to change property values for the current work item, this may have an effect on how the task is distributed, in other words: how it *would have been distributed* if the new property values had been incorporated in a previous task, or when the work item was first saved.

If there is a possibility that there may be changes that require new processing, you can select what to do (*Treatment if change triggers new rule*).

The options are:

- **Full redistribution**: The process goes back to the start and runs with the updated work item.
- **No new distribution**: The process continues, unaffected by any changes for the current step.
- **Redistribution within step**: The process will go back to the previous task, and will be distributed according to the properties of the new work item. Changes made in the work item may lead to a quite different distribution (going to other recipients) but may also cause previous recipients to get the same task once more.

> **Note:** In order to have the **Split row** action available in a user step (Action tab), *User allowed to change data* must be selected.

Approval screen settings

The Approval options sections are related to how the work item must be displayed in the approval window, and how the recipient must approve the current task.

These settings are mainly relevant for transactional work item types, where the work item (for example, an invoice detail) is part of a larger object (represented by the invoice header).

□ *Allow processing on header level*

If a recipient receives several tasks (e.g. invoice details), all parts of the same header object (invoice), you can allow the recipient to approve (or reject) all related tasks in one go.

□ *Display entire transaction*

If selected, the recipient will see all detail rows belonging to the same header, regardless of the recipient. The recipient cannot approve or reject tasks belonging to other recipients.

> **Note:** Note that there is no data control applied to this option, and users may potentially see data they would normally be prevented from seeing.

Advanced or simple mode: The approval window will - by default - open in so-called simple mode, allowing the recipient to approve tasks without going into the transaction details. The window will have a button though, allowing the user to switch between simple and advanced mode.

□ *Advanced mode*: If selected, the window will open in advanced mode, and require approval on a detail level (but the recipient can always switch back to the simple mode).

□ *Disable simple mode*: Only available when the Advanced mode is selected. If you disable the simple mode, it will not be possible to switch between the two modes.

User validation on save: You may require that the approval or rejection action is confirmed by a password.

Approval options

Automatic approval

The automatic approval options of this section allows you to increase workflow efficiency in two special cases:

- When the user who initiated the workflow (registered a new record - work item - or changed a master file record) is identified as the recipient of the current task and the work item hasn’t changed.
- When the task recipient turns out to be the same user as the recipient of the previous task in the process.

The options are explained below:

Option	Consequences when selected
Automatic approval	If the task recipient is the same as the user who initiated the current process instance, and the work item has not changed since its registration, the step will be skipped and the task set to approved.
Two-step approval	If the task recipient is the same as the recipient of the previous task, the step will be skipped and the task set to approved. **Note:** Note: Does not apply to consecutive sub-tasks.
Exclude manual approvers from two-step approval	Only relevant when *Two-step approval* is selected. If selected, manual approvers are excluded from two-step approval. **Note:** Selecting this option, when a user is manually assigned to the current task during a process instance, may receive the same task twice (assuming you have more than one recipient associated to the previous task).
Enable e-mail approval	Enabling e-mail approval allows workflow tasks to be approved by replying to task notification e-mails with actions and comments. "E-mail approval" on the facing page for details.

E-mail approval

This option allows you to handle workflow tasks by replying to task notification e-mails with (custom or user defined) actions and comments.

How to set the E-mail approval function

Set up of e-mail approval consists of a technical configuration performed on the management console (see the technical guidelines for details), and a functional configuration performed on Business World as follows:

1. If not already done so, define the recipient e-mail address in the **User master file** window.
2. Select the *Enable e-mail approval* field for the relevant steps in the workflow process to make it possible to process tasks via e-mail.
3. From the **Alert setup** window, edit the workflow alert template to include:
   - the **[E-mail notification]** tag in the subject field. For steps that have e-mail approval enabled this tag will be replaced with a token that is used to identify replies. For other steps this tag will be blank.
   - the **[Legend]** tag. This is optional but recommended. For steps that have e-mail approval enabled this tag will be replaced with a list of the actions possible on that step. For other steps this tag will be blank.
4. Choose a document type in the **Alert setup** window to include documents attached to items on workflow in task notification e-mails. This functionality can be used independently of e-mail approval.
5. Optionally. to make it easier for users, you can register synonyms in the **Workflow action synonyms** window for actions that can be used when performing actions via e-mail. These synonyms are then recognized as aliases for actions, for example **OK** for **Approve** and **No** for **Reject**.

How users process e-mail approval tasks

Once the workflow process runs for this task, the recipient receives an e-mail with the actions selected in the User step. To complete the task, the user must answer the email with name of the corresponding action to perform on the task and with a mandatory comment if the action was a negative one (with no more than 1000 characters). For example: **Reject. We never received the invoiced goods.**

The task will then disappear from the user's task list (unless the user had parked the task).

The workflow service will send back an email to inform the user that the task has been processed successfully. This e-mail will contain the action performed by the user and the possible comment included.

Note: The following actions are not available for email approval: distribute, inform and forward.

Step instructions

You can enter up to 255 characters with useful information for the task recipient. The text will appear in the task's approval page.

Step options

The *Step options* section provides the following options.

Option	Description
Hide subsequent steps from workflow map	Selecting this option, the recipient will not be able to see in the workflow map any steps following the current one, until the current task is completed. Relevant when the task owners of following tasks are determined via manual distribution and not via a distribution rule.
Include in single user involvement check	Sometimes, although you may have several tasks in a process (intended for different recipients), the distribution logic may find the same recipient for all the tasks. For example, when a section manager acts also as a substitute for the division manager. To avoid this, you can make workflow check any other recipients involved in all the tasks (user steps) for the current process instance. To do so, first you have to select in the corresponding steps the *Include in single user involvement check* option. Then, it is important to select one of the included tasks in *Perform single user involvement after this step* to start the checking process. **Note:** If there is only one user, the task will be treated as un-handled, and be directly transferred to Items to follow up.
Perform single user involvement after this step	

Sharing

By enabling sharing, you allow the original task recipient to hand over the task responsibility to a certain role - after the task is received. The task will then be available for all members of the selected role. Initially, the original recipient will still be considered responsible for the task, and will remain so until someone actively takes over responsibility for the task.

When a role member accepts - or takes over - the task, it will no longer be available as a shared task (but it can be shared again by the new task owner).

Roles: Available roles are limited to *roles enabled for task sharing*, and *where the recipient is a member*.

> **Note:** Currently, task sharing is only available via the **Task list** in the Task management Experience pack.

OR-split setup

OR-splits for alternative routing

An OR-split is used to decide the workflow routing on the basis of certain conditions, either property values in the current work item, or an action performed in the previous step.

A routing direction is called an OR-split branch, or simply a branch. An OR-split can have numerous branches.

OR-splits and distribution rules

There are many similarities between distribution via an OR-split, and distribution via the detailed setup in the Routing section for a distribution rule.

When it is required that a certain task is repeated until some criteria are met, or when you will send the work item directly to a system step on the basis of certain conditions for example, you will need an OR-split. Or if you need advanced conditions for the distribution, there are only available as OR-split functions in OR-splits, providing a range of advanced conditions.

> **Note:** If you have more than one element type available for the process, you can select which one must be used in the Or-split.

Criteria-based or default

You can define several branches based on different criteria. Default branches are mandatory, as in some cases none of the defined criteria may match the current work item.

Business World provides you with two options for an OR-split branch:

- Criteria-based, (*If* branch) i.e. a branch based on an action, property values on the current work item, or functions.
- An Else-branch, used as default when no criteria-based branches match.
  An OR-split must always have one *Else* branch (even if you don't actually need it).

Adding a new OR-split branch to the process diagram

By default, there are two ports on an OR-split shape to indicate a possible *If* or *Else* branch.To create a new branch, just drag one of the ports and connect it to the corresponding shape. To set the properties of each branch, simply click the branch labels (*New split criteria* or *Else*) or click the lines coming out of the Or-split.

Selecting recipient in next step

When the next task after an Or-split is a user step, the recipient can be defined by the distribution rule or by selecting recipients from previous participants in the process.

The possibilities for selecting recipients are:

- **According to rule group from next step** - the recipient of the task is located according to the distribution rule in the next user step. This is the default option when selecting recipients and will not have any real impact on the recipient of the next user step as it was going to be determined by the distribution rule either way. The process will disregard the distribution rule from any previous user steps.
- **Workflow initiator** - the task will be received by the user who originally registered the transaction on workflow. This option prevails over the distribution rule in the user step. One example of usage is when a branch handles an initial registration error.
- **Return to last owner** - the recipient is the user who was the previous owner of the task. This option prevails over the distribution rule in the user step. One example of usage is when the previous task allowed data entry, and the branch finds that a certain limit is exceeded.

  In this case, the Or-split branch must be connected to a previous user step, or the task goes to *Items to follow up* (the rollback function needs to be represented graphically).

> **Note:** Note that even though an OR-split is set up to rollback to the workflow initiator or to the last owner, the next user step, typically a correction task, must still be set up with a distribution rule.

In all the above cases:

- These options are available for both branches (*If* and *Else*).
- All of them must be connected to a user step.

The Else-branch

All OR-splits must have one, and only one, default branch; one branch in an OR-split must have the *Else* option selected. You can select recipients in the next step for an *Else* branch. Apart from that, there is no special configuration of an Else-branch.

Criteria-based OR-split branch

Criteria based on

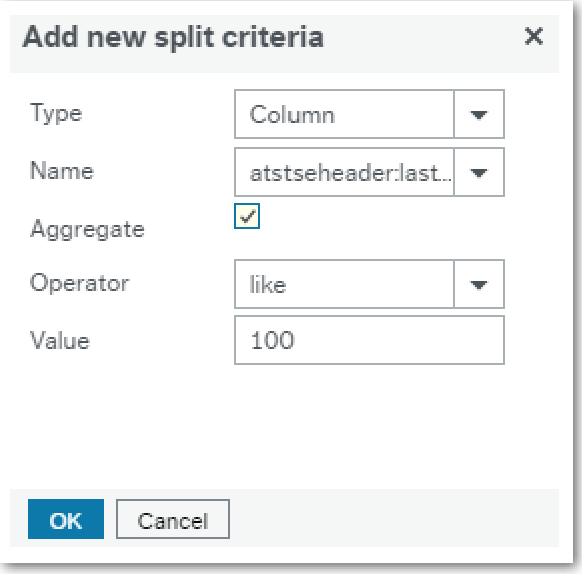
A criteria-based (*If*) branch is either based on:

- an action from the previous step (must be a user step)
- property values in the current work item
- functions

> **Note:** If required, you can use the same action for several branches. This requires that you also add property-based criteria to the other branches in the *Split criteria* section to differentiate them.

Defining criteria

You use the *Split criteria* section to add as many criteria as you need. Here you have an example:



Add new split criteria

Type	Column
Name	atstseheader:last...
Aggregate	☑
Operator	like
Value	100

The following options are available:

Criteria type	Description	Available fields
Attribute/Relation	Allows you to select attribute or relational values. Only available for attributes with the **workflow** checkbox selected.	Attribute Relation Operator Value
Column	Used when you want to move the process in a certain direction based on a column available in the element type of the transaction. Only available for columns with the *OR-split* checkbox selected for that element type in the **Details** tab (in the **Element types** window)	Name Aggregate Operator Value
Function	Used when you want to select functions associated to the element type to set up the criteria (functions for the corresponding element type must be enabled). See OR-split functions for details.	Function Operator Value
Based on action in previous step	Here you can choose as split criteria between any of the actions, either user or system defined, selected in the previous user step. It is not possible to add a split criteria based on actions if more that one user steps are placed immediately before an Or-split shape. You will always find one action available, **Auto approved**. See automatic approval.	Action Some of the following values may be available for selection in the action field: Auto approved Approve Reject Accept Task complete Receive goods Manual user defined actions

Criteria type	Description	Available fields
Master file - new	Only available for element types defined as master file and with the **Master file approval** checkbox selected.	
Master file - approval	Only available for element types defined as master file and with the **Master file approval** checkbox selected.	

All types require that you construct criteria that can be evaluated to **true** or **false** (the exception is the **Auto approve** action). With more than one criteria, the branch will only be selected if all the criteria are **true**.

> **Note:** If you have more than one element type available, it is possible to select the element type for which you are going to set up the split criteria. Note that some element types may not have split criteria available.

OR-split functions

Function result

An OR-split function is a component which performs certain operations on a work item - and possibly related work items for transactional element types - and produces a result that you can use in an OR-split criterion.

You need to know the result type to set up the criterion correctly.

Technical reference: All the available functions are available in the table *asyswfblmethods*, where the return type is indicated in the *data_type* column:

b = boolean (true or false, same as 1 or 0),

d = number (double),

s = string.

Custom components: You can use ACT to create tailored components for all element types. The details are described in the *Customisation Tools* reference manual.

Functions for Incoming invoice

The following functions are available for Incoming invoice (IIN):

Name	Description
Contract invoice	Returns **true** if all incoming invoices are linked to a contract and routes them onwards in the workflow process.
Credit note	Returns **true** if the incoming invoice is a credit note.
Discrepancy	Returns **true** if there is a discrepancy in the invoiced amount, invoiced quantity or tax regarding purchase order. The invoice is subsequently routed onwards in the workflow process.
Discrepancy amount	When invoices are imported through **EI02 Import of invoices** and matched in the **IMS** (Invoice matching service), normal GL lines are generated based on the goods received numbers for the current purchase order. The differences between the actual invoice and the sum of the generated GL lines end up on a special GL line, often referred to as the discrepancy line. This function returns the amount on the discrepancy line and can be used to decide whether an invoice must be

Name	Description
	automatically posted or sent for approval for manual matching. This is only valid for summary invoices and is also depending on how the workflow process is modelled.
Extra invoice lines	Relevant for Purchase orders. Returns **true** if the invoice (all detail lines) contains more lines than the purchase order (if such an order exists). The invoices are subsequently routed onwards in the workflow process. Note that extra lines need to be identified by entering a minus sign (-) before the line number.
Extra order lines	Relevant for Purchase orders. Returns **true** if the supplier's order confirmation contains more lines than the purchase order. Additional lines are identified in the *apodetail.flag* with value **E**. This function identifies those lines and routes them onwards in the workflow process.
Funds check	Returns **true** if calculated spending, including the amount on current work item, exceeds the budget (**false** otherwise). Funds checking is part of Commitment accounting, and the Funds check function is only of interest if you have a relevant commitment setup.
Invoice control is exceeded	Returns **true** if invoice control is exceeded (*Quantity delivered*, *Amount ordered*, *Amount delivered*) according to invoice control setup. This check is run against each order line.
Missing goods received	Returns **true** if there are missing goods.
Overrun amount	Checks the current invoice amount against the amount set up in the *Value* function. Use this function to allow invoices with small amounts to run more smoothly through the approval system without manual interaction, or to force large discrepancies on workflow.
Purchase invoice	Returns **true** if all invoices contain purchase order numbers and routes them onwards in the workflow process.
Supplier invoices	Returns **true** if all invoices are standalone and routes them onwards in the workflow process.
Total invoice amount	Returns the aggregated amount for all detail lines related to the same header.

Functions for Goods received

There are three functions for Goods received (GRN):

Name	Description
Added row in receipt, send all rows to workflow	Returns **true** if a row has been added in the **Goods receipt** window, all received rows will then be sent to workflow.
Additional row(s) added in goods receipt	Returns **true** if row(s) have been added in the **Goods receipt** window and gives you the possibility to route these additional lines.
Stocked	Returns **true** if the received item is stocked.

Functions for Missing goods received

There are three functions for Missing goods received (MGRN):

Name	Description
Credit note	Returns **true** if the current work item belongs to a credit note.
Difference between qty. invoiced and received	Returns **true** as long as the invoiced quantity is different from the received quantity.
Invoice qty. fully received	Returns **true** if the complete invoice quantity is received.

Functions for Purchase invoice

The following functions are available for Purchase invoice (PIN):

Name	Description
Credit note	Returns **true** if the current work item is a credit note.
Discrepancy	Returns **true** if there is a discrepancy in the invoiced amount, invoiced quantity or tax regarding the purchase order. The invoice is subsequently routed onwards in the workflow process.
Exceeds max overrun amount	Returns **true** if the invoice amount exceeds the value of the EI_MAX_OVERRUN_AMT parameter.
Exceeds max post amount	Returns **true** if the invoice amount exceeds the value of the EI_MAX_POST_AMT parameter.
Exceeds overrun pct amount delivered	Returns **true** if the invoice amount exceeds the invoice control *Amount delivered* value.
Exceeds overrun qty delivered	Returns **true** if the invoice amount exceeds the invoice control *Quantity delivered* value.
Exceeds overrun percentage of amount ordered	Returns **true** if the invoice amount exceeds the invoice control *Amount ordered* value.
Extra invoice line(s)	Returns **true** if the invoice (all detail lines) contains more lines

Name	Description
	than the purchase order (if such an order exists). The invoices are subsequently routed onwards in the workflow process. Note that extra lines need to be identified by entering a minus sign (-) before the line number.
Extra order line(s)	Returns **true** if the supplier's order confirmation contains more lines than the purchase order. Additional lines are identified in the *apodetail.flag* with value **E**. This function identifies those lines and routes them onwards in the workflow process.
Full match	Returns **true** if the invoice fully matches the order (there are no differences).
Less than minimum approval amount	Returns **true** if the invoice amount is lower than the value of the EI_MIN_APPR_AMT parameter.
Missing goods received	Returns **true** if the goods are received. If the goods are not received, the function is activated and the invoice is sent to missing goods received.
Ready for posting	Returns **true** if the transaction is ready for posting.

Functions for Order confirmation

There are two functions for Order confirmation (OC):

Name	Description
Order line changed	Returns **true** if the line being processed in workflow has changes. The result is that only the lines with changes will be processed by workflow.
Order changed	Returns **true** if any of the lines being processed in workflow has changes. As a result, with just one line presenting changes, all the lines will be processed by workflow.

Miscellaneous functions

The following element types provide one function:

Element type	Function name	Description
Assignment (BOOK)	Calendar item	Not to be used. Will be removed.
Commitment	Funds check	Returns **true** if calculated spending, including the

Element type	Function name	Description
(COMM)		amount on current work item, will exceed the budget (**false** otherwise). Funds checking is part of Commitment accounting, and the Funds check function is only of interest if you have a relevant commitment setup.
GL transaction (GL)	Funds check	As above.
Purchase order (PO)	Funds check	As above.
Report file (REP)	Report has comments	Return **true** if current work item has comments.
Requisition (REQ)	Funds check	As above.
Travel expenses (TIN)	Funds check	As above.
Timesheet (TS)	Timesheet condition	Returns the name of the screen where the Timesheet was entered. The idea is that a rejected timesheet can be passed back to the workflow initiator - and the correct screen. Possible return values are: **TTS025**, **TTS026**, **TTS054**.

Service call related function

Element type	Function name	Description
All element types with service call set up	Successful HTTP response	Returns **true** if service call was successful, otherwise returns **false**. For more information, see Successful HTTP response OR-split function.

Delay setup

Delay until conditions are met

A delay is usually inserted in a process definition, with the intention to reserve the process for work items that fulfil certain criteria. You also have the possibility of defining an additional delay before work items move to the next task and of setting how often delay conditions must be checked.

> **Note:** If you have more than one element type available, it is possible to select which element type you are going to set up the delay criteria for. Note that some element types may not have criteria available.

You define the criteria - as many as you need - on the basis of property values of the work item:

Criteria type	Description	Possible fields
Attribute/Relation	Here you can either select a value for an attribute property (no relation) or a value for a specific attribute which is related to an attribute value of the work item. Note that relational values may change over time. See Dates on relations for more details. Only available for attributes with the **workflow** checkbox selected.	Attribute Relation Operator Value
Column	Used when you want to move the process in a certain direction based on a column available in the element type. Available columns are selected in the Element type detailswindow, in the**Details**tab, under theDelay / deadline setup column.	Name Operator Value
Time	You can specify before or after intervals for a **DateTime**property (which must have been selected in the**Details**tab).	Value Time type Based on

Additional delay

When required, you can specify a time delay, either as an addition to a set of conditions, or as a general delay for all work items matching the criteria.

An additional delay simply means that the process will not continue to the next task until a certain time period has elapsed.

Delay check interval

You use the *Minutes* field to tell workflow how often it must check the database for records that meet the conditions. Setting this option, work items – matching the criteria – will have to wait to enter the process until the check function is performed.

> **Note:** The default value in *Delay check interval* is **360**. Any value you enter below 1 will always take 360 minutes in code.

New items only

You can choose to set up the delay for *New items only* (i.e. new work items only), or for all items.

When this option is selected, the system will only check delay conditions in new records (work items) saved after the process has been activated for the first time.

As soon as an item has been processed once, it will no longer be new, and therefore not be included the next time workflow checks the delay condition.

Consequences: If you, for example, activate a new process where you have an initial delay reserving the process for customers belonging only to a certain group , and where *New items only* is checked, the process will not be valid for the existing customers in this group.

If you don't check *New items only*, all records (of the relevant element type) in the database are potentially able to meet the delay conditions the first time they are checked. After that however, only new items will be valid for the process.

General rule:

> **Note:** A certain work item (record) can only match a delay condition once. When an item is processed, it can never be included in a new process instance based on the same definition.

Items to follow up (mainly un-handled tasks)

When distribution fails

When the defined distribution logic for a user step fails to identify a valid recipient, the task is taken out of the defined workflow process and marked as un-handled.

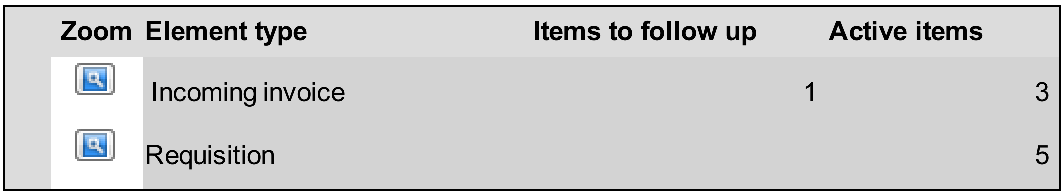
An un-handled task will only be made available for a defined administrator - or administrator group, with access to the **Items to follow up**window. The task - and process instance - will still be alive.

Two task types

When you first open Items to follow up, you get an overview - per element type - of both

- un-handled tasks, as well as
- active tasks awaiting processing by an identified task owner.

(un-handled)) (pending)

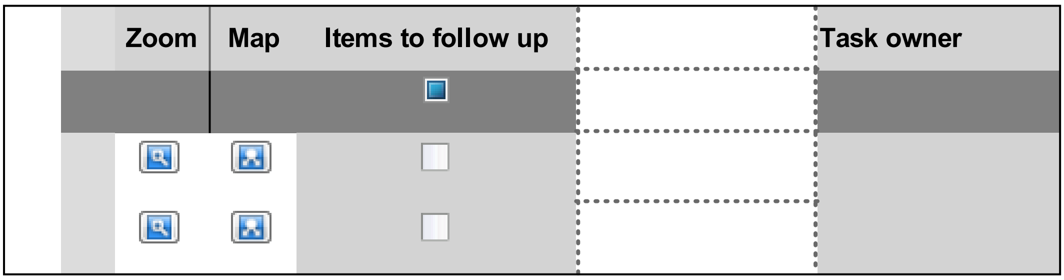


Zoom	Element type	Items to follow up	Active items
🔍	Incoming invoice	1	3
🔍	Requisition		5

When you select an element type to get more details (**Zoom**) , you get exactly the same options to handle the task, regardless of task type (un-handled or pending).

Items to follow up - the tab

You get access to the details about the un-finished tasks, as well as all options, on the **Items to follow up** tab.



Zoom	Map	Items to follow up		Task owner
		■		
🔍	⊞	□		
🔍	⊞	□		

Display columns: If you want to display element type specific data, you must use the *Item details* column in the Element type detailstab (in the**Element types**window). Selected element type columns will appear after*Task owner*.

Options

The following options are available for each un-finished task:

- Analyse the process instance: Take a look at the process instance details leading up to the current task.
- Forward the task to one or more new recipient(s).
- Reset process flow (Redistribute workflow). You can either resume the complete process (as if it was just instantiated, or just the un-finished step.
- Stop the process.

The various options are described below.

Analyse the process instance

The **Map** button gives you access to details (graphical map and various status information) about the process instance - so far. The map will also show the "ideal" way of process continuation, i.e. the workflow diagram if all subsequent user actions were positive.

You can click on any previous tasks in the map to get details about

- any changes made on the work item properties.
- any comments entered by previous task owners.

Forward the task

The **Zoom** button will open the **Manual distribution** tab, where you can add new recipients to the current, un-finished, task. The intention is simply to connect people who will process the task within a reasonable time frame.

Reset process (redistribute)

Note that the **Redistribute workflow** command only works when you first have selected one or more tasks - using the check-box column.

You can reset the process in three ways:

- Start the process again, from the beginning (the first task). A new process instance will be created, as if the original instantiation criteria appeared. The process map, however, will give you all the historic tasks, and where redistribution took place. All previously completed tasks must be processed again.

- Start the task (user step) instance again. This option will create a new instance of the un-finished task, and re-evaluate the distribution logic . Any previously completed sub-tasks must be processed again.
- Start according to a new version of the process definition. This alternative is only relevant if there actually is a newer version of the process definition. If so, however, the new definition will be instantiated, with the original work item.

Stop the process

Note that the two commands **Deactivate workflow** and **Delete workflow** only works when you first have selected one or more tasks - using the check-box column.

Deactivate workflow: when you deactivate workflow, the process instance will be killed, and only historical data will be available. Any further processing is stopped.

Delete workflow: When you delete a workflow, you remove all traces of the process instance. No historical records will be kept.

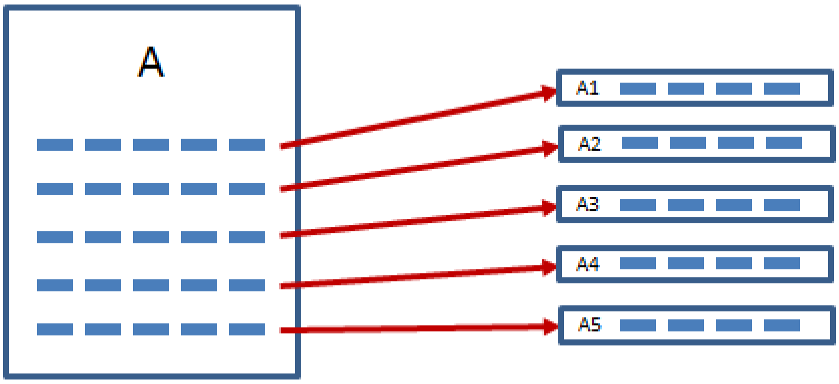
Grouping of tasks

Transactional element types and workflow

When a user works with transactional element types, he or she will normally handle several *related transactions* simultaneously: The user works with an *invoice* to a specific customer, but it is the various *invoice lines* that are the real transactions, and which will be the work items when the invoice is saved and sent on workflow.

A detailed transaction belongs (usually) to a greater entity, a header entity, as an invoice line belongs to an invoice, and all screens in Business World are designed in such a way that we handle the composite object as a whole.

Illustration: The user works with the whole object (A). Each line will be saved as a single transaction (A1 - A5) and instantiate a new workflow process instance.



There may be a large amount of simultaneous processes running, and a single workflow user may therefore be assigned a large amount of single tasks. Some form of grouping is required.

Predefined grouping per logical header (technical)

Business World uses entries in the *asysvalues* table to set up a logical grouping of tasks. All these entries have the name WF_GROUPING.
The grouping is based on table columns from the topmost table in the element type definition, columns often used as foreign keys to the header table.

This basic grouping setup defines:

- The outer limits for aggregation. Aggregation (of amounts) is only possible for work items belonging to the same header.
- How the system must count the number of tasks assigned to a recipient.

Example: All invoice types use *acrtrans* as the topmost table, and have a grouping based on *client* and *voucher_no*, where *voucher_no* identifies the header object (invoice).:

	Description1	Description2	...	Name
1	acrtrans	client, voucher_no		WF_GROUPING
..	...	...	...	...

Grouping tasks in the task list

Per default, Business WorldWorkflow will group all single tasks - as they will appear for a task recipient - according to the properties selected for*Task list*for the current element type in theElement type detailstab (in the**Element types**window), and present them as one line in the recipient's task list.

Example: For the *Incoming invoice* element type, *voucher_no* is selected as the sole property in *Task list*, and a recipient of tasks for incoming invoices, will therefore find one task (line) per invoice in the task list. Every task will be identified with the name given to the user step in the process definition, for example Invoice approval, as well as the value of *voucher_no* (TransNo when displayed).

You can modify the grouping by adding entries to the WF_TASKLIST system setup value.

Use WF_TASKLIST for task list configuration

WF_TASKLIST is a parameter (system setup value) which uses two elements for grouping:

- The Id of the screen used to handle (approve, modify etc) the work items. Some element types, for example the various invoice types, uses the same screens for similar tasks, and an entry in WF_TASKLIST may therefore affect several element types.
- The element type properties selected for *Task list*display in theElement type detailstab (in the**Element types**window).

When you identify a screen in a WF_TASKLIST entry, you will indirectly identify both the affected element types, as well as the relevant task types. The settings will then be valid for all tasks of a certain type, for all element types using the selected screen.

WF_TASKLIST example

Below, we show two entries for WF_TASKLIST:

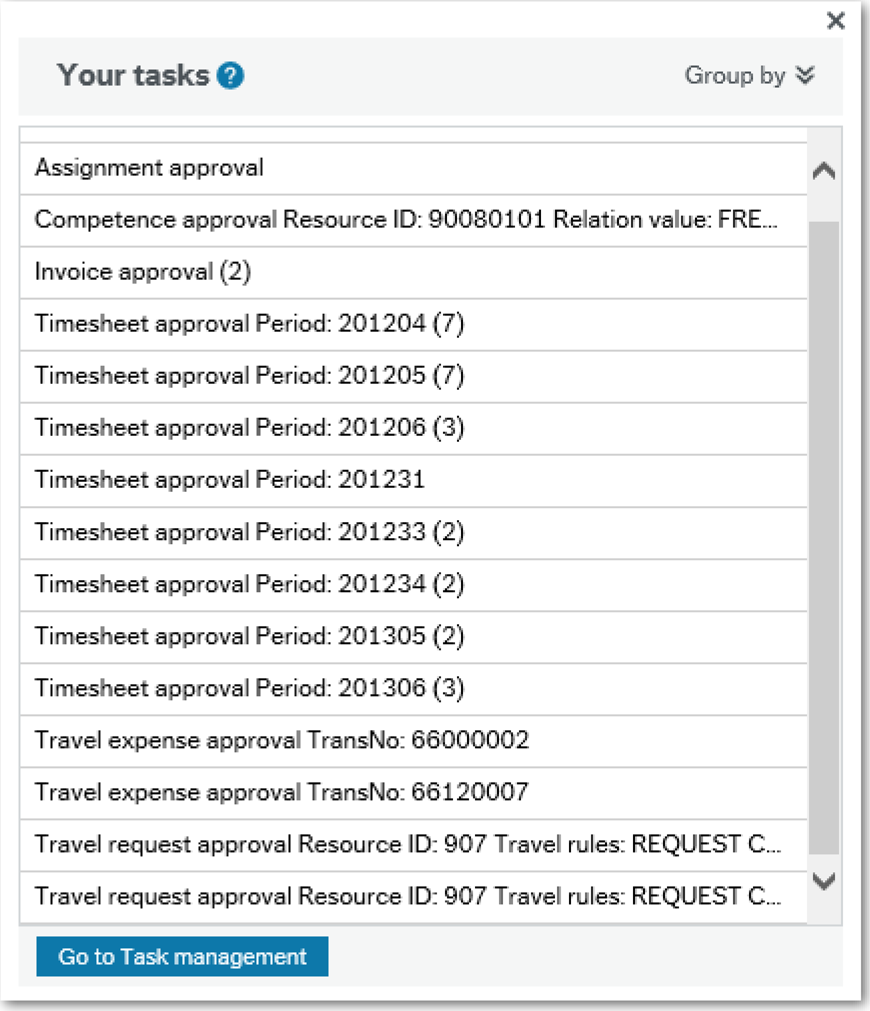
	Name	Pos	Text 1	Number 1	Text 2	...
?						
1	WF_TASKLIST	1	TTS080	0	reg_period	
2	WF_TASKLIST	2	TF1004	0		

Comments: *Text 1* identifies the screen (and indirectly element types and task type), while *Text 2* lists the columns that must be used as grouping criteria in the task list.

The first entry relates to the Timesheets element type and Timesheet approval, while the second entry relates to invoices (all element types for invoices) and invoice approval.

As we see, we have listed no columns in *Text 2* for TF1004, meaning that all tasks that will use this screen (Invoice approval) for task completion, will be grouped by task name only - regardless of *Task list* columns selected for the element type.

The task list may look like this:



Assignment approval

Competence approval Resource ID: 90080101 Relation value: FRE...

Invoice approval (2)

Timesheet approval Period: 201204 (7)

Timesheet approval Period: 201205 (7)

Timesheet approval Period: 201206 (3)

Timesheet approval Period: 201231

Timesheet approval Period: 201233 (2)

Timesheet approval Period: 201234 (2)

Timesheet approval Period: 201305 (2)

Timesheet approval Period: 201306 (3)

Travel expense approval TransNo: 66000002

Travel expense approval TransNo: 66120007

Travel request approval Resource ID: 907 Travel rules: REQUEST C...

Travel request approval Resource ID: 907 Travel rules: REQUEST C...

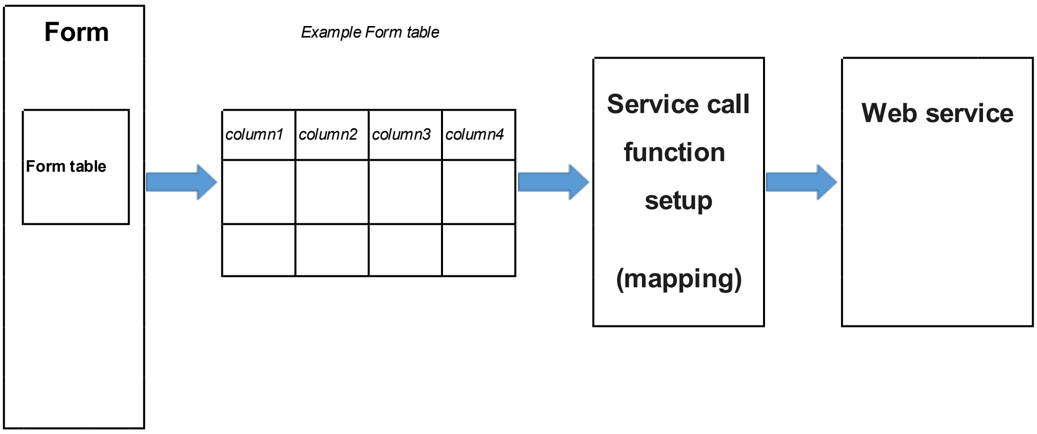
Service call function setup

Introduction

The **Service call function setup** window is designed to enhance Business World functionality with the use of internal SOAP web services and RESTful APIs directly from workflow. It enables users to define operations and parameters that a workflow system step will use when sending requests to web services and APIs. This way, service calls can be performed from different element types, e.g. a transaction of a specific type can be created based on data included in a form.

So far, if a simple form was created to enter invoices from a supplier, the only values a non-functional end user would know about, would be the values entered on the invoice. They would still miss information on accounting periods, etc. This created an additional requirement for another employee (in this case, an accountant) to enter the invoice manually in the **Registration of incoming invoices** window. This step is unnecessary, as the required data is already there, and the remaining part can be filled in while approving the invoice.

The **Service call function setup** window is designed to enable using such services and to create an entry point for data, while a form is being filled in. The functionality allows a consultant to map entry point's parameters (SOAP web service or RESTful web service) to the columns defined in the *Element type* tables. The general concept is as follows:



column1	column2	column3	column4

For the above solution to work, proper mapping definition between *Element type* tables and internal web service needs to be defined. The last step is adding a web service call to the workflow process for the specified element type. The *step-by-step* process is described below.

Defining parameters

Set common parameters

In order to be able to work with the **Service call function setup** window, you need to define two common parameters. These parameters allow for getting definitions of the corresponding web services and their operations. The parameters are:

- **WEBSERVICES_URL** - defines the location of the `Service.svc` file for SOAP Web Services.
- **REST_API_URL** - defines a link to OpenAPI Specification file for REST API Web Services.

To create the parameters:

1. In the menu, go to **System administration** > **System setup** > **System parameters**. Click **Common parameters**.
2. Click the **Add** button.
3. Enter information required for your parameter.
4. In the *Value* field, enter the following:
    - For SOAP services - the path to the `Service.svc` file. The file is used with Windows Communication Foundation (WCF) services hosted in Internet Information Services (IIS). It needs to be created in the application directory. Information on how to create the `.svc` file can be found on third-party websites.
    - For REST services - the link to the OpenAPI Specification JSON file.

Creating those two parameters enables you to select the *Type of service* you want to call during further setup in the **Service call function setup** window. Options are:

- **SOAP** - retrieves services defined by the WEBSERVICES_URL parameter.
- **REST** - retrieves services defined by the REST_API_URL parameter.

Setting up a service call

Working in the Service call function setup window

Once you defined the common parameters for your SOAP and REST API, you can start setting up the service call function. This way, you are able to configure the actual service call you need performed after the workflow enters appropriate system step. The functionality is based on the A29 SERVICE.CALL attribute.

1. In the menu, go to **Common** > **Workflow** > **Fixed registers**, and click **Service call function setup**.
2. Click **New** and start configuring your call. Enter the *Setup code* and description. In order to use the new service call in workflow, the *Status* needs to be set to **Active**.
3. Having defined the common parameters, you can now indicate the *Type of service*. Select **SOAP** or **REST** accordingly.
4. Depending on your choice, the list of available services is displayed once you press space. The services on the list reflect what has been defined in the `Service.svc` file (for SOAP services) or the OpenAPI Specification file (for REST services).
5. Click the **Get service description** button to access the list of operations available for the service you selected. The *Operation* drop-down becomes active. Select the operation for your call.

   > **Note:** For REST APIs, the type of operation is also indicated (POST, GET, PUT, PATCH, DELETE) along with a short description of what the operations actually do.

6. Now you need to decide, how you want your calls handled. Two options are available:
   - **Separate calls for each item in workflow group** - each workflow item triggers call of a web service immediately after it enters the system step.

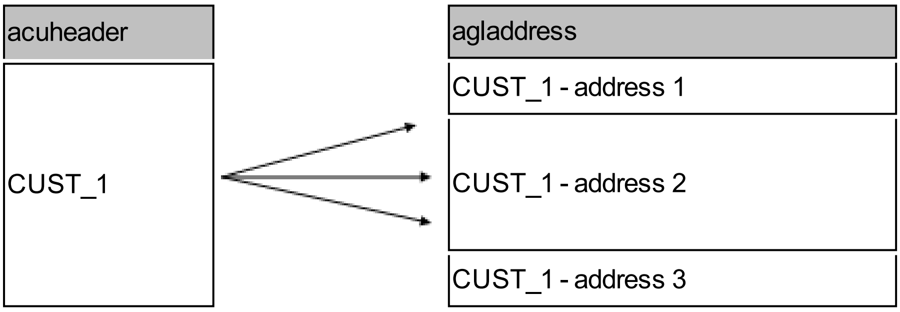
     Example 1:

     Let's assume an employee registers travel expenses for parking, food and hotel in the **Travel expenses** window. Once they save, the task enters the next step in workflow, which is manager's approval. The manager opens the request in the **Travel expenses approval** window and decides to enter **Advanced mode** to approve only hotel and parking expenses, putting the remaining food expense on hold. Each expense type is an item in the expenses workflow group according to workflow logic. This means that **Separate calls for each item in**

workflow group option triggers two separate calls - one for the hotel expense and the other for parking expense. These move forward in workflow according to the map, while for the food expense, another call will be triggered, once it gets approved.

Example 2:

A more complicated example, including the use of grouping functionality, could be as follows: we want to update a customer's existing invoice address or insert a new one. We will use the **Customer** element type in our service call. The element type has two related tables: *acuheader* and *agladdress*:



One *acuheader* table entry (CUST_1) can have multiple addresses assigned. To be able to call the **Addresses** web service using the **Separate calls for each item in workflow group** option, we would need to have the main table row multiplied. In our example, however, we want to update one of the addresses or insert an additional address of a particular, existing type. This would be impossible to do using this setup. We can, however, indicate that we want the *agladdress* table treated as the main row. To do this, we need to set the table name to *agladdress* for root parameter of a complex type:

Name	Cardinality	Data type	Table name (Grouping)
customerObjectList	0..1	Complex	*agladdress*
CustomerObject	0..*	Complex	
AddressList	0..1	Complex	
AddressUnitType	0..*	Complex	
AddressID	1..1	Long	

The system will then iterate by each subsidiary item of the root parameter and perform that many individual calls. This way, using PATCH and

DELETE operations (in case of RESTful APIs), we are able to achieve expected results.

- **Single call for all items in workflow group** - if there are items in the group that haven't passed through the system step, the system marks the processed item as *passed through system step* and proceeds to the next step. When all remaining items go through the system step, a single service call is triggered on behalf of all items in the workflow group.

  > **Note:** If a workflow item is deleted while in workflow (during approval or correction) and all other workflow items already passed through the system step, a single service call is triggered on behalf of all remaining items in the workflow group.

  Example:

  We have entered an invoice with two transaction lines:

agltrans
TRANSLINE_1
TRANSLINE_2

  After save, there will be as many executions of the GL process, as many lines are entered. Having the **Single call for all items in workflow group** option set up, we would like to call our service only once. If parameters for the service we are calling allow to enter more than one object of specified type, we are able to send all rows via this service at once. We need to specify, however, that we are grouping this exact complex parameter via the *agltrans* table, by putting the name of the table in the *Table name* field for the appropriate parameter. To find out more about the grouping functionality, see Using grouping functionality.

7. If authentication is required, enter a unique name of the service account in the *Service account* field. This only applies to REST services.

   > **Note:** To find out more about authentication, read the chapter on web service accounts in the **System administration** Reference manual.

8. Select the element types you want your service call triggered for.

Working with element types

Defining element type mappings

The next step is to define mapping for element types and their associated parameters. Element type is a transfer list that enables you to select on which workflow process types you can use your setup. New mapping of parameters needs to be defined for each selected element type.

In the **Service call function setup** window, you begin your work with element types by transferring them from the *Available* section to the *Selected* section. You then proceed to define the mappings on the **Element type mapping** tab. The tab becomes active only after you select at least one element type for your setup.

> **Note:** Transferring the element types back from *Selected* to *Available* removes all mappings from the setup. Caution is advised when doing that, especially if the setup is already used in a workflow process.

The main part of the tab is the *Parameters mapping* table. It contains columns allowing you to define mapping between service request object properties and selected element type table columns.

In order to understand the way mappings are defined, it is important to know what you can set in particular columns of the table:

- *Name* - names of parameters available for the selected element type. For REST API, there is a virtual name created to differentiate complex type **Array** that stores multiple objects of different types. As JSON arrays start and end with square brackets `[]`, these are added at the end of such name, e.g. `\patch[]`.
- *Path* - nesting level for the parameter. It reflects the structure, hierarchy and parent-child relations for it.
- *Cardinality* - the number of allowed occurrences for a given parameter. A **0..1** value means that the minimum number of occurrences equals zero, and maximum equals one.
    - **0..1** - optional parameter; minimum number of occurrences equals zero, and maximum equals one. The parameter can be removed.
    - **1..1** - mandatory parameter; both minimum and maximum number of occurrences is one, which means the parameter cannot be removed.

  - **0..*** - optional list.
  - **1..*** - mandatory list
  - **0..n**
  - **1..n**
  - **n..m**
- *Data type* - type of object representing the parameter. Possible values are: **Boolean**, **DateTime**, **Float**, **Int**, **Long**, **Money**, **String**. All other types defined in services are marked as **Complex**.
- *Table name* - column listing all tables belonging to the current element type. You can indicate which table the parameter value will be taken from. If you leave this field blank, you will be able to enter a static value for the parameter in the *Value* field.

> **Note:** For complex parameters, entering a table name in this column activates the grouping functionality for those parameters. For more information, see Using grouping functionality. Also note that in case of SOAP web services, the values in the *Table name* field for the root element need to be the same.

- *Value* - value to be assigned to a parameter. Only non-complex parameters can be assigned a value.
  - If a table is selected, this field contains all columns connected to the given element type for the selected table - the parameter gets its value from the selected column.
  - If there is no table selected, you can enter your parameter value manually.

> **Note:** Please note that if there are many rows in the selected table belonging to the element type processed, the system will iterate these rows and generate multiple instances of the parameter. This will happen as soon as cardinality rules are met. For more information see Using grouping functionality.

For complex REST API parameters, you can enter any JSON object in the *Value* area. To find out more, see Handling REST API PATCH operations.

- *Filter* - enabled only for the **Complex** parameter type. It allows for filtering operation data queried from the database while preparing a request in the workflow process. It basically means that you can limit rows being iterated for a given complex parameter by entering certain filtering criteria. The criteria must comply with the syntax of an SQL **WHERE** clause.

What can be included in the filter:

- Column names - if a column name contains any of these special characters: `~ ( ) # \ / = > < + - * % & | ^ ' " [ ]`, you must put the column name in square brackets `[ ]`. If a column name contains a right bracket `]` or a backslash `\`, escape it with a backslash (`\]` or `\\`).

> **Example:** Example: **[column#name] = 10**; **column_name = 10**

- Literals
  - String values - enclosed in single quotes `' '`. If the string contains a single quote character `'`, the quote must be doubled.
  - Numeric values - not enclosed in any characters.
  - Date values - enclosed in hash symbols `# #`.

> **Example:** Example:
> String: **column = 'Some value'**; **column = 'Some "extra" value'**
> Numeric: **column = 110**; **column = 115.9**
> Date: **date = #2017-01-01#**; **date = #31.12.2015 11:30:20#**

- Comparison operators
  - equal: =
  - not equal: <>
  - less than: <
  - less than or equal: <=
  - greater than: >
  - greater than or equal: >=
  - `IN` - used to include only the values from an indicated list of items. You can use this operator for all data types, such as numbers or strings.
  - `LIKE` - only values matching patterns with wildcards are included. A wildcard character is a `*` or `%` character. The pattern can start with it: `*value`, end with it: `value*`, or both: `*value*`. Placing wildcards within a pattern is not allowed.

Example: Example:
`IN`: **id IN (1,3); name IN ('Adam', 'Anna')**
`LIKE`: **name LIKE 'a*'; name LIKE '%an%'; name NOT LIKE 'a*'**

Note: Note: If a pattern in a `LIKE` clause contains these special characters: `* % [ ]`, they need to be escaped using brackets `[ ]`, like this: `[*]`, `[%]`, `[[]` or `[]]`.

- Boolean operators - `AND`, `OR` and `NOT` are used to concatenate expressions. `NOT` has precedence over the `AND` operator and `OR` operator.

Example: Example: **name = 'Adam' AND (age < 20 OR age > 60); NOT name = 'Adam' AND NOT IN (5,6)**

- Arithmetic and string operators
  - addition: `+`
  - subtraction: `–`
  - multiplication: `*`
  - division: `/`
  - modulus: `%`

Example: Example: **gross - net > 0**

- Aggregate functions
  - `SUM`
  - `COUNT`
  - `MIN`
  - `MAX`
  - `AVG` (average)
  - `STDEV` (statistical standard deviation)
  - `VAR` (statistical variance)

Example: Example: **amount > AVG(amount); sequence_no > COUNT (amount > 0)**

Using the grouping functionality

Indicating a table name for **Complex** type parameters activates the grouping functionality for all subsidiary members of the complex parameter. That means the system will iterate all rows in the indicated table belonging to an element type and generate parameter values for each subsidiary set of members.

In our example, let's assume the following:

- there are three lines in the *agladdress* table for the element type being processed;
- grouping functionality is on for the **Address** complex parameter (table name indicated in red);
- we have the following setup:

Name	Cardinality	Data type	Table name (Grouping)	Value (Column)
Customer	1..1	Complex		
Address	1..*	Complex	*agladdress*	
Telephone	0..*	String	*agladdress*	*telephone_1*

The case described above would result in the following API request:

```
<customer>
  <address>
    <telephone>22 73 57 67</telephone>
  </address>
  <address>
    <telephone>12 34 56 78</telephone>
  </address>
  <address>
    <telephone>98 76 54 32</telephone>
  </address>
</customer>
```

However, having the following setup, with grouping functionality off (no table name for any complex parameter):

Name	Cardinality	Data type	Table name (Grouping)	Value (Column)
Customer	1..1	Complex		

Name	Cardinality	Data type	Table name (Grouping)	Value (Column)
Address	1..*	Complex		
Telephone	0..*	String	*agladdress*	*telephone_1*

would result in the following API request:

```
<customer>
  <address>
    <telephone>22 73 57 67</telephone>
    <telephone>12 34 56 78</telephone>
    <telephone>98 76 54 32</telephone>
  </address>
</customer>
```

> **Note:** When using the grouping functionality, additional steps might be required to achieve expected results. In order to go from child element type table (set as the grouping table) to parent element type table (the one that contains data to be mapped), you need to define a table join between element type tables (particularly flexi field tables) using the **Data model** (JD06) window in Desktop. Otherwise, going only from parent element type table (the grouping one) to child element type table (the one that contains data to be mapped) is possible.

Applying filters

In our example, we deal with a SOAP API call. Let's assume the following:

- we are processing the **Customer** element type with three (or more) addresses of different types in the master file;
- we are willing to limit them only to **General** (*address_type*=*1*) and **Delivery** (*address_type*=2) addresses;
- *Grouping* functionality is on;
- we have the following setup:

Name	Cardinality	Data type	Table name (Grouping)	Value (Column)	Filter
UpdateCustomer	1..1	Complex			
CustomerUNITType	1..1	Complex			
CustomerID	0..1	String	*acuheader*	*apar_id*	
CustomerName	0..1	String	*acuheader*	*apar_name*	
CustomerGroupID	0..1	String		**1**	
TaxSystem	0..1	String		**EU**	
FixedTaxSystem	1..1	Boolean		**false**	
Address	1..*	Complex	*agladdress*		agladdress.address_type IN (1,2)
Street	1..1	String	*agladdress*	*address*	
Phone	1..1	String	*agladdress*	*telephone_1*	
City	1..1	String	*agladdress*	*place*	
Company	0..1	String	*acuheader*	*client*	

In the above table, you can see the filter we need to apply in the *SQL filter* column, marked in red.

The case described would result in an API request similar to the one below:

```
<UpdateCustomer>
  <customerUnitType>
```

```
    <CustomerID>1000</CustomerID>
    <CustomerName>Example Company</CustomerName>
    <CustomerGroupID>1</CustomerGroupID>
    <TaxSystem>EU</TaxSystem>
    <FixedTaxSystem>false</FixedTaxSystem>
      <Address>
        <Street>the General Address Street 1234</Street>
        <Phone>23 45 67 89</Phone>
        <City>London</City>
      </Address>
      <Address>
        <Street>the Delivery Address Street 3456</Street>
        <Phone>12 34 56 78</Phone>
        <City>Paris</City>
      </Address>
      <Address>
        <Street>the Delivery Address Street 9876</Street>
        <Phone>98 76 54 32</Phone>
        <City>New York</City>
      </Address>
    <Company>ABC</Company>
  </customerUnitType>
</UpdateCustomer>
```

We can also do the setup as follows (with additional instance of **Address** parameter and grouping on):

Name	Cardinality	Data type	Table name (Grouping)	Value (Column)	Filter
UpdateCustomer	1..1	Complex			
CustomerUNITType	1..1	Complex			

Name	Cardinality	Data type	Table name (Grouping)	Value (Column)	Filter
CustomerID	0..1	String	*acuheader*	*apar_id*	
CustomerName	0..1	String	*acuheader*	*apar_name*	
CustomerGroupID	0..1	String		**1**	
TaxSystem	0..1	String		**EU**	
FixedTaxSystem	1..1	Boolean		**false**	
Address	1..*	Complex	*agladdress*		agladdress.address_type='1'
Street	1..1	String	*agladdress*	*address*	
Phone	1..1	String	*agladdress*	*telephone_1*	
City	1..1	String	*agladdress*	*place*	
Address	1..*	Complex	*agladdress*		agladdress.address_type='2'
Street	1..1	String	*agladdress*	*address*	
Phone	1..1	String	*agladdress*	*telephone_1*	
City	1..1	String	*agladdress*	*place*	
Company	0..1	String	*acuheader*	*client*	

Assuming that we have the same data in the element type, the resulting request will be the same as above.

Inserting parameters

It is possible to insert additional parameters to the existing *Parameters mapping* table. They can be added using the **Insert parameters** dialogue available after you press the **Insert** button for a complex type parameter. The dialogue shows a list of items you can add to the map in context of currently selected row. Only parameters being immediate children of the currently marked one are displayed in the dialogue, and these could be both complex and simple.

1. Enter the name of the parameter you want to filter the results by.
2. Press the **Filter** button.
3. Select the parameters you want to insert from the list available and move them to the *Selected* area.
4. Click the **Insert** button to add the selected parameters to the active row.

Handling REST API PATCH operations

The **Insert parameters** dialogue works a bit differently in case of complex REST API parameters having no specification of child object. After selecting such parameter and clicking **Insert**, the dialogue box allows you to enter mapping for such object in the *Value* area.
The functionality utilizes the JSON PATCH operation. It allows you to update a JSON document without the need to resend the whole document, if only its part has changed. The main operations include:

- **Add** - for adding values to objects or inserting them into arrays.
- **Remove** - for removing values from an object or array.
- **Replace** - for replacing values.
- **Copy** - for copying values within a JSON document.
- **Move** - for moving values within a JSON document.
- **Test** - for testing whether the specified value is set in the document.

In the *Value* area, you can enter any JSON object. It might be a full object, if needed. You can also insert any data from any column within the selected *Table name* in the *Value* row. To do that, you must start with the ‘**@**’ char followed by a simple type, a colon and a column name.

There are 4 simple types supported:

- **s** - string type
- **d** - date type
- **i** - integer type
- **f** - float type

> **Example:** Example: In order to enter a value from the *apar_id* column of the *acuheader* table and treat it as a string, we need to enter **@s:apar_id**. To enter a value from the *credit_age* column of the *acuheader* table and treat it as an integer, we need to enter **@i:credit_age**. To enter a value from the *trans_date* column of the *acuheader* table and treat it as a date, we need to enter **@d:trans_date**. To enter a value from the *pay_delay* column of the *acuheader* table and treat it as a boolean value, we need to enter it as an integer where **0** is false and **1** is true, therefore **@i:pay_delay**.

The above items can be inserted inside any JSON object, on any level. While in workflow, they will be replaced with values from the indicated columns for selected rows. A complete JSON object can also be inserted, e.g.:

```
{"companyID": @s:client, "roleID": @s:role_fx}
```

or

```
{
    “description”: “Payment should be online.”,
    “amount”: 10.33,
    “relations”: {
        “relationToGroup”: “Internal group”
    }
}
```

Using service call in workflow

After you have set up the service call function, you may start defining the workflow process you want to use it in. Note, that for your service call setup to work as intended, you need to set its *Status* to **Active**.

> **Note:** By default, a new service call is set to **Parked**, as the process might be time-consuming and require revisiting. By keeping it **Parked**, you avoid entering an incomplete setup into workflow.

When your service call setup is ready, the next step is selecting the proper *Element type* in the **Process definition** window.

To set up a workflow process:

1. In the menu, go to **Common** > **Workflow**. Click **Process definition**.
2. Click the **New process** button or the **Add process** icon . The **Add process** dialogue window opens.
3. Enter the name for your process in the *Name* field.
4. Select the element type you set up for your service call from the *Available* list.
5. Click **Add**. A new process definition is created.

   > **Note:** If the element type is used in another process definition, you will receive a warning message. This is because using the same element type in several processes may cause multiple distributions of a single item.

6. Add a new user step . In the **Shape details** section, select the *Rule group* and *Screen* for your element type.
7. Select the action for your step on the **Actions** tab.

You are now ready to add the system step that actually triggers the web service call. This means that when the action defined in the user step completes, for example when the **Approve** action is performed, the system will automatically proceed to performing the service call according to setup indicated in the **Service call function setup** window.

1. In the **Process definition** window, add a system step to the new process you have recently created.
2. In the **Shape details** section, in the *Function* column, under *Usage*, select the service call setup you want triggered.

3. Click the **Validate** button to check if your diagram is correct.
4. Click the **Activate** button. The **Activate version** dialogue box is displayed. Enter the summary for your changes and click **OK**. Your process definition is ready.

Successful HTTP response OR-split function

Business World makes it possible to further control the workflow when response from the service call is not successful. For this purpose, you can use the OR-split **Successful HTTP response** function. This way, the workflow can be controlled based on HTTP code included in the response. As the function returns boolean values, these are interpreted as follows:

- successful **2xx** HTTP code is interpreted as **true**;
- HTTP code other than **2xx** is interpreted as **false**.

Successful HTTP response function is available for workflows with element types corresponding to element types defined in service call definitions. In other words, the function is not available, if an element type selected in workflow is not assigned to any service call.

> **Note:** The function has to be placed directly after the system step running the service call. This condition is a must, as OR-spilt only checks the previous node in a workflow map.

Once you place your OR-split on the workflow map:

1. Define the split criteria in the **Shape details** section. Type your description, then click **Add** under *Split criteria*.
2. The **Add new split criteria** dialogue opens. From the *Type* drop-down list, select **Function**.
3. From the *Function* drop-down, select **Successful HTTP response**.
4. From the *Operator* drop-down, select **like**.
5. In the *Value* field, type **true**. Click **OK**.
6. Click **Validate**, to validate your process.
7. Click the **Activate** button. The **Activate version** dialogue box is displayed. Enter the summary for your changes and click **OK**.

Service call function - example business scenarios

In this chapter, you will find step-by-step procedures for utilizing the service call function in workflows. This way, customers and consultants are able see how web services and RESTful APIs can facilitate real-life, day-to-day operations.

The business cases presented are all based on forms and show examples of both SOAP web services and REST APIs. The scenarios described include the following use cases:

- Supplier creation;
- User data management;
- Full transaction reversal in Accounts Receivable;
- Partial transaction reversal in General Ledger.

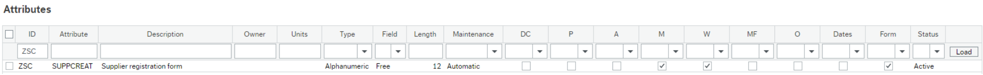
> **Note:** Images in this chapter reflect English user interface, and are available in English only.

Scenario 1: Supplier creation form

In this example scenario, you will register a supplier using a simple form. To achieve this, apart from creating the form, you will also set up the service call function and define the workflow. This way, you will be able to register your supplier automatically upon form submission.

Creating the form

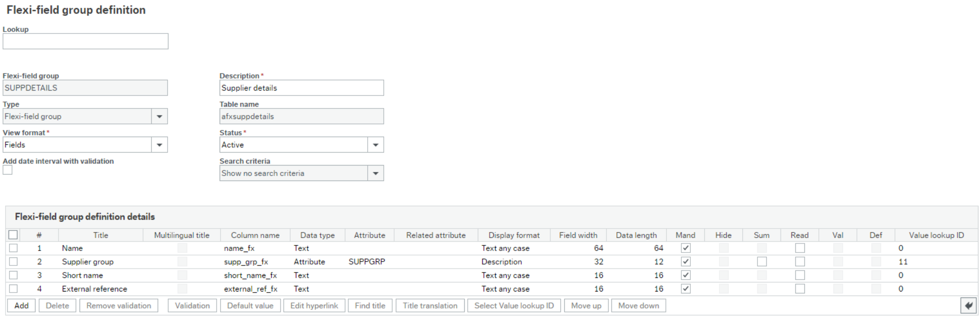
1. In Business World, go to the **Attributes** (TGL004) window. Click **Add** and create attribute **ZSC** SUPPCREAT (Supplier registration form).



Attributes

□	ID	Attribute	Description	Owner	Units	Type	Field	Length	Maintenance	DC	P	A	M	W	MF	O	Dates	Form	Status	
	ZSC																			Load
□	ZSC	SUPPCREAT	Supplier registration form			Alphanumeric	Free	12	Automatic	□	□	□	☑	☑	□	□	□	☑	Active	

2. In the **Flexi-field group definition** (TAG060) window, create a flexi-field group called SUPPDETAILS (Supplier details):



Flexi-field group definition

Lookup

Flexi-field group: SUPPDETAILS

Description *: Supplier details

Type: Flexi-field group

Table name: afxsuppdetails

View format *: Fields

Status *: Active

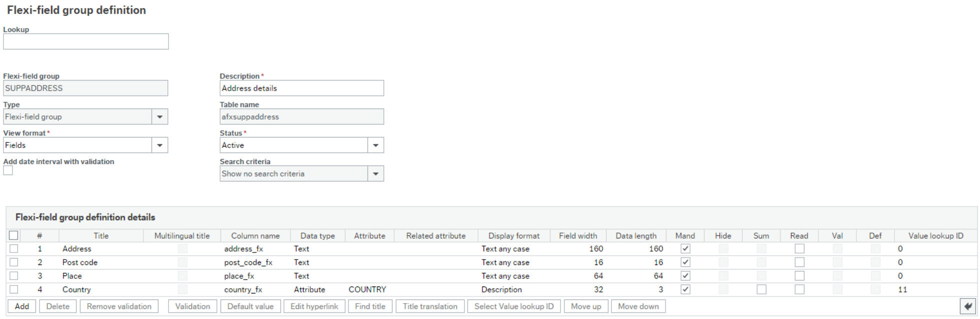
Add date interval with validation: □

Search criteria: Show no search criteria

Flexi-field group definition details

□	#	Title	Multilingual title	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup ID
□	1	Name		name_fx	Text			Text any case	64	64	☑			□			0
□	2	Supplier group		supp_grp_fx	Attribute	SUPPGRP		Description	32	12	☑		□	□			11
□	3	Short name		short_name_fx	Text			Text any case	16	16	☑			□			0
□	4	External reference		external_ref_fx	Text			Text any case	16	16	☑			□			0

3. In the **Flexi-field group definition** (TAG060) window, create a flexi-field group called SUPPADDRESS (Address details):



Flexi-field group definition

Lookup

Flexi-field group: SUPPADDRESS

Description *: Address details

Type: Flexi-field group

Table name: afxsuppaddress

View format *: Fields

Status *: Active

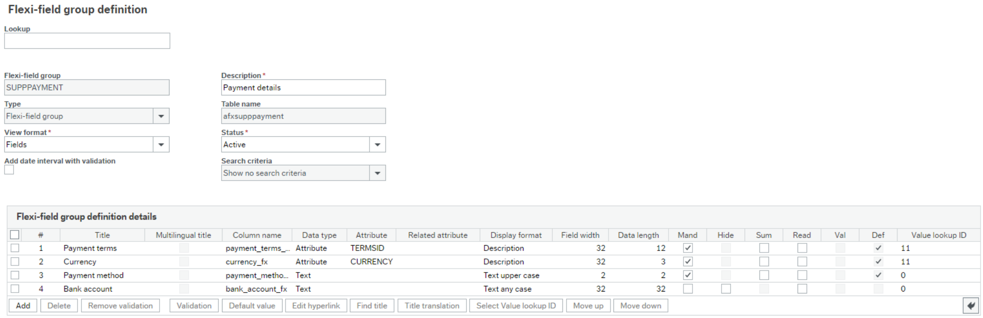
Add date interval with validation: □

Search criteria: Show no search criteria

Flexi-field group definition details

□	#	Title	Multilingual title	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup ID
□	1	Address		address_fx	Text			Text any case	160	160	☑			□			0
□	2	Post code		post_code_fx	Text			Text any case	16	16	☑			□			0
□	3	Place		place_fx	Text			Text any case	64	64	☑			□			0
□	4	Country		country_fx	Attribute	COUNTRY		Description	32	3	☑		□	□			11

4. In the **Flexi-field group definition** (TAG060) window, create a flexi-field group called SUPPPAYMENT (Payment details):



Flexi-field group definition

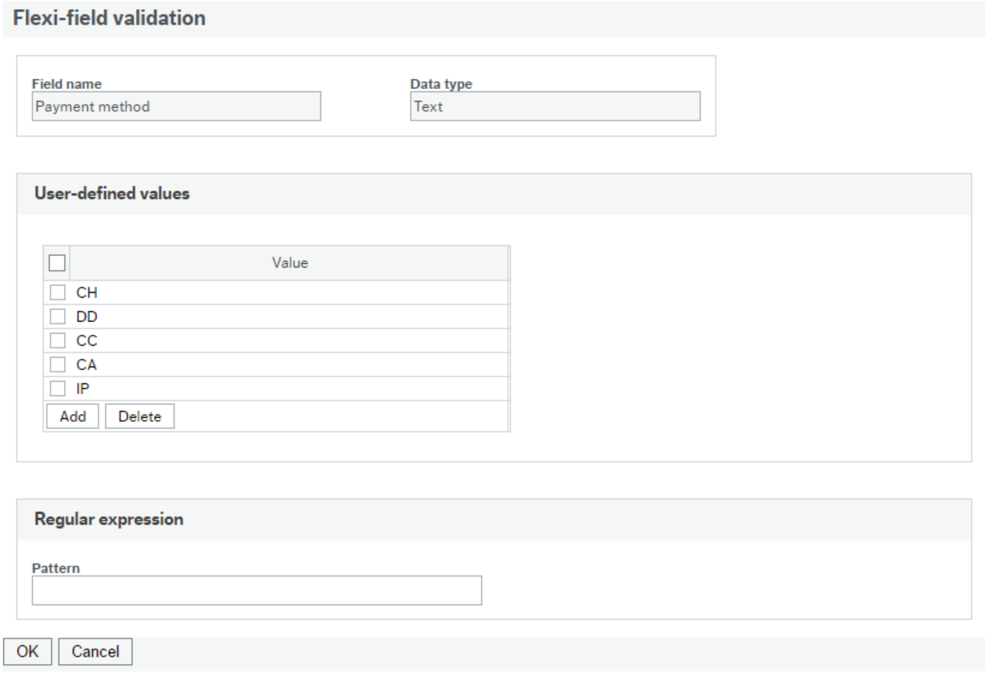
Lookup	
Flexi-field group: SUPPPAYMENT	Description *: Payment details
Type: Flexi-field group	Table name: afxsupppayment
View format *: Fields	Status *: Active
Add date interval with validation: □	Search criteria: Show no search criteria

Flexi-field group definition details

□	#	Title	Multilingual title	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup ID
□	1	Payment terms		payment_terms_...	Attribute	TERMSID		Description	32	12	☑		□	□		✓	11
□	2	Currency		currency_fx	Attribute	CURRENCY		Description	32	3	☑		□	□		✓	11
□	3	Payment method		payment_metho...	Text			Text upper case	2	2	☑		□	□		✓	0
□	4	Bank account		bank_account_fx	Text			Text any case	32	32	□	□		□			0

Assign default values for **Payment terms**, **Currency** and **Payment method** using the **Default value** button.

For **Payment method**, define a list of available values using the **Validation** button:



Flexi-field validation

Field name: Payment method

Data type: Text

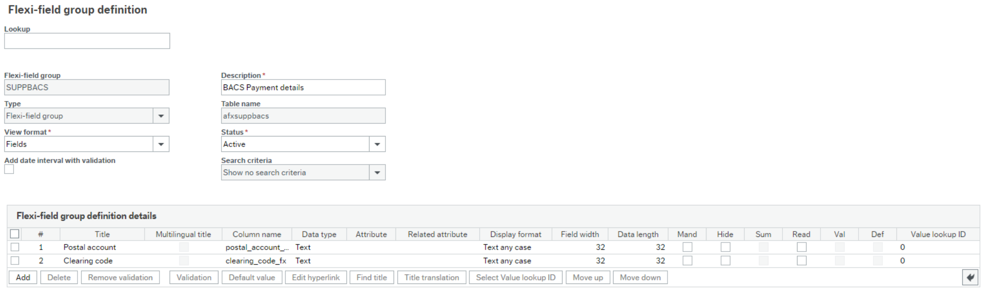
User-defined values

□	Value
□	CH
□	DD
□	CC
□	CA
□	IP

Regular expression

Pattern:

5. In the **Flexi-field group definition** (TAG060) window, create a flexi-field group called SUPPBACS (BACS Payment details):



Flexi-field group definition

Lookup

Flexi-field group: SUPPBACS

Type: Flexi-field group

View format*: Fields

Add date interval with validation: ☐

Description*: BACS Payment details

Table name: afxsuppbacs

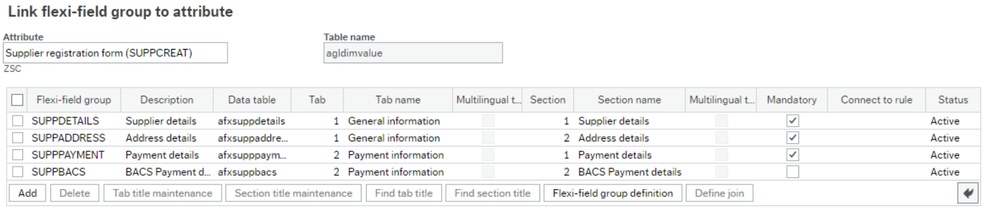
Status*: Active

Search criteria: Show no search criteria

Flexi-field group definition details

☐	#	Title	Multilingual title	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup ID
☐	1	Postal account		postal_account_...	Text			Text any case	32	32	☐	☐		☐			0
☐	2	Clearing code		clearing_code_fx	Text			Text any case	32	32	☐	☐		☐			0

6. In the **Link flexi-field group to attribute** (TCR037) window, link all created flexi-field groups to the **ZSC** SUPPCREAT attribute:



Link flexi-field group to attribute

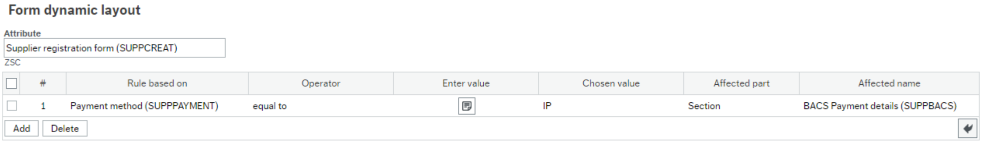
Attribute: Supplier registration form (SUPPCREAT)
ZSC

Table name: agldimvalue

☐	Flexi-field group	Description	Data table	Tab	Tab name	Multilingual t...	Section	Section name	Multilingual t...	Mandatory	Connect to rule	Status
☐	SUPPDETAILS	Supplier details	afxsuppdetails	1	General information		1	Supplier details		☑		Active
☐	SUPPADDRESS	Address details	afxsuppaddre...	1	General information		2	Address details		☑		Active
☐	SUPPPAYMENT	Payment details	afxsupppaym...	2	Payment information		1	Payment details		☑		Active
☐	SUPPBACS	BACS Payment d...	afxsuppbacs	2	Payment information		2	BACS Payment details		☐		Active

Note: Detailed information on working with flexi-field groups can be found in Business World 7.0 **Flexi-fields** Reference manual.

7. In the **Form dynamic layout** (TCR039) window, define the dynamic layout rule as below:

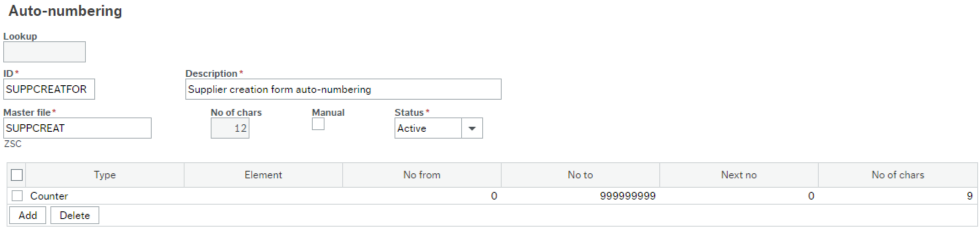


Form dynamic layout

Attribute: Supplier registration form (SUPPCREAT)
ZSC

☐	#	Rule based on	Operator	Enter value	Chosen value	Affected part	Affected name
☐	1	Payment method (SUPPPAYMENT)	equal to		IP	Section	BACS Payment details (SUPPBACS)

8. Define the auto-numbering rule for the **ZSC** SUPPCREAT attribute in the **Auto-numbering** (TAG061) window:



Auto-numbering

Lookup

ID*: SUPPCREATFOR

Description*: Supplier creation form auto-numbering

Master file*: SUPPCREAT
ZSC

No of chars: 12

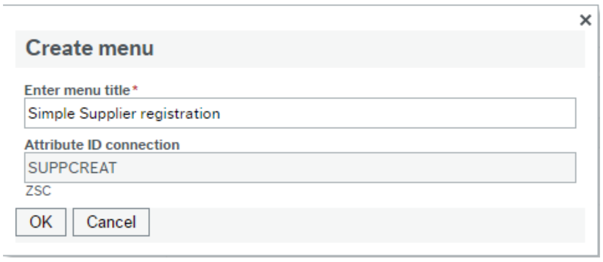
Manual: ☐

Status*: Active

☐	Type	Element	No from	No to	Next no	No of chars
☐	Counter		0	999999999	0	9

9. In the **Form administration** (TCR022) window, select the **ZSC** SUPPCREAT attribute in the *Form* field. Click the **Create menu** button to create a menu item dedicated to

this form:



Create menu

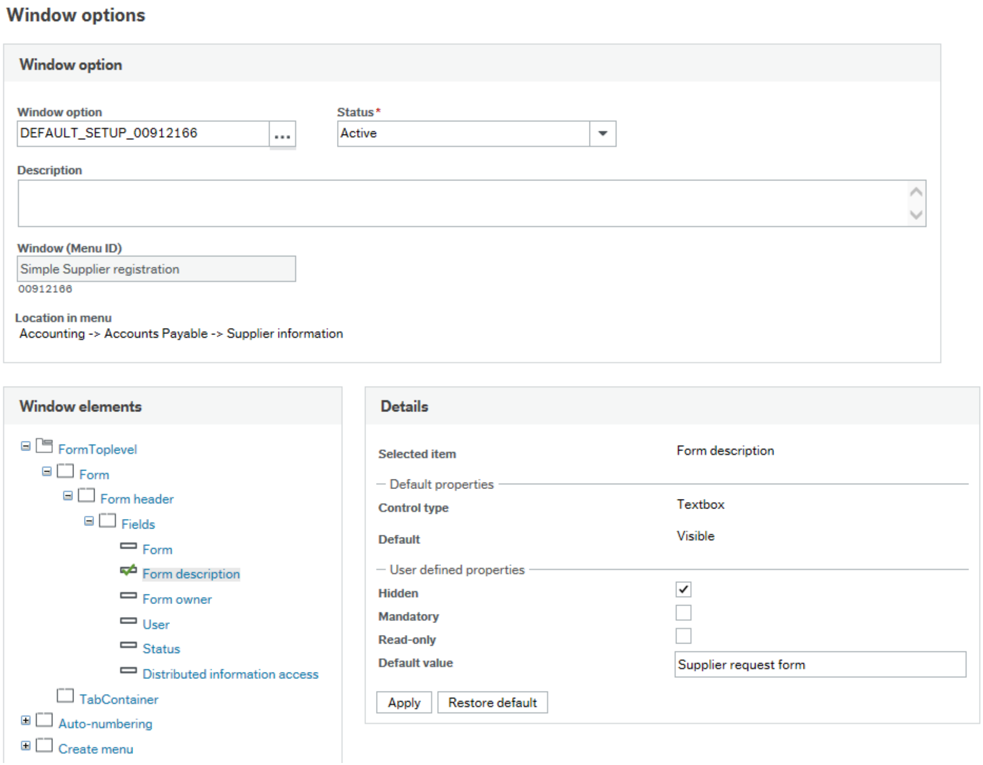
Enter menu title*
Simple Supplier registration

Attribute ID connection
SUPPCREAT
ZSC

10. In **Window Options** (TAG100), hide the *Form description* field for the newly created menu item, as the users do not need to fill it in. Click the item under **Window elements**, and select the *Hidden* check box.

> **Note:** You can also hide the entire *Form header* section if need be.

Assign a *Default* value to the item:



Window options

Window option

Window option: DEFAULT_SETUP_00912166
Status*: Active
Description:
Window (Menu ID): Simple Supplier registration
00912166
Location in menu: Accounting -> Accounts Payable -> Supplier information

Window elements

Details

Selected item: Form description

Default properties
Control type: Textbox
Default: Visible

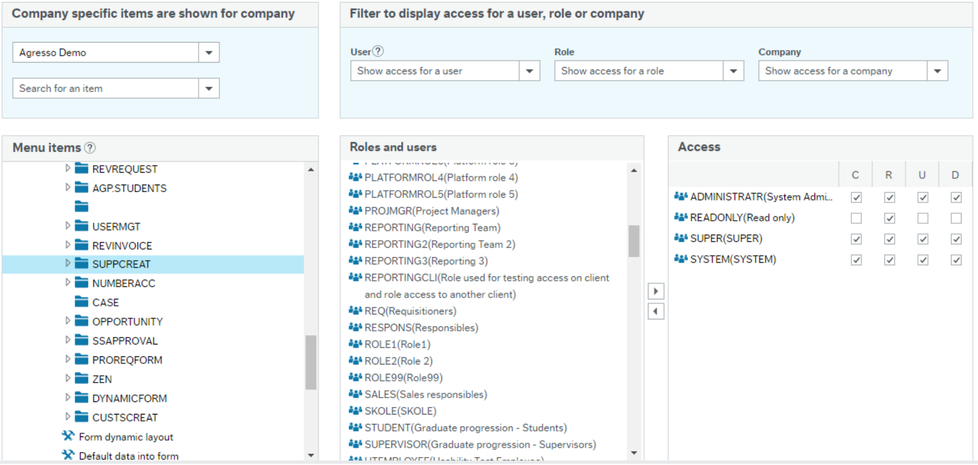
User defined properties
Hidden: ☑
Mandatory: □
Read-only: □
Default value: Supplier request form

Distribute the settings to the companies/users required by using the **Window options distribution** functionality.

> **Note:** Detailed information on working with forms can be found in Business World 7.0 **Forms** Reference manual.

11. In the **Menu access** (XAG003) window, grant access to the following:

    - **Common** > **Forms** > **Form administration** > **SUPPCREAT** and all subsections. This access is optional.
    - **Common** > **Forms** > **Form approval** > **SUPPCREAT** and all subsections
    - **Common** > **Forms** > **Simple Supplier registration** and all subsections:



Company specific items are shown for company

Agresso Demo

Search for an item

Filter to display access for a user, role or company

User

Show access for a user

Role

Show access for a role

Company

Show access for a company

Menu items

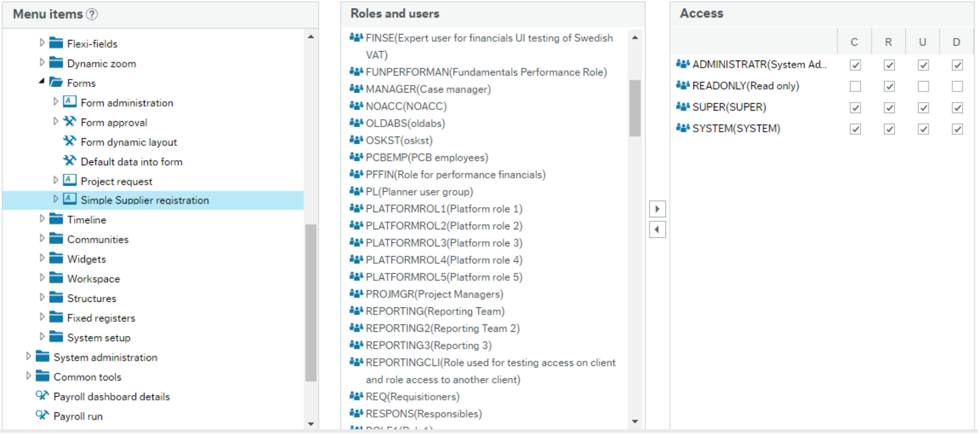
- REVREQUEST
- AGP.STUDENTS
- USERMGT
- REVINVOICE
- SUPPCREAT
- NUMBERACC
- CASE
- OPPORTUNITY
- SSAPPROVAL
- PROREQFORM
- ZEN
- DYNAMICFORM
- CUSTSCREAT
- Form dynamic layout
- Default data into form

Roles and users

- PLATFORMROL4(Platform role 4)
- PLATFORMROL5(Platform role 5)
- PROJMGR(Project Managers)
- REPORTING(Reporting Team)
- REPORTING2(Reporting Team 2)
- REPORTING3(Reporting 3)
- REPORTINGCLI(Role used for testing access on client and role access to another client)
- REQ(Requisitioners)
- RESPONS(Responsibles)
- ROLE1(Role1)
- ROLE2(Role 2)
- ROLE99(Role99)
- SALES(Sales responsibles)
- SKOLE(SKOLE)
- STUDENT(Graduate progression - Students)
- SUPERVISOR(Graduate progression - Supervisors)

Access

	C	R	U	D
ADMINISTRATR(System Admi...	☑	☑	☑	☑
READONLY(Read only)	□	☑	□	□
SUPER(SUPER)	☑	☑	☑	☑
SYSTEM(SYSTEM)	☑	☑	☑	☑



Menu items

- Flexi-fields
- Dynamic zoom
- Forms
  - Form administration
  - Form approval
  - Form dynamic layout
  - Default data into form
  - Project request
  - Simple Supplier registration
- Timeline
- Communities
- Widgets
- Workspace
- Structures
- Fixed registers
- System setup
- System administration
- Common tools
- Payroll dashboard details
- Payroll run

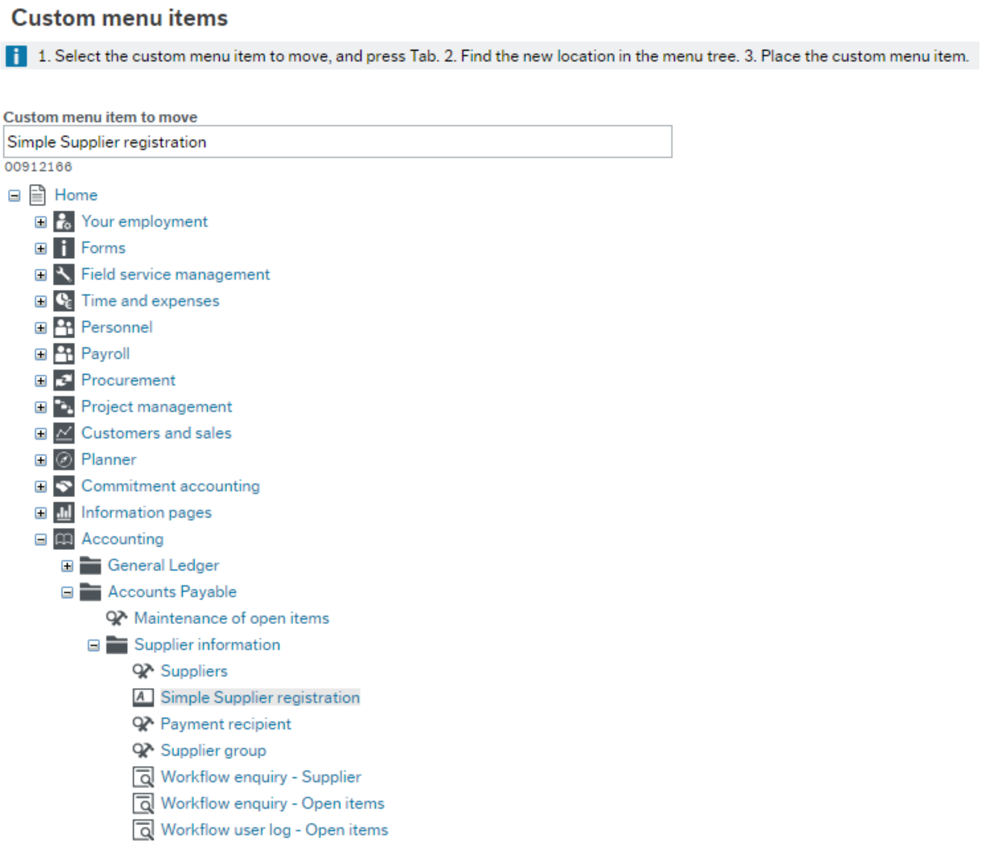
Roles and users

- FINSE(Expert user for financials UI testing of Swedish VAT)
- FUNPERFORMAN(Fundamentals Performance Role)
- MANAGER(Case manager)
- NOACC(NOACC)
- OLDABS(oldabs)
- OSKST(oskst)
- PCBEMP(PCB employees)
- PFFIN(Role for performance financials)
- PL(Planner user group)
- PLATFORMROL1(Platform role 1)
- PLATFORMROL2(Platform role 2)
- PLATFORMROL3(Platform role 3)
- PLATFORMROL4(Platform role 4)
- PLATFORMROL5(Platform role 5)
- PROJMGR(Project Managers)
- REPORTING(Reporting Team)
- REPORTING2(Reporting Team 2)
- REPORTING3(Reporting 3)
- REPORTINGCLI(Role used for testing access on client and role access to another client)
- REQ(Requisitioners)
- RESPONS(Responsibles)

Access

	C	R	U	D
ADMINISTRATR(System Ad...	☑	☑	☑	☑
READONLY(Read only)	□	☑	□	□
SUPER(SUPER)	☑	☑	☑	☑
SYSTEM(SYSTEM)	☑	☑	☑	☑

12. In the **Custom menu items** (TCR028) window, place the **Simple Supplier registration** menu item under **Accounting** > **Accounts Payable** > **Supplier information** (right click over the target item to decide where to place it):



Custom menu items

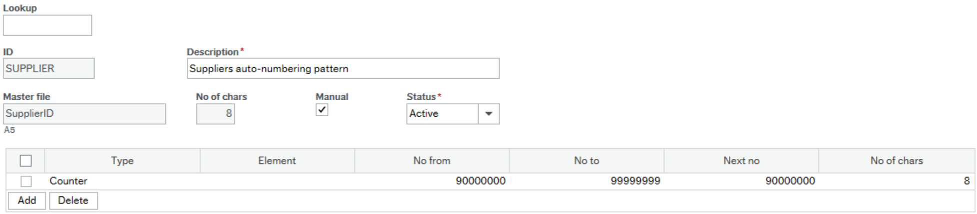
1. Select the custom menu item to move, and press Tab. 2. Find the new location in the menu tree. 3. Place the custom menu item.

Custom menu item to move: Simple Supplier registration

00912166

- Home
  - Your employment
  - Forms
  - Field service management
  - Time and expenses
  - Personnel
  - Payroll
  - Procurement
  - Project management
  - Customers and sales
  - Planner
  - Commitment accounting
  - Information pages
  - Accounting
    - General Ledger
    - Accounts Payable
      - Maintenance of open items
      - Supplier information
        - Suppliers
        - Simple Supplier registration
        - Payment recipient
        - Supplier group
        - Workflow enquiry - Supplier
        - Workflow enquiry - Open items
        - Workflow user log - Open items

13. In the **Auto-numbering** window, enable auto-numbering for the **Supplier** master file:



Lookup:

ID: SUPPLIER

Description*: Suppliers auto-numbering pattern

Master file: SupplierID

A5

No of chars: 8

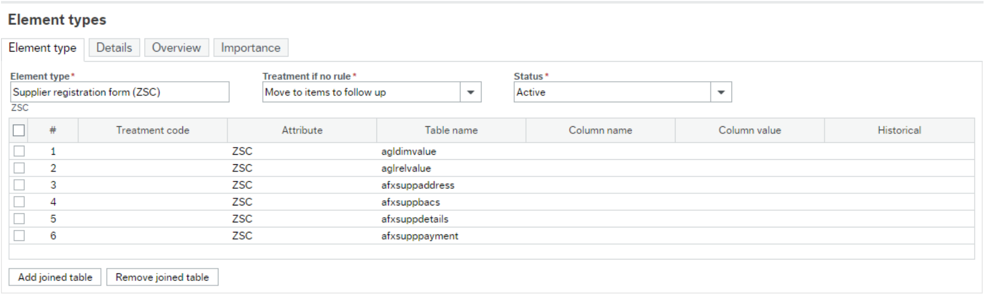
Manual: ☑

Status*: Active

□	Type	Element	No from	No to	Next no	No of chars
□	Counter		90000000	99999999	90000000	8

Setting up the service call

1. In the **Element types** (TWF032) window, create the **Supplier registration form** element type – query for it in the lookup field and just **Save**:



Element types

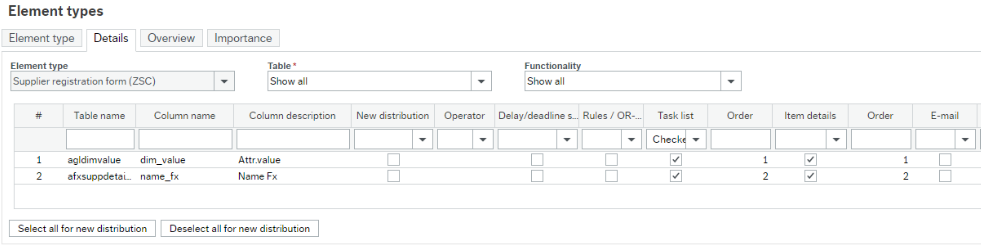
Element type*: Supplier registration form (ZSC)
ZSC

Treatment if no rule*: Move to items to follow up

Status*: Active

□	#	Treatment code	Attribute	Table name	Column name	Column value	Historical
□	1		ZSC	agldimvalue			
□	2		ZSC	aglrelvalue			
□	3		ZSC	afxsuppaddress			
□	4		ZSC	afxsuppbacs			
□	5		ZSC	afxsuppdetails			
□	6		ZSC	afxsupppayment			

On the **Details** tab, select element type properties to be visible in the task list and under item details:



Element types

Element type
Details
Overview
Importance

Element type: Supplier registration form (ZSC)

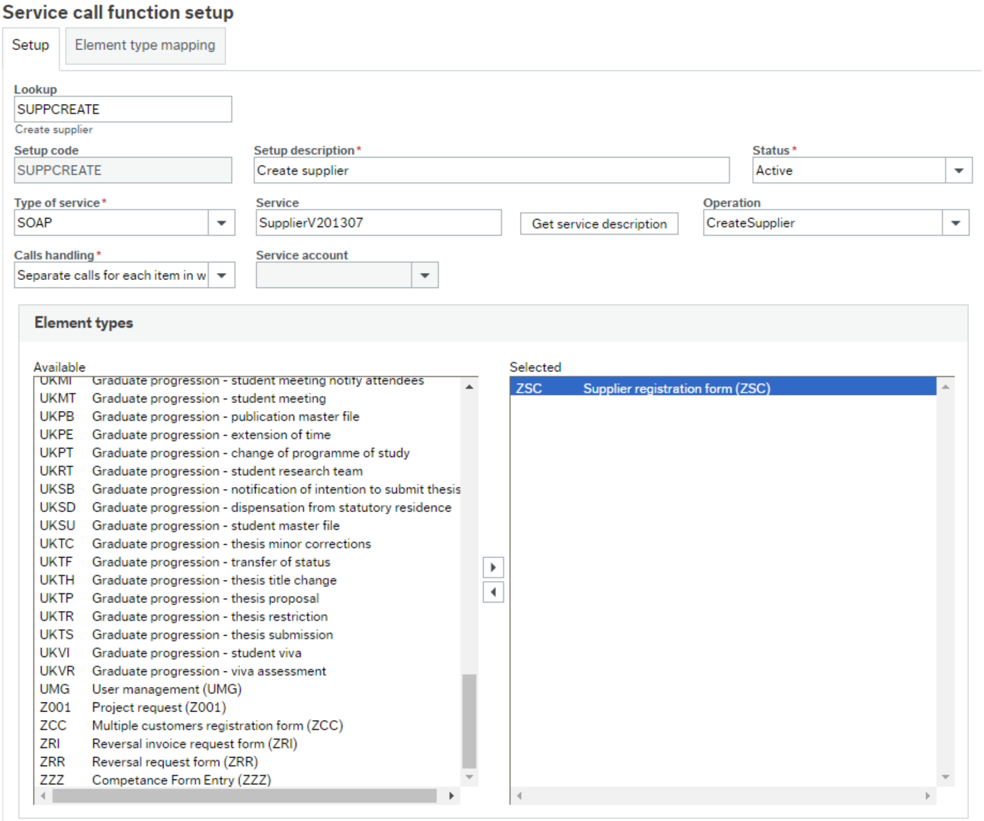
Table*: Show all

Functionality: Show all

#	Table name	Column name	Column description	New distribution	Operator	Delay/deadline s...	Rules / OR-...	Task list	Order	Item details	Order	E-mail
								Checke				
1	agldimvalue	dim_value	Attr.value	□		□	□	☑	1	☑	1	□
2	afxsuppdetai...	name_fx	Name Fx	□		□	□	☑	2	☑	2	□

Select all for new distribution
Deselect all for new distribution

2. In the **Service call function setup** window (TWF260) create service call setup for the **Supplier registration form**:



Service call function setup

Setup | Element type mapping

Lookup: SUPPCREATE
Create supplier

Setup code: SUPPCREATE

Setup description*: Create supplier

Status*: Active

Type of service*: SOAP

Service: SupplierV201307

Get service description

Operation: CreateSupplier

Calls handling*: Separate calls for each item in w

Service account:

Element types

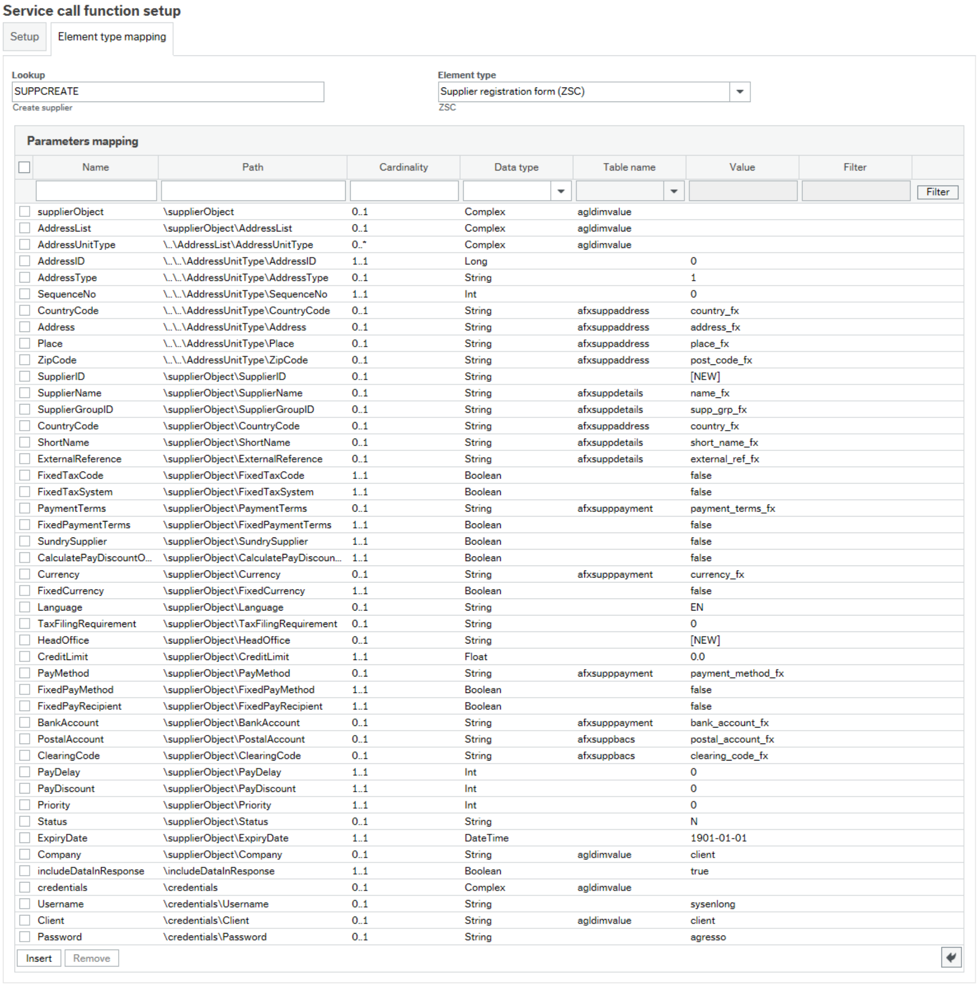
Available

UKMI	Graduate progression - student meeting notify attendees
UKMT	Graduate progression - student meeting
UKPB	Graduate progression - publication master file
UKPE	Graduate progression - extension of time
UKPT	Graduate progression - change of programme of study
UKRT	Graduate progression - student research team
UKSB	Graduate progression - notification of intention to submit thesis
UKSD	Graduate progression - dispensation from statutory residence
UKSU	Graduate progression - student master file
UKTC	Graduate progression - thesis minor corrections
UKTF	Graduate progression - transfer of status
UKTH	Graduate progression - thesis title change
UKTP	Graduate progression - thesis proposal
UKTR	Graduate progression - thesis restriction
UKTS	Graduate progression - thesis submission
UKVI	Graduate progression - student viva
UKVR	Graduate progression - viva assessment
UMG	User management (UMG)
Z001	Project request (Z001)
ZCC	Multiple customers registration form (ZCC)
ZRI	Reversal invoice request form (ZRI)
ZRR	Reversal request form (ZRR)
ZZZ	Competance Form Entry (ZZZ)

Selected

ZSC	Supplier registration form (ZSC)

Map element type properties to parameters:



Service call function setup

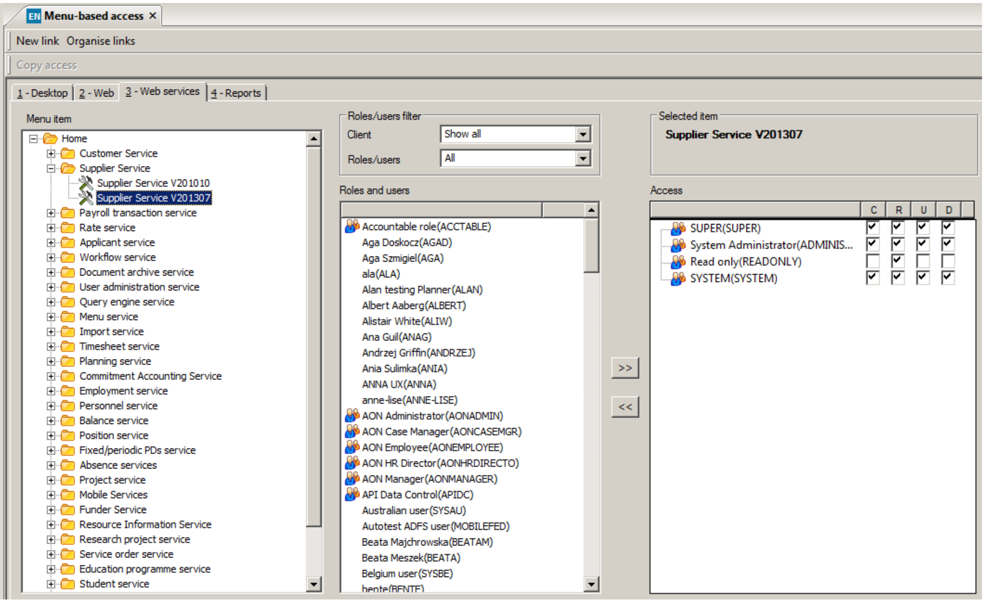
Lookup: SUPPCREATE
Create supplier

Element type: Supplier registration form (ZSC)
ZSC

Parameters mapping

☐	Name	Path	Cardinality	Data type	Table name	Value	Filter
☐	supplierObject	\supplierObject	0..1	Complex	agldimvalue		
☐	AddressList	\supplierObject\AddressList	0..1	Complex	agldimvalue		
☐	AddressUnitType	\..\AddressList\AddressUnitType	0..*	Complex	agldimvalue		
☐	AddressID	\..\..\AddressUnitType\AddressID	1..1	Long		0	
☐	AddressType	\..\..\AddressUnitType\AddressType	0..1	String		1	
☐	SequenceNo	\..\..\AddressUnitType\SequenceNo	1..1	Int		0	
☐	CountryCode	\..\..\AddressUnitType\CountryCode	0..1	String	afxsuppaddress	country_fx	
☐	Address	\..\..\AddressUnitType\Address	0..1	String	afxsuppaddress	address_fx	
☐	Place	\..\..\AddressUnitType\Place	0..1	String	afxsuppaddress	place_fx	
☐	ZipCode	\..\..\AddressUnitType\ZipCode	0..1	String	afxsuppaddress	post_code_fx	
☐	SupplierID	\supplierObject\SupplierID	0..1	String		[NEW]	
☐	SupplierName	\supplierObject\SupplierName	0..1	String	afxsuppdetails	name_fx	
☐	SupplierGroupID	\supplierObject\SupplierGroupID	0..1	String	afxsuppdetails	supp_grp_fx	
☐	CountryCode	\supplierObject\CountryCode	0..1	String	afxsuppaddress	country_fx	
☐	ShortName	\supplierObject\ShortName	0..1	String	afxsuppdetails	short_name_fx	
☐	ExternalReference	\supplierObject\ExternalReference	0..1	String	afxsuppdetails	external_ref_fx	
☐	FixedTaxCode	\supplierObject\FixedTaxCode	1..1	Boolean		false	
☐	FixedTaxSystem	\supplierObject\FixedTaxSystem	1..1	Boolean		false	
☐	PaymentTerms	\supplierObject\PaymentTerms	0..1	String	afxsupppayment	payment_terms_fx	
☐	FixedPaymentTerms	\supplierObject\FixedPaymentTerms	1..1	Boolean		false	
☐	SundrySupplier	\supplierObject\SundrySupplier	1..1	Boolean		false	
☐	CalculatePayDiscountO...	\supplierObject\CalculatePayDiscoun...	1..1	Boolean		false	
☐	Currency	\supplierObject\Currency	0..1	String	afxsupppayment	currency_fx	
☐	FixedCurrency	\supplierObject\FixedCurrency	1..1	Boolean		false	
☐	Language	\supplierObject\Language	0..1	String		EN	
☐	TaxFilingRequirement	\supplierObject\TaxFilingRequirement	0..1	String		0	
☐	HeadOffice	\supplierObject\HeadOffice	0..1	String		[NEW]	
☐	CreditLimit	\supplierObject\CreditLimit	1..1	Float		0.0	
☐	PayMethod	\supplierObject\PayMethod	0..1	String	afxsupppayment	payment_method_fx	
☐	FixedPayMethod	\supplierObject\FixedPayMethod	1..1	Boolean		false	
☐	FixedPayRecipient	\supplierObject\FixedPayRecipient	1..1	Boolean		false	
☐	BankAccount	\supplierObject\BankAccount	0..1	String	afxsupppayment	bank_account_fx	
☐	PostalAccount	\supplierObject\PostalAccount	0..1	String	afxsuppbacs	postal_account_fx	
☐	ClearingCode	\supplierObject\ClearingCode	0..1	String	afxsuppbacs	clearing_code_fx	
☐	PayDelay	\supplierObject\PayDelay	1..1	Int		0	
☐	PayDiscount	\supplierObject\PayDiscount	1..1	Int		0	
☐	Priority	\supplierObject\Priority	1..1	Int		0	
☐	Status	\supplierObject\Status	0..1	String		N	
☐	ExpiryDate	\supplierObject\ExpiryDate	1..1	DateTime		1901-01-01	
☐	Company	\supplierObject\Company	0..1	String	agldimvalue	client	
☐	includeDataInResponse	\includeDataInResponse	1..1	Boolean		true	
☐	credentials	\credentials	0..1	Complex	agldimvalue		
☐	Username	\credentials\Username	0..1	String		sysenlong	
☐	Client	\credentials\Client	0..1	String	agldimvalue	client	
☐	Password	\credentials\Password	0..1	String		agresso	

3. In the Desktop client, grant access to the **Supplier** web service. Go to the **Menu-based access** (AG68) window:



Menu item

- Home
  - Customer Service
  - Supplier Service
    - Supplier Service V201010
    - Supplier Service V201307
  - Payroll transaction service
  - Rate service
  - Applicant service
  - Workflow service
  - Document archive service
  - User administration service
  - Query engine service
  - Menu service
  - Import service
  - Timesheet service
  - Planning service
  - Commitment Accounting Service
  - Employment service
  - Personnel service
  - Balance service
  - Position service
  - Fixed/periodic PDs service
  - Absence services
  - Project service
  - Mobile Services
  - Funder Service
  - Resource Information Service
  - Research project service
  - Service order service
  - Education programme service
  - Student service

Roles/users filter

Client: Show all

Roles/users: All

Roles and users

Accountable role(ACCTABLE)
Aga Doskocz(AGAD)
Aga Szmigiel(AGA)
ala(ALA)
Alan testing Planner(ALAN)
Albert Aaberg(ALBERT)
Alistair White(ALIW)
Ana Guil(ANAG)
Andrzej Griffin(ANDRZEJ)
Ania Sulimka(ANIA)
ANNA UX(ANNA)
anne-lise(ANNE-LISE)
AON Administrator(AONADMIN)
AON Case Manager(AONCASEMGR)
AON Employee(AONEMPLOYEE)
AON HR Director(AONHRDIRECTO)
AON Manager(AONMANAGER)
API Data Control(APIDC)
Australian user(SYSAU)
Autotest ADFS user(MOBILEFED)
Beata Majchrowska(BEATAM)
Beata Meszek(BEATA)
Belgium user(SYSBE)
bente(BENTE)

Selected item

Supplier Service V201307

Access

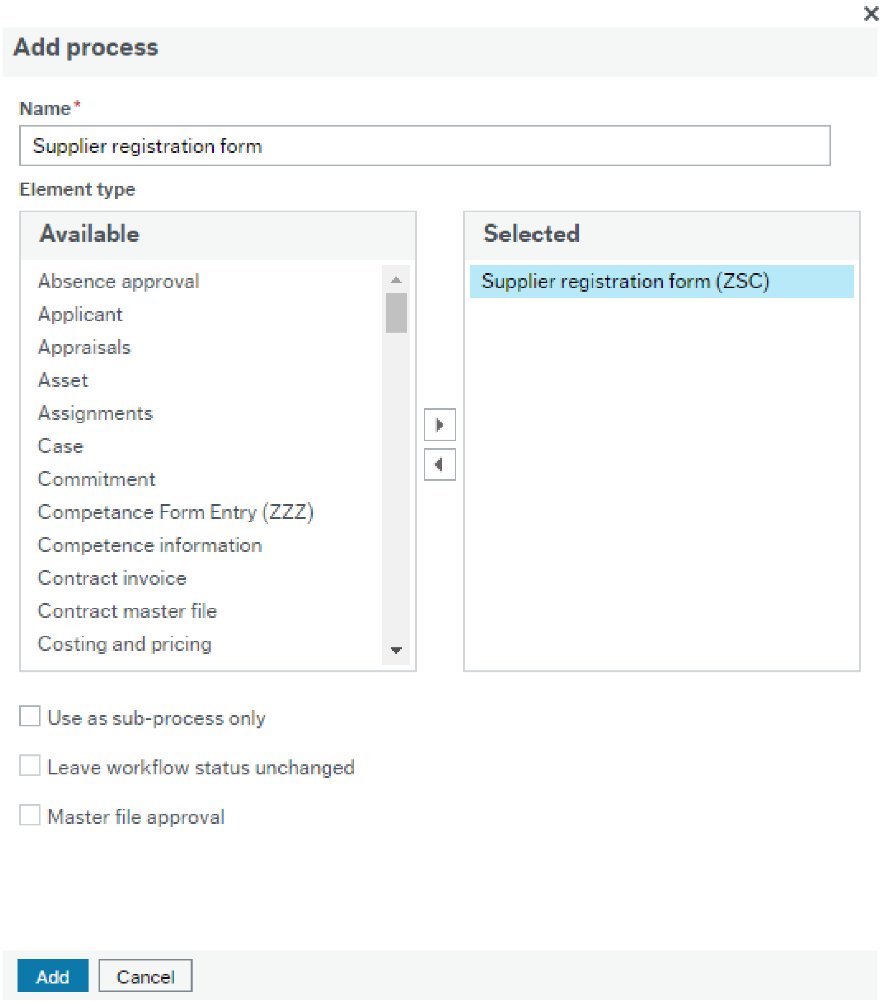
	C	R	U	D
SUPER(SUPER)	☑	☑	☑	☑
System Administrator(ADMINIS...	☑	☑	☑	☑
Read only(READONLY)	□	☑	□	□
SYSTEM(SYSTEM)	☑	☑	☑	☑

4. Clear the web service cache by entering in the browser the URL to the `Service.svc` file followed by `?FlushCache` string, for example:

   http:///test47m7.agresso.int/BusinessWorld-WebServicesAgrM7_491/Service.svc?FlushCache

Setting up the workflow

1. In the **Process definition** (XWF100) window, create a new process, **Supplier registration form**:



Add process

Name*

Supplier registration form

Element type

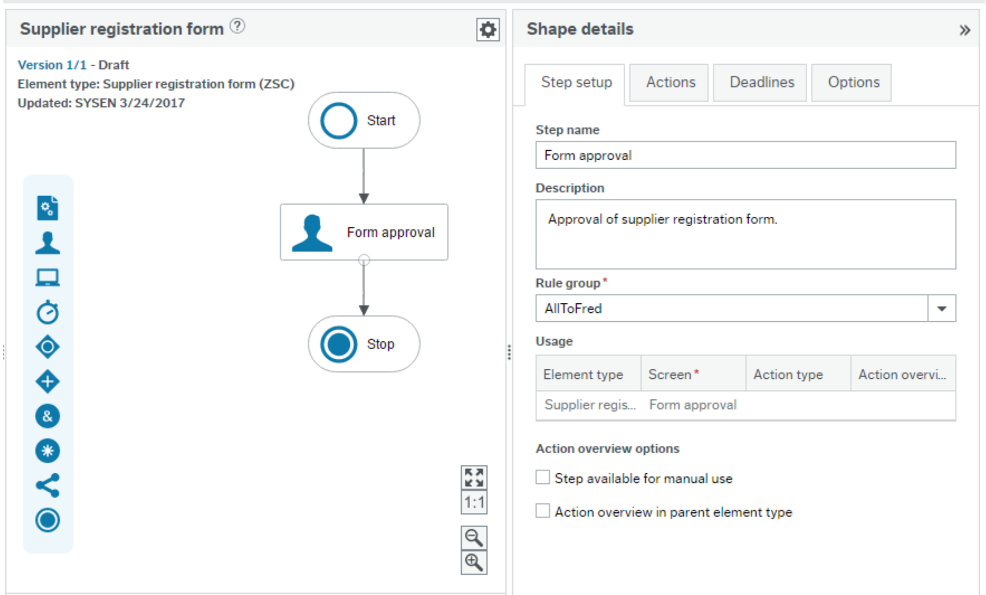
Available	Selected
Absence approval Applicant Appraisals Asset Assignments Case Commitment Competance Form Entry (ZZZ) Competence information Contract invoice Contract master file Costing and pricing	Supplier registration form (ZSC)

□ Use as sub-process only

□ Leave workflow status unchanged

□ Master file approval

2. In the **Process definition** (XWF100) window, add the **Form approval** user step:



Supplier registration form

Version 1/1 - Draft
Element type: Supplier registration form (ZSC)
Updated: SYSEN 3/24/2017

Shape details

Step name: Form approval

Description: Approval of supplier registration form.

Rule group*: AllToFred

Usage

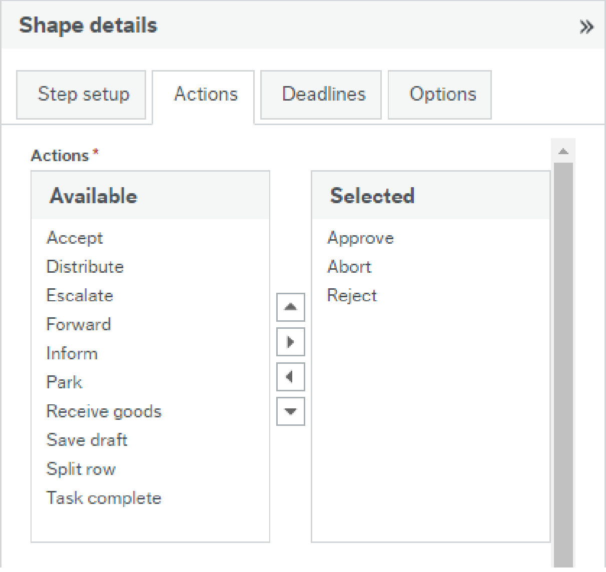
Element type	Screen*	Action type	Action overvi...
Supplier regis...	Form approval		

Action overview options

□ Step available for manual use

□ Action overview in parent element type

with step actions as below:

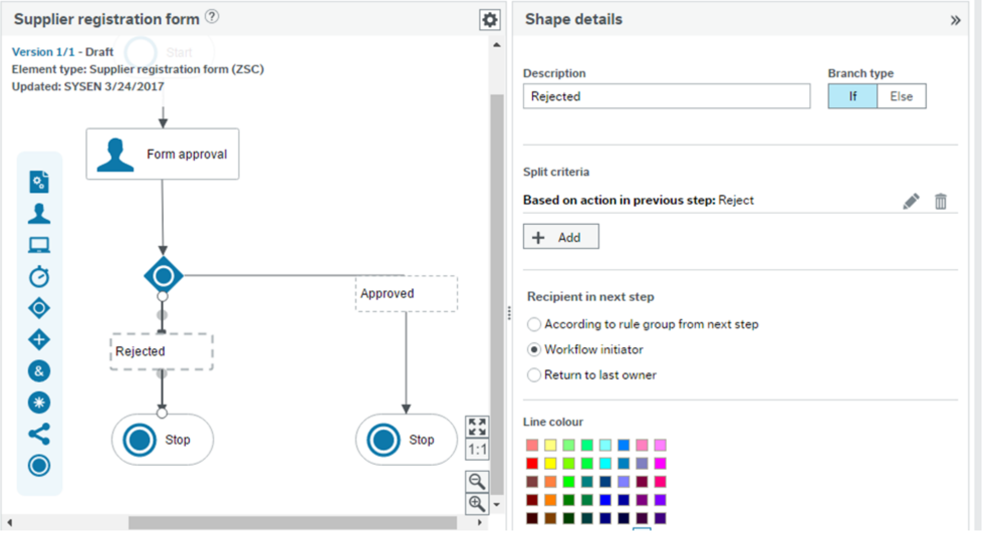


Shape details

Actions*

Available	Selected
Accept	Approve
Distribute	Abort
Escalate	Reject
Forward	
Inform	
Park	
Receive goods	
Save draft	
Split row	
Task complete	

3. In the **Process definition** (XWF100) window, add an OR-split node with the *Rejected* leg defined as **Based on action in previous step: Reject** and the *Accepted* leg defined as **Else**:



Supplier registration form

Version 1/1 - Draft

Element type: Supplier registration form (ZSC)

Updated: SYSEN 3/24/2017

Shape details

Description: Rejected

Branch type: [✓ If] Else

Split criteria

Based on action in previous step: Reject

+ Add

Recipient in next step

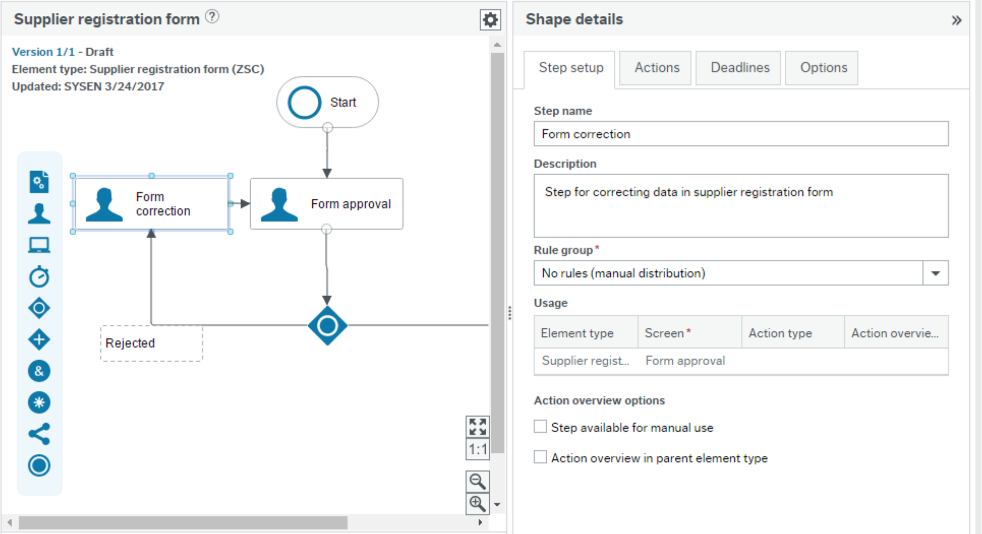
○ According to rule group from next step

● Workflow initiator

○ Return to last owner

Line colour

4. In the **Process definition** (XWF100) window, add a new user step, **Form correction**, and link it to the *Rejected* leg of the OR-split:



Supplier registration form

Version 1/1 - Draft

Element type: Supplier registration form (ZSC)

Updated: SYSEN 3/24/2017

Shape details

Step name: Form correction

Description: Step for correcting data in supplier registration form

Rule group*: No rules (manual distribution)

Usage

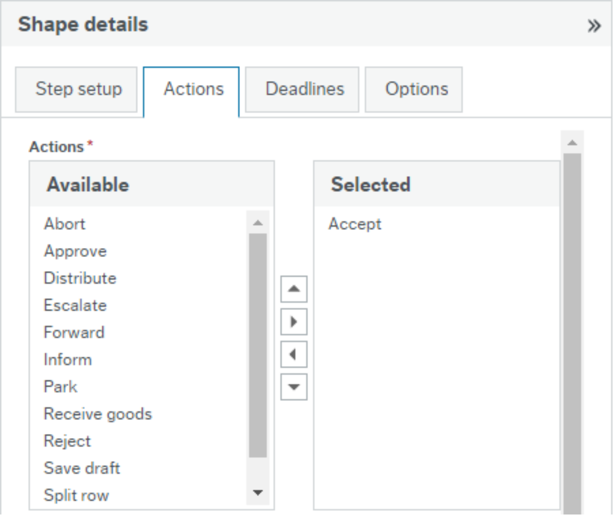
Element type	Screen*	Action type	Action overvie...
Supplier regist...	Form approval		

Action overview options

□ Step available for manual use

□ Action overview in parent element type

with step actions as below:



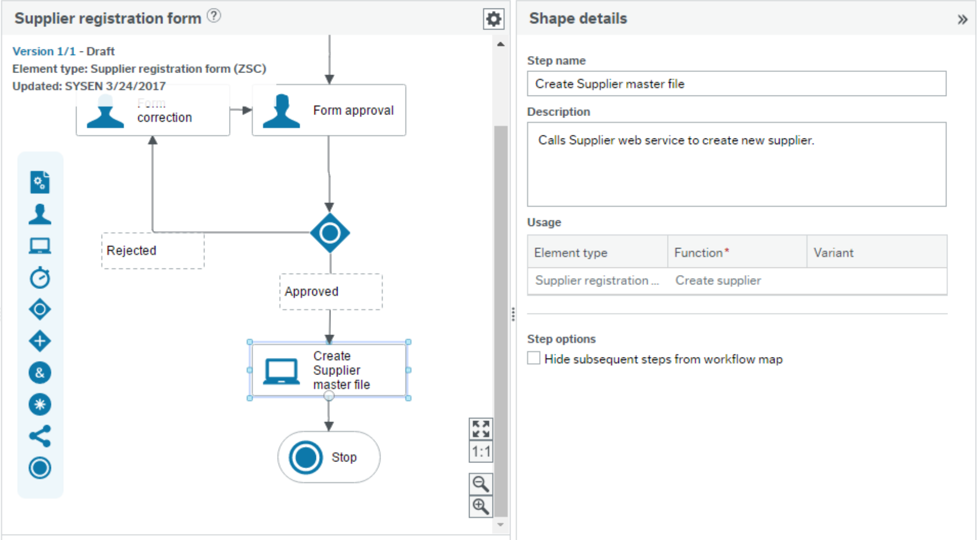
Shape details »

Step setup | Actions | Deadlines | Options

Actions *

Available	Selected
Abort	Accept
Approve	
Distribute	
Escalate	
Forward	
Inform	
Park	
Receive goods	
Reject	
Save draft	
Split row	

5. In the **Process definition** (XWF100) window, add system step **Create Supplier master file** and link it to the *Approved* leg of the OR-split. Select the **Create supplier** function for this step:



Supplier registration form

Version 1/1 - Draft
Element type: Supplier registration form (ZSC)
Updated: SYSEN 3/24/2017

Shape details »

Step name
Create Supplier master file

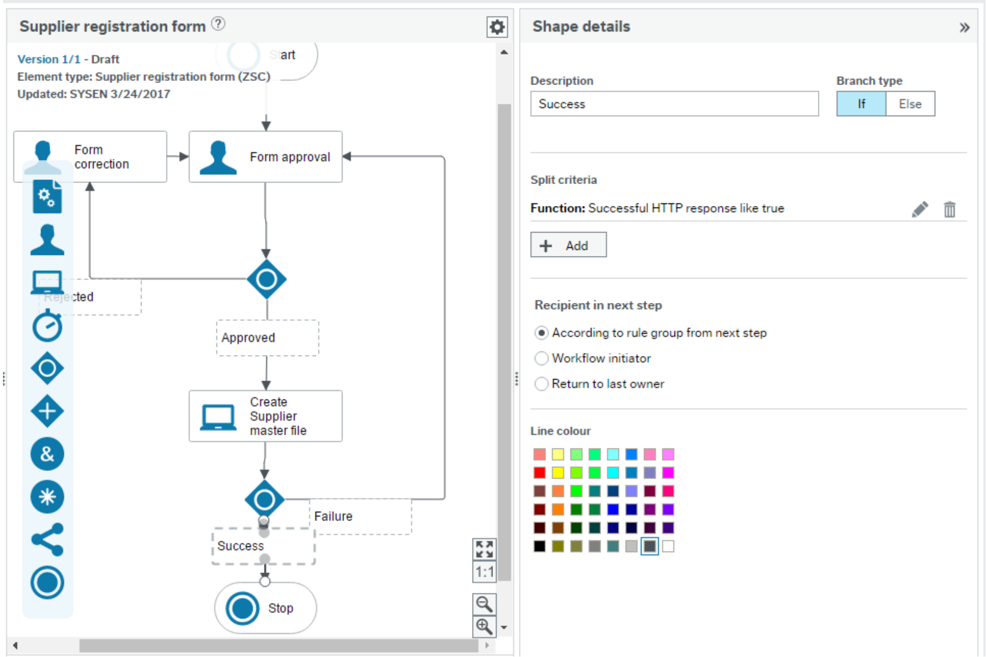
Description
Calls Supplier web service to create new supplier.

Usage

Element type	Function *	Variant
Supplier registration ...	Create supplier	

Step options
□ Hide subsequent steps from workflow map

6. In the **Process definition** (XWF100) window, add OR-split node linked to the **Create Supplier master file** system step with the *Success* leg defined (using the OR-split function) as **Successful HTTP response like true** and the *Failure* leg defined as **Else**:



Supplier registration form

Version 1/1 - Draft
Element type: Supplier registration form (ZSC)
Updated: SYSEN 3/24/2017

Shape details

Description: Success

Branch type: If | Else

Split criteria

Function: Successful HTTP response like true

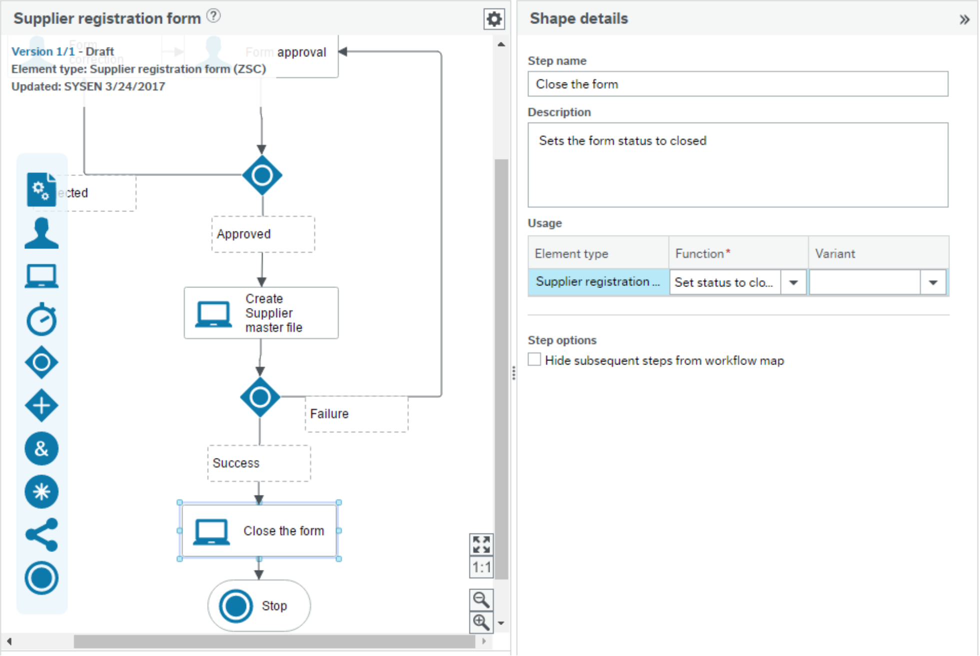
+ Add

Recipient in next step

● According to rule group from next step
○ Workflow initiator
○ Return to last owner

Line colour

7. In the **Process definition** (XWF100) window, add a new system step **Close the form** with **Set status to closed** function selected, and link it to the *Success* leg of the OR-split:



Supplier registration form

Version 1/1 - Draft
Element type: Supplier registration form (ZSC)
Updated: SYSEN 3/24/2017

Shape details

Step name: Close the form

Description: Sets the form status to closed

Usage

Element type	Function*	Variant
Supplier registration ...	Set status to clo...	

Step options

□ Hide subsequent steps from workflow map

8. Save the process definition, validate and activate it. Once the form is filled in, submitted and approved, the results can be viewed in the **Suppliers** window.

Scenario 2: User management form

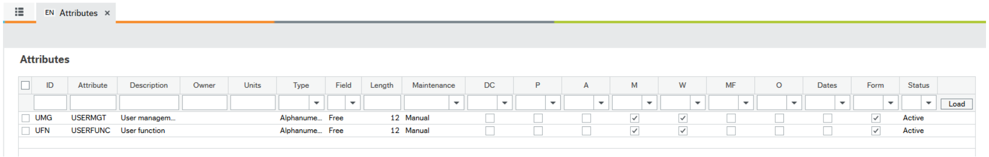
The scenario describes how to automate the process of manipulating user information with the use of a form. You will first create appropriate attributes, relations, flexi-field groups and a form. Next, you will create the service call setup and incorporate it in the workflow, so that user information (in this case, user roles) is updated once the form is filled in and saved.

Creating attributes and flexi-field groups

1. In Business World, go to the **Attributes** (TGL004) window. Click **Add** and create the following attributes:

   - **UMG** USERMGT (User management)
   - **UFN** USERFUNC (User function)

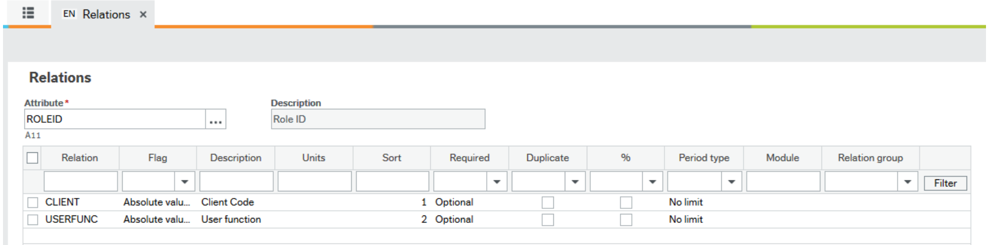
> **Note:** Make sure *Form* and *W* (Workflow) check boxes are selected for the attributes.



Attributes

□	ID	Attribute	Description	Owner	Units	Type	Field	Length	Maintenance	DC	P	A	M	W	MF	O	Dates	Form	Status
□	UMG	USERMGT	User managem...			Alphanume...	Free	12	Manual	□	□	□	☑	☑	□	□	□	☑	Active
□	UFN	USERFUNC	User function			Alphanume...	Free	12	Manual	□	□	□	☑	☑	□	□	□	☑	Active

2. In the **Relations** (TGL006) window, for the ROLEID attribute, add a relation to USERFUNC attribute:



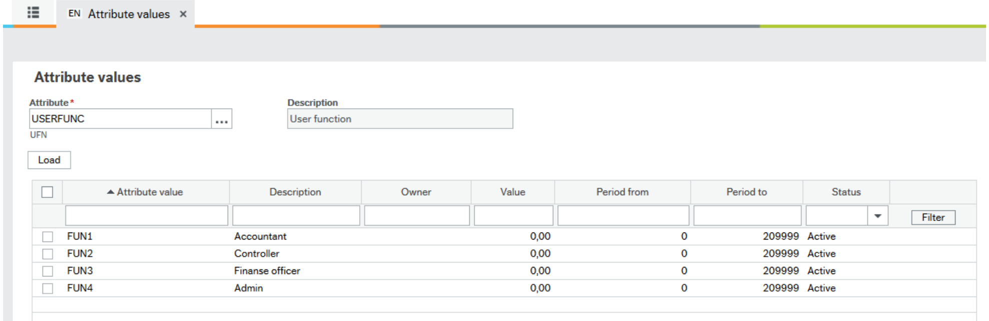
Relations

Attribute* ROLEID A11

Description Role ID

□	Relation	Flag	Description	Units	Sort	Required	Duplicate	%	Period type	Module	Relation group
□	CLIENT	Absolute valu...	Client Code		1	Optional	□	□	No limit		
□	USERFUNC	Absolute valu...	User function		2	Optional	□	□	No limit		

3. In the **Attribute values** (TGL005) window, assign values to the USERFUNC attribute:



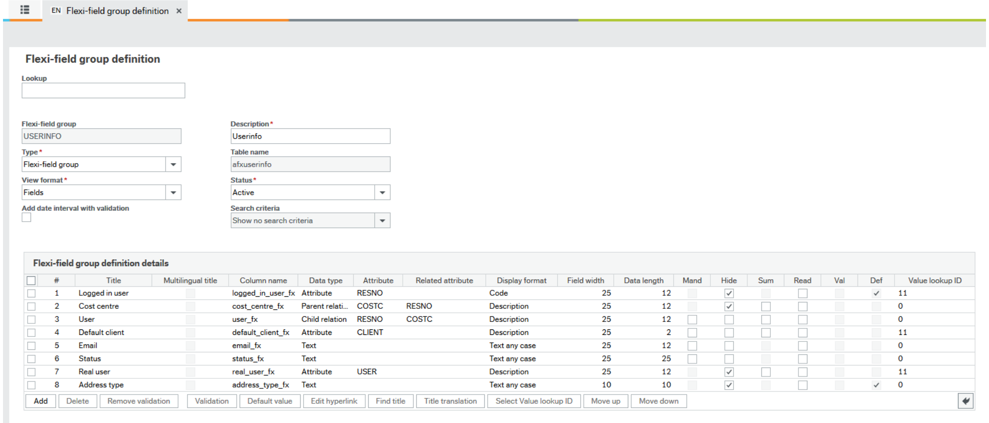
Attribute values

Attribute*: USERFUNC
UFN

Description: User function

□	Attribute value	Description	Owner	Value	Period from	Period to	Status
□	FUN1	Accountant		0,00	0	209999	Active
□	FUN2	Controller		0,00	0	209999	Active
□	FUN3	Finanse officer		0,00	0	209999	Active
□	FUN4	Admin		0,00	0	209999	Active

4. Now you will create flexi-field groups that will be reflected in your form as tabs. In the **Flexi-field group definition** (TAG060) window, create the USERINFO flexi-field group:



Flexi-field group definition

Lookup:

Flexi-field group: USERINFO

Description*: Userinfo

Type*: Flexi-field group

Table name: afxuserinfo

View format*: Fields

Status*: Active

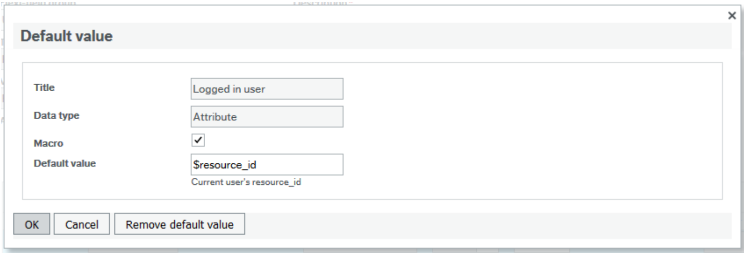
Add date interval with validation: □

Search criteria: Show no search criteria

Flexi-field group definition details

□	#	Title	Multilingual title	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup ID
□	1	Logged in user		logged_in_user_fx	Attribute	RESNO		Code	25	12		☑		□		✓	11
□	2	Cost centre		cost_centre_fx	Parent relati...	COSTC	RESNO	Description	25	12		☑	□	□			0
□	3	User		user_fx	Child relation	RESNO	COSTC	Description	25	12	□	□	□	□			0
□	4	Default client		default_client_fx	Attribute	CLIENT		Description	25	2	□	□	□	□			11
□	5	Email		email_fx	Text			Text any case	25	12	□	□		□			0
□	6	Status		status_fx	Text			Text any case	25	25	□	□		□			0
□	7	Real user		real_user_fx	Attribute	USER		Description	25	12		☑	□	□			11
□	8	Address type		address_type_fx	Text			Text any case	10	10		☑		□		✓	0

Select *Hide* and *Def* check boxes for **Logged in user** and **Address type**. Set the following default values for these fields:



Default value

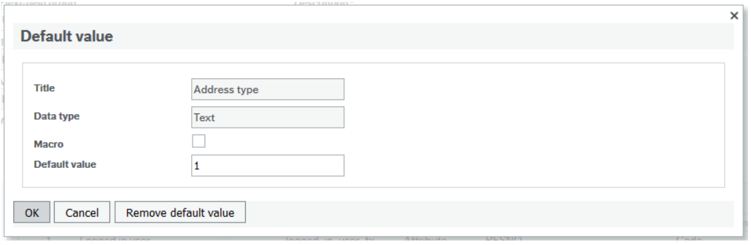
Title: Logged in user

Data type: Attribute

Macro: ☑

Default value: $resource_id
Current user's resource_id

By doing this, you are able to retrieve the resource number for the logged in user, as well as the cost centre, along with the list of other users assigned to it.

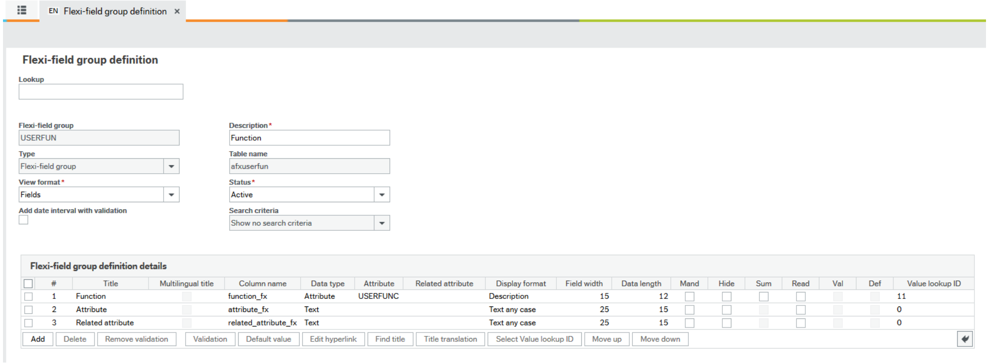


Default value ×

Title	Address type
Data type	Text
Macro	□
Default value	1

By doing this, you restrict the address type to **General** (1) only.

5. In the **Flexi-field group definition** (TAG060) window, create the USERFUN flexi-field group for user functions:



Flexi-field group definition

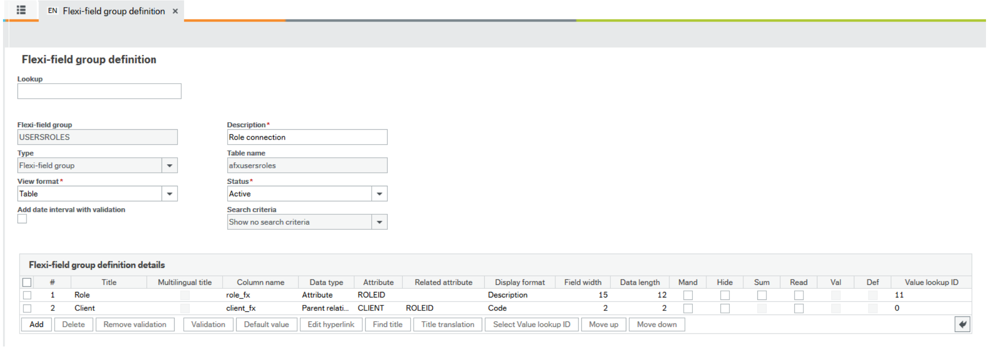
Lookup

Flexi-field group: USERFUN	Description*: Function
Type: Flexi-field group	Table name: afxuserfun
View format*: Fields	Status*: Active
Add date interval with validation: □	Search criteria: Show no search criteria

Flexi-field group definition details

□	#	Title	Multilingual title	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup ID
□	1	Function		function_fx	Attribute	USERFUNC		Description	15	12	□	□	□	□			11
□	2	Attribute		attribute_fx	Text			Text any case	25	15	□	□		□			0
□	3	Related attribute		related_attribute_fx	Text			Text any case	25	15	□	□		□			0

6. In the **Flexi-field group definition** (TAG060) window, create the USERSROLES flexi-field group:



Flexi-field group definition

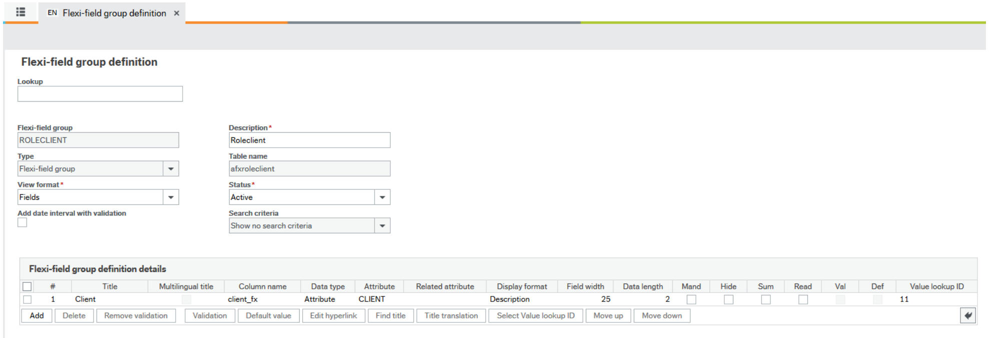
Lookup

Flexi-field group: USERSROLES	Description*: Role connection
Type: Flexi-field group	Table name: afxusersroles
View format*: Table	Status*: Active
Add date interval with validation: □	Search criteria: Show no search criteria

Flexi-field group definition details

□	#	Title	Multilingual title	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup ID
□	1	Role		role_fx	Attribute	ROLEID		Description	15	12	□	□	□	□			11
□	2	Client		client_fx	Parent relati...	CLIENT	ROLEID	Code	2	2	□	□		□			0

7. In the **Flexi-field group definition** (TAG060) window, create the ROLECLIENT flexi-field group. This way, you will add a list of available clients to your form:



Flexi-field group definition

Lookup

Flexi-field group: ROLECLIENT

Description*: Roleclient

Type: Flexi-field group

Table name: afxroleclient

View format*: Fields

Status*: Active

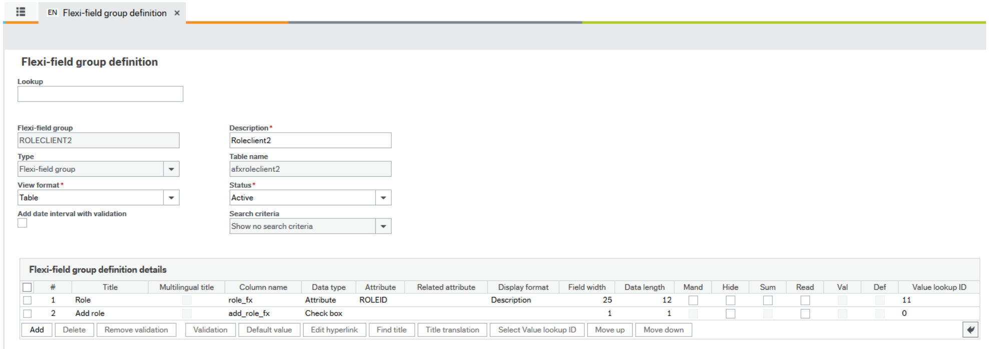
Add date interval with validation: □

Search criteria: Show no search criteria

Flexi-field group definition details

□	#	Title	Multilingual title	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup ID
□	1	Client		client_fx	Attribute	CLIENT		Description	25	2	□	□	□	□			11

8. In the **Flexi-field group definition** (TAG060) window, create the ROLECLIENT2 flexi-field group. This way, you will add to your form a list of roles applied to the selected client and a check box for removing the roles:



Flexi-field group definition

Lookup

Flexi-field group: ROLECLIENT2

Description*: Roleclient2

Type: Flexi-field group

Table name: afxroleclient2

View format*: Table

Status*: Active

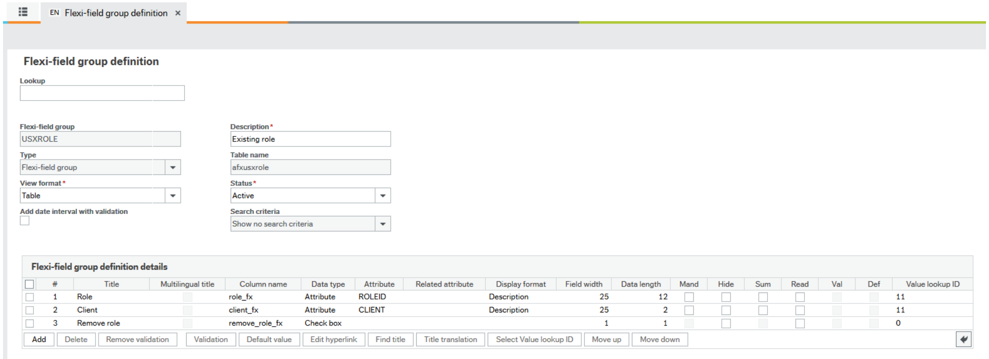
Add date interval with validation: □

Search criteria: Show no search criteria

Flexi-field group definition details

□	#	Title	Multilingual title	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup ID
□	1	Role		role_fx	Attribute	ROLEID		Description	25	12	□	□	□	□			11
□	2	Add role		add_role_fx	Check box				1	1		□		□			0

9. In the **Flexi-field group definition** (TAG060) window, create the USXROLE flexi-field group. This group shows roles already assigned to the client in your form:



Flexi-field group definition

Lookup

Flexi-field group: USXROLE

Description*: Existing role

Type: Flexi-field group

Table name: afxusxrole

View format*: Table

Status*: Active

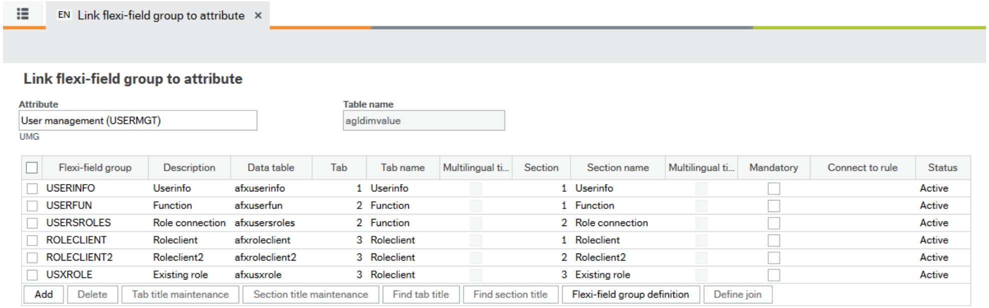
Add date interval with validation: □

Search criteria: Show no search criteria

Flexi-field group definition details

□	#	Title	Multilingual title	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup ID
□	1	Role		role_fx	Attribute	ROLEID		Description	25	12	□	□	□	□			11
□	2	Client		client_fx	Attribute	CLIENT		Description	25	2	□	□	□	□			11
□	3	Remove role		remove_role_fx	Check box				1	1		□		□			0

10. In the **Link flexi-field group to attribute** (TCR037) window, link the User management (USERMGT) attribute to the flexi-field groups you created in previous steps:



Link flexi-field group to attribute

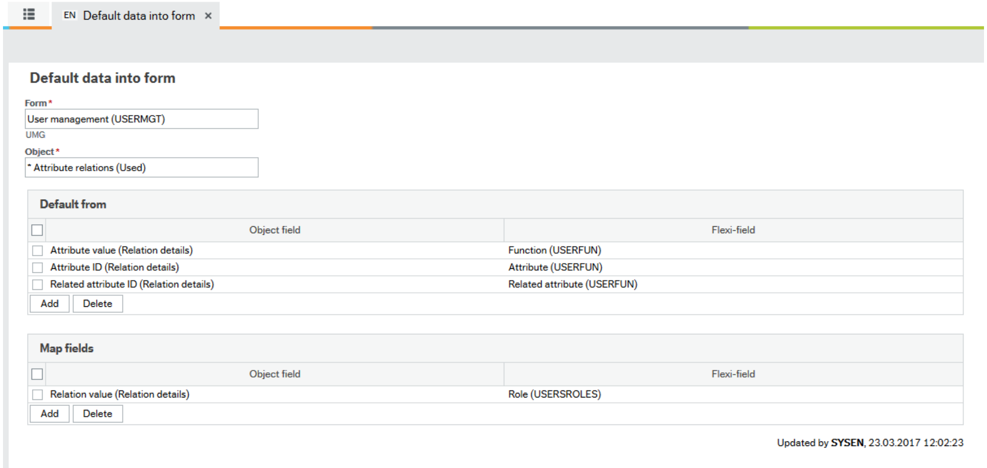
Attribute: User management (USERMGT)
UMG

Table name: agldimvalue

□	Flexi-field group	Description	Data table	Tab	Tab name	Multilingual ti...	Section	Section name	Multilingual ti...	Mandatory	Connect to rule	Status
□	USERINFO	Userinfo	afxuserinfo	1	Userinfo		1	Userinfo		□		Active
□	USERFUN	Function	afxuserfun	2	Function		1	Function		□		Active
□	USERSROLES	Role connection	afxusersroles	2	Function		2	Role connection		□		Active
□	ROLECLIENT	Roleclient	afxroleclient	3	Roleclient		1	Roleclient		□		Active
□	ROLECLIENT2	Roleclient2	afxroleclient2	3	Roleclient		2	Roleclient2		□		Active
□	USXROLE	Existing role	afxusxrole	3	Roleclient		3	Existing role		□		Active

> **Note:** Detailed information on working with flexi-fields can be found in Business World 7.0 **Flexi-fields** Reference manual.

11. If you need your form to be a separate menu item, in the **Form administration** (TCR022) window, select the USERMGT attribute in the *Form* field. Click the **Create menu** button to create a menu item dedicated to this form.
12. In the **Default data into form** (TCR040) window, select the USERMGT form and the **Attribute relations** object from which data will be defaulted into the form (in this case, user roles):



Default data into form

Form *: User management (USERMGT)
UMG

Object *: * Attribute relations (Used)

Default from

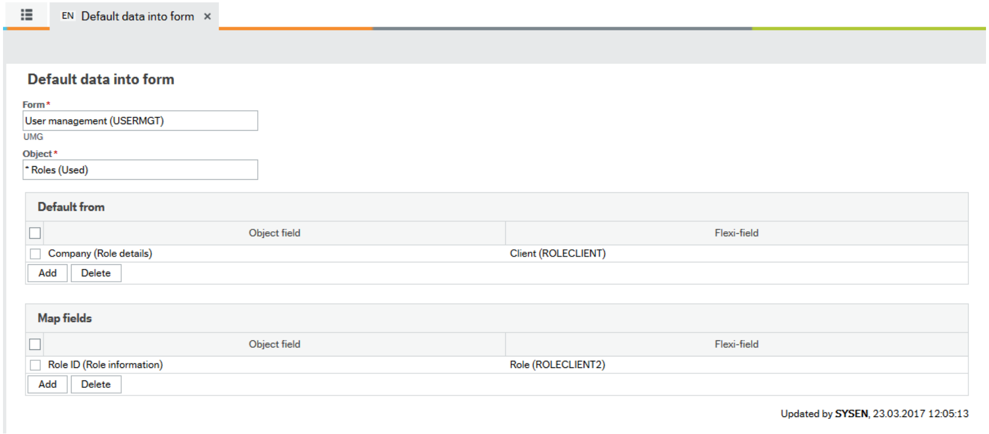
□	Object field	Flexi-field
□	Attribute value (Relation details)	Function (USERFUN)
□	Attribute ID (Relation details)	Attribute (USERFUN)
□	Related attribute ID (Relation details)	Related attribute (USERFUN)

Map fields

□	Object field	Flexi-field
□	Relation value (Relation details)	Role (USERSROLES)

Updated by **SYSEN**, 23.03.2017 12:02:23

13. In the **Default data into form** (TCR040) window, select the USERMGT form and the **Roles** object from which data will be defaulted into the form (in this case, client roles):



Default data into form

Form *
User management (USERMGT)
UMG

Object *
* Roles (Used)

Default from

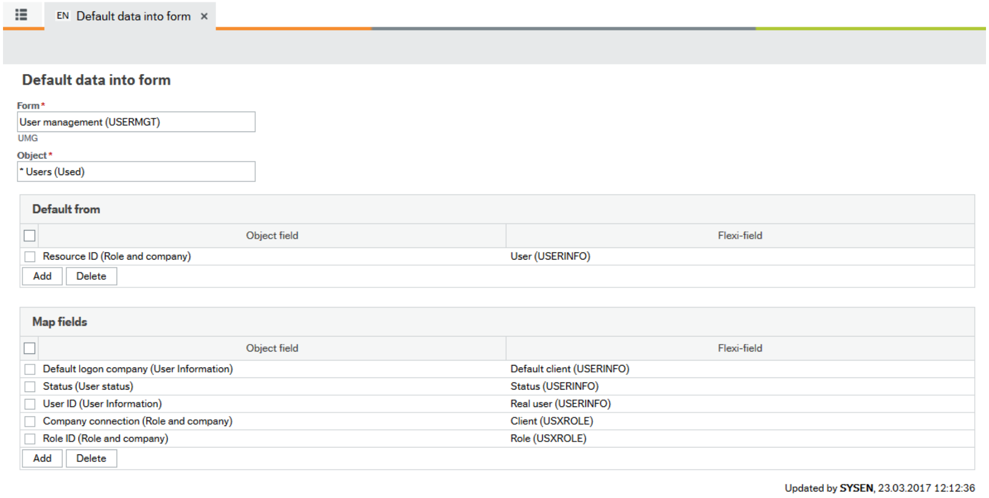
□	Object field	Flexi-field
□	Company (Role details)	Client (ROLECLIENT)

Map fields

□	Object field	Flexi-field
□	Role ID (Role information)	Role (ROLECLIENT2)

Updated by **SYSEN**, 23.03.2017 12:05:13

14. In the **Default data into form** (TCR040) window, select the USERMGT form and the **Users** object from which data will be defaulted into the form (in this case client, default client, role, status and real user):



Default data into form

Form *
User management (USERMGT)
UMG

Object *
* Users (Used)

Default from

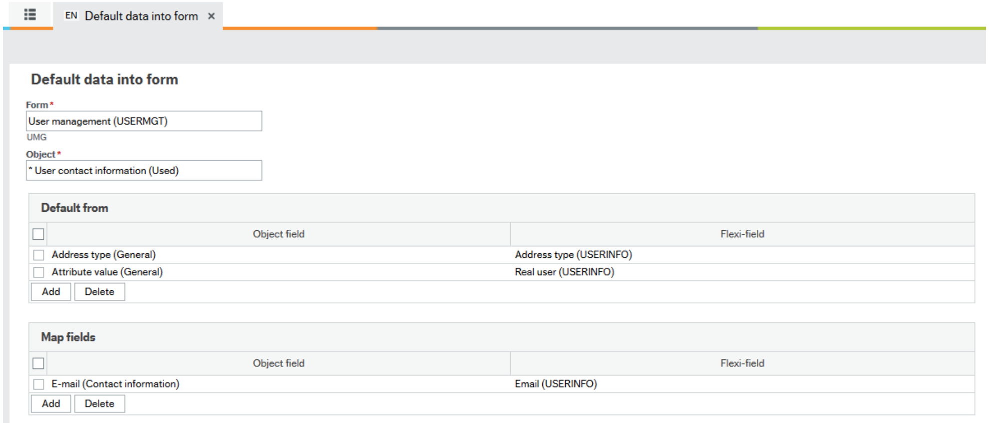
□	Object field	Flexi-field
□	Resource ID (Role and company)	User (USERINFO)

Map fields

□	Object field	Flexi-field
□	Default logon company (User Information)	Default client (USERINFO)
□	Status (User status)	Status (USERINFO)
□	User ID (User Information)	Real user (USERINFO)
□	Company connection (Role and company)	Client (USXROLE)
□	Role ID (Role and company)	Role (USXROLE)

Updated by **SYSEN**, 23.03.2017 12:12:36

15. In the **Default data into form** (TCR040) window, select the USERMGT form and the **User contact information** object from which data will be defaulted into the form (in this case, e-mail address):



Default data into form

Form *
User management (USERMGT)
UMG

Object *
* User contact information (Used)

Default from

□	Object field	Flexi-field
□	Address type (General)	Address type (USERINFO)
□	Attribute value (General)	Real user (USERINFO)

Add Delete

Map fields

□	Object field	Flexi-field
□	E-mail (Contact information)	Email (USERINFO)

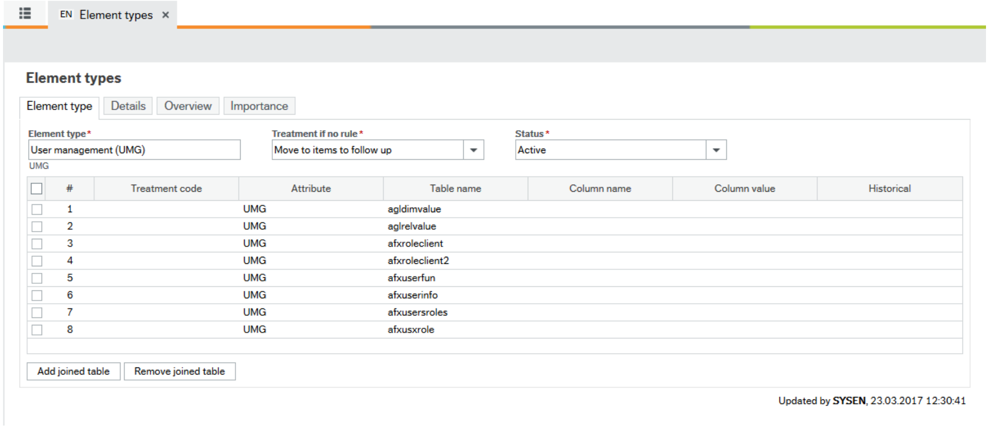
Add Delete

> **Note:** Detailed information on working with forms can be found in Business World7.0 **Forms** Reference manual.

16. In the **Menu access** (XAG003) window, grant appropriate access rights. For more details, see Scenario 1.

Setting up the service call

1. In the **Element types** (TWF032) window, create the **User management** element type:



Element types

Element type *: User management (UMG)
UMG

Treatment if no rule *: Move to items to follow up

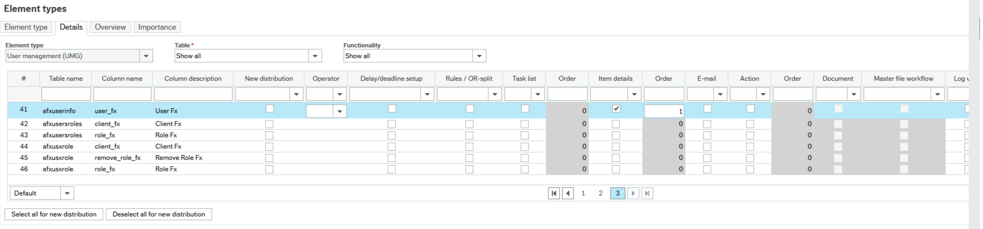
Status *: Active

□	#	Treatment code	Attribute	Table name	Column name	Column value	Historical
□	1		UMG	agldimvalue			
□	2		UMG	aglrelvalue			
□	3		UMG	afxroleclient			
□	4		UMG	afxroleclient2			
□	5		UMG	afxuserfun			
□	6		UMG	afxuserinfo			
□	7		UMG	afxusersroles			
□	8		UMG	afxusxrole			

Add joined table Remove joined table

Updated by **SYSEN**, 23.03.2017 12:30:41

On the **Details** tab, define element type properties:



Element types

Element type: User management (UMG)

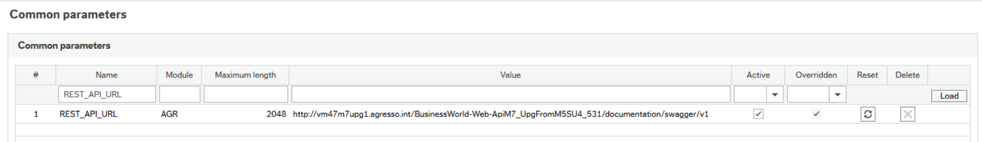
Table *: Show all

Functionality: Show all

#	Table name	Column name	Column description	New distribution	Operator	Delay/deadline setup	Rules / OR-split	Task list	Order	Item details	Order	E-mail	Action	Order	Document	Master file workflow	Log v
41	afxuserinfo	user_fx	User Fx	☐		☐	☐	☐	0	☑	1	☐	☐	0	☐	☐	☐
42	afxusersroles	client_fx	Client Fx	☐		☐	☐	☐	0	☐	0	☐	☐	0	☐	☐	☐
43	afxusersroles	role_fx	Role Fx	☐		☐	☐	☐	0	☐	0	☐	☐	0	☐	☐	☐
44	afxusxrole	client_fx	Client Fx	☐		☐	☐	☐	0	☐	0	☐	☐	0	☐	☐	☐
45	afxusxrole	remove_role_fx	Remove Role Fx	☐		☐	☐	☐	0	☐	0	☐	☐	0	☐	☐	☐
46	afxusxrole	role_fx	Role Fx	☐		☐	☐	☐	0	☐	0	☐	☐	0	☐	☐	☐

Default
1 2 3
Select all for new distribution
Deselect all for new distribution

2. Now you can start configuring your service call setup. Make sure you have set up the REST_API_URL common parameter beforehand:

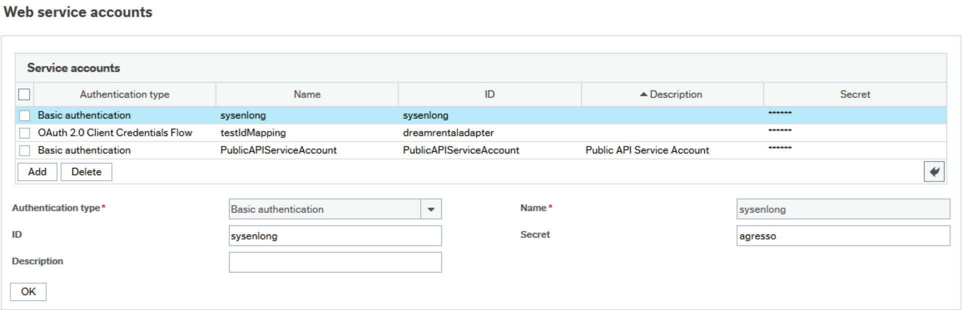


Common parameters

Common parameters

#	Name	Module	Maximum length	Value	Active	Overridden	Reset	Delete
	REST_API_URL							
1	REST_API_URL	AGR	2048	http://vm47m7upg1.agresso.int/BusinessWorld-Web-ApiM7_UpgFromM5SU4_531/documentation/swagger/v1	☑	✓		☒

3. Make sure you have your web service accounts set up, too:



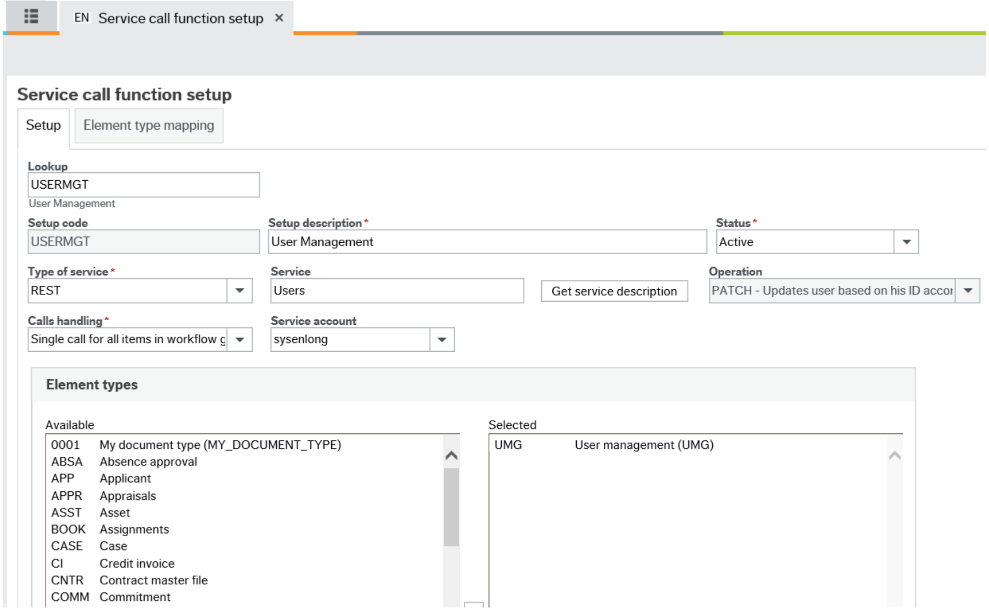
Web service accounts

Service accounts

☐	Authentication type	Name	ID	Description	Secret
☐	Basic authentication	sysenlong	sysenlong		******
☐	OAuth 2.0 Client Credentials Flow	testIdMapping	dreamrentaladapter		******
☐	Basic authentication	PublicAPIServiceAccount	PublicAPIServiceAccount	Public API Service Account	******

Authentication type *	Basic authentication	Name *	sysenlong
ID	sysenlong	Secret	agresso
Description			

4. In the **Service call function setup** (TWF260) window, create the **User management** setup:



Service call function setup

Lookup: USERMGT
User Management

Setup code: USERMGT

Setup description *: User Management

Status *: Active

Type of service *: REST

Service: Users

Operation: PATCH - Updates user based on his ID accor

Calls handling *: Single call for all items in workflow c

Service account: sysenlong

Element types

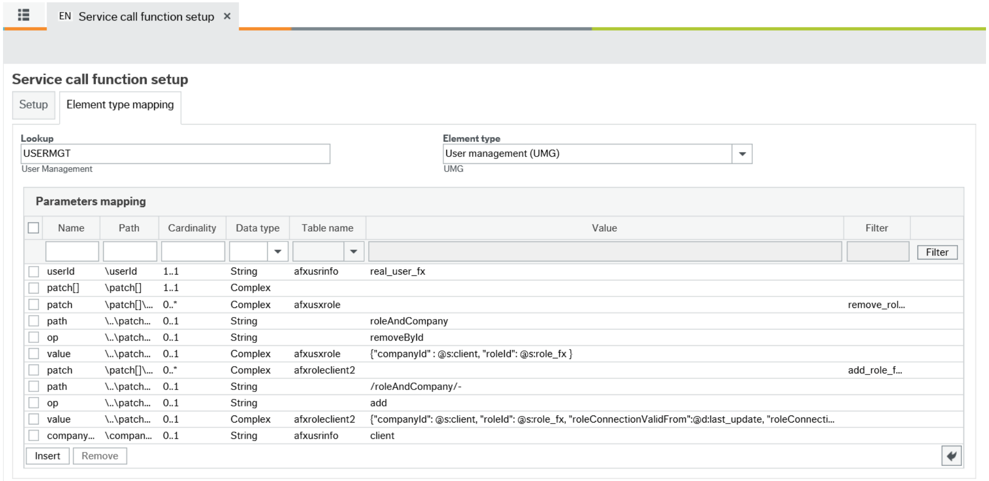
Available

0001	My document type (MY_DOCUMENT_TYPE)
ABSA	Absence approval
APP	Applicant
APPR	Appraisals
ASST	Asset
BOOK	Assignments
CASE	Case
CI	Credit invoice
CNTR	Contract master file
COMM	Commitment

Selected

UMG	User management (UMG)

5. On the **Element type mapping** tab, set up mapping for the parameters. In our case we will be adding and removing roles, so we need the following setup:



Service call function setup

Lookup: USERMGT
User Management

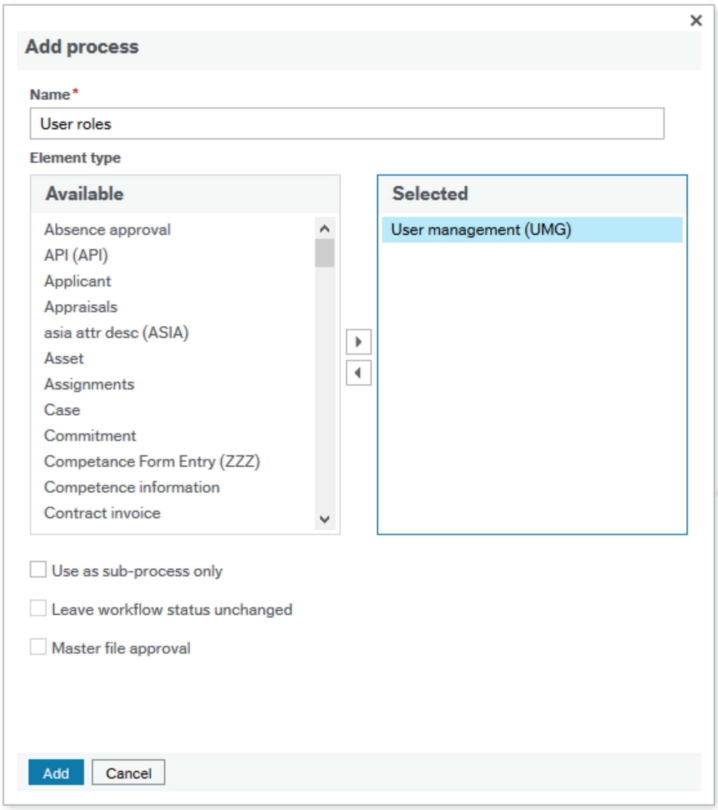
Element type: User management (UMG)
UMG

Parameters mapping

□	Name	Path	Cardinality	Data type	Table name	Value	Filter
□	userId	\userId	1..1	String	afxusrinfo	real_user_fx	
□	patch[]	\patch[]	1..1	Complex			
□	patch	\patch[]\...	0..*	Complex	afxusxrole		remove_rol...
□	path	\..\patch...	0..1	String		roleAndCompany	
□	op	\..\patch...	0..1	String		removeById	
□	value	\..\patch...	0..1	Complex	afxusxrole	{"companyId" : @s:client, "roleId": @s:role_fx }	
□	patch	\patch[]\...	0..*	Complex	afxroleclient2		add_role_f...
□	path	\..\patch...	0..1	String		/roleAndCompany/-	
□	op	\..\patch...	0..1	String		add	
□	value	\..\patch...	0..1	Complex	afxroleclient2	{"companyId": @s:client, "roleId": @s:role_fx, "roleConnectionValidFrom":@d:last_update, "roleConnecti...	
□	company...	\compan...	0..1	String	afxusrinfo	client	

Setting up the workflow

1. In the **Process definition** (XWF100) window, create a new process, **User roles**:



Add process

Name*: User roles

Element type

Available:
- Absence approval
- API (API)
- Applicant
- Appraisals
- asia attr desc (ASIA)
- Asset
- Assignments
- Case
- Commitment
- Competance Form Entry (ZZZ)
- Competence information
- Contract invoice

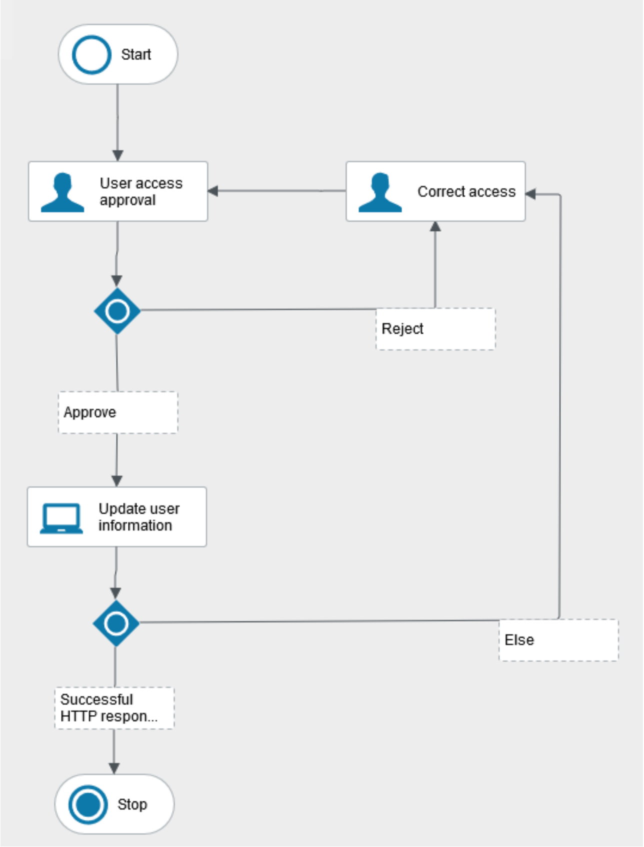
Selected:
- User management (UMG)

□ Use as sub-process only

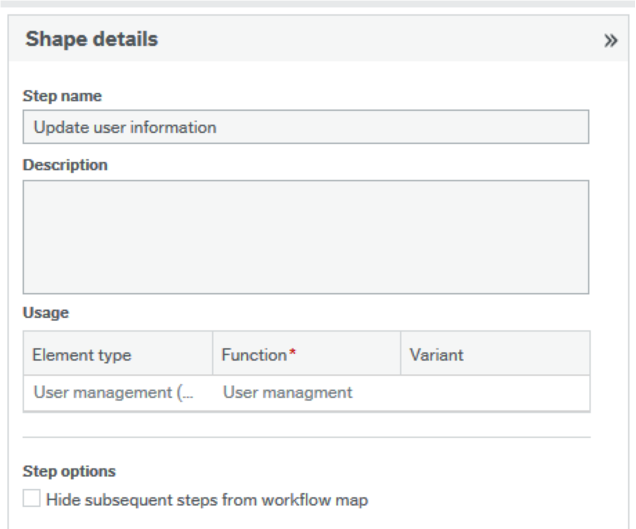
□ Leave workflow status unchanged

□ Master file approval

2. In the **Process definition** (XWF100) window, create a standard workflow for filling in and approving a form. Add the **Update user information** system step:



3. Set the following step actions for the system step:



Shape details »

Step name

Update user information

Description

Usage

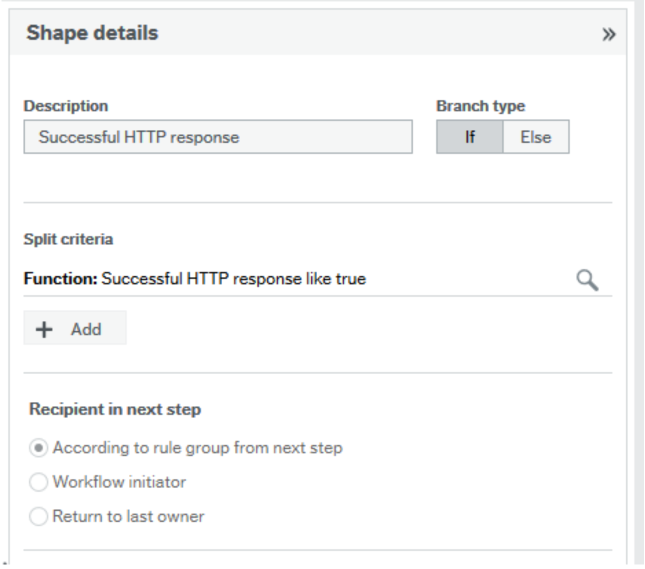
Element type	Function*	Variant
User management (...	User managment	

Step options

□ Hide subsequent steps from workflow map

The step will call the RESTful API, as defined in the **User management** service call setup created in previous steps.

4. Set the following step actions for the OR-split:



Shape details

Description: Successful HTTP response

Branch type: [✓ If] Else

Split criteria

Function: Successful HTTP response like true

Recipient in next step

● According to rule group from next step

○ Workflow initiator

○ Return to last owner

This means that if the system step returns a successful HTTP response, the workflow will continue to the next step and stop.

> **Note:** If the HTTP response code is different than 2xx, you can look for possible reasons in the workflow log. The workflow log functionality is delivered with Business World.

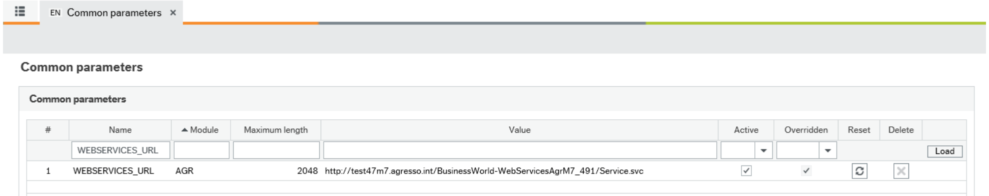
5. Save the process definition, validate and activate it. Once the form is filled in (e.g. a role is added or removed by selecting appropriate check boxes in the form), submitted and approved, the results will be visible in the **User master file** window (in this case, on the **Role and company** tab).

Scenario 3: Invoice reversal

This scenario describes how to reverse customer invoices using a form. Please note that some procedures have already been described in the previous scenarios, and can be referred to, if more details are needed.

Creating attributes and flexi-field groups

1. In the **Common parameters** (TAG023) window, set up and activate the WEBSERVICES_URL common parameter:

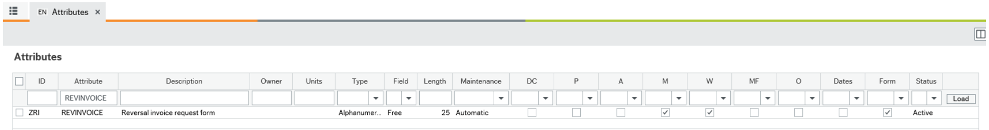


Common parameters

Common parameters

#	Name	▲ Module	Maximum length	Value	Active	Overridden	Reset	Delete	
	WEBSERVICES_URL								Load
1	WEBSERVICES_URL	AGR	2048	http://test47m7.agresso.int/BusinessWorld-WebServicesAgrM7_491/Service.svc	☑	✓		☒	

2. Go to the **Attributes** (TGL004) window. Click **Add** and create the **ZRI** REVINVOICE attribute:

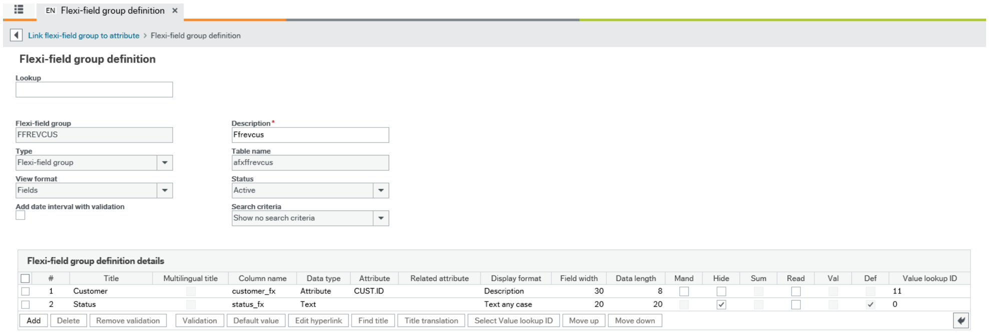


Attributes

□	ID	Attribute	Description	Owner	Units	Type	Field	Length	Maintenance	DC	P	A	M	W	MF	O	Dates	Form	Status	
		REVINVOICE																		Load
□	ZRI	REVINVOICE	Reversal invoice request form			Alphanumer...	Free	25	Automatic	□	□	□	☑	☑	□	□	□	☑	Active	

> **Note:** Make sure *Form* and *W* (Workflow) check boxes are selected for the attribute.

3. In the **Flexi-field group definition** (TAG060) window, create the FFREVCUS flexi-field group. Hide the *Status* field:


EN Flexi-field group definition ×
Link flexi-field group to attribute > Flexi-field group definition

Flexi-field group definition

Lookup

Flexi-field group: FFREVCUS

Type: Flexi-field group

View format: Fields

Add date interval with validation: □

Description*: Ffrevcus

Table name: afxffrevcus

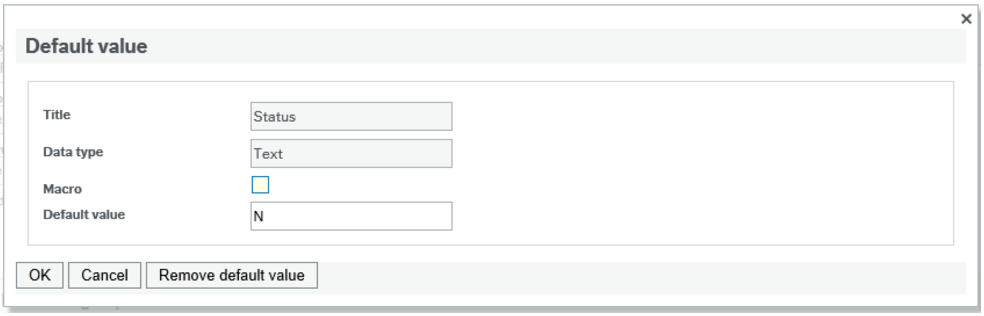
Status: Active

Search criteria: Show no search criteria

Flexi-field group definition details

□	#	Title	Multilingual title	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup ID
□	1	Customer		customer_fx	Attribute	CUST.ID		Description	30	8	□	□		□			11
□	2	Status		status_fx	Text			Text any case	20	20		☑		□		✓	0

4. Set up default value for *Status*.

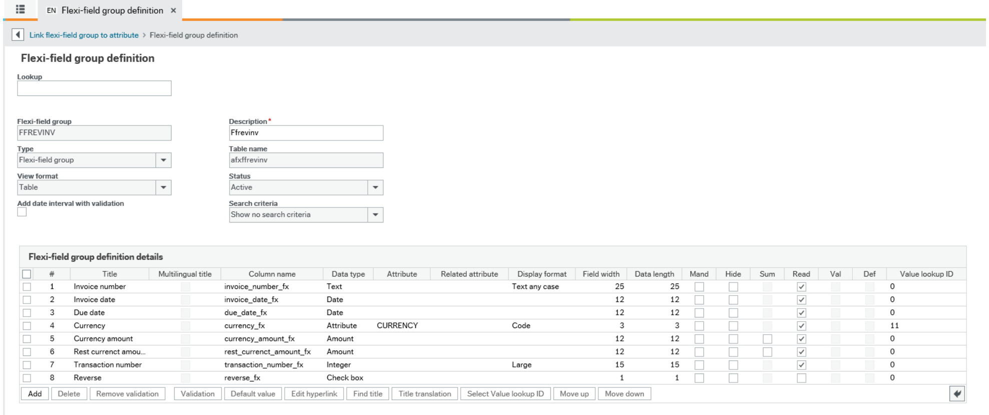


Default value

Title	Status
Data type	Text
Macro	□
Default value	N

By doing this, you restrict the status to **N** (Active) only.

5. In the **Flexi-field group definition** (TAG060) window, create the FFREVINV flexi-field group. Hide the *Status* field:



EN Flexi-field group definition
Link flexi-field group to attribute > Flexi-field group definition

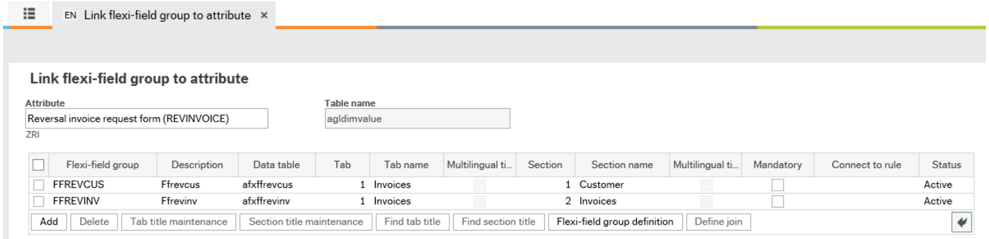
Flexi-field group definition

Lookup	
Flexi-field group	FFREVINV
Description*	Ffrevinv
Type	Flexi-field group
Table name	afxffrevinv
View format	Table
Status	Active
Add date interval with validation	□
Search criteria	Show no search criteria

Flexi-field group definition details

□	#	Title	Multilingual title	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup ID
□	1	Invoice number		invoice_number_fx	Text			Text any case	25	25	□	□		☑			0
□	2	Invoice date		invoice_date_fx	Date				12	12	□	□		☑			0
□	3	Due date		due_date_fx	Date				12	12	□	□		☑			0
□	4	Currency		currency_fx	Attribute	CURRENCY		Code	3	3	□	□		☑			11
□	5	Currency amount		currency_amount_fx	Amount				12	12	□	□	□	☑			0
□	6	Rest currenct amou...		rest_currenct_amount_fx	Amount				12	12	□	□	□	☑			0
□	7	Transaction number		transaction_number_fx	Integer			Large	15	15	□	□		☑			0
□	8	Reverse		reverse_fx	Check box				1	1	□	□		□			0

6. In the **Link flexi-field group to attribute** (TCR037) window, link the REVINVOICE attribute to the flexi-field groups you created in previous steps:



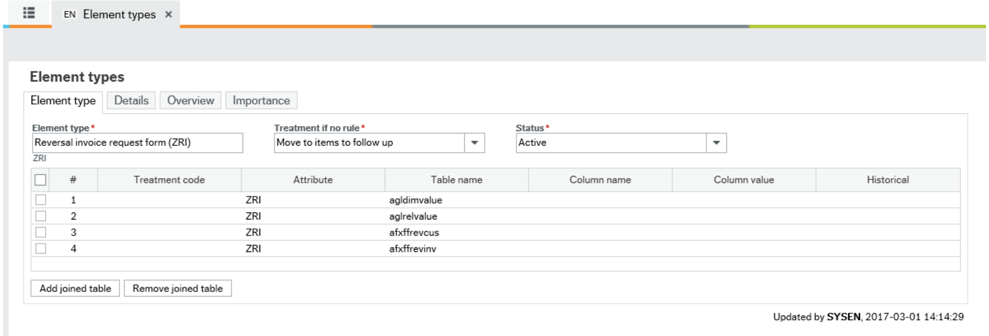
Link flexi-field group to attribute

Attribute	Reversal invoice request form (REVINVOICE) ZRI
Table name	agldimvalue

□	Flexi-field group	Description	Data table	Tab	Tab name	Multilingual ti...	Section	Section name	Multilingual ti...	Mandatory	Connect to rule	Status
□	FFREVCUS	Ffrevcus	afxffrevcus	1	Invoices		1	Customer		□		Active
□	FFREVINV	Ffrevinv	afxffrevinv	1	Invoices		2	Invoices		□		Active

> **Note:** Detailed information on working with flexi-fields can be found in Business World 7.0 **Flexi-fields** Reference manual.

7. In the **Element types** (TWF032) window, create the element type based on the **ZRI** attribute and linked flexi-field groups:



EN Element types ×

Element types

Element type | Details | Overview | Importance

Element type *
Reversal invoice request form (ZRI)
ZRI

Treatment if no rule *
Move to items to follow up

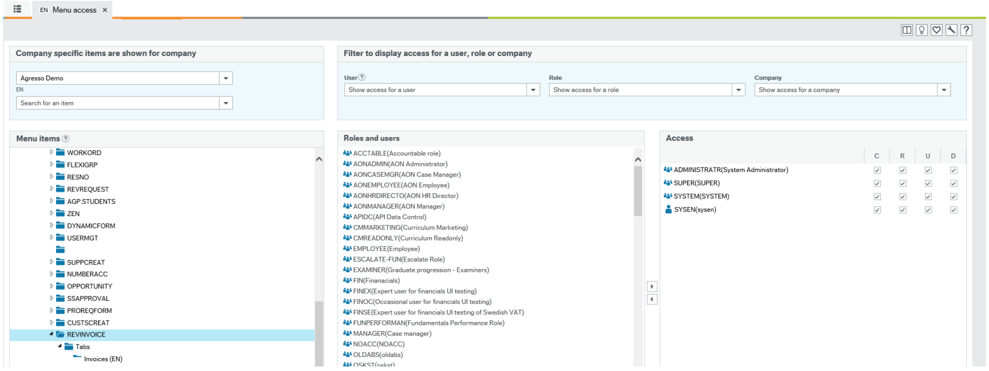
Status *
Active

□	#	Treatment code	Attribute	Table name	Column name	Column value	Historical
□	1		ZRI	agldimvalue			
□	2		ZRI	aglrelvalue			
□	3		ZRI	afxffrevcus			
□	4		ZRI	afxffrevinv			

Add joined table | Remove joined table

Updated by **SYSEN**, 2017-03-01 14:14:29

8. In the **Menu access** (XAG003) window, grant access rights to the form:



EN Menu access ×

Company specific items are shown for company

Agresso Demo
EN
Search for an item

Filter to display access for a user, role or company

User
Show access for a user
Role
Show access for a role
Company
Show access for a company

Menu items

- WORKORD
- FLEXIGRP
- RESNO
- REVREQUEST
- AGP.STUDENTS
- ZEN
- DYNAMICFORM
- USERMGT
- SUPPCREAT
- NUMBERACC
- OPPORTUNITY
- SSAPPROVAL
- PROREQFORM
- CUSTSCREAT
- REVINVOICE
  - Tabs
    - Invoices (EN)

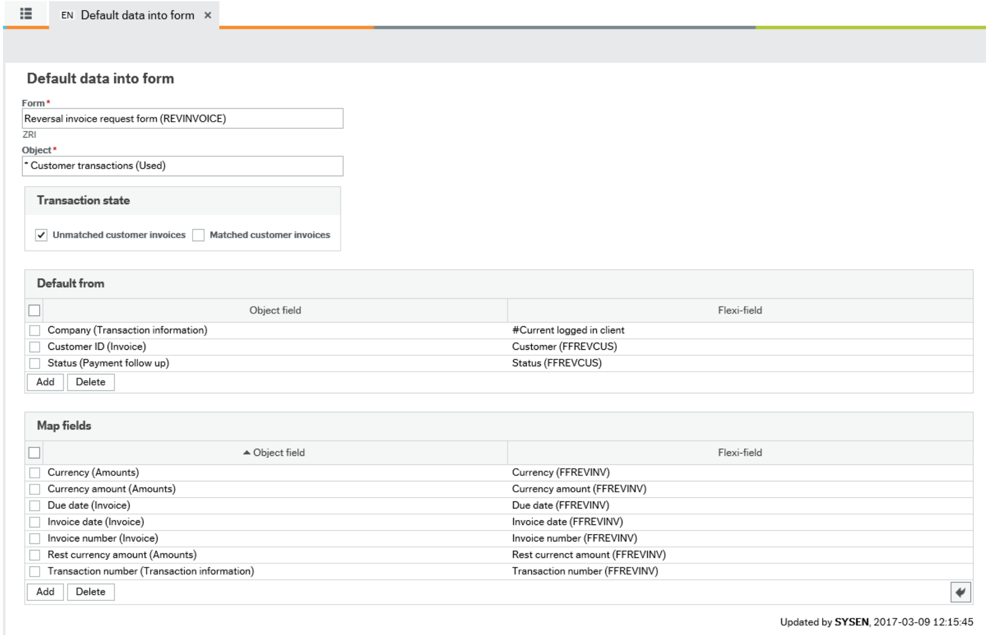
Roles and users

- ACCTABLE(Accountable role)
- AONADMIN(AON Administrator)
- AONCASEMGR(AON Case Manager)
- AONEMPLOYEE(AON Employee)
- AONHRDIRECTO(AON HR Director)
- AONMANAGER(AON Manager)
- APIDC(API Data Control)
- CMMARKETING(Curriculum Marketing)
- CMREADONLY(Curriculum Readonly)
- EMPLOYEE(Employee)
- ESCALATE-FUN(Escalate Role)
- EXAMINER(Graduate progression - Examiners)
- FIN(Finanacials)
- FINEX(Expert user for financials UI testing)
- FINOC(Occasional user for financials UI testing)
- FINSE(Expert user for financials UI testing of Swedish VAT)
- FUNPERFORMAN(Fundamentals Performance Role)
- MANAGER(Case manager)
- NOACC(NOACC)
- OLDABS(oldabs)
- OSKST(oskst)

Access

	C	R	U	D
ADMINISTRATR(System Administrator)	☑	☑	☑	☑
SUPER(SUPER)	☑	☑	☑	☑
SYSTEM(SYSTEM)	☑	☑	☑	☑
SYSEN(sysen)	☑	☑	☑	☑

9. In the **Default data into form** (TCR040) window, create relationship between the **Customer transactions** object and the form. This means that for our form, the company and customer ID will be defaulted from this object:



EN Default data into form ×

Default data into form

Form *
Reversal invoice request form (REVINVOICE)
ZRI

Object *
* Customer transactions (Used)

Transaction state

☑ Unmatched customer invoices □ Matched customer invoices

Default from

□	Object field	Flexi-field
□	Company (Transaction information)	#Current logged in client
□	Customer ID (Invoice)	Customer (FFREVCUS)
□	Status (Payment follow up)	Status (FFREVCUS)

Add | Delete

Map fields

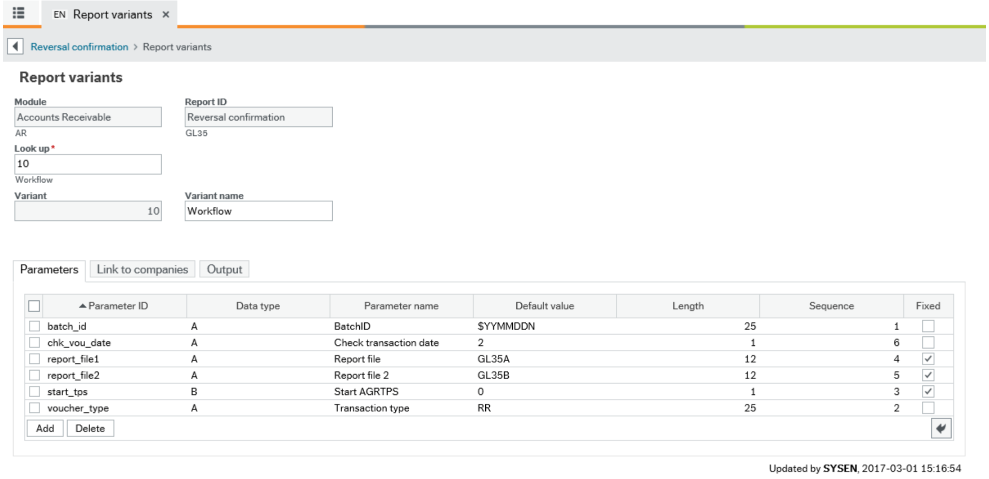
□	▲ Object field	Flexi-field
□	Currency (Amounts)	Currency (FFREVINV)
□	Currency amount (Amounts)	Currency amount (FFREVINV)
□	Due date (Invoice)	Due date (FFREVINV)
□	Invoice date (Invoice)	Invoice date (FFREVINV)
□	Invoice number (Invoice)	Invoice number (FFREVINV)
□	Rest currency amount (Amounts)	Rest current amount (FFREVINV)
□	Transaction number (Transaction information)	Transaction number (FFREVINV)

Add | Delete

Updated by **SYSEN**, 2017-03-09 12:15:45

> **Note:** Detailed information on working with forms can be found in Business World 7.0 **Forms** Reference manual.

10. Under **Accounting** > **Accounts Receivable** > **Reversal** > **Reversal confirmation**, create a new report variant for the process. Click the **Report variants** button and save the following report variant:



Report variants

Module: Accounts Receivable
AR

Report ID: Reversal confirmation
GL35

Look up *: 10
Workflow

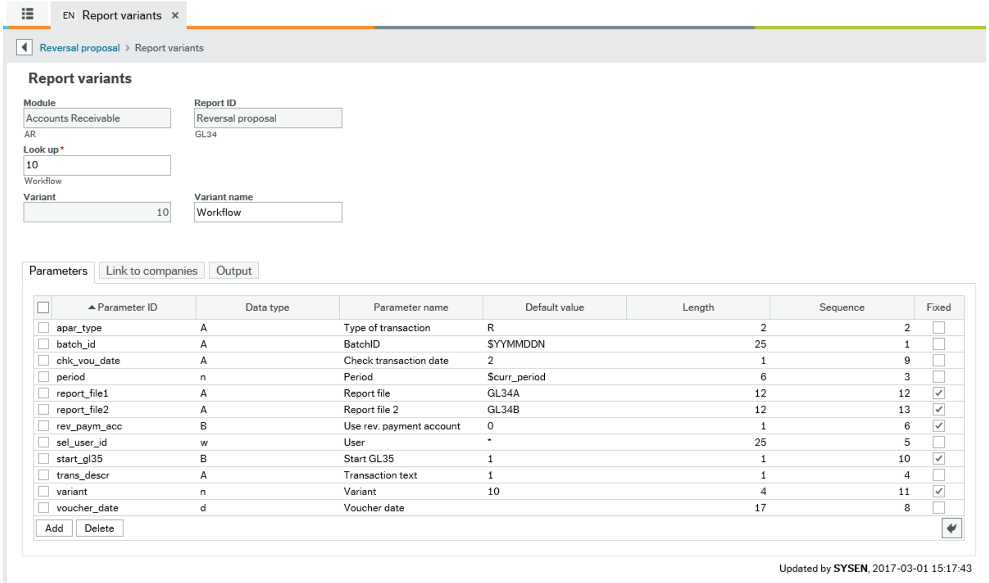
Variant: 10

Variant name: Workflow

□	▲Parameter ID	Data type	Parameter name	Default value	Length	Sequence	Fixed
□	batch_id	A	BatchID	$YYMMDDN	25	1	□
□	chk_vou_date	A	Check transaction date	2	1	6	□
□	report_file1	A	Report file	GL35A	12	4	☑
□	report_file2	A	Report file 2	GL35B	12	5	☑
□	start_tps	B	Start AGRTPS	0	1	3	☑
□	voucher_type	A	Transaction type	RR	25	2	□

Updated by **SYSEN**, 2017-03-01 15:16:54

11. Under **Accounting** > **Accounts Receivable** > **Reversal** > **Reversal proposal**, create a new report variant for the process. Make sure that you start the GL35 process with the correct variant:



Report variants

Module: Accounts Receivable
AR

Report ID: Reversal proposal
GL34

Look up *: 10
Workflow

Variant: 10

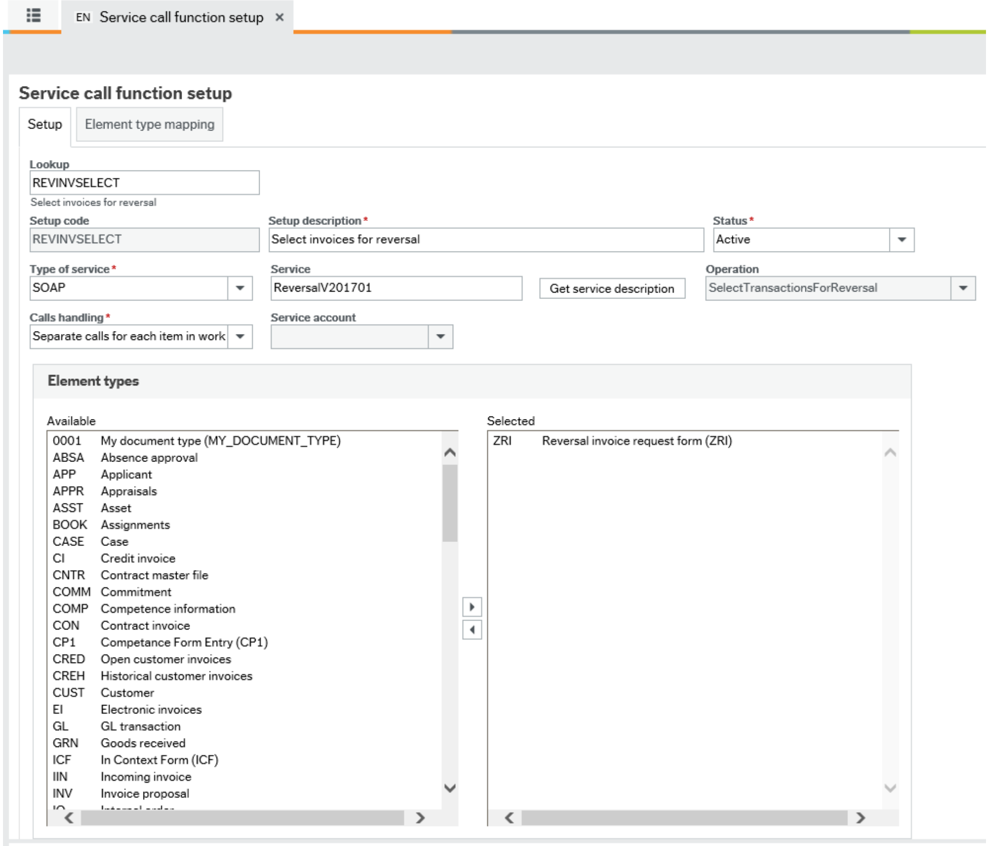
Variant name: Workflow

□	▲Parameter ID	Data type	Parameter name	Default value	Length	Sequence	Fixed
□	apar_type	A	Type of transaction	R	2	2	□
□	batch_id	A	BatchID	$YYMMDDN	25	1	□
□	chk_vou_date	A	Check transaction date	2	1	9	□
□	period	n	Period	$curr_period	6	3	□
□	report_file1	A	Report file	GL34A	12	12	☑
□	report_file2	A	Report file 2	GL34B	12	13	☑
□	rev_paym_acc	B	Use rev. payment account	0	1	6	☑
□	sel_user_id	w	User	*	25	5	□
□	start_gl35	B	Start GL35	1	1	10	☑
□	trans_descr	A	Transaction text	1	1	4	□
□	variant	n	Variant	10	4	11	☑
□	voucher_date	d	Voucher date		17	8	□

Updated by **SYSEN**, 2017-03-01 15:17:43

Setting up the service call

1. In the **Service call function setup** (TWF260) window, create the **REVINVSELECT** setup. Connect it with the **Reversal invoice request form** element type and the **Reversal** web service:



EN Service call function setup

Service call function setup

Setup | Element type mapping

Lookup: REVINVSELECT
Select invoices for reversal

Setup code: REVINVSELECT

Setup description*: Select invoices for reversal

Status*: Active

Type of service*: SOAP

Service: ReversalV201701

Get service description

Operation: SelectTransactionsForReversal

Calls handling*: Separate calls for each item in work

Service account:

Element types

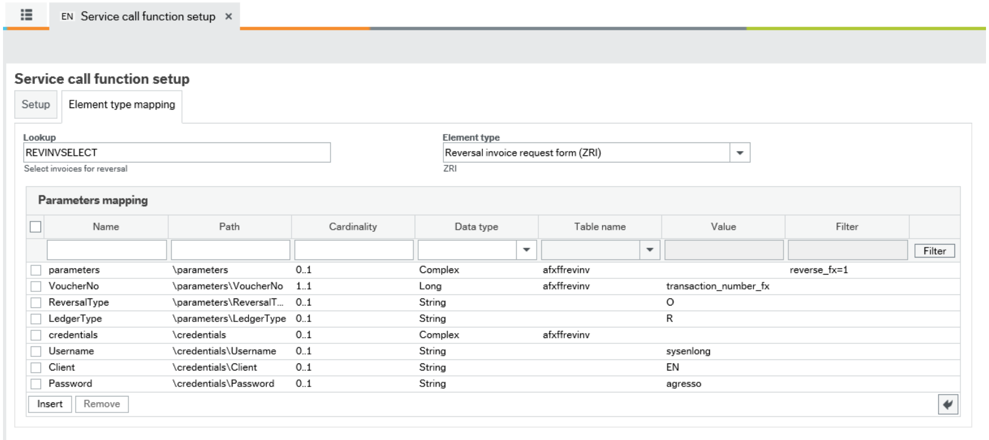
Available

0001	My document type (MY_DOCUMENT_TYPE)
ABSA	Absence approval
APP	Applicant
APPR	Appraisals
ASST	Asset
BOOK	Assignments
CASE	Case
CI	Credit invoice
CNTR	Contract master file
COMM	Commitment
COMP	Competence information
CON	Contract invoice
CP1	Competance Form Entry (CP1)
CRED	Open customer invoices
CREH	Historical customer invoices
CUST	Customer
EI	Electronic invoices
GL	GL transaction
GRN	Goods received
ICF	In Context Form (ICF)
IIN	Incoming invoice
INV	Invoice proposal
IO	Internal order

Selected

ZRI	Reversal invoice request form (ZRI)

2. On the **Element type mapping** tab, set up mapping for the parameters:



EN Service call function setup

Service call function setup

Setup | Element type mapping

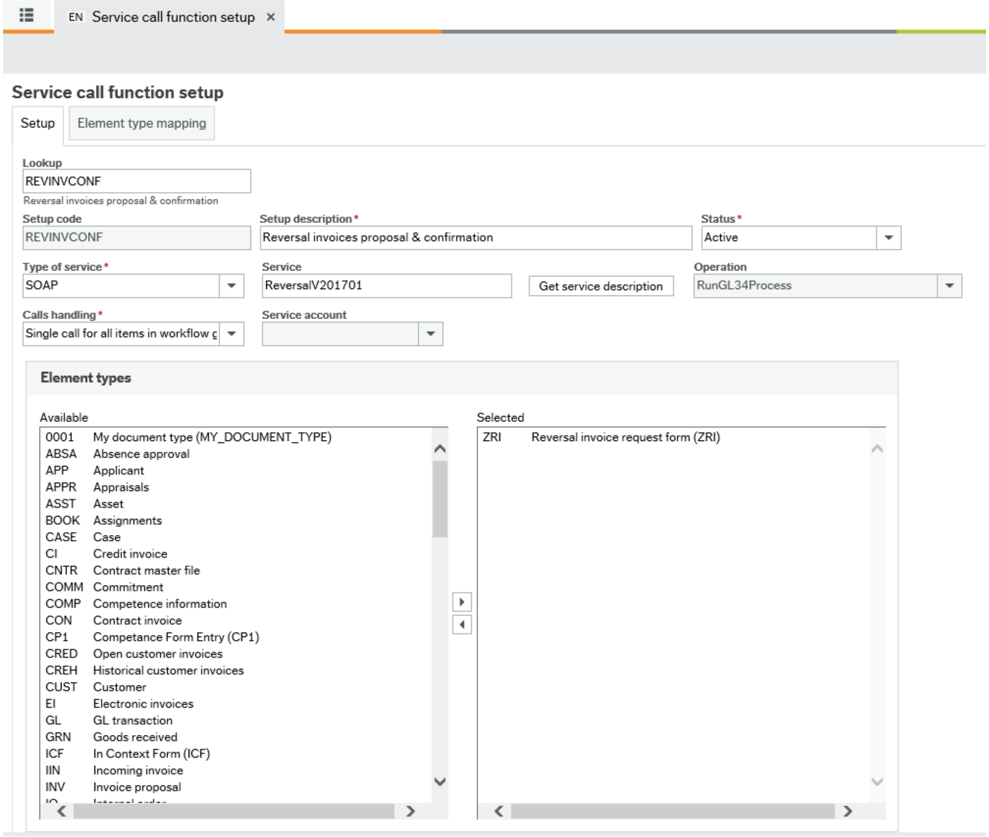
Lookup: REVINVSELECT
Select invoices for reversal

Element type: Reversal invoice request form (ZRI)
ZRI

Parameters mapping

□	Name	Path	Cardinality	Data type	Table name	Value	Filter
□	parameters	\parameters	0..1	Complex	afxffrevinv		reverse_fx=1
□	VoucherNo	\parameters\VoucherNo	1..1	Long	afxffrevinv	transaction_number_fx	
□	ReversalType	\parameters\ReversalT...	0..1	String		O	
□	LedgerType	\parameters\LedgerType	0..1	String		R	
□	credentials	\credentials	0..1	Complex	afxffrevinv		
□	Username	\credentials\Username	0..1	String		sysenlong	
□	Client	\credentials\Client	0..1	String		EN	
□	Password	\credentials\Password	0..1	String		agresso	

3. In the **Service call function setup** (TWF260) window, create the **REVINVCONF** setup. Connect it with the **Reversal invoice request form** element type and the **Reversal** web service:



EN Service call function setup ×

Service call function setup

Setup | Element type mapping

Lookup: REVINVCONF
Reversal invoices proposal & confirmation

Setup code: REVINVCONF

Setup description *: Reversal invoices proposal & confirmation

Status *: Active

Type of service *: SOAP

Service: ReversalV201701

Get service description

Operation: RunGL34Process

Calls handling *: Single call for all items in workflow g

Service account:

Element types

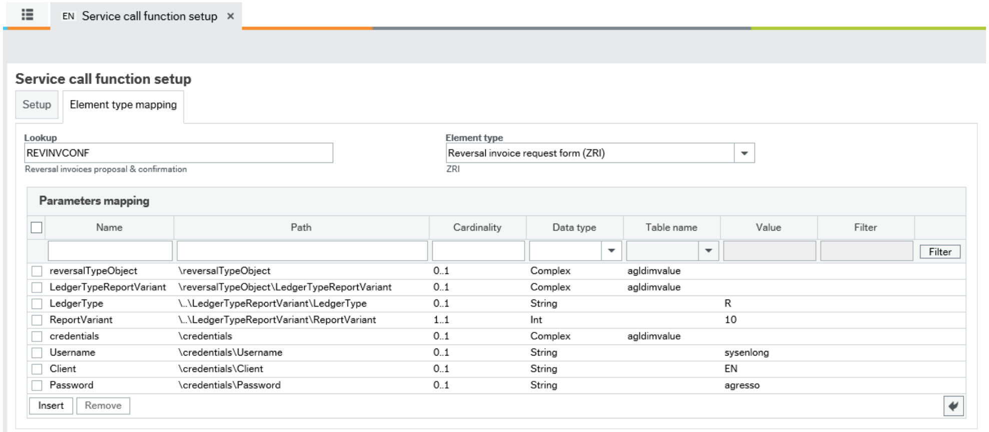
Available

0001	My document type (MY_DOCUMENT_TYPE)
ABSA	Absence approval
APP	Applicant
APPR	Appraisals
ASST	Asset
BOOK	Assignments
CASE	Case
CI	Credit invoice
CNTR	Contract master file
COMM	Commitment
COMP	Competence information
CON	Contract invoice
CP1	Competance Form Entry (CP1)
CRED	Open customer invoices
CREH	Historical customer invoices
CUST	Customer
EI	Electronic invoices
GL	GL transaction
GRN	Goods received
ICF	In Context Form (ICF)
IIN	Incoming invoice
INV	Invoice proposal
IO	Internal order

Selected

ZRI	Reversal invoice request form (ZRI)

4. On the **Element type mapping** tab, set up mapping for the parameters. Set up the same report variant (10) for the **Run34GLProcess** operation:



EN Service call function setup ×

Service call function setup

Setup | Element type mapping

Lookup: REVINVCONF
Reversal invoices proposal & confirmation

Element type: Reversal invoice request form (ZRI)
ZRI

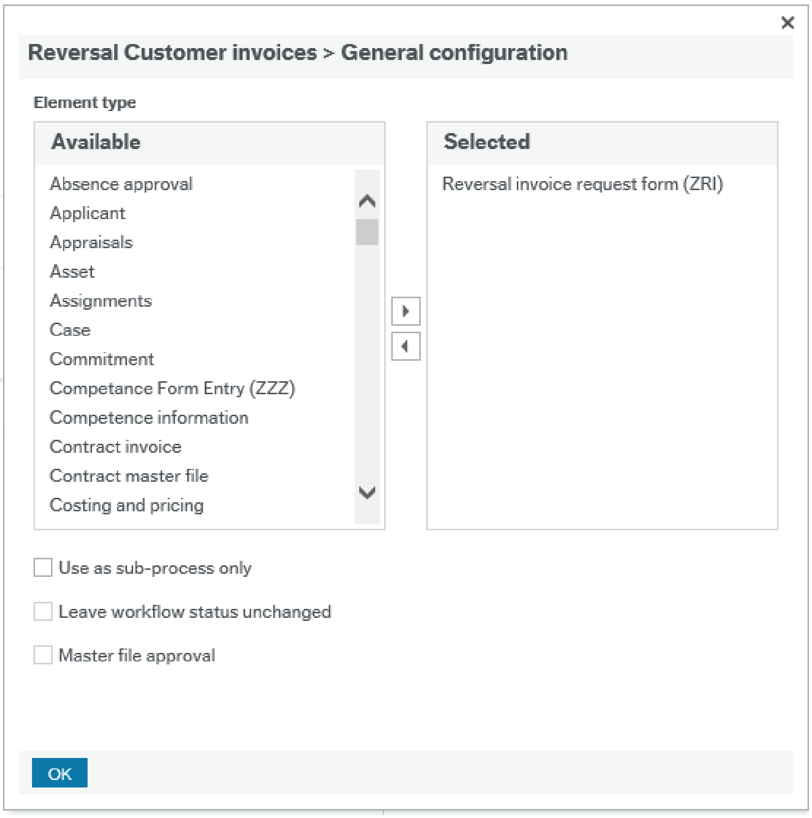
Parameters mapping

□	Name	Path	Cardinality	Data type	Table name	Value	Filter
□	reversalTypeObject	\reversalTypeObject	0..1	Complex	agldimvalue		
□	LedgerTypeReportVariant	\reversalTypeObject\LedgerTypeReportVariant	0..1	Complex	agldimvalue		
□	LedgerType	\..\LedgerTypeReportVariant\LedgerType	0..1	String		R	
□	ReportVariant	\..\LedgerTypeReportVariant\ReportVariant	1..1	Int		10	
□	credentials	\credentials	0..1	Complex	agldimvalue		
□	Username	\credentials\Username	0..1	String		sysenlong	
□	Client	\credentials\Client	0..1	String		EN	
□	Password	\credentials\Password	0..1	String		agresso	

Insert | Remove

Setting up the workflow

1. In the **Process definition** (XWF100) window, create a new process, **Reversal Customer invoices**. Select the **Reversal invoice request form (ZRI)** element type for it:



Reversal Customer invoices > General configuration

Element type

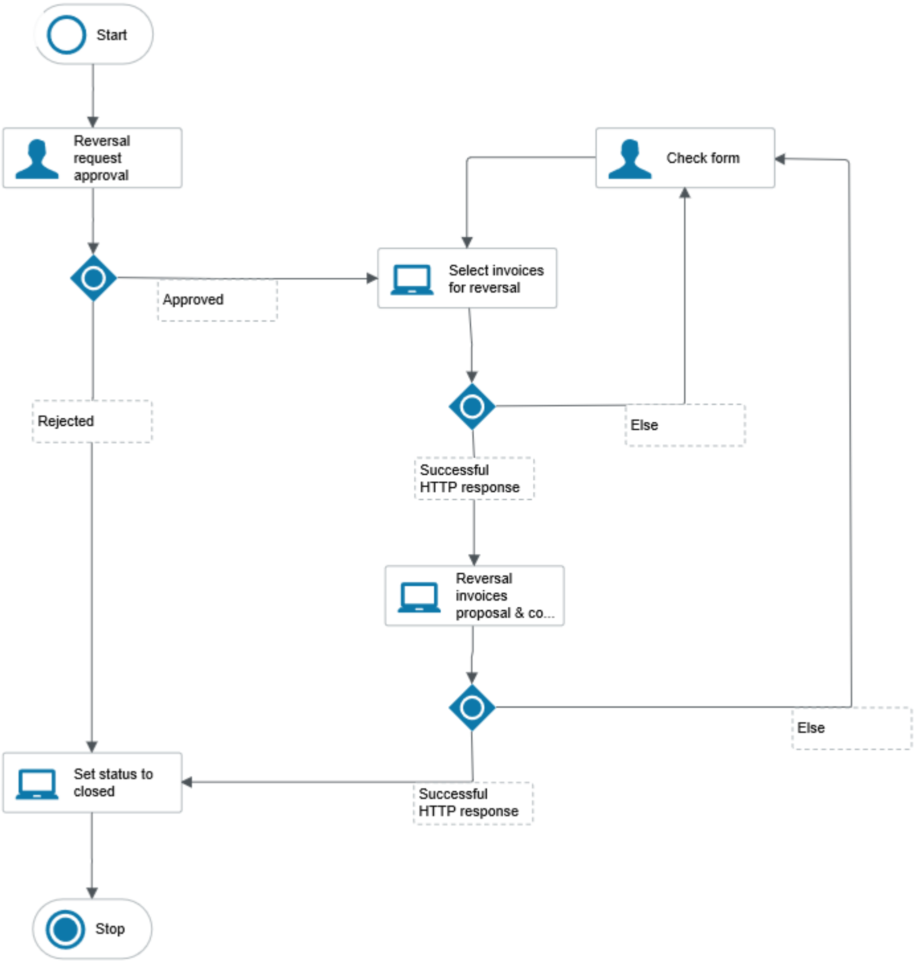
Available	Selected
Absence approval Applicant Appraisals Asset Assignments Case Commitment Competance Form Entry (ZZZ) Competence information Contract invoice Contract master file Costing and pricing	Reversal invoice request form (ZRI)

□ Use as sub-process only

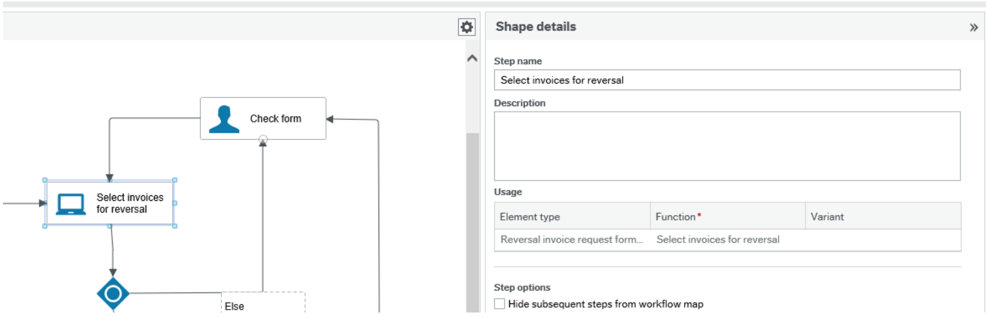
□ Leave workflow status unchanged

□ Master file approval

2. In the **Process definition** (XWF100) window, create the workflow for invoice reversal:



3. For the **Select invoices for reversal** system step, indicate the corresponding service call setup in the *Function* column:



Shape details

Step name

Select invoices for reversal

Description

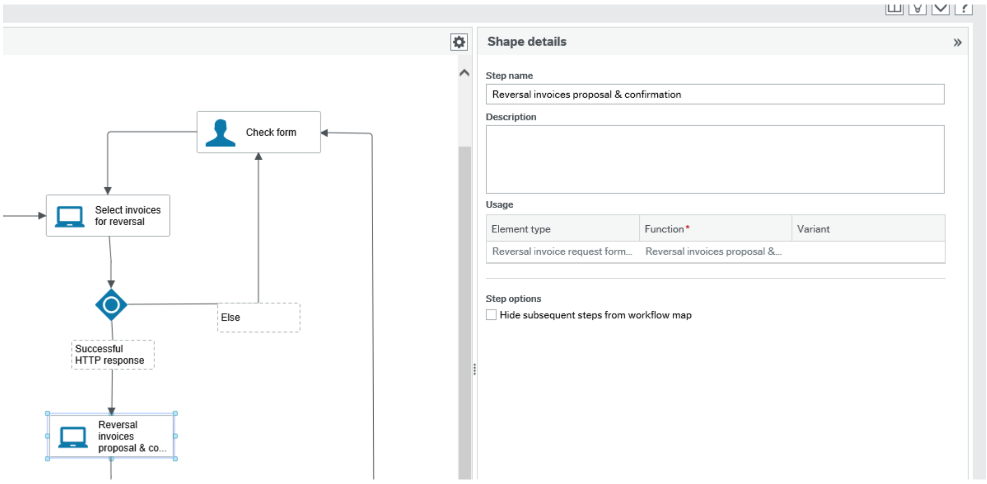
Usage

Element type	Function*	Variant
Reversal invoice request form...	Select invoices for reversal	

Step options

□ Hide subsequent steps from workflow map

4. For the **Reversal invoices proposal & confirmation** system step, indicate the corresponding service call setup in the *Function* column:


Check form
Select invoices for reversal
Else
Successful HTTP response
Reversal invoices proposal & co...

Shape details

Step name

Reversal invoices proposal & confirmation

Description

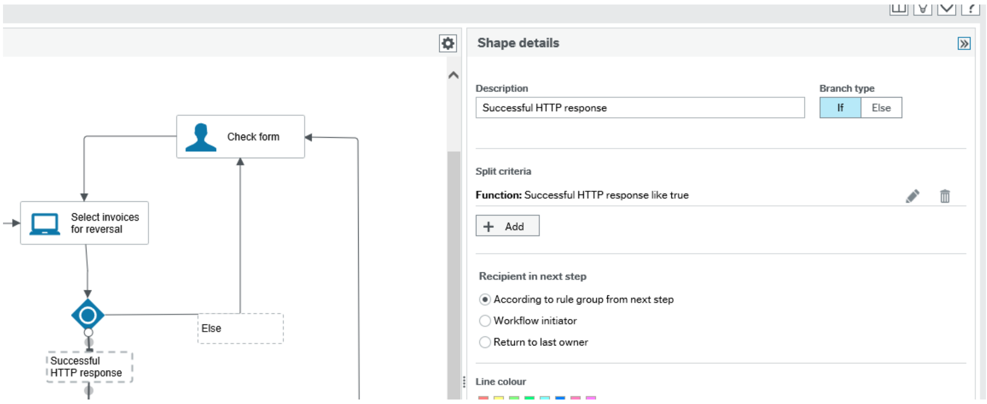
Usage

Element type	Function*	Variant
Reversal invoice request form...	Reversal invoices proposal &...	

Step options

☐ Hide subsequent steps from workflow map

5. In the OR-split, use the **Successful HTTP response** function:



Shape details

Description: Successful HTTP response

Branch type: If | Else

Split criteria

Function: Successful HTTP response like true

+ Add

Recipient in next step

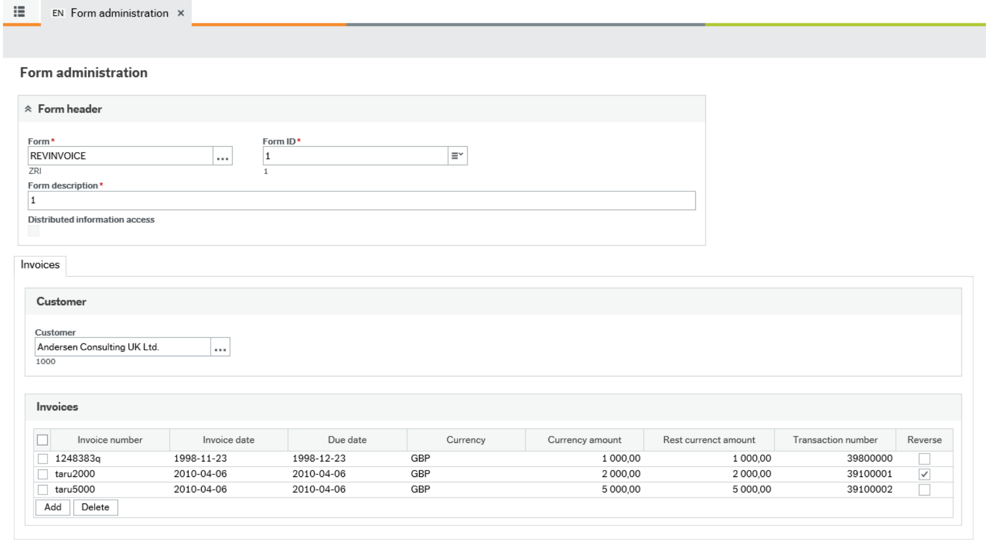
● According to rule group from next step

○ Workflow initiator

○ Return to last owner

Line colour

6. To verify your setup, go to the **Form administration** (TCR022) window. Once you enter a customer ID and tab out, the system gets all invoices with status **N**. In the *Reverse* column, select the check boxes next to the invoices you want to reverse:



Form administration

Form header

Form *: REVINVOICE
ZRI

Form ID *: 1
1

Form description *: 1

Distributed information access: □

Customer

Customer: Andersen Consulting UK Ltd.
1000

Invoices

□	Invoice number	Invoice date	Due date	Currency	Currency amount	Rest current amount	Transaction number	Reverse
□	1248383q	1998-11-23	1998-12-23	GBP	1 000,00	1 000,00	39800000	□
□	taru2000	2010-04-06	2010-04-06	GBP	2 000,00	2 000,00	39100001	☑
□	taru5000	2010-04-06	2010-04-06	GBP	5 000,00	5 000,00	39100002	□

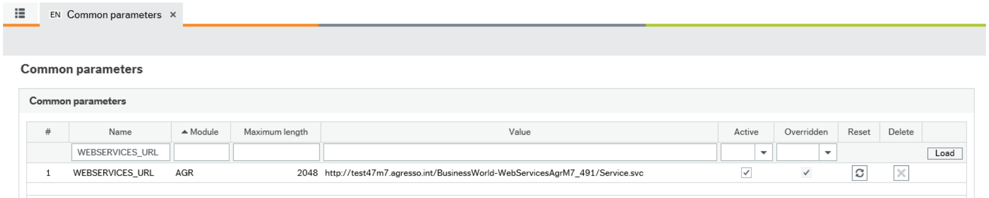
7. Submit the form. Follow the worklow to have the required invoices reversed.

Scenario 4: Partial reversal of GL transaction

In this scenario, you will reverse a part of a General Ledger transaction using a form for triggering the Reversal web service. Please note that some procedures have already been described in the previous scenarios, and can be referred to, if more details are needed.

Creating attributes and flexi-field groups

1. In the **Common parameters** (TAG023) window, set up and activate the WEBSERVICES_URL common parameter:

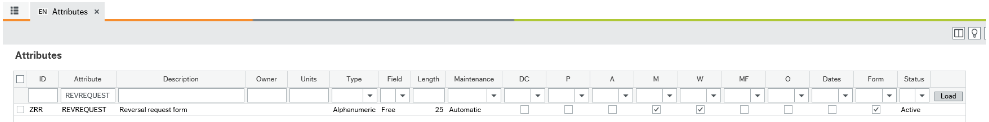


Common parameters

Common parameters

#	Name	Module	Maximum length	Value	Active	Overridden	Reset	Delete	
	WEBSERVICES_URL								Load
1	WEBSERVICES_URL	AGR	2048	http://test47m7.agresso.int/BusinessWorld-WebServicesAgrM7_491/Service.svc	☑	✓		☒	

2. Go to the **Attributes** (TGL004) window. Click **Add** and create the **ZRR** REVREQUEST attribute:

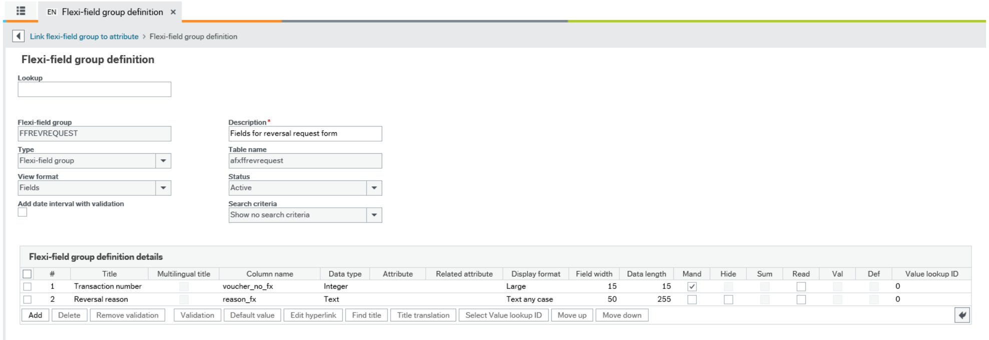


Attributes

☐	ID	Attribute	Description	Owner	Units	Type	Field	Length	Maintenance	DC	P	A	M	W	MF	O	Dates	Form	Status	
		REVREQUEST																		Load
☐	ZRR	REVREQUEST	Reversal request form			Alphanumeric	Free	25	Automatic	☐	☐	☐	☑	☑	☐	☐	☐	☑	Active	

> **Note:** Make sure *Form* and *W* (Workflow) check boxes are selected for the attributes.

3. In the **Flexi-field group definition** (TAG060) window, create the FFREVREQUEST flexi-field group:


EN Flexi-field group definition ×
Link flexi-field group to attribute > Flexi-field group definition

Flexi-field group definition

Lookup

Flexi-field group: FFREVREQUEST

Description*: Fields for reversal request form

Type: Flexi-field group

Table name: afxffrevrequest

View format: Fields

Status: Active

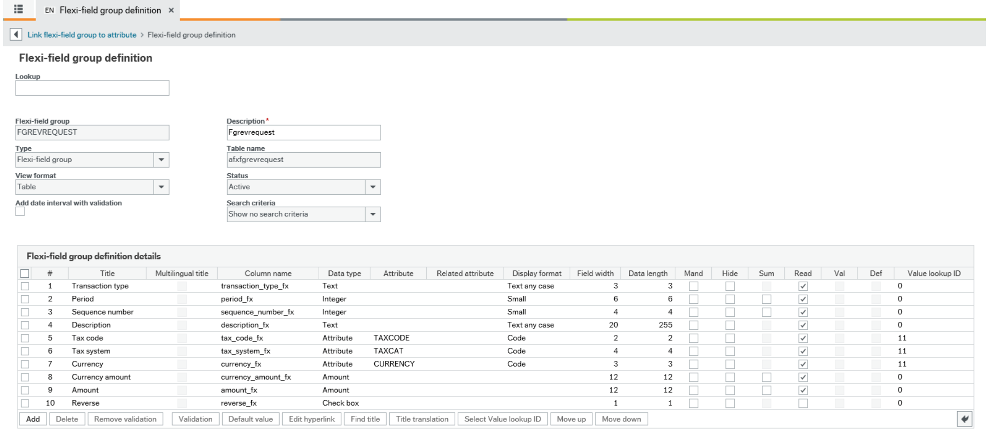
Add date interval with validation ☐

Search criteria: Show no search criteria

Flexi-field group definition details

☐	#	Title	Multilingual title	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup ID
☐	1	Transaction number		voucher_no_fx	Integer			Large	15	15	☑			☐			0
☐	2	Reversal reason		reason_fx	Text			Text any case	50	255	☐	☐		☐			0

4. In the **Flexi-field group definition** (TAG060) window, create the FGREVREQUEST flexi-field group:



Flexi-field group definition

Lookup:

Flexi-field group: FGREVREQUEST

Description*: Fgrevrequest

Type: Flexi-field group

Table name: afxfgrevrequest

View format: Table

Status: Active

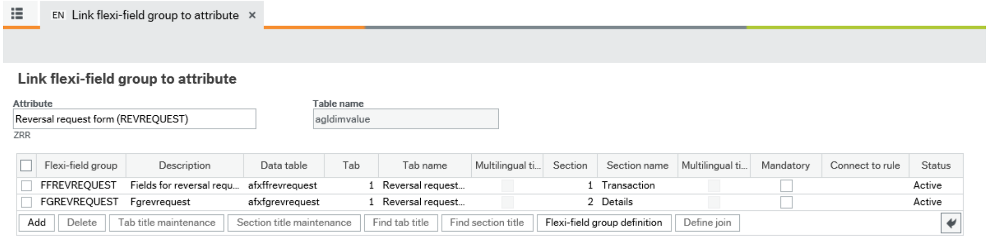
Add date interval with validation: □

Search criteria: Show no search criteria

Flexi-field group definition details

□	#	Title	Multilingual title	Column name	Data type	Attribute	Related attribute	Display format	Field width	Data length	Mand	Hide	Sum	Read	Val	Def	Value lookup ID
□	1	Transaction type		transaction_type_fx	Text			Text any case	3	3	□	□		☑			0
□	2	Period		period_fx	Integer			Small	6	6	□	□	□	☑			0
□	3	Sequence number		sequence_number_fx	Integer			Small	4	4	□	□	□	☑			0
□	4	Description		description_fx	Text			Text any case	20	255	□	□		☑			0
□	5	Tax code		tax_code_fx	Attribute	TAXCODE		Code	2	2	□	□		☑			11
□	6	Tax system		tax_system_fx	Attribute	TAXCAT		Code	4	4	□	□		☑			11
□	7	Currency		currency_fx	Attribute	CURRENCY		Code	3	3	□	□		☑			11
□	8	Currency amount		currency_amount_fx	Amount				12	12	□	□	□	☑			0
□	9	Amount		amount_fx	Amount				12	12	□	□	□	☑			0
□	10	Reverse		reverse_fx	Check box				1	1	□	□		□			0

5. In the **Link flexi-field group to attribute** (TCR037) window, link the Reversal request form (REVREQUEST) attribute to the flexi-field groups you created in previous steps:


EN Link flexi-field group to attribute ×
Add
Delete
Tab title maintenance
Section title maintenance
Find tab title
Find section title
Flexi-field group definition
Define join

Link flexi-field group to attribute

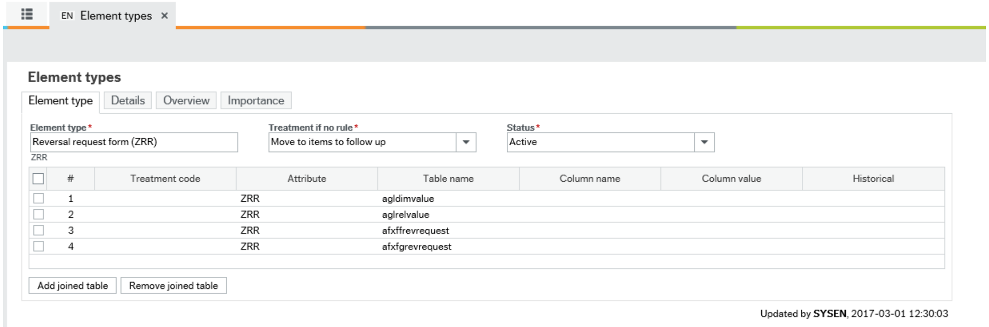
Attribute: Reversal request form (REVREQUEST)
ZRR

Table name: agldimvalue

□	Flexi-field group	Description	Data table	Tab	Tab name	Multilingual ti...	Section	Section name	Multilingual ti...	Mandatory	Connect to rule	Status
□	FFREVREQUEST	Fields for reversal requ...	afxffrevrequest	1	Reversal request...		1	Transaction		□		Active
□	FGREVREQUEST	Fgrevrequest	afxfgrevrequest	1	Reversal request...		2	Details		□		Active

> **Note:** Detailed information on working with flexi-fields can be found in Business World 7.0 **Flexi-fields** Reference manual.

6. In the **Element types** (TWF032) window, create the element type based on the **ZRR** attribute and linked flexi-field groups:



Element types

Element type*: Reversal request form (ZRR)
ZRR

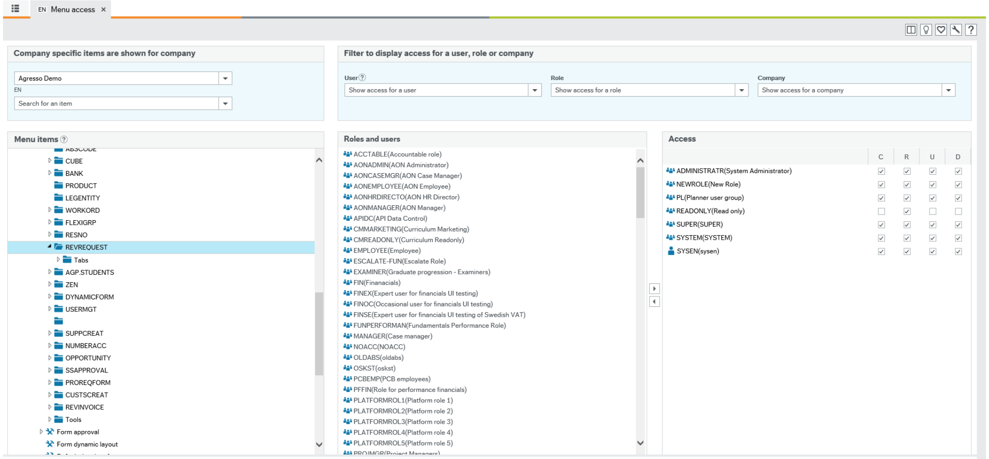
Treatment if no rule*: Move to items to follow up

Status*: Active

□	#	Treatment code	Attribute	Table name	Column name	Column value	Historical
□	1		ZRR	agldimvalue			
□	2		ZRR	aglrelvalue			
□	3		ZRR	afxffrevrequest			
□	4		ZRR	afxfgrevrequest			

Updated by **SYSEN**, 2017-03-01 12:30:03

7. In the **Menu access** (XAG003) window, grant access rights to the form:



EN Menu access

Company specific items are shown for company

Agresso Demo

EN

Search for an item

Filter to display access for a user, role or company

User

Show access for a user

Role

Show access for a role

Company

Show access for a company

Menu items

- ABSCODE
- CUBE
- BANK
- PRODUCT
- LEGENTITY
- WORKORD
- FLEXIGRP
- RESNO
- REVREQUEST
  - Tabs
- AGP.STUDENTS
- ZEN
- DYNAMICFORM
- USERMGT
- SUPPCREAT
- NUMBERACC
- OPPORTUNITY
- SSAPPROVAL
- PROREQFORM
- CUSTSCREAT
- REVINVOICE
- Tools
- Form approval
- Form dynamic layout

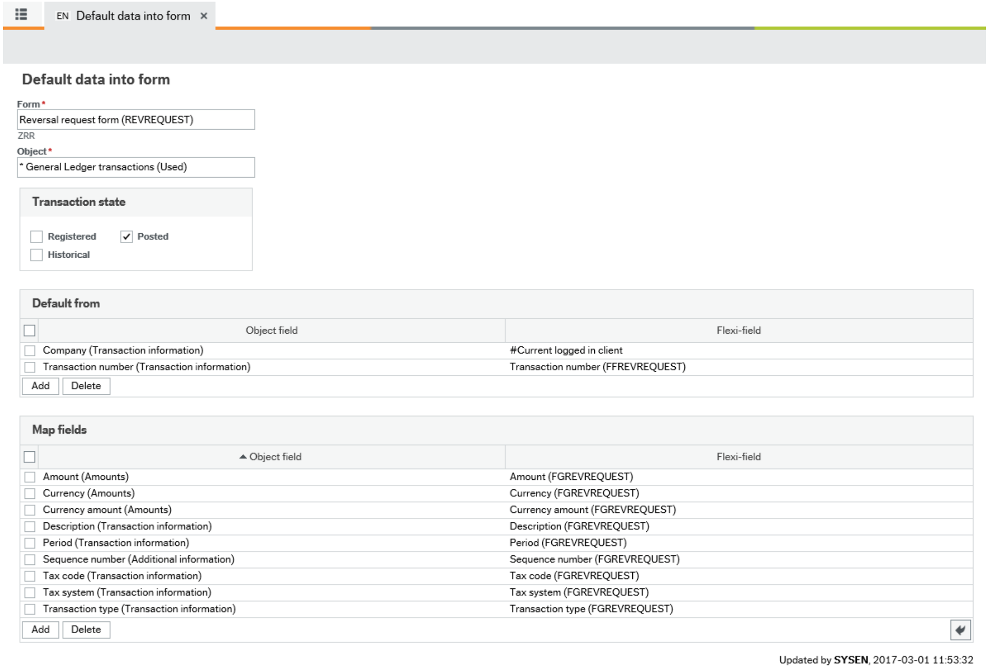
Roles and users

ACCTABLE(Accountable role)
AONADMIN(AON Administrator)
AONCASEMGR(AON Case Manager)
AONEMPLOYEE(AON Employee)
AONHRDIRECTO(AON HR Director)
AONMANAGER(AON Manager)
APIDC(API Data Control)
CMMARKETING(Curriculum Marketing)
CMREADONLY(Curriculum Readonly)
EMPLOYEE(Employee)
ESCALATE-FUN(Escalate Role)
EXAMINER(Graduate progression - Examiners)
FIN(Finanacials)
FINEX(Expert user for financials UI testing)
FINOC(Occasional user for financials UI testing)
FINSE(Expert user for financials UI testing of Swedish VAT)
FUNPERFORMAN(Fundamentals Performance Role)
MANAGER(Case manager)
NOACC(NOACC)
OLDABS(oldabs)
OSKST(oskst)
PCBEMP(PCB employees)
PFFIN(Role for performance financials)
PLATFORMROL1(Platform role 1)
PLATFORMROL2(Platform role 2)
PLATFORMROL3(Platform role 3)
PLATFORMROL4(Platform role 4)
PLATFORMROL5(Platform role 5)
PROJMGR(Project Managers)

Access

	C	R	U	D
ADMINISTRATR(System Administrator)	☑	☑	☑	☑
NEWROLE(New Role)	☑	☑	☑	☑
PL(Planner user group)	☑	☑	☑	☑
READONLY(Read only)	□	☑	□	□
SUPER(SUPER)	☑	☑	☑	☑
SYSTEM(SYSTEM)	☑	☑	☑	☑
SYSEN(sysen)	☑	☑	☑	☑

8. In the **Default data into form** (TCR040) window, create relationship between the **General Ledger transactions** object and the form:



EN Default data into form

Default data into form

Form *

Reversal request form (REVREQUEST)

ZRR

Object *

* General Ledger transactions (Used)

Transaction state

□ Registered ☑ Posted

□ Historical

Default from

□	Object field	Flexi-field
□	Company (Transaction information)	#Current logged in client
□	Transaction number (Transaction information)	Transaction number (FFREVREQUEST)

Add Delete

Map fields

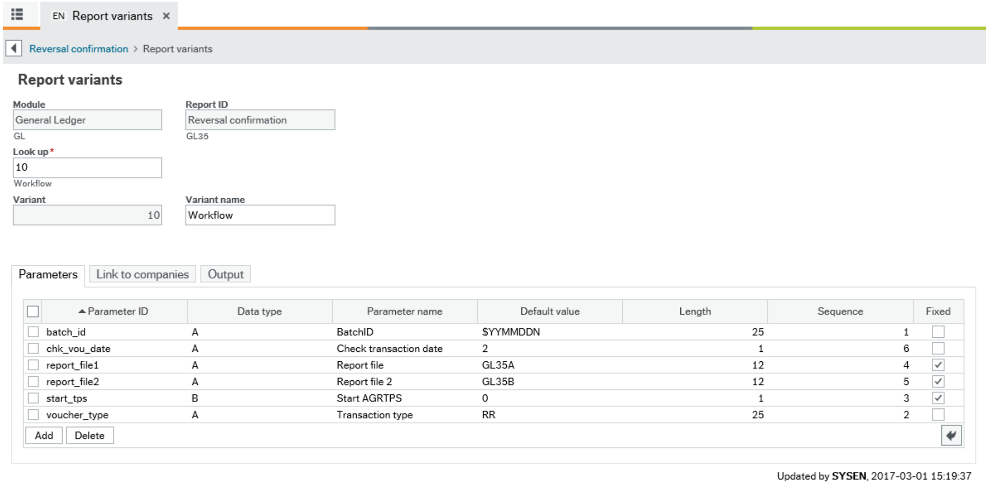
□	Object field	Flexi-field
□	Amount (Amounts)	Amount (FGREVREQUEST)
□	Currency (Amounts)	Currency (FGREVREQUEST)
□	Currency amount (Amounts)	Currency amount (FGREVREQUEST)
□	Description (Transaction information)	Description (FGREVREQUEST)
□	Period (Transaction information)	Period (FGREVREQUEST)
□	Sequence number (Additional information)	Sequence number (FGREVREQUEST)
□	Tax code (Transaction information)	Tax code (FGREVREQUEST)
□	Tax system (Transaction information)	Tax system (FGREVREQUEST)
□	Transaction type (Transaction information)	Transaction type (FGREVREQUEST)

Add Delete

Updated by **SYSEN**, 2017-03-01 11:53:32

> **Note:** Detailed information on working with forms can be found in Business World 7.0 **Forms** Reference manual.

9. Under **Accounting** > **General Ledger** > **Reversal** > **Reversal confirmation**, create a new report variant for the process. Click the **Report variants** button and save the following report variant:



Report variants

Module: General Ledger (GL)

Report ID: Reversal confirmation (GL35)

Look up *: 10 (Workflow)

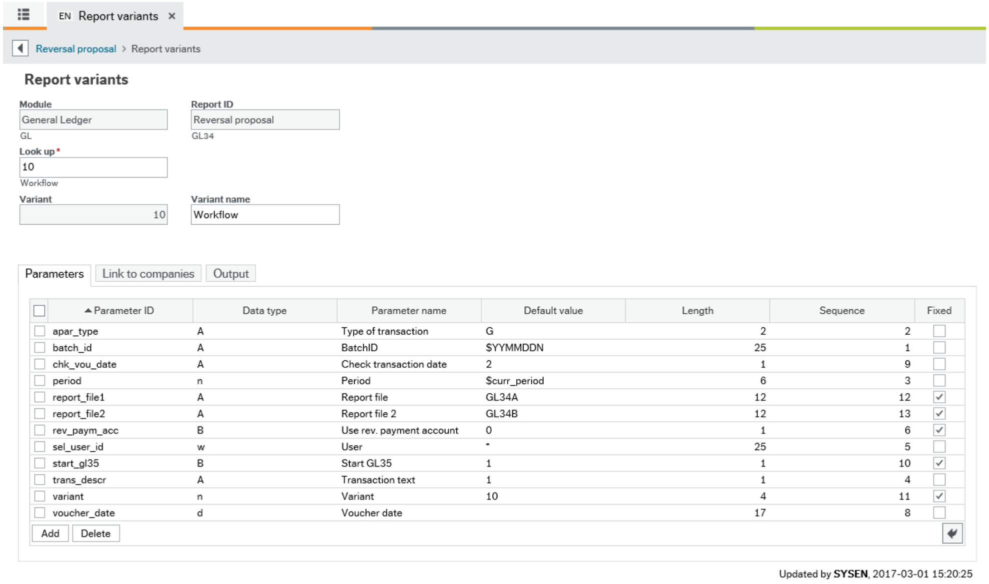
Variant: 10

Variant name: Workflow

☐	Parameter ID	Data type	Parameter name	Default value	Length	Sequence	Fixed
☐	batch_id	A	BatchID	$YYMMDDN	25	1	☐
☐	chk_vou_date	A	Check transaction date	2	1	6	☐
☐	report_file1	A	Report file	GL35A	12	4	☑
☐	report_file2	A	Report file 2	GL35B	12	5	☑
☐	start_tps	B	Start AGRTPS	0	1	3	☑
☐	voucher_type	A	Transaction type	RR	25	2	☐

Updated by **SYSEN**, 2017-03-01 15:19:37

10. Under **Accounting** > **General Ledger** > **Reversal** > **Reversal proposal**, create a new report variant for the process. Make sure that you start the GL35 process with the correct variant:



Report variants

Module: General Ledger (GL)

Report ID: Reversal proposal (GL34)

Look up *: 10 (Workflow)

Variant: 10

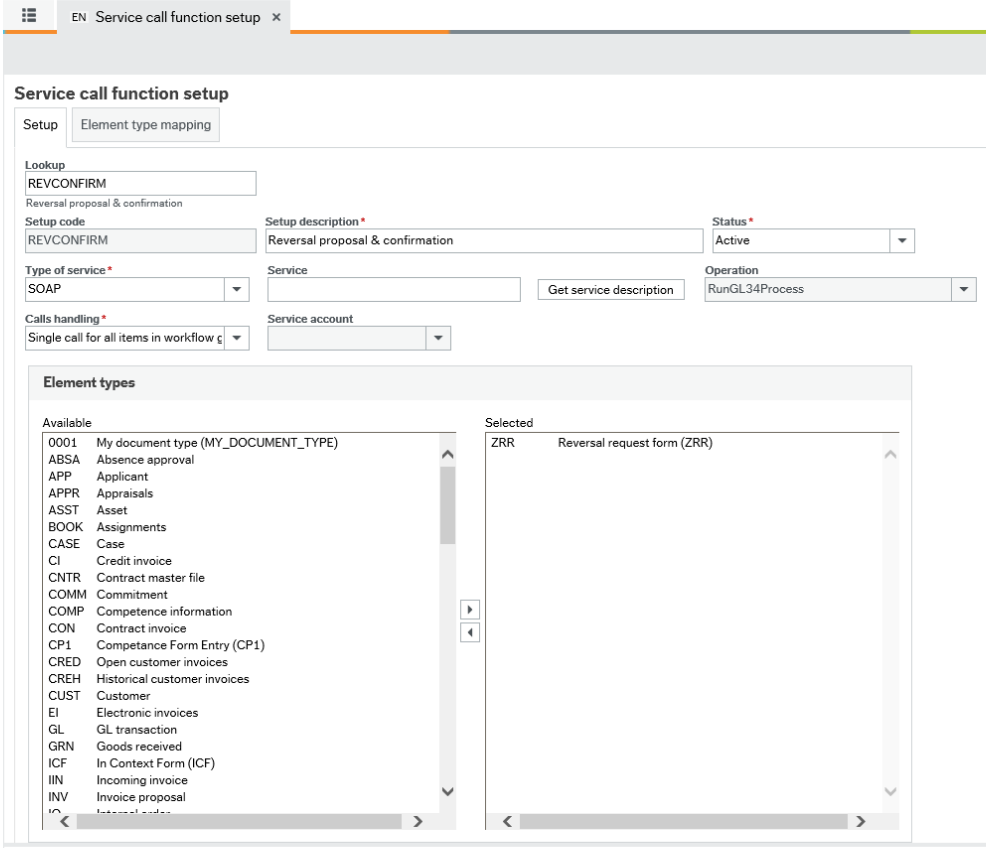
Variant name: Workflow

☐	Parameter ID	Data type	Parameter name	Default value	Length	Sequence	Fixed
☐	apar_type	A	Type of transaction	G	2	2	☐
☐	batch_id	A	BatchID	$YYMMDDN	25	1	☐
☐	chk_vou_date	A	Check transaction date	2	1	9	☐
☐	period	n	Period	$curr_period	6	3	☐
☐	report_file1	A	Report file	GL34A	12	12	☑
☐	report_file2	A	Report file 2	GL34B	12	13	☑
☐	rev_paym_acc	B	Use rev. payment account	0	1	6	☑
☐	sel_user_id	w	User	*	25	5	☐
☐	start_gl35	B	Start GL35	1	1	10	☑
☐	trans_descr	A	Transaction text	1	1	4	☐
☐	variant	n	Variant	10	4	11	☑
☐	voucher_date	d	Voucher date		17	8	☐

Updated by **SYSEN**, 2017-03-01 15:20:25

Setting up the service call

1. In the **Service call function setup** (TWF260) window, create the **REVSELECT** setup. Connect it with the **Reversal request form** element type and the **Reversal** web service:



EN Service call function setup ×

Service call function setup

Setup | Element type mapping

Lookup: REVCONFIRM

Reversal proposal & confirmation

Setup code: REVCONFIRM

Setup description*: Reversal proposal & confirmation

Status*: Active

Type of service*: SOAP

Service:

Get service description

Operation: RunGL34Process

Calls handling*: Single call for all items in workflow g

Service account:

Element types

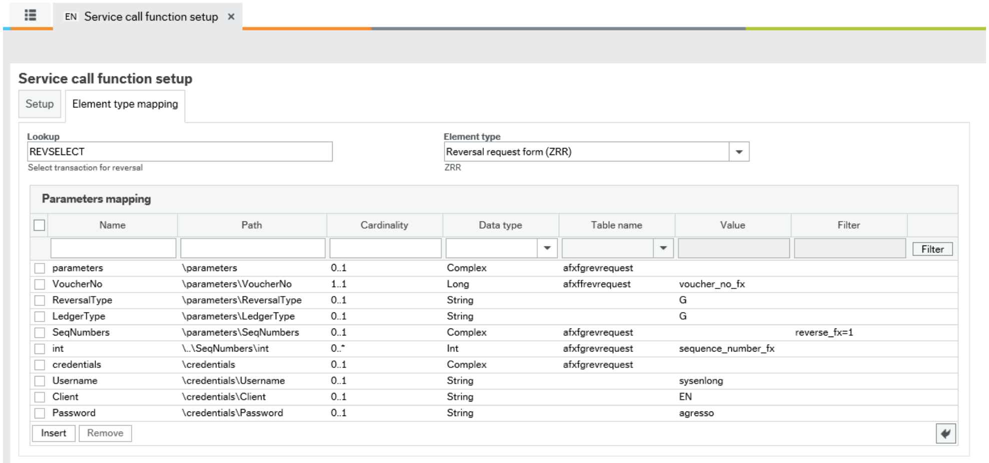
Available

0001	My document type (MY_DOCUMENT_TYPE)
ABSA	Absence approval
APP	Applicant
APPR	Appraisals
ASST	Asset
BOOK	Assignments
CASE	Case
CI	Credit invoice
CNTR	Contract master file
COMM	Commitment
COMP	Competence information
CON	Contract invoice
CP1	Competance Form Entry (CP1)
CRED	Open customer invoices
CREH	Historical customer invoices
CUST	Customer
EI	Electronic invoices
GL	GL transaction
GRN	Goods received
ICF	In Context Form (ICF)
IIN	Incoming invoice
INV	Invoice proposal
IO	Internal order

Selected

ZRR	Reversal request form (ZRR)

2. On the **Element type mapping** tab, set up mapping for the parameters:



EN Service call function setup ×

Service call function setup

Setup | Element type mapping

Lookup: REVSELECT

Select transaction for reversal

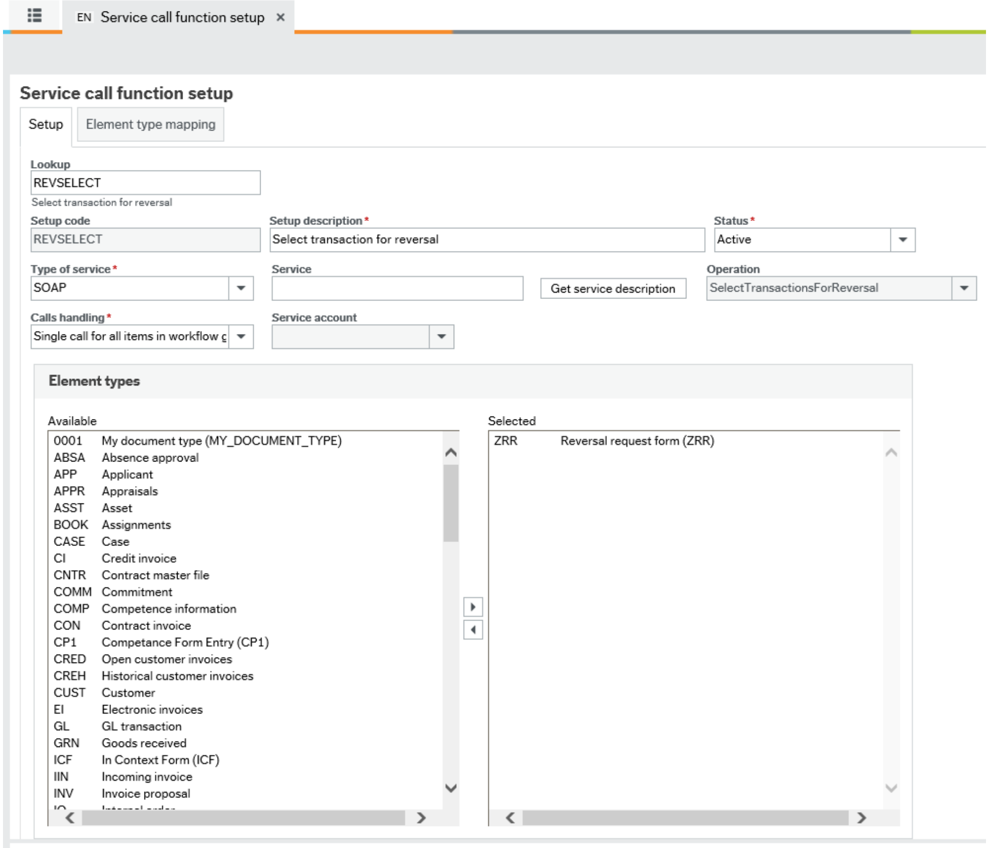
Element type: Reversal request form (ZRR)

ZRR

Parameters mapping

Name	Path	Cardinality	Data type	Table name	Value	Filter
parameters	\parameters	0..1	Complex	afxfgrevrequest		
VoucherNo	\parameters\VoucherNo	1..1	Long	afxffrevrequest	voucher_no_fx	
ReversalType	\parameters\ReversalType	0..1	String		G	
LedgerType	\parameters\LedgerType	0..1	String		G	
SeqNumbers	\parameters\SeqNumbers	0..1	Complex	afxfgrevrequest		reverse_fx=1
int	\..\SeqNumbers\int	0..*	Int	afxfgrevrequest	sequence_number_fx	
credentials	\credentials	0..1	Complex	afxfgrevrequest		
Username	\credentials\Username	0..1	String		sysenlong	
Client	\credentials\Client	0..1	String		EN	
Password	\credentials\Password	0..1	String		agresso	

3. In the **Service call function setup** (TWF260) window, create the **REVCONFIRM** setup. Connect it with the **Reversal request form** element type and the **Reversal** web service:



EN Service call function setup

Service call function setup

Setup | Element type mapping

Lookup: REVSELECT
Select transaction for reversal

Setup code: REVSELECT

Setup description *: Select transaction for reversal

Status *: Active

Type of service *: SOAP

Service:

Get service description

Operation: SelectTransactionsForReversal

Calls handling *: Single call for all items in workflow g

Service account:

Element types

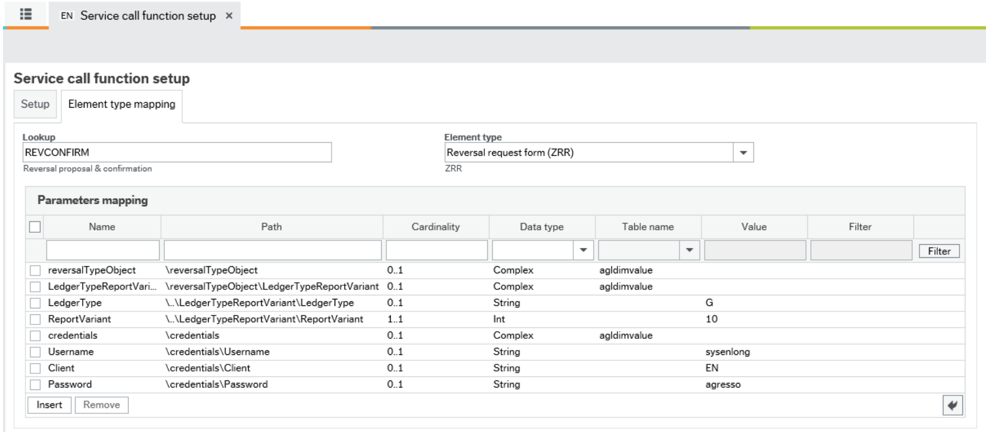
Available

0001	My document type (MY_DOCUMENT_TYPE)
ABSA	Absence approval
APP	Applicant
APPR	Appraisals
ASST	Asset
BOOK	Assignments
CASE	Case
CI	Credit invoice
CNTR	Contract master file
COMM	Commitment
COMP	Competence information
CON	Contract invoice
CP1	Competance Form Entry (CP1)
CRED	Open customer invoices
CREH	Historical customer invoices
CUST	Customer
EI	Electronic invoices
GL	GL transaction
GRN	Goods received
ICF	In Context Form (ICF)
IIN	Incoming invoice
INV	Invoice proposal

Selected

ZRR	Reversal request form (ZRR)

4. On the **Element type mapping** tab, set up mapping for the parameters. Make sure you indicate the appropriate report variant for the **Run34GLProcess** operation:



EN Service call function setup

Service call function setup

Setup | Element type mapping

Lookup: REVCONFIRM
Reversal proposal & confirmation

Element type: Reversal request form (ZRR)
ZRR

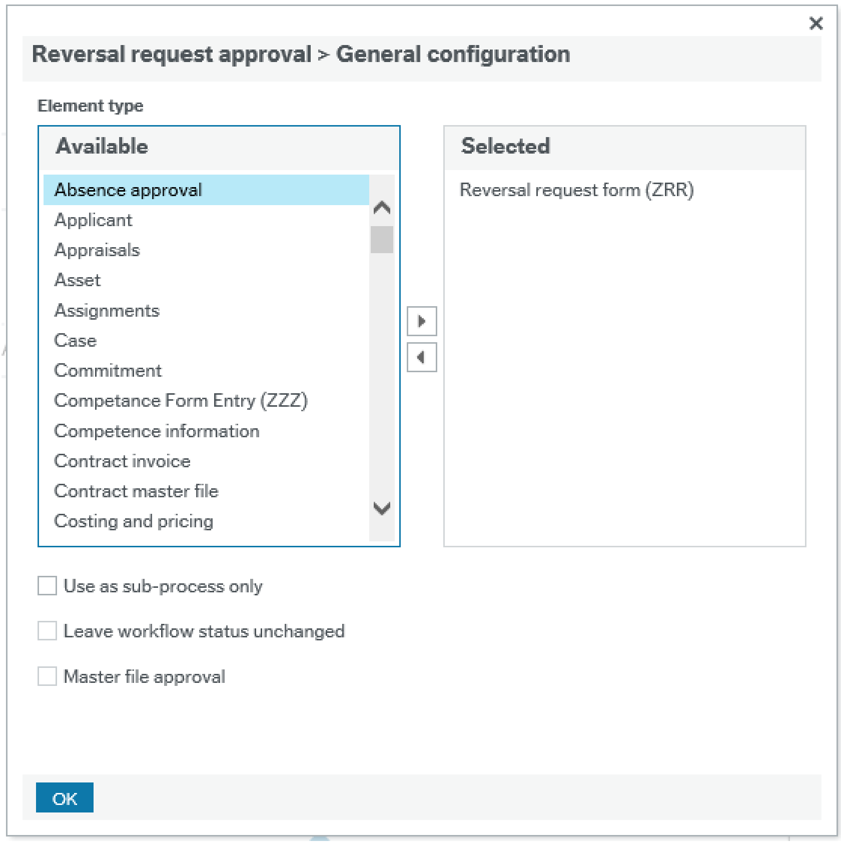
Parameters mapping

□	Name	Path	Cardinality	Data type	Table name	Value	Filter
□	reversalTypeObject	\reversalTypeObject	0..1	Complex	agldimvalue		
□	LedgerTypeReportVari...	\reversalTypeObject\LedgerTypeReportVariant	0..1	Complex	agldimvalue		
□	LedgerType	\..\LedgerTypeReportVariant\LedgerType	0..1	String		G	
□	ReportVariant	\..\LedgerTypeReportVariant\ReportVariant	1..1	Int		10	
□	credentials	\credentials	0..1	Complex	agldimvalue		
□	Username	\credentials\Username	0..1	String		sysenlong	
□	Client	\credentials\Client	0..1	String		EN	
□	Password	\credentials\Password	0..1	String		agresso	

Insert | Remove

Setting up the workflow

1. In the **Process definition** (XWF100) window, create a new process, **Reversal request approval**. Select the **Reversal request form** element type for it:



Reversal request approval > General configuration

Element type

Available

- Absence approval
- Applicant
- Appraisals
- Asset
- Assignments
- Case
- Commitment
- Competance Form Entry (ZZZ)
- Competence information
- Contract invoice
- Contract master file
- Costing and pricing

Selected

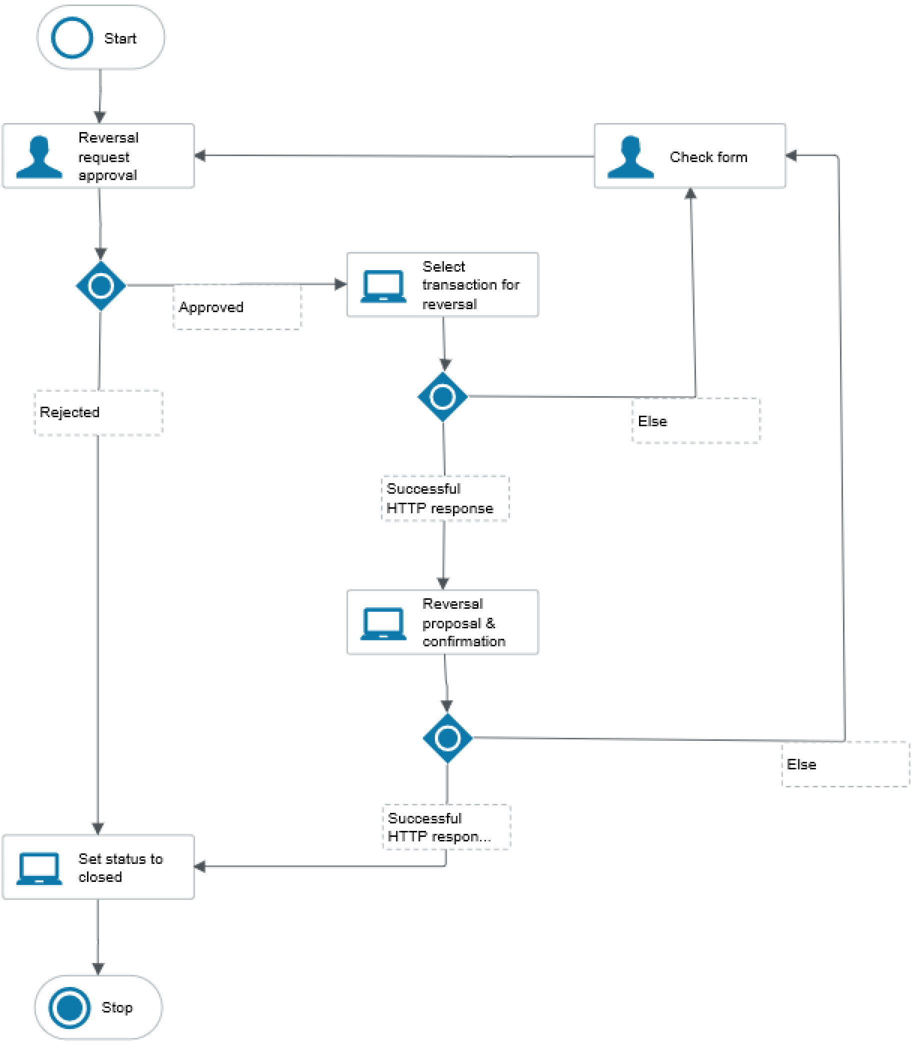
- Reversal request form (ZRR)

□ Use as sub-process only

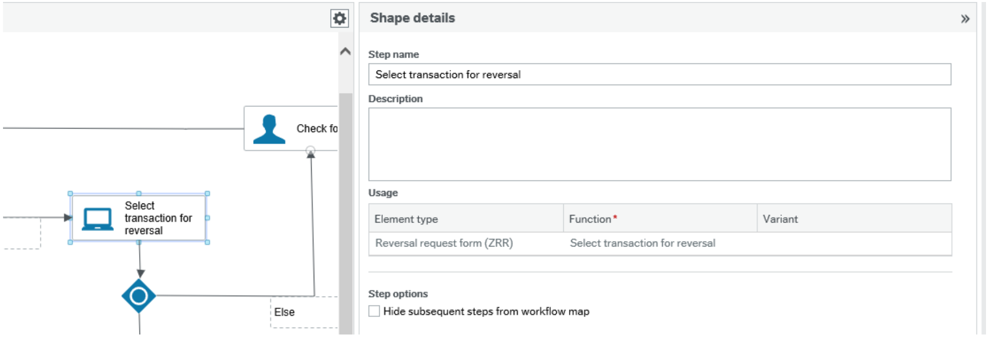
□ Leave workflow status unchanged

□ Master file approval

2. In the **Process definition** (XWF100) window, create the workflow for reversal request:


Start
Reversal request approval
Check form
Approved
Select transaction for reversal
Rejected
Else
Successful HTTP response
Reversal proposal & confirmation
Else
Successful HTTP respon...
Set status to closed
Stop

3. For the **Select transaction for reversal** system step, indicate the corresponding service call setup in the *Function* column:



Shape details

Step name

Select transaction for reversal

Description

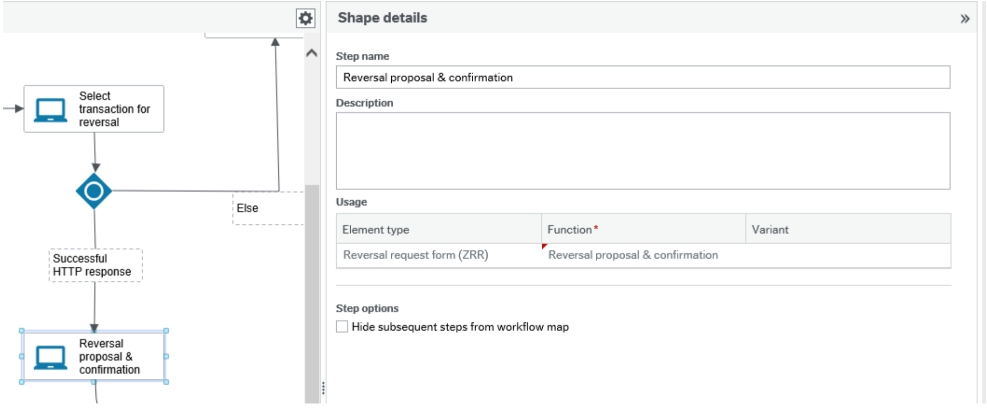
Usage

Element type	Function*	Variant
Reversal request form (ZRR)	Select transaction for reversal	

Step options

☐ Hide subsequent steps from workflow map

4. For the **Reversal proposal & confirmation** system step, indicate the corresponding service call setup in the *Function* column:



Select transaction for reversal
Else
Successful HTTP response
Reversal proposal & confirmation

Shape details

Step name: Reversal proposal & confirmation

Description:

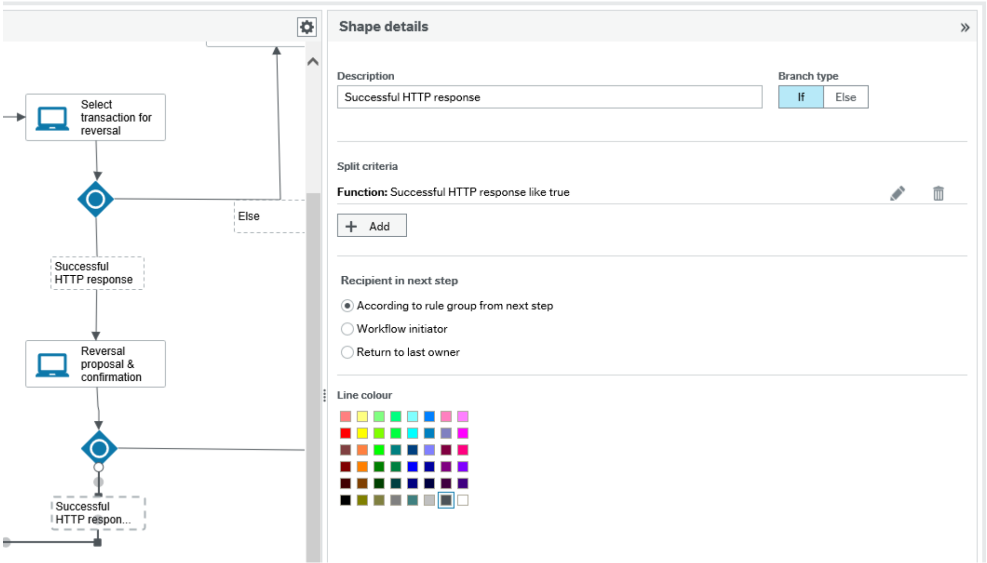
Usage

Element type	Function*	Variant
Reversal request form (ZRR)	Reversal proposal & confirmation	

Step options

□ Hide subsequent steps from workflow map

5. In the OR-split, use the **Successful HTTP response** function:



Select transaction for reversal
Else
Successful HTTP response
Reversal proposal & confirmation
Successful HTTP respon...

Shape details

Description: Successful HTTP response

Branch type: If | Else

Split criteria

Function: Successful HTTP response like true

+ Add

Recipient in next step

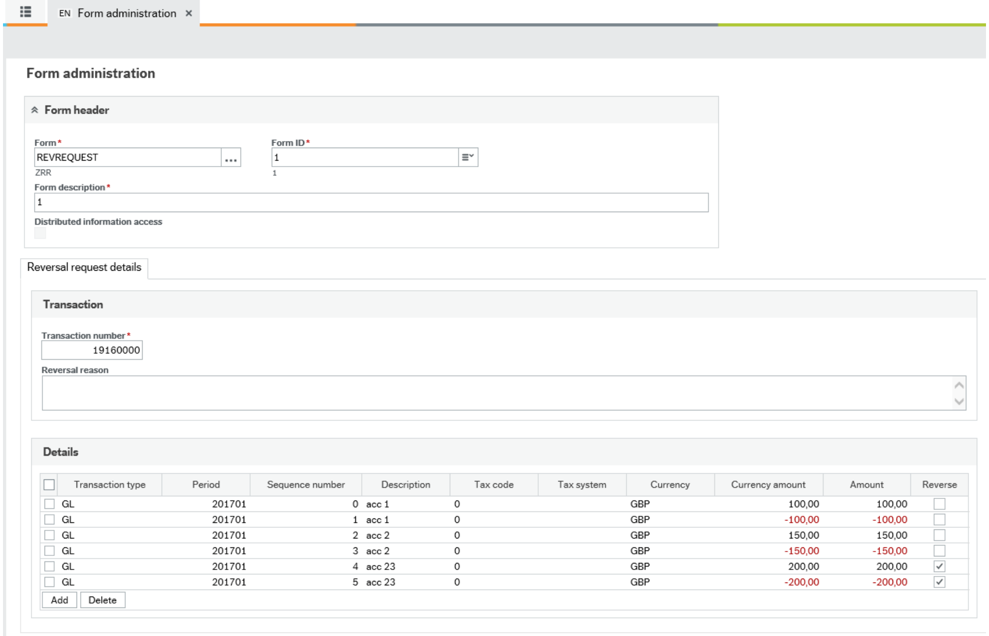
● According to rule group from next step

○ Workflow initiator

○ Return to last owner

Line colour

6. To verify your setup, go to the **Form administration** (TCR022) window. Once you enter a transaction number and tab out, the system fills in all positions (sequence numbers) for that transaction in the **Details** section. In the *Reverse* column, select the check boxes next to transactions you want to reverse:



Form administration

Form header

Form*: REVREQUEST
ZRR

Form ID*: 1
1

Form description*: 1

Distributed information access: □

Reversal request details

Transaction

Transaction number*: 19160000

Reversal reason:

Details

□	Transaction type	Period	Sequence number	Description	Tax code	Tax system	Currency	Currency amount	Amount	Reverse
□	GL	201701	0	acc 1	0		GBP	100,00	100,00	□
□	GL	201701	1	acc 1	0		GBP	-100,00	-100,00	□
□	GL	201701	2	acc 2	0		GBP	150,00	150,00	□
□	GL	201701	3	acc 2	0		GBP	-150,00	-150,00	□
□	GL	201701	4	acc 23	0		GBP	200,00	200,00	☑
□	GL	201701	5	acc 23	0		GBP	-200,00	-200,00	☑

7. Submit the form. Follow the worklow to have the required transactions reversed.