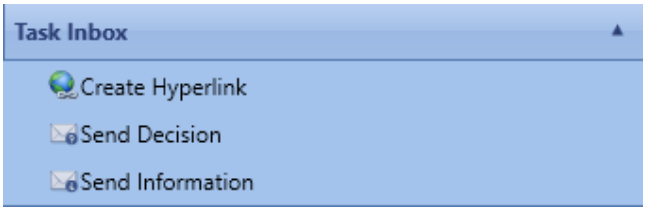
Task Inbox

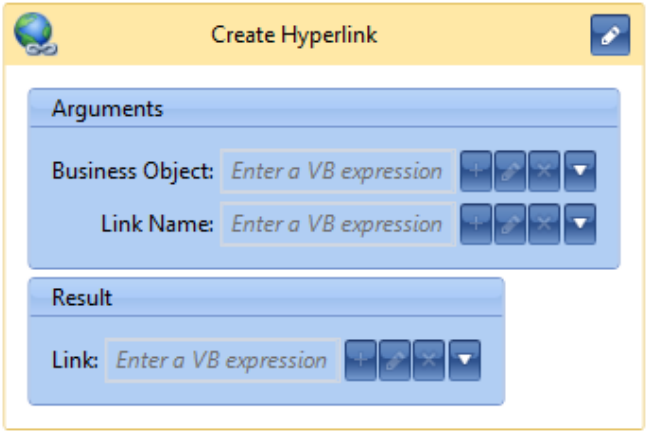
Tab *Task Inbox* contains three activities



Task Inbox activities category

Create Hyperlink

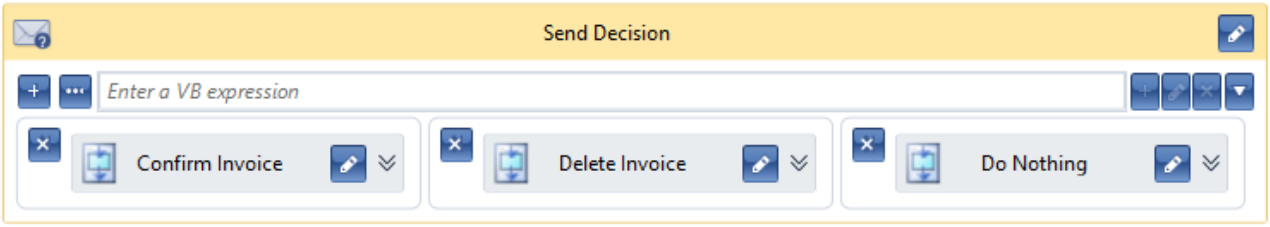
With the use of this activity, it is possible to create a hyperlink to a business object which can be used later, e.g., when sending information to the user. From the level of the task inbox, with the use of that reference, it is possible to open an object form.



Create Hyperlink activity

Send Decision

The activity stops a process activity and sends to the operator/operator group a message informing about the necessity of making a decision. The working of the process will be continued after the user selects one of the options available in the task inbox.

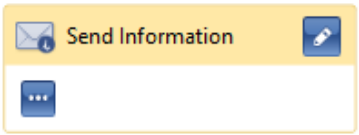


Send decision activity

Example of a decision in the task inbox

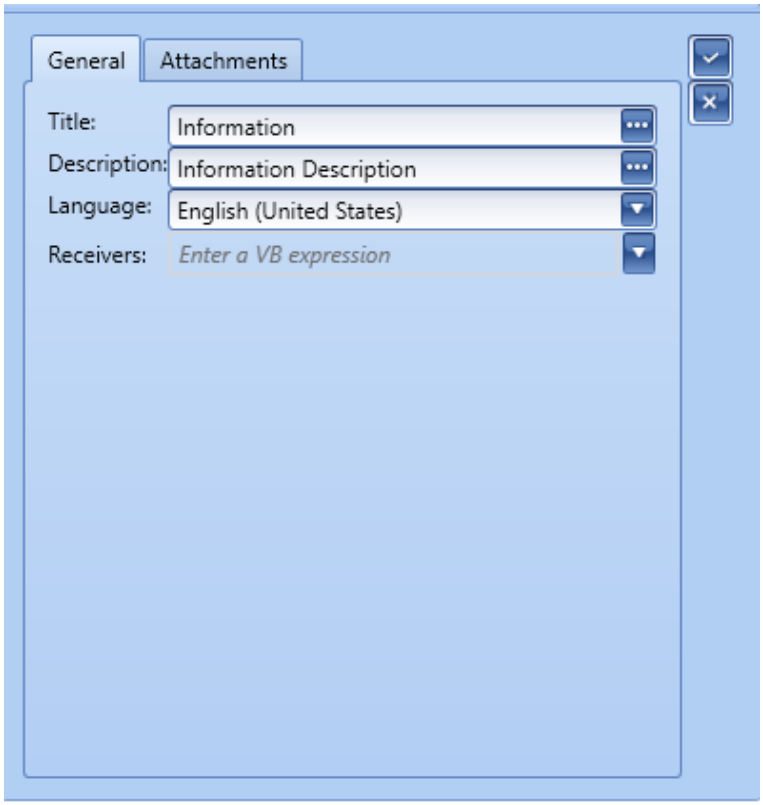
Send Information

The activity allows for sending information to the task inbox of a specific operator, operators or operator groups. Unlike the *Send Decision* activity, the action does not stop a process and does not require any operator's action.



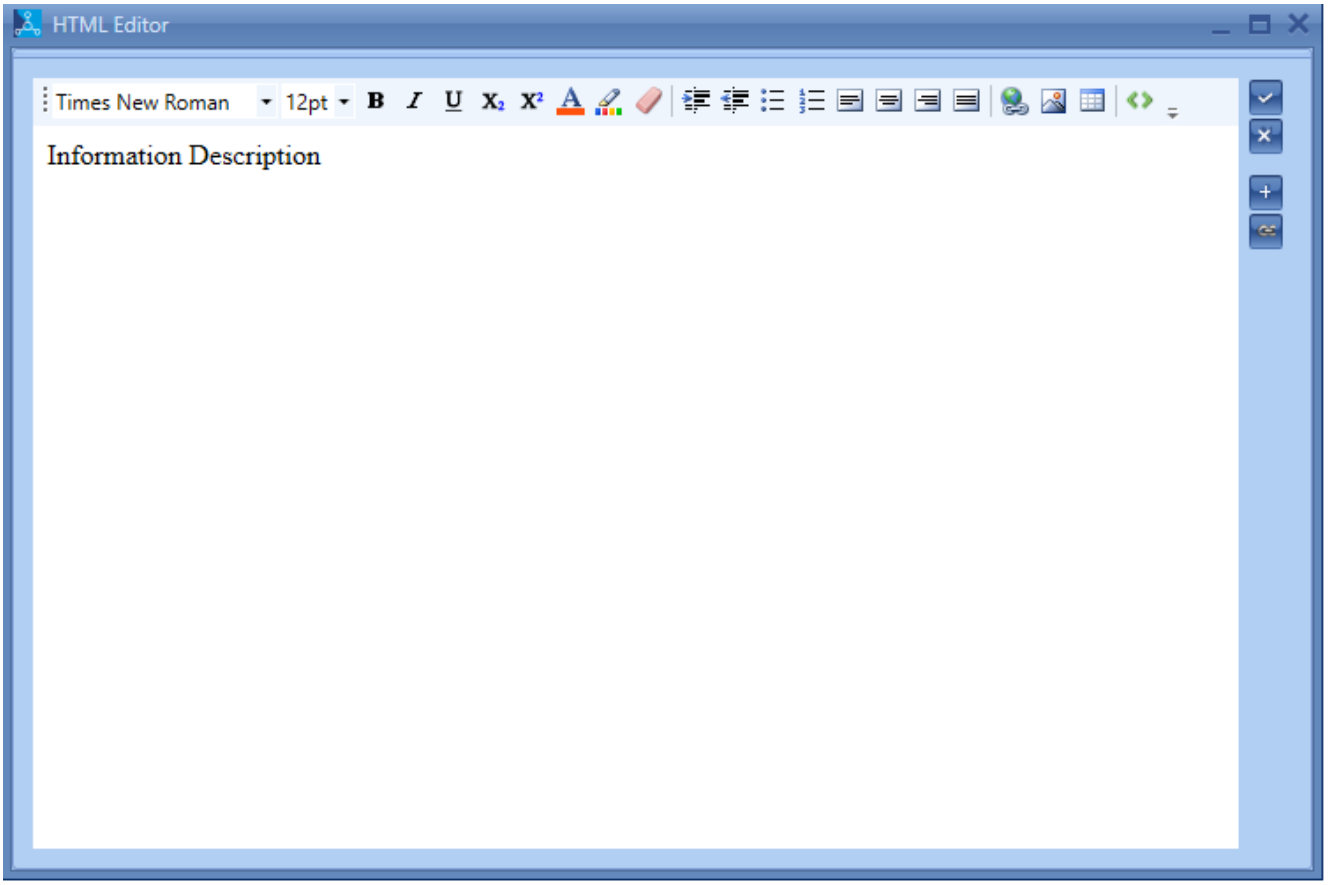
Send Information activity

After selecting […] button, a window for editing message content, recipients, subject, is opened. In the window it is also possible to add attachments to the message being sent.



Edition of an information

After clicking on […] button placed next to the information description, a window for editing message content appears. The HTML editor provides the elementary edit options, such as: selection of the type, size or color of a font, setting of bold, underline, italic, creation of bulleted list and tables, insertion of hyperlinks and images. It allows also for opening the source code view (HTML) to edit manually (button with red frame in the image below) when the basic functions of the editor are not sufficient.



Edition of information content

After switching to the mode of manual edition of the HTML code, it is possible to use tags compliant with the HTML, CSS standard handled by the Internet Explorer browser. Style sheets should be defined directly in the code or, optionally, they can be imported from external WWW sources.

On the right side, there are buttons used for adding variables and hyperlinks to business objects. After selecting a variable or a hyperlink, the content will be added automatically in braces.

Hint
It is possible to contain .NET expressions in a message content. Example of use of the *if.* statement.
@{if(SalesInvoice.IsNullOrDBNull,"Generation of sales invoice failed","Generated sales invoice:"+SalesInvoice.Numerator.Text)}

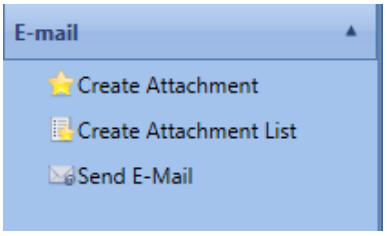
For creating messages, it is also possible to use JavaScript

scripts.

The basic information about writing expressions with the use of the Visual Basic .NET syntax can be found in article *Basic elements of the Visual Basic .NET syntax*.

E-mail

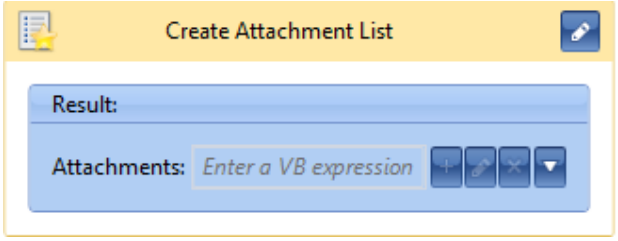
In *E-mail* category, there are 3 activities.



Activities in E-mail category

Create Attachment List

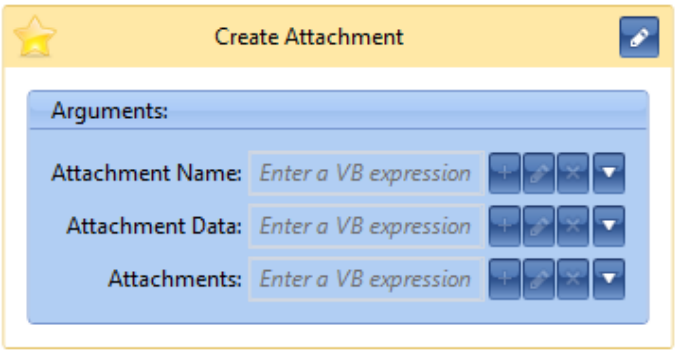
The activity allows for initiating a list of attachments. Attachments can be added to the list and sent by means of an e-mail.



Create Attachment List activity

Create Attachment

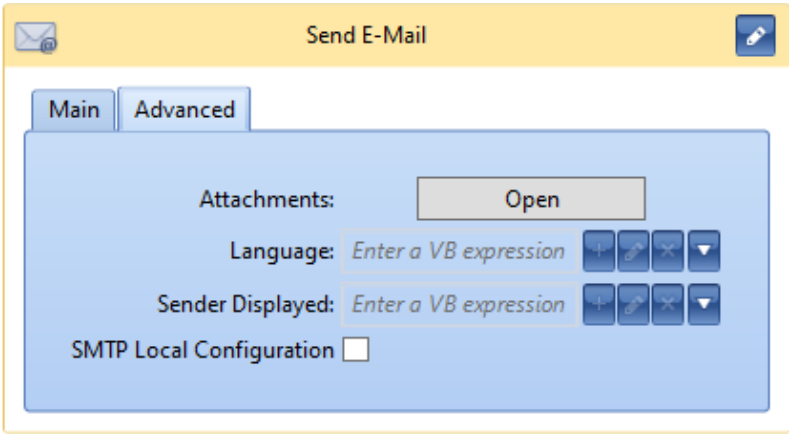
Allows for creating new attachment which, after being added to the list of attachments, can be sent in an e-mail. To create an attachment, it is necessary to specify its name, data and previously created list to which the attachment will be added.



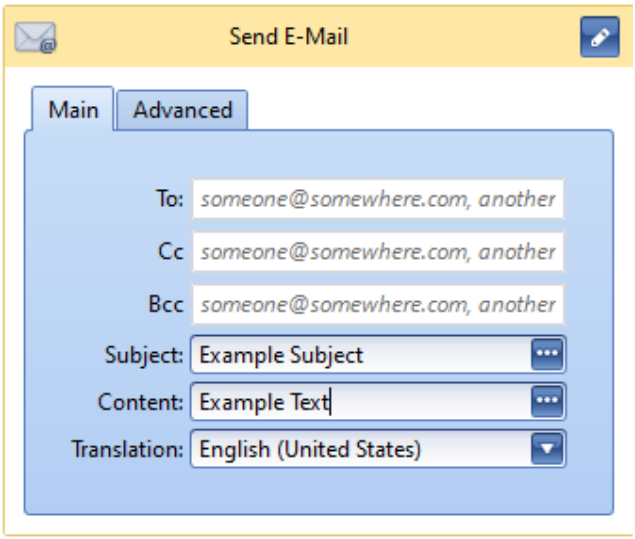
Create Attachment activity

Send E-Mail

With the use of this activity, it is possible to send a message to a specific addresses. The content of a message can be edited in a HTML editor, the same as is used for information and decisions. Completing of textual field *To* is mandatory. In *Advanced* tab, it is possible to add attachments, set language and sender, as well as use SMTP Local Configuration



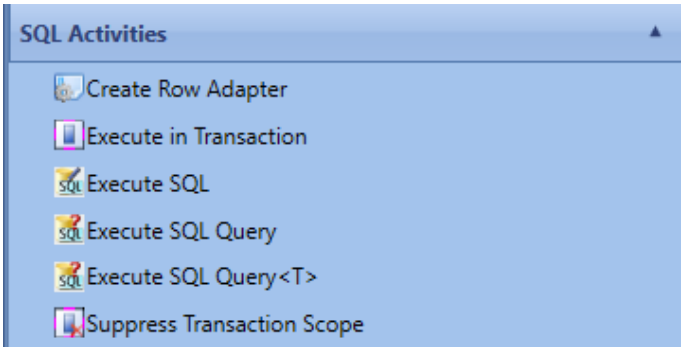
Send E-mail activity



Advanced settings -f e-mail message

SQL Activities

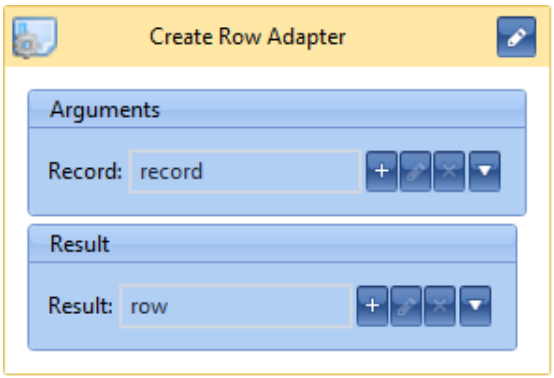
SQL Activities category contains categories executing operations directly on a database.



SQL Activities activities category

Create Row Adapter

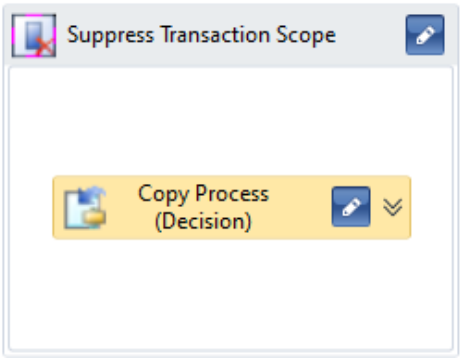
The activity is used for creating *SqlRecordAdapter* object from *IDataReader* object, which allows for an easier analysis of an SQL query results.



Create Row Adapter activity

Suppress Transaction Scope

After using this activity, all operations within it are executed beyond a transaction, which means that in case an execution error occurs, a current object status will be saved in a database and no changes will be undone.

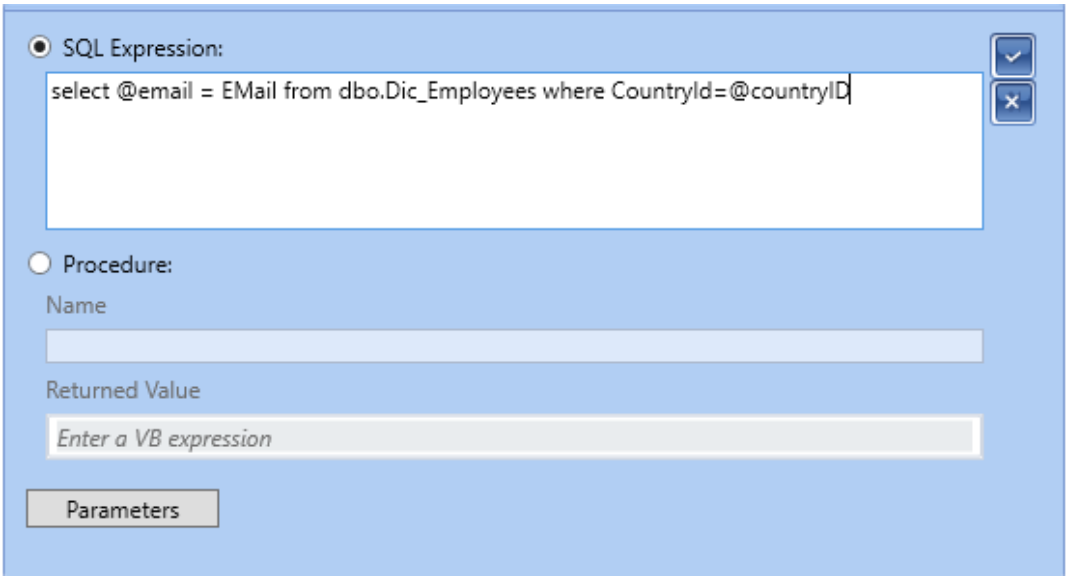


Suppress Transaction Scope activity

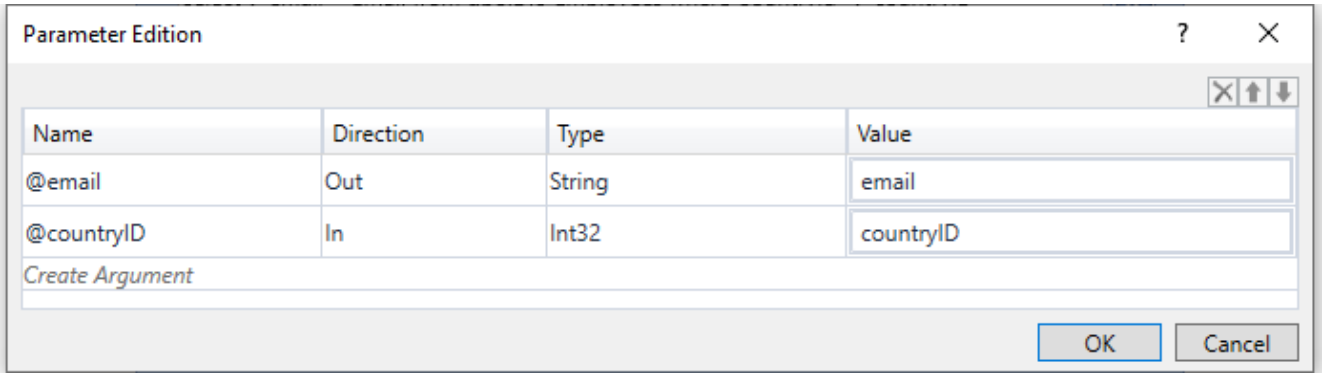
Hint
If a global process is started from a local process, then *Start BPM Process* activity should be placed in the *Suppress Transaction Scope* activity.

Execute SQL

The activity allows for executing an SQL query on a company database from which a process is started. Input parameters can be transferred to such query and data can be retrieved to variables with the use of output parameters.



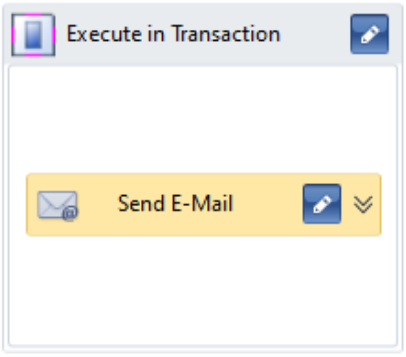
Activity Execute SQL



Execute SQL activity parameters

Execute in Transaction

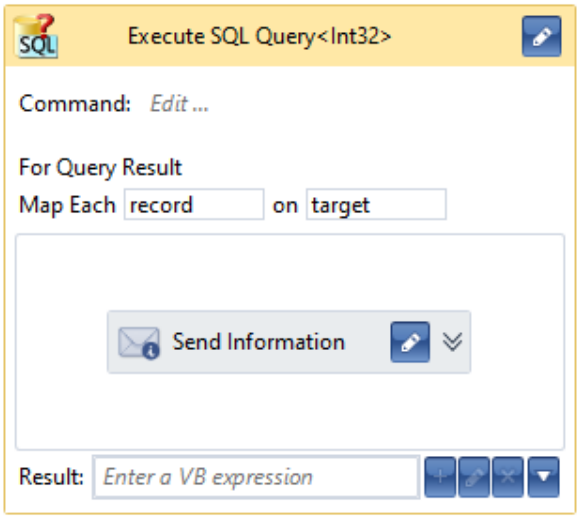
Using of this activity opens an internal transaction in an already existing transaction of a BPM process. Changes introduced by means of the activity to a database will be canceled in case and error (exception) occurs during its execution.



Execute in Transaction activity

Execute SQL Query

This activity allows for executing an SQL query analogic to the *Execute SQL*. Additionally, for each returned row, with the use of a query, it is possible to define specific transactions.

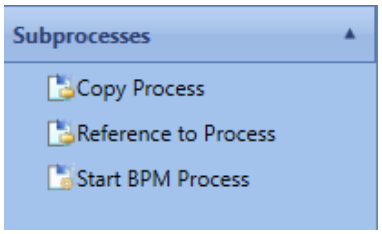


Execute SQL Query activity

Besides the activities described for *Execute SQL Query*, it allows for creating a list of objects of a given type and adding to it objects on the basis of rows which are results of an SQL query.

Subprocesses

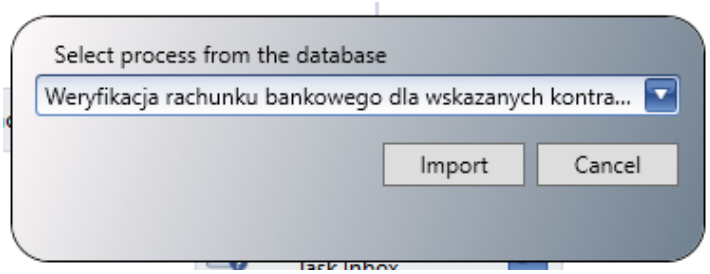
Subprocesses group contains activities responsible for associating of a given process with other processes available in a database.



Subprocesses activity category

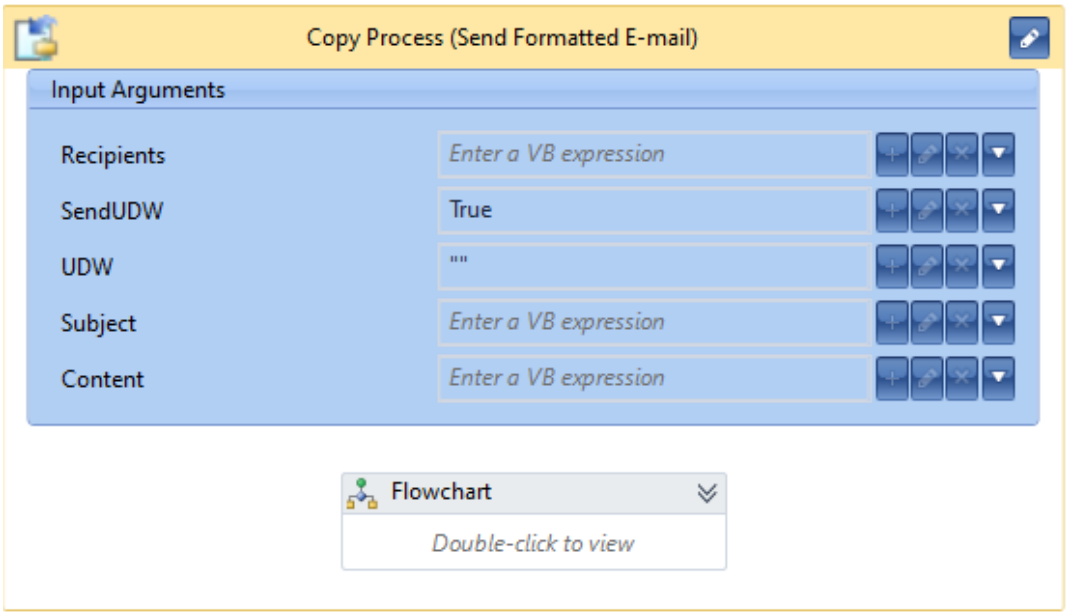
Copy Process

The activity adds an editable copy of a process from the assembly. When adding the activity, the user must indicate the process which is supposed to be added.



Importing of a process during the addition of the Copy Process activity

In the activity window, it is possible to set input and output parameters, preview and edit process definition.



Copy Process activity

Note
The Copy Process activity is not available in the local processes.

Reference to Process

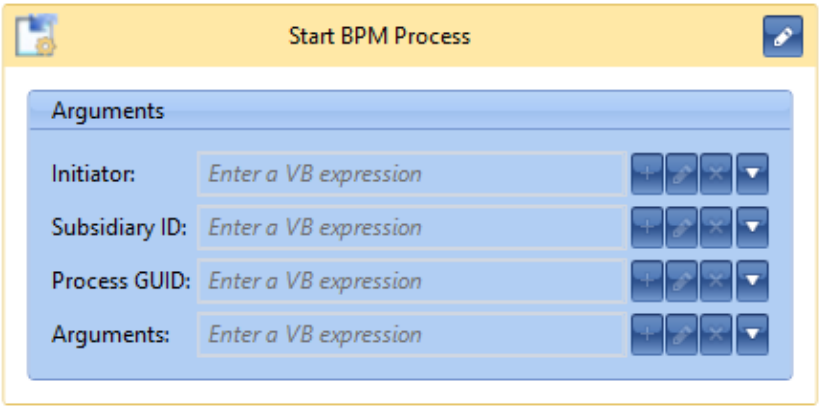
This activity is similar to the *Copy Process* activity. An indicated process becomes subprocess. When creating a reference, current input and output arguments, that is subprocess signature, are read. After changing the signature, it is necessary to refresh manually the references in all process which refer to the reference.

Note
The *Reference to Process* activity is not available in the local processes.

Start BPM Process

The activity allows for starting a BPM process from the assembly for selected subsidiary. As arguments, it is necessary to specify the ID of a distant center or insert

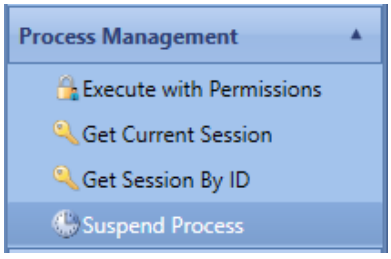
Nothing (in such case the process will start in a local subsidiary. It is also necessary to complete the global ID (*GUID)* which can be found in a hidden by default column of the process assembly. The initiator is the user with whose permissions the process will be started. It is also possible to transfer process arguments as a dictionary, where the key is the parameter name and the value is an object.



Start BPM Process activity

Process management

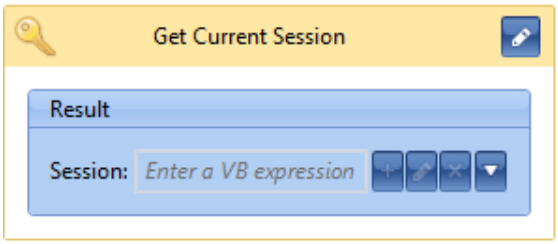
The activities in *Process Management* group are used for handling permissions.



Process Management activities category

Get Current Session

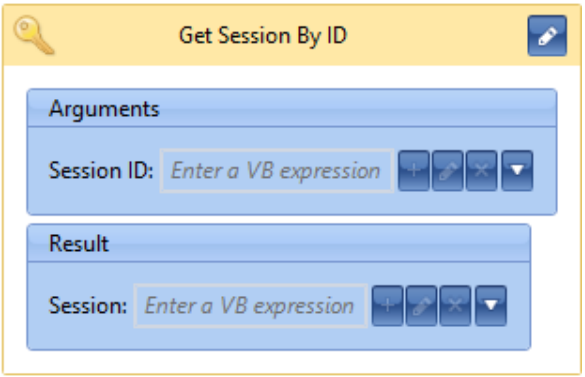
The activity retrieves an object containing information about the current session within which a process is being executed. Such session contains, e.g., information about the center with whose permissions the process is being executed.



Get Current Session activity

Get Session By ID

This activity allows for retrieving of any session on the basis of its ID.

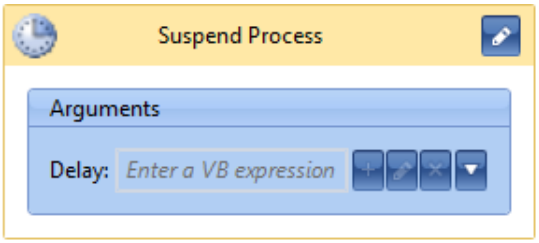


Get Session By ID activity

Suspend Process

With the use of the *Suspend Process* activity, it is possible to suspend execution of a process for a determined time. When a process is suspended, the BPM server queued is released, which allows for starting next process. An important property of the activity is the fact that after process suspension its

status is saved in database. It is strictly related to the parameter *Continuation of interrupted instances,* which is described in article *Advanced settings*.



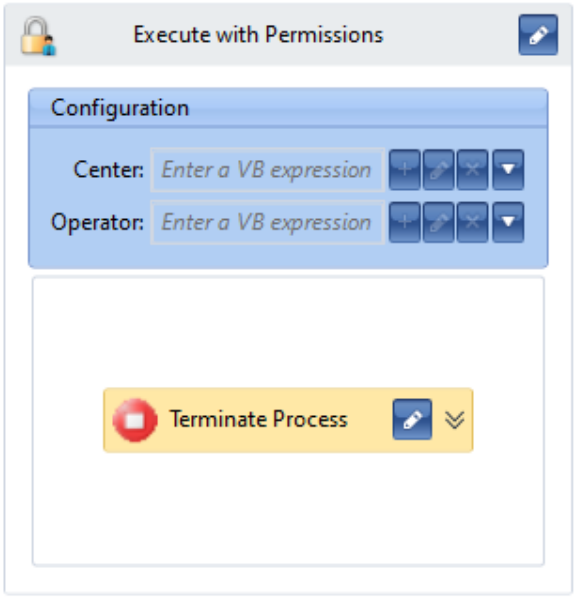
Suspend Process activity

Hint
Before suspending a process it is recommended to set variables to *null,* if it is possible. It is a good practice which allows for saving memory resources.

Note
It is not possible to use the activity *Suspend Process* in a local process.

Execute with Permissions

By default, a process is executed with the permissions of the operator and center set for that BPM process (global processes) or of the Comarch ERP Standard application (local processes). To be able to execute a part of a process with permissions of another center and/or operator, it is necessary to use the *Execute with Permissions* activity.

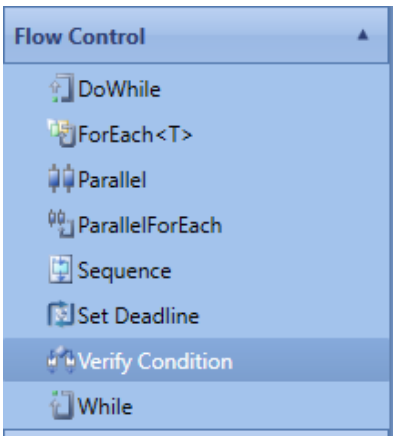


Execute with Permissions activity

Flow control

Flow control

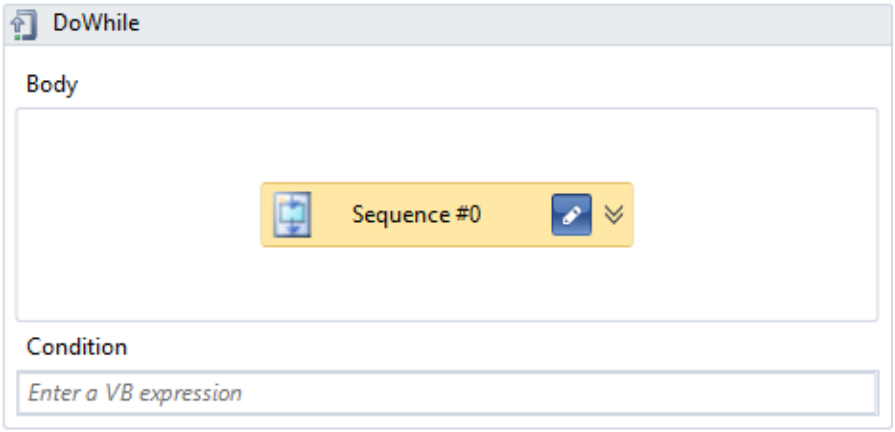
The activitires in *Flow Control* group are used for multiple repetition of operations, introduction of conditionals, limitation of processes in time or definition of operations for parallel execution.



Flow control activities category

DoWhile

The activity executes operations in a loop until a specific condition is fulfilled.



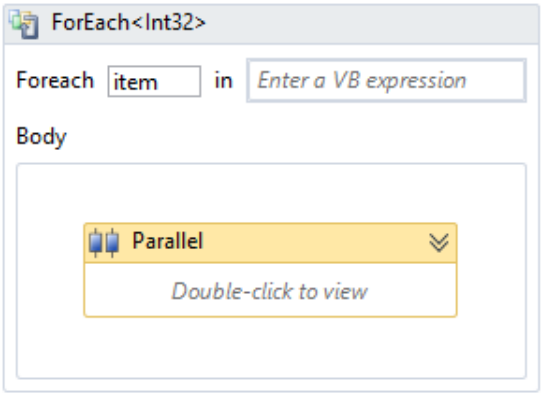
DoWhile activity

Note
The activity *DoWhile* will be executed always at least once, because the condition is verified every time after execution of instructions defined in *Body* sections.

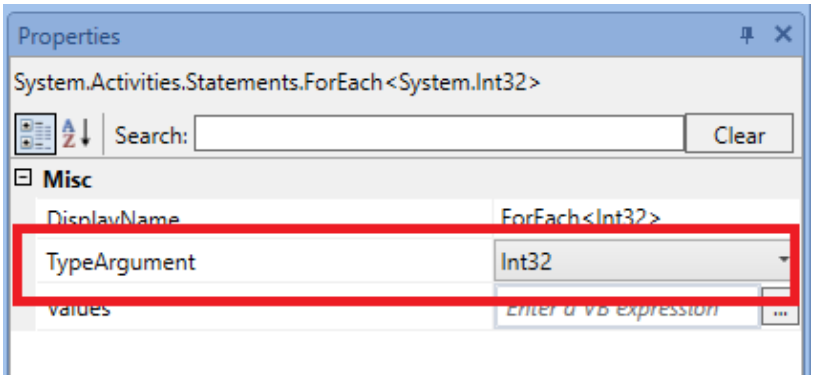
ForEach<T>

The activity allows for executing operation for each item in collection. To use the activity in a proper way, it is

necessary to remember to set in the tab *Properties* the appropriate type of variables included in the collection.



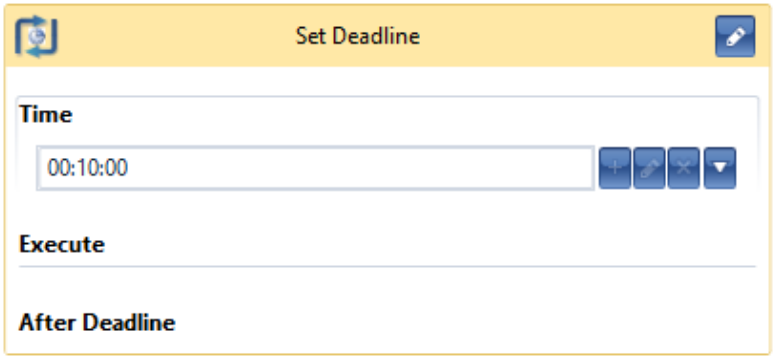
For each activity



Setting types of collection elements for ForEach activity

Set Deadline

The activity allows for limiting the time of executed operations. The process will be executing instructions declared in *Execute* tab maximally for the period specified in *Time* tab. If the time gets exceeded, the process will execute operations from *After Deadline* tab.

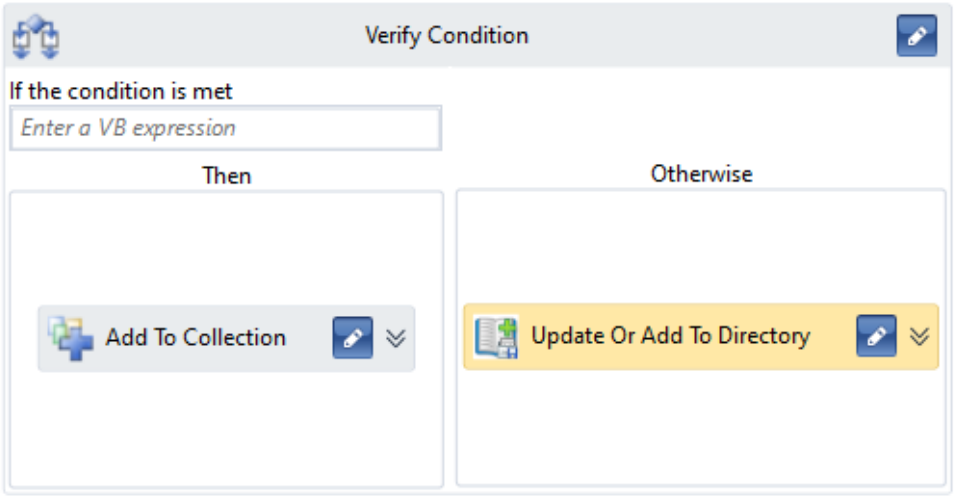


Set Deadline activity

Note
Set Deadline activity is not available for local processes.

Parallel

Allows for parallel execution of operations, activities or subprocesses.



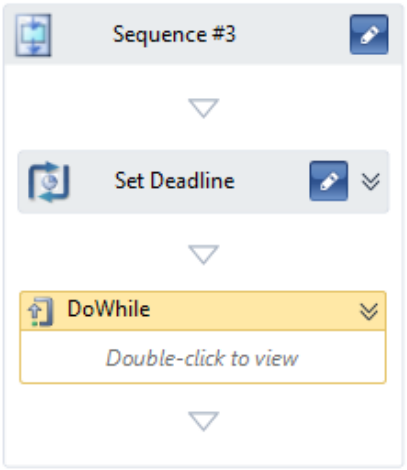
Parallel activity

ParallelForEach

The activity allows for defining parallel execution of operations for each element. It is a combination of *ForEach* and *Parallel* activities.

Sequence

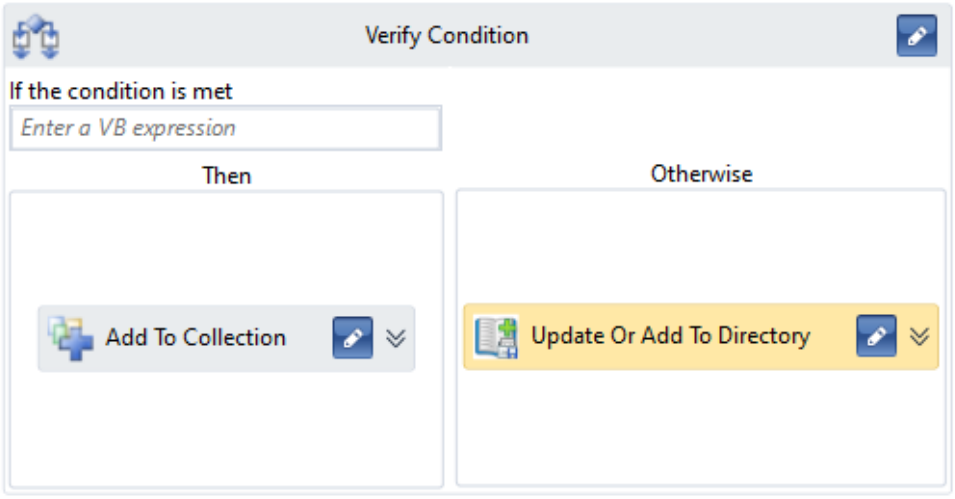
In a sequence, operations are executed one-by-one. It is not possible to branch them. To change execution order, it is sufficient to drag an activity higher or lower.



Sequence activity

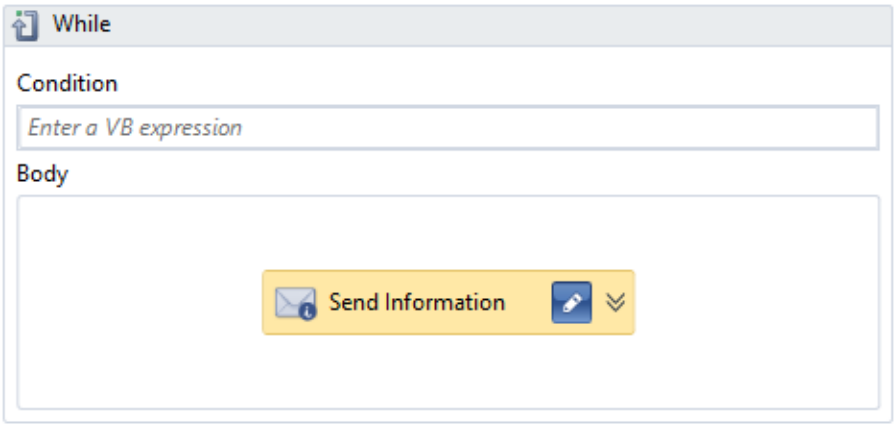
Verify Condition

The activity verifies whether a given condition is real. If yes, it executes instructions placed on the left side, if not – on the right. Upon clicking with the right mouse button on the activity, it is possible to select *Add Branch* option. It allows for executing a subsequent conditional, if the first condition is not fulfilled (analogically to *elseif* instruction)



Verify Condition activity

Activity analogical to the activity *DoWhile*. The difference consists of the fact that in this case a condition is verified always before a loop is executed.



While activity

Flowchart

Flowchart group contains activities allowing for adding additional activity trees to a process.

Flowchart activity category

FlowSwitch

The activity allows for executing specific operations depending on the value of the input parameter which can be also an expression. *Default* value is set at the moment when the input parameter is not equal to any other value defined in *FlowSwitch.* It is possible to select any parameter type from among types available in *Comarch ERP Standard* or added by the user in references.

FlowSwitch activity

Flowchart

This activity contains a process scheme analogical to the one created automatically by the editor when adding a process, if no sequence is selected. More information regarding creating a process can be found in article <<Adding new process>>.

Flowchart activity

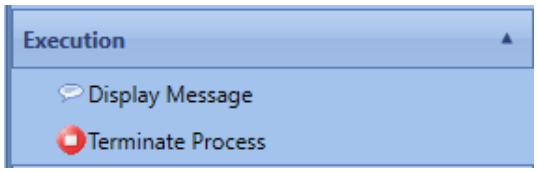
Condition

This activity is similar to the *FlowSwitch* activity. The difference consists of the fact that a condition which can be true or not, is verified. Depending on the result, the process executes indicated operations.

Condition activity

Execution

Execution group contains two activities:



Execution activities category

Display message

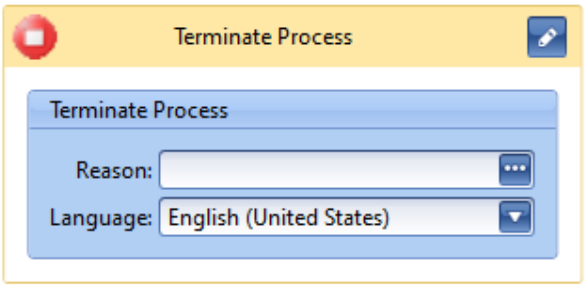
The activity allows for displaying messages for users in the interface. As arguments, it is necessary to enter subject and content of a message. Both values can be added along with their translations.

Display Message activity

Note
The *Display message* activity is available for local processes only.

Terminate Process

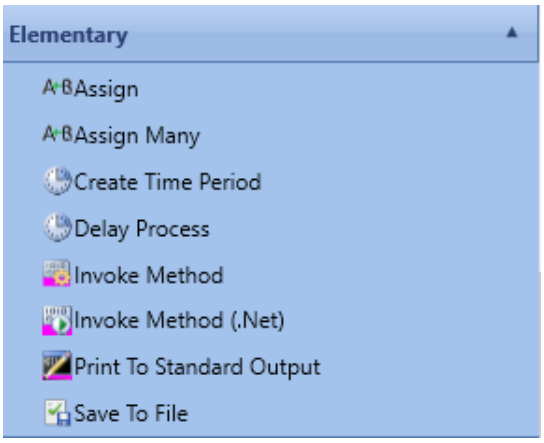
The activity allows for permanent interruption of execution of a process instance. To add the activity in a proper way, it is necessary to complete field *Reason,* entering a text containing the reason for the interruption of the process. It will be visible for the user form whom the process will execute that activity.



Terminate Process acttivity

Elementary

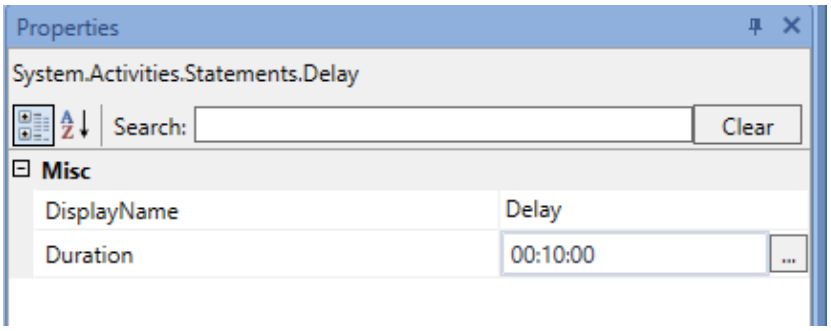
This group contains activities which make it possible to assign values to variables or to execute methods.



Elementary activities category

Delay Process

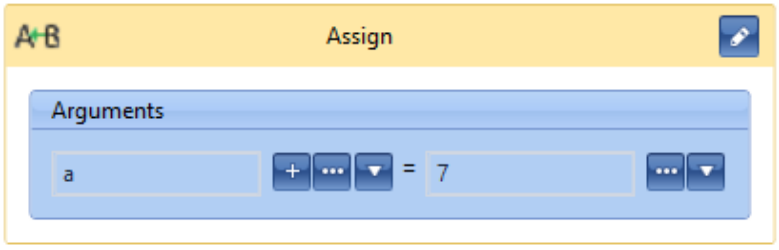
The activity allows for stopping a process for a time specified in the property window. Such process remains with active status, which means that it is not possible to execute another process making part of a current thread. The process status is not saved in a database.



Properties of the activity Delay Process

Assign

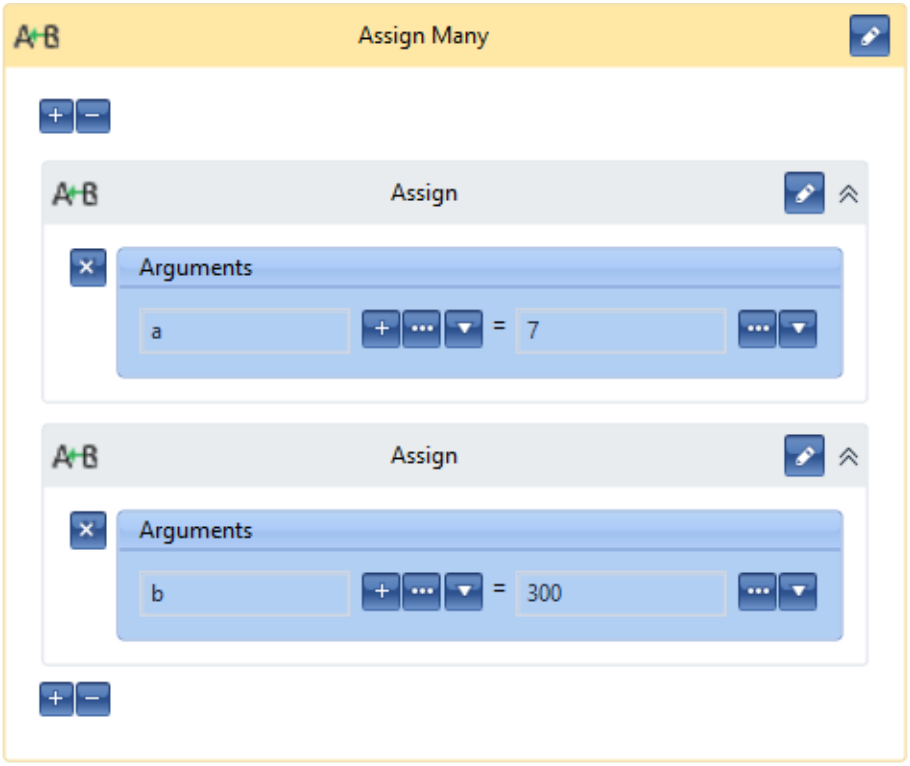
With the use of this activity, it is possible to assign value to a selected variable.



Assign activity

Assign Many

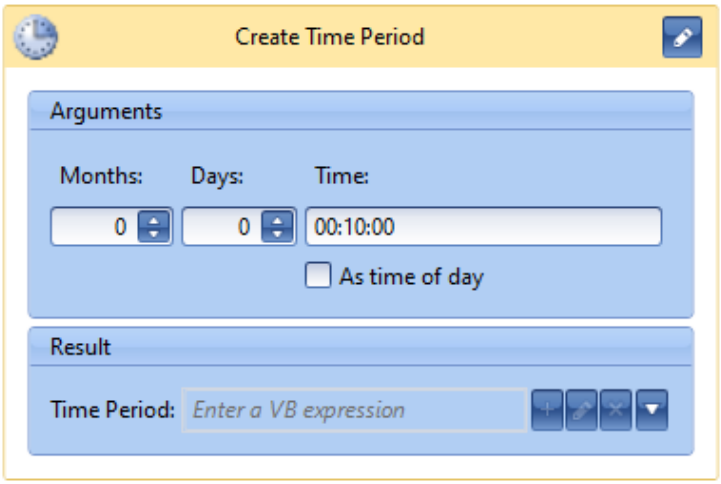
Consists of assigning values to variables in batch, thanks to which it is not necessary to use many *Assign* activities, one after the other.



Assign Many activity

Create Time Period

The activity allows for generating a variable of *TimeSpan* type which stores a time period and can be used as an argument, for example, in the *Delay Process* activity.



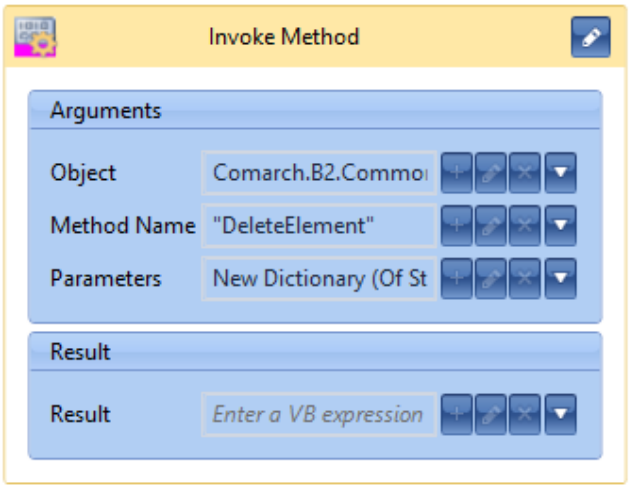
Create Time Period activity

Invoke Method

The activity allows for executing any system method inside of a process.

Example
Calling *DeleteElement* method on a sales invoice document.
To the *Flowchart*, it is necessary to transfer the *Invoke Method* activity and complete the following fields:

- Object – *B2.Common.Locator.GetService(Of Comarch.B2.Sales.Interfaces.Presentation.ISalesInvoiceService)*
- Method Name – *“DeleteElement”*
- Parameters – new dictionary initiation: *New Dictionary (Of String, Object) From {{“document”, FS},{“elementId”, FS.Elements(0).Id}}*
- Result – a blank field can be left

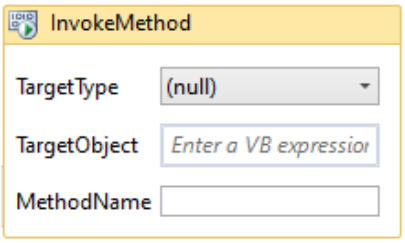


Invoke Method activity

The activity *Invoke Method* completed this way will delete the first item from the invoice which was previously uploaded to the process.

Invoke Method (.Net)

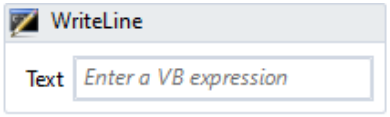
This activity is similar to the *Invoke Method* activity. One of the fields *TargetType* or *TargetObject* must be specified to enable execution of a non-instance or instance method. In the property window, it is possible to specify method parameters or assign its result to a variable.



Invoke Method (.Net) activity

Print To Standard Output

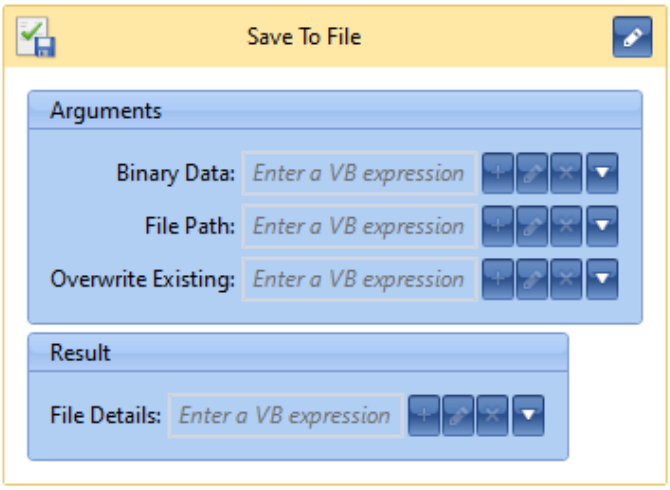
Allows for writing a given text in the BPM server console. Optionally, in the activity properties, it is also possible to specify the author of the text.



Print To Standard Output activity

Save To File

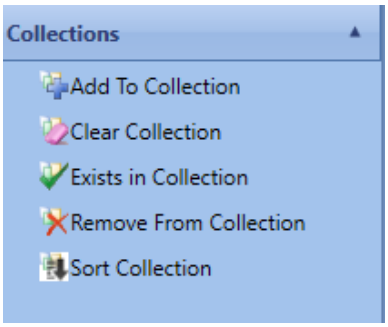
With the use of this activity, it is possible to save data to a file. The data must be given in the binary form (byte array). It is also necessary to ender the file path along with the file name and specify whether the file has to be overwritten, if it already exists in the given location.



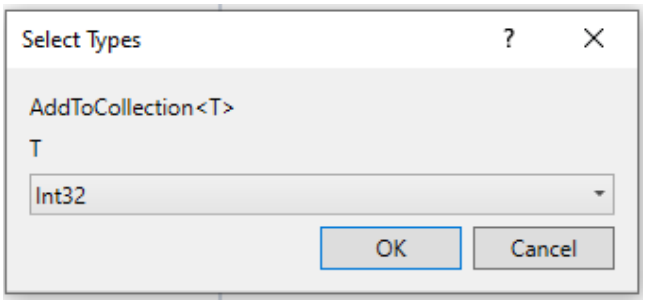
Save To File activity

Collections

This group contains activities executing operations on collections, e.g., on lists. Before adding any activity regarding a collection, the user has to specify its type. A type can be any Comarch ERP Standard object, numeric or textual variable etc.



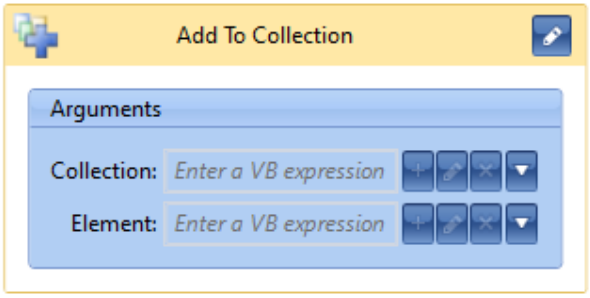
Collections activity category



Selecting collection type

Add to Collection

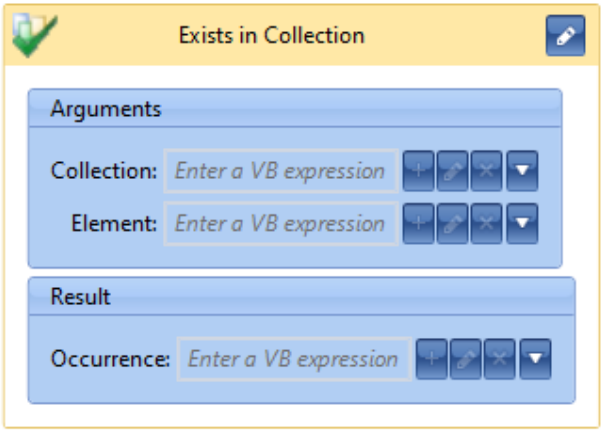
The activity adds an item to a collection. It is necessary to indicate a collection to which the item is to be added.



Add to Collection activity

Exists in Collection

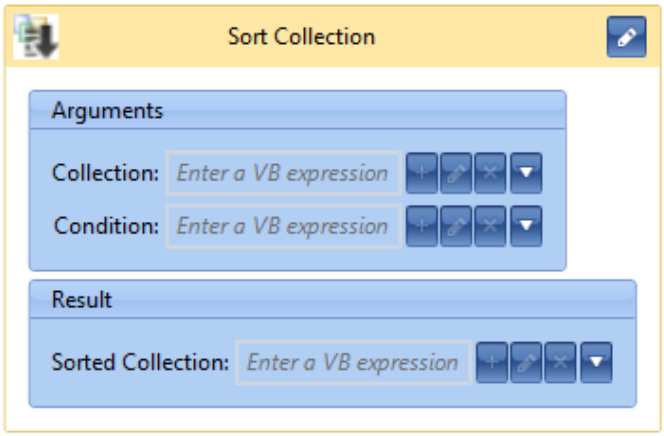
The activity verifies whether a given item exists in a collection. As a result, it returns the logical value *True*, if such item exists in the collection. Otherwise, the value *False* is returned.



Exists in Collection activity

Sort Collection

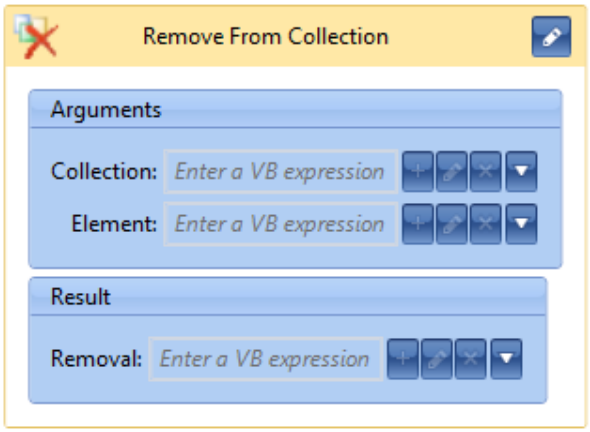
With the use of this activity, it is possible to sort a collection. It is necessary to specify a condition in textual form (*“ASC”* or *“DESC”*) which will determine whether a collection should be ordered in ascending/descending way. In case a collection contains more than one filed, it is possible to select the filed by which the sorting is supposed to be carried out, by entering its name under the (“Name DESC”) condition. After a comma, it is possible to add another condition, if a collection is to be sorted, e.g., first, by name and then by ID.



Sort Collection activity

Remove From Collection

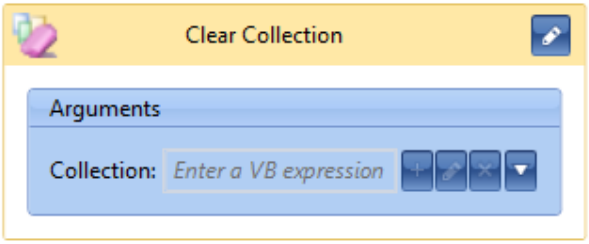
The activity allows for removing a selected item from a collection. As a result, a logical value specifying whether the process has executed the operation correctly, is returned.



Remove From Collection activity

Clear Collection

With the use of this activity, it is possible to remove all items from a given collection.



Clear Collection activity

Error handling

The group allows for handling exceptions returned by the application in case an unexpected error occurs during the execution of an operation or in process places planned by the

author.

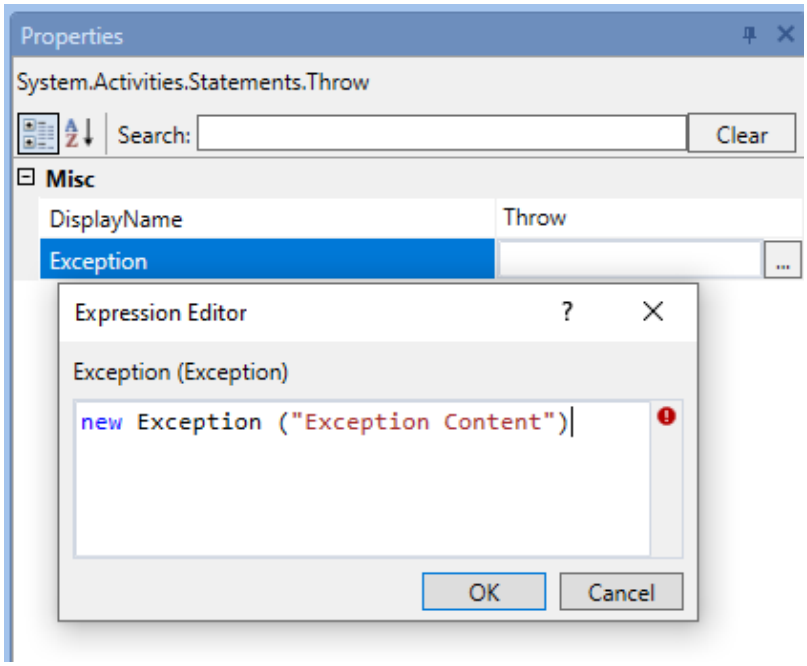
Error Handling activity category

Rethrow

The activity can be used only in *Catch* field of *TryCatch* activity. It throws again an exception which was already handled.

Throw

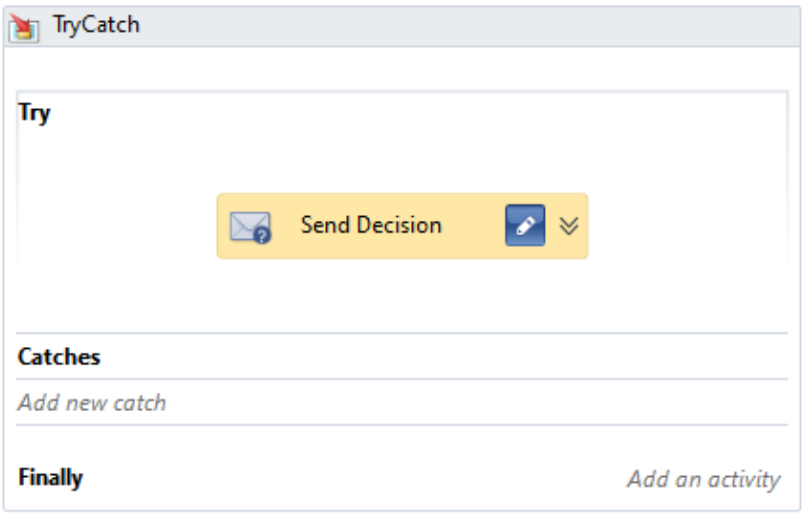
The activity throws an exception indicated by the process author. The exception should be defined in *Properties* window. It is also possible to use exceptions available in Comarch ERP Standard API.



Properties of the activity Throw

TryCatch

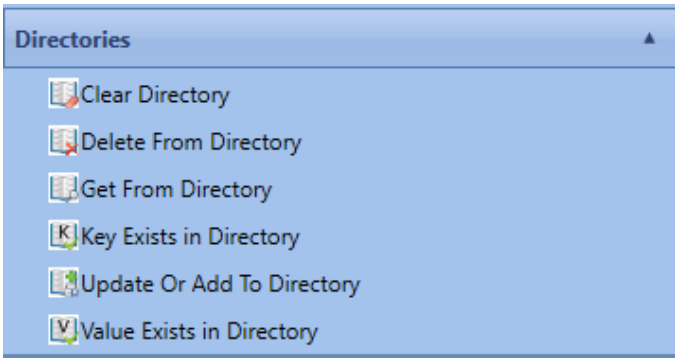
In case an exception occurs during the execution of operations contained in *Try* field, execution of the action will be interrupted and actions defined in *Catch* field will be started. Regardless of whether an exception occurs or not, at the end the operations indicated in *Finally* field will be executed.



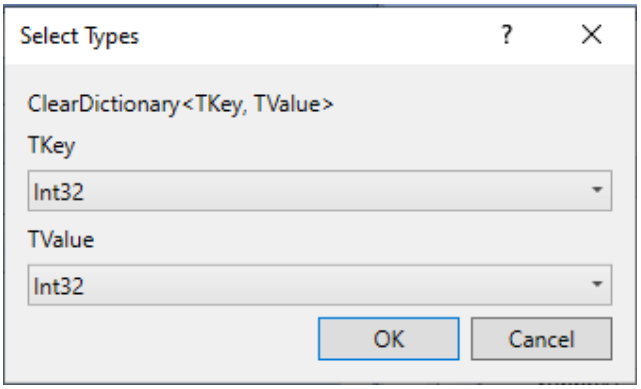
TryCatch activity

Directories

In the group, there are activities allowing for executing of operations on directories defined as groups of key-value pairs. Before adding an activity from that group, it is necessary to specify key and value type.



Directories activity category



Selecting keys and values type for a directory

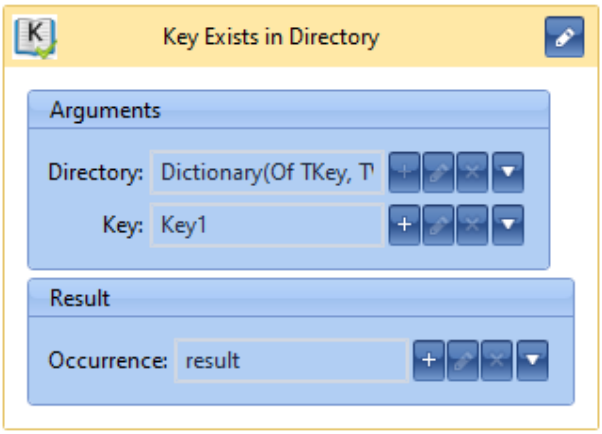
Update Or Add To Directory

Allows for adding values for a specific key to a directory. If the key already exists in the directory, the value will be updated.

Update Or Add To Directory activity

Key Exists in Directory

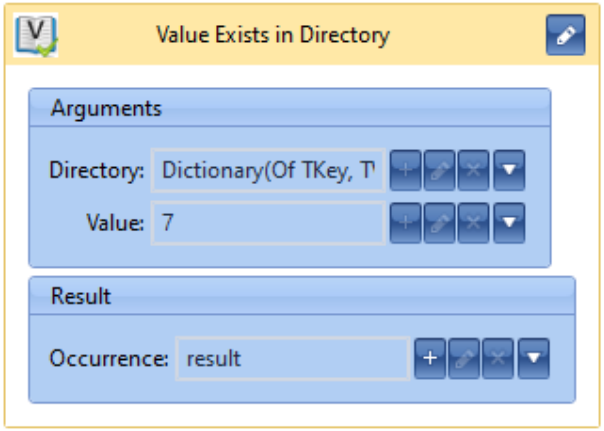
The activity verifies whether in a given directory, there is an entry containing a specific key. The result is a logical variable assuming *True* value, if the key exists in the directory and *False* value, if it does not exist.



Key Exists in Directory activity

Value Exists in Directory

The activity verifies whether in a given directory, there is an entry containing a specific value. The result is a logical variable assuming *True* value, if the value exists in the directory and *False* value, if it does not exist.



Key Exists in Directory activity

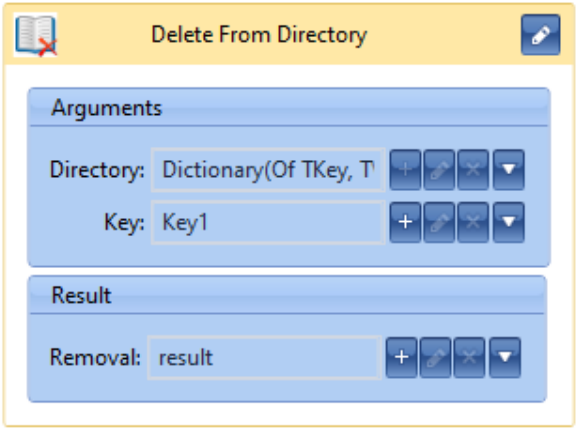
Get From Directory

The activity retrieves a value from a directory on the basis of a specific key. The result, besides the value, is a logical variable containing the information whether the entry was retrieved.

Get From Directory activity

Delete From Directory

The activity deletes an entry from a directory on the basis of a specific key. The result, besides the value, is a logical variable containing the information whether the entry was deleted.



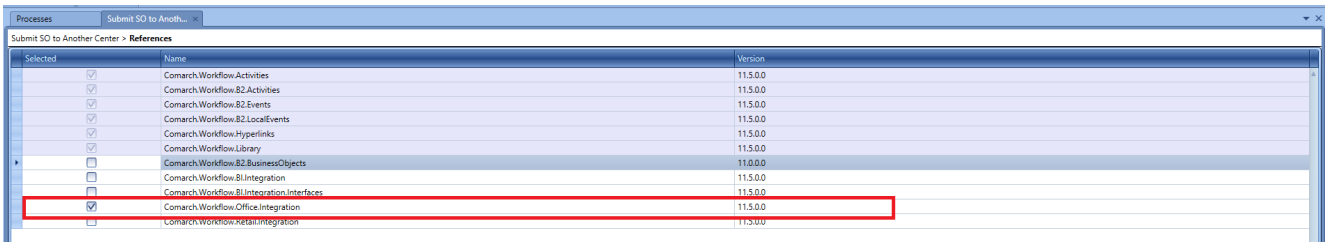
Delete From Directory activity

Clear Directory

The activity deletes all entries from a directory.

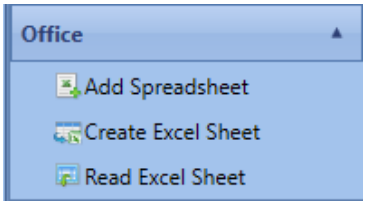
Office

The activities of *Office* group allow for integrating spreadsheets of the Microsoft Office system with the application Comarch ERP Standaed BPM. Thanks to them, the user can download and save data to spreadsheets or create new ones. To be able to use the actvities of the *Office* group, first, it is necessary to <<add reference>> *Comarch.Workflow.Office.Integration.* It is a standard reference, already uploaded to the system. The user must only select a reference for a given process.



Reference for integration with the Microsoft Office system

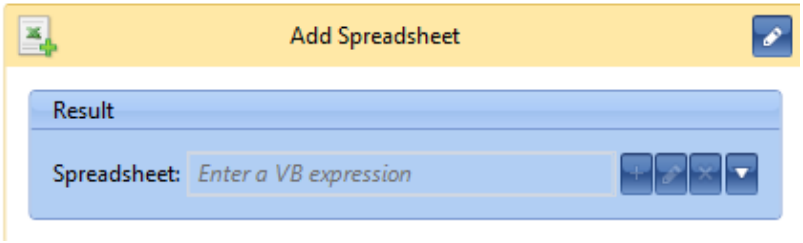
The group contains three activities:



Office activities catgory

Add Spreadsheet

The activity adds a spreadsheet structure A variable of *SpreadsheetDocument* type is created The property of an object of *SpreadsheetDocument*. type is *Records* which contains a list of rows of a given spreadsheet. Each row object contains *Cells* property which contains the list of columns in a given row.

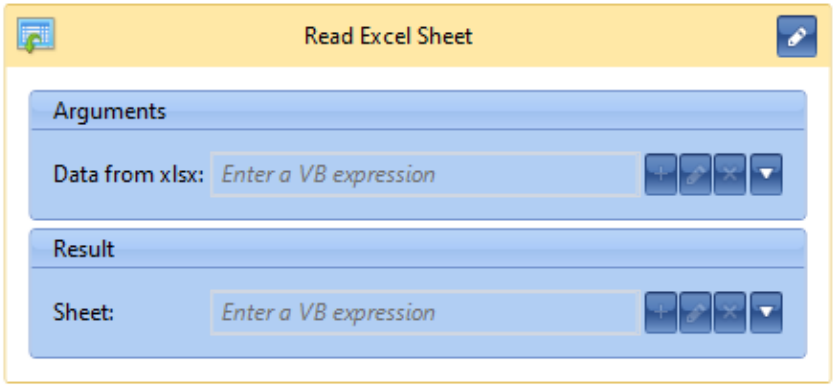


Add Spreadsheet activity

Read Excel Sheet

The activity allows for reading data from a spreadsheet. As an arguments, it is necessary to enter the data from the file. The result is a spreadsheet – variable of *SpreadsheetDocument*. document type.

Hint
A file can be read, for example, from a <<parameter>> which must be previously defined as *FileDataType* type. The file data is stored in the *FileDataType.Data* property.

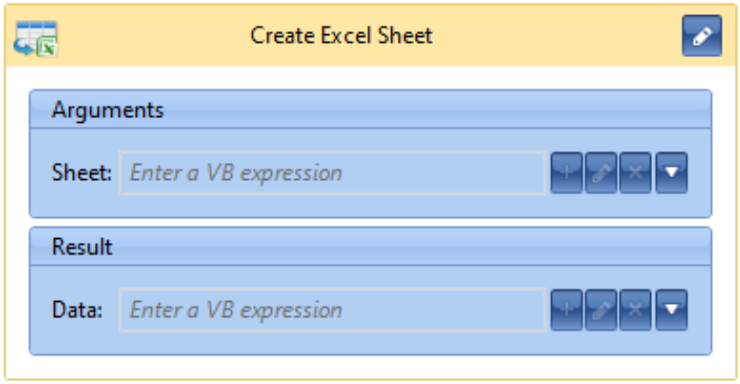


Read Excel Sheet activity

Create Excel Sheet

The activity generates binary data on the basis of an indicated sheet.

Example
After generating binary data it is possible to generate an attachment from it, by *Crate Attachment* activity. An attachment created this way is added to a list of attachment previously created by *Create Attachment List* activity. A sheet can be sent to the task inbox of an employee or send via e-mail.



Read Excel Sheet activity