
Workflow Automation 3.4

Process Authoring Guide

July 2018

Legal Notice

Workflow Automation powered by Aegis is protected by United States Patent No(s): 5829001, 5999178, 6708224, 6792462.

© Copyright 2007-2018 Micro Focus or one of its affiliates.

The only warranties for products and services of Micro Focus and its affiliates and licensors ("Micro Focus") are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Contents

About this Book and the Library	5
1 Getting Started with Workflow Automation	7
1.1 What is Workflow Automation?	7
1.2 How Workflow Automation Works	8
1.3 Process Author Checklist	9
1.4 Starting the Configuration Console the First Time	9
2 Understanding Events	11
2.1 Understanding Base Events	11
2.2 Understanding Named Events	12
2.3 Understanding Scheduled Events	12
2.3.1 Understanding Event Schedule Types	12
2.3.2 Understanding Event Schedule Activation	13
2.3.3 Understanding Timetables and Time Ranges	14
2.3.4 Scheduled Event Example	15
3 Understanding Triggers	23
3.1 Understanding Trigger Operators	24
3.2 Understanding Trigger Expressions	24
3.2.1 Understanding Initiation Expressions	25
3.2.2 Understanding Blocking Expressions	31
3.2.3 Understanding Appending Expressions	34
3.3 Using the “Is Not Equal To” Trigger Comparison	37
3.4 Understanding Trigger Custom Attributes	38
3.5 Revising Existing Triggers	40
3.6 Understanding Abstract Trigger Revisions	40
3.7 Understanding the Manual Trigger	41
4 Working with Processes and Workflows	43
4.1 Understanding Process Terminology	44
4.2 Understanding Workflow Activities	45
4.3 Understanding the Input Builder	46
4.4 Understanding User Input Forms	47
4.4.1 Linking to a User Input Form	47
4.4.2 Customizing a User Input Form	48
4.5 Understanding Workflow Connectors	48
4.5.1 Configuring a Connector	49
4.5.2 Understanding Conditional Expressions	49
4.5.3 Understanding How Join Activities Interact with Conditional Connectors	51
4.6 Working with Strings in Workflow Activities	53
4.7 Creating a Process	53
4.8 Workflow Examples	54
4.8.1 Find the Management Server with the Fewest AppManager Agents	54
4.8.2 Creating a Snapshot on a Virtual Machine	61
4.8.3 Creating a New User Account	65

4.9	Debugging a Workflow	75
4.10	Working with Times in Workflow Automation	76
4.11	Validating a Workflow	77
4.12	Understanding Supporting Analysis	77
4.13	Understanding Workflow Priority	77
	4.13.1 Trigger Processing Order	78
	4.13.2 Event Consumption Policy	78
	4.13.3 Setting the Policies for a Workflow	78
4.14	Understanding Revision Control	79
	4.14.1 Putting a Workflow Revision into Production	79
	4.14.2 Cloning a Workflow Revision	79
	4.14.3 Taking a Workflow Revision Out of Production	80
	4.14.4 Exporting a Workflow	80
	4.14.5 Importing a Workflow	80
	4.14.6 Customizing Work Item Properties for a Workflow	81
	4.14.7 Customizing Work Item Attributes	81
4.15	Deploying Activity Module Updates	82
5	Understanding Activity Libraries	83
5.1	Creating a Custom Library	83
5.2	Customizing an Existing Activity Library	84
5.3	Importing and Exporting Activity Libraries	84

About this Book and the Library

The *Process Authoring Guide* provides conceptual information about the Workflow Automation product (powered by Aegis). This book defines terminology and various related concepts. It also provides step-by-step guidance for many process-related tasks.

Intended Audience

This book provides information for individuals responsible for any of the following tasks:

- ♦ Understanding Workflow Automation concepts
- ♦ Designing and implementing processes that represent your IT policies

NOTE: Workflow Automation processes can interact with various monitoring products, such as AppManager or Security Manager. Process Authors should have a thorough understanding of the monitoring products they use in a process.

Other Information in the Library

The library provides the following information resources:

Administrator Guide

Provides conceptual information related to installing Workflow Automation, using the Configuration Console, and step-by-step guidance for many configuration tasks.

User Guide

Provides conceptual information related to the Operations Console and step-by-step guidance for many Process Operator tasks.

Reporting Center Reporting Guide

Provides conceptual information about the Reporting Center product. Intended for individuals responsible for understanding and using Workflow Automation reports.

Help for Configuration Console

Provides context-sensitive information and step-by-step guidance for common tasks, as well as definitions for each field on each window.

Help for Operations Console

Provides conceptual information and step-by-step guidance for common tasks.

1 Getting Started with Workflow Automation

IT operations managers are under increasing pressure to control costs while delivering services at a faster pace than ever before. In the quest to get “more for less,” many IT professionals are exploring ways to automate time-consuming, labor-intensive tasks that increasingly occupy valuable and expensive IT staff resources.

Many enterprises seek to automate routine workflow practices at the level of specific tasks and procedures, sometimes referred to as **run books**. While run books represent proven manual processes, they are subject to human error and can be very expensive by wasting expert staff time on repetitive, menial tasks. Run books can also be abstract, representing “tribal knowledge” that resides with individuals. Tribal knowledge may not be accurately recorded and can be easily lost when individuals leave an organization.

At the same time, IT operations deal with an increasingly diverse and complex IT environment due to highly distributed operations and business acquisitions. Often, these operational environments contain many diverse tools. The lack of integration for these tools can hinder the ability to efficiently respond to events and conduct routine operational activities.

We designed Workflow Automation from the ground up as an IT workflow automation platform, to enable the automation of IT run books and processes.

1.1 What is Workflow Automation?

Workflow Automation, powered by Aegis, is an IT process automation platform that allows you to model, automate, measure, and continuously improve run books and processes. Workflow Automation allows you to:

- ◆ Define automated IT processes, using your documented IT processes and run books as the foundation for automation
- ◆ Automatically execute process steps on behalf of personnel
- ◆ Coordinate work between different IT functions

Integrating with other enterprise products*, Workflow Automation retrieves normalized information from your operations management systems to provide a cohesive “big picture” view of your IT operation processes.

* Integration products include adapters for Databases, Directory and Resources Administration, SNMP Activity, VMware vCenter Server, LDAP, and ServiceNow.

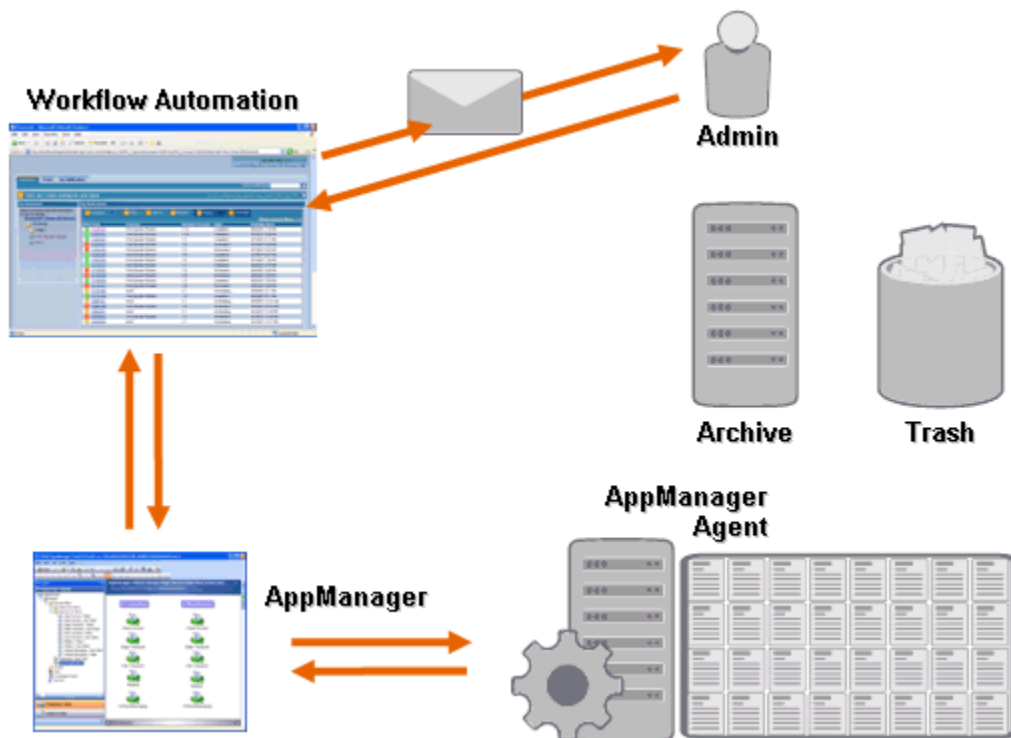
By aggregating this information into a central console, Workflow Automation lets IT operations personnel more easily accomplish the following objectives:

- ◆ Automatically initiate standard IT processes to address incidents
- ◆ Document all response activities
- ◆ Report on adherence to accepted IT policies
- ◆ Provide a complete audit history of incident management

By allowing users to quickly identify and automatically group and suppress related events, Workflow Automation reduces the number of false positives users must address and manage.

1.2 How Workflow Automation Works

Workflow Automation funnels information from data sources into a central repository, evaluating incoming data against processes that your team defines to match your IT policies. A process runs automatically when event information matches its trigger criteria. If user input is required, the user logs on to the Operations Console to tell the process what to do. For example, if temporary file growth causes disk space to fall below threshold, Workflow Automation can command AppManager to perform disk cleanup.



In this example, AppManager detects that available disk space has fallen below threshold and generates an event, which triggers a process in Workflow Automation and creates a **work item** (an instance of a running process). Workflow Automation requests a disk usage analysis from AppManager, identifying the top N culprits by folder, file type, and age, with extra attention paid to known temporary file storage areas. Workflow Automation sends an email with the results of the analysis to the appropriate user, requesting approval to perform disk cleanup. The email includes a link to the Operations Console. The user logs on to the Operations Console and approves partial cleanup. Workflow Automation commands AppManager to delete the approved files and analyze new disk space status. Workflow Automation waits for confirmation of success, which it then sends to the user. Workflow Automation closes the work item.

Workflow Automation processes are flexible. A typical process may automatically complete the following types of steps:

- ◆ Check for auxiliary information about the event
- ◆ Determine the priority and urgency of the problem
- ◆ Allow users to drill down for details about the event stream

1.3 Process Author Checklist

Process Authors are responsible for creating and maintaining processes and their associated workflows. After the Workflow Automation Administrator has completed the initial configuration tasks, the Process Author can start creating triggers and process workflows in Workflow Automation. Use the following checklist to help you complete these tasks.

	Checklist Items
<input type="checkbox"/>	1. For triggers that reference specific types of events, create named events. For more information, see Understanding Named Events .
<input type="checkbox"/>	2. Create triggers. For more information, see Understanding Triggers .
<input type="checkbox"/>	3. Create new processes. For more information, see Creating a Process .
<input type="checkbox"/>	4. If you are responsible for ensuring servers are resources in Workflow Automation, set up resources and business services. For more information about resources, see the Workflow Automation Administrator Guide .

1.4 Starting the Configuration Console the First Time

The Configuration Console is a Windows-based application delivered to your desktop using Microsoft ClickOnce publishing and deployment technology. To start the console the first time, launch it from the Operations Console. The initial ClickOnce deployment creates a shortcut in the NetIQ program group.

NOTE: If your environment does not allow ClickOnce installation, you can install the Configuration Console locally on any computer. For more information about installing the Configuration Console locally, see the [Workflow Automation Administrator Guide](#).

To launch the Configuration Console from the Operations Console:

- 1 If you are logged on to the Web Server computer, on the Start menu, click **All Programs > NetIQ**, and then click **NetIQ Aegis Operations Console**.
- 2 If you want to log on to the Operations Console remotely, complete the following steps:
 - 2a Start Internet Explorer.
 - 2b Specify the following URL, where *hostcomputer* is the name of the Web Server computer:
`http://hostcomputer/Aegis`
- 3 On the Operations Console Login window, provide your user name and password.

NOTE: If you do not have a user name and password for Workflow Automation, or if you do not know the name of the Web Server computer, contact your Workflow Automation Administrator.

- 4 To ensure the local computer has the prerequisites for installing the Configuration Console, complete the following steps:
 - 4a Click **Configuration Console Prerequisites**.
 - 4b If you need to install the prerequisites, click **Install Prerequisites**.
 - 4c Close the Aegis Configuration Console Prerequisites window.

- 5 Click **Launch Configuration Console**.
- 6 On the Configuration Console Login window, provide your user name and password.

2 Understanding Events

An event initiates an action in a process by matching one of the following:

- ♦ **Trigger** - Triggers evaluate events to determine whether to start a process or append the event to an existing work item (an instance of a running process). A single event might trigger multiple processes. A combination of events can trigger a single process.
- ♦ **Wait for Event activity** - The Wait for Event activity responds to an event that occurs while a process is running. Typically, each adapter has a customized version of the activity to match events from that adapter. For example, if your Workflow Automation environment has multiple adapters, you might see a Wait for Email Event activity or a Wait for AppManager Event activity.

Workflow Automation provides the following event types:

- ♦ Base event
- ♦ Named event
- ♦ Scheduled event

2.1 Understanding Base Events

A base event is an event from Workflow Automation or a Workflow Automation adapter. A base event has a list of available attributes with no pre-defined values. You can associate base events with trigger expressions and named event expressions to match incoming events. When you associate a base event with a trigger expression or named event expression, you specify the values for specific attributes you want to compare to attribute values on incoming events.

Workflow Automation provides the following base events:

- ♦ Aegis.Event
- ♦ Aegis.Generic Integration Event
- ♦ Email.Message
- ♦ Exchange.New Message Event
- ♦ Scheduler.Event

Each Workflow Automation adapter provides its own base events. For more information about the base events specific to an adapter, see the [NetIQ Aegis Adapter Documentation](#). For more information about trigger expressions, see [Understanding Trigger Expressions](#).

2.2 Understanding Named Events

A **named event** (also known as a **triggering event definition**) is a custom event you create with specific attribute values. For example, a named event can check for AppManager events with a severity of `Emergency` and from a specific Knowledge Script. When Process Authors create triggers, they can include named events in the expressions governing when the trigger starts, suppresses, or groups a work item. For more information about triggers, see [Understanding Triggers](#).

By default, a trigger does not require one event to meet all specified trigger expressions. A trigger requires only that *one or more* events meet all specified expressions *within the specified time window*. You can use named events in some trigger expressions to ensure Workflow Automation initiates a work item when one specific event meets all conditions. For more information about trigger expressions, see [Understanding Trigger Expressions](#).

You can edit a named event that is not associated with a workflow revision currently in production at any time. To edit a named event associated with a workflow revision in production, you must take the workflow revision out of production. After you edit the named event, you can put the workflow revision back in production. Workflow Automation checks for updates to named events when you put a workflow revision in production. For more information about taking a workflow revision out of production, see [Taking a Workflow Revision Out of Production](#).

2.3 Understanding Scheduled Events

Scheduled events allow you to initiate a work item at a specific date and time. For example, if your company sets up, manages, and maintains servers for multiple customers, you can create a workflow that:

- 1 Starts at the beginning of a maintenance period for a particular set of servers.
- 2 Places the servers in maintenance mode.
- 3 Performs routine maintenance, such as performing a backup or rebooting a server.
- 4 Waits until the maintenance period ends.
- 5 Puts the servers back into production mode.

2.3.1 Understanding Event Schedule Types

Workflow Automation provides the following types of event schedules:

One Time	The event schedule generates an event once at the specified date and time.
Recurring	The event schedule generates an event on a regular basis, such as the last Friday of every month. You can also configure the event schedule to generate an event multiple times on the specified day, such as every hour from 10:00 AM to 2:00 PM.
Start/End of Time Range	The event schedule generates an event at the beginning or end of one or more specified time ranges. The selected time ranges determine whether the event schedule is recurrent. You can also configure the event schedule to generate an event at both the beginning and the end of the specified time ranges.

2.3.2 Understanding Event Schedule Activation

When an event schedule is active, it generates an event by adding a task to the Task Scheduler in Microsoft Windows, which sends the event to Workflow Automation at the appropriate date and time. By default, an event schedule is active only if a workflow revision in production references the event schedule by name in one of the following activities:

- ◆ Start of Workflow activity (in rules for triggers or named events)
 - ◆ **If you want to reference an event schedule by name in a trigger**, use either of the following expression templates:
 - ◆ Scheduled event
 - ◆ Any event of a specific type and attribute value
 - ◆ **If you want to reference an event schedule by name in a named event**, specify the `Scheduler.Event` event type and the **Event Schedule Name** event attribute.
- ◆ Wait for Scheduled Event activity

Workflow Automation activates or deactivates the event schedule automatically when you change the status of an eligible workflow revision.

You can configure an event schedule to be active at all times. The **Keep event schedule active if Aegis is shut down** check box allows the event schedule to generate events under the following conditions:

- ◆ The Aegis Namespace Provider service is not running. For example, if you modify settings in the Adapter Configuration Utility, you must restart the Namespace Provider service.
- ◆ No workflow revisions in production reference the event schedule.

If the event schedule generates an event under either of these conditions, the event waits in Microsoft Message Queuing. When the Namespace Provider service is running again, Workflow Automation receives the event and displays it in the Events list in the Operations Console. If a workflow revision that references the event schedule is in production, Workflow Automation initiates a work item.

NOTE

- ◆ Active event schedules generate events only if they are enabled. If you disable an event schedule, the event schedule does not generate events even if it is configured to be always active.
 - ◆ If you do not configure an event schedule to be always active, the event schedule is available “on demand.” If you modify an on demand event schedule while a workflow revision in production is using it, the changes do not take effect until the event schedule is no longer in use.
 - ◆ If you configure an event schedule to be always active in a cluster environment, to ensure the event schedule generates events in the case of a failover, you need to perform a manual failover to each node in the cluster.
-

2.3.3 Understanding Timetables and Time Ranges

A **timetable** is a collection of one or more time ranges. For example, a timetable called `National Holidays` has a time range for each holiday in the year, such as `New Years Day`.

Using Time Ranges to Restrict an Event Schedule

You can use time ranges to restrict when an event schedule generates events.

Positive Constraints

The event schedule generates an event only if the scheduled occurrence is within a specified time range. For example, if you want a scheduled event to occur every day except for weekends, you can add a timetable called `Business Days`, which has a time range for each weekday, as a positive constraint. The scheduled event occurs Monday through Friday, but not on weekends.

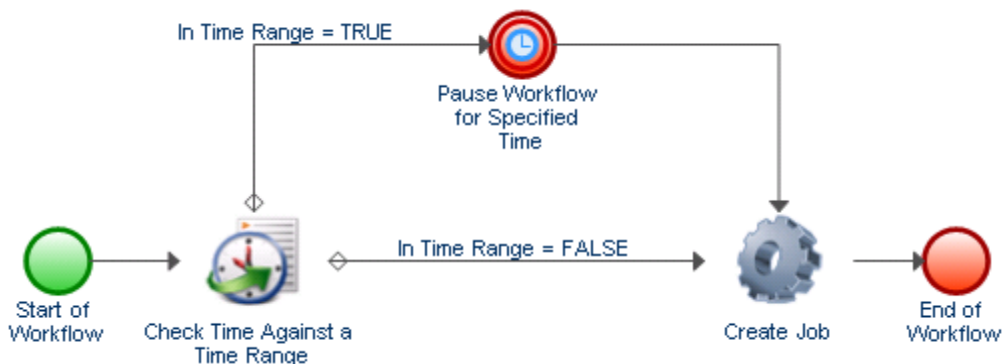
Negative Constraints

The event schedule does not generate an event if the scheduled occurrence is within a specified time range. For example, if you want a scheduled event to occur every Friday except for holidays, you can add a timetable called `National Holidays`, which has a time range for each holiday in the year. The scheduled event does not occur on any Friday that is a holiday.

Using Time Ranges in Workflows

Time ranges allow the `Check Time Against a Time Range` activity to check whether a specified time falls within one or more pre-defined time ranges. The activity's **In Time Range** output parameter is either `TRUE` or `FALSE`. Process Authors can use the output parameter in another activity at a later point in the workflow or in a conditional connector to determine the path of the workflow immediately after the `Check Time Against a Time Range` activity.

For example, a Process Author places a `Check Time Against a Time Range` activity immediately after a `Start of Workflow` activity and configures it to retrieve the current time and check it against a timetable that contains time ranges related to national holidays. The Process Author configures a conditional connector to connect the `Check Time Against a Time Range` activity to a `Pause for Specified Time` activity. When an event triggers the workflow, the `Check Time Against a Time Range` activity checks the current time against the specified time range and finds that the current time is on New Year's Day. The `Pause for Specified Time` activity delays the flow until the next business day.



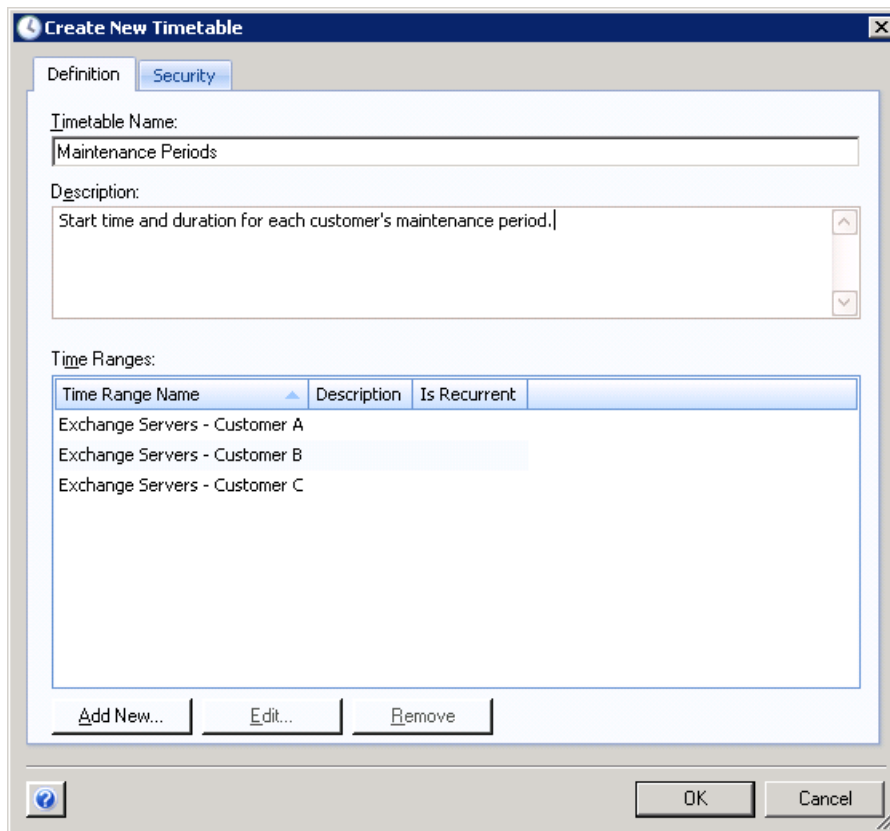
2.3.4 Scheduled Event Example

The following example provides an overview of the major steps involved in setting up a scheduled event and associating it with a trigger:

- 1 Create a Timetable
- 2 Create a Time Range
- 3 Create an Event Schedule
- 4 Create a Named Event
- 5 Create a Trigger
- 6 Create a Process

1. Create a Timetable

Create a timetable called `Maintenance Periods` that contains a time range for each client's maintenance period, such as `Exchange Servers - Customer A`, which is the same name your environment uses for the resource group that contains the computers.



2. Create a Time Range

Create the `Exchange Servers - Customer A` time range, and specify when the maintenance period starts, how long it lasts, and how often it recurs.

Edit Time Range

Time Range Name: Exchange Servers - Customer A

Description: Recurring maintenance time range for Customer A

Time Zone: (GMT-06:00) Central Time (US & Canada)

Start Time: 2:00 AM

End Time: 4:00 AM

Duration: 2 hours

Is Recurrent

Recurrence Frequency: _____

Occurs: Monthly

Day 1 of every 1 month(s)

The first Saturday of every 1 month(s)

Recurrence Duration: _____

Start Date: 10/30/2010

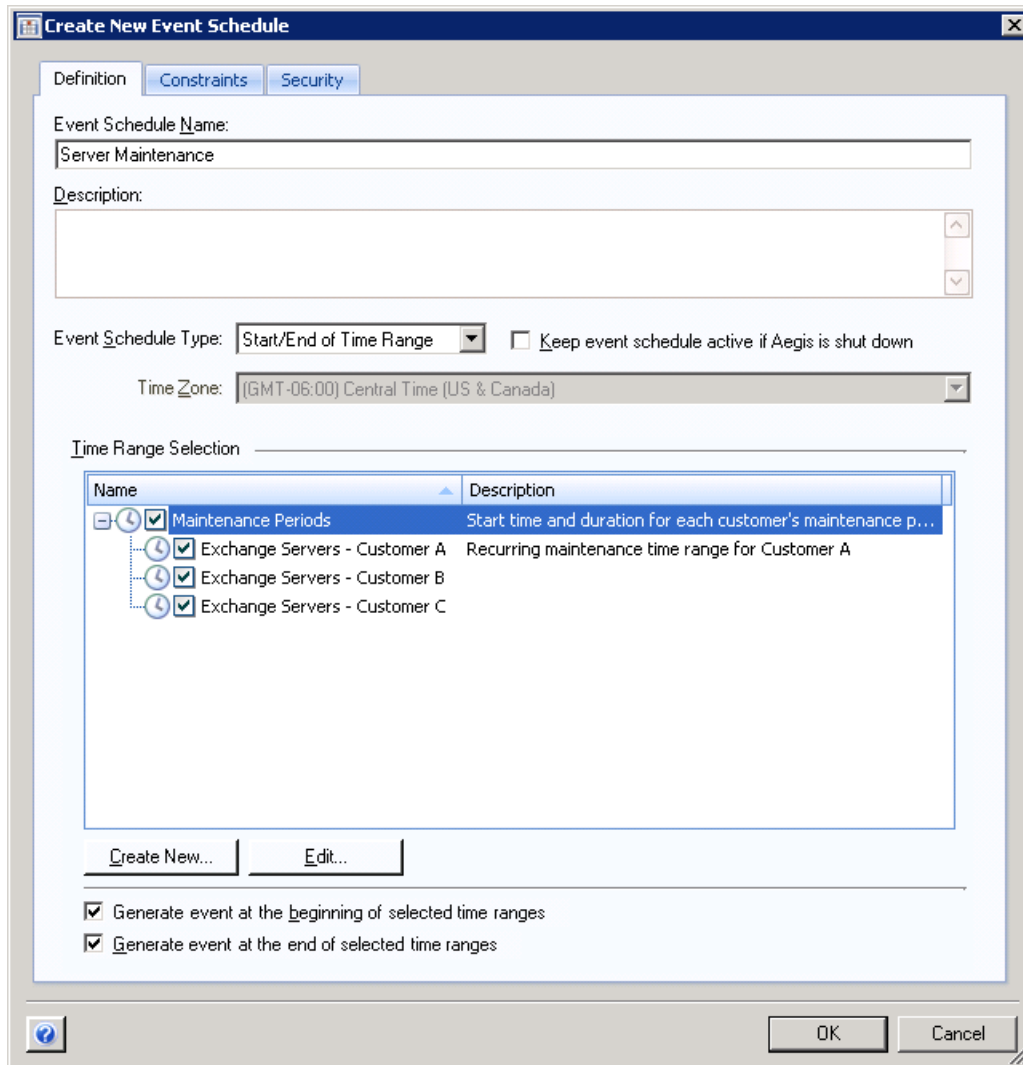
End Date: 11/ 2/2010

No end date

OK Cancel

3. Create an Event Schedule

Create an event schedule called `Server Maintenance`, and configure it to generate an event at the start and end of each time range in the `Maintenance Periods` timetable.



The event schedule generates an event by adding a task to the Task Scheduler in Microsoft Windows, which sends the event to Workflow Automation at the appropriate date and time.

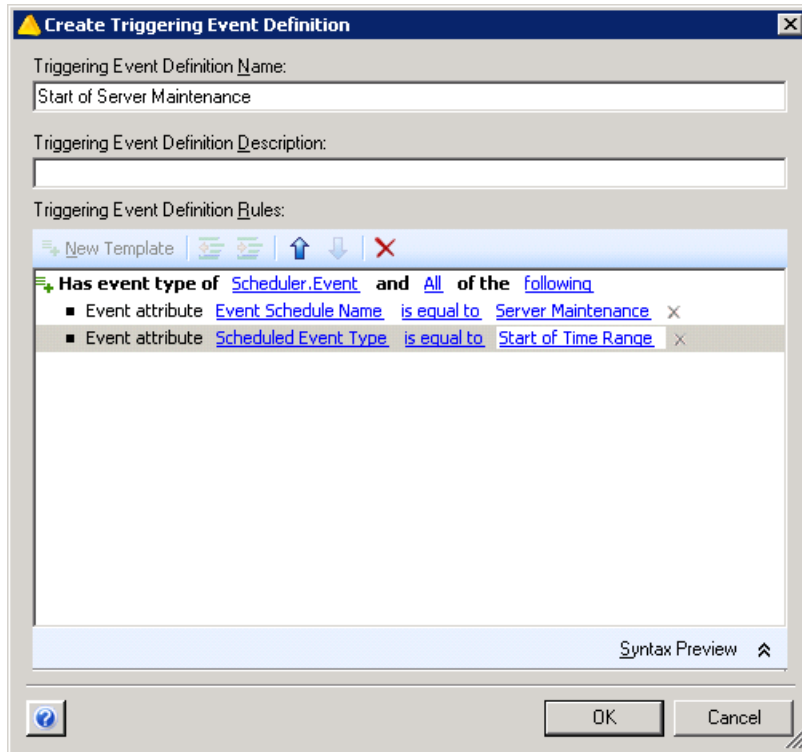
4. Create a Named Event

Create a named event called Start of Server Maintenance with the following settings:

Has event type of Scheduler.Event and All of the following

Event attribute Event Schedule Name is equal to Server Maintenance

Event attribute Scheduled Event Type is equal to Start of Time Range

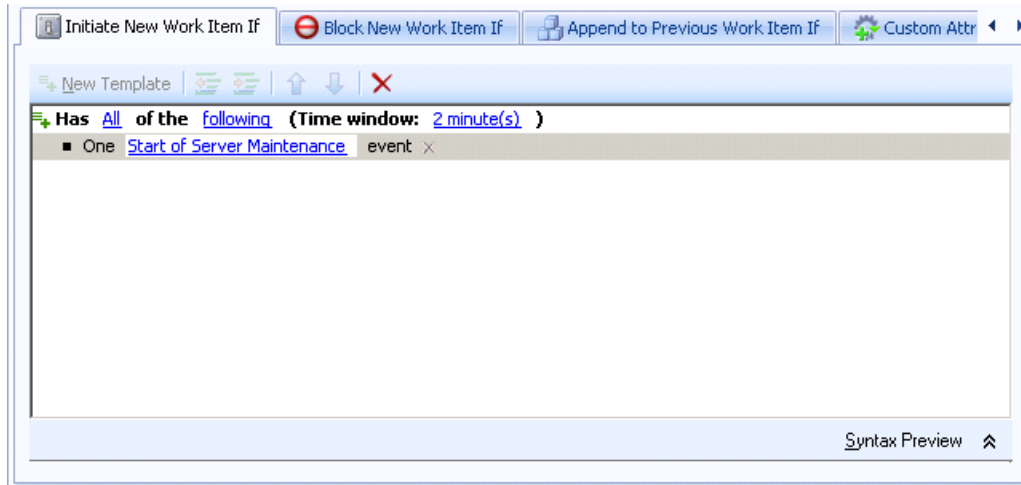


For more information about named events, see [Understanding Named Events](#).

5. Create a Trigger

Create a trigger called `Scheduled Server Maintenance`.

On the `Initiate New Work Item If` tab, use the `One` event of a `specific type` expression template to match the `Start` of `Server Maintenance` named event.



On the `Custom Attributes` tab, create a custom attribute that retrieves the `Time Range Name` attribute from the event that initiates the work item. Ensure the following values are set:

	Custom Attribute
Name	TriggeringEventTimeRange
Display Name	Time Range of Triggering Event
Type	String
Value	Use the Input Builder to retrieve the <code>Time Range Name</code> attribute from the event that initiates the work item. For more information about using the Input Builder, see Understanding the Input Builder .

For more information about triggers, see [Understanding Triggers](#). For more information about custom attributes, see [Understanding Trigger Custom Attributes](#).

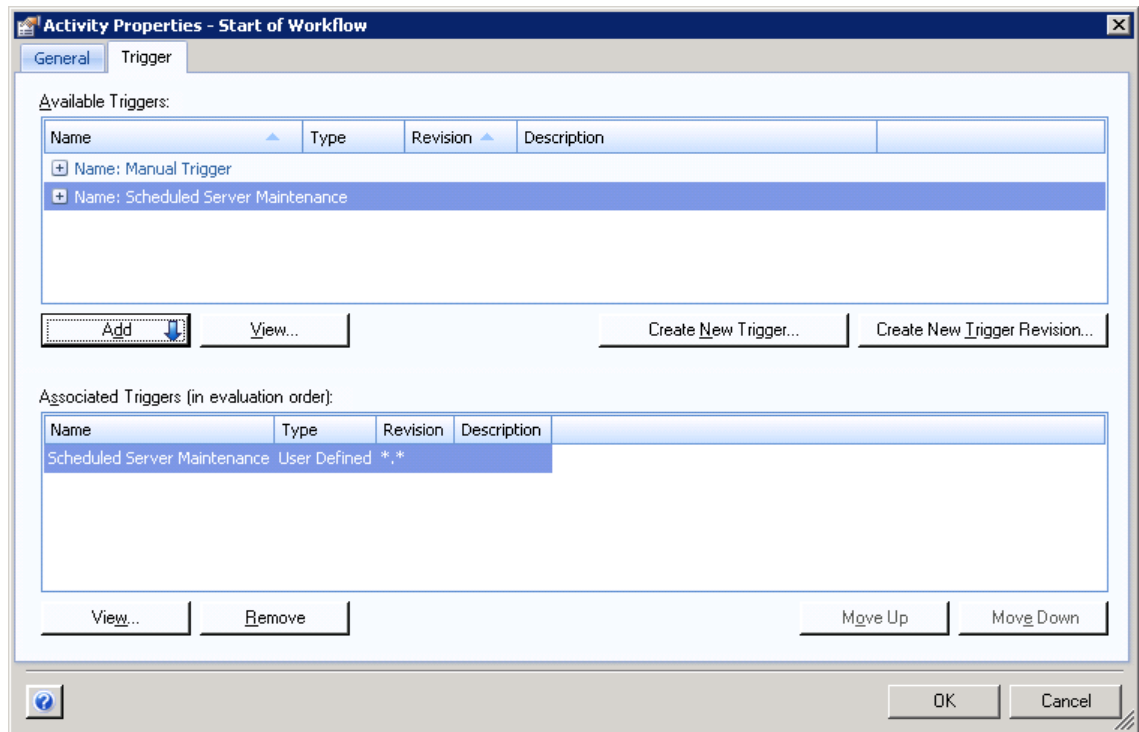
6. Create the Process

Create a new process and open the Workflow Designer.

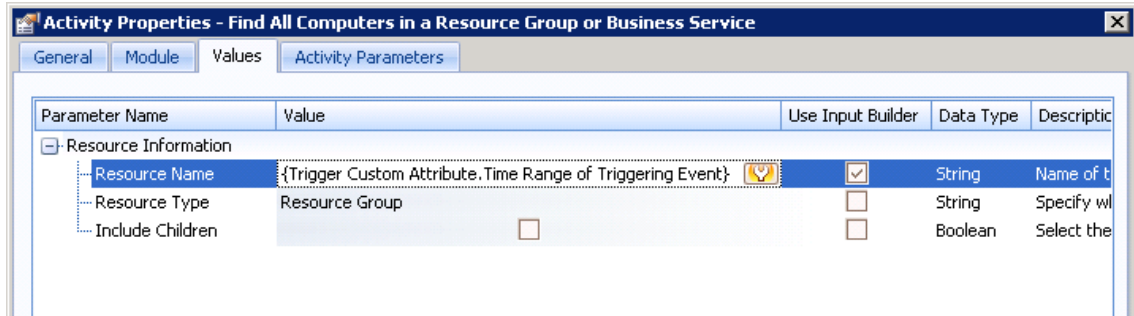
- 1 In the Library Browser, drag the following activities from the specified activity libraries to the design grid:

Activity	Library
Start of Workflow	Basic Workflow Control
Find All Computers in a Resource Group or Business Service	Resources Management
Set Ad Hoc Maintenance Mode	NetIQ AppManager
Wait for Scheduled Event	Times and Dates
Clear Ad Hoc Maintenance Mode	NetIQ AppManager
End of Workflow	Basic Workflow Control

- 2 In the Start of Workflow activity, add the trigger you created.

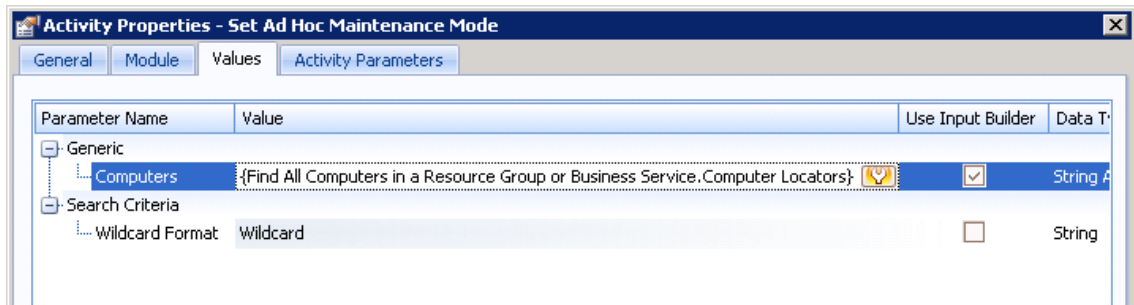


- In the Find All Computers in a Resource Group or Business Service activity, for the **Resource Name** parameter, use the Input Builder to retrieve the custom attribute you created on the trigger.



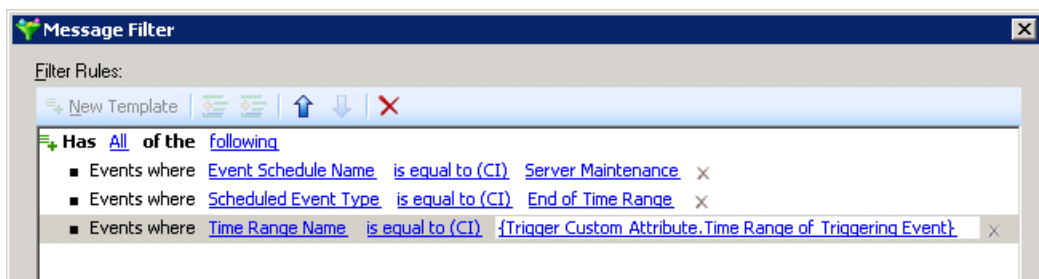
The activity retrieves a list of all computers in the resource group with the same name as the time range associated with the triggering event. The list of computers is available in an output parameter. For more information about resource groups, see the [Workflow Automation Administrator Guide](#).

- In the Set Ad Hoc Maintenance Mode activity, for the **Computers** parameter, use the Input Builder to retrieve the **Computer Locators** output parameter from the Find All Computers in a Resource Group or Business Service activity.



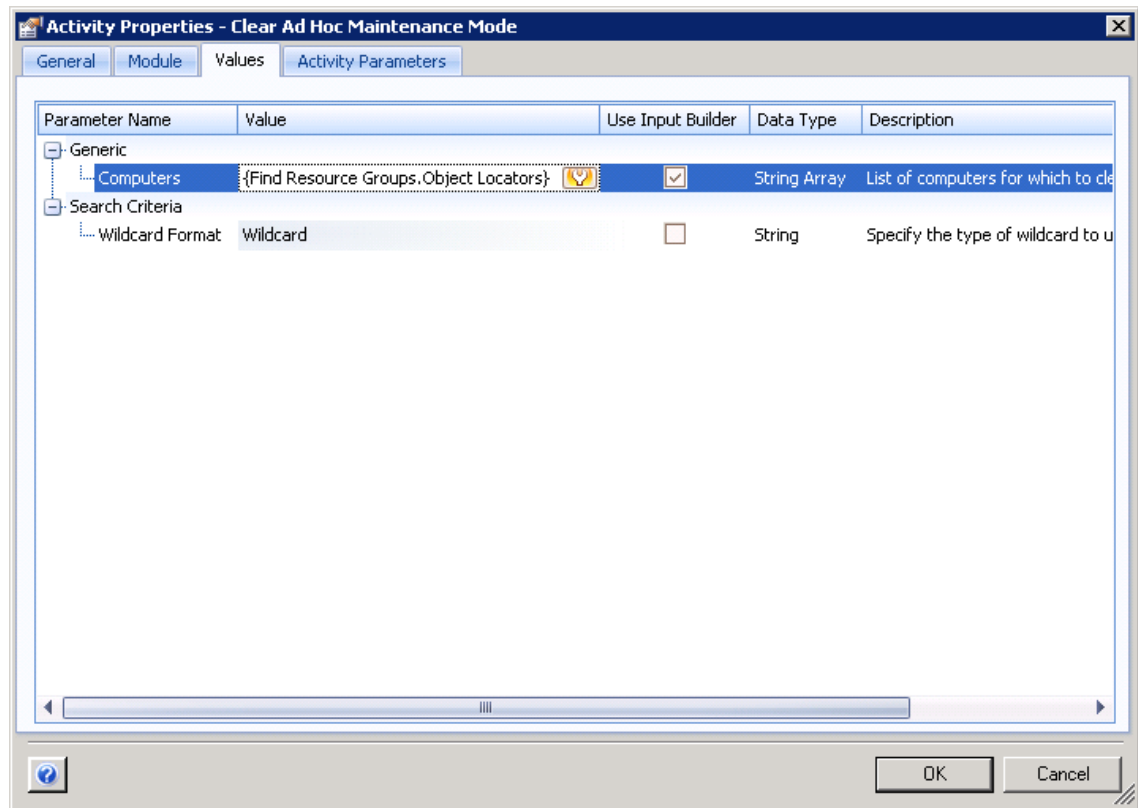
- In the Wait for Scheduled Event activity, create a filter with the following rules:
 - Events where Event Schedule Name is equal to Server Maintenance
 - Events where Scheduled Event Type is equal to End of Time Range
 - Events where Time Range Name is equal to {Trigger Custom Attribute.Time Range of Triggering Event}

NOTE: The brackets represent a portion of the filter for which you must use the Input Builder to specify the custom attribute you created previously. For more information about using the Input Builder, see [Understanding the Input Builder](#).

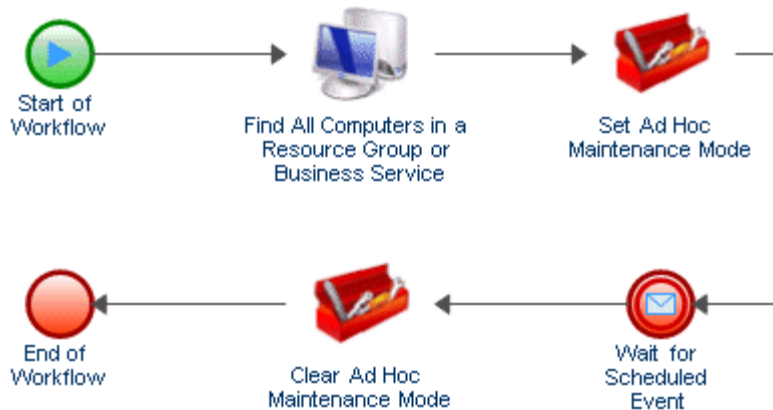


The activity pauses the workflow until the event schedule sends an event to indicate the end of the maintenance period.

- In the Clear Ad Hoc Maintenance Mode activity, for the **Computers** parameter, use the Input Builder to retrieve the **Object Locators** output parameter from the Find Resource Groups activity.



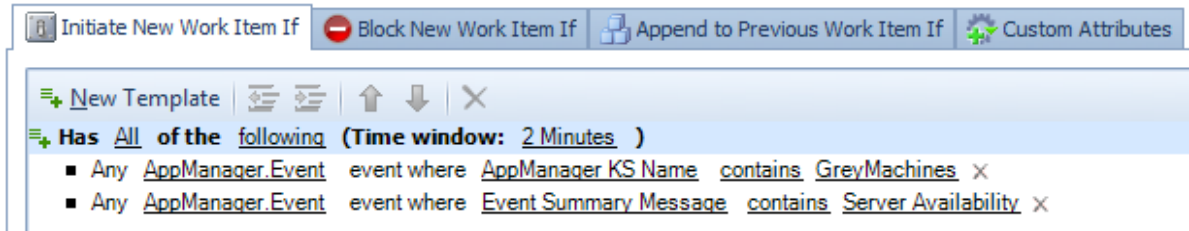
- Ensure the workflow has an End of Workflow activity, and then connect the activities with unconditional connectors.



For more information about connectors, see [Understanding Workflow Connectors](#).

3 Understanding Triggers

A trigger evaluates incoming events and determines whether to initiate a work item or append one or more events to existing work items. When you create or revise a trigger, you specify the trigger rules on the following tabs:



Initiate New Work Item If

If events match the rules on this tab, the trigger initiates a work item and attaches the triggering event to the work item as a related event. If multiple events match the rules, the trigger attaches all of the events to the work item. For more information about initiating a work item, see [Understanding Initiation Expressions](#).

Block New Work Item If

If events match the rules on this tab, the trigger blocks the creation of a new work item. For more information about blocking a work item, see [Understanding Blocking Expressions](#).

Append to Previous Work Item If

If events match the rules on this tab, the trigger appends the events to an existing work item. For more information about appending events to a work item, see [Understanding Appending Expressions](#).

By default, a trigger does not require a single event to meet all rules on any of the tabs. The trigger requires only that *one or more* events meet all specified rules *within the specified time window*. For example, the rules in the previous image initiate a new work item if either of the following occurs:

- ♦ One incoming `AppManager.Event` event has a `GreyMachines` KS and an event summary message that contains `Server Availability`:
- ♦ Two separate incoming events occur within two minutes of each other:
 - ♦ One `AppManager.Event` event with a `GreyMachines` KS
 - ♦ One `AppManager.Event` event with an event summary message that contains `Server Availability`:

If you want to ensure a trigger requires one event to meet all rules, you can create a named event and assign it to the rule. For more information about named events, see [Understanding Named Events](#).

3.1 Understanding Trigger Operators

An operator is a complex rule that ties other rules together and governs how they are related. Workflow Automation provides the following default operators.

Has All of the following

Requires one or more events to match all rules. For example, the following combination of rules on the **Initiate New Work Item If** tab tells the trigger to initiate a work item only if both descendant expressions are TRUE:

```
Has All of the following
  Expression 1
  Expression 2
```

Has Any of the following

Requires one event to match any of the rules. For example, the following combination of rules on the **Block New Work Item If** tab tells the trigger to block a new work item if either of the descendant expressions is TRUE:

```
Has Any of the following
  Expression 1
  Expression 2
```

Operators can have other operators as descendants, creating unlimited “nested” levels. For example, the following combination of rules on the **Append to Previous Work Item If** tab tells the trigger to append matching events to a work item only if Expression 1, Expression 2, and either Expression 3 or Expression 4 is TRUE.

```
Has All of the following
  Expression 1
  Expression 2
  Has Any of the following
    Expression 3
    Expression 4
```

3.2 Understanding Trigger Expressions

An expression is a simple rule that cannot contain other rules. When you configure an expression, you specify the criteria the trigger uses to match an incoming event, such as:

- ♦ The type of event the trigger must match, either a base event or a named event. For more information about event types, see [Understanding Events](#).
- ♦ The event attribute the trigger must evaluate on the incoming events.
- ♦ How the value of the event attribute compares to a specified value, such as `is equal to` or `is less than`.
- ♦ The value to which the trigger compares the attribute value on the incoming event.

3.2.1 Understanding Initiation Expressions

The **Initiate New Work Item If** tab provides the following expressions:

Any event of a specific type and attribute value

This expression matches a single base event with a specific value for one of the attributes. The time window does not apply. For example, to initiate a work item when Workflow Automation receives any AppManager event for which the KS name is `NT_ServiceDown`, configure the expression as follows:

```
Has All of the following (Time window: 2 Minutes )
  ■ Any AppManager.Event event where AppManager KS Name is equal to NT_ServiceDown x
```

Any event of a specific type which fails to match other triggers within the time window

This expression matches a specific event that does not match any other triggers during the specified time window.

NOTE: This trigger expression is not related to workflow priority. For more information about workflow priority, see [Understanding Workflow Priority](#).

If you use a base event, the trigger does not evaluate any attribute values on the incoming event. For example, if you specify `AppManager.Event`, the trigger initiates a work item when it matches any AppManager event that does match another trigger within the specified time window.

```
Has All of the following (Time window: 2 Minutes )
  ■ Any AppManager.Event event which fails to match other triggers within the time window x
```

If you use a named event, the trigger ensures the event matches the specified attribute values on the named event. For example, if you create a named event called `NT_ServiceDown`, where the event type is `AppManager.Event` and the value for the **AppManager KS Name** attribute is `NT_ServiceDown`, the trigger initiates a work item only if an event with the specified values does not match any other trigger within the specified time window.

```
Has All of the following (Time window: 2 Minutes )
  ■ Any NT_ServiceDown event which fails to match other triggers within the time window x
```

To ensure the event does not match other triggers during the specified time window, Workflow Automation requires approximately one minute of additional processing time. For example, if the time window is 2 minutes, Workflow Automation initiates a work item approximately three minutes after the event arrives.

Any event which fails to match other triggers within the time window

This expression matches any non-specified event that does not match any other triggers during the specified time window. The trigger ignores the event type and all attribute values. For example, you can use this expression to ensure Workflow Automation responds to all events.

NOTE: This trigger expression is not related to workflow priority. For more information about workflow priority, see [Understanding Workflow Priority](#).

To ensure the event does not match other triggers during the specified time window, Workflow Automation requires approximately one minute of additional processing time. For example, if the time window is 2 minutes, Workflow Automation initiates a work item approximately three minutes after the event arrives.

Any N or more events from the same business service

This expression matches a minimum number of non-specified events from the same business service within the specified time window. N must be greater than 1. The trigger ignores the event type and all attribute values. For example, to initiate a work item from three or more events that arrive from the same business service within two minutes, regardless of the event type or the relationship of the events, configure the template as follows:

```
Has All of the following (Time window: 2 Minutes )
  ■ Any 3 or more events from the same business service X
```

Any N or more events from the same cluster

This expression matches a minimum number of non-specified events from the same cluster within the specified time window. N must be greater than 1. The trigger ignores the event type and all attribute values. For example, to initiate a work item from two or more events that arrive from the same cluster within two minutes, regardless of the event type or the relationship of the events, configure the template as follows:

```
Has All of the following (Time window: 2 Minutes )
  ■ Any 2 or more events from the same cluster X
```

Any N or more events from the same computer

This expression matches a minimum number of non-specified events from the same computer within the specified time window. N must be greater than 1. The trigger ignores the event type and all attribute values. For example, to initiate a work item from two or more events from the same server within two minutes, regardless of the event type or the relationship of the events, configure the template as follows:

```
Has All of the following (Time window: 2 Minutes )
  ■ Any 2 or more events from the same computer X
```

Any N or more events of a specific type

This expression matches a minimum number of events of the same type within the specified time window. N must be greater than or equal to 1.

NOTE: If you set N to 1, the trigger ignores the specified time window and the expression behaves the same as `One event of a specific type`.

If you use a base event, the trigger does not evaluate any attribute values on the incoming event. For example, if you set N to 2 and specify `AppManager.Event`, the trigger initiates a work item when it matches at least two `AppManager` events within the specified time window.

```
Has All of the following (Time window: 2 Minutes )
  ■ Any 2 or more AppManager.Event events X
```

If you use a named event, the trigger ensures the event matches the specified attribute values on the named event. For example, if you set *N* to 2 and create a named event called `NT_ServiceDown`, where the event type is `AppManager.Event` and the value for the **AppManager KS Name** attribute is `NT_ServiceDown`, the trigger initiates a work item when at least two events with the specified values arrive within the specified time window.

```
Has All of the following (Time window: 2 Minutes )
  Any 2 or more NT_ServiceDown events x
```

Any N or more events of a specific type with identically valued attributes

This expression matches a minimum number of events within the specified time window, if the values of the specified attributes are identical on all events. *N* must be greater than 1.

If you use a base event, the trigger compares only the specified attributes. For example, to initiate a work item if two or more `Exchange.New Message Event` events occurring within two minutes of each other have the same subject, configure the expression as follows:

```
Has All of the following (Time window: 2 Minutes )
  Any 2 or more Exchange.New Message Event events with identically valued attributes Subject x
```

If you use a named event, the trigger evaluates the attributes associated with the named event as well as the attributes you specify for the expression. For example, if you create a named event called `Exchange Server Down`, where the event type is `Exchange.New Message Event` and the value for the **Subject** attribute is `Exchange Server Down`, you can specify additional attributes in the trigger expression, such as `Sender Email Address` and `Status`. If you set *N* to 2, the trigger initiates a work item only if two or more events with identical values for all specified attributes arrive within the specified time window.

```
Has All of the following (Time window: 2 Minutes )
  Any 2 or more Exchange Server Down events with identically valued attributes Sender Email Address, Status x
```

Combination of events with different attributes having common values

This expression matches any combination of at least two events within the specified time window, if the specified attributes have identical values.

NOTE: This expression has its own time window. By default, the expression inherits the value of the time window from the parent operator.

If you use base events, the trigger compares only the specified attributes. For example, the following configuration compares the **AppManager Machine Name** attribute on an `AppManager` event to the **Virtual Machine Name** attribute on a `vCenter` event. The trigger ignores all other attributes and values.

```
Has All of the following (Time window: 2 Minutes )
  Any 2 or more events with different attributes having common values defined as follows (Time window: inherited ) x
    Match key = AppManager.Event . AppManager Machine Name x
    Match key = VMware.vCenter Event . Virtual machine name x
```

If you use named events, the trigger evaluates the attributes associated with the named events as well as the attributes you specify for the expression. For example, the following configuration initiates a work item only if two events matching the attributes on the named events and having identical values for the specified attributes in the expression occur within two minutes of each other.

```

Has All of the following (Time window: 2 Minutes )
├─ Any 2 or more events with different attributes having common values defined as follows (Time window: inherited ) ×
│   └─ Match key = Poor Web Service Quality . AppManager Machine Name ×
│   └─ Match key = VM Detected Down . Virtual machine name ×

```

Combination of events with same attributes having common values

This expression matches any combination of at least two events within the specified time window, if each event has an identical value for the specified attribute.

NOTE: This expression has its own time window. By default, the expression inherits the value of the time window from the parent operator.

If you use base events, the trigger compares only the specified attributes. For example, the following configuration compares the **Resource Group Name** attribute on an AppManager event, an Aegis event, and a vCenter event. The trigger ignores all other attributes and values.

```

Has All of the following (Time window: 2 Minutes )
├─ Combination of following events with same attributes Resource Group Name having common values (Time window: inherited ) ×
│   └─ AppManager.Event event ×
│   └─ Aegis.Event event ×
│   └─ VMware.vCenter.Event event ×

```

If you use named events, the trigger evaluates the attributes associated with the named events as well as the attributes you specify for the expression. For example, the following configuration initiates a work item only if events matching the attributes on at least two of the named events and having identical values for the **Resource Group Name** attribute occur within two minutes of each other.

```

Has All of the following (Time window: 2 Minutes )
├─ Combination of following events with same attributes Resource Group Name having common values (Time window: inherited ) ×
│   └─ NT.ServiceDown event ×
│   └─ Poor Web Service Quality event ×
│   └─ VM Detected Down event ×

```

Compare multiple attribute values of two different events

This expression matches a combination of two specific events within the specified time window, if at least two specified attributes (one from each event) have identical values. To make the comparisons more restrictive, you can add more attributes to the expression.

This expression provides the following operators for the list of attributes you want to compare:

- ◆ **All** requires the two events to have identical values for all of the attributes.
- ◆ **Any** requires the two events to have identical values for any or all of the attributes.
- ◆ **None** requires the two events to not have identical values for all of the attributes.
- ◆ **Not Any** requires the events to not have identical values for any or all of the attributes.

If you use base events, the trigger compares only the specified attributes. For example, the following configuration compares the **Event Message** attribute on a vCenter event to the **Resource Name** attribute on an AppManager event. The trigger ignores all other attributes and values.

```
Has All of the following (Time window: 2 Minutes )
  Two events (Event 1: VMware.vCenter.Event and Event 2: AppManager.Event ) have All of the following x
    Event 1. Event Message is equal to Event 2. Resource Name x
```

If you use named events, the trigger evaluates the attributes associated with the named events as well as the attributes you specify for the expression. For example, the following configuration initiates a work item only if two events matching the attributes on the named events and having identical values for the specified attributes in the expression occur within two minutes of each other.

```
Has All of the following (Time window: 2 Minutes )
  Two events (Event 1: VM Detected Down and Event 2: Poor Web Service Quality ) have All of the following x
    Event 1. Event Message is equal to Event 2. Resource Name x
```

Compare two attribute values of two different events

This expression matches a combination of two specific events within the specified time window, if two specific attributes (one from each event) have identical values. If you want to compare more than two attributes, use the `Compare multiple attribute values of two different events` expression.

If you use base events, the trigger compares only the specified attributes. For example, the following configuration compares the **Event Message** attribute on a vCenter event to the **Computer Name** attribute on an AppManager event. The trigger ignores all other attributes and values.

```
Has All of the following (Time window: 2 Minutes )
  VMware.vCenter.Event . Event Message is equal to AppManager.Event . Resource Name x
```

If you use named events, the trigger evaluates the attributes associated with the named events as well as the attributes you specify for the expression. For example, the following configuration initiates a work item only if two events matching the attributes on the named events and having identical values for the specified attributes in the expression occur within two minutes of each other.

```
Has All of the following (Time window: 2 Minutes )
  VM Detected Down . Event Message is equal to Poor Web Service Quality . Resource Name x
```

One event of a specific type

This expression matches one specific event. The time window does not apply.

If you use a base event, the trigger does not evaluate any attribute values on the incoming event. For example, to initiate a work item when Workflow Automation receives any vCenter event, configure the expression as follows:

```
Has All of the following (Time window: 2 minutes )
  One VMware.vCenter.Event event x
```

If you use a named event, the trigger ensures the event matches the specified attribute values on the named event. For example, to initiate a work item when Workflow Automation receives a named event called `VM Detected Down`, where the event type is `VMware.vCenter Event` and the value for the `Status` attribute is `Closed`, configure the expression as follows:

```

Has All of the following (Time window: 2 Minutes )
  ■ Scheduled event defined by event schedule AE: Maintenance Mode Check x
  
```

Scheduled event

This expression matches any scheduled event from a specific event schedule. The time window does not apply. For example, the following configuration initiates a work item when the `AE: Maintenance Mode Check` event schedule generates an event.

```

Has All of the following (Time window: 2 Minutes )
  ■ Scheduled event defined by event schedule AE: Maintenance Mode Check x
  
```

Sequence of events

This expression matches a specific sequence of events within the specified time window. The expression compares the values for the same time-related sequence attribute on each of the events to determine the order of occurrence.

NOTE: This expression has its own time window. By default, the expression inherits the value of the time window from the parent operator.

If you use base events, the trigger evaluates only the specified sequence attribute. For example, the following configuration initiates a work item if an `AppManager` event, a `vCenter` event, and an `Aegis` event occur in the specified order within three minutes of each other according to the `Creation Time` sequence attribute. The trigger ignores all other attributes and values.

```

Has All of the following (Time window: 3 minutes )
  Sequence of following events (Time window: inherited ) x
    o AppManager.Event event (Sequence attribute: Creation Time ) x
    o VMware.vCenter Event event (Sequence attribute: Creation Time ) x
    o Aegis.Event event (Sequence attribute: Creation Time ) x
  
```

If you use named events, the trigger evaluates the attributes associated with the named events as well as the specified sequence attribute. For example, the following configuration initiates a work item only if three events matching the attributes on the named events occur in the specified order within three minutes of each other according to the `Creation Time` sequence attribute.

```

Has All of the following (Time window: 3 minutes )
  Sequence of following events (Time window: inherited ) x
    o Poor Web Service Quality event (Sequence attribute: Creation Time ) x
    o VM Detected Down event (Sequence attribute: Creation Time ) x
    o Server Rebooted event (Sequence attribute: Creation Time ) x
  
```

Sequence of events with same attributes having common values

This expression matches a specific sequence of events within the specified time window, if each event has an identical value for the specified attribute. The expression compares the values for the same time-related sequence attribute on each of the events to determine the order of occurrence.

NOTE: This expression has its own time window. By default, the expression inherits the value of the time window from the parent operator.

If you use base events, the trigger evaluates only the specified common and sequence attributes. For example, the following configuration initiates a work item if an AppManager event, a vCenter event, and an Aegis event have the same **Resource Group** attribute value, and occur in the specified order within three minutes of each other according to the **Creation Time** sequence attribute. The trigger ignores all other attributes and values.

```
Has All of the following (Time window: 3 minutes )
├─ Sequence of following events with same attributes Resource Group having common values (Time window: inherited ) ×
│  └─ AppManager.Event event (Sequence attribute: Creation Time ) ×
│     └─ VMware.vCenter Event event (Sequence attribute: Creation Time ) ×
│        └─ Aegis.Event event (Sequence attribute: Creation Time ) ×
```

If you use named events, the trigger evaluates the attributes associated with the named events as well as the specified sequence attribute. For example, the following configuration initiates a work item only if three events matching the attributes on the named events have the same **Resource Group** attribute value, and occur in the specified order within three minutes of each other according to the **Creation Time** sequence attribute.

```
Has All of the following (Time window: 3 minutes )
├─ Sequence of following events with same attributes Resource Group having common values (Time window: inherited ) ×
│  └─ Poor Web Service Quality event (Sequence attribute: Creation Time ) ×
│     └─ VM Detected Down event (Sequence attribute: Creation Time ) ×
│        └─ Server Rebooted event (Sequence attribute: Creation Time ) ×
```

3.2.2 Understanding Blocking Expressions

Workflow Automation can prevent a large number of unnecessary work items by blocking the following types of events:

- | | |
|---------------------------|--|
| Repetitive events | A data source may repeatedly send identical events during the course of an outage but the first notice is usually sufficient. |
| Symptomatic events | One failure may have downstream impacts, each of which generates a symptomatic event. Fix the root cause and many of these events go away. |
| False warnings | Performance management systems often have static and inexact thresholds that lead to a large number of warnings when there is no real problem. |

A trigger can block the creation of new work items by ignoring events that match the rules on the **Block New Work Item If** tab. By default, the trigger blocks matching events for the life of the work item, but you can configure the trigger to block matching events for a specific period of time after the work item starts. For example, if you expect a work item to run for an extended period of time after the trigger initiates it, you can configure the trigger to block subsequent matching events for the first 10 minutes the work item runs. After 10 minutes, the trigger stops blocking matching events.

NOTE: Process Authors can also define work item-level blocks when designing a workflow. A work item block terminates a running work item if it meets pre-defined conditions. For example, a conditional connector can take the workflow to an End of Workflow activity.

The **Block New Work Item If** tab provides the following expressions:

Any event from event source if this event initiated new work item

This expression matches any event from the same computer, business service, resource group, or business service role as the event that previously initiated the work item. The trigger does not evaluate any other attributes on the event. For example, if the trigger previously initiated a work item when Workflow Automation received an event in the `Exchange Servers` resource group, you can block additional matching events from initiating new work items during the first 10 minutes of the life of the existing work item by configuring the expression as follows:

```
All of the following occur within 2 Minutes of each other during the first 10 minutes of the work item
  ■ Any event from Resource Group if this event initiated new work item X
```

Any event matching attributes of the event that initiated new work item

This expression has the following sub-expressions:

- ◆ **Event attribute matching attribute of event(s) that initiated new work item** - Compares the value of an attribute on the current event to the value of an attribute on the event that initiated the work item, similar to the ON clause of a SQL JOIN statement. To evaluate multiple attributes on the current event, add multiple instances of this sub-expression. If multiple events initiated the work item, the attribute value on the current event can match the attributes on one or more of the triggering events. This is the primary sub-expression. The other sub-expressions are qualifiers.
- ◆ **Attribute of event(s) that initiated new work item matching event value** - Compares an attribute on the triggering event to a specific value, similar to the WHERE clause of a SQL JOIN statement. To evaluate multiple attributes on the triggering event, add multiple instances of this sub-expression. If multiple events initiated the work item, the trigger evaluates all of them. For best results, use this sub-expression as a qualifier for the `Event attribute matching attribute of event(s) that initiated new work item` sub-expression. Do not use it by itself.
- ◆ **Event attribute matching event value** - Compares the value of an attribute on the current event to a specific value, similar to the WHERE clause of a SQL JOIN statement. To evaluate multiple attributes on the current event, add multiple instances of this sub-expression. For best results, use this sub-expression as a qualifier for the `Event attribute matching attribute of event(s) that initiated new work item` sub-expression. Do not use it by itself.

For example, if the trigger previously initiated a work item when Workflow Automation received an AppManager event, you can block a new AppManager event from initiating a new work item during the first 10 minutes of the life of the existing work item if:

- ◆ The new event is from the same computer as the original event
- ◆ The triggering event belongs to a specific resource group, such as `Resource Group A`
- ◆ The new event has a specific Knowledge Script associated with it, such as `NT_NetworkBusy`

Configure the sub-expressions as follows:

```
All of the following occur within 2 Minutes of each other during the first 10 minutes of the work item
  ■ Any AppManager.Event event with All of the following X
    ○ Resource Name is equal to AppManager.Event . Resource Name of event(s) that initiated new work item X
    ○ AppManager.Event . Resource Group Name of event(s) that initiated new work item is equal to Resource Group A X
    ○ AppManager KS Name is equal to NT_NetworkBusy X
```

This configuration functions like a SQL JOIN statement with the following syntax:


```

SELECT *
FROM NewEvent
INNER JOIN TriggeringEvent
ON NewEvent.ComputerName=TriggeringEvent.ComputerName
WHERE TriggeringEvent.ResourceGroup = ResourceGroupA AND NewEvent.KSName =
NT_NetworkBusy

```

Where *NewEvent* is the new AppManager event and *TriggeringEvent* is the AppManager event that initiated the work item.

Any event of a specific type and attribute value

This expression matches a single event with a specific value for one of the attributes. For example, if the trigger previously initiated a work item when Workflow Automation received an AppManager event for which the KS name is *NT_SystemUptime*, you can block events where the KS name is *NT_ServiceDown* from initiating new work items during the first 10 minutes of the life of the existing work item by configuring the expression as follows:

```

All of the following occur within 2 Minutes of each other during the first 10 minutes of the work item
  ■ Any AppManager.Event event where AppManager KS Name is equal to NT_ServiceDown x

```

To match multiple attributes on the new event, add multiple instances of this expression. For example, if you want to block new AppManager events where the KS name is *NT_ServiceDown* only if the event is from a specific computer, such as *Computer01*, configure the expression as follows:

```

All of the following occur within 2 Minutes of each other during the first 10 minutes of the work item
  ■ Any AppManager.Event event where AppManager KS Name is equal to NT_ServiceDown x
  ■ Any AppManager.Event event where Resource Name is equal to Computer01 x

```

Any event of a specific type from event source if this event initiated new work item

This expression matches a single event from a specific computer, business service, resource group, or business service role.

If you use a base event, the trigger does not evaluate any other attributes on the event. For example, if the trigger previously initiated a work item when Workflow Automation received an event in the *Exchange Servers* resource group, you can block additional matching events from the same resource group from initiating new work items by configuring the expression as follows:

```

All of the following occur within 2 Minutes of each other during the life of the work item
  ■ Any Exchange.New Message Event event from Resource Group if this event initiated new work item x

```

If you use a named event, the trigger ensures the event matches the specified attribute values on the named event. For example, if the trigger previously initiated a work item when Workflow Automation received an *Exchange Server Down* event for the *Exchange Servers* resource group, you can block additional matching events from the same resource group from initiating new work items by configuring the expression as follows:

```

All of the following occur within 2 Minutes of each other during the life of the work item
  ■ Any Exchange Server Down event from Resource Group if this event initiated new work item x

```

One event of a specific type

This expression matches one specific event. The time window does not apply.

If you use a base event, the trigger does not evaluate any attribute values on the incoming event. For example, if the trigger previously initiated a work item when Workflow Automation received a vCenter event, you can block additional matching events from initiating new work items by configuring the expression as follows:

```
All of the following occur within 2 Minutes of each other during the life of the work item
  ■ One VMware.vCenter Event event x
```

If you use a named event, the trigger ensures the event matches the specified attribute values on the named event. For example, if the trigger previously initiated a work item when Workflow Automation received a VM Detected Down named event, you can block additional matching events from initiating new work items by configuring the expression as follows:

```
All of the following occur within 2 Minutes of each other during the life of the work item
  ■ One VM Detected Down event x
```

3.2.3 Understanding Appending Expressions

A trigger can append events that match the rules on the **Append to Previous Work Item If** tab to the work item. By default, the trigger appends matching events for the life of the work item, but you can configure the trigger to append matching events for a specific period of time after the work item starts. For example, if you expect a work item to run for an extended period of time after the trigger initiates it, you can configure the trigger to append subsequent matching events for the first 10 minutes the work item runs. After 10 minutes, the trigger stops appending matching events.

NOTE: A single event can trigger or be appended to multiple work items.

The **Append to Previous Work Item If** tab provides the following expressions:

Any event from event source if this event initiated new work item

This expression matches any event from the same computer, business service, resource group, or business service role as the event that previously initiated the work item. The trigger does not evaluate any other attributes on the event. For example, if the trigger previously initiated a work item when Workflow Automation received an event in the Exchange Servers resource group, you can append additional matching events to the existing work item during its first 10 minutes by configuring the expression as follows:

```
All of the following occur within 2 Minutes of each other during the first 10 minutes of the work item
  ■ Any event from Resource Group if this event initiated new work item x
```

Any event matching attributes of the event that initiated new work item

This expression has the following sub-expressions:

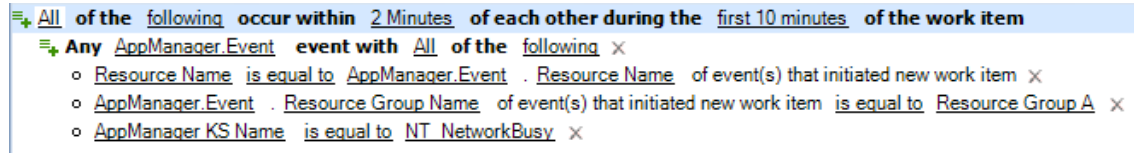
- ◆ **Event attribute matching attribute of event(s) that initiated new work item** - Compares the value of an attribute on the current event to the value of an attribute on the event that initiated the work item, similar to the ON clause of a SQL JOIN statement. To evaluate multiple attributes on the current event, add multiple instances of this sub-expression. If multiple events initiated the work item, the attribute value on the current event can match the attributes on one or more of the triggering events. This is the primary sub-expression. The other sub-expressions are qualifiers.

- ◆ **Attribute of event(s) that initiated new work item matching event value** - Compares an attribute on the triggering event to a specific value, similar to the WHERE clause of a SQL JOIN statement. To evaluate multiple attributes on the triggering event, add multiple instances of this sub-expression. If multiple events initiated the work item, the trigger evaluates all of them. For best results, use this sub-expression as a qualifier for the `Event attribute matching attribute of event(s) that initiated new work item` sub-expression. Do not use it by itself.
- ◆ **Event attribute matching event value** - Compares the value of an attribute on the current event to a specific value, similar to the WHERE clause of a SQL JOIN statement. To evaluate multiple attributes on the current event, add multiple instances of this sub-expression. For best results, use this sub-expression as a qualifier for the `Event attribute matching attribute of event(s) that initiated new work item` sub-expression. Do not use it by itself.

For example, if the trigger previously initiated a work item when Workflow Automation received an AppManager event, you can append a new AppManager to the existing work item during its first 10 minutes if:

- ◆ The new event is from the same computer as the original event
- ◆ The triggering event belongs to a specific resource group, such as `Resource Group A`
- ◆ The new event has a specific Knowledge Script associated with it, such as `NT_NetworkBusy`

Configure the sub-expressions as follows:



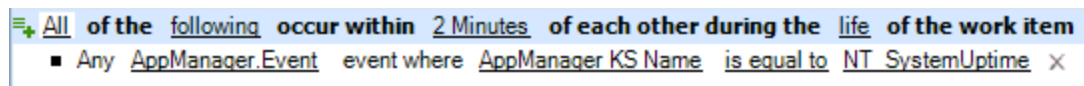
This configuration functions like a SQL JOIN statement with the following syntax:

```
SELECT *
FROM NewEvent
INNER JOIN TriggeringEvent
ON NewEvent.ComputerName=TriggeringEvent.ComputerName
WHERE TriggeringEvent.ResourceGroup = ResourceGroupA AND NewEvent.KSName =
NT_NetworkBusy
```

Where `NewEvent` is the new AppManager event and `TriggeringEvent` is the AppManager event that initiated the work item.

Any event of a specific type and attribute value

This expression matches a single event with a specific value for one of the attributes. For example, if the trigger previously initiated a work item when Workflow Automation received an AppManager event for which the KS name is `NT_SystemUptime`, you can append additional matching events where the KS name is `NT_SystemUptime` to the existing work item by configuring the expression as follows:



Any event of a specific type from event source if this event initiated new work item

This expression matches a single event from a specific computer, business service, resource group, or business service role.

If you use a base event, the trigger does not evaluate any other attributes on the event. For example, if the trigger previously initiated a work item when Workflow Automation received an event in the `Exchange Servers` resource group, you can append additional matching events from the same resource group to the existing work item by configuring the expression as follows:

```
All of the following occur within 2 Minutes of each other during the life of the work item
  ■ Any Exchange.New Message Event event from Resource Group if this event initiated new work item x
```

If you use a named event, the trigger ensures the event matches the specified attribute values on the named event. For example, if the trigger previously initiated a work item when Workflow Automation received an `Exchange Server Down` event for the `Exchange Servers` resource group, you can append additional matching events from the same resource group to the existing work item by configuring the expression as follows:

```
All of the following occur within 2 Minutes of each other during the life of the work item
  ■ Any Exchange Server Down event from Resource Group if this event initiated new work item x
```

One event of a specific type

This expression matches one specific event. The time window does not apply.

If you use a base event, the trigger does not evaluate any attribute values on the incoming event. For example, if the trigger previously initiated a work item when Workflow Automation received a vCenter event, you can append additional matching events to the existing work item by configuring the expression as follows:

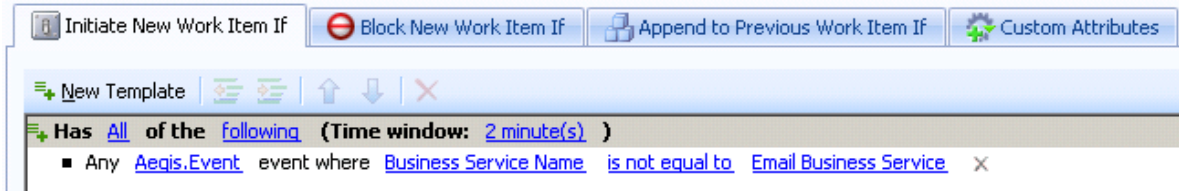
```
All of the following occur within 2 Minutes of each other during the life of the work item
  ■ One VMware.vCenter Event event x
```

If you use a named event, the trigger ensures the event matches the specified attribute values on the named event. For example, if the trigger previously initiated a work item when Workflow Automation received a `VM Detected Down` named event, you can append additional matching events to the existing work item by configuring the expression as follows:

```
All of the following occur within 2 Minutes of each other during the life of the work item
  ■ One VM Detected Down event x
```

3.3 Using the “Is Not Equal To” Trigger Comparison

When configuring a trigger, you can use the `is not equal to` trigger comparison to tell a trigger to act only if the new event’s attribute values are not equal to the specified values. For example, the following trigger initiates a work item only if the value of the **Business Service Name** attribute on the new event is not equal to `Email Business Service`:

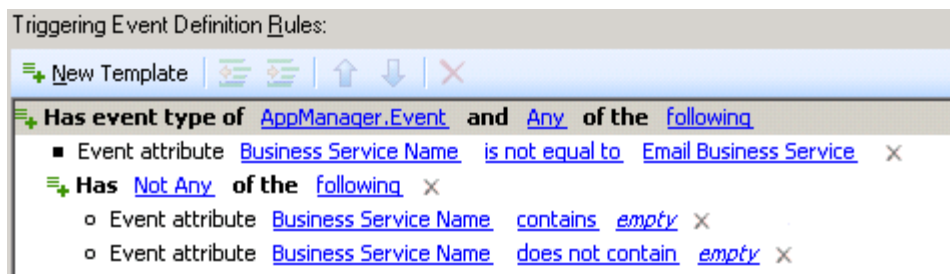


When you use the `is not equal to` trigger comparison, the trigger ignores empty values in the following event attributes:

- ◆ Business Service ID
- ◆ Business Service Name
- ◆ Business Service Role
- ◆ Computer ID
- ◆ Computer Name
- ◆ Knowledge
- ◆ Resource Group ID
- ◆ Resource Group Name

For example, if the **Business Service Name** attribute in the previous graphic has an empty value, the trigger ignores the event and does not initiate a work item, even though the empty value is not equal to the specified comparison value.

To configure a trigger to initiate work items for events with empty values that do not equal the specified comparison values, create a named event and associate it with the trigger. The following example specifies a named event that matches AppManager events with **Business Service Name** values not equal to `Email Business Service`, including empty values:



Where *empty* represents any text string you do not use when naming your business services.

For more information about named events, see [Understanding Named Events](#).

3.4 Understanding Trigger Custom Attributes

When you configure a trigger, you can use the **Custom Attributes** tab to create custom attributes and values the trigger passes to each work item it initiates. Process Authors can use the Input Builder to reference the trigger custom attributes in conditional connectors and activities.

NOTE

- ◆ Process Authors can also define custom attributes at the work item level. For more information about defining custom work item attributes in a workflow, see [Customizing Work Item Attributes](#).
- ◆ If a custom work item attribute has the same name as a trigger custom attribute, the value in the custom work item attribute overrides the value in the trigger custom attribute.

If multiple triggers pass custom attributes with the same name, the Input Builder displays one **set**, which represents all trigger custom attributes with the same name.

For example, you can associate the following triggers with the Start of Workflow activity:

Associated Triggers (in evaluation order):

Name	Type	Revision	Description
Trigger 1	User Defined	1.0	
Trigger 2	User Defined	1.0	

If Trigger 1 initiates a work item from the workflow, it passes the following trigger custom attributes:

Trigger Name:
Trigger 1

Description:

Initiate New Work Item If Block New Work Item If Append to Previous Work Item If Custom Attributes

Generate the following custom attributes in work items created by this trigger:

Name	Display Name	Type	Value	Description
A	A	String		
B	B	String		
C	C	String		

If Trigger 2 initiates a work item from the workflow, it passes the following trigger custom attributes:

Trigger Name:
Trigger 2

Description:

Initiate New Work Item If Block New Work Item If Append to Previous Work Item If Custom Attributes

Generate the following custom attributes in work items created by this trigger:

Name	Display Name	Type	Value	Description
A	A	String		
B	B	String		
D	D	String		

When you use the Input Builder to reference a trigger custom attribute, you see the following list:

The screenshot shows the 'Input Builder' dialog box. At the top, there is an 'Input Editor' section with a toolbar containing a 'New Input Element' button, a list icon, a refresh icon, and up/down arrow icons. Below this is the 'Input Elements' section, which displays a list of available trigger custom attributes. The list has a header with columns for 'Custom Attribute Name', 'Data Type', and 'Description'. The list contains four entries: 'A' (String), 'B' (String), 'C' (String), and 'D' (String). The 'A' entry is currently selected. At the bottom of the dialog, there are 'OK' and 'Cancel' buttons.

Custom Attribute Name	Data Type	Description
A	String	
B	String	
C	String	
D	String	

In the list of available trigger custom attributes, A and B represent sets of multiple trigger custom attributes with the same name from different triggers. Combining multiple trigger custom attributes into a single set allows you to reduce the amount of logic you need to add to the workflow at design time. You can configure the workflow to reference only one trigger custom attribute, regardless of which trigger initiated the work item. For more information about using the Input Builder to reference trigger custom attributes, see [Understanding the Input Builder](#).

3.5 Revising Existing Triggers

If you need to update an existing trigger, you can create a new trigger revision. The new trigger revision is available for new processes or workflow revisions, but does not replace the original trigger in any workflow revisions in production. To revise a trigger for a workflow revision in production, clone the workflow revision, and then revise the trigger. For more information about cloning a workflow revision, see [Cloning a Workflow Revision](#).

3.6 Understanding Abstract Trigger Revisions

Abstract trigger revisions allow Workflow Automation to upgrade workflow revisions using the trigger to the latest trigger revision. Workflow Automation provides the following abstract trigger revisions in the Start of Workflow activity:

- ♦ **N.*** represents the latest trigger revision with a specific major revision number (N) and highest minor revision number. For example, when you create Workflow Revision A and assign the abstract revision 1.* of Trigger A, the current trigger revision is 1.1. When Workflow Automation upgrades the workflow revision, Workflow Automation upgrades Trigger A to revision 1.5, which is the most recent 1.* revision, rather than to the latest revision of Trigger A, which is 2.2.
- ♦ ***.*** represents the latest trigger revision with the highest major and minor revision numbers. For example, when you create Workflow Revision B and assign the abstract revision *.* of Trigger A, the current trigger revision is 1.1. When Workflow Automation upgrades the workflow revision, Workflow Automation upgrades to the latest revision of Trigger A, which is 2.2.

Workflow Automation upgrades to the latest trigger revisions in the following scenarios:

Putting a Workflow Revision into Production

When you put a workflow revision with abstract trigger revisions into production for the first time, Workflow Automation assigns the latest trigger revision according to the rule for the specified abstract revision. For more information about putting a workflow into production, see [Putting a Workflow Revision into Production](#).

Creating a New Trigger Revision

When you create a new trigger revision, Workflow Automation verifies whether any workflow revisions reference one of the trigger's abstract revisions, and displays the Deploy Trigger Updates window. Workflow Automation ignores workflow revisions that are currently Under Construction.

Manually Deploying Trigger Updates

When you manually deploy trigger updates for a specific trigger, Workflow Automation verifies whether any workflow revisions reference one of the trigger's abstract revisions, and displays the Deploy Trigger Updates window. Workflow Automation ignores workflow revisions that are currently Under Construction.

In each scenario, Workflow Automation restricts the list of workflow revisions to the ones you have the permissions to view, clone, and put into production.

When you deploy trigger updates, Workflow Automation clones the specified workflow revision, and then puts the clone into production, which upgrades all triggers to use their latest trigger revisions. For more information about cloning a workflow revision, see [Cloning a Workflow Revision](#).

If you upgrade a workflow revision that is currently out of production, the clone remains out of production.

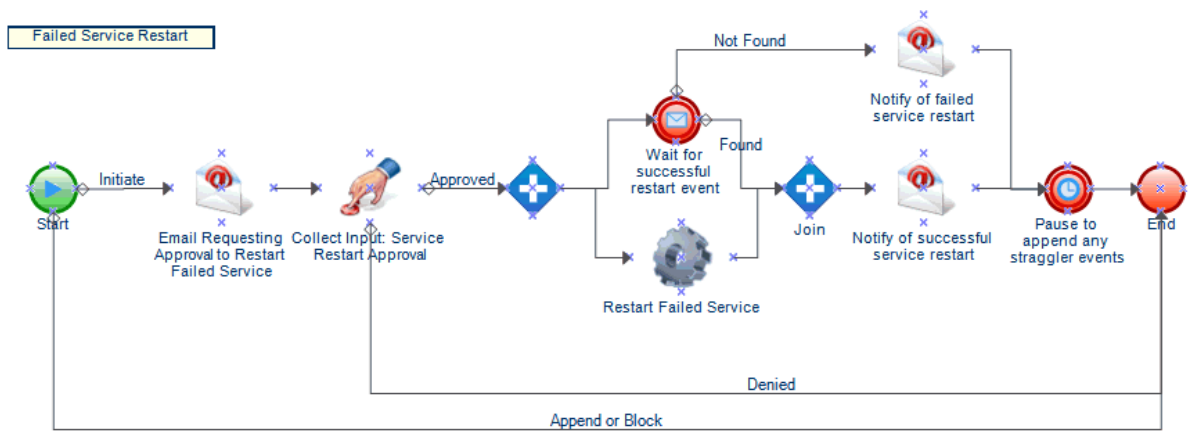
3.7 Understanding the Manual Trigger

Workflow Automation provides a manual trigger you can associate with any process. A manual trigger requires human intervention to initiate a work item and allows users to initiate a work item from the process at any time. A process can have only one manual trigger.

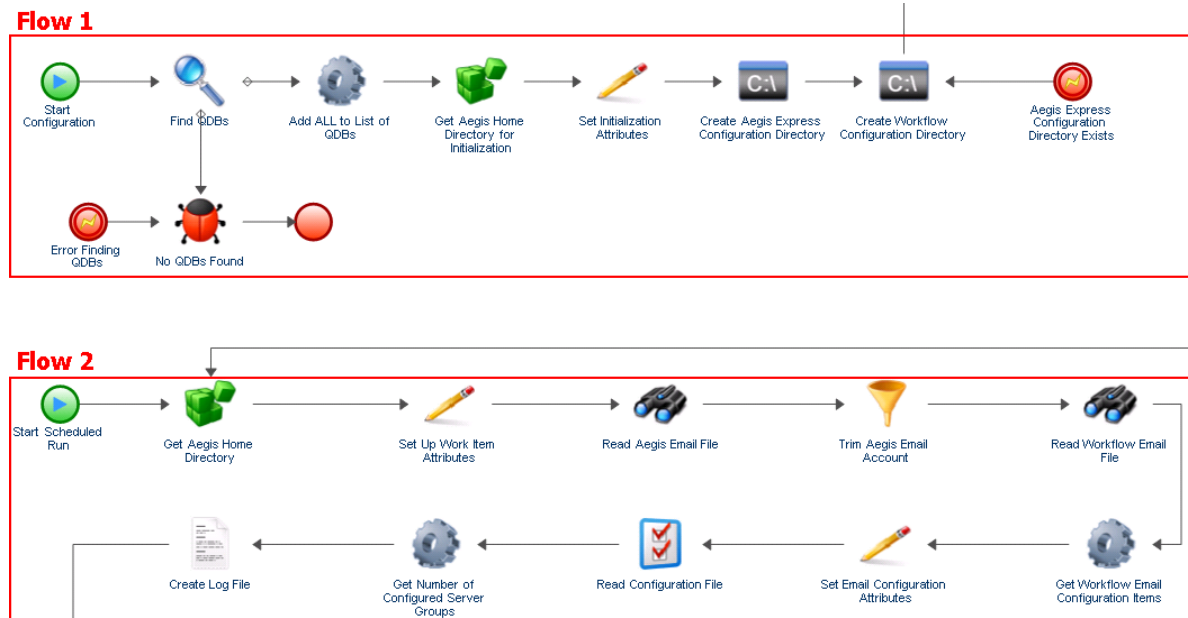
4 Working with Processes and Workflows

A “process” consists of the steps taken to respond to an event sent to Workflow Automation by a data source. To establish your IT policy governance model, you must define the IT processes you use when you respond to specific events throughout your environment. Workflow Automation evaluates incoming events against the processes your team defines to represent your IT policies.

Each process has an associated workflow. A “workflow” is a graphical representation of the steps in a process, made up of activities and connectors, designed to reflect a specific IT policy, as shown in the following figure. When one or more events from a data source match the trigger criteria of a process, Workflow Automation executes the process and initiates a work item.



Use the Workflow Designer to refine the process workflow until it matches your IT policy for addressing, responding to, and closing an incident. During execution, a workflow visits activities and traverses connectors in a specific order (called a **flow**). The Start of Workflow activity is the first activity the workflow executes. If the workflow contains multiple Start of Workflow activities, the workflow can have multiple simultaneous flows, as shown in the following figure.



The workflow can also create multiple flows when it traverses more than one connector from an activity or when the trigger appends a new event to the work item. When there are no more active flows, execution terminates.

NOTE: Multiple flows with the same termination point can terminate at different times.

4.1 Understanding Process Terminology

To understand how Workflow Automation uses the processes you define, you should understand the following terms:

Work Item

A single instance of a process, triggered by either an incoming event or a manual trigger. A process stands idle until a trigger initiates a work item. A single process may have multiple work items running simultaneously.

Activity








A step in a process that performs a specific function, such as starting the workflow, joining multiple flows, waiting for an incoming event, or stopping the workflow. For more information about activities, see [Understanding Workflow Activities](#).

Trigger

A set of rules associated with a process that determine how to respond to incoming events, such as initiating a new work item or appending an event to an existing work item. A manual trigger requires human intervention to initiate a work item. For more information about triggers, see [Understanding Triggers](#).

4.2 Understanding Workflow Activities

The Workflow Designer provides the following basic activity types you can use to build workflows.

	<p>Start of Workflow</p>	<p>Starts the workflow when its associated triggers match incoming events from a data source.</p> <p>A workflow can have multiple Start of Workflow activities that create more than one flow at the same time.</p> <p>When you associate a trigger with the activity, you can choose an abstract trigger revision. For more information about abstract trigger revisions, see Understanding Abstract Trigger Revisions.</p>
	<p>Task</p>	<p>Performs a specific task, such as collecting user input or sending an email, before proceeding. An associated piece of code, or module, performs the task.</p> <p>Workflow Automation provides several activity libraries that group Task activities according to function. Each Workflow Automation adapter provides its own library of product-specific activities.</p> <p>Task activities in an Workflow Automation or adapter library have an icon that represents their functions, such as the Send Email activity.</p>
	<p>Join</p>	<p>Allows the process to split into multiple parallel flows or combine two or more parallel flows into one.</p> <p>Join activities require the workflow to meet the conditions on all incoming conditional connectors before it proceeds to the next activity. For more information about satisfying multiple conditional connectors with a Join activity, see Understanding How Join Activities Interact with Conditional Connectors.</p>
	<p>Wait for New Email Message</p>	<p>Waits for an incoming message that matches the defined filter. If a matching message does not arrive in the specified time frame, the process continues.</p>
	<p>Pause for Specified Time</p>	<p>Waits for a specified period of time or until a specified time before continuing.</p>
	<p>Capture Workflow Errors</p>	<p>Waits for errors to occur in the workflow.</p> <p>This activity sits outside the main flow. When an error occurs, the flow “jumps” to this activity and proceeds down an alternate flow that handles the error before rejoining the main flow.</p>
	<p>End of Workflow</p>	<p>Marks the end of the process.</p> <p>A workflow can have multiple End activities, depending on the number of sub-flows.</p> <p>If the workflow has multiple flows but only one End of Workflow activity, a Completed icon displays when the first flow terminates, even if other flows are still active.</p>

4.3 Understanding the Input Builder

The Input Builder allows you to construct dynamic values for activity input parameters. A dynamic value is a variable for which the Input Builder obtains a value at run time. You can also construct complex literal values. The Input Builder is also available for some fields in the Configuration Console. For detailed examples of using the Input Builder to construct input values, see the sample workflows in [Workflow Examples](#).

The Input Builder allows you to construct a value with one or more of the following input elements:

Activity Output Parameter

Retrieves an output value from another activity in the workflow. You can retrieve any output parameter from any activity in the workflow, such as the name of the trigger that activated the Start of Workflow activity. This element is available only in the Workflow Designer.

Activity Execution Parameter

Retrieves information related to a specific activity's start time, end time, or duration during the course of a workflow. This element is available only in the Workflow Designer.

Event Attribute

Retrieves the specified attribute value from the event that triggered the work item, such as the subject line on an email event.

Work Item Attribute

Retrieves the specified attribute value from the current work item, such as the subject or description. You can also retrieve custom work item attributes.

Trigger Custom Attribute

Retrieves the specified attribute value from the trigger that initiated the work item. This element is available only in the Workflow Designer.

Global Setting

Retrieves the value from the specified global setting at run time, such as the name of the email server or the URL for the Operations Console.

Input Form Link

Retrieves the URL of the user input form associated with the specified Collect Input from User activity. For more information, see [Linking to a User Input Form](#).

Free-Form Text

Allows you to add text to the constructed value. For example, if you use the Send Email activity in a workflow, you can use the Input Builder to construct the message body with a combination of free-form text and activity output parameters. The text you type is the same for each work item. The variables retrieve values at run time.

New Line

Adds a new line to the constructed value. For example, if you use the Send Email activity in a workflow, you can use the Input Builder to construct the message body with a combination of literal values and variables. You can separate blocks of text with one or more new lines. This element is only available in the Workflow Designer.

Space

Adds a space to the constructed value. For example, if you want to specify the full name of a user in an input parameter, and you must retrieve the first and last names as separate values, you can use the Input Builder to concatenate the two values into one value at run time, separated by a space.

Empty String

Adds an empty string to the constructed value. For example, if the input parameter requires an entry at design time, but you want to specify a null value, you can use the Input Builder to populate the input parameter with an empty string.

Tab

Adds a tab to the constructed value. For example, if you want to construct the contents of a text file, you can insert a tab to indent the start of a paragraph.

4.4 Understanding User Input Forms

User input forms allow Workflow Automation users, typically Process Operators, to interact with a work item and provide feedback the associated workflow needs to complete its task. For example, if you are building a workflow to restart a service, and the restart requires approval, you can use the Collect Input from User activity to pause the workflow and wait for approval. Depending on the user's response, the workflow can either restart the service or stop.

4.4.1 Linking to a User Input Form

If you do not want the user to interact with the Operations Console, you can send an email with a link to a specific user input form. When the user clicks the link in the email and logs in, the input form displays in a browser window. The user does not see the full workflow or any other data in the Operations Console.

Workflow Automation supports user input forms in the following Web browsers:

- ♦ Mozilla Firefox
- ♦ Google Chrome
- ♦ Microsoft Internet Explorer 11, Edge

To create a link to a user input form:

- 1 In the Workflow Designer, drag a Send Email activity and a Collect Input from User activity to the design grid.
- 2 Use a connector to join the activities, from the Send Email activity to the Collect Input from User activity.
- 3 Double-click the Send Email activity, and then complete the following steps:
 - 3a On the Values tab, next to the Body parameter, select the **Use Input Builder** check box.
 - 3b In the Value field for the Body parameter, click **Input Builder**.
 - 3c On the Input Builder window, click **New Input Element**, and then click **Input Form Link**.
 - 3d Under Input Form Activity Name, select **Collect Input from User**, and then click **OK**.

NOTE: If you have configured multiple Collect Input from User activities, ensure you select the one connected to the Send Email activity you are configuring.

- 3e On the Input Builder window, click **OK**.
- 3f On the activity properties window, configure the remaining input parameters, and then click **OK**.
- 4 Configure the Collect Input from User activity and any other activities you have added to the workflow.

NOTE: For detailed examples of creating workflows, see [Workflow Examples](#)

4.4.2 Customizing a User Input Form

You can change the appearance of the user input forms in your environment by modifying the following cascading style sheets:

- ♦ `InputForm.css` - styles related to the input form and form fields
- ♦ `LoginDialog.css` - styles related to the login window
- ♦ `MainLayout.css` - styles related to the banner, footer, and main panel
- ♦ `Page.css` - styles related to the login view and the input form view inside the main panel

The `.css` files are typically located in the `C:\Program Files\NetIQ\Aegis\AegisInputForm\css` folder on the Web Server computer.

NOTE: If you modify the `.css` files in your environment, ensure you back them up before upgrading to the next version. After the upgrade is complete, you can restore them and replace the `.css` files installed by the upgrade.

4.5 Understanding Workflow Connectors

Workflows use connectors to travel from one activity to another. When you add a connector to a workflow, you can specify whether it is conditional or unconditional. An unconditional connector always allows the workflow to traverse it from one activity to another. A conditional connector uses a conditional expression to determine whether the workflow can traverse it.

4.5.1 Configuring a Connector

The basic steps for configuring a connector are the same for both conditional and unconditional connectors.

To configure a connector:

- 1 On the Workflow tab, in the Tools group, click **Connect Activities Tool**.
- 2 On the activity from which you want the connector to originate, click the appropriate port, and then drag the pointer to the appropriate port on the activity to which you want the workflow to travel.

TIP

- ◆ An activity's port turns into a red box when the tool hovers over it.
- ◆ The connector automatically picks the best "route," based on the ports you select. To switch to manual mode, press the space bar.
- ◆ You can draw all the connectors you need while the Connect Activities tool is active.

-
- 3 To switch back to the standard pointer tool, on the Workflow tab, in the Tools group, click **Selection Tool**.
 - 4 If the Connector Properties window does not display automatically, right-click the connector, and then click **Properties**.
 - 5 Verify that the **From** and **To** fields specify the correct activities.

TIP: You can use the **From** and **To** fields to connect the appropriate activities without manually positioning the ends of the connector.

- 6 Choose the desired action to:
 - ◆ reverse the flow of the connector, click **Reverse Connector**
 - ◆ configure an unconditional connector, click **Always traverse this connector**
 - ◆ configure a conditional connector, complete the following steps:
 1. Click **Only traverse this connector when the following conditions are met**.
 2. Under Conditional Expression, click **Create Expression** to launch the Expression Editor window.
 3. On the Expression Editor window, configure the conditional expression, and then click **OK**. For more information about conditional expressions, see [Understanding Conditional Expressions](#).
- 7 On the Connector Properties window, click **OK**.

4.5.2 Understanding Conditional Expressions

A conditional expression contains one or more of the following components:

Operator

An operator is a complex rule, such as `Has All of the following`, that ties rules together and governs how they are related using Boolean logic. Operators may have other operators as descendants, creating unlimited nested levels.

To add a new operator to a conditional expression, click **New Template**, and then select one of the following:

- ◆ **All** to require the workflow to match a combination of the following rules
- ◆ **Any** to require the workflow to match any or all of the following rules
- ◆ **None** to require the workflow to not match a combination of the following rules
- ◆ **Not Any** to require the workflow to not match any or all of the following rules

Parameter Expression

Evaluates the value of an output parameter from a previous activity in the workflow. For example, you can specify that the connector allows the workflow to traverse it only if the value for the **Status of the send** parameter on the Send Email activity equals `Succeeded`.

Activity Execution Attribute Expression

Evaluates the value of a specific activity's start time, end time, or duration. For example, you can specify that the connector allows the workflow to traverse it only if the Find Objects activity took less than 60 seconds to complete.

Event Attribute Expression

Evaluates the value of an attribute on the triggering event. Attributes from triggering events are attached to the work item. For example, you can specify that the connector allows the workflow to traverse it only if the event type equals `Aegis.Event` and the value for the **Business Service Name** attribute equals `Messaging Service`.

Work Item Attribute Expression

Evaluates the value of a work item attribute. For example, you can specify that the connector allows the workflow to traverse it only if the value for the **Work Item Resource Groups** attribute equals `Exchange Servers`.

Trigger Custom Attribute Expression

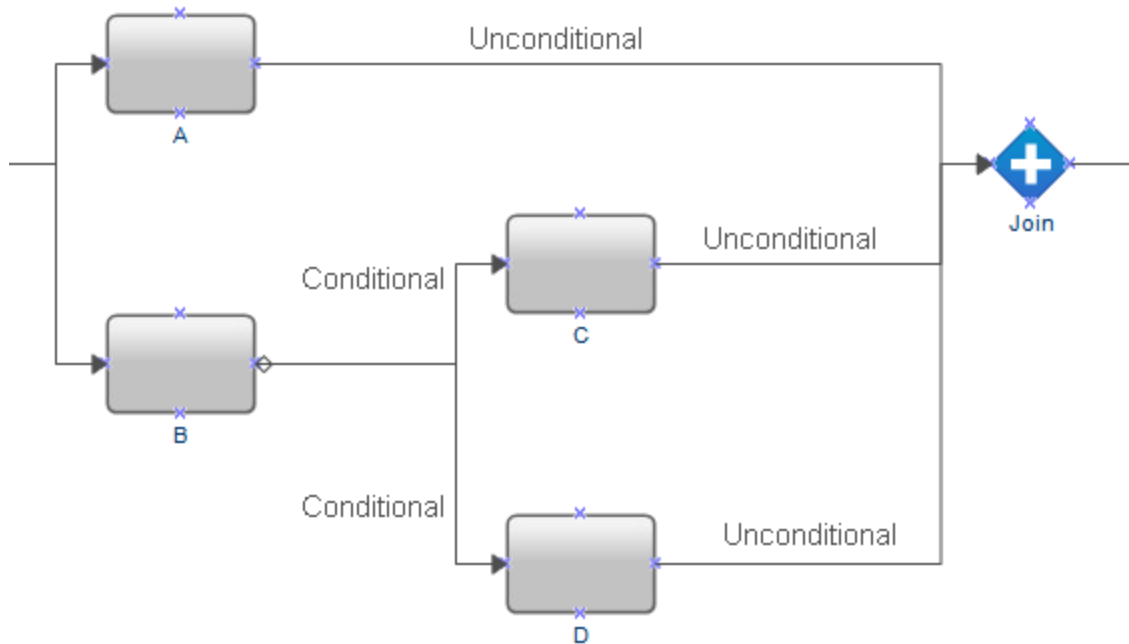
Evaluates the value of a custom attribute defined on a trigger. For example, you can specify that the connector allows the workflow to traverse it only if the value for the trigger custom attribute **Cause** equals `Exchange server down`.

Global Setting Expression

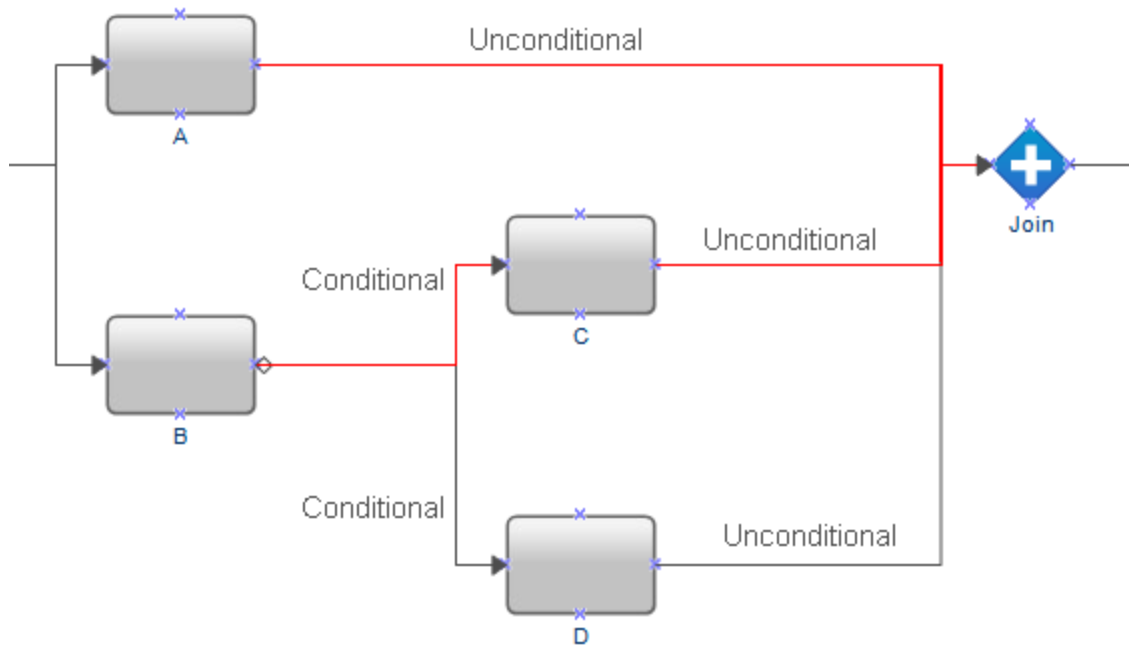
Evaluates the value of the specified global setting, such as the name of the email server or the URL for the Operations Console. For example, you can specify that the connector allows the workflow to traverse it only if name of the email server computer matches the value you provide.

4.5.3 Understanding How Join Activities Interact with Conditional Connectors

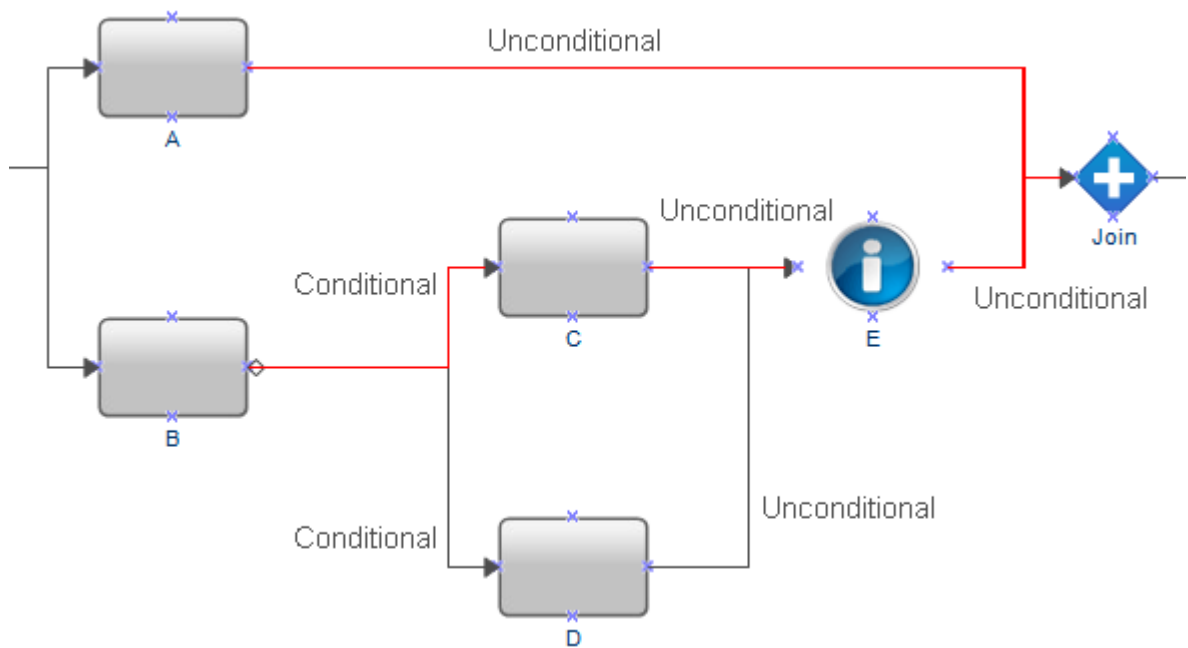
A Join activity requires the workflow to traverse all incoming connectors before it can proceed to the next activity in the workflow. If a conditional expression prevents the workflow from traversing all connectors, the Join activity cannot proceed. For example, if the workflow splits into two parallel flows, one starting with activity A, the other starting with activity B, the Join activity has three incoming connectors.



The workflow always traverses the unconditional connector between A and the Join activity, but the conditional connectors after B require the workflow to executing either C or D. If the workflow cannot traverse the connector between D and the Join activity, the Join activity cannot proceed.



To ensure the workflow is able to traverse all connectors coming in to the Join activity, add a No Operation activity (activity E) between activities C and D and the Join activity. As a result, the workflow is always able to traverse all connectors coming in to the Join activity, and the Join activity is able to proceed to the next activity in the workflow.



4.6 Working with Strings in Workflow Activities

When designing a workflow, if you manually type a string with single quotes into an input parameter, Workflow Automation strips the single quotes from the string. For example, if you type ' , ' in the Echo activity, the output is the comma without the single quotes.

To ensure Workflow Automation does not strip the single quotes from the string, do one of the following:

- ♦ Escape the single quote with another single quote. For example, ' ' , ' ' .
- ♦ Use the Free-Form Text option in the Input Builder to type the string.

4.7 Creating a Process

The following checklist provides an overview of the steps for creating a new process and its associated workflow.

	Checklist Items
<input type="checkbox"/>	1. In the Navigation pane, click Getting Started .
<input type="checkbox"/>	2. In the Getting Started pane, click Create Process to launch the Process Creation Wizard.
<input type="checkbox"/>	3. Complete the Process Creation Wizard.
<input type="checkbox"/>	4. If you want to open the Workflow Designer automatically upon finishing the wizard, on the Process Properties tab, select the Open Workflow Designer check box.
<input type="checkbox"/>	5. Click Finish to save the process and launch the Workflow Designer.
<input type="checkbox"/>	6. Configure the activities for the workflow. For step-by-step examples of configuring activities, see Workflow Examples .
<input type="checkbox"/>	7. Configure the connectors. For more information about configuring connectors, see Understanding Workflow Connectors .
<input type="checkbox"/>	8. Validate the workflow. For more information about validating a workflow, see Validating a Workflow .
<input type="checkbox"/>	9. When you are ready to use the process, put the workflow revision into your production environment. For more information, see Putting a Workflow Revision into Production .

4.8 Workflow Examples

These topics provide step-by-step guidance for creating sample workflows from start to finish.

4.8.1 Find the Management Server with the Fewest AppManager Agents

If you install the AppManager Integration Adapter (AppManager adapter), you can design a workflow to determine which management server (MS) in your AppManager environment to use when deploying the next AppManager agent. Each AppManager agent in the environment has a primary MS and an optional secondary MS. Ideally, each MS handles a similar workload with a similar number of agents.

To help balance the number of AppManager agents associated with each MS in your AppManager environment, you can create the following workflows:

- ♦ A **parent workflow** that uses the Add Computer to Deployment Rule activity to deploy a new AppManager agent to a server you want to monitor. Deployment rules in AppManager can differ from one server to the next, and the workflow determines which rules to use based on the MS you want to configure as the primary MS for the new agent. When the parent workflow needs the name of the MS to use as the primary MS for the new agent, it launches the child workflow. After the child workflow provides the host name of the MS, the parent workflow deploys the agent and assigns the specified MS as the agent's primary MS.
- ♦ A **child workflow** that determines which MS is currently the primary MS for the fewest number of agents. The child workflow sends the host name of the MS to the parent workflow.

This example walks you through creating the child workflow.

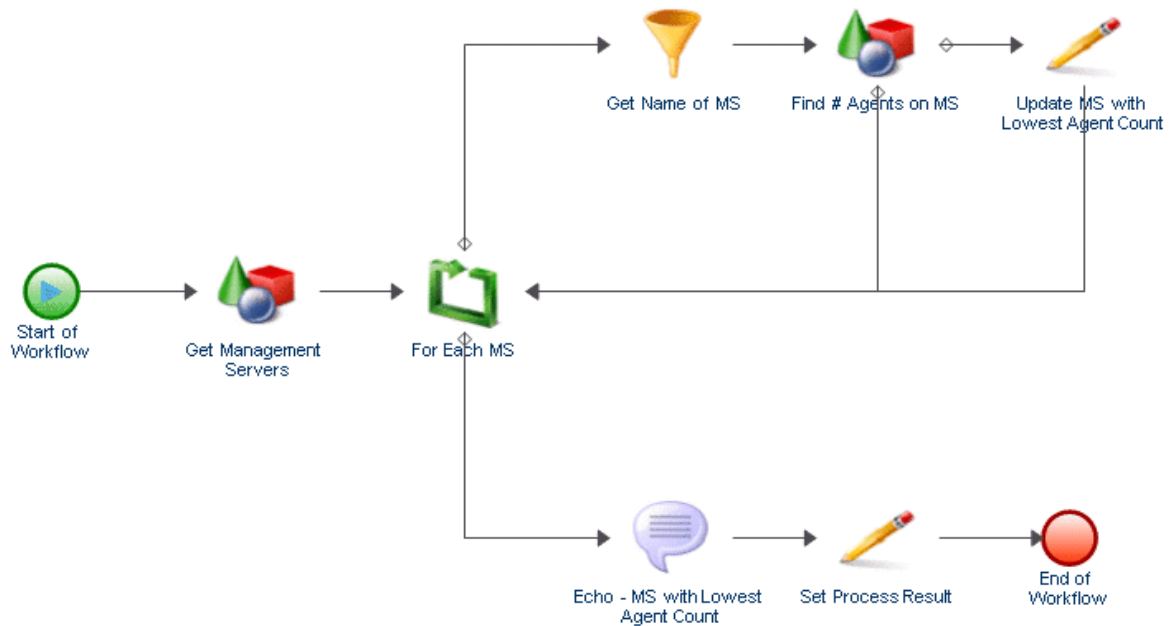
Assumptions

This example assumes the following:

- ♦ A parent workflow exists to launch this workflow
- ♦ Workflow Automation communicates with only one AppManager environment
- ♦ Deployment rule names are based on the MS they assign as the primary MS to the agent
- ♦ There are no UNIX computers in the AppManager environment

Building the Workflow

When you complete this sample procedure, your workflow should look like this:



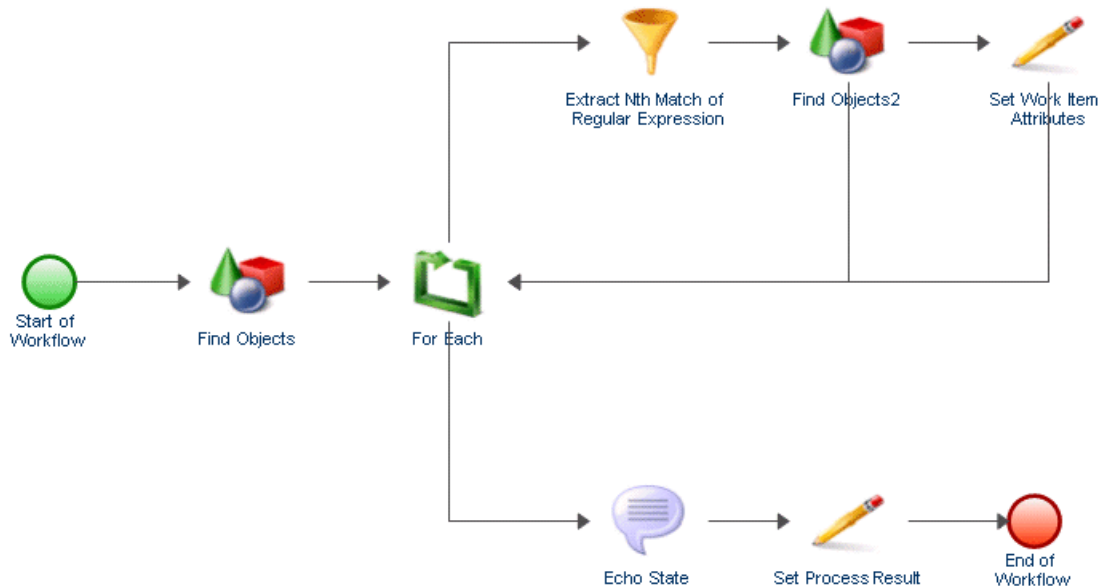
To create a workflow to determine where to deploy the next AppManager agent:

- 1 In the Library Browser, drag the following activities from the specified activity libraries to the design grid:

Activity	Library
Start of Workflow	Basic Workflow Control
Find Objects (2)	Basic Workflow Control
For Each	Basic Workflow Control
Extract Nth Match of Regular Expression	Data Manipulation
Set Work Item Attributes	Data Manipulation
Echo State	Basic Workflow Control
Set Process Result	Basic Workflow Control
End of Workflow	Basic Workflow Control

- 2 Use connectors to join the activities, in the following order:
 - ◆ Start of Workflow to Find Objects (1)
 - ◆ Find Objects (1) to For Each
 - ◆ For Each to Extract Nth Match of Regular Expression
 - ◆ Extract Nth Match of Regular Expression to Find Objects (2)
 - ◆ Find Objects (2) to Set Work Item Attributes
 - ◆ Find Objects (2) to For Each

- ◆ Set Work Item Attributes to For Each
- ◆ For Each to Echo State
- ◆ Echo State to Set Process Result
- ◆ Set Process Result to End of Workflow



- 3 On the View tab in the Properties group, click **Workflow Properties**.
- 4 On the **Work Item** tab, add two custom work item attributes with the following properties:

	Attribute 1	Attribute 2
Name	minimumAgentCount	MSwithLowestNumberOfAgents
Display Name	Minimum Agent Count	MS with Lowest Number of Agents
Type	Integer Number	String
Value	1000000	MSwithLowestNumberOfAgents
Description	Stores the value of the least number of agents on an MS (default is 1000000 so will always be reset on first loop)	
Read Only	Yes	Yes

- 5 On the Workflow Properties window, under **Custom Work Item Attributes**, use the **Move Up** or **Move Down** buttons to ensure **minimumAgentCount** is in the first evaluation position.
- 6 Click **OK**.
- 7 Double-click the Start of Workflow activity, and then complete the following steps:
 - 7a On the Trigger tab, click **Add**.
 - 7b Under Available Triggers, select **Manual Trigger**, and then click **Add**.
 - 7c Click **OK**.

- 8 Double-click the first Find Objects activity, and then complete the following steps:
 - 8a On the General tab, in the Name field, type `Get Management Servers`.
 - 8b On the Query tab, under Object Type, select **AppManager > AppManager Repository > Windows Computer**, and then click **OK**.
 - 8c Click the **Windows Computer** sub-tab.
 - 8d Click **Create Filter**.
 - 8e On the Expression Editor window, click **New Template**, and then click **Expression**.
 - 8f Click `<attribute>`, select **Management Server**, and then click **OK**.
 - 8g Click `<attribute value>`, and then click **Enter Simple Value**.
 - 8h Click **true**.
 This expression ensures the activity only retrieves management server computers in the AppManager adapter namespace.
 - 8i On the Expression Editor window, click **OK**.
 - 8j Select the **Generate Object Locators** check box.
 - 8k Click **OK**.
- 9 Double-click the For Each activity, and then complete the following steps:
 - 9a On the General tab, in the Name field, type `For Each MS`.
 - 9b Click the **Values** tab.
 - 9c For the Input Array parameter, select the **Use Input Builder** check box.
 - 9d In the Value field for the Input Array parameter, click the **Input Builder** button.
 - 9e On the Input Builder window, click **New Input Element**, and then click **Activity Output Parameter**.
 - 9f Under Output Parameter Name, expand **Get Management Servers**, select **Object Locators**, and then click **OK**.
 The Object Locators output parameter is a string array that contains a list of the management server computers in the AppManager adapter namespace.
 - 9g On the Input Builder window, click **OK**.
 - 9h In the Value field for the Start Index parameter, specify the position in the input string array from which you want the For Each activity to retrieve a value on the first pass through the loop. The default value is 0, which references the first element in the string array.
 On each pass through the loop, the activity adds the specified increment to the current index to calculate the Next Index output parameter. The activity uses **Next Index** on the next pass through the loop. On the first pass, the activity assigns the start index to the **Current Index** output parameter. For example, if **Start Index** is 0 and **Increment** is 1, the first time the activity executes, it retrieves the value from index position 0 in the string array, and then sets **Next Index** to 1. The second time the activity executes, the activity retrieves the value from index position 1 in the string array, and then sets **Next Index** to 2. The activity increments the index value on each execution until Next Index is equal to or greater than the last index position in the string array.
 - 9i On the activity properties window, click **OK**.
- 10 Double-click the connector between For Each MS and Extract Nth Match of Regular Expression, and then complete the following steps:
 - 10a Select **Only traverse this connector when the following conditions are met**.
 - 10b Under Conditional Expression, click **Create Expression**.

- 10c** On the Expression Editor window, click **New Template**, and then click **Parameter Expression**.
- 10d** Click **<parameter name>**, expand **For Each MS**, select **Complete**, and then click **OK**.
- 10e** Click **<attribute value>**, and then click **Enter Simple Value**.
- 10f** Click **false**.
- The expression allows the workflow to take this path as long as the For Each MS activity is processing the list of management server computers. When the For Each MS activity processes the last management server computer in the string array, it sets the **Complete** output parameter to `true`, and the expression prevents the workflow from taking this path.
- 10g** On the Expression Editor window, click **OK**.
- 10h** Uncheck **Show Label**.
- 10i** Click **OK**.
- 11** Double-click the connector between For Each MS and Echo State, and then complete the following steps:
- 11a** Select **Only traverse this connector when the following conditions are met**.
- 11b** Under Conditional Expression, click **Create Expression**.
- 11c** On the Expression Editor window, click **New Template**, and then click **Parameter Expression**.
- 11d** Click **<parameter name>**, expand **For Each MS**, select **Complete**, and then click **OK**.
- 11e** Click **<attribute value>**, and then click **Enter Simple Value**.
- 11f** Click **true**.
- The expression allows the workflow to take this path only after the For Each MS activity processes the last management server computer in the string array and sets the Complete Output parameter to `true`.
- 11g** On the Expression Editor window, click **OK**.
- 11h** Uncheck **Show Label**.
- 11i** Click **OK**.
- 12** Double-click the Extract Nth Match of Regular Expression activity, and then complete the following steps:
- 12a** On the General tab, in the Name field, type `Get Name of MS`.
- 12b** Click the **Values** tab.
- 12c** For the Input String parameter, select the **Use Input Builder** check box.
- 12d** In the Value field for the Input String parameter, click the **Input Builder** button.
- 12e** On the Input Builder window, click **New Input Element**, and then click **Activity Output Parameter**.
- 12f** Under Output Parameter Name, expand **For Each MS**, select **Output Value**, and then click **OK**.
- The Output Value output parameter contains a locator for a management server computer.
- 12g** On the Input Builder window, click **OK**.
- 12h** In the Regular Expression parameter, type the following:
`/IQAM_WindowsComputer=(\w+)`
- 12i** In the **Segments** parameter, type `1(1)`.
- 12j** Select the **Output as String** check box.

- 12k In the Output String Separator parameter, type a comma (,).
- 12l Click **OK**.
- 13 Double-click the second Find Objects activity, and then complete the following steps:
 - 13a On the General tab, in the Name field, type `Find # Agents on MS`.
 - 13b On the Query tab, under Object Type, select **AppManager > AppManager Repository > Windows Computer**, and then click **OK**.
 - 13c Click the **Windows Computer** sub-tab.
 - 13d Click **Create Filter**.
 - 13e On the Expression Editor window, click **New Template**, and then click **Expression**.
 - 13f Click **<attribute>**, select **Primary Management Server**, and then click **OK**.
 - 13g Click **is equal to**, and then click **contains**.
 - 13h Click **<attribute value>**, and then click **Use Input Builder**.
 - 13i On the Input Builder window, click **New Input Element**, and then click **Activity Output Parameter**.
 - 13j Expand **Get Name of MS**, click **Result String**, and then click **OK**.
 - 13k On the Input Builder window, click **OK**.
 - 13l On the Expression Editor window, click **OK**.
- 13m Ensure **Generate Object Locators** is checked.
- 13n Click **OK**.
- 14 Double-click the connector between Find # Agents on MS and For Each MS, and then complete the following steps:
 - 14a Select **Only traverse this connector when the following conditions are met**.
 - 14b Under Conditional Expression, click **Create Expression**.
 - 14c On the Expression Editor window, click **New Template**, and then click **Parameter Expression**.
 - 14d Click **<parameter name>**, expand **Find # Agents on MS**, select **Number of Object Locators**, and then click **OK**.
 - 14e Click **is equal to**, and then click **is greater than or equal to**.
 - 14f Click **<attribute value>**, and then click **Use Input Builder**.
 - 14g On the Input Builder window, click **New Input Element**, and then click **Work Item Attribute**.
 - 14h Under Custom Work Item Attributes, click **Minimum Agent Count**, and then click **OK**.
 - 14i On the Input Builder window, click **OK**.
 - 14j On the Expression Editor window, click **OK**.
 - 14k Uncheck **Show Label**.
 - 14l Click **OK**.
- 15 Double-click the connector between Find # Agents on MS and Set Work Item Attributes, and then complete the following steps:
 - 15a Select **Only traverse this connector when the following conditions are met**.
 - 15b Under Conditional Expression, click **Create Expression**.
 - 15c On the Expression Editor window, click **New Template**, and then click **Parameter Expression**.

- 15d Click **<parameter name>**, expand **Find # Agents on MS**, select **Number of Object Locators**, and then click **OK**.
- 15e Click **is equal to**, and then click **is less than**.
- 15f Click **<attribute value>**, and then click **Use Input Builder**.
- 15g On the Input Builder window, click **New Input Element**, and then click **Work Item Attribute**.
- 15h Under Custom Work Item Attributes, click **Minimum Agent Count**, and then click **OK**.
- 15i On the Input Builder window, click **OK**.
- 15j On the Expression Editor window, click **OK**.
- 15k Uncheck **Show Label**.
- 15l Click **OK**.
- 16 Double-click the Set Work Item Attributes activity, and then complete the following steps:
 - 16a On the General tab, in the Name field, type `Update MS with Lowest Agent Count`.
 - 16b Click the **Attributes** tab.
 - 16c Under Available Custom Attributes, select each of the following attributes, and then click **Add**:
 - ◆ Minimum Agent Count
 - ◆ MS with Lowest Number of Agents
 - 16d Under Configured Custom Attributes, for the Minimum Agent Count attribute, select the **Use Input Builder** check box.
 - 16e In the Value field for the Minimum Agent Count parameter, click the **Input Builder** button.
 - 16f On the Input Builder window, click **New Input Element**, and then click **Activity Output Parameter**.
 - 16g Under Output Parameter Name, expand **Find # Agents on MS**, select **Number of Object Locators**, and then click **OK**.
 - 16h On the Input Builder window, click **OK**.
 - 16i Under Configured Custom Attributes, for the MS with Lowest Number of Agents attribute, select the **Use Input Builder** check box.
 - 16j In the **Value** field for the MS with Lowest Number of Agents parameter, click the **Input Builder** button.
 - 16k On the Input Builder window, click **New Input Element**, and then click **Activity Output Parameter**.
 - 16l Under Output Parameter Name, expand **Get Name of MS**, select **Result String**, and then click **OK**.
 - 16m On the Input Builder window, click **OK**.
 - 16n Click **OK**.
- 17 Double-click the Echo State activity, and then complete the following steps:
 - 17a On the General tab, in the Name field, type `Echo - MS with Lowest Agent Count`.
 - 17b Click the **Values** tab.
 - 17c For the Echo parameter, select the **Use Input Builder** check box.
 - 17d In the Value field for the **Echo** parameter, click the **Input Builder** button.
 - 17e On the Input Builder window, click **New Input Element**, and then click **Work Item Attribute**.
 - 17f Under Custom Work Item Attributes, click **MS with Lowest Number of Agents**, and then click **OK**.

- 17g On the Input Builder window, click **OK**.
- 17h Click **OK**.
- 18 Double-click the Set Process Result activity, and then complete the following steps:
 - 18a Click the **Values** tab.
 - 18b Set the **Workflow Result** parameter to `Success`.
 - 18c For the Workflow Output Value parameter, select the **Use Input Builder** check box.
 - 18d In the Value field for the Workflow Output Value parameter, click the **Input Builder** button.
 - 18e On the Input Builder window, click **New Input Element**, and then click **Activity Output Parameter**.
 - 18f Under Output Parameter Name, expand **Echo - MS with Lowest Agent Count**, select **Echo**, and then click **OK**.
 - 18g On the Input Builder window, click **OK**.
 - 18h Click **OK**.
- 19 In the Workflow Designer, on the Workflow tab, in the File group, click **Save Workflow**.
- 20 On the Workflow tab, in the File group, click **Close Workflow Designer**.

Adding Functionality to the Workflow

To increase the complexity and capability of this workflow you might add one or more of the following:

- ♦ **Multiple environments** - Workflow Automation can communicate with multiple AppManager environments. For example, your company might divide its management servers between three AppManager environments, each in a different geographic location.
- ♦ **Agents on UNIX computers** - If your AppManager environment deploys agents to both Windows and UNIX computers, you can add another Find Objects activity to retrieve all agents on UNIX computers, and then use the Simple Calculator activity to add the results to the output of the Find # Agents on MS activity.
- ♦ **Server types** - Your AppManager environment might divide management servers into similar groups based on their function, such as UNIX, Windows, Exchange Server, or SQL Server. The workflow first determines to which group it should deploy the new agent, and then determines the specific MS.

4.8.2 Creating a Snapshot on a Virtual Machine

If you install the Workflow Automation Adapter for VMware vCenter Server (vCenter adapter), you can design a workflow that creates a new snapshot for an existing virtual machine (VM). In this example, the workflow has a manual trigger. When a Process Operator manually triggers the workflow in the Operations Console, the workflow presents a user input form that asks for the following information:

- ♦ Name of the data center that hosts the VM
- ♦ Name of the VM for which to create the snapshot
- ♦ Name of the new snapshot
- ♦ Optional description of the new snapshot
- ♦ Whether to include the VM memory in the snapshot
- ♦ Whether to quiesce the file system while creating the snapshot

After the Process Operator provides the requested input, the workflow looks for the VM and creates the snapshot.

Assumptions

This example assumes your environment has only one vCenter Server and that all datacenters have unique names.

Building the Workflow

When you complete this sample procedure, your workflow should look like this:



To create a workflow to create a new snapshot on a virtual machine:

- 1 In the Library Browser, drag the following activities from the specified activity libraries to the design grid:

Activity	Library
Start of Workflow	Basic Workflow Control
Collect Input from User	Email and User Interaction
Find Virtual Machines	VMware vCenter
Create Snapshot	VMware vCenter
End of Workflow	Basic Workflow Control

- 2 Use connectors to join the activities, in the following order:
 - ♦ Start of Workflow to Collect Input from User
 - ♦ Collect Input from User to Find Virtual Machines
 - ♦ Find Virtual Machines to Create Snapshot
 - ♦ Create Snapshot to End of Workflow
- 3 Double-click the Start of Workflow activity, and then complete the following steps:
 - 3a On the Trigger tab, click **Add**.
 - 3b Under Available Triggers, select **Manual Trigger**, and then click **Add**.
 - 3c Click **OK**.
- 4 Double-click the Collect Input from User activity, and then complete the following steps:
 - 4a On the Input Form tab, click **Edit**.
 - 4b On the General tab of the Create/Edit User Input Form wizard, specify a title for the user input form, such as `Snapshot Information`.
 - 4c Type a description, and then click **Next**.

- 4d On the Form Input Fields tab of the Create/Edit User Input Form wizard, add the following field definitions in the specified order:

Type	Parameter ID	Parameter Name	Required
Single Line Text	DataCenterName	Datacenter Name	Yes
Single Line Text	VMName	VM Name	Yes
Single Line Text	SnapshotName	Name of New Snapshot	Yes
Single Line Text	Snapshot Description	Snapshot Description (optional)	No
Check Box	IncludeMemory	Include VM Memory in Snapshot	
Check Box	Quiesce	Quiesce File System	

- 4e Click **Finish**.
- 4f Click **OK**.
- 5 Double-click the Find Virtual Machines activity, and then complete the following steps:
- 5a On the Query tab, select **Generate Object Locators**.
- 5b Click the **Datacenter** sub-tab.
- 5c Click **Edit Filter**.
- 5d On the Expression Editor window, click **New Template**, and then click **Expression**.
- 5e Click **<attribute>**, select **Datacenter name**, and then click **OK**.
- 5f Click **<attribute value>**, and then click **Use Input Builder**.
- 5g On the Input Builder window, click **New Input Element**, and then click **Activity Output Parameter**.
- 5h Expand **Collect Input from User**, click **Datacenter Name**, and then click **OK**.
- 5i On the Input Builder window, click **OK**.
- 5j On the Expression Editor window, click **OK**.
- 5k Click the **Virtual Machine** sub-tab.
- 5l Click **Edit Filter**.
- 5m On the Expression Editor window, click **New Template**, and then click **Expression**.
- 5n Click **<attribute>**, select **Inventory name**, and then click **OK**.
- 5o Click **<attribute value>**, and then click **Use Input Builder**.
- 5p On the Input Builder window, click **New Input Element**, and then click **Activity Output Parameter**.
- 5q Expand **Collect Input from User**, click **VM Name**, and then click **OK**.
- 5r On the Input Builder window, click **OK**.
- 5s On the Expression Editor window, click **OK**.
- 5t Click **OK**.
- 6 Double-click the Create Snapshot activity.
- 7 On the **Values** tab, complete the following steps for each input parameter:
- 7a Select the **Use Input Builder** check box.
- 7b In the Value field, click the **Input Builder** button.

7c On the Input Builder window, click **New Input Element**, and then click **Activity Output Parameter**.

7d For each input parameter, expand the specified activity, and then click the specified output parameter.

Input Parameter	Activity Name	Output Parameter Name
Virtual Machines	Find Virtual Machines	Object Locators
Snapshot Name	Collect Input from User	Name of New Snapshot
Snapshot Description	Collect Input from User	Snapshot Description (optional)
Memory Snapshot	Collect Input from User	Include VM Memory in Snapshot
Quiesce File System	Collect Input from User	Quiesce File System

7e On the Input Builder window, click **OK**.

7f Click **OK**.

8 In the Workflow Designer, on the Workflow tab, in the File group, click **Save Workflow**.

9 On the Workflow tab, in the File group, click the **Close Workflow Designer**.

Adding Functionality to the Workflow

To increase the complexity and capability of this workflow, you might add one or more of the following:

- ♦ **Automatic trigger** - You can add a trigger that initiates a work item when a user sends an email with the subject `Request Snapshot` to a monitored email address. If the user correctly formats the message body, you can use an Extract Nth Match of Regular Expression activity to parse the contents and retrieve the following:
 - ♦ Name of the datacenter hosting the VM
 - ♦ Name of the VM for which to create the snapshot
 - ♦ Name of the new snapshot
 - ♦ Optional description of the new snapshot
 - ♦ Whether to include the VM memory in the snapshot
 - ♦ Whether to quiesce the file system
- ♦ **Error checking** - If the contents of the triggering email are not in the correct format, the Extract Nth Match of Regular Expression activity cannot parse the contents and generates an error. A Capture Workflow Errors activity can retrieve the error and proceed down an alternate flow to a Send Email activity. The Send Email activity notifies the person who requested the new snapshot that the email request was in the wrong format.
- ♦ **Multiple vCenter Servers** - If your environment has multiple vCenter servers, you can add a field definition to the user input form to request the name of the vCenter server. Datacenters in different vCenter servers can have the same name. To ensure the Find Virtual Machines activity finds the VM, you must add a filter that matches the specified vCenter server name to one in your environment.

4.8.3 Creating a New User Account

If you install the Workflow Automation Adapter for Directory and Resource Administrator (DRA adapter), you can design a workflow that creates a new user account in response to an email.

Assumptions

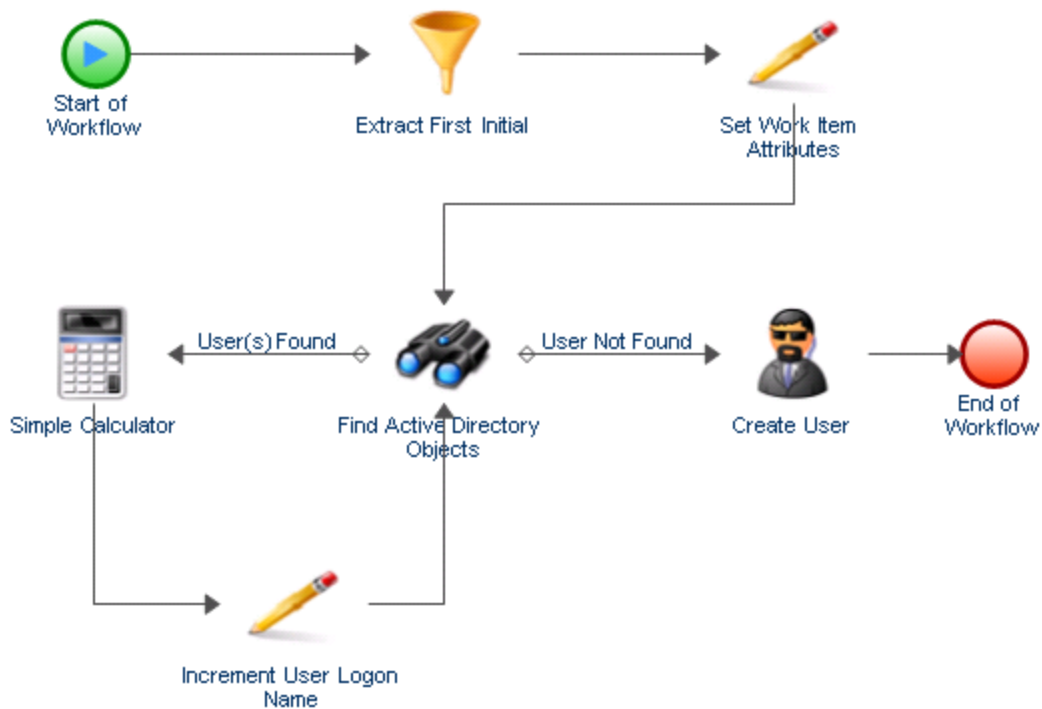
This example assumes the following:

- Your Workflow Automation environment monitors Microsoft Exchange Server mailboxes
- The email subject is *Create new user*
- The body of the email is formatted as follows:

```
FirstName LastName
Domain
Location
Department
```

Building the Workflow

When you complete this sample procedure, your workflow should look like this:



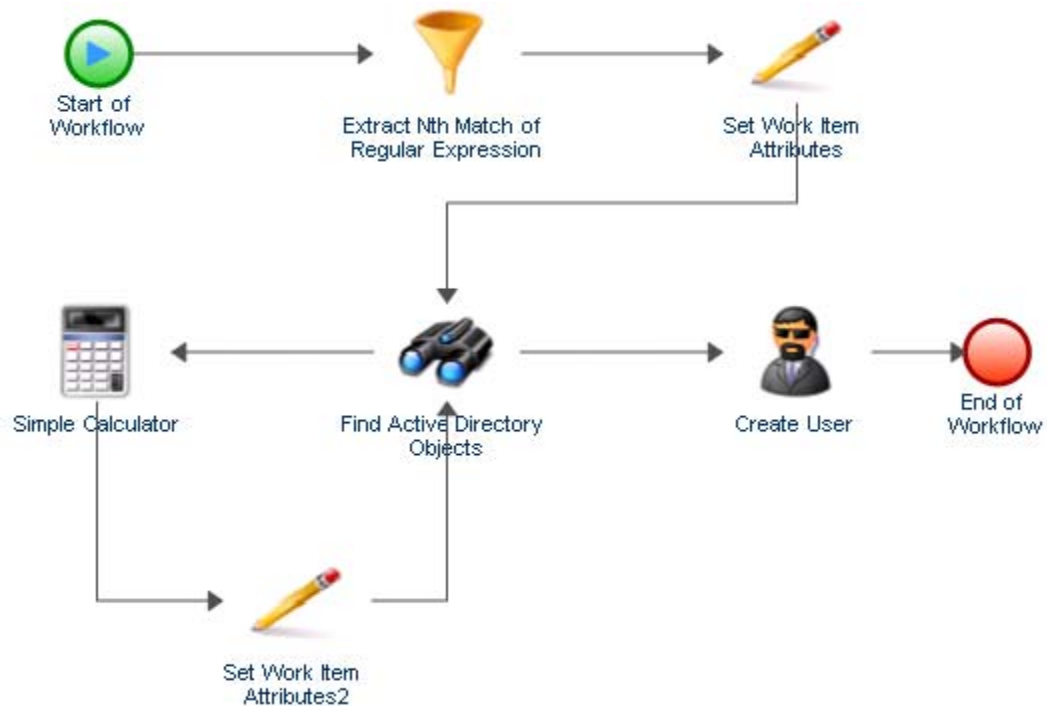
To create a workflow that creates a new user account using DRA:

- 1 In the Library Browser, drag the following activities from the specified activity libraries to the design grid:

Activity	Library
Start of Workflow	Basic Workflow Control
Extract Nth Match of Regular Expression	Data Manipulation
Set Work Item Attributes (2)	Data Manipulation
Find Active Directory Objects	NetIQ Directory and Resource Administrator
Create User	NetIQ Directory and Resource Administrator
Simple Calculator	Data Manipulation
End of Workflow	Basic Workflow Control

- 2 Use connectors to join the activities, in the following order:

- ◆ Start of Workflow to Extract Nth Match of Regular Expression
- ◆ Extract Nth Match of Regular Expression to Set Work Item Attributes (1)
- ◆ Set Work Item Attributes (1) to Find Active Directory Objects
- ◆ Find Active Directory Objects to Set Work Item Attributes (2)
- ◆ Set Work Item Attributes (2) of Simple Calculator
- ◆ Simple Calculator to Find Active Directory Objects
- ◆ Find Active Directory Objects to Create User
- ◆ Create User to End of Workflow



- 3 On the View tab, in the Properties group, click **Workflow Properties**.
- 4 On the Work Item tab, under Custom Work Item Attributes, add eight attributes with the following properties:

	Attribute 1	Attribute 2	Attribute 3	Attribute 4
Name	FirstName	FirstInitial	LastName	UserLogon
Display Name	First Name	First Initial	Last Name	User Logon
Type	String	String	String	String
Value				
Description				
Hidden	No	Yes	No	No
Read Only	Yes		Yes	Yes

	Attribute 5	Attribute 6	Attribute 7	Attribute 8
Name	Department	Createln	Searchln	Counter
Display Name	Department	Create In	Search In	Counter
Type	String	String	String	Integer Number
Value				0
Description				
Hidden	Yes	No	Yes	Yes
Read Only		Yes		

At run time, the workflow populates the custom work item attributes with values it retrieves from the email that initiated the work item. Activities in the workflow retrieve the values from the work item attributes during the course of the workflow.

- 5 On the Workflow Properties window, click **OK**.
- 6 Double-click the **Start of Workflow** activity, and then complete the following steps:
 - 6a On the Trigger tab, click **Add**.
 - 6b Under Available Triggers, click **Create New Trigger**.
 - 6c On the Create Trigger window, in the Trigger Name field, type `New User Email`.
 - 6d On the Initiate New Work Item If tab, click **New Template**.
 - 6e Under Expression Templates, click **Any event of a specific type and attribute value**, and then click **OK**.
 - 6f Click **<event type>**.
 - 6g Under Event Type, click **Exchange.New Message Event**, and then click **OK**.
 - 6h Click **<event attribute>**.
 - 6i Under Event Attribute, click **Subject**, and then click **OK**.
 - 6j Click **<event attribute value>**.
 - 6k In the text box, type `New user account`, and then click **OK**.

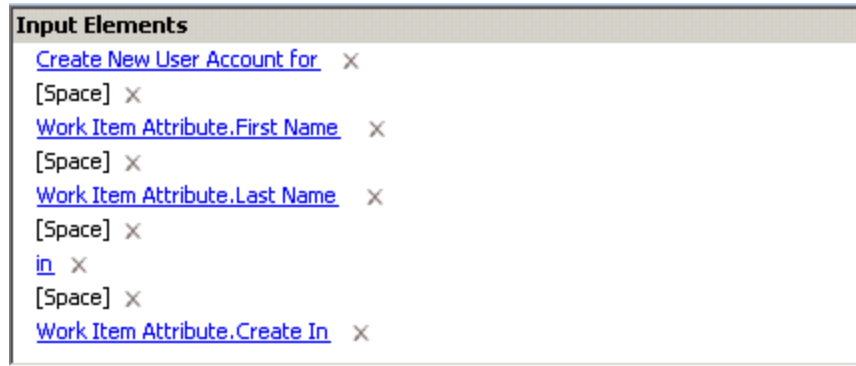
- 6l On the Create New Trigger window, click **OK**.
- 6m Under Associated Triggers, verify the new trigger is present, and then click **OK**.
The trigger only initiates a work item if an email message with a matching subject line arrives in a monitored mailbox.
- 7 Double-click the **Extract Nth Match of Regular Expression** activity, and then complete the following steps:
 - 7a On the General tab, in the Name field, type `Extract First Initial`.
 - 7b Click the **Values** tab.
 - 7c For the Input String parameter, select the **Use Input Builder** check box.
 - 7d In the Value field for the Input String parameter, click the **Input Builder** button.
 - 7e On the Input Builder window, click **New Input Element**, and then click **Event Attribute**.
 - 7f Under Event Type, select **Exchange.New Message Event**, and then click **OK**.
 - 7g Click **<Event Attribute>**.
 - 7h Under Event Attribute, click **Text Body**.
 - 7i On the Attribute Qualifiers tab, select the **Use Word** check box.
 - 7j In the Value field, click the **Browse** button.
 - 7k On the Use Word window, select the **Use word #** check box.
 - 7l In the Use word # field, type `1`, and then click **OK**.
 - 7m On the Input Builder window, click **OK**.
 - 7n In the Regular Expression parameter, type `(.)`.
 - 7o In the Segments parameter, type `1(1)`.
 - 7p Select the **Output as String** check box.
 - 7q In the Output String Separator parameter, type a comma `,`.
 - 7r Click **OK**.
- 8 Double-click the first **Set Work Item Attributes** activity.
- 9 On the Attributes tab, under Available Custom Attributes, select each of the following attributes, and then click **Add**, in the following order:
 - ◆ First Initial
 - ◆ First Name
 - ◆ Last Name
 - ◆ User Logon Name
 - ◆ Create In
 - ◆ Search In
 - ◆ Department
 - ◆ Subject
- 10 Under Configured Custom Attributes, for the First Initial attribute, complete the following steps:
 - 10a Select the **Use Input Builder** check box.
 - 10b In the Value field for the First Initial attribute, click the **Input Builder** button.
 - 10c On the Input Builder window, click **New Input Element**, and then click **Activity Output Parameter**.

- 10d Under Output Parameter Name, expand Extract First Initial, select **Result String**, and then click **OK**.
- 10e On the Input Builder window, click **OK**.
- 11 Under Configured Custom Attributes, for the First Name attribute, complete the following steps:
 - 11a Select the **Use Input Builder** check box.
 - 11b In the Value field for the First Name attribute, click the **Input Builder** button.
 - 11c On the Input Builder window, click **New Input Element**, and then click **Event Attribute**.
 - 11d Under Event Type, select **Exchange.New Message Event**, and then click **OK**.
 - 11e Click **<Event Attribute>**.
 - 11f Under Event Attribute, click **Text Body**.
 - 11g On the Attribute Qualifiers tab, select the **Use Word** check box.
 - 11h In the Value field, click the **Browse** button.
 - 11i On the Use Word window, select the **Use word #** check box.
 - 11j In the Use word # field, type 1, and then click **OK**.
 - 11k On the Input Builder window, click **OK**.
- 12 Under Configured Custom Attributes, for the Last Name attribute, complete the following steps:
 - 12a Select the **Use Input Builder** check box.
 - 12b In the Value field for the Last Name attribute, click the **Input Builder** button.
 - 12c On the Input Builder window, click **New Input Element**, and then click **Event Attribute**.
 - 12d Under Event Type, select **Exchange.New Message Event**, and then click **OK**.
 - 12e Click **<Event Attribute>**.
 - 12f Under Event Attribute, click **Text Body**.
 - 12g On the Attribute Qualifiers tab, select the **Use Word** check box.
 - 12h In the Value field, click the **Browse** button.
 - 12i On the Use Word window, select the **Use word #** check box.
 - 12j In the Use word # field, type 2, and then click **OK**.
 - 12k On the Input Builder window, click **OK**.
- 13 Under Configured Custom Attributes, for the User Logon Name attribute, complete the following steps:
 - 13a Select the **Use Input Builder** check box.
 - 13b In the Value field for the User Logon Name attribute, click the **Input Builder** button.
 - 13c On the Input Builder window, click **New Input Element**, and then click **Event Attribute**.
 - 13d Under Event Type, select **Exchange.New Message Event**, and then click **OK**.
 - 13e Click **<Event Attribute>**.
 - 13f Under Event Attribute, click **Text Body**.
 - 13g On the Attribute Qualifiers tab, select the **Use Word** check box.
 - 13h In the Value field, click the **Browse** button.
 - 13i On the Use Word window, select the **Use word #** check box.
 - 13j In the Use word # field, type 2, and then click **OK**.
 - 13k Click **New Input Element**, and then click **Activity Output Parameter**.

- 13l Under Output Parameter Name, expand **Extract First Initial**, select **Result String**, and then click **OK**.
- 13m On the Input Builder window, click **OK**.
- 14 Under Configured Custom Attributes, for the Create In attribute, complete the following steps:
 - 14a Select the **Use Input Builder** check box.
 - 14b In the Value field for the Create In attribute, click the **Input Builder** button.
 - 14c On the Input Builder window, click **New Input Element**, and then click **Event Attribute**.
 - 14d Under Event Type, select **Exchange.New Message Event**, and then click **OK**.
 - 14e Click **<Event Attribute>**.
 - 14f Under Event Attribute, click **Text Body**.
 - 14g On the Attribute Qualifiers tab, select the **Use Word** check box.
 - 14h In the Value field, click the **Browse** button.
 - 14i On the Use Word window, select the **Use word #** check box.
 - 14j In the Use word # field, type 3, and then click **OK**.
 - 14k Click **New Input Element**, and then click **Free-Form Text**.
 - 14l In the free-form text box, type \, and then click **OK**.
 - 14m Click **New Input Element**, and then click **Event Attribute**.
 - 14n Under Event Type, select **Exchange.New Message Event**, and then click **OK**.
 - 14o Click **<Event Attribute>**.
 - 14p Under Event Attribute, click **Text Body**.
 - 14q On the Attribute Qualifiers tab, select the **Use Word** check box.
 - 14r In the Value field, click the **Browse** button.
 - 14s On the Use Word window, select the **Use word #** check box.
 - 14t In the Use word # field, type 4, and then click **OK**.
 - 14u Click **New Input Element**, and then click **Free-Form Text**.
 - 14v In the free-form text box, type \, and then click **OK**.
 - 14w Click **New Input Element**, and then click **Event Attribute**.
 - 14x Under Event Type, select **Exchange.New Message Event**, and then click **OK**.
 - 14y Click **<Event Attribute>**.
 - 14z Under Event Attribute, click **Text Body**.
 - 14aa On the Attribute Qualifiers tab, select the **Use Word** check box.
 - 14ab In the Value field, click the **Browse** button.
 - 14ac On the Use Word window, select the **Use word #** check box.
 - 14ad In the Use word # field, type 5, and then click **OK**.
 - 14ae On the Input Builder window, click **OK**.
- 15 Under Configured Custom Attributes, for the Search In attribute, complete the following steps:
 - 15a Select the **Use Input Builder** check box.
 - 15b In the Value field for the Search In attribute, click the **Input Builder** button.
 - 15c On the Input Builder window, click **New Input Element**, and then click **Event Attribute**.
 - 15d Under Event Type, select **Exchange.New Message Event**, and then click **OK**.

- 15e Click **<Event Attribute>**.
 - 15f Under Event Attribute, click **Text Body**.
 - 15g On the Attribute Qualifiers tab, select the **Use Word** check box.
 - 15h In the Value field, click the **Browse** button.
 - 15i On the Use Word window, select the **Use word #** check box.
 - 15j In the Use word # field, type 3, and then click **OK**.
 - 15k On the Input Builder window, click **OK**.
- 16 Under Configured Custom Attributes, for the Department attribute, complete the following steps:
- 16a Select the **Use Input Builder** check box.
 - 16b In the Value field for the Department attribute, click the **Input Builder** button.
 - 16c On the Input Builder window, click **New Input Element**, and then click **Event Attribute**.
 - 16d Under Event Type, select **Exchange.New Message Event**, and then click **OK**.
 - 16e Click **<Event Attribute>**.
 - 16f Under Event Attribute, click **Text Body**.
 - 16g On the Attribute Qualifiers tab, select the **Use Word** check box.
 - 16h In the Value field, click the **Browse** button.
 - 16i On the Use Word window, select the **Use word #** check box.
 - 16j In the Use word # field, type 5, and then click **OK**.
 - 16k On the Input Builder window, click **OK**.
- 17 Under Configured Custom Attributes, for the Subject attribute, complete the following steps:
- 17a Select the **Use Input Builder** check box.
 - 17b In the Value field for the Subject attribute, click the **Input Builder** button.
 - 17c On the Input Builder window, click **New Input Element**, and then click **Free-Form Text**.
 - 17d In the free-form text box, type *Create New User Account for*, and then click **OK**.
 - 17e Click **New Input Element**, and then click **Space**.
 - 17f Click **New Input Element**, and then click **Work Item Attribute**.
 - 17g Under Custom Work Item Attributes, click **First Name**, and then click **OK**.
 - 17h Click **New Input Element**, and then click **Space**.
 - 17i Click **New Input Element**, and then click **Work Item Attribute**.
 - 17j Under Custom Work Item Attributes, click **Last Name**, and then click **OK**.
 - 17k Click **New Input Element**, and then click **Space**.
 - 17l Click **New Input Element**, and then click **Free-Form Text**.
 - 17m In the free-form text box, type *in*, and then click **OK**.
 - 17n Click **New Input Element**, and then click **Space**.
 - 17o Click **New Input Element**, and then click **Work Item Attribute**.
 - 17p Under Custom Work Item Attributes, click **Create In**, and then click **OK**.

17q On the Input Builder window, click **OK**.



At run time, the workflow populates the subject of the work item with the new user's name and location. Process Operators can check the progress of the work item in the Operations Console.

- 18 On the activity properties window, click **OK**.
- 19 Double-click the Find Active Directory Objects activity, and then complete the following steps:
 - 19a Click the **Values** tab.
 - 19b For the Search In parameter, select the **Use Input Builder** check box.
 - 19c In the Value field for the Search In parameter, click the **Input Builder** button.
 - 19d On the Input Builder window, click **New Input Element**, and then click **Work Item Attribute**.
 - 19e Under Custom Work Item Attributes, click Search In, and then click **OK**.
 - 19f Select the **Search Specific Object Types** radio button.
 - 19g Select the **User Accounts** check box.
 - 19h Expand **User Accounts**.
 - 19i For the User Logon Name Pattern parameter, select the **Use Input Builder** check box.
 - 19j In the Value field for the User Logon Name Pattern parameter, click the **Input Builder** button.
 - 19k On the Input Builder window, click **New Input Element**, and then click **Work Item Attribute**.
 - 19l Under Custom Work Item Attributes, click **User Logon Name**, and then click **OK**.
 - 19m On the activity properties window, click **OK**.
- 20 Double-click the connector between Find Active Directory Objects and Simple Calculator, and then complete the following steps:
 - 20a In the Name field, type `User ID already in use`.
 - 20b Select **Only traverse this connector when the following conditions are met**.
 - 20c Under Conditional Expression, click **Edit Expression**.
 - 20d On the Expression Editor window, click **New Template**, and then click **Parameter Expression**.
 - 20e Click `<parameter name>`, expand **Find Active Directory Objects**, select **Number of Object Locators**, and then click **OK**.
 - 20f Click **is equal to**, and then click **is greater than**.
 - 20g Click `<attribute value>`, and then click **Edit Simple Value**.
 - 20h In the text field, type `0`, and then click **OK**.

- 20i On the Expression Editor window, click **OK**.
- 20j On the Connector Properties window, click **OK**.
- 21 Double-click the **Simple Calculator** activity, and then complete the following steps:
 - 21a Click the **Values** tab.
 - 21b For the Expression parameter, select the **Use Input Builder** check box.
 - 21c In the Value field for the Expression parameter, click the **Input Builder** button.
 - 21d On the Input Builder window, click **New Input Element**, and then click **Work Item Attribute**.
 - 21e Under Custom Work Item Attributes, click **Counter**, and then click **OK**.
 - 21f Click **New Input Element**, and then click **Free-Form Text**.
 - 21g In the free-form text box, type +1, and then click **OK**.
 - 21h On the Input Builder window, click **OK**.
 - 21i On the activity properties window, click **OK**.
- 22 Double-click the second **Set Work Item Attributes** activity.
- 23 On the General tab, in the Name field, type `Increment User Logon Name`.
- 24 On the Attributes tab, under Available Custom Attributes, select the following:
 - ◆ User Logon Name
 - ◆ Counter
- 25 Under Configured Custom Attributes, for the User Logon Name attribute, complete the following steps:
 - 25a Select the **Use Input Builder** check box.
 - 25b In the Value field for the User Logon Name attribute, click the **Input Builder** button.
 - 25c On the Input Builder window, click **New Input Element**, and then click **Work Item Attribute**.
 - 25d Under Custom Work Item Attributes, click **User Logon Name**, and then click **OK**.
 - 25e Click **New Input Element**, and then click **Activity Output Parameter**.
 - 25f Under Output Parameter Name, expand **Simple Calculator**, select **Result**, and then click **OK**.
 - 25g On the Input Builder window, click **OK**.
- 26 Under Configured Custom Attributes, for the Counter attribute, complete the following steps:
 - 26a Select the **Use Input Builder** check box.
 - 26b In the Value field for the Counter attribute, click the **Input Builder** button.
 - 26c On the Input Builder window, click **New Input Element**, and then click **Activity Output Parameter**.
 - 26d Under Output Parameter Name, expand **Simple Calculator**, select **Result**, and then click **OK**.
 - 26e On the Input Builder window, click **OK**.
- 27 On the activity properties window, click **OK**.
- 28 Double-click the connector between Find Active Directory Objects and Create User, and then complete the following steps:
 - 28a In the Name field, type `User ID available`.
 - 28b Select **Only traverse this connector when the following conditions are met**.
 - 28c Under Conditional Expression, click **Edit Expression**.

- 28d On the Expression Editor window, click **New Template**, and then click **Parameter Expression**.
- 28e Click **<parameter name>**, expand **Find Active Directory Objects**, select **Number of Object Locators**, and then click **OK**.
- 28f Click **<attribute value>**, and then click **Edit Simple Value**.
- 28g In the text field, type 0, and then click **OK**.
- 28h On the Expression Editor window, click **OK**.
- 28i On the Connector Properties window, click **OK**.
- 29 Double-click the **Create User** activity, and then click the **Values** tab.
- 30 For the Create In parameter, complete the following steps:
 - 30a Select the **Use Input Builder** check box.
 - 30b In the Value field for the Create In parameter, click the **Input Builder** button.
 - 30c On the Input Builder window, click **New Input Element**, and then click **Work Item Attribute**.
 - 30d Under Custom Work Item Attributes, click **Create In**, and then click **OK**.
 - 30e On the Input Builder window, click **OK**.
- 31 For the Full Name parameter, complete the following steps:
 - 31a Select the **Use Input Builder** check box.
 - 31b In the Value field for the Full Name parameter, click the **Input Builder** button.
 - 31c On the Input Builder window, click **New Input Element**, and then click **Work Item Attribute**.
 - 31d Under Custom Work Item Attributes, click **First Name**, and then click **OK**.
 - 31e Click **New Input Element**, and then click **Space**.
 - 31f Click **New Input Element**, and then click **Work Item Attribute**.
 - 31g Under Custom Work Item Attributes, click **Last Name**, and then click **OK**.
 - 31h On the Input Builder window, click **OK**.
- 32 For the First Name parameter, complete the following steps:
 - 32a Select the **Use Input Builder** check box.
 - 32b In the Value field for the First Name parameter, click the **Input Builder** button.
 - 32c On the Input Builder window, click **New Input Element**, and then click **Work Item Attribute**.
 - 32d Under Custom Work Item Attributes, click **First Name**, and then click **OK**.
 - 32e On the Input Builder window, click **OK**.
- 33 For the Last Name parameter, complete the following steps:
 - 33a Select the **Use Input Builder** check box.
 - 33b In the Value field for the Last Name parameter, click the **Input Builder** button.
 - 33c On the Input Builder window, click **New Input Element**, and then click **Work Item Attribute**.
 - 33d Under Custom Work Item Attributes, click **Last Name**, and then click **OK**.
 - 33e On the Input Builder window, click **OK**.
- 34 For the Pre-Windows 2000 Logon Name parameter, complete the following steps:
 - 34a Select the **Use Input Builder** check box.
 - 34b In the Value field for the Pre-Windows 2000 Logon Name parameter, click the **Input Builder** button.
 - 34c On the Input Builder window, click **New Input Element**, and then click **Work Item Attribute**.

- 34d Under Custom Work Item Attributes, click **User Logon Name**, and then click **OK**.
- 34e On the Input Builder window, click **OK**.
- 35 For the User Logon Name parameter, complete the following steps:
 - 35a Select the **Use Input Builder** check box.
 - 35b In the Value field for the User Logon Name parameter, click the **Input Builder** button.
 - 35c On the Input Builder window, click **New Input Element**, and then click **Work Item Attribute**.
 - 35d Under Custom Work Item Attributes, click **User Logon Name**, and then click **OK**.
 - 35e On the Input Builder window, click **OK**.
- 36 For the Department parameter, complete the following steps:
 - 36a Select the **Use Input Builder** check box.
 - 36b In the Value field for the Department parameter, click the **Input Builder** button.
 - 36c On the Input Builder window, click **New Input Element**, and then click **Work Item Attribute**.
 - 36d Under Custom Work Item Attributes, click **Department**, and then click **OK**.
 - 36e On the Input Builder window, click **OK**.
- 37 On the activity properties window, click **OK**.
- 38 In the Workflow Designer, on the Workflow tab, in the File group, click **Save Workflow**.
- 39 On the Workflow tab, in the File group, click the **Close Workflow Designer**.

Adding Functionality to the Workflow

To increase the complexity and capability of this workflow you might add one or more of the following:

- ♦ **POP3 email support** - You can add another trigger to the Start of Workflow activity to respond to a properly formatted email message arriving in a monitored POP3 mailbox.
- ♦ **Complex email formatting** - The email message matching the trigger can contain more complex formatting, from which you can retrieve information to populate more input parameters on the Create User activity or additional custom work item attributes.
- ♦ **Send credentials to system administrator** - You can send an email containing the new user's ID and password to the system administrator, who can then provide the credentials to the user.

4.9 Debugging a Workflow

Debug mode in the Workflow Designer allows you to test a workflow for errors at design time, without putting the workflow revision into production. In debug mode, you can:

- ♦ View the status of each activity and connector.
- ♦ Open an activity while it is executing, and view the values of its input and output parameters on the **Execution Results** tab.

NOTE: If the activity executes multiple times, such as in a loop, the **Execution Results** tab displays the results of each execution, in reverse chronological order. If the activity executes again while you are viewing the results, you can refresh the view to add the results of the most recent execution.

- ♦ Test and respond to user input forms.

NOTE: To respond to a user input form in debug mode, double-click the appropriate Collect Input from User activity. When the activity is waiting for input in debug mode, the input form on the **Input Form** tab of the activity is fully functional.

- ◆ View the current value of custom work item attributes.
- ◆ View the dynamic values assigned by Input Builder in conditional connector and event filter expressions. For example, the following conditional connector expression allows the workflow to traverse the connector only if the number of object locators retrieved by the Find Computers activity is equal to the number of rows found in a table by the Get Table Size activity:

```
Activity Output Parameter Find Computers.Number of Object Locators is equal to {Get Table Size.Number of Rows}
```

In debug mode, if you double-click the connector, you can see the value for {Get Table Size.Number of Rows}.

- ◆ Add, remove, enable, disable breakpoints.
- ◆ Step through activities and connectors one at a time.

If another revision of the workflow is in production, you can debug the current revision without conflict.

NOTE: When you execute a workflow in debug mode, Workflow Automation creates a temporary work item. Process Operators do not see the temporary work item in the Operations Console.

Breakpoints allow you to pause the workflow. You can associate breakpoints with activities and connectors in either design mode or debug mode. The workflow pauses at the breakpoint, prior to executing or traversing the associated activity or connector. For example, if you want to review the results of an activity, you can add a breakpoint to the connector immediately after the activity. After the activity executes, the workflow pauses at the breakpoint, and does not traverse the connector. After you review the results of the activity, you can continue debugging the workflow. If you need to make changes to the activity, you can exit debug mode and modify the activity.

Breakpoints remain associated with the revision until you put it into production.

4.10 Working with Times in Workflow Automation

The Workflow Automation consoles display times and allow you to type times according to the regional settings on your computer, but Workflow Automation stores all times in GMT (Greenwich Mean Time). When working with activities in the Workflow Designer, all activity output parameters related to time return a `time_t` value, which is the number of seconds since the start of the Unix epoch (1/1/1970 12:00:00 AM).

The **Time (LOCAL)** output parameter on the Get Current Time and Date activity returns a `time_t` value for your local time zone. All other time-related output parameters, including the Get Current Time and Date activity's other output parameter, **Time**, return a `time_t` value that includes the GMT offset.

For example, if the Get Current Time and Date activity retrieves a date and time of 3:00 PM CST on Wednesday, July 16, 2008, the **Time (LOCAL)** parameter returns 1216177200 and the **Time** parameter returns 1216195200. The difference is 18,000 seconds, or five hours, which is the time zone offset between CST and GMT.

If you use the Input Builder to reference an output parameter from the Get Current Time and Date activity, ensure you select the appropriate parameter. For example, if you configure the Check Time Against a Schedule activity to retrieve the time from the Get Current Time and Date activity, reference the **Time** parameter to correctly account for the GMT offset.

4.11 Validating a Workflow

A workflow must pass validation before you can put it into production. You can manually validate a workflow at any time during the design session. Workflow Automation automatically validates a workflow when you save the workflow or attempt to put a workflow revision into production.

NOTE

- ♦ Due to the complexity of some workflows, it may not be possible to complete a workflow in one design session. You can save the workflow and return to it at a later time to perform validation.
- ♦ The Workflow Designer checks that you have a Start of Workflow activity, an End of Workflow activity, and that the connectors are properly attached. The Workflow Designer does not validate workflow logic.

For more information about putting a workflow into production, see [Putting a Workflow Revision into Production](#).

4.12 Understanding Supporting Analysis

When designing a workflow, you can configure one or more data source-specific activities to collect information related to the process, either in the initial stages of investigation or after the work item has been closed. Process Operators can view supporting analysis results in the Operations Console.

In the initial stages of investigation, supporting analysis activities can collect information from all data sources related to a work item to ensure that a Process Operator has as much information as possible to identify the root cause.

After a work item completes, supporting analysis activities can collect information from all related data sources to verify the incident has been correctly resolved.

NOTE: Supporting analysis activities are specific to a data source. For more information about a particular activity, see its related Help.

If a Process Operator views supporting analysis results before all supporting analysis steps are complete, the Operations Console displays the completed results (if any) and indicates which steps are still running.

4.13 Understanding Workflow Priority

When Workflow Automation receives an event from a data source, Workflow Automation evaluates the workflows currently in production to find a matching trigger. If a workflow has multiple triggers associated with it, Workflow Automation evaluates the triggers in the order they are specified on the Start of Workflow activity. If the event matches a trigger, Workflow Automation initiates a work item.

If the event matches triggers for multiple workflows, Workflow Automation prioritizes the workflows based on their assigned trigger processing order, and then starts the appropriate workflows based on the event consumption policy assigned to each matching workflow.

4.13.1 Trigger Processing Order

When an incoming event matches the same trigger on multiple workflows, Workflow Automation prioritizes the workflows using the trigger processing order setting for each workflow, placing the workflow with the highest setting at the top of the list. Workflow Automation checks the event consumption policy for the first workflow, and then continues as directed. The maximum setting is 100.

WARNING: When setting the trigger processing order for a group of workflows matching the same events, ensure only one has the highest setting. If you configure two workflows to be at the top of the list, Workflow Automation picks the first one it reads from the database, based on database syntax.

4.13.2 Event Consumption Policy

The event consumption policy for a workflow determines how the workflow responds to an incoming event that matches its assigned trigger. Each policy marks the event at the time of use, even if the policy prevents the event from matching any workflows lower in the trigger processing order.

To set the consumption policy for multiple workflows that use the same trigger, use one of the following options:

- ◆ **Consume** - Use the event and stop. The event will not start any matching workflows lower in the trigger processing order.
- ◆ **Mark and Pass** - Use the event and continue. The event may start additional matching workflows lower in the trigger processing order.

To set the consumption policy for multiple workflows that use different triggers, use one of the following options:

- ◆ **Use if Not Marked** - Use the event if it has not already started a matching workflow higher in the trigger processing order, and then stop. The event will not start any matching workflows lower in the trigger processing order.
- ◆ **Use if Not Marked and Pass** - Use the event if it has not already started a matching workflow higher in the trigger processing order, and then continue. The event may start additional matching workflows lower in the trigger processing order.

4.13.3 Setting the Policies for a Workflow

You can set the event consumption policy and the trigger processing order for a workflow revision that is under construction.

To set the trigger processing and event consumption policies for a workflow:

- 1 In the Navigation pane, click **Processes**.
- 2 In the left pane, select the appropriate process.
- 3 In the Workflow Revisions list, select the workflow revision for which you want to set the event consumption policy.
- 4 In the Workflow Revision Tasks list, click **View Workflow Revision Properties**.
- 5 On the Workflow Revision Properties window, click the **Workflow** tab.
- 6 Under Policies, provide the appropriate information, and then click **OK**.

4.14 Understanding Revision Control

A workflow must be in production to receive and respond to events. Multiple workflows can be in production, but a process can have only one workflow revision in production at a time.

4.14.1 Putting a Workflow Revision into Production

When you have completed a workflow, you can put it into production. Once the workflow is in production, Workflow Automation immediately searches for matches among new events.

NOTE

- ◆ Once you put a workflow into production, it cannot return to being under construction. If you want to edit a workflow that is currently in production, you must clone it and create a new revision. For more information about cloning workflow revisions, see [Cloning a Workflow Revision](#).
 - ◆ A process may have only one workflow revision in production at a time. If a revision is already in production, Workflow Automation automatically replaces it with the new revision.
 - ◆ When you put a workflow revision with abstract trigger revisions into production for the first time, Workflow Automation assigns the latest trigger revision according to the rule for the specified abstract revision. For more information about abstract trigger revisions, see [Understanding Abstract Trigger Revisions](#).
 - ◆ Once you put a workflow revision into production, Workflow Automation does not automatically upgrade the Task activities in the workflow to use newer versions of their associated activity modules. You must manually deploy these updates. For more information about updating Task activities with the latest activity modules, see [Deploying Activity Module Updates](#).
-

Workflow Automation automatically validates the revision's workflow when you put it into production. If the workflow passes validation, Workflow Automation assigns a version number and puts the revision into production.

4.14.2 Cloning a Workflow Revision

If you need to edit a workflow revision after putting it into production, you can clone it, creating an exact copy of the workflow in a new revision. The new workflow revision remains under construction until it is put into production.

To clone a workflow revision:

- 1 In the Navigation pane, click **Processes**.
- 2 In the left pane, select the process that contains the revision you want to clone.
- 3 In the Workflow Revisions list, select the revision you want to clone.
- 4 In the Workflow Revision Tasks list, click **Clone Revision**.
- 5 If you are cloning a workflow revision you built with a previous version of Workflow Automation, upgrade the workflow to use the latest versions of the activities. For more information about upgrading an older workflow, see [Deploying Activity Module Updates](#).
- 6 In the Workflow Designer, make the appropriate changes.
- 7 On the Workflow tab, in the File group, click **Save Workflow**.
- 8 Close the Workflow Designer.

- 9 In the Workflow Revision Tasks list, click **Check In Revision**.
- 10 If you are finished making changes, put the revision into production.

4.14.3 Taking a Workflow Revision Out of Production

You can manually take a workflow revision out of production at any time. For example, you might want to take a workflow revision out of production to prevent incoming events from initiating work items while you clone the workflow and customize the new workflow revision.

To take a workflow revision out of production:

- 1 In the Navigation pane, click **Processes**.
- 2 In the left pane, select the process that contains the workflow revision you want to take out of production.
- 3 In the Workflow Revisions list, select the workflow revision you want to take out of production.
- 4 In the Workflow Revision Tasks list, click **Take Revision Out of Production**.

4.14.4 Exporting a Workflow

You can export the workflow associated with a workflow revision to an `.nxpr` file to share across your organizational and geographical structure. The export file includes the workflow's associated triggers, trigger event definitions, and custom work item attributes. Process Authors may import the `.nxpr` file to another process and modify the workflow to suit their needs. For more information about importing a workflow revision, see [Importing a Workflow](#).

TIP: You can also export any business services referenced in the workflow.

To export a workflow:

- 1 In the Navigation pane, click **Processes**.
- 2 In the left pane, select the appropriate process.
- 3 In the Workflow Revisions list, select the revision associated with the workflow you want to export.
- 4 In the Workflow Revision Tasks list, click **Export Workflow**.
- 5 On the Export Workflow window, specify the location and file name, and then click **Save**.

4.14.5 Importing a Workflow

Using the Import Workflow wizard, you can import a workflow revision into a process. The imported workflow displays in the **Workflow Revisions** list as the most recent revision.

NOTE: Process Authors create import files when they export a workflow to a file. For more information about exporting a workflow, see [Exporting a Workflow](#).

To import a workflow revision into a process:

- 1 In the Navigation pane, click **Processes**.
- 2 In the left pane, select the process for which you want to import a workflow.

- 3 In the Workflow Revision Tasks list, click **Import Workflow**.
- 4 Complete the Import Workflow Wizard, and then click **Finish**.

4.14.6 Customizing Work Item Properties for a Workflow

Before you put a workflow into production, you can specify the custom properties it assigns to the work items it creates. For more information about putting a workflow into production, see [Putting a Workflow Revision into Production](#).

To customize work item properties for a workflow:

- 1 In the Navigation pane, click **Processes**.
- 2 In the left pane, select the appropriate process.
- 3 In the Workflow Revisions list, select the workflow revision for which you want to create or modify custom work item properties.
- 4 In the Workflow Revision Tasks list, click **View Revision Properties**.
- 5 On the Workflow Revision Properties window, click the **Work Item** tab.
- 6 Under Properties, provide the appropriate information, and then click **OK**.

4.14.7 Customizing Work Item Attributes

Before you put a workflow into production, you can assign custom attributes to the work items it creates. These attributes can be internal to the workflow or visible to users on the Operations Console.

NOTE: If a custom work item attribute with the same name is passed to the work item by a trigger, the value you assign here overrides the value from the trigger.

To customize work item attributes for a workflow:

- 1 In the Navigation pane, click **Processes**.
- 2 In the left pane, select the appropriate process.
- 3 In the Workflow Revisions list, select the workflow revision for which you want to create or modify custom work item attributes.
- 4 In the Workflow Revision Tasks list, click **View Revision Properties**.
- 5 On the Workflow Revision Properties window, click the **Work Item** tab.
- 6 If you want to create a custom work item attribute, complete the following steps:
 - 6a Under Custom Work Item Attributes, click **Add**.
 - 6b On the New Custom Work Item Attribute window, provide the appropriate information, and then click **OK**.
- 7 If you want to edit a custom work item attribute, complete the following steps:
 - 7a Under Custom Work Item Attributes, select the attribute you want to edit, and then click **Edit**.
 - 7b On the Edit Custom Work Item Attribute window, make the appropriate changes, and then click **OK**.
- 8 On the Workflow Revision Properties window, click **OK**.

4.15 Deploying Activity Module Updates

When you install a new version of an activity module, you can update existing processes to use the latest version. For more information about activity modules, see [Understanding Workflow Activities](#).

To deploy activity module updates:

- 1 In the Navigation pane, click **Processes**.
- 2 If you want to deploy activity module updates to all processes, complete the following steps:
 - 2a In the Global Tasks list, click **Deploy Activity Module Updates**.
 - 2b Complete the Deploy Activity Module Updates Wizard.
- 3 If you want to deploy activity module updates to a specific process, complete the following steps:
 - 3a In the left pane, select the appropriate process.
 - 3b In the Workflow Revisions list, select the revision you want to update.
 - 3c In the Workflow Revision Tasks list, click **Deploy Activity Module Updates**.
 - 3d Complete the Deploy Activity Module Updates Wizard.
- 4 Click **Finish** to deploy the updated activity modules.

5 Understanding Activity Libraries

Activity libraries in the Workflow Designer contain pre-configured workflow activities and connectors that you can use to build a workflow.

When you install newer versions of activity modules, Workflow Automation automatically updates activity libraries to use the latest version, ensuring Process Authors have the most recent versions at design time. Workflow Automation does not upgrade existing workflows. For more information about manually deploying activity module updates to existing workflows, see [Deploying Activity Module Updates](#).

The Workflow Designer supports the following types of activity libraries:

Product-specific Libraries

Contain activities related to a specific adapter, such as the AppManager adapter. The Workflow Automation Administrator installs product-specific libraries as part of installing and configuring an adapter.

Default Libraries

Contain Workflow Automation activities that you can use in any workflow.

Custom Libraries

A library you create to contain pre-configured workflow activities that you use often. Custom libraries are specific to your environment and can contain a combination of default activities, and product-specific activities. For more information about creating your own libraries, see [Creating a Custom Library](#).

5.1 Creating a Custom Library

You can create a custom activity library containing a set of pre-configured workflow activities you use often.

To create a custom library in the Workflow Designer:

- 1 Drag the activities and connectors you want from existing libraries to the design grid.
- 2 Configure the activities to suit your needs.
- 3 On the Activity Library tab, in the Library group, click **Create New Library**.
- 4 Drag the customized activities from the design grid to the new library.

NOTE: When you add a customized activity to a library, Workflow Automation checks for a newer version of the activity. If a newer version is available, Workflow Automation automatically updates the activity.

- 5 On the Activity Library tab, in the Properties group, click **Library Properties**.
- 6 On the Library Properties window, provide the appropriate information, and then click **OK**.
- 7 On the Activity Library tab, in the Library group, click **Save Library**.

5.2 Customizing an Existing Activity Library

After you create an activity library, you can add, delete, or customize the activities associated with it at any time.

To customize an existing library in the Workflow Designer:

- 1 If you want to add new activities to the existing library, drag the activities you want from existing libraries to the design grid.
- 2 If the library is already open, select it in the **Library Browser**.
- 3 If the library is not already open, complete the following steps:
 - 3a On the Activity Library tab, in the Library group, click **Open Libraries**.
 - 3b On the Open Activity Library window, select one or more libraries, and then click **OK**.
- 4 Configure the activities to suit your needs.
- 5 Drag the customized activities from the design grid to the library.
- 6 On the Activity Library tab, in the Library group, click **Save Library**.
- 7 If the Save Activity Library As window displays, type a new name in the **Name** field.

NOTE: When you customize a built-in library, you must change the name and save it as a new custom library.

- 8 Click **OK**.

5.3 Importing and Exporting Activity Libraries

You can import and export any activity library to share across your organizational and geographical structure. If newer versions of the activities in the imported library are available, Workflow Automation automatically updates the activity library to use the latest versions of the activities.