Data Control in Workflow

Workflow Designer

IBM FileNet Business Process Manager

Using expressions in a workflow

- An expression is a formula or set of symbols that returns a single result and can be used when you design the following: – Data fields definitions
  - –Routing conditions
  - –Workflow field assignments
  - Expression parameters
- A simple expression is a single variable or a literal.
  - –A variable can be one of the supported simple data types or an array.
- A complex expression is a combination of workflow fields, system fields, operators, literals, and functions.
- Expressions can also be used at run time, when searching for work in Process Administrator.

Data types and arrays

- An array is one-dimensional list of elements of the same simple data type.
  - Must be of type Boolean, float, integer, string, or time
- Array elements are delimited by braces: { }
  - Example: {13.29, 98.34, 1.5} is a float array with three elements
- Arrays must always have at least one element (which might be empty).
- Array size is adjusted automatically to contain all elements.
  - Indexes assigned out of sequence create all intermediate elements
- Specialized functions are available that work only with arrays.
  - Example: **`elementcount(array_field)`**returns the number of non-null and empty elements in array_field

Literals and operators in expressions

- Literals of all data types, except time, can be used
  - –Boolean examples: **`true false`** (These values are case-sensitive.)
  - –Float examples: **`3.25 0.2536 -1.5e-25`**
  - –Integer example: **`477`**
  - –String example: **`"a string field"`** (Quotation marks are not considered part of the string.)
  - –Array example: **`{1.25, 3.67, 333.2}`**
- Operators can be used when building expressions and are evaluated in order of preference
  - –Parentheses () for logical grouping
  - –Arithmetic operators: **`+ - / *`**
  - –String concatenation: **`+`**
  - –Relative operators: **`< > = <= >=`**
  - –Logical operators: **`not, and, or, like, is null, is not null`**

Functions

- System-provided functions are available to build expressions

–**Not** the same as the system functions used in a system step and found in the General System palette (example: Assign step)

- Selected examples of available functions

  –General functions

- **`if (boolean_expr, expr2, expr3)`**
- **`max (expr_1, expr2_, ... expr_n)`** –String functions
- **`len (string_expression)`** returns the length of a string

  –Data type conversion functions

- **`convert (source_exp, type_name)`**

  –Time functions

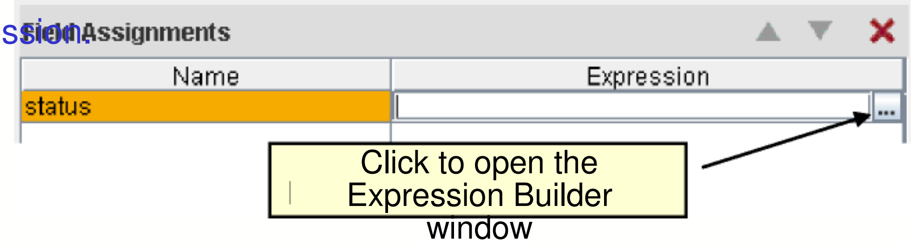
- **`adddays(time_expr, number_of_days)`** and similar time units

Data type conversion

- Functions are available for explicit data type conversion
  - Convert an expression from one type to another
- Example: **`convert (0, boolean)`** evaluates to false
  - Convert using a mask: number to string, string to time, time to string
- Example: **`numbertostring (Num1, "###.##")`**
  evaluates to the value 45.33 if Num1 = 45.3394
  - Determine if an expression is a valid value that can be converted
- Example: **`is_valid ("34.55", float)`** evaluates to true
- Implicit conversion occurs for certain data type combinations to ensure that the data type satisfies the operator or function
  - Float to integer conversion
- Example: 123.57 assigned as the initial value of an integer data field is converted to 123
  - Integer to float conversion
- Example: 2 is converted to 2.0 in the expression 10.75 + 2

Expression Builder

- Expression Builder window is used to create complex expressions. –Access is available from a field where complex expressions are permitted. –You can choose from several options to develop the expression.

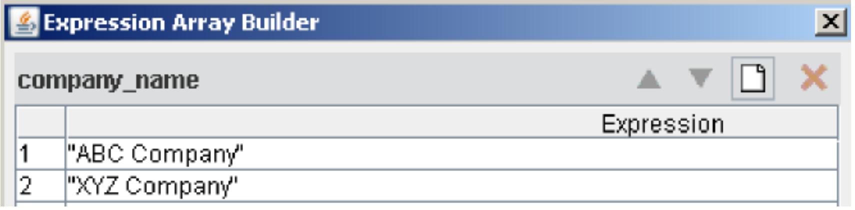


Field Assignments

Name	Expression
status	

Click to open the Expression Builder window

- To set initial values of array fields, use the Expression Array Builder window to build an expression for each element in an array.



Expression Array Builder

company_name

	Expression
1	"ABC Company"
2	"XYZ Company"

Expression examples

- Time evaluation
  - `adddays(systemtime(),7)` returns a time 7 days ahead of current time
  - `weekday(systemtime())` returns an integer representing the day of the week for current time
- Array evaluation
  - To test for equivalency of two array elements use `array1_name[1] = array2_name[1]`
  - Do **not** use `array1_name = array2_name` because it results in a malfunction at run time.
- String concatenation operator (+)
  - Example where customer_name is a string data field: `"The order is for " + customer _ name + "."`

- Verify assignment of a single attachment
  - Create an empty_attachment field and make a comparison:

```
if (attachment_field1 = empty_attachment,
"Attachment not assigned", "Attachment is assigned")
```

Step parameters and workflow data fields

- You can control how data is used at each step by defining the following:
  - Which data fields are used as parameters at a step
  - Access rights to each step parameter at runtime
  - How data fields are assigned
- System data fields are generally assigned automatically by the system.
  - Some can be explicitly assigned.
  - Example: F_Subject

Select step parameters and access rights

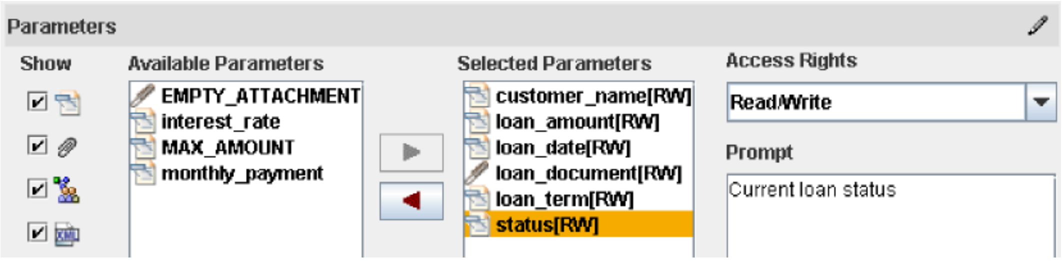
- In step properties Parameters tab, you select which parameters can be used in the step.

–Assign access rights to control whether the current parameter value is displayed or can be assigned.

–Options to display types of parameters:

- Data fields
- Attachments
- Workflow groups
- XML fields (used for Web Services operations)

–Prompt is a tool tip displayed for the user in the Workplace XT step processor.



Parameters

Show

☑

☑

☑

☑

Available Parameters

- EMPTY_ATTACHMENT
- interest_rate
- MAX_AMOUNT
- monthly_payment

Selected Parameters

- customer_name[RW]
- loan_amount[RW]
- loan_date[RW]
- loan_document[RW]
- loan_term[RW]
- status[RW]

Access Rights: Read/Write

Prompt: Current loan status

Exposed data fields

- The workflow administrator exposes data fields in a queue, roster, or event log.
- An exposed field
  - Is readable in its native format without opening the work item in a step processor
  - Is available to use in a search filter, to define an index, and to store information in an event log
  - Changes the way work items are contained and used
- Why expose data fields?
  - To create indexes from the fields that can be used to track, search, and sort
  - To export field values to other IBM FileNet components, such as Process Analyzer and Rules Engine

Exposed Data Fields window

- Access in Process Designer Workflow Properties window

–Helps achieve consistency in exposed field names –Field

names and data types must match

- To use the Exposed Data Fields window

1.Open Workflow Properties and select the Data Fields tab.

2.On the Data Fields tab, click the Exposed Data Fields icon.

The Exposed Data Fields window opens and lists all exposed fields defined by the administrator and committed in the region.

1.To see where a field is defined, select a field and click Details.

2.Choose from the list to create a workflow data field of the same name and data type as the exposed field.

Assignment before and after a step

- Two assignment options are available in step properties that are executed relative to the current step.
  - Assignment before execution
  - Assignment after completion
- Access from step properties pane
  - Select Assignments tab > Before Execution or After Completion.
  - Select the workflow field to assign and then use the Expression Builder window to define the expression.
- Use these options as an alternative to the Assign system function.

Example: Assign F_Comment after step completion

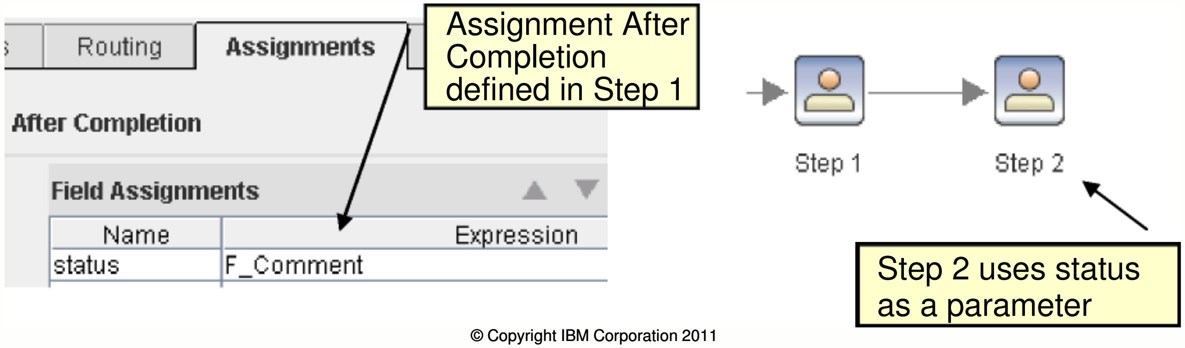
- Use case scenario

–You want to display user comments from one step to users who process the next step.

–You do **not** want to expose all previous comments or history. –View History option is not suitable for this case.

- Design solution example

–Use Assignment After Completion option and a data field to persist the F_Comment field from Step 1 to Step 2.



Name	Expression
status	F_Comment

Assignment After Completion defined in Step 1

Step 2 uses status as a parameter

Limitation of the Assign system function

- The Assign system function **cannot** be used to assign any of the transitory system fields that have a value only at a step.
  - Examples: F_Responses and F_Comment have null values outside of a step.
  - If you use them in an Assign step, the assigned values do **not** persist to the next step.
- No Process Designer validation error or runtime error occurs.

Workflow participants

- A workflow participant is any user or group who is assigned to process steps in a workflow.
- When you assign a step to specific participants you have direct control over which participant processes the step.
  - Decision of who processes the work is made at **design time.**
- When you assign a step to a work queue, the work might be processed by any one of the users who has processing access rights to the queue.
  - Decision of who processes the work is made at **run time.**
- Use a workflow group in the step destination to provide flexibility of runtime assignment along with direct control over who is allowed to process the work.

Workflow groups

- Placeholders for one or more users or groups

–Define the group in Workflow Properties > Workflow Groups tab. –Assign group members at design time or dynamically at run time. –**Not** the same as an LDAP group

–Used in Process Designer to assign the step destination for a step

- Use case and benefits –Provide flexibility to assign participants at design time or run time

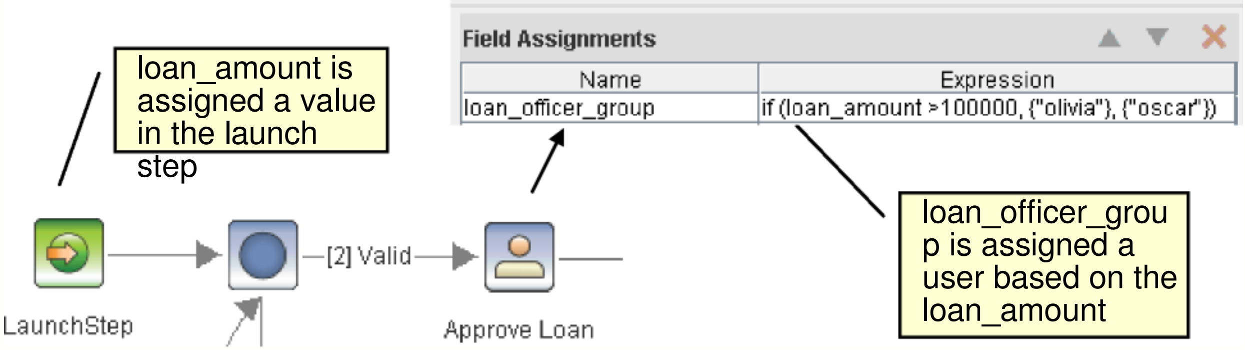
–Create a workflow group for a collection of users who perform a particular job function in a workflow

–For example, supervisors or loan officers

- At run time, the group contains unique values for each workflow.

Example: Dynamic assignment of workflow groups

- Use case scenario
  - Loan processing workflow contains an Approve Loan step.
  - A specific loan officer is assigned to approve, based on the amount.
- Design solution example
  - The loan_officer_group workflow group is assigned as step destination for the Approve Loan Amount step.
  - Before execution of the step, a specific loan officer is assigned, based on the loan_amount data field value.



Name	Expression
loan_officer_group	if (loan_amount >100000, {"olivia"}, {"oscar"})

loan_amount is assigned a value in the launch step

loan_officer_group is assigned a user based on the loan_amount

System-provided workflow group: F_Trackers

- Members are assigned in Workflow Properties > Workflow Groups.
- All assigned members are workflow trackers.
- Members can also be assigned at run time using Process Administrator and the full access view of Process Tracker.
- Consider assigning at least one tracker for a workflow definition.
  - To monitor workflow and help resolve runtime problems

System-provided workflow group: F_Originator

- Represents the name of the user who launched the workflow
- Can be used in assignment of the step destination for a step
- System automatically assigns value at launch time
- Not generally used for workflows launched by the system (for example, through a workflow subscription)
  - If you assign a step to F_Originator in a workflow launched using a workflow subscription, the work item is assigned to the Content Engine administrative user.

Set participant privileges

- In General step properties, you can control how a workflow participant interacts with the workflow at a step.
- For a particular step, you can allow a workflow participant to do the following:
  - –Reassign work to another user.
  - –View status using Process Tracker.
  - –View history using the History window in the step processor.

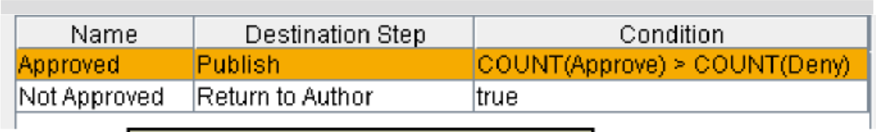
Participant voting

- A multiparticipant step with responses specified
  - All participants must complete the step before responses are evaluated.
  - You can base route conditions on the aggregated outcome of the participant responses.
- Use COUNT in the route conditions to test the number of participant responses.
  - Examples:
- COUNT (Approve) > COUNT (Reject)
- COUNT (Reject) >= 1

Example: Counting responses

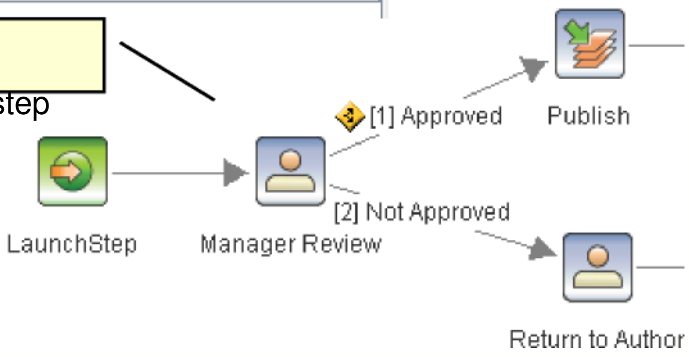
- Responses for the Manager Review step
  - –Are specified in the order [1] Approve, [2] Deny
- Two outgoing routes from Manager Review step
  - –[1] Approved route is taken if COUNT (Approve) > COUNT (Deny) –

Otherwise, route [2] Not Approved is taken



Name	Destination Step	Condition
Approved	Publish	COUNT(Approve) > COUNT(Deny)
Not Approved	Return to Author	true

Outgoing Routing Information for Manager Review step


[1] Approved
Publish
LaunchStep
Manager Review
[2] Not Approved
Return to Author