

# EXPRESSION REFERENCE

---

## Table of Contents

---

<b>1</b>	<b>Introduction .....</b>	<b>3</b>
1.1	Access to Ticket Fields.....	3
<b>2</b>	<b>Messaging &amp; Collaboration .....</b>	<b>4</b>
2.1	Prefix and Suffix .....	4
2.2	Expressions .....	4
<b>3</b>	<b>Workflow Management.....</b>	<b>5</b>
3.1	Using Expressions in Workflow Plug-Ins .....	5
3.2	Using the Special Data Fields .....	6
<b>4</b>	<b>Expressions.....</b>	<b>9</b>
4.1	General Expressions.....	9
4.2	Expressions for Configuration Items.....	9
4.3	Expressions for Dynamic Placeholders .....	11
4.4	Expressions for the Early Warning System.....	11
4.5	Expressions for Expenses.....	11
4.6	Expressions for the Knowledge Base .....	12
4.7	Expressions for Messages .....	13
4.8	Expressions for Process Roles.....	14
4.9	Expressions for Tasks .....	14
4.10	Expressions for Task Life Cycles .....	16
4.11	Expressions for Task Schemas .....	16
4.12	Expressions for Task Statuses .....	17
4.13	Expressions for Tickets .....	17
4.14	Expressions for Workflow Items .....	22
<b>5</b>	<b>Examples .....</b>	<b>24</b>
5.1	Delimitation of Date/Time .....	24
5.2	Solve Ticket Directly.....	24
<b>6</b>	<b>Operators .....</b>	<b>25</b>
6.1	Introduction .....	25
6.2	Using the Operators .....	25
6.3	Relational Operators .....	26
6.4	Operators for Strings.....	27
6.5	Arithmetic Operators .....	28
6.6	Logical Operators.....	29
6.7	Brackets.....	29
<b>7</b>	<b>Statistics and Change Management.....</b>	<b>30</b>
7.1	Statistics.....	30
7.2	Change Management.....	30

# 1 Introduction

---

*TicketXpert.NET* (TXP) and *WorkflowXpert.NET* (WXP) have their own expression language, which can be used for individual customization or substantial extensions at different locations. Expressions are particularly helpful, when it is not possible to solve a complex configuration problem by only using the user interface – the expression system thus offers extension options specially tailored to power users.

Expressions mainly can be used in three modules: in the *Messaging & Collaboration* module for the creation of custom message templates, in the *Service Level Management* for the computation of non-standard escalation dates, and in the *Workflow Management*. Because not all elements of the expression language can be used in every module, the following sections describe the application of the expression language in the respective module.

## 1.1 Access to Ticket Fields

You can access the content of ticket fields via miscellaneous expressions. At this, please make sure that the ticket fields you want to access via expression do not use any mathematical operators (e.g. “-” or “+”) or similar special characters. These special characters can prevent the proper access to the ticket fields.

## 2 Messaging & Collaboration

For every message type, there are several placeholders available, which will be replaced by the respective text on sending. Additionally to the default placeholders that are located on the right side (placeholder bar) and faded in dynamically on mouseover, there is an option of adding individually defined placeholders.

### 2.1 Prefix and Suffix

Every placeholder defined individually begins with the prefix {\$ and ends with the suffix \$} (start and end token). Between the prefix and the suffix, an arbitrary expression can be entered, which can be interpreted in context to the message to be sent.

If an expression cannot be interpreted, NO text is inserted into the message in this spot.

### 2.2 Expressions

The following screenshot shows the use of expressions in a message template.

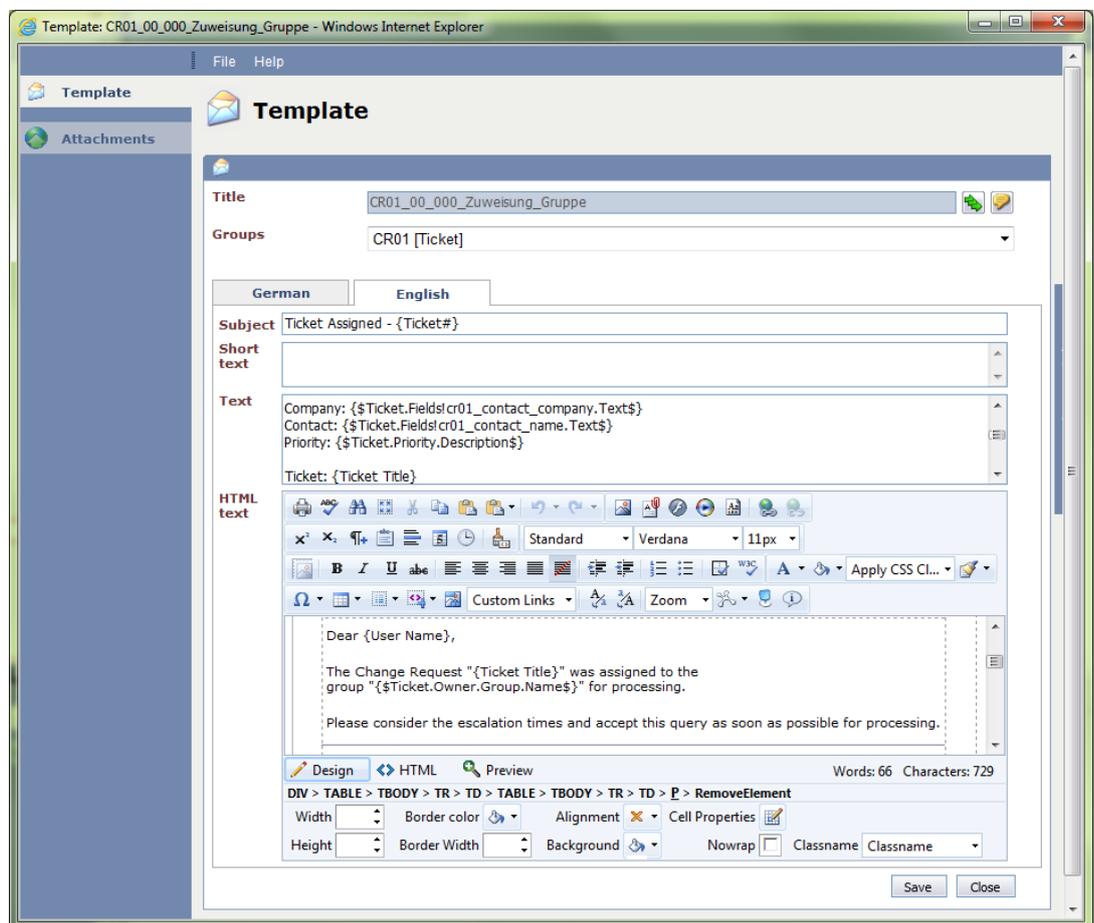


Figure 2-1

A listing of the available expressions can be found in Chapter 4.

## 3 Workflow Management

### 3.1 Using Expressions in Workflow Plug-Ins

The following example shows the use of an expression in the plug-in *Conditional Switch on Expression*. It has two exits and decides whether the ticket should be directed to the exit *Condition fulfilled* or not, depending on the underlying expression. When phrasing the expression, it is necessary to express a relation, which enables the plug-in to make a yes/no decision. In this very example, the condition is fulfilled if the ticket owns a field called *RequiresAnswer* and if it has the content *True*.

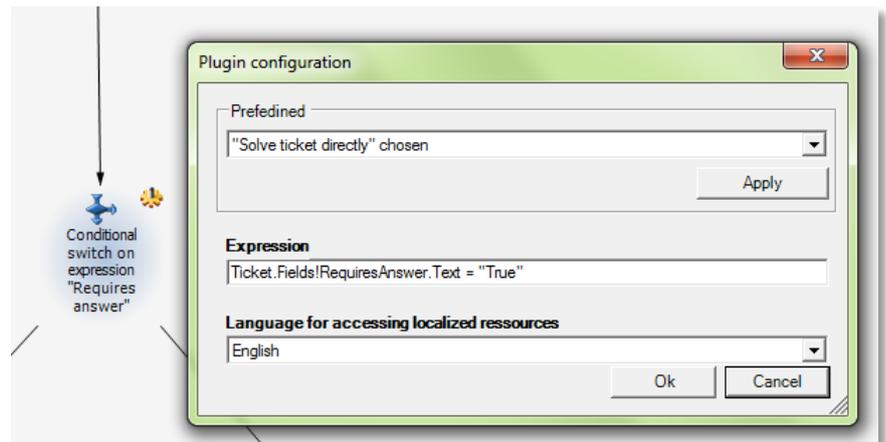


Figure 3-1

The next example illustrates another use of the *Conditional Switch on Expression* plug-in. In this case, it should decide whether a ticket is to be assigned to a user. The special key word *NULL* is used here. This condition is fulfilled if the ticket is assigned to **no** user.

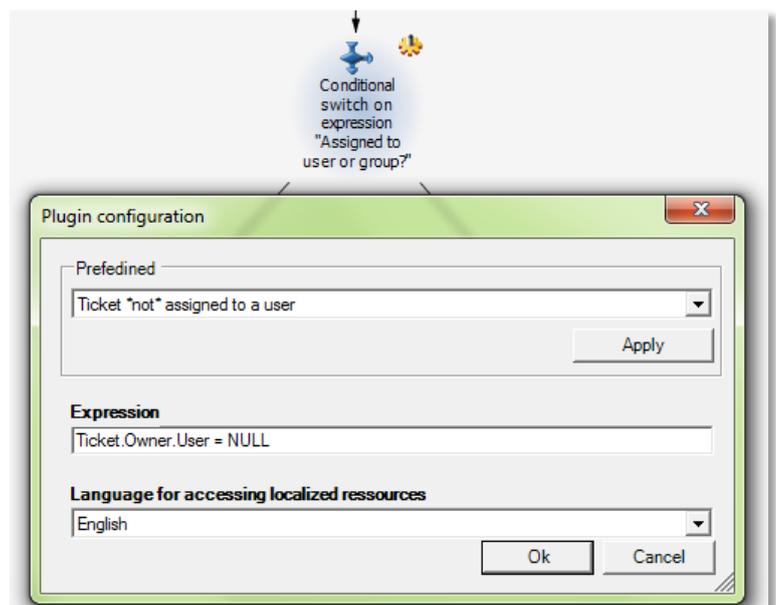
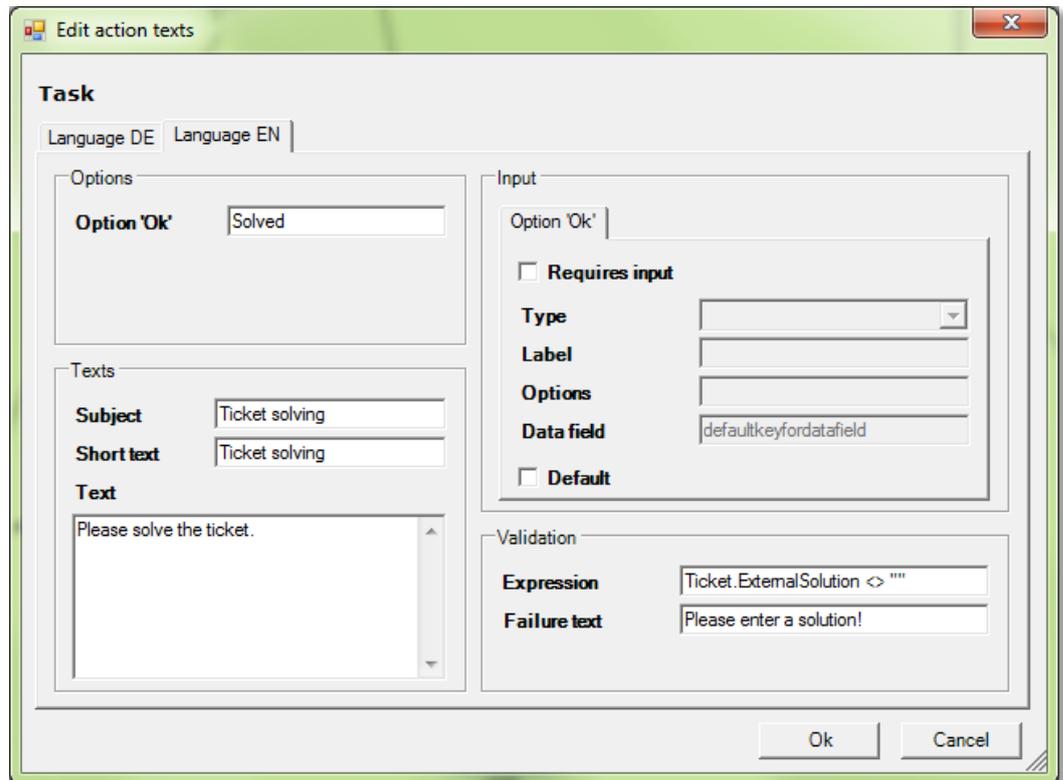


Figure 3-2

When defining workflow activities, expressions can be used for validation. If the validation fails (i.e. the condition phrased in the expression is not fulfilled), the workflow activity will not be processed. In the following example, pressing the **OK** button is only accepted, if the ticket contains a solution.



**Figure 3-3**

### 3.2 Using the Special Data Fields

Special data fields can be used in the workflow, in order to implement the solution of the last user/group action into the following action, for example. These data fields have to be distinct and there should not be multiple data fields of the same name, as the content of a field could be overwritten by another field otherwise.

The data fields are stated during the configuration without any additional characters, as seen in the first figure. If the content of the field is to be displayed, the name of the data field has to be set in double curly braces:

```
{{DataFieldName}}
```

An action of approval is used as an example. For this purpose, two user actions are configured consecutively in the workflow. The first action is to be processed by an approving authority, which is to approve a request.

If the approval is not given, the user has to enter a reason. This reason is displayed as a selection box with three alternatives.

In the following figure, the configuration of this action is displayed:

**Edit action texts**

**Task**

Language DE | Language EN

Options

Option 'Yes' Yes

Option 'No' No

Texts

Subject Approval

Short text Approval

Text

Check the ticket, then approve it.

Input

Option 'Yes' | Option 'No'

Requires input

Type DropDownList

Label Reason

Options High complexity, Budged overdraw

Data field ApprovalReason

Default

Validation

Figure 3-4 Configuration of the ticket action for selecting a reason

After performing this action, the ticket would return from the approving authority to the requester awaiting the result of the approval in order to be able to make a decision concerning the further process. In order not to have to browse several different tabs needlessly to find the solution, the second action is configured in a way to make the solution visible directly in the user task.

**Edit action texts**

**Task**

Language DE | Language EN

Options

Option 'Ok' Ok

Texts

Subject Forward ticket

Short text Forward ticket - Approval den

Text

Approval was denied.  
Reason: {{ApprovalReason}}

Figure 3-5 Configuration of the output

The fields framed in red demonstrate the configuration of the data field. The data field *ApprovalReason* is configured to save the user's selection and to display it later on (or use it in another way).

In the following figure, the data field has been configured for display in the activities list. Thus, the user can see the result of the last action immediately – without having to open the ticket first.



Activities until 28/09/2011	
Ticket id	Activity
mwi_94	Forward ticket - Approval denied
PS_0025	Accept ticket

Figure 3-6 Display of the selection of the last action in the activities list

It is furthermore possible to configure the entry of the reason for approval as a free text field, in which the user can enter the reasons.

Displaying these entries is then possible using the same measures as described in the example of the selection box before.

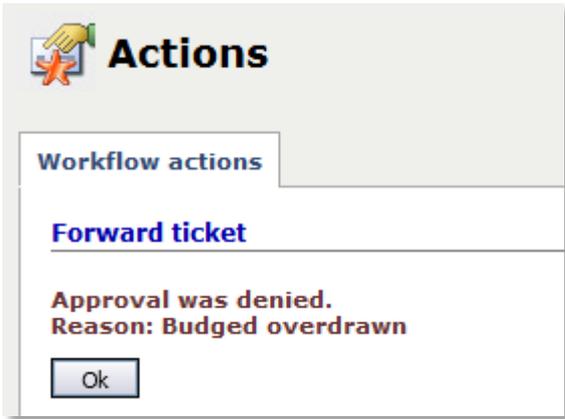


Figure 3-7 Display of a data field, which has been filled via a free text field

## 4 Expressions

The following expressions are available in *Xpert.NET*:

### 4.1 General Expressions

EXPRESSION	DESCRIPTION
<b>General.&lt;Attribute&gt;</b>	Returns general information. The following attributes are available:
CurrentDate	Current date (Format: YYYY-MM-dd)
CurrentTime	Current time (Format: hh:mm)
CurrentDate['<format>']	Current date (formatted); formatting options can be found on the following web page: <a href="http://msdn.microsoft.com/de-de/library/8kb3ddd4.aspx">http://msdn.microsoft.com/de-de/library/8kb3ddd4.aspx</a>
CurrentDate.UTC['<format>']	Current date in UTC (formatted)
CurrentTime['<format>']	Current time (formatted)
CurrentTime.UTC['<format>']	Current time in UTC (formatted)
helpdesklink	URL to <i>TicketXpert/WorkflowXpert</i>
CurrentUser	Access to the current user
CurrentUser.< UserField >	See expressions for the affected user of a ticket.
NewLine	Inserts a line break.

Table 4-1

### 4.2 Expressions for Configuration Items

EXPRESSION	DESCRIPTION
<b>Ci.Affected.&lt;Attribute&gt;</b>	Returns information on the affected user of a CI under specification of an additional attribute. The following attributes are available:
Email	Returns the email address of the affected user.
FirstName	Returns first name of the affected user.
LastName	Returns last name of the affected user.
LoginName	Returns login name of the affected user.
FullName	Returns the full name of the affected user.
MainGroup.<GroupField>	Returns the main group of the affected user (for <GroupField> see the expressions for the affected group (ticket)).
TextBuildingBlock!<Name>.Plaintext TextBuildingBlock!<Name>.HTMLText	Display of the user-dependent text boilerplates as plain text or HTML respectively (substitute <Name> with the boilerplate's name).
<b>Ci.Creator.&lt;User field&gt;</b>	For <User field> see the expressions for affected user.
<b>Ci.Creator.MainGroup.&lt;GroupField&gt;</b>	Returns the creator's main group (for <GroupField> see the expressions for the affected group (ticket))
<b>Ci.Datald</b>	Internal ID of the CI (primary key)
<b>Ci.Fields!&lt;Field name&gt;.&lt;Attribute&gt;</b>	Returns information on CI fields under specification of an additional attribute. The following attributes are available:
Datald	Internal field ID (primary key)
ObjectGuid	Unique field ID for all databases
Text	Text of the specified field
Value	Value of the specified field

<b>Ci.Fields.Count</b>	Number of CI fields
<b>Ci.NumberOfChilds</b>	Number of contained CIs
<b>Ci.ObjectGuid</b>	Unique ID of a CI for all databases
<b>Ci.Owner.&lt;UserField&gt;</b>	For <UserField> see expressions for the affected user.
<b>Ci.Owner.MainGroup.&lt;GroupField&gt;</b>	Returns the owner's main group (for <GroupField> see the expressions in the affected group (ticket)).
<b>Ci.Supplier.&lt;UserField &gt;</b>	For <User field> see the expressions for affected user.
<b>Ci.Supplier.MainGroup.&lt;GroupField&gt;</b>	Returns the supplier's main group (for <GroupField> see the expressions in the affected group (ticket)).
<b>Ci.ParentCi.&lt;Attribute&gt;</b>	Returns information on parent CIs in connection with the attribute. If there are more parents than one, the first one is used.
Schema	Returns the schema of the parent CI.
State	Returns the status of the parent CI.
<b>Ci.Relations.Count</b>	Number of relations of the CI
<b>Ci.Schema.&lt;Attribute&gt;</b>	Returns information on a CI schema in connection with an attribute.
DataId	Internal ID of the CI schema (primary key)
FriendlyName	Name of the schema (language-independent)
IsVirtual	Denotes whether the schema is a virtual schema ( <i>true</i> or <i>false</i> ).
NumberOfChilds	Denotes the number of subschemas.
ObjectGuid	Unique ID of the schema in all databases.
<b>Ci.Status.&lt;Attribute&gt;</b>	Returns information on the status of a CI in connection with an attribute.
DataId	Internal ID of the status (primary key)
FriendlyName	Name of the CI status (language-independent)
ObjectGuid	Unique ID of the status in the entire database
<b>Ci.Uniqueld</b>	Unique ID of the CI visible in the interface
<b>Ci.VisibleId</b>	Visible ID of the CI
<b>Ci.Zone.&lt;Attribute&gt;</b>	Returns information on CMDB zones in connection with an attribute.
DataId	Returns the internal ID of the zone (primary key).
Fields!<FieldName>.DataId	Returns the internal ID of the specified field (primary key). (<Field name> denotes the respective field of a CI zone.)
Fields!<FieldName>.ObjectGuid	Returns the unique ID of the zone field in all databases. (<Field name> denotes the respective field of a CI zone.)
Fields!<FieldName>.Text	Returns the text of the specified field. (<Field name> denotes the respective field of a CI zone.)
Fields!<FieldName>.Value	Returns the value of the specified field. (<Field name> denotes the respective field of a CI zone.)
FriendlyName	Returns the name of the CI zone (language-independent).
NumberOfChilds	Returns the number of direct subzones.
ObjectGuid	Returns the unique ID of the zone in all databases.
ParentZone.<ZoneExpression>	Returns the "parent zone". For <ZoneExpression>, all expressions of <i>Ci.Zone</i> can be used.
Schema.DataId	Returns the internal ID of the zone schema (primary key).
Schema.FriendlyName	Returns the name of the zone schema (language-independent).
Schema.ObjectGuid	Returns the unique ID of the zone schema in all databases.
ZoneType	Returns the zone type (either <i>TopLevelZone</i> or <i>CiZone</i> )

**Table 4-2**

### 4.3 Expressions for Dynamic Placeholders

EXPRESSION	DESCRIPTION
Placeholder.<Name>	Returns the text of the dynamic placeholder (replace <Name> with the placeholder's name).
Placeholder.<Name>.HTML	Returns the text of the dynamic placeholder (replace <Name> with the placeholder's name). Line breaks are replaced with  .

Table 4-3

### 4.4 Expressions for the Early Warning System

EXPRESSION	DESCRIPTION
<b>EarlyWarnAlert.&lt;Attribute&gt;</b>	Returns information on early warning rules in connection with the attribute.
Rule!Name	Returns the name of the early warning rule.
Subscription!Name	Returns the name of the early warning subscription.
Tickets!Url	Returns the URLs of the tickets affected by this rule as plain text.
Tickets!Url.Html	Returns the URLs of the tickets affected by this rule as HTML.
Tickets[<Index>]	Returns the respective ticket (<Index> starts with 0).
TriggerTicket	Returns the ticket that has triggered the early warning system.
TriggerTicket.Ticket#	Returns the number of the ticket that has triggered the early warning system.

Table 4-4

### 4.5 Expressions for Expenses

EXPRESSION	DESCRIPTION
<b>Expense.&lt;Attribute&gt;</b>	Returns information on expenses in connection with an attribute.
DatId	Returns the unique ID of an expense in the database.
ExecutionDate	Returns the execution date of an expense in the format dd.mm.yyyy hh:mm:ss.
<b>Expense.User.&lt;Attribute&gt;</b>	Returns information on the expense creator in connection with an attribute. Instead of the attribute, all expressions for the affected user can be used in the ticket expressions.
<b>Expense.Ticket.&lt;TicketExpression&gt;</b>	Returns information on the ticket linked to the expense in connection with the ticket expression.
<b>Expense.Task.&lt;TaskExpression&gt;</b>	Returns information on the task linked to the expense in connection with the task expression.
<b>Expense.External.&lt;Attribute&gt;</b>	Returns information on external details of an expense in connection with the attribute.
<b>Expense.Internal.&lt;Attribute&gt;</b>	Returns information on internal details of an expense in connection with the attribute.
Title	Returns the (external or internal) title of an expense.
Description	Returns the (external or internal) description of an expense.
Total	Returns the (external or internal) total of an expense.
<b>AllocationType.&lt;Attribute&gt;</b>	Returns information on the allocation type in connection with the attribute.
ObjectGuid	Returns the ObjectGuid of the allocation type.
Title	Returns the name of the allocation type.
Description	Returns the description of the allocation type.
Charge	Returns the charges of the allocation type.

Factor	Returns the factor of the allocation type.
<b>Expense.Project.&lt;Attribute&gt;</b>	Returns information on the project of an expense in connection with the attribute.
ObjectGuid	Returns the ObjectGuid of the project.
Title	Returns the name of the project.
Start	Returns the start date of the project.
End	Returns the end date of the project.
<b>Expense.CostCenter.&lt;Attribute&gt;</b>	Returns information on the cost center used in the expense in connection with the attribute.
ObjectGuid	Returns the ObjectGuid of the cost center.
Number	Returns the number of the cost center assigned in the project.
Title	Returns the title of the cost center.
Start	Returns the start date of the cost center.
End	Returns the end date of the cost center.
EnableTimeAccount	Returns <i>true</i> if the time account has been activated for the cost center, otherwise <i>false</i> .
<b>Expense.Type.&lt;Attribute&gt;</b>	Returns information on the expense type used in an expense in connection with the attribute.
Title	Returns the title of the expense type.
<b>DefaultAllocationType.&lt;AllocationTypeExpression&gt;</b>	Returns information on the default allocation type for the used expense type in connection with the <i>AllocationTypeExpression</i> .

Table 4-5

## 4.6 Expressions for the Knowledge Base

EXPRESSION	DESCRIPTION
<b>Language-independent</b>	
<b>KB.Article.&lt;Attribute&gt;</b>	Returns information on a KB article in connection with the attribute.
ID	Returns the ID of the article.
CreatedDate	Returns the creation date of the article.
ChangedDate	Returns the date of change of the article.
RevisedDate	Returns the revision date of the article.
Creator	Returns the full name of the creator.
Creator.MainGroup.<GroupField>	Returns the main group of the creator (for <GroupField> see the expression for the affected group (ticket)).
Modifier	Returns the full name of the user who has modified the article last.
Modifier.MainGroup.<GroupField>	Returns the main group of the user who has modified the article last (for <GroupField> see the expression for the affected group (ticket)).
Revisor	Returns the full name of the user who revised the article.
Revisor.MainGroup.<GroupField>	Returns the main group of the user who has revised the article (for <GroupField> see the expression for the affected group (ticket)).
KnowledgeOwner	Returns the knowledge owner of the article.
KnowledgeOwner.MainGroup.<GroupField>	Returns the main group of the knowledge owner (for <GroupField> see the expression for the affected group (ticket)).
KnowledgeOwnerGroup	Returns the group the knowledge owner is a member of.

Owner	Returns the owner (knowledge owner) of a KB article as an object.
RatingCount	Returns the number of ratings of a KB article.
RatingAverage	Returns the average rating of a KB article.
VisitsCount	Returns the number of persons having already viewed the article.
ValidStartDate	Returns the start date of the period of validity.
ValidEndDate	Returns the end date of the period of validity.
Faq	Returns <i>true/false</i> , depending on whether the article has been marked as a FAQ article.
Inherit	Returns whether the article is visible in sub-categories as well.
ReviewDate	Returns the date of resubmission of the article.
Url	Returns the direct URL to the article.
<b>KB.Article.Categories.ObjectGuids</b>	Returns a comma-separated list of all ObjectGuids of KB categories an article is in.
<b>KB.Stage.Url</b>	Returns the direct URL to the knowledge base.
<b>Language-dependent</b>	
<b>KB.Article.&lt;Attribute&gt;</b>	Returns language-dependent information on a KB article in connection with the attributes below.
Details.Title!<Language>	Returns the title of the article. (<Language> = the language code of the desired language (EN = English) or <i>All</i> for all published languages) <i>Examples:</i> - KB.Article.Details.Title!EN - KB.Article.Details.Title!EN,FR - KB.Article.Details.Title!All
Details.Keywords!< Language >	Returns keywords of the article. (<Language>: See <i>Title</i> )
Details.ContentHTML!< Language>	Returns the content of the article. (<Language>: See <i>Title</i> )
Details.Boilerplate!< Language>	Returns the text boilerplate of the article. (<Language>: See <i>Title</i> )
<b>KB.Article.Categories.Names</b>	Returns all localized names of KB categories containing the article. The categories are separated as follows: "   " (space pipe space).

**Table 4-6**

## 4.7 Expressions for Messages

EXPRESSION	DESCRIPTION
<b>Message.Attachments.&lt;Attribute&gt;</b>	Returns information on file attachments of messages in connection with the attribute.
Count	Returns the number of file attachments.
Names	Returns a comma-separated list of the names of the file attachments.
Size	Returns the total size of file attachments in Byte.
<b>Message.Body.Plain (*)</b>	Returns the message text of a previous message in plaintext format.
<b>Message.Body.Html (*)</b>	Returns the message text of a previous message in HTML format.
<b>Message.CreatedDateTime</b>	Returns the localized date (and time), on which the e-mail has been processed by the system.
<b>Message.Priority.Name</b>	Returns the priority the initial message has been sent with.

<b>Message.SendDateTime</b>	Returns the localized date (and time) the message initially has been sent at.
<b>Message.Sender</b>	Returns the sender of the initial message.
<b>Message.Subject (*)</b>	Returns the subject of the previous message.

**Table 4-7 Note: (\*) – These expressions are only available for the ticket actions *Response e-mail* and *Forward e-mail***

## 4.8 Expressions for Process Roles

EXPRESSION	DESCRIPTION
<b>ProcessRole.&lt;Attribute&gt;</b>	Returns information on process roles in connection with the attribute.
DatId	Returns the unique ID of the process role in the database.
Description	Returns the description of the process role.
FriendlyName	Returns the language-independent name.
IsDeleted	Returns <i>true</i> , if the process role has been deleted, otherwise <i>false</i> .
Name	Returns the language-dependent name.
ObjectGuid	Returns the unique ID of the process role in the database as a GUID.
ProcessSchemaDatId	Returns the unique ID (database) of the process schema this process role has been assigned to.

**Table 4-8**

## 4.9 Expressions for Tasks

EXPRESSION	DESCRIPTION
<b>Task.Changed</b>	Date and time of the last change
<b>Task.ClientObjectGuid</b>	Returns the unique ID of the client the task has been created in.
<b>Task.Closed</b>	Returns the date and time of closing the task (NULL, if it has not been closed yet).
<b>Task.Comments.&lt;Attribute&gt;</b>	Returns information on task comments in connection with the attribute. Date/time information on comments is specified in the local time.
All!Text	Returns all comments of the task including user and date/time.
Count	Returns the number of comments of the task.
Last!CreatedDateTime	Returns the creation time of the last comment.
Last!Text	Returns the text of the last comment.
Last!User.<UserExpression>	Returns the user who has created the last comment (for User Expression see <i>Ticket.AffectedUser</i> in chapter 4.13 <i>Expressions for Tickets</i> ).
<b>Task.CommentsHtml.All!Text</b>	Returns all comments in HTML format including user and date/time. Date/time information on comments is specified in the local time.
<b>Task.CommentsHtml.Last!Text</b>	Returns the text of the last comment in HTML format (without user and date/time).
<b>Task.CommentsPlainText.All!Text</b>	Returns all comments in plain text format including user and date/time. Date/time information on comments is specified in the local time.
<b>Task.CommentsPlainText.Last!Text</b>	Returns the text of the last comment in plain text format (without user and date/time).

<b>Task.Created</b>	Returns the date and time of the task's creation.
<b>Task.Creator.&lt;UserExpression&gt;</b>	Returns the creator of the task (for user expression, see <i>Ticket.AffectedUser</i> in Section 4.13 <i>Expressions for Tickets</i> ).
<b>Task.Creator.MainGroup.&lt;GroupField&gt;</b>	Returns the creator's main group (for <GroupField> see <i>Ticket.AffectedGroup</i> ).
<b>Task.DataId</b>	Returns the unique ID of the task in the database.
<b>Task.Description</b>	Returns the description of the task.
<b>Task.DueDate</b>	Returns the end date of the task.
<b>Task.Executor.&lt;UserExpression&gt;</b>	Returns the current task executor (for user expression, see <i>Ticket.AffectedUser</i> ).
<b>Task.Executor.MainGroup.&lt;GroupField&gt;</b>	Returns the executor's main group (for <GroupField> see <i>Ticket.AffectedGroup</i> ).
<b>Task.ExecutorGroup.&lt;GroupExpression&gt;</b>	Returns the current task executor group (NULL, if the task has been assigned to a user; for group expression, see <i>Ticket.AffectedGroup</i> ).
<b>Task.ExecutorRole.&lt;RoleExpression&gt;</b>	Returns the current task executor process role (only set, if the task has been created within a process schema; for <RoleExpression> see 4.8 <i>Process Role</i> ).
<b>Task.FullId</b>	Returns the ID of the task as displayed on the interface (e.g. TSK_5).
<b>Task.HasNonClosedSubTasks</b>	Returns <i>true</i> , if there are no open subtasks; otherwise <i>false</i> .
<b>Task.HasNonClosedSuccessorTickets</b>	Returns <i>true</i> , if there are no open successor tickets for the task; otherwise <i>false</i> .
<b>Task.HasSubTasks</b>	Returns <i>true</i> , if the task contains further subtasks; otherwise <i>false</i> .
<b>Task.HasSuccessorTickets</b>	Returns <i>true</i> , if the task contains further successor tickets; otherwise <i>false</i> .
<b>Task.LverDefaultCostCenterObjectGuid</b>	Returns the internal ID (database) of the default cost center of the task (can be NULL).
<b>Task.LverDefaultProjectObjectGuid</b>	Returns the internal ID (database) of the default project of the task (can be NULL).
<b>Task.LverDefaultsAreReadOnly</b>	Returns <i>true</i> , if the Lver default values are read-only; otherwise <i>false</i> .
<b>Task.ObjectGuid</b>	Returns the internal ID (database) of the task (GUID).
<b>Task.OlaTimeSpan</b>	Returns the timespan of the task's OLA (can be NULL).
<b>Task.ParentTask.&lt;TaskExpression&gt;</b>	Returns the parent task, if there is one (for <TaskExpression>, all of the subexpressions in this list can be used, e.g. <i>Task.ParentTask.Description</i> ).
<b>Task.ParentTicket.&lt;TicketExpression&gt;</b>	Returns the parent ticket of the task and solves the ticket expressions accordingly (for <TicketExpression>, see 4.13 <i>Expressions for Tickets</i> ) Examples: - Task.ParentTicket.Status.FriendlyName - Task.ParentTicket.Fields!<FieldName>.Text
<b>Task.Start</b>	Returns the starting time of the task.
<b>Task.TaskAux.&lt;Attribute&gt;</b>	Returns information on the elements linked to the task in connection with the following attributes.
AttachmentsCount	Returns the number of the task's file attachments.
CommentsCount	Returns the number of the task's comments.
ExpenseCount	Returns the number of the task's booked expenses.
SubTasksCount	Returns the number of the subtasks.
TicketsCount	Returns the number of the task's successor tickets.

<b>Task.TaskFolder.&lt;Attribute&gt;</b>	Returns information on task folders in connection with the attribute.
Datald	Returns the unique ID (database) of the task folder.
Description	Returns the description of a task folder.
ObjectGuid	Returns the unique ID (database) of the task folder (GUID).
ParentTaskFolder. <TaskFolderexpression>	Returns the parent task folder (for <TaskFolderexpression>, all of the listed TaskFolder subexpressions can be used, e.g. <i>Task.TaskFolder.ParentTaskFolder.Description</i> ).
Title	Returns the name of the task folder.
<b>Task.TaskLifeCycle.&lt;LifecycleExpression&gt;</b>	Returns the used life cycle (for <LifecycleExpression> see 4.10 <i>Expressions for Task Life Cycles</i> ).
<b>Task.TaskSchema.&lt;SchemaExpression&gt;</b>	Returns the used schema of a task (for <SchemaExpression>, see 4.11 <i>Expressions for Task Schemas</i> ).
<b>Task.TaskStatus.&lt;StatusExpression&gt;</b>	Returns the current status of the task (for <StatusExpression>, see 4.12 <i>Expressions for Task Statuses</i> ).
<b>Task.Title</b>	Returns the title of a task.

**Table 4-9**

## 4.10 Expressions for Task Life Cycles

EXPRESSION	DESCRIPTION
<b>TaskLifeCycle.&lt;Attribute&gt;</b>	Returns information on task life cycles in connection with the attribute.
Datald	Returns the unique ID (database) of the life cycle used by the task.
Description	Returns the description of the life cycle.
FriendlyName	Returns the language-independent name of the life cycle.
StartTaskStatus. <StatusExpression>	Returns the configured start status of the life cycle (for <StatusExpression>, see 4.12 <i>Expressions for Task Statuses</i> ).
StartWorkflow	Returns the GUID (ID in the database) of the workflow configured as a start workflow for the life cycle.

**Table 4-10**

## 4.11 Expressions for Task Schemas

EXPRESSION	DESCRIPTION
<b>TaskSchema.&lt;Attribute&gt;</b>	Returns information on task schemas in connection with the attribute.
Datald	Returns the unique ID (database) of the task schema.
Description	Returns the description of the task schema.
FriendlyName	Returns the language-independent name of the schema.
LifeCycle.<Lifecycleexpression>	Returns the life cycle used in the task schema (for <LifecycleExpression> see 4.10 <i>Expressions for Task Life Cycles</i> ).
Name	Returns the language-dependent name of the task schema.
ObjectGuid	Returns the unique ID (database) of the task schema (GUID).

**Table 4-11**

## 4.12 Expressions for Task Statuses

EXPRESSION	DESCRIPTION
<b>TaskStatus.&lt;Attribute&gt;</b>	Returns information on task statuses in connection with the attribute.
DataId	Returns the unique ID (database) of the status.
FriendlyName	Returns the language-independent name of the status.
StatusTypeDataId	Returns the ID of the status type.
Visible	Returns <i>true</i> , if the status has been marked as visible; otherwise <i>false</i> .

Table 4-12

## 4.13 Expressions for Tickets

EXPRESSION	DESCRIPTION
<b>Ticket.AffectedGroup.&lt;Attribute&gt;</b>	Returns attributes of the affected group. Further attributes can be added to the group. These attributes can then be called via the name of the attribute. (A tool is necessary for adding further attributes. Please contact your project manager or support.)
Name	Returns the name of the affected group.
email	Returns the e-mail address of the affected group.
TextBuildingBlock!<Name>.Plaintext	Returns a group-dependent boilerplate in plain text (replace <Name> with the name of the boilerplate).
TextBuildingBlock!<Name>.HTMLText	Returns a group-dependent boilerplate in HTML format (replace <Name> with the name of the boilerplate).
<b>Ticket.AffectedUser.&lt;Attribute&gt;</b>	Returns information on the affected user. Further attributes can be added to the group. These attributes can then be called via the name of the attribute. (A tool is necessary for adding further attributes. Please contact your project manager or support.)
email	Returns the e-mail address of the affected user.
firstname	Returns the first name of the affected user.
FullName	Returns the full name of the affected user.
lastname	Returns the last name of the affected user.
LoginName	Returns the login name of the affected user.
TextBuildingBlock!<Name>.Plaintext	Returns a user-dependent boilerplate in plain text (replace <Name> with the name of the boilerplate).
TextBuildingBlock!<Name>.HTMLText	Returns a user-dependent boilerplate in HTML format (replace <Name> with the name of the boilerplate).
MainGroup.<GroupField>	Returns user's main group (for <GroupField> see the expressions for the affected group).
ObjectGuid	Returns the ObjectGuid of the affected user.
<b>Ticket.Attachments.&lt;Attribute&gt;</b>	Returns information on file attachments in tickets.
Names.HTML	Returns a list of the file attachment names in HTML format.
Names.PlainText	Returns a list of the file attachment names in plain text.
Count	Returns the number of the ticket's file attachments.
<b>Ticket.CIs.&lt;Attribute&gt;</b>	Returns information on CIs.
Count	Returns the number of the CIs linked to the ticket.
[<Index>].<CiExpression>	Access to the linked CI in the specified index. Value of the respective CI expression (substitute <CiExpression> with one of the CIs' expressions, e.g. <i>Ticket.Cis[0].Affected.FirstName</i> ).
<b>Ticket.Clones.Count</b>	Returns the number of the ticket's clones.

<b>Ticket.Comments.&lt;Attribute&gt;</b>	Returns information on ticket comments in connection with the attributes below. Date/time information on comments is specified in the local time.
All!Text	Returns all comments formatted as saved.
Count	Returns the number of comments.
Last!CreatedDateTime	Returns the creation date of the last comment.
Last!Text	Returns the last comment formatted as saved.
Last!User.<UserField>	Instead of < <i>UserField</i> > all expressions for affected users can be entered.
<b>Ticket.CommentsHtml.All!Text</b>	Returns all comments formatted as HTML (line breaks are replaced with  ). Date/time information on comments is specified in the local time.
<b>Ticket.CommentsHtml.Last!Text</b>	Returns the last comment formatted as HTML (line breaks are replaced with  ).
<b>Ticket.CommentsPlainText.All!Text</b>	Returns all comments formatted as plain text.
<b>Ticket.CommentsPlainText.Last!Text</b>	Returns the last comment formatted as plain text.
<b>Ticket.CompletionDateTime</b>	Returns the completion date.
<b>Ticket.Correspondence.Count</b>	Returns the number of correspondences linked to this ticket.
<b>Ticket.CreatedDateTime.&lt;Attribute&gt;</b>	Returns information on the creation date of the ticket in connections with the attributes below.
['<format>']	Returns the creation time (formatted).
UTC['<format>']	Returns the creation time in UTC (formatted), e.g. ['yyyy/MM/dd HH:mm:ss']. Additionally, date formats supported by the .NET framework are supported as well, e.g.: ['d']: 6/15/2008 ['D']: Sunday, June 15, 2008 ['f']: Sunday, June 15, 2008 9:15 PM ['F']: Sunday, June 15, 2008 9:15:07 PM ['g']: 6/15/2008 9:15 PM ['G']: 6/15/2008 9:15:07 PM ['m']: June 15 ['o']: 2008-06-15T21:15:07.0000000 ['R']: Sun, 15 Jun 2008 21:15:07 GMT ['s']: 2008-06-15T21:15:07 ['t']: 9:15 PM ['T']: 9:15:07 PM ['u']: 2008-06-15 21:15:07Z ['U']: Monday, June 16, 2008 4:15:07 AM ['y']: June, 2008 Some of these expressions depend on the regional settings. If they are used without relation to the user (e.g. in the workflow), the server settings are decisive.
DateDDMMYYYY	Returns the creation time in the format DD.MM.YYYY (e.g. 25.11.2011).
<b>Ticket.CreationMethod</b>	Creation method of a ticket. Possible values: <ul style="list-style-type: none"> <li>• Mail2Ticket</li> <li>• Manual</li> <li>• ContainerTicketWizard</li> </ul>
Mail2TicketAccount.Name	Returns the name of the Mail2Ticket account the ticket has been created with. If the ticket has not been created via Mail2Ticket, the expression returns "".

<b>Ticket.Creator.&lt;Attribute&gt;</b>	Returns the creator of the ticket in connection with the attributes below.
email	Returns the e-mail address of the ticket creator.
firstname	Returns the first name of the ticket creator.
lastname	Returns the last name of the ticket creator.
FullName	Returns the full name of the ticket creator.
LoginName	Returns the login name of the ticket creator.
TextBuildingBlock!<Name>.Plaintext	Returns a user-dependent boilerplate in plain text (replace <Name> with the name of the boilerplate).
TextBuildingBlock!<Name>.HTMLText	Returns a user-dependent boilerplate in HTML format (replace <Name> with the name of the boilerplate).
MainGroup.<GroupField>	Returns user's main group of the ticket creator (for <GroupField> see the expressions for the affected group).
<b>Ticket.CustomAttributes!IsDirectSolve.Value</b>	Possible return values: <i>true</i> and <i>false</i> . If <i>true</i> is returned, the ticket wizard has been closed by clicking the button <i>Solve ticket directly</i> .
<b>Ticket.DataId</b>	Returns the ticket ID from the database (unique ID).
<b>Ticket.Escalations.&lt;Attribute&gt;</b>	Returns information on escalations in connection with the attribute. In contrast to <i>Ticket.PendingEscalations</i> , information on the already achieved or escalated escalations is returned as well.
Count	Returns the number of escalations.
Text	Returns information on the current escalation.
!<EscalationName>.Deadline	Returns the date and the time (localized) of the specified escalation.
<b>Ticket.EstimatedTimeExpense</b>	Returns the estimated time expense for a ticket (if set, otherwise NULL).
<b>Ticket.Expenses.Count</b>	Returns the number of expenses booked on a ticket.
<b>Ticket.ExternalSolution.&lt;Attribute&gt;</b>	Returns the external solution of the ticket in HTML format.
PlainText	Returns the external solution of the ticket as plain text.
PlainText.Length	Returns the number of characters contained in the external solution.
<b>Ticket.Fields!&lt;Name&gt;.&lt;Attribute&gt;</b>	Returns information on ticket fields using the attributes below. (<Name> has to be replaced by the name of the ticket field; ticket fields containing umlauts, spaces, or hyphens are not supported here).
Text	Returns the text value of the respective ticket field formatted as saved.
Ci.<CiExpression>	Returns the value of the CI expression (replace <CiExpression> with an expression for the CIs, e.g. <i>Ticket.Fields!CiList1.Ci.Affected.FirstName</i> ). This expression is only valid, if the ticket field has been configured as a CI list in the Ticket Wizard.
TextHTML	Text value of the respective ticket field formatted as HTML (line breaks are replaced with  ; replace <Name> with the name of the ticket field)
TextPlain	Returns the text value of the respective ticket field formatted as plain text (replace <Name> with the name of the ticket field).
Value	Returns the value of the respective ticket field. This value is ONLY set for selection fields and for <i>Multiple choice fields</i> .
ValuePlain	Returns the value of the respective ticket field as plain text. This is valid for the <i>Multiple choice field</i> as well.

ZonedDateTimeDDMMYY  
 ZonedDateTimeDDMMYYHHMM  
 ZonedDateTimeDDMMYYHHMMSS  
 ZonedDateTimeDDMMYYYY  
 ZonedDateTimeDDMMYYYYHHMM  
 ZonedDateTimeDDMMYYYYHHMMSS

Returns ticket fields with date content in user-defined format. The requirement for using these expressions is the ticket field having the type *DateTime*. Via the expression *ZonedDateTime*, the date is converted from the universal to the correct time zone.

<b>Ticket.Fields!&lt;Name&gt;.User.&lt;Attribute&gt;</b>	This expression allows for accessing user information via a ticket field, if the fields contains data like the ObjectGuid, UserBrowser path, login name, or an e-mail address. Via the attribute, this allows for accessing user information.
Ticket.Fields!<Field>.User.New.Password	Via this expression, the new password of a user can be returned.
<b>Ticket.Fields!&lt;Name&gt;.Group.&lt;Attribute&gt;</b>	This expression allows for accessing group information via a ticket field, if the fields contains data like the ObjectGuid, UserBrowser path, login name, or an e-mail address. Via the attribute, this allows for accessing group information.
<b>Ticket.HasNonClosedSubTasks</b>	Returns the value <i>true</i> , if the ticket contains further open subtasks; otherwise <i>false</i> .
<b>Ticket.HasParent</b>	Checks whether the current ticket has a parent ticket. <i>True</i> if yes, <i>false</i> if no.
<b>Ticket.HasSubTasks</b>	Returns the value <i>true</i> , if the ticket contains subtasks; otherwise <i>false</i> .
<b>Ticket.InternalSolution.&lt;Attribute&gt;</b>	Returns the internal solution of the ticket formatted as HTML.
InternalSolution.PlainText	Returns the internal solution of the ticket formatted as plain text.
<b>Ticket.KB.Count</b>	Returns the number of KB articles linked to this ticket.
<b>Ticket.Mails.Count</b>	Returns the number of messages contained in this ticket.
<b>Ticket.MessageTag</b>	Returns the specific expression to identify the ticket distinctly.
<b>Ticket.ObjectGuid</b>	Returns the Guid of the ticket from the database (unique Guid of the ticket).
<b>Ticket.Owner.Group.&lt;Attribute&gt;</b>	Returns the owner group of a ticket. It is additionally possible to enter the following attributes.
Description	Returns the description of the group.
Name	Returns the name of the group.
TextBuildingBlock!<Name>.Plaintext	Returns a user-dependent boilerplate in plain text (replace <Name> with the name of the boilerplate).
TextBuildingBlock!<Name>.HTMLText	Returns a user-dependent boilerplate in HTML format (replace <Name> with the name of the boilerplate).
<b>Ticket.OriginalMessage.&lt;Attribute&gt;</b>	<p>Returns information on the original message (the first incoming e-mail to a ticket), including ticket images, in connection with the attribute. All parts available for message expressions can be used here as well, e.g.:</p> <ul style="list-style-type: none"> <li>• Ticket.OriginalMessage.Subject</li> <li>• Ticket.OriginalMessage.Body.Html</li> <li>• Ticket.OriginalMessage.Body.Plain</li> </ul> <p>Warning: These expressions can only be used with tickets generated via Mail2Ticket. It is advisable to check in advance whether the ticket actually is such a ticket.</p>
<b>Ticket.Owner.User.&lt;Attribute&gt;</b>	If this value equals NULL, the owner of the ticket is currently a group; if not, it is a user. Additionally, the following attributes can be specified.
< UserField >	All expressions that can be used for the affected user can be used here as well.
MainGroup.<GroupField>	Returns the owner's main group (for <GroupField> see the expressions for the affected group).

<b>Ticket.Parent.&lt;Attribute&gt;</b>	Returns a possible parent ticket and allows access to it. Otherwise, NULL is returned. Additionally, the following attributes are available.
Clones.Count	Returns the number of clones of the parent ticket.
<b>Ticket.PendingEscalations.&lt;Attribute&gt;</b>	Returns information on pending escalations in connection with the attribute.
Count	Returns the number of escalations.
Text	Returns information on the current escalation.
!<EscalationName>.Deadline	Returns the date and the time (localized) of the specified escalation.
<b>Ticket.Priority.&lt;Attribute&gt;</b>	Returns information on ticket priorities using the following attributes.
Description	Returns the description of the priority.
EscalationTimes.Count	Returns the number of escalation times for the current ticket.
EscalationTimes.Text	Returns the escalation times of the current ticket, including the escalation time name and the configured escalation time.
FriendlyName	Returns the name of the priority of the ticket (language-independent).
IsCritical	Denotes whether the priority of the ticket is marked as critical.
Name	Returns the name of the priority.
<b>Ticket.Responsible.User.&lt;Attribute&gt;</b>	Returns attributes of the responsible user. The same attributes as in <i>Ticket.AffectedUser.*</i> can be used here.
<b>Ticket.Responsible.Group.&lt;Attribute&gt;</b>	Returns attributes of the responsible group. The same attributes as in <i>Ticket.AffectedUser.*</i> can be used here.
<b>Ticket.Schema.FriendlyName</b>	For one activated language: name of the ticket schema. For multiple activated languages: unique name of the ticket schema without localization.
<b>Ticket.Service.&lt;Attribute&gt;</b>	Returns information on the services used in the service portfolio in connection with an attribute. For these expressions, it is required to check whether a ticket has been created via the Service Portfolio.
Manager.<Attribute>	Returns the Service Manager group.
Manager.Group.<Attribute>	Returns the Service Manager group. <Attribute> can be replaced by the attributes available for a normal group (e.g. <i>name</i> , see <i>Affected Group</i> ).
HasServiceCi	Returns a Boolean value. <i>True</i> if the superordinate service of the service transaction the ticket has been created in owns a service CI, otherwise <i>false</i> .
ServiceCi.<Attribute>	Returns information on the service CI of the superordinate service of the service transaction the ticket has been created in. All expressions for CIs can be used as attributes.
<b>Ticket.Schema.ProcessRoles! &lt;FriendlyName&gt;.DefaultGroup.&lt;Attribute&gt;</b>	Returns the desired default process role from a process schema along with its properties. <FriendlyName> is to be replaced by the name of the process role. The <Attribute> can be replaced by the available attributes of the group (see <i>Affected Group</i> )
<b>Ticket.ServiceTransaction.ProcessRoles! &lt;FriendlyName&gt;.Group.&lt;Attribute&gt;</b>	Returns the desired process role along with its properties. <FriendlyName> can be replaced by the name of the process role. The <Attribute> can be replaced by the available attributes of the group (see <i>Affected Group</i> )
<b>Ticket.StartDateTime</b>	This expression returns the start time of a ticket.
<b>Ticket.StateObjectGuid</b>	Returns a unique ID (in all databases) of the current ticket status.

<b>Ticket.Status.&lt;Attribute&gt;</b>	Returns information on the current ticket status only in connection with the attributes listed below.
FriendlyName	Returns the friendly name of the current ticket status (language-independent).
Name	Returns the language-dependent name of the current ticket status.
<b>Ticket.Successors.&lt;Attribute&gt;</b>	Returns information on successor tickets in connection with the attributes listed below.
Count	Returns the number of successor tickets
[<Index>]	Access to the successor ticket at the specified index (the index starts at 0). Example: Ticket.Successors1
[<Index>].ObjectGuid	Access to the ObjectGuid of the successor ticket at the specified index (the index starts with 0). Example: Ticket.Successors1.ObjectGuid
<b>Ticket.Ticket#</b>	Returns the ticket number.
<b>Ticket.Title</b>	Returns the title field of the ticket.
<b>Ticket.Url.&lt;Attribute&gt;</b>	Returns the direct URL to the ticket. Additionally, an attribute can be entered.
<ul style="list-style-type: none"> <li>Details</li> <li>Fields</li> <li>UploadFiles</li> <li>Actions</li> <li>History</li> <li>Comments</li> <li>Solution</li> <li>Expenses</li> <li>Overview</li> <li>Todos</li> </ul>	Returns the direct URL to the ticket, focusing on the specified tab.
<b>Ticket.UrlQueryString</b>	Returns the query string for the URL access of the respective ticket (i.e. everything after ? without the actual address to the site).

**Table 4-13**

## 4.14 Expressions for Workflow Items

The following expressions are available for workflow items. However, they can only be used in workflows. These expressions are not case sensitive.

EXPRESSION	DESCRIPTION
<b>WorkflowItem.AffectedGroup.&lt;Attribute&gt;</b>	Returns information on the affected group. All expressions for the affected group can be used as an attribute here.
<b>WorkflowItem.AffectedGroupObjectGuid</b>	Returns the ObjectGuid of the affected group of the workflow item.
<b>WorkflowItem.AffectedUser.&lt;Attribute&gt;</b>	Returns information on the affected user. All expressions for the affected user can be used as an attribute here.
<b>WorkflowItem.AffectedUserObjectGuid</b>	Returns the ObjectGuid of the affected user of the workflow item.
<b>WorkflowItem.ClientObjectGuid</b>	Returns the unique ID of the client this workflow item has been created in.
<b>WorkflowItem.ConnectorObjectGuid</b>	Returns the ObjectGuid of the connector.
<b>WorkflowItem.CreatorUser.&lt;Attribute&gt;</b>	Returns information on the creator. All expressions for users can be used as an attribute here.
<b>WorkflowItem.CreatorUserObjectGuid</b>	Returns the ObjectGuid of the creator of the workflow item.
<b>WorkflowItem.DataId</b>	Returns the data ID of the workflow item.

<b>WorkflowItem.IdDescriptive</b>	Returns the descriptive ID of a workflow item. For a ticket: CR_00367, for example.
<b>WorkflowItem.IdUniqueForType</b>	Returns the unique ID for the type of the workflow item.
<b>WorkflowItem.IsCompleted</b>	Returns whether a workflow item has been completed (Boolean).
<b>WorkflowItem.IsErroneous</b>	Returns whether the item has an error (Boolean).
<b>WorkflowItem.IsSuspended</b>	Returns whether the item has been suspended (Boolean).
<b>WorkflowItem.IsWaitingForJoin</b>	Returns whether the item is waiting for a join (Boolean).
<b>WorkflowItem.LastActionInput</b>	Returns the input of the last action.
<b>WorkflowItem.LastActionResult</b>	Returns the result of the last action.
<b>WorkflowItem.LastException</b>	Returns the last error of the workflow item.
<b>WorkflowItem.ObjectType</b>	Returns the ObjectType of the workflow item.
<b>WorkflowItem.Owner.Group.&lt;Attribute&gt;</b>	Returns information on the owner group. All expressions for the owner group can be used as an attribute here.
<b>WorkflowItem.OwnerGroupObjectGuid</b>	Returns the ObjectGuid of the owner group of the workflow item.
<b>WorkflowItem.Owner.User.&lt;Attribute&gt;</b>	Returns information on the owner. All expressions for the owner can be used as an attribute here.
<b>WorkflowItem.OwnerUserObjectGuid</b>	Returns the ObjectGuid of the owner of the workflow item.
<b>WorkflowItem.PluginObjectGuid</b>	Returns the ObjectGuid of the plug-in the workflow item is currently in.
<b>WorkflowItem.PreviousConnectorObjectGuid</b>	Returns the ObjectGuid of the previous connector.
<b>WorkflowItem.PreviousPluginObjectGuid</b>	Returns the ObjectGuid of the plug-in the workflow item has been in previously.
<b>WorkflowItem.PreviousWorkflowObjectGuid</b>	Returns the ObjectGuid of the workflow the workflow item has been in previously.
<b>WorkflowItem.ProcessSchemaDataId</b>	Returns the data ID of the process schema the workflow item has been created with.
<b>WorkflowItem.SuspendedUntil</b>	Returns the date, until which a workflow item has been suspended.
<b>WorkflowItem.WorkflowObjectGuid</b>	Returns the ObjectGuid of the current workflow of the workflow item.

**Table 4-14**

## 5 Examples

---

### 5.1 Delimitation of Date/Time

In order to query whether a ticket has been created between 07:00 and 18:00, for example, the following expression can be used:

```
Ticket.CreatedDateTime['HH.MM'] > "07.00"&&
```

```
Ticket.CreatedDateTime['HH.MM'] <"18.00"
```

This expression queries, whether the ticket has been created *after* 07:00 and *before* 18:00. Thus, for example, a switch in the workflow can be controlled.

### 5.2 Solve Ticket Directly

The workflow provides an option to query whether the Ticket Wizard has been completed via the button *Solve directly*:

```
Ticket.CustomAttributes!IsDirectSolve.Value = "true"
```

The expression above checks whether the Ticket Wizard has been completed with the button *Solve directly*. In this case, the whole expression returns *true*.

## 6 Operators

### 6.1 Introduction

In conjunction with expressions, operators can be used in order to perform comparisons, apply mathematical operations or connect strings.

An operator is a sign symbolizing a certain function, mainly in the form of an operation. Among these operators are e.g. basic arithmetic operations, relations, and logical operators.

Examples:

- < is less than
- <> does not equal
- + add/combine

On both sides of the operator, the so-called operands can be found. There is one on the left side of the operator (*left hand*) and one on the right side (*right hand*).

In order to process operator functions, it is necessary that both sides have the same data type in most cases.  $5 > \text{Hallo}$ , for example, does generally not make any sense. The result of such an operation is of a certain data type as well. Relations like  $5 > 4$ , for example, result in a logical value (*True* or *False*). The result data type of an arithmetic operation can, however, consist of several different data types. It depends on the input data types and the arithmetic operation performed.

The tables starting from Chapter 6.3 display an overview of the possible operations with matching examples. Please remember that, in order to have easily displayable characters, some operators in TXP/WXP can differ from the generally known ones. Moreover, there are operators mostly originating in the field of IT.

### 6.2 Using the Operators

Operators can be used for the display of data, for example. This option is available in message templates, workflow actions for users and groups, and for workflow switches depending on such an expression as well.

A short example will demonstrate how operators can be used in a workflow action:

Texts

**Subject**

**Short text**

**Text**

Correspondence: {\$ Ticket.Correspondence.Count \$}  
E-mails: {\$ Ticket.Mails.Count \$}  
Comments: {\$ Ticket.Comments.Count \$}

---

Sum: {\$ Ticket.Correspondence.Count +  
Ticket.Mails.Count +  
Ticket.Comments.Count\$}

Figure 6-1 Configuring the workflow action

The following data has been entered into the text field in the figure above:

*Correspondence: {\$ Ticket.Correspondence.Count \$}*

*E-mails: {\$ Ticket.Mails.Count \$}*

*Comments: {\$ Ticket.Comments.Count \$}*

-----

Sum: {\$Ticket.Correspondence.Count + Ticket.Mails.Count + Ticket.Comments.Count\$}

In the ticket, this configuration is displayed as follows:

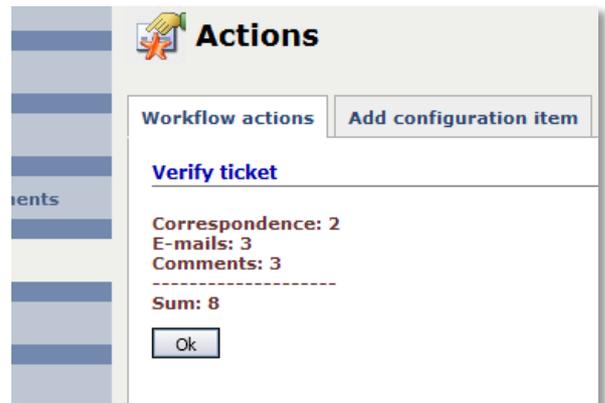


Figure 6-2

The example shows that when using operators, the start token {\$ and the end token \$} can only be used at the beginning and at the end of the entire operation. If the expression token is closed after every expression, each expression is processed separately, which will result in no operation.

### 6.3 Relational Operators

Relational operators generally have the result type *Boolean* (logical value), which can have the following three values:

- True
- False
- Null

These relational operators are often used in switch plug-ins in the workflow. There, tickets can take different directions in the workflow – depending on the expression.

OPERATOR	RESULT TYPE	RESULT
<=	Boolean	[Left] is less than or equals [right]
	<i>Example:</i>	
	Ticket.CreatedDateTime <= Ticket.CompletionDateTime	
<>	Boolean	[Left] does not equal [right]

	<i>Example:</i> Ticket.CreatedDateTime <> Ticket.CompletionDateTime	
>=	Boolean	[Left] is greater than or equals [right]
	<i>Example:</i> Ticket.CreatedDateTime >= Ticket.CompletionDateTime	
=	Boolean	[Left] equals [right]
>	Boolean	[Left] is greater than [right]
<	Boolean	[Left] is less than [right]

Table 6-1

## 6.4 Operators for Strings

One part of the operators for strings is their combination. All data types that can be displayed as strings in any way can be combined.

➔ In one expression chain, only similar operators can be used – thus either && or ||. It is not possible to use && and || simultaneously.

OPERATOR	RESULT TYPE	RESULT
&	String	[Left] is combined with [right]
	<i>Note:</i> Unnecessary characters (e.g. space characters, line breaks, etc.) are removed at the beginning and at the end of every of string before combining the strings. <i>Example:</i> Ticket.AffectedUser.firstname & ", " & Ticket.AffectedUser.lastname  Result example: Max, Muster	
+	String	[Left] is combined with [right]
	<i>Note:</i> Unnecessary characters (e.g. space characters, line breaks, etc.) are removed at the beginning and at the end of every of string before combining the strings.  <i>Example:</i> Ticket.AffectedUser.lastname + ", " + Ticket.AffectedUser.firstname  Result example: Muster, Max	
~+	String	[Left] is combined with [right]
	<i>Note:</i> Unnecessary characters are not removed when using this operator.  <i>Example:</i> Ticket.AffectedUser.lastname ~+ ", " ~+ Ticket.AffectedUser.firstname  Result example: Muster, Max	

Table 6-2

## 6.5 Arithmetic Operators

OPERATOR	LEFT HAND	RIGHT HAND	RESULT TYPE	RESULT
/	TimeSpan	Double Integer	TimeSpan	TimeSpan([left].Ticks/[right])
<i>Note:</i> TimeSpan is a special data type. It denotes the time span between two dates. The result type is converted into the type TimeSpan again here.				
/	Double Integer	Double Integer	Double Integer	[Left] is divided by [right]
<i>Note:</i> The result type is determined by the more precise data type in this constellation. If the most precise data type is an integer, the result type is an integer as well. If the most precise data type is a double (and thus more precise than an integer), the result type is a double as well.				
*	TimeSpan	Double Integer	TimeSpan	TimeSpan([left].Ticks * [right])
<i>Note:</i> TimeSpan is a special data type. It denotes the time span between two dates. The result type is converted into the type TimeSpan again here.				
*	Double Integer	Double Integer	Double Integer	[Left] multiplied by [right]
<i>Note:</i> The result type is determined by the more precise data type in this constellation. If the most precise data type is an integer, the result type is an integer as well. If the most precise data type is a double (and thus more precise than an integer), the result type is a double as well.				
-	TimeSpan	TimeSpan	TimeSpan	[Right] is subtracted from [left]
-	DateTime	TimeSpan	DateTime	[Right] is subtracted from [left]
-	IsoDateTime	TimeSpan	DateTime	[Left].UtcDateTime - [right]
-	Double Integer	Double Integer	Double Integer	[Right] is subtracted from [left]
+	Double Integer	Double Integer	Double Integer	[Left] and [right] are added
<i>Note:</i> If the data type of one operand is a string type, the combination is used. <i>Example:</i> Ticket.CustomAttributes.Count + 1				
~+	Double Integer	Double Integer	Double Integer	[Left] and [right] are added
<i>Note:</i> If the data type of one operand is a string type, the combination is used.				

**Table 6-3**

## 6.6 Logical Operators

OPERATOR	MEANING	RESULT
&&	and	With this operator, two expressions for relations can be combined.
	<i>Example:</i> Ticket.AffectedUser.lastname = "Muster" && Ticket.AffectedUser.firstname = "Max" Here, both expressions have to be fulfilled ( <i>true</i> ) in order to fulfill the combined expression with <i>true</i> .	
	or	With this operator, two expressions for relations can be combined.
	<i>Example:</i> Ticket.Owner.User.lastname = "Muster"    Ticket.Owner.User.lastname = "Tester" Here, only one of the two sub-expressions has to be fulfilled in order to fulfill the combined expression.	

Table 6-4

## 6.7 Brackets

When working with expressions, brackets can be used as well. Thus arithmetic operations and complex expressions containing multiple expression conditions can be compiled. The brackets will be interpreted in the same way as is known from mathematics or computer languages.

### Examples

The expression examples are printed in bold and a short explanation can be found below them.

**$((2+3)*4)+1$**

The result after the evaluation of the expression is 21.

**(Ticket.Fields!MyField1.Text = "1" || Ticket.Fields!MyField2.Text = "2") &&  
Ticket.Fields!MyField3.Text = "3"**

The ticket field *MyField3* has to have the value 3, here. Moreover, one of the conditions within the brackets has to be fulfilled (thus either the field *MyField1* has to have the value 1 or the field *MyField2* has to have the value 2).

## 7 Statistics and Change Management

### 7.1 Statistics

Created by	Marco Mehl
Creation date	29.12.2005
Doc-ID	DOC-041212-010
Version	3.8-2
Status	Approved
Replaces version	3.8-1
Release date	13.3.2014
Valid from	Immediately
Valid until	Cancellation
Document name	Expression Reference 3.8-2_EN.docx

### 7.2 Change Management

VERSION	DATE	EXECUTED BY	COMMENT
1.0	29.12.2005	Thomas Luck	
1.1	11.01.2006	Alexander Schmidt	Adapted to new layout
1.2	22.02.2006	Alexander Schmidt	
1.3	28.02.2006	Thomas Luck	
1.4	15.06.2006	Alexander Schmidt	
1.4a	15.06.2006	Alexander Schmidt	
1.5	23.07.2006	Thomas Luck	
1.6	07.09.2006	Ralph Pfister	
1.7	29.11.2006	Thomas Luck	
1.8	08.12.2006	Thomas Luck	
1.9	21.12.2006	Ralph Pfister	Knowledgebase and Tasktool
1.10	11.01.2007	Michael Schuler	Tasktool
1.11	11.07.2007	Luca Marinucci	
1.12	26.10.2007	Thomas Luck	
1.13	02.05.2008	Marco Mehl	Layout and Additionsn (General, Ticket- and Ci-Expressions, Operators)
2.0	06.05.2008	Marco Mehl	Additional documentation (Operators and data fields)
2.1	08.05.2008	Marco Mehl	Minor non-conformities corrected
2.2	08.08.2008	Marco Mehl	Expressions for UTC Date/Timeadded
2.3	19.08.2008	Marco Mehl	Logicaloperatorsadded, diversification
2.4	08.01.2009	Marco Mehl	Two new CI-expressions added, CI-expressions corrected
2.5	20.03.2009	Marco Mehl	Message-expressions and example added
2.6	23.03.2009	Luca Marinucci	Expression added, to access logged in users

2.7	24.06.2009	Marco Mehl	UrlQueryString-Expression added; Ticket.Status.FriendlyName modified and Ticket.Status. Name added
2.8	14.09.2009	Maik Wisatzke	Layout changed
2.9	03.11.2009	Marco Mehl	General.NewLine, Ticket.DataId, Ticket.ObjectGuid, Ticket.Successors.Count added; removed 2 deprecated expressions
2.10	20.04.2010	Marco Mehl	Advice for accessing ticket fields added; Example added; Expression for Escalation-deadline added; expression for ExpenseCount added
3.0	18.08.2010	Maik Wisatzke	Adjustments for version 3.4
3.1	03.11.2010	Marco Mehl/Maik Wisatzke	4 new expressions for comments; obsolete CI-expressions deleted; new CI-expressions added; 2 early warning expressions added
3.2	29.04.2011	Marco Mehl	Hint for expression chains, task expressions revised
3.3	30.06.2011	Marco Mehl	Task.Title added, new messaging expressions added, EarlyWarnAlert.TriggerTicket.Ticket# corrected
3.4	18.08.2011	Marco Mehl	Ticket.Attachments.Count added; Beschreibung für added the use of brackets; Expressions for clones added; Expressions for process roles and Service Manager added
3.5	19.09.2011	Marco Mehl/Maik Wisatzke	Added ParentTicket Expression to Task Expressions; Added new figures
3.6	05.10.2011	Marco Mehl	Added MainGroup-Expression; Added EarlyWarnAlert.Subscription!Name
3.7	04.11.2011	Marco Mehl	Added expressions for task comments
3.8	29.12.2011	Maik Wisatzke	Added the Ticket.OriginalMessage expression, minor non-conformities corrected
3.9	09.05.2012	Maik Wisatzke/Steffi Kurnot	Restructuring of the expressions, Added expense expressions, and several ticket and service portfolio expressions
3.10	11.07.2012	Maik Wisatzke/Stephanie Henze	Added Workflowitem Expressions; upgraded ticket escalation expressions; upgraded CiParentCI expressions
3.11	04.12.2012	Maik Wisatzke/Steffi Kurnot	Added date formats
3.8-1	13.02.2013	Maik Wisatrke	Corrections
3.8-2	12.03.2014	Anna Hajduk	Kb.Article.Owner added; CiZoneDateTime Expressions removed; linguistic revision

**Copyright © 2014**

**isonet ag, Engineering und Beratung, Zürich**

This document is intellectual property of isonet AG and protected by copyright. It must not be saved, copied, duplicated, or passed on photo mechanically, digitally, or through any other means. Nor must it be used for the execution of projects. The companies directly addressed in this project have the right to use this document for their purposes exclusively within the scope of this offer.