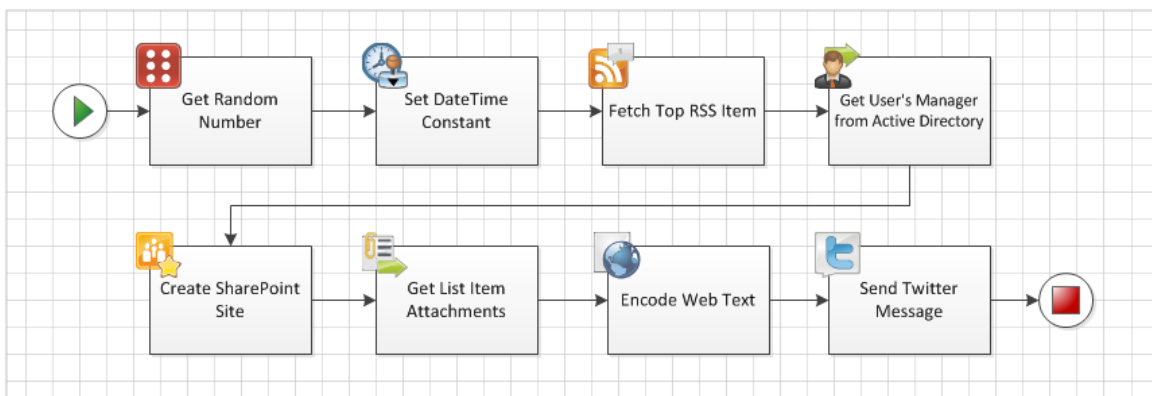


HarePoint Workflow Extensions Manual

For SharePoint Server 2010 and SharePoint Foundation 2010



Product version 1.4

28.05.2012

(This Page Intentionally Left Blank)

Table of Contents

Table of Contents	3
I. Introduction	11
II. Deployment.....	12
1. System requirements.....	12
2. Licensing and trial limitations	13
3. Installing HarePoint Workflow Extensions.....	14
4. Installing Client Applications.....	17
5. How to repair, upgrade or remove product	19
III. Product Overview	21
1. Getting started	21
2. Creating workflows in SharePoint Designer	22
3. Designing workflows in Microsoft Visio.....	23
4. Creating workflows in Visual Studio	24
5. Security issues.....	25
6. How to disable certain HarePoint activities.....	26
IV. HarePoint Activities and Conditions	28
1. Active Directory activities	28
1.1 Add Member to Active Directory Group.....	28
1.2 Change Active Directory Account Settings.....	28
1.3 Create Group in Active Directory.....	29
1.4 Create User in Active Directory.....	29
1.5 Delete Group from Active Directory	30
1.6 Delete User from Active Directory.....	30
1.7 Find User in Active Directory	30
1.8 Get Active Directory Groups Where User is Member.....	31
1.9 Get E-Mails of Active Directory Group Members	32
1.10 Get User's Attribute from Active Directory	32
1.11 Get User's Manager from Active Directory.....	33
1.12 Get User's Multivalued Attribute from Active Directory	33

1.13 Is User a Member of an Active Directory Group (HarePoint)	34
1.14 Move User into Organizational Unit	34
1.15 Remove Member from Active Directory Group.....	35
1.16 Set Active Directory Authentication (impersonate).....	35
1.17 Set User's Attribute in Active Directory	36
1.18 Set User's Multivalued Attribute in Active Directory.....	36
2. Administration activities	37
2.1 Activate Feature	37
2.2 Add Users to SharePoint Group	37
2.3 Add Users to Site.....	37
2.4 Create SharePoint Group	38
2.5 Create SharePoint Site	38
2.6 Create Site Collection.....	39
2.7 Deactivate Feature.....	39
2.8 Delete SharePoint Group	40
2.9 Delete SharePoint Site	40
2.10 Delete Site Collection.....	40
2.11 Get Site User Property	41
2.12 Is Role Assigned to User (HarePoint)	41
2.13 Is User a Member of a SharePoint Group (HarePoint)	42
2.14 Remove User from SharePoint Group	42
2.15 Remove User from Site	42
2.16 Rename Site at Url	42
2.17 Set SharePoint List Anonymous Access	43
2.18 Set SharePoint Site Anonymous Access.....	43
2.19 Set Site User Property	44
3. Arrays activities.....	44
3.1 Add Array Item	44
3.2 Clear Array.....	45
3.3 Create Array	45
3.4 Get Array Item.....	46

3.5 Get Array Items Count	46
3.6 Get Index of Array Item.....	46
3.7 Insert Array Item	47
3.8 Remove Array Item	48
3.9 Remove Array Item at	48
3.10 Remove Duplicates from Array	48
4. Date/Time activities	49
4.1 Converts Date from Local Time to UTC.....	49
4.2 Converts Date from UTC to Local Time.....	49
4.3 Converts Date to Text	49
4.4 Get Date Property	51
4.5 Set DateTime Constant	52
5. Development activities	52
5.1 Execute Custom Code	52
5.2 Execute PowerShell Script.....	54
5.3 Execute SQL Command	55
5.4 Query XML.....	57
5.5 Transform XML.....	59
5.6 Update XML.....	61
6. Document Conversion activities	63
6.1 Convert Document.....	63
6.2 Convert Folder at Url.....	64
6.3 Convert Library.....	65
6.4 Get Conversion Job Status	66
6.5 Wait Conversion Job	67
7. Document Library activities	67
7.1 Check In Item at Url.....	67
7.2 Check Out Item at Url.....	68
7.3 Copy or Move Document	68
7.4 Create Document from File.....	69
7.5 Create Document from Text	69

7.6 Create Folder in the Document Library	70
7.7 Create Folder in the Document Library at Url.....	70
7.8 Delete Folder in the Document Library.....	71
7.9 Delete Folder in the Document Library at Url.....	71
7.10 Discard Check Out Item at Url.....	71
7.11 Discard Publish Item	72
7.12 Download Documents to Folder	72
7.13 Get Document Content	73
7.14 Get Entry From Folder Property Bag.....	73
7.15 Publish Item	74
7.16 Rename Document at Url	74
7.17 Save Document to File	74
7.18 Set Network File Access	75
7.19 Upload Documents from Folder	76
8. E-Mail activities.....	76
8.1 Get E-mail Property.....	76
8.2 Receive All E-Mails to List	77
8.3 Receive E-Mail.....	78
8.4 Save E-Mail Attachments to List Item	79
8.5 Send E-Mail with Document as Attachment.....	80
8.6 Send E-Mail with File Attachments	80
8.7 Send E-Mail with Http File Attachment	81
8.8 Send E-Mail with InfoPath Attachments.....	82
8.9 Send E-Mail with List Item Attachments.....	84
8.10 Send E-Mail without Attachments	84
8.11 Set SMTP Server Options	85
9. Images activities.....	86
9.1 Convert Image.....	86
9.2 Convert Images in Folder	87
9.3 Convert Images in Library	87
9.4 Flip Image	88

9.5 Get EXIF Property from Image	88
9.6 Get Image Size.....	89
9.7 Is Image Compressed (HarePoint).....	91
9.8 Is Image Type Supported (HarePoint)	92
9.9 Resize Image	94
9.10 Resize Images in Folder	94
9.11 Resize Images in Library	95
9.12 Rotate Image	96
9.13 Set Transparent Color	96
10. Lists activities	97
10.1 Approve Item at Url	97
10.2 Calculate Aggregation Function	98
10.3 Copy InfoPath Attachments to Document Library.....	99
10.4 Copy List Item Attachments to Document Library.....	101
10.5 Copy List Item Attachments to List Item.....	101
10.6 Create List	101
10.7 Create List Item at Url	102
10.8 Delete All List Items	103
10.9 Delete Item at Url.....	104
10.10 Download InfoPath Attachments to Folder	105
10.11 Find List Item at Url	105
10.12 Find List Item by Caml.....	106
10.13 Get Field Value at Url	107
10.14 Get List Item Attachment Content.....	107
10.15 Get List Item Attachments	108
10.16 Get List Item Url	108
10.17 Get List Url.....	109
10.18 Is Current Item Checked Out (HarePoint)	110
10.19 Is Specified Item Checked Out (HarePoint).....	110
10.20 Reject Item at Url	110
10.21 Remove All Attachments.....	111

10.22 Remove InfoPath Attachments	111
10.23 Update List Item at Url	111
10.24 Upload Files to List Item Attachments	112
11. Network activities	113
11.1 Create Directory on FTP Server	113
11.2 Fetch RSS Feed to Html	113
11.3 Fetch RSS Item	114
11.4 Send HTTP GET Request	114
11.5 Send HTTP POST Request	115
11.6 Send Twitter Message	116
11.7 Send XMPP Instant Message	117
11.8 Set Encrypted Passwords	118
11.9 Set FTP Server Options	118
11.10 Set HTTP Request Options	119
11.11 Upload Document to FTP Server	120
11.12 Upload List Item Attachments to FTP Server	120
12. Office activities	121
12.1 Find Text in Word Document	121
12.2 Get Cell Value	121
12.3 Get Office Document Property	122
12.4 Read Word Document	123
12.5 Read Word Document To Text	123
12.6 Replace Text in Word Document	123
12.7 Set Cell Value	124
12.8 Set Office Document Property	124
12.9 Update Word Document	125
13. Permissions activities	127
13.1 Change Permissions Inheritance at Url	127
13.2 Change Permissions Inheritance for List	127
13.3 Change Permissions Inheritance for List Item	128
13.4 Delete All Permissions at Url	128

13.5 Delete All Permissions from List	128
13.6 Delete All Permissions from List Item	129
13.7 Delete All User Permissions at Url	129
13.8 Delete All User Permissions from List	129
13.9 Delete All User Permissions from List Item.....	130
13.10 Delete Permissions at Url.....	130
13.11 Delete Permissions from List	131
13.12 Delete Permissions from List Item	132
13.13 Grant Permissions at Url	133
13.14 Grant Permissions on List.....	134
13.15 Grant Permissions on List Item	135
14. Printing activities.....	136
14.1 Print Document.....	136
14.2 Set Printing Options	137
15. Strings activities	137
15.1 Capture All Occurrences of Text	137
15.2 Capture Text.....	138
15.3 Convert HTML To Text	139
15.4 Convert Number To Text.....	139
15.5 Converts Base64 to Text	140
15.6 Converts Text to Base64	141
15.7 Decode Web Text.....	142
15.8 Encode Web Text	143
15.9 Extract Substring	144
15.10 Format Field of Current Item	145
15.11 Get Random Number	146
15.12 Get Random String.....	146
15.13 Is Value Matches Regular Expression (HarePoint).....	147
15.14 Join Text	147
15.15 Replace Text	148
15.16 Set Regular Expression Options	149

15.17 Split Text.....	149
15.18 Trim Text	150
16. Workflow activities	150
16.1 Get Workflow Association Data	150
16.2 Get Workflow Status	151
16.3 Start Workflow	151
16.4 Terminate Workflow	153
16.5 Wait Workflow To Complete	153
16.6 Wait Workflow To Complete By Name	154
Appendix A. List of activities from Free Set	156

I. Introduction



HarePoint Workflow Extensions is a software product that substantially expands the capacity to create workflows for SharePoint and in the most of cases allows avoiding custom development and coding when creating them.

The product offers a set of ready activities with a fundamentally new functionality to workflows - managing document libraries and lists, reading and writing Microsoft Office documents, document conversion to PDF and other formats, image conversion, processing and resizing, SharePoint administration, working with Active Directory, working with SQL databases and XML, executing just-in-time compiled C# or VB.NET code, operating through such networking protocols as HTTP, SMTP, POP3, FTP, RSS, XMPP and Twitter, improved activities for emailing messages and a large set of activities that facilitate work with string arrays, strings and regular expressions, date and time.

At the same time, the product is fully supported by all Microsoft workflow development tools and there is no need to migrate and adapt current workflows to a new development environment, master and deploy a new development environment.

Three key features of HarePoint Workflow Extensions are as follows:

- ✓ over 160 new activities in addition to 30 built-in SharePoint activities;
- ✓ over 30 activities of Free Set, which you can use without any fee or limitation;
- ✓ full support in SharePoint Designer, Microsoft Visio, Microsoft Visual Studio;
- ✓ easy deployment, integration and minimum training costs.

We are confident that within an hour after deployment you will learn and be able to use new features of the product to the in full.

II. Deployment

1. System requirements

HarePoint Workflow Extensions is designed to operate in SharePoint Server 2010 and SharePoint Foundation 2010 (hereinafter referred to as SharePoint 2010) environments, and it is not compatible with SharePoint Server 2007 and Windows SharePoint Services 3.0.

The product has no special requirements to operating system and its components, RAM memory space and CPU type. The product may operate at any configurations where SharePoint 2010 is deployed. The product requires 10 Mb of disk space on each SharePoint 2010 server.

HarePoint Workflow Extensions includes a number of supporting Client Applications and components (hereinafter referred to as the Client Applications) that can be installed on the workstations used for workflow development (hereinafter referred to as the Workstations).

Minimum workstation system requirements:

- Microsoft Windows XP Service Pack 2 (x86, x64).
- Microsoft .NET Framework v2.0.
- 10 MB of free disk space.

In addition to the minimum requirements the Client Applications have a number of special requirements:

Client Program	Description	Special requirements
Twitter Authorization Tool	Utility to generate authorization token for “Send Twitter Message” activity (see description of this activity in Section IV).	Microsoft Internet Explorer 6.0 or higher
HarePoint Visio Add-In	Library of shapes and support module for HarePoint activities in Visio (see chapter 4 of Section III).	Microsoft Visio Premium x86, x64. See also “Introduction to designing and customizing workflows”: http://office.microsoft.com/en-us/sharepoint-designer-help/introduction-to-designing-and-customizing-workflows-HA101859249.aspx?CTT=3
Library for Visual Studio	HarePoint library for Visual Studio and toolbar utility (see chapter 5 of Section III).	Microsoft Visual Studio 2010 with installed project template “Sequential Workflow” for SharePoint 2010. See also “Requirements for Developing SharePoint Solutions”: http://msdn.microsoft.com/en-us/library/ee231582.aspx

2. Licensing and trial limitations

HarePoint Workflow Extensions is licensed for a number of SharePoint 2010 servers where “Microsoft SharePoint Foundation Workflow Timer Service” is installed. The number of licenses per product should correspond to a number of servers. A list of services installed on the server can be viewed by opening System Settings tab and clicking the link “Manage servers in this farm” on Central Administration site.

A number of Workstations with installed Client Applications for workflow development aren’t accounted by product licensing.

You may use the product on the basis of one of 4 licenses:

1. *Trial License*. Provided automatically for a 30-day period to use the product in organization’s environment without any limitation by number of servers and product functionality.
2. *Free License*. This license provides free use of the product in the organization’s environment without limitation by number of servers, usage time, but with limited set of activities: a set of free activities only (see Appendix A).
3. *Limited License*. This license is purchased for a number of SharePoint 2010 servers and allows using product in the organization’s environment without limitation by usage time, however with limited set of available activities: only the set of free activities and any five paid activities at your choice.
4. *Full License*. This license is purchased for a number of SharePoint 2010 servers and allows using the product in the organization’s environment without limitation by usage time and any limitation of available activities.

After the first installation of the product on SharePoint 2010 it will automatically start operating under Trial License. At the same time all the activities supplied with the products are available.

After expiration of the Trial License, the product will continue to operate under Free License. The set of available activities will be limited to the free set (see Appendix A). Execution of workflows created by using paid activities will be terminated with error and record “This workflow action requires a license” in the history list.

After purchasing a Limited License and entering registration keys on the administration page HarePoint Workflow Extensions of the Central Administration site it will be possible to choose five activities from

The screenshot displays the administration interface for HarePoint Workflow Extensions. At the top, it states "1 server was found where the product is installed." Below this, a text box contains the license key "WLQH8-". A message follows: "You have entered the limited license key(s). You need to specify five activities you plan to use and redeploy the solution in order to changes take effect." Below the message are three dropdown menus for selecting activities: "Change Permissions Inheritance for List Item", "Create Group in Active Directory", and "Get User's Attribute from Active Directory". A fourth dropdown menu, "Is Specified Item Checked Out", is expanded, showing a list of activities under two categories: "Conditions" and "Active Directory". The "Active Directory" category is currently selected and highlighted in blue.

1 server was found where the product is installed.

WLQH8-

You have entered the limited license key(s). You need to specify five activities you plan to use and redeploy the solution in order to changes take effect.

Change Permissions Inheritance for List Item

Create Group in Active Directory

Get User's Attribute from Active Directory

Is Specified Item Checked Out

Conditions

- Is Current Item Checked Out
- Is Role Assigned to User
- Is Specified Item Checked Out
- Is User a Member of a SharePoint Group

Active Directory

- Add Member to Active Directory Group
- Change Active Directory Account Settings
- Create Group in Active Directory
- Create User in Active Directory
- Delete Group from Active Directory
- Delete User from Active Directory
- Get Active Directory Groups Where User is Member
- Get E-Mails of Active Directory Group Members
- Get User's Attribute from Active Directory
- Get User's Manager from Active Directory
- Get User's Multivalued Attribute from Active Directory
- Remove Member from Active Directory Group
- Set Active Directory Authentication (impersonate)
- Set User's Attribute in Active Directory

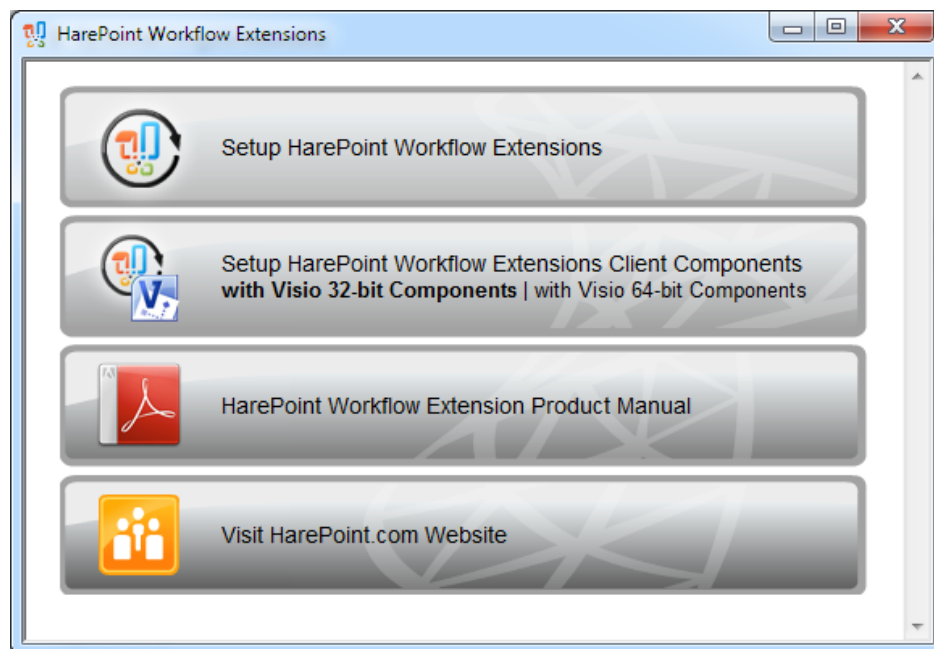
the paid list (in addition to activities from the free set). This selection can be changed later. The rest of paid activities will be inaccessible for creation of workflows, while execution of workflows created with their help will be terminated with error and record “This workflow action requires a license” in the history list.

After purchasing Full License and entering Registration Keys on the administration page HarePoint Workflow Extensions of the Central Administration site, all workflow activities will become available. All previously created workflows using the paid activities will be executed without errors.

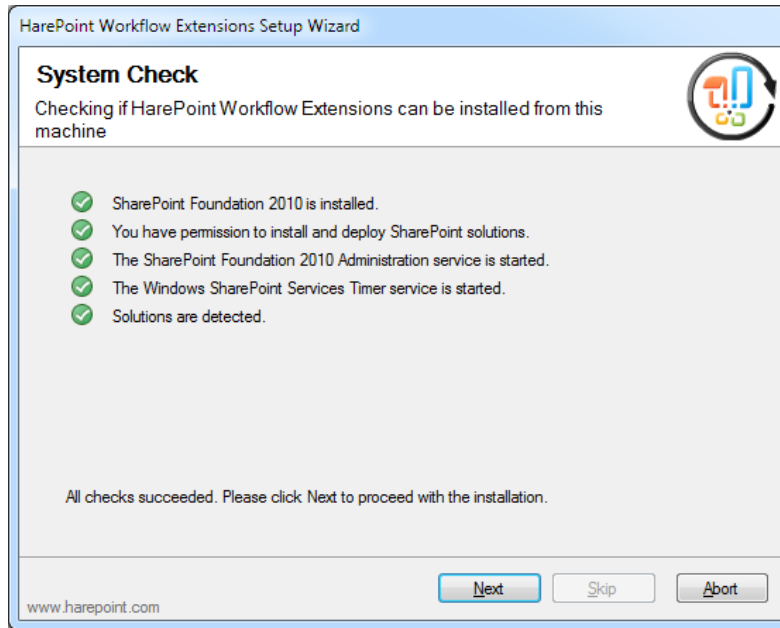
3. Installing HarePoint Workflow Extensions

To install the product you’ll need administrator rights for SharePoint 2010 farm.

Please access any server of the SharePoint 2010 farm, unpack the product archive into a folder on the local disk. Run SETUP.EXE in the selected folder. The following window will be opened:



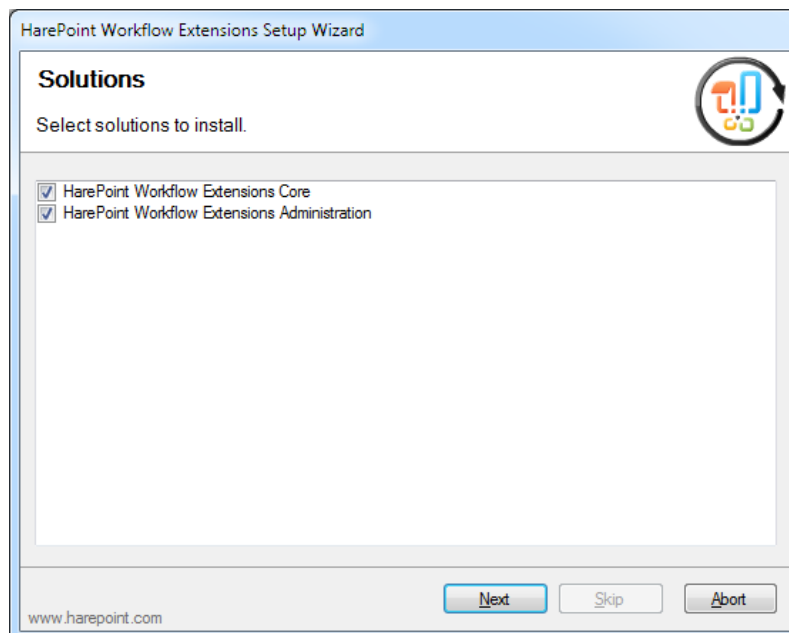
Select item “Setup HarePoint Workflow Extensions”. Installation Wizard will be started. Press Next button on the wizard’s welcome window and the wizard will check if the product can be installed:



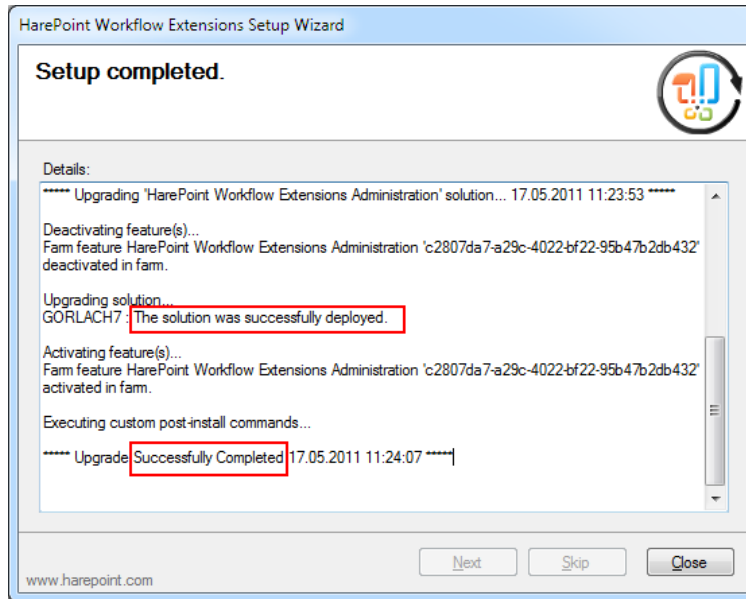
Should any problems be detected, please rectify them and restart the installation wizard.

At the next step of the Wizard, you will be offered the solutions to install. In general case for successful product installation you'll need to install both solutions offered:

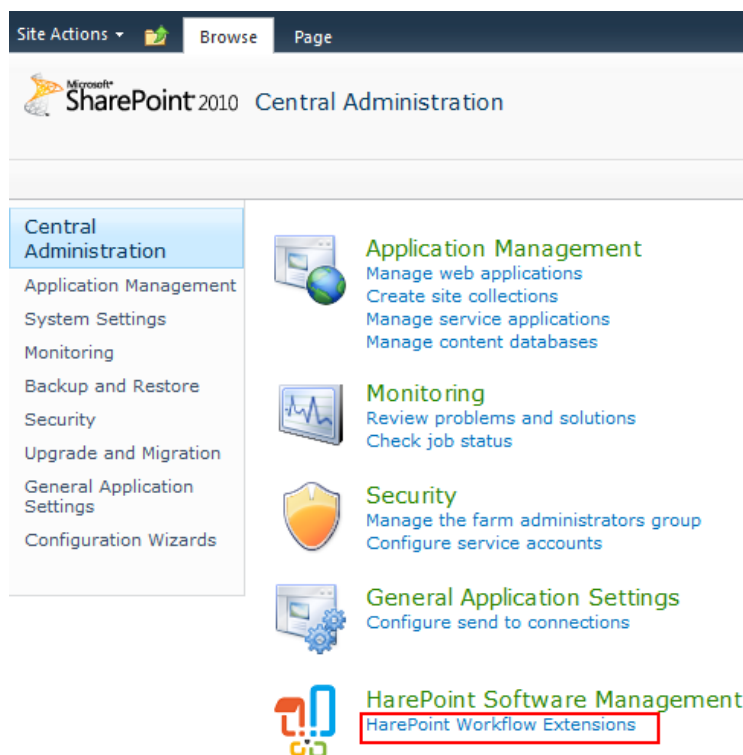
- HarePoint Workflow Extensions Core – solution containing activity library.
- HarePoint Workflow Extensions Administration – solution for managing the product on the Central Administration site.



Follow the instructions of the Wizard and complete installation of the solutions selected:

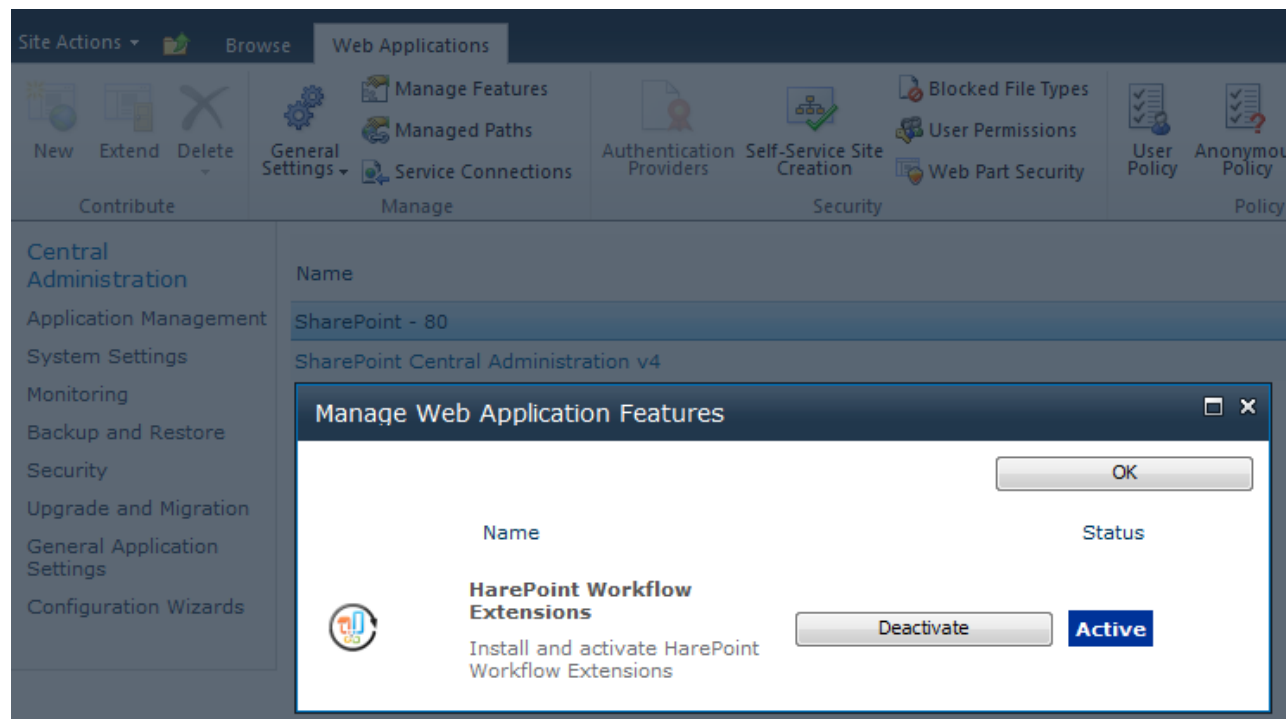


Open Central Administration site. Open the “HarePoint Workflow Extensions” link in HarePoint Software Management section:



If you have product license, please enter the license code in “Licenses” section. This section also describes the type of your current license and in case of Trial License, the number of days left to its expiration.

Please be advised that deactivation of “HarePoint Workflow Extensions” in the “Manage Features” section of the web application will not result in deactivation of product features:

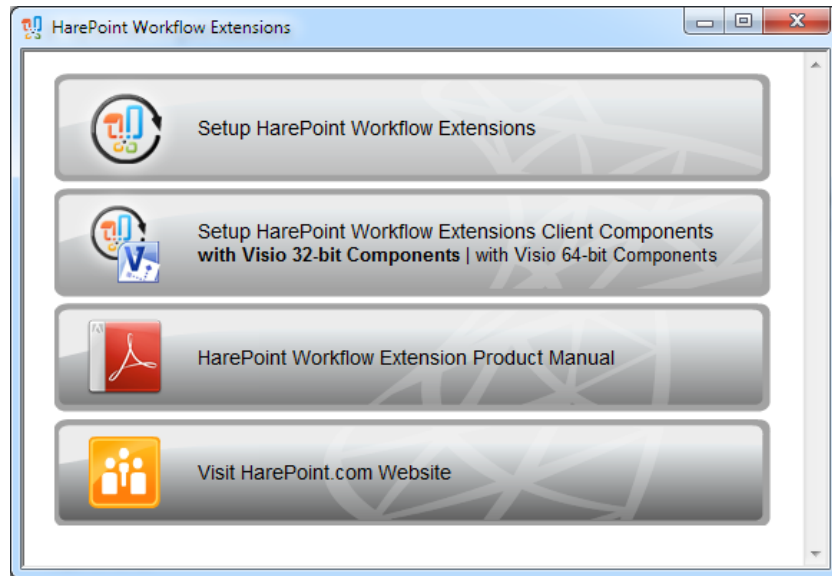


Manual on forced deactivation of full or partial functionality of the products for the site is given in chapter 6, Section III of this Manual.

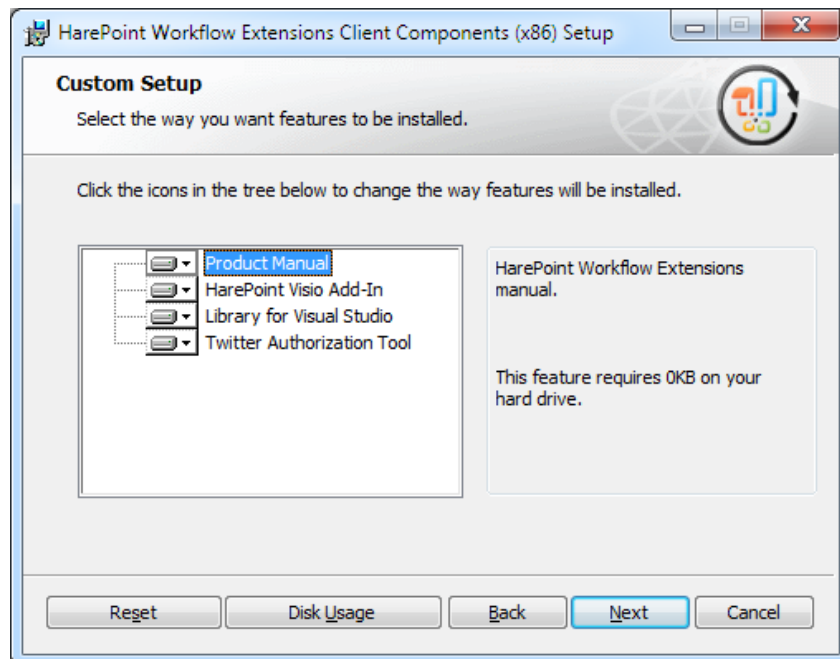
4. Installing Client Applications

Installation of Client Applications is not mandatory for workflow development using the product activities (see chapter 1 of this Section and chapter 1 of Section III).

Enter the Workstation; unpack the product archive into a folder on the local disk. Run SETUP.EXE in the selected folder. The following window will be opened:



Select the “Setup HarePoint Workflow Extensions Client Components” item. Installation Wizard will be launched. Follow instructions of the Wizard to select the required Client Applications:



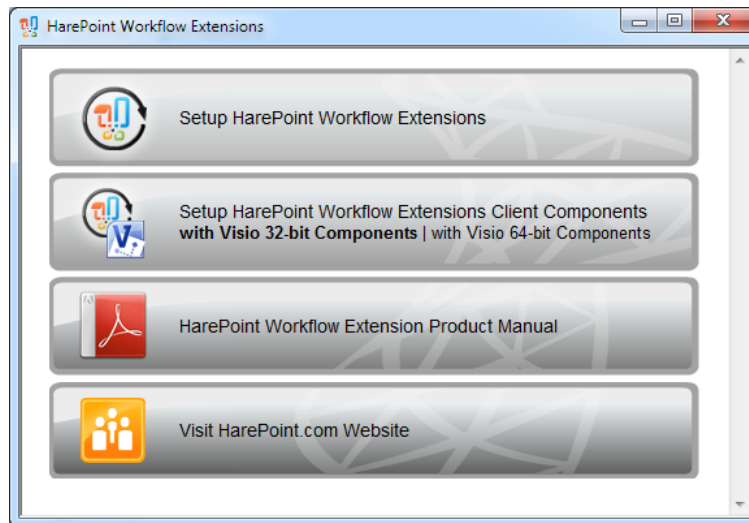
Then follow instructions of the Wizard and complete the installation.

The operation with Client Application is described in Section III of this manual.

5. How to repair, upgrade or remove product

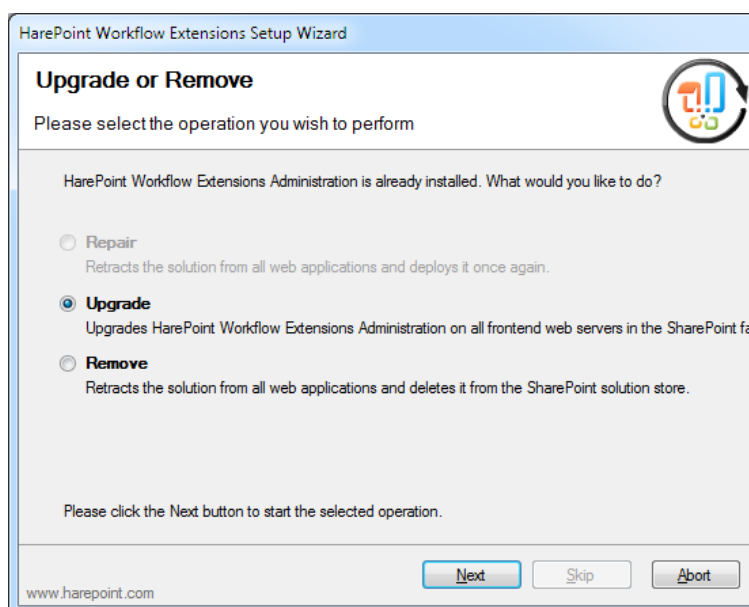
To repair, upgrade or remove the product you'll need administrator rights of SharePoint 2010 farm.

Please enter any server of the SharePoint 2010 farm, unpack the product archive into a folder on the local disk. Run SETUP.EXE in the selected folder. The following window will be opened:

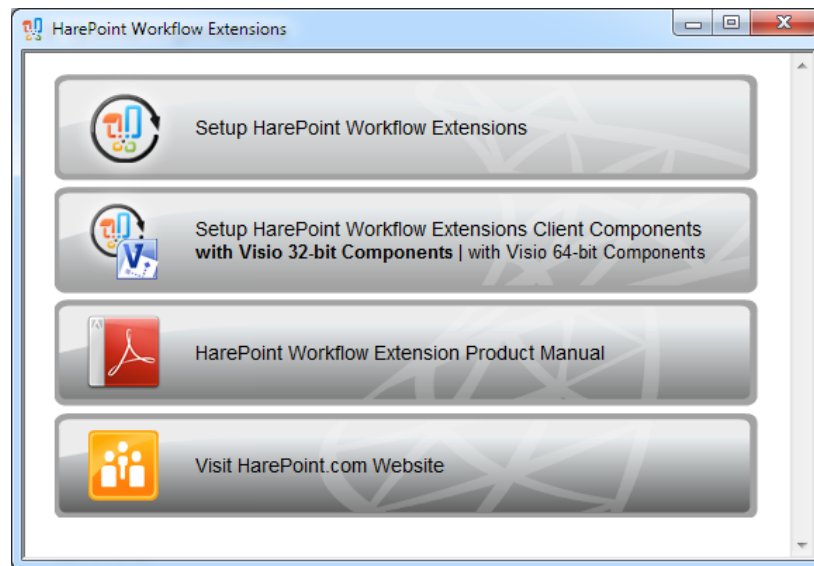


Please select "Setup HarePoint Workflow Extensions" item. Installation Wizard will be launched. Following instructions of the Wizard, pass system check and solution selection steps similar to those during the product installation (see chapter 3).

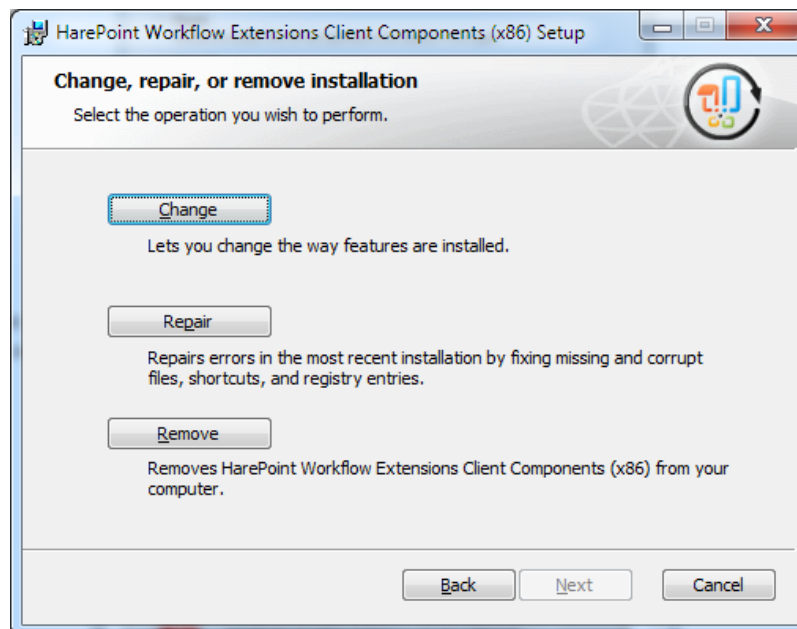
Upon completing these steps the Wizard will ask to repair, upgrade or remove the product components. Select the required type of operation and follow the further instructions of the Wizard.



To repair, upgrade or remove a Client Application, please enter a Workstation, unpack the product archive into a folder on the local disk. Run SETUP.EXE in the selected folder. The following window will be opened:



Select "Setup HarePoint Workflow Extensions Client Components" item. Installation Wizard will be launched. Following instructions of the Wizard, select the required Client Applications and type of operation:



Follow instructions of the Wizard and complete installation.

III. Product Overview

1. Getting started

Microsoft offers three tools that create workflows for SharePoint. Two of them, SharePoint Designer and Microsoft Visio, are intended for no-code workflow automation. A third one, Microsoft Visual Studio, includes more functionality but requires quite complex programming and deep knowledge of SharePoint architecture.

HarePoint Workflow Extensions can be successfully used to develop workflows in Visual Studio, but this product's main goal is to avoid programming as this requires not only deep knowledge but also ten times more time for workflow development, debugging and support.

If you are new to workflow development in SharePoint Designer and Microsoft Visio, before you further read this material we recommend you the article “Introduction to designing and customizing workflows” at <http://office.microsoft.com/en-us/sharepoint-designer-help/introduction-to-designing-and-customizing-workflows-HA101859249.aspx?CTT=3>

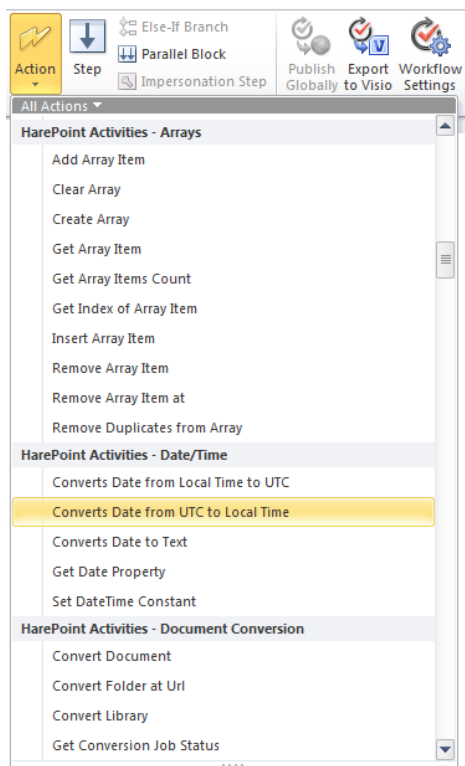
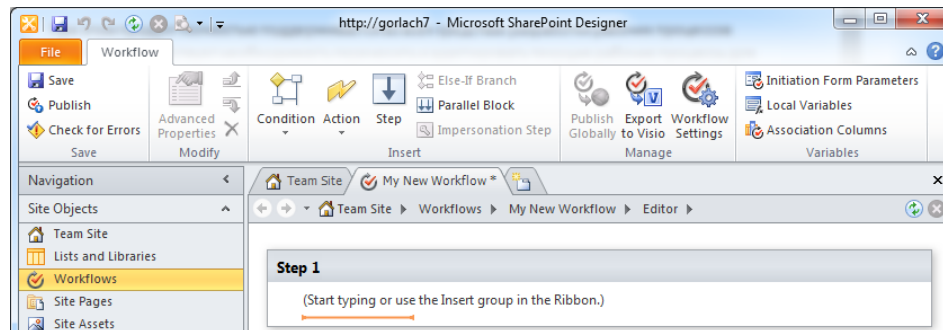
Please note that the product is made up of five components:

- The library of the activities. Installed on servers of SharePoint farm.
- The administration module. Installed on the Central Administration site of SharePoint farm.
- A plug-in for Microsoft Visio. Installed on the work station where it requires use of HarePoint Workflow Extensions activities in Microsoft Visio.
- The library for Visual Studio. Installed on the work station where it requires use of HarePoint Workflow Extensions activities in Visual Studio.
- Utility for Twitter. Installed on the work station, required for generation of an authorization token.

The product files do not have to be installed on the work station to develop workflows only in SharePoint Designer without the use of Microsoft Visio or Visual Studio. The exception is the utility for Twitter required to generate an authorization token for the “Send Twitter Message” activity (see description in Section IV).

2. Creating workflows in SharePoint Designer

Within SharePoint Designer, get connected to the site where the product is installed, create a new workflow or open the existing workflow:



Click “Action” on the control panel and scroll the “All Actions” list that opens. If you see categories of activities the titles of which start with “HarePoint Activities” – congratulations, as everything has been successfully installed and you can start creating workflows.

Please note that apart from activities the product includes a few conditions. They are available in the drop-down list of Conditions and have the names of this kind:

Is Item Field Matched to Regular Expression (HarePoint)

Working with HarePoint activities and conditions is not different from working with in-built SharePoint activities and conditions. A complete user manual for activities and conditions is provided in Section IV of this user manual.

If you are new to workflow development in SharePoint Designer, please read the article referred to in “Getting Started”.

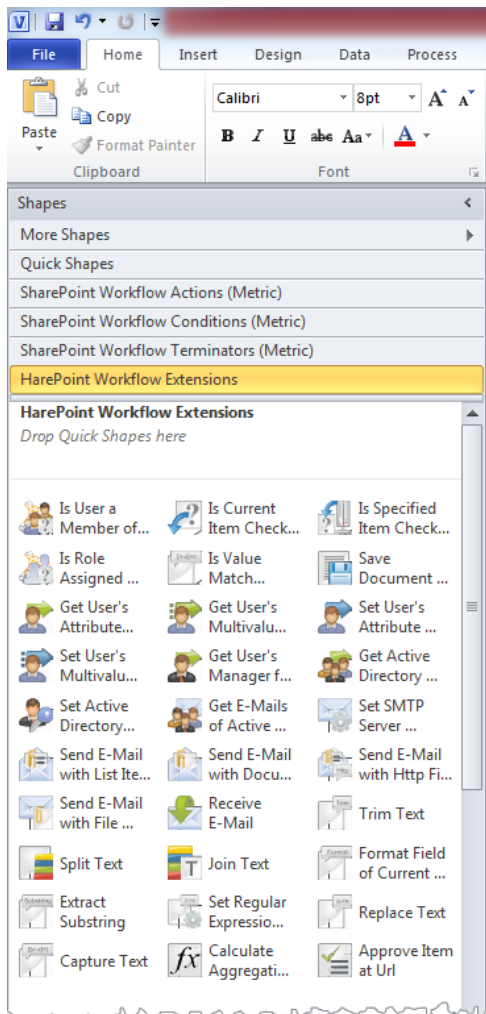
Please note that a number of activities and conditions are context-sensitive and sometimes you may not find them in the menu. For example, the “Create Document from File” activity is available only when creating a workflow for the Document Library and it won’t be listed in the menu when a workflow for the Calendar or the Task List is created.

3. Designing workflows in Microsoft Visio

Microsoft Visio allows easily designing large and complex workflows. You cannot set parameters for activities in Visio. However, it visually represents the workflow algorithm.

If you are new to designing workflows in Visio, please read the following articles:

- Introduction to designing and customizing workflows: <http://office.microsoft.com/en-us/sharepoint-designer-help/introduction-to-designing-and-customizing-workflows-HA101859249.aspx?CTT=3>
- Create, import, and export SharePoint workflows in Visio: <http://office.microsoft.com/en-us/visio-help/create-import-and-export-sharepoint-workflows-in-visio-HA101888007.aspx>



To design workflows in Visio, Microsoft Visio Premium is required. Besides, HarePoint Visio Plug-In should be installed on the computer for full support of HarePoint Workflow Extensions activities (see Section II of this manual).

Select File -> Options in Microsoft Visio and go to “Add-Ins” in the “Visio Options” window that opens. There should be “HarePoint Visio Add-In” in the “Active Application Add-Ins” section.

When a new document is being created from a “Microsoft SharePoint Workflow” template or a SharePoint workflow is being imported, “HarePoint Workflow Extensions” tab will appear in Shapes.

Using HarePoint activities and conditions in Microsoft Visio is not different from using in-built SharePoint Workflow activities and conditions.

Please note that HarePoint Visio Plug-In is used for import and export of a workflow file. The plug-in is not used to save files of other types. When there is some failure during import or export of workflow files, please disable the plug-in in the “Visio Options” menu.

Please always back up workflow files.

4. Creating workflows in Visual Studio

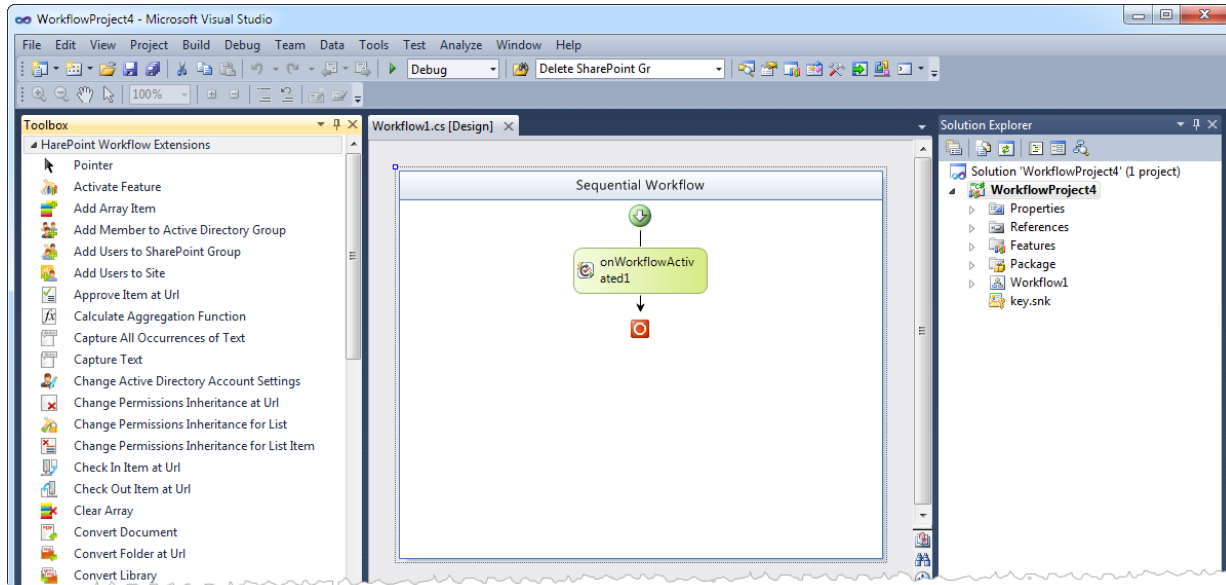
Creating workflows in Visual Studio is rather complex and required deep knowledge of SharePoint architecture. Whenever possible, it's better to confine with "Execute Custom Code" activity or with development of a custom activity rather than create a workflow in Visual Studio.

If you are new to developing workflows in Visual Studio, please first read the following articles:

- Requirements for Developing SharePoint Solutions: <http://msdn.microsoft.com/en-us/library/ee231582.aspx>
- Creating SharePoint 2010 Sequential Workflows in Visual Studio 2010: <http://msdn.microsoft.com/en-us/library/gg265727.aspx>

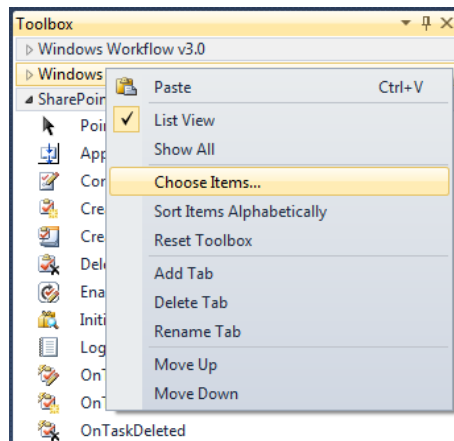
When developing a workflow in Visual Studio, you can use HarePoint activities and in-built SharePoint activities alike. For this, all components should be installed on the computer as required by the article "Requirements for Developing SharePoint Solutions" as well as HarePoint libraries for Visual Studio (see Section II of this manual).

Select "Add HarePoint toolbox to Visual Studio 2010" in Start -> Programs. A console application will be launched to add the "HarePoint Workflow Extensions" toolbox for all languages and all projects created on the basis of "Sequential Workflow" and "State Machine Workflow" templates for SharePoint 2010.



The toolbox can be removed through the context menu, "Delete Tab" in the toolbox, or by running "Remove HarePoint toolbox from Visual Studio 2010" in Start -> Programs.

A toolbox can also be added manually. For this, select “View \ Toolbox” in the project based on the “Sequential Workflow” template created as described in the article referred to above. Click the right button of the mouse in the work area in the Toolbox window that opens and select “Choose Items...”.



Click “Browse...” in the “.NET Framework Components” tab of the window that opens to select the *HarePoint.Activities.dll* file in the directory selected for installation of the HarePoint library for Visual Studio. Then HarePoint activities appear in the list.

After you press OK and close the window, a link to HarePoint.Activities will appear in the References section of the project, the “General” tab with HarePoint Workflow Extensions activities will appear in the Toolbox window.

After components are added manually, they can be renamed manually so that their names will be more readable and look like

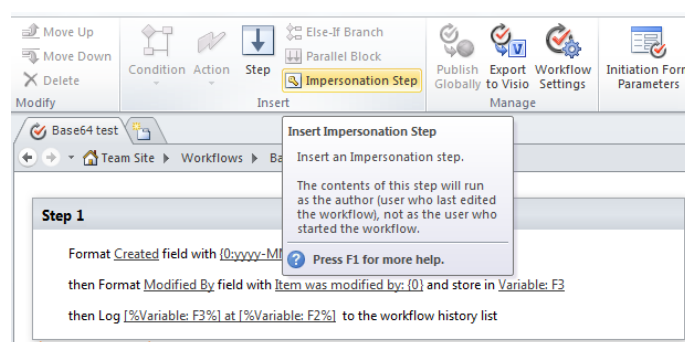
in SharePoint Designer and sorted by name by selecting the “Sort Items Alphabetically” item in the context menu. Please note that when components are added by a command “Add HarePoint toolbox to Visual Studio 2010”, they will be automatically renamed and sorted.

Using HarePoint activities in Visual Studio is not different from using in-built SharePoint activities.

5. Security issues

HarePoint activities are performed in the context of the user who initiated a workflow. If we have created a workflow that automatically runs when a new document is added to a library, this workflow will always run in the context of the user who added the document.

In this case, a number of activities such as “Create User in Active Directory” or “Add User to SharePoint Group” require adequate rights and permissions. The user who has the permission for adding documents to a library may not have rights and permissions for creating a user in Active Directory or adding a user in the SharePoint group and the workflow will be run with errors.



Such mistakes can be avoided in two ways.

A general method is to move such activities to the Impersonation Step, which is run in the context of the workflow author (or last user who edited the workflow).

The second method is available only for activities working with the Active Directory.

It involves adding the “Set Active Directory Authentication (impersonate)” activity which saves the login and password for access to the Active Directory in the current workflow step.

Step 1

Set Active Directory authentication (impersonate) using user name GORLACH7\gorlach and password Mk29GfS3eh .

then Add Current Item:Created By to members of GORLACH7\docmanagers group in Active Directory.

All HarePoint activities working with the Active Directory check if the current workflow step has the above “Set Active Directory Authentication (impersonate)” activity and, if it does, use access parameters set there. Otherwise, they work with the Active Directory in the context of the user who initiated workflow launch or the workflow author if they were moved to the Impersonation Step.

Apart from “Set Active Directory Authentication (impersonate)”, the HarePoint library has a number of activities which also require a password. These are activities for FTP, receiving messages through POP3 and sending email through SMTP. In all cases, the password is stored as open text in the workflow XOML file.

This means that under certain conditions the password can be obtained by third parties through SharePoint Designer or even a browser through a link of this kind:

<http://sharepoint/Workflows/workflowname/workflowname.xoml>

Setting up security for SharePoint is outside the scope of this article. If you have questions, please see the official user manual at <http://technet.microsoft.com/en-us/library/ee428324.aspx>

Please note that you can disable some HarePoint activities. See chapter 6 of this Section.

6. How to disable certain HarePoint activities

In some cases you may need to disable certain HarePoint activities. What you need to do is open the *web.config* file on the relevant server and add a string as shown in the example below (italicized) to the *authorizedTypes* section:

```
<System.Workflow.ComponentModel.WorkflowCompiler>
<authorizedTypes>
....
<authorizedType Assembly="HarePoint.Activities, Version=2.0.0.0, Culture=neutral,
    PublicKeyToken=73210640c0bcfb88"
    Namespace="HarePoint.Activities" TypeName="*" Authorized="True" />
<authorizedType Assembly="HarePoint.Activities, Version=2.0.0.0, Culture=neutral, PublicKeyToken=73210640c0bcfb88"
    Namespace="HarePoint.Activities" TypeName="Actions.FormatField" Authorized="False" />
</authorizedTypes>
```

```
</System.Workflow.ComponentModel.WorkflowCompiler>
```

After IIS is restarted, all workflows with the "Format Field of Current Item" activity (HarePoint.Activities.Actions.FormatField) will terminate with an error - "Workflow was canceled by System Account". In response to an attempt to add a new workflow with this activity, SharePoint Designer will show an error message *"(0, 0) Type HarePoint.Activities.Actions.FormatField, HarePoint.Activities, Version=2.0.0.0, Culture=neutral, PublicKeyToken=73210640c0bcfb88 is not marked as authorized in the application configuration file.)"*.

To remove an activity from the list of those available for SharePoint Designer, open the file "HarePointActivities.Actions" in the templates directory (such as: "C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\TEMPLATE\1033\Workflow\") and remove the following block from it:

```
<Action Name="Format Field of Current Item"
  ClassName="HarePoint.Activities.Actions.FormatField"
  Assembly="HarePoint.Activities, Version=2.0.0.0, Culture=neutral, PublicKeyToken=73210640c0bcfb88"
  AppliesTo="all"
  Category="HarePoint Activities - Strings">
  <RuleDesigner Sentence="Format %1 field with %2 and store in %3">
    <FieldBind Field="Input" DesignerType="FieldNames" Text="this" Id="1"/>
    <FieldBind Field="Format" DesignerType="TextArea" Text="this format" Id="2"/>
    <FieldBind Field="OutputText" DesignerType="ParameterNames" Text="FormattedText" Id="3"/>
  </RuleDesigner>
  <Parameters>
    <Parameter Name="__Context"
      Type="Microsoft.Sharepoint.WorkflowActions.WorkflowContext, Microsoft.SharePoint.WorkflowActions"
      Direction="In"/>
    <Parameter Name="__ListId" Type="System.String, mscorlib" Direction="In" />
    <Parameter Name="__ListItem" Type="System.Int32, mscorlib" Direction="In" />
    <Parameter Name="Input" Type="System.String, mscorlib" Direction="In"
      Description="Field name of current item"/>
    <Parameter Name="Format" Type="System.String, mscorlib" Direction="In" Description="Format string"/>
    <Parameter Name="OutputText" Type="System.String, mscorlib" Direction="Out" Description="Output text"/>
  </Parameters>
</Action>
```

After IIS is restarted, there'll be no activity in the list of activities available in SharePoint Designer for all sites on the server.

IV. HarePoint Activities and Conditions

This section contains manual on activities and conditions of the product. How the action is initially displayed in a workflow step (appears in SharePoint Designer upon adding to the workflow) is given for each activity or condition after its name. Some descriptions contain a paragraph of examples with sample workflows and how they are displayed in SharePoint Designer unless otherwise specified.

1. Active Directory activities

1.1 Add Member to Active Directory Group

Add current user to members of specified group in Active Directory.

This workflow action is used to add user or group to specified Active Directory security group.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

current user - Logon name of the user or group: DOMAIN\user or user@domain.com. Leave the field empty to use current user.

specified - Logon name of the group: DOMAIN\group or group@domain.com.

1.2 Change Active Directory Account Settings

Change logon name account settings: enable, change password at next logon: keep existing, password never expires: keep existing.

This workflow action is used to enable or disable user account and change its settings.

See Also: "Create User in Active Directory" activity.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

logon name - Logon name of the user: DOMAIN\user or user@domain.com.

enable/disable - Enable or disable user account.

change password - User must change password at next logon: yes/no or keep existing settings

password never expires - Password never expires: yes/no or keep existing settings.

1.3 Create Group in Active Directory

Create new group in this container with specified group name.

This workflow action will create new group in specified Active Directory container.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

this - Distinguished name of the Active Directory container: OU=Test,DC=domain,DC=com. Use Active Directory Explorer to get correct path: <http://technet.microsoft.com/en-us/sysinternals/bb963907>

specified - Logon name of the group: DOMAIN\group or group@domain.com.

1.4 Create User in Active Directory

Create new user in this container with specified logon name, this first name, this last name, this password.

This workflow action will create new user account in specified Active Directory container. Created account will be enabled, you can disable account and change its settings with "Change Active Directory Account Settings" activity.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

this container - Distinguished name of the Active Directory container, like OU=Test,DC=domain,DC=com.

specified - User's logon name. Domain name in the user's logon name is ignored (correct syntax is DOMAIN\username or username@domain).

this first name - User's first name.

this last name - User's last name.

this password - User's password. Password may be blank. Use "Change Active Directory Account Settings" activity to ask user to change password at first logon.

1.5 Delete Group from Active Directory

Delete group with this group name from Active Directory.

This workflow action will delete specified security group from Active Directory.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

this - Logon name of the group: DOMAIN\group or group@domain.com.

1.6 Delete User from Active Directory

Delete user with this logon name from Active Directory.

This workflow action will delete specified user from Active Directory.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

this - Logon name of the user: DOMAIN\user or user@domain.com.

1.7 Find User in Active Directory

Find current user in Active Directory and store logon name in Variable: variable

This workflow action is used to find a SharePoint user in Active Directory and store his logon name in the output variable. Action returns an empty string if the user is not found in Active Directory.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow, or workflow author context in the case of impersonation step.

Parameters:

current user - Name of SharePoint user.

Variable: variable - Output string variable with a logon name like DOMAIN\username. Empty string if user is not found in Active Directory.

1.8 Get Active Directory Groups Where User is Member

Get list of the Active Directory groups where user current user is member. Store list of these groups in Variable: GroupsArray, success of the searching in Variable: UserFound.

This workflow action will return the list of groups where specified user is a member.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

current user - The name of Active Directory user for reading names of his groups (like DOMAIN\Account). Leave the field empty to use current user.

Variable: GroupsArray - Output array variable with the list of groups. Use "Join Text" or array activities to convert this array to string.

Variable: UserFound - Output boolean variable, *true* indicates that user was found in Active Directory.

Examples:

Get list of the Active Directory groups where user current user is member. Store list of these groups in Variable: GroupsArray, success of searching in Variable: UserFound.

then Get the number of items in the Variable: GroupsArray and store in Variable: groupsCount.

then Log [%Workflow Context: Current User%] is the member of [%Variable: groupsCount%] groups to the workflow history list

1.9 Get E-Mails of Active Directory Group Members

Get user e-mails from specified group. Expand nested groups: No. Store result in Variable: EmailsArray.

This workflow action will return Active Directory group member's email addresses to array variable.

Use "Join Text" or array activities to get string values.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

specified group - Name of the group from which we want to read member e-mails (like: DOMAIN\group).

expand nested groups - Include or not members of nested groups.

Variable: EmailsArray - Output array variable with the list of the e-mails of Active Directory group members.

1.10 Get User's Attribute from Active Directory

Get this attribute of current user from Active Directory. Store attribute value in Variable: variable1, success of searching in Variable: variable2.

This workflow action will return a value of Active Directory user specified attribute.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

See Also: "Get User's Multivalued Attribute from Active Directory" action.

Parameters:

this attribute - Name of the attribute.

current user - Name of the user from which we want to read attribute value (like DOMAIN\Account). Leave the field empty to use current user.

Variable: variable1 - Output variable for the found attribute value.

Variable: variable2 - Output boolean variable, *true* indicates that user (not attribute!) was found in Active Directory.

1.11 Get User's Manager from Active Directory

Get manager of current user from Active Directory. Store manager's display name in Variable: variable1, success of the searching in Variable: variable2.

This workflow action is used to get a manager's display name for specified Active Directory user.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

current user - The name of Active Directory user for reading his manager's name (like DOMAIN\Account). Leave the field empty to use current user.

Variable: variable1 - Output variable for resultant manager display name.

Variable: variable2 - Output boolean variable, *true* indicates that user (not manager or display name) was found in Active Directory.

1.12 Get User's Multivalued Attribute from Active Directory

Get multi-valued attribute this attribute of current user from Active Directory. Store attribute value in Variable: ArrayVariable, success of the searching in Variable: variable.

This workflow action is used to get multi-valued attribute of specified Active Directory user.

Note: This action returns array variable. You can use "Get Array Items Count", "Get Array Item" or "Join Text" actions to get items from output array.

See Also: "Get User's Attribute from Active Directory" action.

Parameters:

this attribute - Name of attribute.

current user - User logon name (like DOMAIN\Account) or current user by default.

Variable: ArrayVariable - Output array variable with attribute values.

Variable: variable - Output boolean variable, indicates that user was found in Active Directory.

1.13 Is User a Member of an Active Directory Group (HarePoint)

If current user is a member of Active Directory group this group (use this user and this password to connect Active Directory)

This condition returns true if specified user is the member of specified Active Directory group.

You can specify optional credentials for Active Directory access. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Note: The password will be stored as a plain text in the workflow XOML file and may be accessed by the URL like *http://sharepoint/Workflows/workflowname/workflowname.xml* or by SharePoint Designer.

Parameters:

current user - User logon name, like DOMAIN\user.

this group - Group logon name, like DOMAIN\group.

this user - User name to be used for Active Directory access (like DOMAIN\Account). Optional.

this password - Password to be used for Active Directory access. Optional.

1.14 Move User into Organizational Unit

Move Active Directory user with this logon name into specified organizational unit.

This workflow action is used to move specified AD user into specified AD organizational unit or container.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

with this logon name - Logon name of the Active Directory user (e.g. DOMAIN\username).

specified - Distinguished name or canonical name of the Organizational Unit or Active Directory container (e.g. OU=Test,DC=company,DC=local or company.local/Users)

Examples:

Move Active Directory user COMPANY\user1 into company.local/Staff organizational unit.

1.15 Remove Member from Active Directory Group

Remove current user from members of specified group in Active Directory.

This workflow action will remove user or group from an Active Directory security group.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

current user - Logon name of the user or group: DOMAIN\user or user@domain.com. Leave the field empty to use current user.

specified - Logon name of the group: DOMAIN\group or group@domain.com.

Examples:

Remove COMPANY\jsmith from finance@company.com group in Active Directory

1.16 Set Active Directory Authentication (impersonate)

Set Active Directory authentication (impersonate) using user name this user and password this password.

This workflow action is used to specify AD credentials for all AD-related HarePoint activities (like "Remove Member from Active Directory Group" or "Delete Group from Active Directory") in the current workflow step. This activity must be placed above all other AD-related activities in the step. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Note: The password will be stored as a plain text in the workflow XOML file and may be accessed by the URL like <http://sharepoint/Workflows/workflowname/workflowname.xoml> or by SharePoint Designer.

Parameters:

this user - User name that will be used for Active Directory access (like DOMAIN\Account).

this password - Password that will be used for Active Directory access (see Note).

1.17 Set User's Attribute in Active Directory

Set this attribute of current user in Active Directory to this value.

This workflow action is used to get attribute of specified user from Active Directory.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

See Also: "Set User's Multivalued Attribute in Active Directory" action.

Parameters:

this attribute - Name of attribute.

current user - User logon name (like DOMAIN\Account) or current user by default.

this value - Value for attribute.

1.18 Set User's Multivalued Attribute in Active Directory

Set multi-valued attribute this attribute of current user in Active Directory to this value.

This workflow action is used to get multi-valued attribute of specified Active Directory user.

Note: This action use array variable. Use "Create Array" or "Split Text" actions to create array variable.

See Also: "Set User's Attribute in Active Directory" action.

Parameters:

this attribute - Name of attribute.

current user - User logon name (like DOMAIN\Account) or current user by default.

this value - Array variable with values.

2. Administration activities

2.1 Activate Feature

Activate this feature in feature collection at this url. Force: yes.

This workflow action is used to activate feature on SharePoint site or site collection.

Parameters:

this feature - Feature GUID or name, as it displayed at "Site Settings \ Manage site features" or at "Site Collection Administration \ Site collection features".

this url - Site URL.

Yes/No - Yes to force activation of the feature even if there are errors; otherwise, No.

2.2 Add Users to SharePoint Group

Add specified users to specified group at this site.

This action allows you to add one or more users to SharePoint group.

Parameters:

specified users - One or more logon names of users separated by semicolon.

specified group - Group name, as it displayed at People and Groups \ All Groups.

this - Full URL of a site.

2.3 Add Users to Site

Add specified users to this site

This workflow action is used to add local or Active Directory users to SharePoint site.

Parameters:

specified users - One or more logon user names separated by semicolon. For example, DOMAIN\account or MACHINE\account.

this site - Full URL of the site.

2.4 Create SharePoint Group

Create specified group at site url and make current user the owner, give this permission level, can view: Group Member, can edit: Group Owner.

This workflow action is used to create new SharePoint group. You can add members to the created group with "Add Users to SharePoint Group" action.

Parameters:

specified group - Group name.

site url - Full URL of the site.

current user - Logon name (like DOMAIN\account) of group owner. Current user by default.

this permission level - Site permissions for the new group. You can specify one or more permissions separated by semicolon, including custom permission levels as its listed at Site Settings \ Permission Levels.

can view: **Group Member** - Who can view the membership of the group. Available values: *Group Members, Everyone*.

can edit: **Group Owner** - Who can edit the membership of the group. Available values: *Group Owner, Group Members*.

2.5 Create SharePoint Site

Create new site with specified title at this url, using this template and this description, use unique permissions No.

This action allows you to create new site anywhere within the SharePoint environment.

Parameters:

specified - Title of new site.

this url - Full URL for new site.

this template - Template for the new site. It might be name of site template, e.g. Team Site, Blank Site, Blog and etc. Either it might be ID of site template, e.g. STS#0, STS#1 and so on.

this description - New site description.

No - By default new site inherits permissions from parent Site or Site Collection or unique permissions can be created. If you select No, one set of user permissions is shared by both sites. Consequently, you cannot change user permissions on your new site unless you are an administrator of this parent site.

2.6 Create Site Collection

Create new site collection with specified title at this url, using this template and specified description. Site collection administrators: primary and secondary.

This action workflow action is used to create new site collection.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

specified title - Title of new site collection.

this url - URL of new site collection.

this template - Template for the new site. It might be name of site template, e.g. Team Site, Blank Site, Blog and etc. Either it might be ID of site template, e.g. STS#0, STS#1 and so on.

specified description - Description of new site collection.

primary - Login name of administrator for this site collection. Security groups are not supported.

secondary - Optionally specify a secondary site collection administrator. Only one user login can be provided; security groups are not supported.

2.7 Deactivate Feature

Deactivate this feature in feature collection at this url. Force: yes.

This workflow action is used to deactivate feature on SharePoint site or site collection.

Parameters:

this feature - Feature GUID or name, as it displayed at "Site Settings \ Manage site features" or at "Site Collection Administration \ Site collection features".

this url - Full site URL.

Yes/No - Yes to force deactivation of the feature even if there are errors; otherwise, No.

2.8 Delete SharePoint Group

Delete specified group from this site url.

This workflow action is used to delete SharePoint groups from specified site.

Parameters:

specified - Name of group.

site url - Full URL of a site.

2.9 Delete SharePoint Site

Delete site at this url.

This action allows you to delete site anywhere within the SharePoint environment.

Parameters:

this url - Full URL of site to delete.

2.10 Delete Site Collection

Delete site collection at this url.

This workflow action is used to delete site collection.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

this url - Full URL to site collection.

2.11 Get Site User Property

Get specified property for this user and store result in Variable: variable.

This workflow action is used to get specified property value from user's profile on the site.

For MOSS, you can get the list of all available properties in Central Administration \ Manage User Properties.

Parameters:

specified - Property display name.

this user - User, in whose profile property value is getting.

Variable: variable - Output string variable.

Examples:

Get About me property for Current Item:Created By and store result in Variable: am

then Format Created field with * Create item at {0:dd-MM-yyyy hh:mm} and store in Variable: ds

then Set Variable: ns to [%Variable: am%][%Variable: ds%]

then Set Variable: ns to About me property for Current Item: Created By

2.12 Is Role Assigned to User (HarePoint)

If role is assigned to this user

This condition is used to check is specified permission level (role) is assigned to user on the current site. This action obtains effective permission level by combining permissions granted directly to user with permissions granted to groups where user is a member.

Parameters:

role - User role (combination of permission levels).

this user - User logon name.

2.13 Is User a Member of a SharePoint Group (HarePoint)

If this user is a member of this SharePoint group

This condition allows you to check user membership in SharePoint group.

Parameters:

this user - User account name, like DOMAIN\logon.

this - SharePoint group name.

2.14 Remove User from SharePoint Group

Remove specified user from specified group at this site.

This workflow action is used to remove user from SharePoint group.

Parameters:

specified user - Logon of user to remove.

specified group - Group name, as it displayed at People and Groups \ All Groups.

this - Full URL of a site.

2.15 Remove User from Site

Remove specified user from this site.

This workflow action is used to remove user from SharePoint site and from all site groups.

Parameters:

specified - Logon name of user.

this - Full URL of a site.

2.16 Rename Site at Url

Rename site at this url to this title.

This workflow action is used to set new title for site specified by URL anywhere within the SharePoint environment.

Parameters:

this url - Full URL of site.

this title - New title.

2.17 Set SharePoint List Anonymous Access

Set anonymous users can Add: **No**, Edit: **No**, Delete: **No**, View: **No** items at list at **list url**

This workflow action is used to set anonymous access for the list specified by URL. To use this action, anonymous access should be enabled for the web site: <http://office.microsoft.com/en-au/windows-sharepoint-services-help/enable-anonymous-access-HA010113018.aspx>

Parameters:

Add: **No** - Anonymous user can add items. Available values: *Yes, No*.

Edit: **No** - Anonymous user can edit items. Available values: *Yes, No*.

Delete: **No** - Anonymous user can delete items. Available values: *Yes, No*.

View: **No** - Anonymous user can view items. Available values: *Yes, No*.

list url - Full URL of target list.

2.18 Set SharePoint Site Anonymous Access

Set anonymous users can access: **Nothing** at **site url**.

This workflow action is used to set anonymous access for the site specified by URL. To use this action, anonymous access should be enabled for the web site: <http://office.microsoft.com/en-au/windows-sharepoint-services-help/enable-anonymous-access-HA010113018.aspx>

Parameters:

Nothing - What anonymous users can access at target web site. Available values: *Nothing, Lists and Libraries, All Web Site*.

site url - Full URL of target web site.

2.19 Set Site User Property

Set value to specified property for this user

This workflow action is used to set specified property value in user's profile on the site.

For MOSS, you can get the list of all available properties in Central Administration \ Manage User Properties.

See Also: "Set User's Multivalued Attribute in Active Directory" action.

Parameters:

value - Text, expression, item field or variable.

specified - Property display name.

this user - User, in whose profile property value is setting.

Examples:

The workflow below allows to user automatically change his picture in profile by uploading JPG file to documents library.

If Current Item:File Type equals jpg

then Set [%Current Item:URL Path%] to Picture property for Current Item:Created By

3. Arrays activities

3.1 Add Array Item

Add item to the end of the array variable

This workflow action is used to add a new item to the end of array or to copy source array specified in the first parameter to the end of the target array specified in the second parameter.

Parameters:

item - New item to add or source array to copy.

array variable - Target array variable.

3.2 Clear Array

Remove all items from the array variable.

This workflow action is used to remove all items from specified array variable.

Parameters:

array variable - Array variable.

3.3 Create Array

Create String array with initial items and store in Variable: ArrayVariable

This workflow action is used to create array variable.

Parameters:

String - Type of the array. Available types are: *String*, *Date/Time* - date and time values, *Integer* - positive or negative integer values, *Number* - floating point numbers.

initial - One or more items for array separated by semicolon.

Variable: ArrayVariable - Output array variable.

Examples:

Create String array with John;Mark;Alex items and store in Variable: array1

then Create String array with Jane;Kate;Olga items and store in Variable: array2

then Insert Variable: array2 into Variable: array1 at 1 index

then Get index of Mark item in Variable: array1 and store in Variable: index

then Log Variable: index to the workflow history list

Result in workflow history list: 4

3.4 Get Array Item

Get item at 0 index from array variable and store in variable.

This workflow action is used to get an item at specified index of array.

Non-negative values of index are zero-based and means 'from beginning of array', so 1 means 'second item of array'.

Negative value of index means 'from the end of array', so -1 means 'last item of array'.

Parameters:

0 - Index of array item.

array variable - Array variable.

variable - Output variable to store result.

3.5 Get Array Items Count

Get the number of items in the array variable and store in Variable: variable1

This action allows you to get the number of array items.

Parameters:

array variable - Array variable.

Variable: variable1 - Output integer variable.

Examples:

Get list of the Active Directory groups where user current user is member. Store list of these groups in Variable: GroupsArray, success of searching in Variable: UserFound.

then Get the number of items in the Variable: GroupsArray and store in Variable: groupsCount.

then Log [%Workflow Context: Current User%] is the member of [%Variable: groupsCount%] groups to the workflow history list

3.6 Get Index of Array Item

Get index of item item in the array variable and store in Variable: variable1

This workflow action is used to get zero-based index of specified item in array variable.

Result -1 means that item is not found.

Parameters:

<u>item</u> - Item to get index of.
<u>array variable</u> - Array variable.
<u>Variable: variable1</u> - Output integer variable to store result. -1 means that item is not found.

Examples:

Create String array with John;Mark;Alex items and store in Variable: array1

then Create String array with Jane;Kate;Olga items and store in Variable: array2

then Insert Variable: array2 into Variable: array1 at 1 index

then Get index of Mark item in Variable: array1 and store in Variable: index

then Log Variable: index to the workflow history list

Result in workflow history list: 4

3.7 Insert Array Item

Insert array item into array variable at 0 index.

This workflow action is used to insert new item to specified position of array variable. Or to insert source array specified in the first parameter to the specified position of target array specified in second parameter.

Parameters:

<u>item</u> - New item or source array to insert.
<u>array variable</u> - Target array variable.
<u>0</u> - Zero-based array index.

Examples:

Create String array with John;Mark;Alex items and store in Variable: array1

then Create String array with Jane;Kate;Olga items and store in Variable: array2

then Insert Variable: array2 into Variable: array1 at 1 index

then Get index of Mark item in Variable: array1 and store in Variable: index

then Log Variable: index to the workflow history list

Result in workflow history list: 4

3.8 Remove Array Item

Remove item from the array variable

This workflow action is used to delete first occurrence of specified item from array variable.

Parameters:

item - Item to remove from array.

array variable - Array variable.

3.9 Remove Array Item at

Remove the item at 0 index from array variable

This workflow action is used to remove item at specified zero-based index from array.

Parameters:

0 - Zero-based index of array item.

array variable - Array variable.

3.10 Remove Duplicates from Array

Remove duplicates from the array variable

This workflow action is used to delete duplicated values in array.

Parameters:

array variable - Array variable.

4. Date/Time activities

4.1 Converts Date from Local Time to UTC

Converts specified date from Local Time to UTC and store result in Variable: ConvertedToUTCDate.

This workflow action converts local date/time variable to Coordinated Universal Time (UTC) date/time. The UTC is equal to the local time minus the UTC offset. The conversion also takes into account the daylight saving time rule.

Parameters:

specified date - Input date or Date/Time variable.

Variable: ConvertedToUTCDate - Output Date/Time variable.

4.2 Converts Date from UTC to Local Time

Converts specified date from UTC to Local Time and store result in Variable: ConvertedToLocalTimeDate

This workflow action is used to convert Coordinated Universal Time (UTC) to local time. The local time is equal to the UTC time plus the UTC offset. The conversion also takes into account the daylight saving time rule.

Parameters:

specified date - Input UTC date or date/time variable

Variable: ConvertedToLocalTimeDate - Output local time/date variable

4.3 Converts Date to Text

Converts specified date to text with specified format and store result in Variable: TextDate

This workflow action is used to convert date/time variable to string using specified format.

Detailed description of format string syntax is contained in following articles:

- Standard DateTime Format Strings - [http://msdn.microsoft.com/en-us/library/aa720651\(v=VS.71\).aspx](http://msdn.microsoft.com/en-us/library/aa720651(v=VS.71).aspx)

- Custom DateTime Format Strings - [http://msdn.microsoft.com/en-us/library/aa719856\(v=VS.71\).aspx](http://msdn.microsoft.com/en-us/library/aa719856(v=VS.71).aspx)
- DateTime Formatting Sample - [http://msdn.microsoft.com/en-us/library/aa326720\(v=VS.71\).aspx](http://msdn.microsoft.com/en-us/library/aa326720(v=VS.71).aspx)

In the case of the wrong format string, the date will be converted with default format and warning message will be stored into workflow log file.

Parameters:

specified date - Input date/time variable.

specified format - Format string.

Variable: TextDate - Output string variable.

Examples:

Set DateTime constant Now and store result in Variable: d

then Convert Variable: d to text with MMMM dd yyyy HH:MM:ss.ffffff and store result in Variable: strDate

then Get Variable: d day property and store result in Variable: day

then Get Variable: d day of week property and store result in Variable: dow

then Get Variable: d day of year property and store result in Variable: doy

then Get Variable: d hour property and store result in Variable: h

then Get Variable: d millisecond property and store result in Variable: ms

then Get Variable: d minute property and store result in Variable: min

then Get Variable: d month property and store result in Variable: m

then Get Variable: d second property and store result in Variable: sec

then Get Variable: d year property and store result in Variable: y

then Log [%Variable: strDate%] : [%Variable: day%], [%Variable: dow%], [%Variable: doy%], [%Variable: h%], [%Variable: ms%], [%Variable: min%], [%Variable: m%], [%Variable: sec%], [%Variable: y%] to the workflow history list

Result in workflow history file:

4.4 Get Date Property

Get this date day property and store result in Variable: DateProperty.

This workflow action is used to get one from numeric values (like day of week or year) of date/time variable. See examples section for details.

Parameters:

this date - Input date/time variable.

day - What number from the date to store: *day, day of week, day of year, hour, millisecond, minute, month, second, year*. See examples section for details.

Variable: DateProperty - Output integer variable.

Examples:

Set DateTime constant Now and store result in Variable: d

then Convert Variable: d to text with MMMM dd yyyy HH:MM:ss.ffffff and store result in Variable: strDate

then Get Variable: d day property and store result in Variable: day

then Get Variable: d day of week property and store result in Variable: dow

then Get Variable: d day of year property and store result in Variable: doy

then Get Variable: d hour property and store result in Variable: h

then Get Variable: d millisecond property and store result in Variable: ms

then Get Variable: d minute property and store result in Variable: min

then Get Variable: d month property and store result in Variable: m

then Get Variable: d second property and store result in Variable: sec

then Get Variable: d year property and store result in Variable: y

then Log [%Variable: strDate%] : [%Variable: day%], [%Variable: dow%], [%Variable: doy%], [%Variable: h%], [%Variable: ms%], [%Variable: min%], [%Variable: m%], [%Variable: sec%], [%Variable: y%] to the workflow history list

Result in workflow history list:

April 25 2011 14:04:26.1626432 : 25, 2, 115, 14, 162, 16, 4, 26, 2011

4.5 Set DateTime Constant

Set DateTime constant Now and store result in Variable: DateConstant.

This workflow action is used to initialize date/time variables see parameters and examples sections for details.

Parameters:

Now - One from following date/time values: *Now, UtcNow, Today, Yesterday, Tomorrow, Beginning of this Week, End of this Week, Beginning of next Week, End of next Week, Beginning of this Month, End of this Month, Beginning of next Month, End of next Month, Beginning of this Year, End of this Year, Beginning of next Year, End of next Year*. See examples section for details.

Variable: DateConstant - Output date/time variable.

Examples:

Set DateTime constant Now and store result in Variable: vNow

then Set DateTime constant Today and store result in Variable: vToday

then Set DateTime constant End of this Week and store result in Variable: vEndOfWeek

then Set DateTime constant Beginning of next Year and store result in Variable: vBeginningNextYear

*then Log [%Variable: vNow%] * [%Variable: vToday%] * [%Variable: vEndOfWeek%] * [%Variable: vBeginningNextYear%] to the workflow history list*

Result in workflow history list:

4/25/2011 2:56:12 PM * 4/25/2011 12:00:00 AM * 4/30/2011 12:00:00 AM * 1/1/2012 12:00:00 AM

5. Development activities

5.1 Execute Custom Code

Execute this code in C# with these referenced assemblies and store result in variable.

This workflow action is used to execute custom C# or VB.NET code within workflow.

In the code, the object named Context with IExecuteCustomCodeLink interface is available. With this object, you can easily get access to workflow context, output Result variable and to any other workflow variable, activity execution context, and workflow history list (using LogComment function).

```
public interface IExecuteCustomCodeLink
{
    WorkflowContext WorkflowContext { get; }

    object Result { get; set; }

    object GetWorkflowVariable(string variableName);

    void SetWorkflowVariable(string variableName, object value);

    void LogComment(string comment);

    ActivityExecutionContext ExecutionContext { get; }
}
```

See examples section below.

Note: This workflow action is not safe. If you need to disable it due to security reasons, see article "How to disable certain HarePoint activities" in Section III of this Manual.

Parameters:

this code - C# or VB.NET code, see examples below. Use *using* C# keyword or *Imports* VB.NET statement to get access to additional namespaces. The following namespaces are already available in the code: System, Microsoft.SharePoint, System.Workflow.ComponentModel.

C# - Choose of two available languages: C# or VB.NET.

these referenced assemblies - Optional multi-line string with assemblies. Each line should contain a single file name. References to the following assemblies already exist even when this parameter is empty: System.dll, System.Data.dll, System.Xml.dll, System.Workflow.Activities.dll, System.Workflow.ComponentModel.dll, Microsoft.Sharepoint.WorkflowActions.dll, Microsoft.SharePoint.dll, HarePoint.Activities.dll.

variable - Output variable. This variable is accessible from code as *Context.Result*.

Examples:

Set DateTime constant Now and store result in Variable: MyDate

the Execute int DateShift = 10; DateTime odt = ... in C# with these referenced assemblies and store result in Variable: DateShift

then Log The date is moved for [%Variable: DateShift%] years, to [%Variable: MyDate%] to the workflow history list

This sample moves the date in MyDate variable for ten years from the code. The complete code for Execute Custom Code action is the following:

```
int DateShift = 10;
DateTime odt = (DateTime)Context.GetWorkflowVariable("MyDate");
DateTime ndt = odt.AddYears(DateShift);
Context.SetWorkflowVariable("MyDate", ndt);
Context.Result = DateShift;

Context.LogComment("Date is moved from custom code");
```

Result in workflow history list:

Date is moved from custom code

The date is moved for 10 years, to 6/30/2021 5:49:54 PM

5.2 Execute PowerShell Script

Execute this PowerShell script and store result in Variable: results. Get credentials from this Secure Store App ID.

This workflow action is used to execute custom PowerShell script within a workflow.

In the code, the object named \$Context is available. With this object, you can easily get access to workflow context, output Result variable and to any other workflow variable, activity execution context, and workflow history list (using Write-Host).

Additionally, two variables with user credentials available in the code: \$credential - NetworkCredential object, \$pscredential - PSCredential object.

See examples section below.

Note: This workflow action is not safe. If you need to disable it due to security reasons, see article "How to disable certain HarePoint activities" in Section III of this Manual.

Parameters:

this PowerShell script - PowerShell script, see example below.

Variable: results - Output variable. This variable is accessible from code as \$Context.Result.

this Secure Store App ID - Optional. Secure Store Service (SharePoint Server 2010) application ID. See <http://technet.microsoft.com/en-us/library/ee806889.aspx> for details.

Examples:

Execute `$x = $Context.GetWorkflowVariable("v... and store result in Variable: results. Get credentials from this Secure Store App ID.`

The complete code for Execute PowerShell Script action is the following:

```
$x = $Context.GetWorkflowVariable("v1")
$Context.LogComment("test comment")
Write-Host "test"
$Context.Result = "result string"
$Context.SetWorkflowVariable("v1", "medved")
Write-Host $Context.WorkflowContext.Web.Title
Get-ExecutionPolicy -Scope Process
Get-ExecutionPolicy -Scope CurrentUser
Get-ExecutionPolicy -Scope LocalMachine
Get-ExecutionPolicy
```

5.3 Execute SQL Command

Execute this SQL command using specified connection string for MS SQL database and store result in Variable: variable, number of records in Variable: recordsAffected.

This workflow action is used to execute SQL command on Microsoft SQL Server, Oracle, ODBC or OLE DB data sources.

For SQL language reference, see your DMBS documentation.

Parameters:

this SQL command - String, expression or variable with valid SQL command.

specified connection string - Database connection string. See your DMBS documentation for connection string format.

MS SQL - Type of database: MS SQL, ODBC, OLE DB or Oracle.

Variable: variable - Output array variable for retrieved records. If only one record was retrieved by SELECT command (value of *recordsAffected* is 1) then this array contains the record. If more than one record was retrieved by SELECT command (value of *recordsAffected* is 2 or greater) then this array contains values of the first column of recordset.

Variable: recordsAffected - Output integer variable with the number of records affected by command execution.

Examples:

Execute *INSERT INTO [Sheet1\$] (ModifiedAt, FileName, UserName) VALUES ('[%Current Item:Modified%]', '[%Current Item:Name%]', '[%Current Item:ModifiedBy%]');* using *DSN=FileDb* for *ODBC* database and store result in *Variable: variable2*, number of records in *Variable: affectedRecords*

then Execute *SELECT TOP 1 ModifiedAt, UserName FROM [Sheet1\$] WHERE FileName='[%Current Item:Name%]' ORDER BY ModifiedAt* using *DSN=FileDb* for *ODBC* database and store result in *Variable: variable3*, number of records in *Variable: affectedRows*

then Get item at *0* index from *Variable: variable3* and store in *Variable: FirstMod*

If *Variable: FirstMod* not equals *Current Item:Modified*

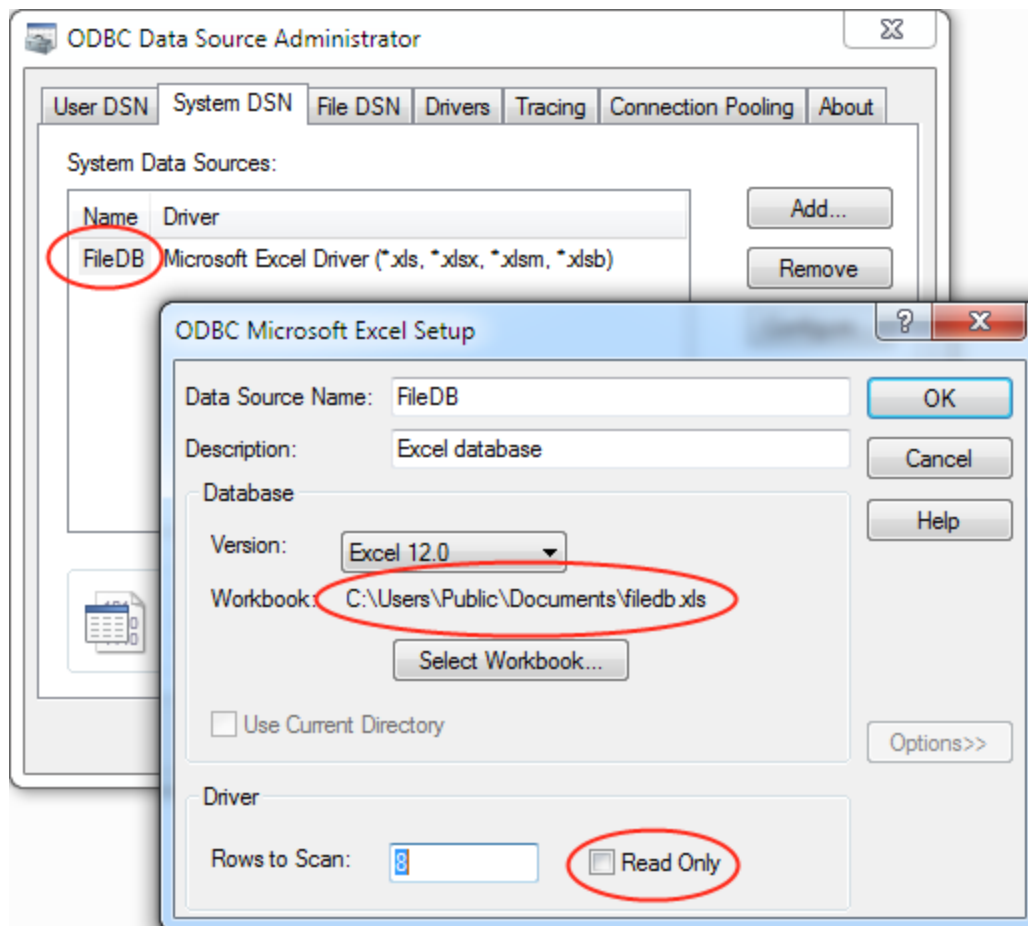
Get item at *1* index from *Variable: variable3* and store in *Variable: FirstModificationBy*

then Log *This document was first time created by [%Variable: FirstModificationBy%] at [%Variable: FirstMod%]* to the workflow history list

This sample uses a simple Excel file with the structure shown below as the database.

	A	B	C
1	ModifiedAt	FileName	UserName
2	6/30/2011 3:53:20 PM	2011-06-04 15-48-35 - Sasha - IMG_7591	GORLACH7\gorlach
3	7/1/2011 12:07:04 PM	2011-06-04 15-47-33 - Sasha - IMG_7588	GORLACH7\gorlach
4	7/1/2011 12:09:04 PM	2011-06-04 15-47-57 - Sasha - IMG_7590	GORLACH7\gorlach
5	7/1/2011 12:09:51 PM	2011-06-04 15-47-54 - Sasha - IMG_7589	GORLACH7\gorlach
6	7/1/2011 12:11:57 PM	2011-06-04 15-48-35 - Sasha - IMG_7591	GORLACH7\gorlach
7	7/1/2011 12:13:50 PM	2011-06-04 15-48-37 - Sasha - IMG_7592	GORLACH7\gorlach
8	7/1/2011 12:14:41 PM	2011-06-04 15-47-21 - Sasha - IMG_7586	GORLACH7\gorlach
9	7/1/2011 12:16:06 PM	2011-06-05 15-53-43 - Sasha - IMG_7600	GORLACH7\gorlach
10	7/1/2011 12:16:16 PM	2011-06-04 15-47-24 - Sasha - IMG_7587	GORLACH7\gorlach
11	7/1/2011 12:17:26 PM	2011-06-04 15-47-33 - Sasha - IMG_7588	GORLACH7\gorlach
12			

To run this sample, you should also create System DSN with the FileDB name for the file at Administrative Tools \ Data Sources (ODBC).



Configure this workflow to start automatically when document is added or changed in document library and it will log all activity with the documents to an Excel file.

Result in workflow history list:

This document was first time created by GORLACH7\gorlach at 7/1/2011 12:07:04 PM

5.4 Query XML

Query this XML using XPath query this query and store resulting text in Variable: variable

This workflow action is used to query XML document using XPath query language.

See Also:

- XPath Language Reference: <http://www.w3.org/TR/xpath/>

- XPath Online Tutorial: <http://www.w3schools.com/xpath/default.asp>
- Online XPath Query Tool: <http://www.bit-101.com/xpath/>

Parameters:

this XML - Input string, expression or variable with XML document.

this query - XPath query string. See examples section for details.

text - One from three available options: *text*, *inner XML* or *outer XML*.

Variable: variable - Output string variable.

Examples:

Send HTTP request to <http://www.ecb.int/stats/eurofxref/eurofxref-daily.xml> using GET method with this headers. Store response in Variable: CurrencyXml and status code in Variable: variable

then Query [%Variable: CurrencyXml%] using XPath query //Cube[@currency='USD']/@rate and store resulting text in Variable: EuroUsd

then Log European Central Bank EUR to USD conversion rate is: [%Variable: EuroUsd%] to the workflow history list

This sample takes XML document with the currency conversion rates from European Central Bank, and gets actual EUR/USD conversion rate.

Result in the workflow history list:

The request was sent successfully: <http://www.ecb.int/stats/eurofxref/eurofxref-daily.xml>

European Central Bank EUR to USD conversion rate is: 1.4453

Example of XML document of European Central Bank:

```
<gesmes:Envelope xmlns:gesmes="http://www.gesmes.org/xml/2002-08-01"
  xmlns="http://www.ecb.int/vocabulary/2002-08-01/eurofxref">
```

```
<gesmes:subject>Reference rates</gesmes:subject>
```

```
<gesmes:Sender>
```

```
<gesmes:name>European Central Bank</gesmes:name>
```

```
</gesmes:Sender>
```

```
<Cube>
```

```

<Cube time="2011-06-30">

<Cube currency="USD" rate="1.4453"/>

<Cube currency="JPY" rate="116.25"/>

<Cube currency="BGN" rate="1.9558"/>

<Cube currency="CZK" rate="24.345"/>

<Cube currency="DKK" rate="7.4587"/>

<Cube currency="GBP" rate="0.90255"/>

</Cube>

</Cube>

</gesmes:Envelope>

```

5.5 Transform XML

Transform this XML using XSLT this transform and store result in Variable: variable

This workflow action is used for XML transformations using XSLT template.

See Also:

- XSL Transformations Version 1.0: <http://www.w3.org/TR/xslt>
- Online XSLT Tutorial: <http://www.w3schools.com/xsl/default.asp>
- Online XSLT Design
Tool: <http://www.w3schools.com/xsl/tryxslt.asp> xmlfile=catalog&xsltfile=catalog

Parameters:

this XML - Input string, expression or variable with XML document.

this transform - Input string, expression or variable with XSLT template.

Variable: variable - Output string variable.

Examples:

Send HTTP request to <http://www.ecb.int/stats/eurofxref/eurofxref-daily.xml> using GET method with this headers. Store response in Variable: Xml and status code in Variable: Code

then Transform [%Variable: Xml%] using XSLT `<xsl:stylesheet version="1.0" xmlns:xsl=...` and store result in Variable: Html

*then Create document from Variable: Html in <http://gorlach7/Shared%20Documents/Rates.Html>.
Overwrite existing document.*

This sample takes XML document with currency conversion rates from European Central Bank and generates HTML document with currency rates in document library.

For a sample of XML document, please see the examples section of 'Query XML' workflow action overview.

Full text of XSLT template for this sample:

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:gesmes="http://www.gesmes.org/xml/2002-08-01"
xmlns:cubes="http://www.ecb.int/vocabulary/2002-08-01/eurofxref">
<xsl:template match="/">
  <html>
  <body>
    <h2><xsl:value-of select="//gesmes:name"/>: <xsl:value-of select="//cubes:Cube/@time" /></h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th>Currency</th>
        <th>Rate</th>
      </tr>
      <xsl:for-each select="//cubes:Cube/cubes:Cube/cubes:Cube">
        <tr>
          <td><xsl:value-of select="@currency"/></td>
          <td><xsl:value-of select="@rate"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
```

```

</xsl:for-each>

</table>

</body>

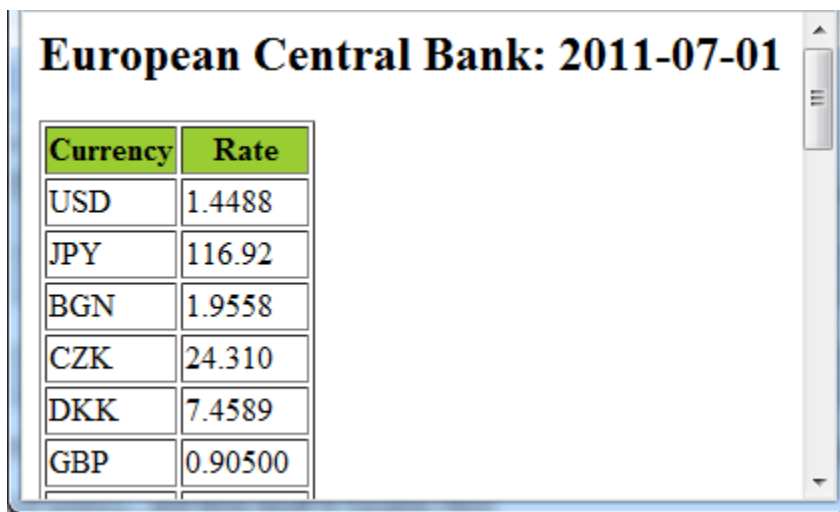
</html>

</xsl:template>

</xsl:stylesheet>

```

Example of output HTML document:



Currency	Rate
USD	1.4488
JPY	116.92
BGN	1.9558
CZK	24.310
DKK	7.4589
GBP	0.90500

5.6 Update XML

Update this XML, update node value using XPath this query with this value, store result in Variable: variable

This workflow action is used to update node value or append new child node to XML document using XPath query.

See Also:

- XPath Language Reference: <http://www.w3.org/TR/xpath/>
- XPath Online Tutorial: <http://www.w3schools.com/xpath/default.asp>
- Online XPath Query Tool: <http://www.bit-101.com/xpath/>

Parameters:

this XML - Input text, expression or variable with XML document.

update node value - Two options are available: *update node value* or *append child node*.

this query - Input text, expression or variable with XPath query.

this value - New node value or new child node XML.

Variable: variable - Output string variable for updated XML document.

Examples:

1. Update node sample

Send HTTP request to <http://www.ecb.int/stats/eurofxref/eurofxref-daily.xml> using GET method with this headers. Store response in Variable: Xml and status code in Variable: Code

then Query [%Variable: Xml%] using XPath query //Cube[@currency='USD']/@rate and store resulting text in Variable: EuroUsd

then Set Variable: EuroUsdNumber to Variable: EuroUsd

then Calculate Variable: EuroUsdNumber multiply by 1,1 (Output to Variable: EuroUsdCorrected)

then Update [%Variable: Xml%], update node value using XPath //Cube[@currency='USD']/@rate with [%Variable: EuroUsdCorrected%], store result in Variable: NewXml

then Transform [%Variable: NewXml%] using XSLT <xsl:stylesheet version="1.0" xmlns:xsl=... and store result in Variable: Html

then Create document from Variable:

Html in <http://gorlach7/Shared%20Documents/Rates.Html>. Overwrite existing document.

This sample takes XML document with currency conversion rates from European Central Bank, increases EUR/USD conversion rate by 10%, and generates HTML document with currency rates in document library.

Note that type of EuroUsdNumber variable is Number. For the full text of XSLT template and example of output HTML see examples section of 'Transform XML' activity overview. For the sample of source XML document see examples section of 'Query XML' activity overview.

2. Append child node sample

Send HTTP request to <http://www.ecb.int/stats/eurofxref/eurofxref-daily.xml> using GET method with this headers. Store response in Variable: Xml and status code in Variable: Code

then Update [%Variable: Xml%], append child node value using XPath //Cube/Cube[@currency='USD'] with <Cube currency="EUR" rate="1.0000"/>, store result in Variable: NewXml

then Transform [%Variable: NewXml%] using XSLT <xsl:stylesheet version="1.0" xmlns:xsl=... and store result in Variable: Html

then Create document from Variable: Html in http://gorlach7/Shared%20Documents/Rates.Html. Overwrite existing document.

This sample takes XML document with currency conversion rates from European Central Bank, add new node with EUR/EUR rate 1.0000, and generates HTML document with currency rates in document library.

For the full text of the XSLT template and example of output HTML, see examples section of the 'Transform XML' activity overview. For a sample of source XML document, please see examples section of the 'Query XML' activity overview.

6. Document Conversion activities

6.1 Convert Document

Convert this document to PDF format and store job ID in Variable: ConversionJobId. Overwrite existing document.

This workflow action is used to convert specified document to different format and store output file in the same document library.

This action process only the files of supported formats, files of other formats will be skipped. List of supported formats:

- Open XML Document (.docx, .docm, .dotx, .dotm)
- Word 97-2003 Document (.doc, .dot)
- Rich Text Format (.rtf)
- Web Page (.htm, .html, .mht, .mhtml)
- Word 2003 XML Document (.xml)

Action returns empty string in ConversionJobId output variable if no files were found to convert.

Note: This action require "Word Automation Services" running on server. To check this service, open Central Administration, go to System Settings and open "Manage services on server" page. Unfortunately, this service is not available on SharePoint Foundation.

See Also: "Wait Conversion Job" and "Get Conversion Job Status" actions.

Parameters:

this - Document to convert, e.g. Current Item.

PDF - Choose one from available formats: *Word Document (docx)*, *Word 97-2003 Document (doc)*, *PDF*, *RTF*, *XPS*.

Variable: ConversionJobId - Output string variable with ID of conversion job. Action returns empty string if no files were found for conversion.

Overwrite - If the folder already contains file with the same name, then you can apply one from two options. *Overwrite* - replace file by new one. *Do not overwrite* - skip this file.

Examples:

See example at "Wait Conversion Job" action description.

6.2 Convert Folder at Url

Convert documents in folder at this url to PDF format and store documents to folder at specified url, store job ID in Variable: ConversionJobId. Overwrite existing document.

This workflow action is used to convert all documents from document library specified by URL. Output files maybe stored in the same or at another document library.

This action process only the files of supported formats, files of other formats will be skipped. List of supported formats:

- Open XML Document (.docx, .docm, .dotx, .dotm)
- Word 97-2003 Document (.doc, .dot)
- Rich Text Format (.rtf)
- Web Page (.htm, .html, .mht, .mhtml)
- Word 2003 XML Document (.xml)

Action returns empty string in ConversionJobId output variable is no files were found to convert.

Note: This action require "Word Automation Services" running on server. To check this service, open Central Administration, go to System Settings and open "Manage services on server" page. Unfortunately, this service is not available on SharePoint Foundation.

See Also: "Wait Conversion Job" and "Get Conversion Job Status" actions.

Parameters:

this url - Full URL of source document library or URL of a folder of document library.

PDF - Choose one from available formats: *Word Document (.docx)*, *Word 97-2003 Document (.doc)*, *PDF*, *RTF*, *XPS*.

specified url - Full URL of target document library or URL of a folder of document library.

Variable: ConversionJobId - Output string variable with ID of conversion job. Action returns empty string if no files were found for conversion.

Overwrite - If the folder already contains file with the same name, then you can apply one from two options. *Overwrite* - replace file by new one. *Do not overwrite* - skip this file.

6.3 Convert Library

Convert documents in this library to PDF format and store converted documents to specified library, store job ID in Variable: ConversionJobId. Overwrite existing document.

This workflow action is used to convert all documents from document library. Output files maybe stored in the same or at another document library.

This action process only the files of supported formats, files of other formats will be skipped. List of supported formats:

- Open XML Document (.docx, .docm, .dotx, .dotm)
- Word 97-2003 Document (.doc, .dot)
- Rich Text Format (.rtf)
- Web Page (.htm, .html, .mht, .mhtml)
- Word 2003 XML Document (.xml)

Action returns empty string in ConversionJobId output variable is no files were found to convert.

Note: This action require "Word Automation Services" running on server. To check this service, open Central Administration, go to System Settings and open "Manage services on server" page. Unfortunately, this service is not available on SharePoint Foundation.

See Also: "Wait Conversion Job" and "Get Conversion Job Status" actions.

Parameters:

this library - Source document library on current site.

PDF - Choose one from available formats: *Word Document (docx)*, *Word 97-2003 Document (doc)*, *PDF*, *RTF*, *XPS*.

specified library - Target document library on current site.

Variable: ConversionJobId - Output string variable with ID of conversion job. Action returns empty string if no files were found for conversion.

Overwrite - If the folder already contains file with the same name, then you can apply one from two options. *Overwrite* - replace file by new one. *Do not overwrite* - skip this file.

6.4 Get Conversion Job Status

Get conversion job with this job ID status and store in Variable: JobStatus; store links to converted documents in Variable: DocumentsArray.

This workflow action is used to get a string with status of conversion job, one from three values is possible: *Completed*, *Failed* or *Running*. For completed job, it also returns array with URLs of new documents.

See Also: "Wait Conversion Job" and "Get Array Items Count" actions.

Parameters:

with this job ID - ID of conversion job returned by "Convert Document", "Convert Folder at Url" or "Convert Library" actions.

Variable: JobStatus - Output string variable for the job status, contains one from three possible values: *Completed*, *Failed* or *Running*.

Variable: DocumentsArray - Output array variable for the full URLs of new files. Use "Join Text" or "Get Array Item" actions to get string values.

Examples:

See example at "Wait Conversion Job" action description.

6.5 Wait Conversion Job

Wait while conversion job with this job ID is running.

This workflow action is used to wait while specified conversion job is running. Note that conversion job may take from few minutes to several hours depends on different factors.

See Also: "Convert Document" and "Get Conversion Job Status" actions.

Parameters:

with this job ID - ID of conversion job returned by "Convert Document", "Convert Folder at Url" or "Convert Library" actions.

Examples:

Convert document Current Item to PDF format and store job ID in Variable: ConversionJobId. Overwrite existing document.

then Wait while conversion job Variable: ConversionJobId is running

then Get conversion job Variable: ConversionJobId status and store in Variable: JobStatus; store links to converted documents in Variable: DocumentsArray

then Get item at Q index from Variable: DocumentsArray and store in Variable: FileName

then Log Status: [%Variable: JobStatus%]. File name: [%Variable: FileName%] to the workflow history list

Result in workflow history list:

Status: Completed. File name: http://sp14.vlab/Shared Documents/Invoice 1730A.pdf

7. Document Library activities

7.1 Check In Item at Url

Do minor check in of list item at specified url with specified comment

This workflow action is used to check in specified item with a comment.

Note: This action is unavailable in site workflows or list workflows of different type, e.g. Calendar or Tasks.

Parameters:

minor check in - Check in type, one from following options: *minor check in*, *major check in*, *overwrite check in*.

specified url - Full list item URL including site name.

specified comment - Comment text.

7.2 Check Out Item at Url

Check out item at specified url

This workflow action is used to check out list item specified by URL.

Note: This action is unavailable in site workflows or list workflows of different type, e.g. Calendar or Tasks.

Parameters:

specified url - Full URL of list item including site name.

7.3 Copy or Move Document

Copy document at this url to the folder at specified url and with the same name. Overwrite existing document.

This workflow action is used to copy, move or rename documents in SharePoint document libraries.

Note: This action is unavailable in site workflows or list workflows of different type, e.g. Calendar or Tasks.

Parameters:

Copy - Copy or move document.

this url - URL of the document at SharePoint Server.

specified url - URL of the target folder, may be the same as source folder.

the same - Optional parameter, new file name and extension for the document.

Overwrite - If the folder already contains file with the same name, then you can apply one from two options. *Overwrite* - replace file by new one. *Do not overwrite* - skip this file.

Examples:

Move document at <http://gorlach7/Shared%20Documents/1.png> to the folder at <http://gorlach7/Shared%20Documents/> and with [2.png](#) name. Overwrite existing document.

This sample workflow renames 1.PNG file to 2.PNG.

7.4 Create Document from File

Create document from file path in parent folder url. Overwrite existing document. Trigger events.

This workflow action is used to add new document or replace existing document by file from disk or network.

Note: This action is unavailable in site workflows or list workflows of different type, e.g. Calendar or Tasks.

Note: In order to specify file access credentials, add the "Set Network File Access" activity to the same workflow step first.

Parameters:

file path - Disk file path or network file path.

parent folder url - Full URL including site name of document library of the folder in document library.

Overwrite - If the folder already contains file with the same name, then you can apply one from three options. *Overwrite* - replace file in library by specified file. *Do not overwrite* - generate an error. *Append unique identifier* - add string like _20e0ad06-4261-4349-8a37-894fa7574f42 to the end of file name.

Trigger - Specifies whether to trigger an item create event.

7.5 Create Document from Text

Create document from this text in file url. Overwrite existing document. Trigger events.

This workflow action is used to add new document or replace existing document by new file created from string variable.

Note: This action is unavailable in site workflows or list workflows of different type, e.g. Calendar or Tasks.

Parameters:

this text - Document body.

file url - Full URL for the new document, including site name and file name.

Overwrite - If the folder already contains file with the same name, then you can apply one from three options. *Overwrite* - replace file in library by specified file. *Do not overwrite* - generate an error. *Append unique identifier* - add string like _20e0ad06-4261-4349-8a37-894fa7574f42 to the end of file name.

Trigger - Specifies whether to trigger an item create event.

7.6 Create Folder in the Document Library

Create specified folder in the current folder.

This workflow action is used to create folder in document library.

Note: This action is unavailable in site workflows or list workflows of different type, e.g. Calendar or Tasks.

Parameters:

specified - Name of the folder to create.

current/root - Specifies where to create a new folder: in the current folder of document library or in the root folder.

7.7 Create Folder in the Document Library at Url

Create specified folder at folder url.

This workflow action is used to create subfolder in document library or in folder of document library.

Note: This action is unavailable in site workflows or list workflows of different type, e.g. Calendar or Tasks.

Parameters:

specified - Folder name.

url - Full URL with site name of document library or folder in document library.

7.8 Delete Folder in the Document Library

Delete specified folder in the current folder.

This workflow action is used to delete specified folder with all its content in document library. The error happens if the folder is not found.

Note: This action is unavailable in site workflows or list workflows of different type, e.g. Calendar or Tasks.

Parameters:

specified - Name of the folder to delete.

current/root - Specifies where to delete the folder: in the current folder of document library or in the root folder.

7.9 Delete Folder in the Document Library at Url

Delete specified folder at folder url

This workflow action is used to delete subfolder in document library or in folder of document library.

Note: This action is unavailable in site workflows or list workflows of different type, e.g. Calendar or Tasks.

Parameters:

specified - Subfolder name to delete.

url - Full URL with site name of document library or name of the folder in the document library.

7.10 Discard Check Out Item at Url

Discard Check Out item at specified url

This workflow action is used to discard check out of specified list item.

Note: This action is unavailable in site workflows or list workflows of different type, e.g. Calendar or Tasks.

Parameters:

specified url - Full list item URL including site name.

7.11 Discard Publish Item

Discard publish list item at this url with specified comment

This workflow action is used to discard list item publishing.

Note: This action is unavailable in site workflows or list workflows of different type, e.g. Calendar or Tasks.

Parameters:

this url - Full URL of target list item.

specified comment - Comment text.

7.12 Download Documents to Folder

Recursively download all documents from this folder url to specified path. Overwrite existing files.

This workflow action is used to save all documents from document library or folder of document library to local disk or to network share.

Note: This action is unavailable in site workflows or list workflows of different type, e.g. Calendar or Tasks.

Note: In order to specify file access credentials, add the "Set Network File Access" activity to the same workflow step first.

Parameters:

Recursively - One from two options. *Recursively*: save the document library or folder content including all subfolders, subfolders tree will be created in *specified path*. *Non-recursively*: save files only from specified document library or folder.

all - File mask, like *.jpg. You can specify only one file mask.

this folder url - Document library or folder of document library.

specified path - Local disk or network path.

Overwrite - If the folder already contains file with the same name, then you can apply one from four options. *Overwrite* - replace file in library by new file. *Do not overwrite* - generate an error. *Skip* - do not

overwrite file in library without error. *Append unique identifier for* - add string like _20e0ad06-4261-4349-8a37-894fa7574f42 to the end of file name.

7.13 Get Document Content

Get content of the document at this url and store in Variable: variable.

This workflow action is used to retrieve contents of specified document in to output variable.

Parameters:

this url - URL of the document at SharePoint Server.

Variable: variable - Output variable.

7.14 Get Entry From Folder Property Bag

Get specified entry value from list item at this url parent folder property bag and store in Variable: variable

This workflow action is used to get specified entry value from property bag of parent folder of specified list item.

For the list of property bag entries see: [http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.spfolder.properties\(v=office.12\).aspx](http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.spfolder.properties(v=office.12).aspx)

Error happens if property is not found.

Note: This action is unavailable in site workflows or list workflows of different type, e.g. Calendar or Tasks.

Parameters:

specified - Name of the entry in property bag, e.g. vti_timelastmodified.

this url - Full URL of list item.

Variable: variable - Output variable.

7.15 Publish Item

Publish list item at this url with specified comment

This workflow action is used to submit the list item for content approval with the specified comment.

Note: This action is unavailable in site workflows or list workflows of different type, e.g. Calendar or Tasks.

Parameters:

this url - Full URL of target list item.

specified - Comment text.

7.16 Rename Document at Url

Rename document at specified url to this file name

This workflow action is used to rename list item specified by URL.

Note: This action is unavailable in site workflows or list workflows of different type, e.g. Calendar or Tasks.

Parameters:

specified url - Full URL of list item including site name.

this file name - New file name without extension.

7.17 Save Document to File

Save list item at this url to output folder. Overwrite existing file.

This workflow action is used to save document from documents library to local disk or network share.

Note: This action is unavailable in site workflows or list workflows of different type, e.g. Calendar or Tasks.

Note: In order to specify file access credentials, add the "Set Network File Access" activity to the same workflow step first.

Parameters:

this url - Full URL of list item.

output folder - Local disk folder or network share.

Overwrite - If the folder already contains file with the same name, then you can apply one from four options. *Overwrite* - replace file in library by new file. *Do not overwrite* - generate an error. *Skip* - do not overwrite file in library without error. *Append unique identifier for* - add string like _20e0ad06-4261-4349-8a37-894fa7574f42 to the end of file name.

Examples:

Save [%Current Item:Encoded Absolute URL%] to C:\temp\ *Overwrite* existing file.

7.18 Set Network File Access

Set network file access using this user and this password.

This workflow action is used to specify credentials for a group of Document Library activities that works with files:

- Save Document to File
- Create Document From File
- Upload Documents From Folder
- Download Documents To Folder
- Upload Files To List Item Attachments

Note: The password will be stored as a plain text in the workflow XOML file and may be accessed by the URL like *http://sharepoint/Workflows/workflowname/workflowname.xoml* or by SharePoint Designer.

Parameters:

this user - User name that will be used for file access (like DOMAIN\Account).

this password - Password that will be used for file access (see Note).

Examples:

Set network file access using *HAREPOINT\testuser* and *pwd123*.

7.19 Upload Documents from Folder

Recursively upload all files from specified path to this folder url. Overwrite existing documents. Trigger events.

This workflow action is used to upload files from local disk or network share to document library.

Note: This action is unavailable in site workflows or list workflows of different type, e.g. Calendar or Tasks.

Note: In order to specify file access credentials, add the "Set Network File Access" activity to the same workflow step first.

Parameters:

Recursively - One from two options. *Recursively*: upload files from specified path including all subfolders, subfolders tree will be created in specified document library. *Non-recursively*: upload files only from specified path.

all - File mask, like *.jpg. You can specify only one file mask.

specified path - Disk or network path.

this folder url - Document library or folder of document library.

Overwrite - If the folder already contains file with the same name, then you can apply one from four options. *Overwrite* - replace file in library by new file. *Do not overwrite* - generate an error. *Skip* - do not overwrite file in library without error. *Append unique identifier for* - add string like _20e0ad06-4261-4349-8a37-894fa7574f42 to the end of file name.

Trigger - Specifies whether to trigger an item create/update event.

Examples:

Recursively upload all files from C:\ to http://mysite/Shared%20Documents/. Overwrite existing documents

Never try this example.

8. E-Mail activities

8.1 Get E-mail Property

Get HTML body from this mime body and store in Variable: variable.

This workflow action is used to retrieve message properties from the mime body. The list of available properties: HTML body, TEXT body, sender name, sender e-mail, CC recipients, reply to, date sent, importance, mail headers.

See Also: "Receive E-Mail" action.

Parameters:

HTML body - One of the available message properties.

this mime - Mime body of the received e-mail message .

Variable: variable - String value of the specified property .

Examples:

Get HTML body from Variable: mimeType body and store in Variable: variable.

8.2 Receive All E-Mails to List

Download all e-mails from this server, secure connection type None, login this login and password this password. Store them with attachments into the list at this url. Store inline images in folder at this url.

This workflow action is used to download all messages from the specified POP3 mailbox to the prepared SharePoint list. The messages are deleted from the mailbox after addind into SharePoint list.

The prepared SharePoint list may contain following fields, that will be filled by the values from the downloaded messages:

Column	Data Type	E-mail header
Title	Single line of text	Subject
Description	Multiple lines of text / enhanced rich text	Html Body
FromName	Single line of text	Sender display name
FromAddress	Single line of text	Sender e-mail address
FromUser	Person or Group	The program tries to find SharePoint user with the message sender e-mail address.
Date	Date and Time	Date Sent
Importance	Single line of text	Message importance: Low, Normal, High
TextBody	Multiple lines of text	Text body of the message
ToString	Single line of text	"To" field with names and e-mail addresses
CCString	Single line of text	"CC" field with names and e-mail addresses
ReplyToString	Single line of text	"Reply-To" header
MailHeaders	Multiple lines of text	All message headers

Parameters:

this server - POP3 server name or IP-address. By default port 110 is used, you can specify port with colon, like mail.company.com:1110.

type **None** - Secure connection type. Available values: None, SSL. Note, that POP3 server with SSL usually listen for incoming connections at port 995.

this login - User name for the POP3 server.

this password - Password for the POP3 server.

with - Specifies whether to attach message attachments to list item or not.

this url - Full URL of the list.

this url - Full URL of the document library where inline attachments will be stored. Optional.

Examples:

Download all e-mails from pop.gmail.com:995, secure connection type SSL, login harepoint@gmail.com and password pwd123. Store them with attachments into the list at Variable: listUrl. Store inline images in folder at Variable: folderUrl.

8.3 Receive E-Mail

Receive email from this server, secure connection type None, login this login and password this password. Store e-mail sender in Variable:variable1, recipient in Variable:variable2, subject in Variable:variable3 and HTML body in Variable:variable4.

This workflow action is used to receive e-mail message from POP3 mail server. Action gets the list of messages in mailbox, receives the first one and delete received message from mailbox.

Note: The password will be stored as a plain text in the workflow XOML file and may be accessed by the URL like http://sharepoint/Workflows/workflowname/workflowname.xoml or by SharePoint Designer.

See Also: "Save E-Mail Attachments to List Item" and "Get E-Mail Property" actions.

Parameters:

this server - POP3 server name or IP-address. By default port 110 is used, you can specify port with colon, like mail.company.com:1110.

type **None** - Secure connection type. Available values: *None*, *SSL*. Note, that POP3 server with SSL usually listen for incoming connections at port 995.

this login - User name for the POP3 server.

this password - Password for the POP3 server.

Variable: variable1 - Sender address of received e-mail message.

Variable: variable2 - Recipients addresses of received e-mail message.

Variable: variable3 - Subject of received e-mail message.

Variable: variable4 - Body of received e-mail message. The type of body is specified by the previous parameter. It can be one of the following: HTML, TEXT or MIME.

Examples:

Receive email from *pop.gmail.com:995*, secure connection type *SSL*, login *harepoint@gmail.com* and password *pwd123*. Store e-mail sender in *Variable: sender*, recipient in *Variable: recipient*, subject in *Variable: subject* and *HTML* body in *Variable: body*.

8.4 Save E-Mail Attachments to List Item

Save all attachments from this mime body to list item at this url. Store names of saved attachments to Variable: array.

This workflow action is used to retrieve attachments from the e-mail message and attach them to the specified list item.

See Also: "Receive E-Mail" action.

Parameters:

all - File masks, separated by semicolon. Leave the field empty to save all attachments.

this mime - String with the message mime body.

this url - Full URL of list item.

Variable: array - Array of the saved attachment names.

Examples:

Save *.jpg;*.png attachments from Variable: mimeType body to list item at Variable: itemUrl. Store names of saved attachments to Variable: array.

8.5 Send E-Mail with Document as Attachment

Send e-mail to this address with Normal importance, blind carbon copy to this address, display this address as sender and include document this document as attachment in the e-mail.

This workflow action is used to send e-mail with the document from document library as attachment.

By default, farm settings for outgoing emails (Central Administration \ Outgoing E-Mail Settings) will be used.

See Also: "Set SMTP Server Options" action.

Parameters:

e-mail to this address - Click here to specify message recipients, carbon copy recipients, message subject and body.

Normal - E-mail importance. Available options: *Normal, Low, High*.

blind carbon copy to this address - Blind carbon copy recipients, maybe blank.

this address as sender - Sender address. If this value is not specified then value from farm settings for outgoing emails will be used.

this document - Item from document library. For example, *Current Item* for the document library workflow.

8.6 Send E-Mail with File Attachments

Send e-mail to this address with Normal importance, blind carbon copy to this address, use this address as sender and include a file this file as attachment.

This workflow action is used to send e-mail with one or more file attachments from local disk or network share.

By default, farm settings for outgoing emails (Central Administration \ Outgoing E-Mail Settings) will be used.

See Also: "Set SMTP Server Options" action.

Parameters:

e-mail to this address - Click here to specify message recipients, carbon copy recipients, message subject and body.

Normal - E-mail importance. Available options: *Normal, Low, High*.

blind carbon copy to this address - Blind carbon copy recipients, maybe blank.

this address as sender - Sender address. If this value is not specified then value from farm settings for outgoing emails will be used.

this file - Multi-line string with one or more file names from local disk or network share to attach. Each file name must be on a separate line.

Examples:

Use SMTP server exchange.local:125, secure connection type None, login this login and password this password.

then Send e-mail to gorlach@exchange.local with Normal importance, blind carbon copy to this address, use noreply@exchange.local as sender and include a file C:\temp\report.pdf as attachment.

8.7 Send E-Mail with Http File Attachment

Send e-mail to this address with Normal importance, blind carbon copy to this address, use this address as sender and include a file from this url as attachment named file name.

This workflow action is used to send e-mail with file attachment specified by URL.

By default, farm settings for outgoing emails (Central Administration \ Outgoing E-Mail Settings) will be used.

See Also: "Set SMTP Server Options" action.

Parameters:

e-mail to **this address** - Click here to specify message recipients, carbon copy recipients, message subject and body.

Normal - E-mail importance. Available options: *Normal, Low, High*.

blind carbon copy to **this address** - Blind carbon copy recipients, maybe blank.

this address as sender - Sender address. If this value is not specified then value from farm settings for outgoing emails will be used.

this URL - Full URL to file.

file name - Optional new name and extension for attached file, leave blank to use original name.

Examples:

Send e-mail to harepoint@gmail.com with Normal importance, blind carbon copy to this address, use noreply@harepoint.com as sender and include a file from Workflow Context:Current Item URL as attachment named file name.

8.8 Send E-Mail with InfoPath Attachments

Send e-mail to **this address** with **Normal** importance, blind carbon copy to **this address**, use **this address** as sender and add InfoPath attachment **specified by XPATH** of list item at **this url** to attachments of e-mail.

This workflow action is used to send e-mail with InfoPath attachments of list item.

By default, farm settings for outgoing emails (Central Administration \ Outgoing E-Mail Settings) will be used.

See Also: "Set SMTP Server Options" action.

Parameters:

e-mail to **this address** - Click here to specify message recipients, carbon copy recipients, message subject and body.

Normal - E-mail importance. Available options: *Normal, Low, High*.

blind carbon copy to **this address** - Blind carbon copy recipients, maybe blank.

this address as sender - Sender address. If this value is not specified then value from farm settings for outgoing emails will be used.

specified by XPATH - XPath query string. See examples section for details.

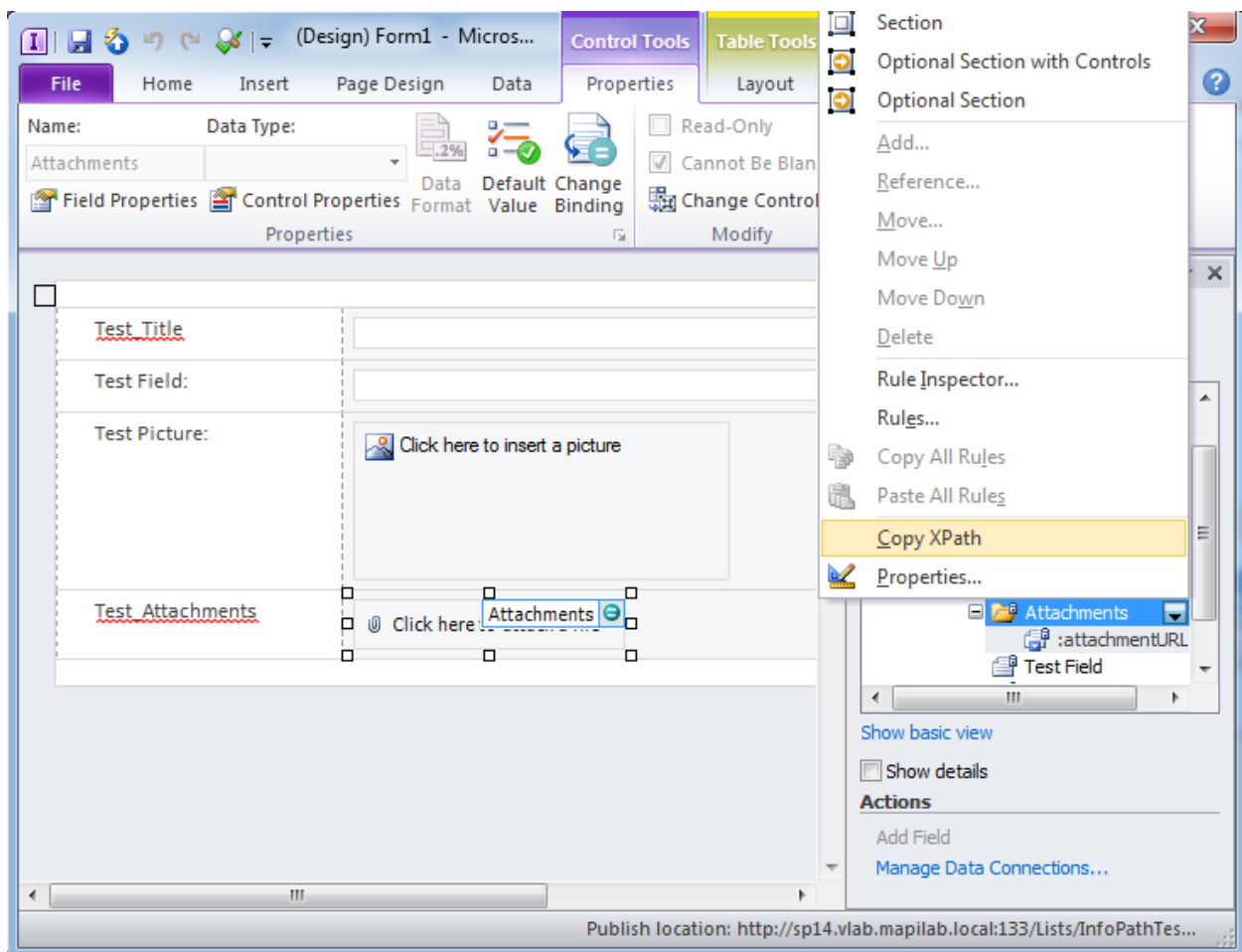
this item - InfoPath attachments of this list item will be attached to e-mail.

Examples:

Send e-mail to harepoint@gmail.com with *Normal* importance, blind carbon copy to [this address](#), use noreply@harepoint.com as sender and add InfoPath attachment [/dfs:myFields/dfs:dataFields/my:SharePointListItem_RW/my:Attachments](#) of list item at *Variable: thisUrl* to attachments of e-mail.

You can combine multiple XPath queries using vertical bar ('|') sign, e.g.
[/my:myFields/my:MoreAttachments](#) | [/my:myFields/my:AttachFile](#).

You can retrieve XPath string using Microsoft InfoPath Designer:



8.9 Send E-Mail with List Item Attachments

Send e-mail to this address with Normal importance, blind carbon copy to this address, use this address as sender and include all files attached to this item to attachments of e-mail.

This workflow action is used to send e-mail with attachments of list item. For example, you can send attachments of Task or Calendar item with this action.

By default, farm settings for outgoing emails (Central Administration \ Outgoing E-Mail Settings) will be used.

See Also: "Set SMTP Server Options" action.

Parameters:

e-mail to this address - Click here to specify message recipients, carbon copy recipients, message subject and body.

Normal - E-mail importance. Available options: *Normal, Low, High*.

blind carbon copy to this address - Blind carbon copy recipients, maybe blank.

this address as sender - Sender address. If this value is not specified then value from farm settings for outgoing emails will be used.

this item - Attachments of this list item (like Calendar or Task) will be attached to e-mail.

8.10 Send E-Mail without Attachments

Send e-mail to this address with Normal importance, blind carbon copy to this address, display this address as sender.

This workflow action is used to send e-mail without any attachments.

By default, farm settings for outgoing emails (Central Administration \ Outgoing E-Mail Settings) will be used.

See Also: "Set SMTP Server Options" action.

Parameters:

e-mail to this address - Click here to specify message recipients, carbon copy recipients, message subject and body.

Normal - E-mail importance. Available options: *Normal, Low, High*.

blind carbon copy to **this address** - Blind carbon copy recipients, maybe blank.

this address as sender - Sender address. If this value is not specified then value from farm settings for outgoing emails will be used.

8.11 Set SMTP Server Options

Use SMTP server this server, secure connection type None, login this user and password this password.
Do not convert message body to plain text.

This workflow action is used to set SMTP server settings for HarePoint SMTP-related activities like "Send E-Mail with List Item Attachments". This activity must be placed above all other SMTP-related activities in the current workflow step.

Otherwise, farm settings for outgoing emails (Central Administration \ Outgoing E-Mail Settings) will be used for SMTP-related HarePoint activities.

Note: The password will be stored as a plain text in the workflow XOML file and may be accessed by the URL like <http://sharepoint/Workflows/workflowname/workflowname.xoml> or by SharePoint Designer.

Parameters:

this server - SMTP server name or IP-address. By default port 25 is used, you can specify port with colon, like mail.company.com:125.

type **None** - Secure connection type. Available values: *None, SSL*. Note, that SMTP server with SSL usually listen for incoming connections at port 465.

this user - Login for SMTP server, may be blank.

this password - Password for SMTP server, may be blank.

Do not convert - Specifies whether to convert message body to plain text or not.

Examples:

Use SMTP server exchange.local:125, secure connection type None, login this login and password this password. Do not convert message body to plain text.

then Send e-mail to gorlach@exchange.local with Normal importance, blind carbon copy to this address, use noreply@exchange.local as sender and include a file C:\temp\report.pdf as attachment.

9. Images activities

9.1 Convert Image

Convert image at specified url to JPG format and store URL to converted image in Variable: ConvertedImageUrl. Overwrite existing image.

This workflow action converts image to one of supported output formats: Bitmap Picture (bmp), GIF, JPG, PNG, TGA, TIFF.

Output file will have the same name and the extension according your selection in the second parameter, and will be placed in the same document library or folder where the source file is stored.

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

specified url - Full URL of the picture in Document Library.

JPG - One of supported output formats: Bitmap Picture (bmp), GIF, JPG, PNG, TGA, TIFF.

Variable: ConvertedImageUrl - Output string variable with the URL of a new picture.

Overwrite - Choose overwrite the file or do not, if the document library already contains a file with the same name as output file.

Examples:

If image at Current Item:Encoded Absoulte URL is supported

If image at Current Item:Encoded Absoulte URL uses compression

Log Conversion is not required to the workflow history list

else

Convert image at Current Item:Encoded Absoulte URL to JPG format and store URL of converted image in Variable: ConvertedImageUrl. Overwrite existing image.

then Delete item at Current Item:Encoded Absoulte URL list item url.

This workflow automatically converts uncompressed images to JPG format.

9.2 Convert Images in Folder

Convert images in folder at this url to JPG format and store converted images in specified url. Overwrite existing image. Do not delete source images

This workflow action is used to convert all images of supported formats in specified Document Library or folder of Document Library. Source and target folders or Document Libraries may be the same.

Pay attention that source images which already have a type specified by second parameter will not be copied to the target folder. Use built-in "Copy List Item" activity if required.

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

this url - Full URL of Document Library or folder of Document Library with source files.

JPG - One of supported output formats: Bitmap Picture (bmp), GIF, JPG, PNG, TGA, TIFF.

specified url - Full URL of target Document Library or folder of Document Library to store converted images. May be the same as source URL.

Overwrite - Overwrite or do not overwrite pictures in target Document Library.

Do not delete - Delete or do not delete source images.

9.3 Convert Images in Library

Convert images in this library to JPG format and store converted images in specified library. Overwrite existing image. Do not delete source images

This workflow action is used to convert all images of supported formats in specified Document Library. Source and target Document Libraries may be the same.

Pay attention that source images which already have the type specified by the second parameter will not be copied to target Document Library. Use built-in "Copy List Item" activity if required.

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

this library - Source Document Library.

JPG - One from supported output formats: Bitmap Picture (bmp), GIF, JPG, PNG, TGA, TIFF.

specified library - Target Document Library. Note that source and target Document Libraries may be the same.

Overwrite - Overwrite or do not overwrite pictures in target Document Library.

Do not delete - Delete or do not delete source images.

9.4 Flip Image

Flip image at specified url horizontally.

This workflow action is used to flip image in one from supported formats horizontally or vertically.

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

specified url - URL of the image in Document Library.

horizontally - How to flip the image: horizontally or vertically.

9.5 Get EXIF Property from Image

Get EXIF property Title from image at this url and store in Variable: variable, raw value in Variable: variable1.

This workflow action is used to get EXIF property from an image in one from supported formats. With this action you can get the following properties:

- Title
- Camera Manufacturer
- Camera Model
- Exposure Time
- F number
- Date and Time
- Focal Length
- Orientation

- Exposure program
- ISO speed rating
- Flash
- Exposure mode
- Artist
- White balance
- Scene capture type
- Exposure bias
- Aperture
- Maximum lens aperture
- North or South Latitude
- East or West Longitude
- Scene type
- Software

See also: Exchangeable image file format at Wikipedia
(http://en.wikipedia.org/wiki/Exchangeable_image_file_format).

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

<u>Title</u> - Name of the EXIF property (see list above).
<u>this url</u> - URL of the picture in Document Library.
<u>Variable: variable</u> - Output string variable. Empty string if the property is not found.
<u>Variable: variable</u> - Output string variable. Empty string if the property is not found.

9.6 Get Image Size

Get size of the image in specified url and store width in Variable: ImageWidth and height in Variable: ImageHeight

This workflow action is used to get image size in pixels. In case of an unknown format of the file, this action will generate an error.

File formats supported by the Product image processing engine:

- BMP files [reading, writing]
- DDS files [reading]
- EXR files [reading, writing]
- Raw Fax G3 files [reading]
- GIF files [reading, writing]
- HDR files [reading, writing]
- ICO files [reading, writing]
- IFF files [reading]
- JNG files [reading]
- JPEG/JIF files [reading, writing]
- JPEG-2000 File Format [reading, writing]
- JPEG-2000 codestream [reading, writing]
- KOALA files [reading]
- Kodak PhotoCD files [reading]
- MNG files [reading]
- PCX files [reading]
- PBM/PGM/PPM files [reading, writing]
- PFM files [reading, writing]
- PNG files [reading, writing]
- Macintosh PICT files [reading]
- Photoshop PSD files [reading]
- RAW camera files [reading] (see note below)
- Sun RAS files [reading]

- SGI files [reading]
- TARGA files [reading, writing]
- TIFF files [reading, writing]
- WBMP files [reading, writing]
- XBM files [reading]
- XPM files [reading, writing]

Note: Product can load RAW files from digital photo cameras (CRW/CR2, NEF, RAF, DNG, MOS, KDC, DCR, etc), virtually all RAW formats are supported).

Parameters:

specified url - Full URL of the picture in Document Library.

Variable: ImageWidth - Output variable for image width in pixels.

Variable: ImageHeight - Output variable for image height in pixels.

9.7 Is Image Compressed (HarePoint)

If image at this URL uses compression

This workflow condition analyzes image file content to check whether the image is compressed or is not. In some cases, e.g. in case of TIFF images, it is impossible to determine if the image is compressed or is not only by file extension.

Parameters:

this URL - Full URL of the picture in Document Library.

Examples:

If image at Current Item:Encoded Absoulte URL is supported

If image at Current Item:Encoded Absoulte URL uses compression

Log Conversion is not required to the workflow history list

else

Convert image at Current Item:Encoded Absoulte URL to JPG format and store URL of converted image in Variable: ConvertedImageUrl. Overwrite existing image.

then Delete item at Current Item:Encoded Absoulte URL list item url.

This workflow automatically converts uncompressed images to JPG format.

9.8 Is Image Type Supported (HarePoint)

If image at this URL is supported

This workflow condition analyzes file extension and returns true if the file type is supported by the Product image processing engine.

File formats supported by the Product image processing engine:

- BMP files [reading, writing]
- DDS files [reading]
- EXR files [reading, writing]
- Raw Fax G3 files [reading]
- GIF files [reading, writing]
- HDR files [reading, writing]
- ICO files [reading, writing]
- IFF files [reading]
- JNG files [reading]
- JPEG/JIF files [reading, writing]
- JPEG-2000 File Format [reading, writing]
- JPEG-2000 codestream [reading, writing]
- KOALA files [reading]
- Kodak PhotoCD files [reading]
- MNG files [reading]
- PCX files [reading]
- PBM/PGM/PPM files [reading, writing]

- PFM files [reading, writing]
- PNG files [reading, writing]
- Macintosh PICT files [reading]
- Photoshop PSD files [reading]
- RAW camera files [reading] (see note below)
- Sun RAS files [reading]
- SGI files [reading]
- TARGA files [reading, writing]
- TIFF files [reading, writing]
- WBMP files [reading, writing]
- XBM files [reading]
- XPM files [reading, writing]

Note: Product can load RAW files from digital photo cameras (CRW/CR2, NEF, RAF, DNG, MOS, KDC, DCR, etc), virtually all RAW formats are supported).

Parameters:

this URL - Full URL of the picture in Document Library.

Examples:

If image at Current Item:Encoded Absoulte URL is supported

If image at Current Item:Encoded Absoulte URL uses compression

Log Conversion is not required to the workflow history list

else

Convert image at Current Item:Encoded Absoulte URL to JPGformat and store URL of converted image in Variable: ConvertedImageUrl. Overwrite existing image.

then Delete item at Current Item:Encoded Absoulte URL list item url.

This workflow automatically converts uncompressed images to JPG format.

9.9 Resize Image

Resize image at specified url. Set width: 0, height: 0 (keep aspect ratio: yes) and append 'resize' to filename. Do not process image if it is smaller than target.

This workflow action is used to resize image. With this action you can easily avoid huge images from modern digital cameras in SharePoint galleries.

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

specified url - Full URL of the picture in Document Library.

width 0 - Maximum width of converted picture, in pixels.

height 0 - Maximum height of converted picture, in pixels.

keep aspect ratio - Yes or no. In the first case, the output picture will have one dimension exactly as specified by width/height parameters, and another dimension may be smaller than the one specified by width/height.

append 'resize' to filename - To options are available: save output file with the new name with prefix, or to overwrite source file.

Do not process - Process or not process source file if it has smaller dimensions than the one specified by height and width parameters.

Examples:

If Current Item: File Type equals jpg

Resize image at Current Item: Encoded Absolute URL. Set width: 800, height: 600 (keep aspect ratio: yes) and overwrite existing file. Do not process image if it is smaller than target.

9.10 Resize Images in Folder

Resize images in folder at this url. Set width: 0, height: 0 (keep aspect ratio: yes). Store resized images in folder at specified url and append 'resize' to filename. Do not process image if it is smaller than target. Do not delete source images

This workflow action is used to resize all images of supported formats at Document Library or folder of Document Library.

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

this url - URL of source Document Library or folder.

width 0 - Maximum width of converted picture, in pixels.

height 0 - Maximum height of converted picture, in pixels.

keep aspect ratio - Yes or no. In the first case, the output picture will have one dimension exactly as specified by width/height parameters, and another dimension may be smaller than the one specified by width/height.

specified url - URL of the target Document Library or folder. Source and target Document Libraries may be the same.

append 'resize' to filename - To options are available: save output file with the new name with prefix, or to overwrite source file in case the source and the target Document Libraries are the same.

Do not process - Process or do not process, or just copy source file to the target folder if it has smaller dimensions than specified by height and width parameters.

Do not delete - Delete or not delete source images.

9.11 Resize Images in Library

Resize images in folder at this library. Set width: 0, height: 0 (keep aspect ratio: yes). Store resized images in specified library and append 'resize' to filename. Do not process image if it is smaller than target. Do not delete source images

This workflow action is used to resize all images of supported formats at Document Library or folder of Document Library.

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

this library - Document Library with source images.

width 0 - Maximum width of converted picture, in pixels.

height 0 - Maximum height of converted picture, in pixels.

keep aspect ratio - Yes or no. In the first case, the output picture will have one dimension exactly as specified by width/height parameters one, and another dimension may be smaller than the specified by width/height.

specified library - Target Document Library for converted images. Source and target Document Libraries may be the same.

append 'resize' to filename - To options are available: save output file with the new name with prefix, or to overwrite source file in case the source and the target Document Libraries are the same.

Do not process - Process or not process, or just copy source file to the target folder if it has smaller dimensions than the specified by height and width parameters.

Do not delete - Delete or do not delete source images.

9.12 Rotate Image

Rotate image at specified url 90 degrees Clockwise.

This workflow action is used to rotate the image in one from supported formats to 90, 180 or 270 degrees clockwise or counter-clockwise.

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

specified url - URL of the image in Document Library.

90 - How many degrees to rotate the image: 90, 180, 270.

Clockwise - How to rotate the image: clockwise or counter-clockwise.

9.13 Set Transparent Color

Set #FF00FF color of the image at specified url transparent, convert to PNG.

This workflow action is used to set transparent color for an image in one of supported formats. Image with the transparency can be saved in PNG or GIF formats only.

The transparent color is a property of the image, transparency is not a property of the color. Once you have set a color to be the transparent color, any regions of the image in that color that were drawn previously will be transparent.

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

#FF00FF - Color to mark as transparent. You can use hexadecimal RGB values (like #RRGGBB) or the name of system-defined color from this list: [http://msdn.microsoft.com/en-us/library/system.drawing.knowncolor\(VS.90\).aspx](http://msdn.microsoft.com/en-us/library/system.drawing.knowncolor(VS.90).aspx)

specified url - URL of the image in Document Library.

PNG - Output format for the image with transparency: PNG or GIF.

10. Lists activities

10.1 Approve Item at Url

Approve **this list item url** item with **specified** comment.

This workflow action is used to approve list item in specified by URL.

Note: To get list item URL you can use “Get List Item Url” action.

Parameters:

this list item url - Full URL of list item.

specified - Comment text.

Examples:

Get Display Form url for list item Current Item and store result in Variable: F1

Get Edit Form url for list item Current Item and store result in Variable: F2

Get Native Form url for list item Current Item and store result in Variable: F3

*then Log [%Variable: F1%] * [%Variable: F2%] * [%Variable: F3%] to the workflow history list*

then Approve [%Variable: F3%] item with Approved by workflow comment

Result workflow history list:

`http://gorlach7/Shared Documents/Forms/DispForm.aspx ID=214 * http://gorlach7/Shared Documents/Forms/EditForm.aspx ID=214 * http://gorlach7/Shared Documents/2144750538 Doc064.pdf`

10.2 Calculate Aggregation Function

Calculate SUM on this field from all items in this list and store result in Variable:variable1

This workflow action is used to calculate SUM, COUNT, AVG, MIN or MAX aggregation function on list items field values. Note that empty field values are skipped by all these functions including COUNT.

Aggregation functions and supported field formats:

- SUM, calculate summarized value: Number, Integer, String.
- COUNT, count non-empty items: Any format.
- AVG, calculate average value: Number, Integer.
- MIN, seeking the minimum value: Number, Integer, Date/Time, String.
- MAX, seeking the maximum value: Number, Integer, Date/Time, String.

The third parameter of this action contains multi-line string with list items filter in one from four available formats:

1. Default empty string means 'all items'.
2. List of GUIDs of list items, each line contains one GUID (value of GUID property of list item).
3. List of case-insensitive masks (with * and wildcards) for names of list items or list items names (value of Name property of list item). Each line contains one mask or name.
4. Filter in the CAML (Collaborative Application Markup Language) Query format, see below.

CAML format is a bit complicated, but very flexible. It is based on XML. The example of correct CAML filter, it is a body of CAML Query element:

<Where>

<And>

<Eq>

<FieldRef Name='FirstName' />

<Value Type='Text'>Alexey</Value>

</Eq>

<Contains>

<FieldRef Name='Email' />

```

    <Value Type='Text'>@gmail.com</Value>

  </Contains>

</And>

</Where>

```

Note, that you should use internal names of fields in the CALM Query.

For the CAML reference, see:

- Writing CAML queries: <http://sharepointmagazine.net/articles/writing-caml-queries-for-retrieving-list-items-from-a-sharepoint-list>
- CAML Query Schema: <http://msdn.microsoft.com/en-us/library/ms467521.aspx>
- CAML Query Element: <http://msdn.microsoft.com/en-us/library/ms471093.aspx>
- Freeware tool to build CAML
Query: <http://www.u2u.be/Tools/wincamlquerybuilder/CamlQueryBuilder.aspx>

Parameters:

SUM - Aggregation function. One from following options: *SUM*, *COUNT*, *AVG*, *MIN*, *MAX*.

this field - Field name.

all - Items filter. All items by default, or the list of items names, or the list of list items GUIDs, or the CAML query.

this list - SharePoint list.

Variable:variable1 - Output variable to store result. Note that type of output variable should match with the type of result. E.g. attempt to store Date/Time result to Number variable will cause an exception.

10.3 Copy InfoPath Attachments to Document Library

Copy Info Path attachment specified by XPATH of list item at this url to document library at specified url. Overwrite existing items.

This workflow action is used to copy InfoPath attachments of specified list item to document library.

Parameters:

specified by XPATH - XPath query string. See examples section for details.

this url - Full URL of list item.

specified url - Full URL of target document library.

Overwrite - If the document library already contains file with the same name, you can choose to *Overwrite* it or to skip this file (*Do not overwrite*).

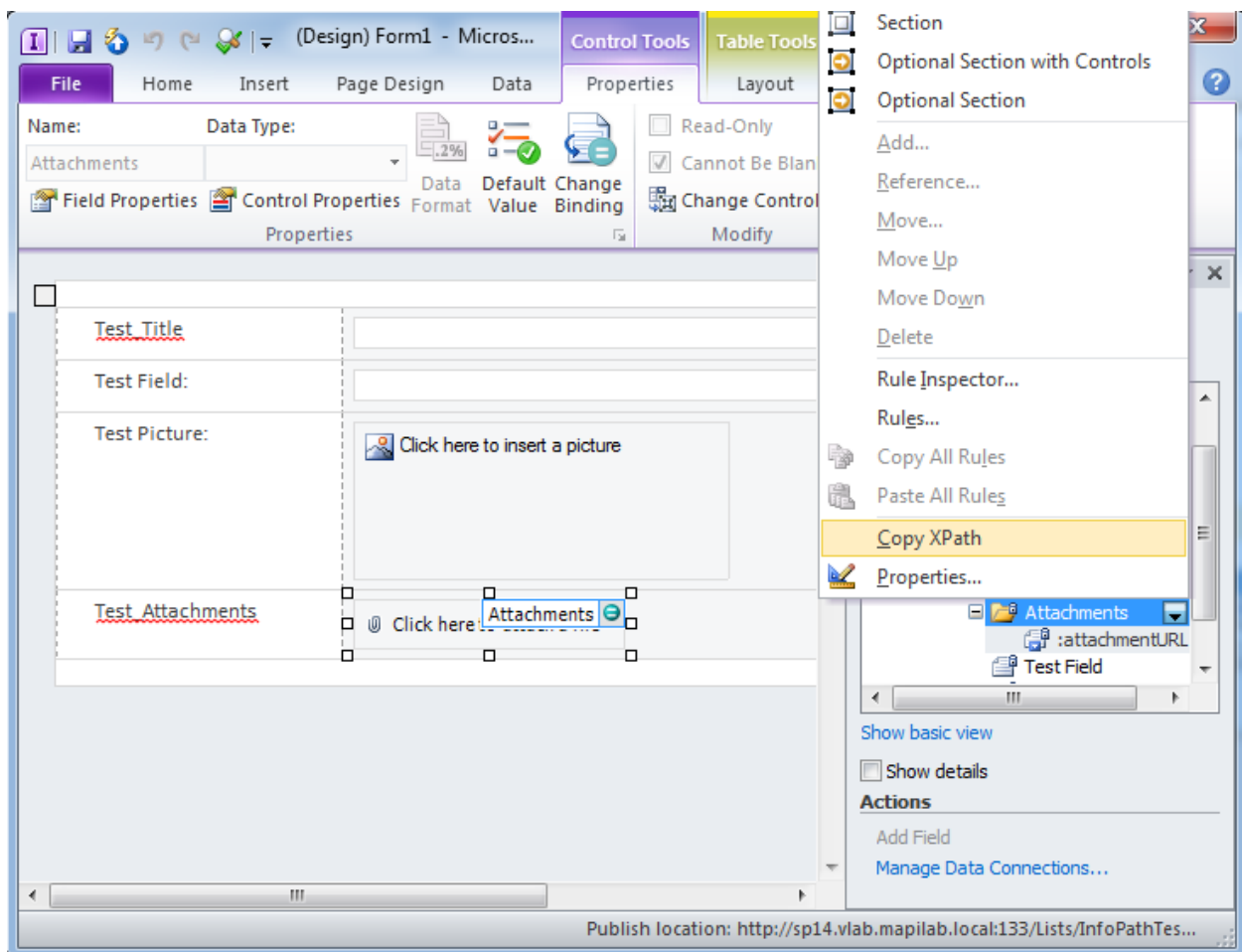
Examples:

Copy Info Path attachment /dfs:myFields/dfs:dataFields/my:SharePointListItem_RW/my:Attachments of list item at Variable: listItemUrl to document library at Variable: docLibUrl. Overwrite existing items.

You can combine multiple XPath queries using vertical bar ('|') sign, e.g.

`/my:myFields/my:MoreAttachments | /my:myFields/my:AttachFile`.

You can retrieve XPath string using Microsoft InfoPath Designer:



10.4 Copy List Item Attachments to Document Library

Copy attachments of list item at this url to document library at specified url. Overwrite existing items.

This workflow action is used to copy attachments from specified list item to document library specified by URL.

See Also: "Upload Files to List Item Attachments" action.

Parameters:

this url - Full URL of list item.

specified url - Full URL of target document library.

Overwrite - If the document library already contains file with the same name, you can choose to *Overwrite* it or to skip this file (*Do not overwrite*).

10.5 Copy List Item Attachments to List Item

Copy attachments of list item at this url to list item at specified url. Overwrite existing items.

This workflow action is used to copy attachments of one list item to attachments of another list item.

See Also: "Upload Files to List Item Attachments" action.

Parameters:

this url - Full URL of source list item.

specified url - Full URL of target list item.

Overwrite - If target item already contains attachment with the same name, then you can choose one from two options. *Overwrite* - to replace file by new one. *Do not overwrite* - to skip this file.

10.6 Create List

Create list with this title and this description based on template with this name. Store list url to Variable: variable.

Creates new SharePoint list or document library using specified template.

Parameters:

<u>this title</u> - Title of new list.
<u>this description</u> - Description of the new list.
<u>this name</u> - Name of the list of document library template.
<u>Variable: variable</u> - This variable will contain the full URL of the created list or document library

10.7 Create List Item at Url

Create list item with specified values at this list url. Store new list item url in Variable:variable1. Trigger events.

This workflow action is used to create new list item and set the values of list item fields. Values are specified as multi-line string of one from two available formats.

First, the format in style of INI file. For example:

Position=Team Leader

First Name=John

Last Name=Smith

Age=23

Second, is XML format, like:

<item>

<field title="Position">Team Leader</Field>

<field title="First Name">John</Field>

<field title="Last Name">Smith</Field>

<field title="Age">23</Field>

</item>

See Also: "Update List Item at Url" action.

Parameters:

specified values - Values for list item fields in INI or XML format.

this list url - Full URL of list.

Variable: variable1 - Output string variable for the URL of created list item.

Trigger - Specifies whether to trigger an item create event.

10.8 Delete All List Items

Delete all list items. Store number of deleted items in Variable: variable.

This workflow action is used to delete list items of current list.

The first parameter of this action contains multi-line string with list items filter in one from four available formats:

1. Default empty string means 'all items'.
2. List of GUIDs of list items, each line contains one GUID (value of GUID property of list item).
3. List of case-insensitive masks (with * and wildcards) for names of list items or list items names (value of Name property of list item). Each line contains one mask or name.
4. Filter in the CAML (Collaborative Application Markup Language) Query format, see below.

CAML format is a bit complicated, but very flexible. It is based on XML. The example of correct CAML filter, it is a body of CAML Query element:

<Where>

<And>

```

<Eq>

  <FieldRef Name='FirstName' />

  <Value Type='Text'>Alexey</Value>

</Eq>

<Contains>

  <FieldRef Name='Email' />

  <Value Type='Text'>@gmail.com</Value>

</Contains>

</And>

</Where>

```

Note, that you should use internal names of fields in the CALM Query.

For the CAML reference, see:

- Writing CAML queries: <http://sharepointmagazine.net/articles/writing-caml-queries-for-retrieving-list-items-from-a-sharepoint-list>
- CAML Query Schema: <http://msdn.microsoft.com/en-us/library/ms467521.aspx>
- CAML Query Element: <http://msdn.microsoft.com/en-us/library/ms471093.aspx>
- Freeware tool to build CAML Query: <http://www.u2u.be/Tools/wincamlquerybuilder/CamlQueryBuilder.aspx>

Parameters:

all - Items filter. All items by default, or the list of items names, or the list of list items GUIDs, or the CAML query.

Variable: variable - Output integer variable to store number of deleted items.

10.9 Delete Item at Url

Delete item at specified list item url.

This workflow action is used to delete list item specified by URL.

Note: To get list item URL you can use “Get List Item Url” action.

Parameters:

specified - Full URL of list item.

10.10 Download InfoPath Attachments to Folder

Download Info Path attachment specified by XPATH of list item at this url to this folder. Overwrite existing items.

This workflow action is used to save InfoPath attachments of specified list item to local disk or to network share.

Note: In order to specify file access credentials, add the "Set Network File Access" activity to the same workflow step first.

Parameters:

specified by XPATH - XPath query string. See "Copy InfoPath Attachments to Document Library" action examples section for details.

this url - Full URL of list item.

this folder - Local disk or network path.

Overwrite - If the document library already contains file with the same name, you can choose to *Overwrite* it or to skip this file (*Do not overwrite*).

10.11 Find List Item at Url

Find item in the list at this url where item's field equals value. Store result urls in Variable: array.

This workflow action is used to find list items by the value of the specified field.

The list of supported operators and types:

Operator	String	Integer	Date/Time
equals	+, case insensitive	+	+
not equals	+, case insensitive	+	+
is greater than	-	+	+
is greater than or equals to	-	+	+
is less than	-	+	+
is less than or equals to	-	+	+
begins with	+, case insensitive	-	-
ends with	+, case insensitive	-	-
contains	+, case insensitive	-	-
doesn't contain	+, case insensitive	-	-
is empty	+	+	+
is not empty	+	+	+
is in array	+, case insensitive	+	-

Parameters:

this url - Full URL of list

field - The name of the field to search for.

equals - Comparison operator.

value - Value to search for.

Variable: array - Output array variable with URLs of list items.

Examples:

Find item in the list at Variable: listUrl where item's Company equals MAPILab Ltd. Store result urls in Variable: array.

10.12 Find List Item by Caml

Find item in the list at this url by this CAML query. Store result urls in Variable: array.

This workflow action is used to find list items using CAML query.

See "Calculate Aggregation Function" activity description for examples and references on CAML queries.

See Also: "Find List Item at Url" activity.

Parameters:

this url - Full URL of list.

this CAML query - String with the CAML query.

Variable: array - Output array variable with URLs of list items.

Examples:

Find item in the list at Variable: listUrl by <Where><And><Eq><FieldRef>.... Store result urls in Variable: array.

10.13 Get Field Value at Url

Get value from this field of list item at this url and store result in Variable: variable.

This workflow action is used to get a field value from list item specified by URL.

Parameters:

this field - Field name.

this url - Full URL of list item.

Variable: variable - Output variable for the field value.

10.14 Get List Item Attachment Content

Get content of the attachment with this name at this list item url and store in Variable: variable.

This workflow action is used to retrieve contents of specified list item attachment into output variable.

Parameters:

with this name - Name of the attachment.

this list item url - Full URL of list item.

Variable: variable - Output variable.

Examples:

Get content of the attachment OriginalMessage.eml at Current Item: Encoded Absolute URL and store in Variable: eml.

then Get sender email from Variable: eml body and store in Variable: Sender Address.

then Get reply to from Variable: eml body and store in Variable: Reply To.

the Update list item with Sender Address=[%Variable: Sender Add... at Current Item: Encoded Absolute URL list item url. Do not trigger events.

10.15 Get List Item Attachments

Get Names of attachments of list item at this url and store to Variable: ArrayVariable.

This workflow action is used to get names, file sizes or URLs of list item attachments.

See Also: "Join Text" and "Get Array Items Count" activities.

Parameters:

Names - One from three options: *Names*, *URLs* or *File Sizes*.

this - Full URL of list item.

Variable: ArrayVariable - Output string array with names or URLs, or number array with file sizes.

Examples:

Get Names of attachments of list item at Current Item:Encoded Absolute URL url and store to Variable: namesArray.

then Get the number of items in Variable: namesArray and store in Variable: attachmentsCount.

then Log Variable: attachmentsCount to the workflow history list

10.16 Get List Item Url

Get Display Form url for list item this list item and store result in Variable:variable1.

This workflow action is used to get URL of specified list item, or URL of specified list item edit form or URL of specified list item view form.

Parameters:

Display Form - One from three options: *Display Form*, *Edit Form*, *Native Form*. In the case of documents library, *Native Form* is the URL of file.

this list item - List item, e.g. Current Item.

Variable:variable1 - Output string variable.

Examples:

Get Display Form url for list item Current Item and store result in Variable: F1

Get Edit Form url for list item Current Item and store result in Variable: F2

Get Native Form url for list item Current Item and store result in Variable: F3

then Log [%Variable: F1%] * [%Variable: F2%] * [%Variable: F3%] to the workflow history list

then Approve [%Variable: F3%] item with Approved by workflow comment

Result workflow history list:

http://gorlach7/Shared Documents/Forms/DispForm.aspx ID=214 * http://gorlach7/Shared Documents/Forms/EditForm.aspx ID=214 * http://gorlach7/Shared Documents/2144750538 Doc064.pdf

10.17 Get List Url

Get Display Form url of this list and store to Variable: variable.

This workflow action is used to get one from seven possible URL types for a list on current SharePoint site:

- Display Form
- Edit Form
- New Form
- Native Form
- Mobile Display Form
- Mobile Edit Form
- Mobile New Form

Note: Most workflow actions are works with Native Form of URL.

Parameters:

Display Form - Type of URL to return.

this - List on current site.

Variable: variable - Output string variable.

10.18 Is Current Item Checked Out (HarePoint)

If current item is checked out

This workflow condition returns true if the current item is checked out.

10.19 Is Specified Item Checked Out (HarePoint)

If item at specified url is checked out

This workflow condition returns true if the item specified by URL is checked out.

Parameters:

specified - Full URL of list item.

10.20 Reject Item at Url

Reject list item at this url item with specified comment.

This workflow action is used to reject items in specified list.

Note: To get list item URL you can use “Get List Item Url” action.

Parameters:

this - Full URL of list item.

specified - Comment text.

10.21 Remove All Attachments

Remove all attachments from list item at this url. Store number of removed attachments to Variable: variable.

This workflow action is used to remove attachments by name or by file mask from the list item specified by URL.

Parameters:

all - Multi-line string with one attachment file name or file mask at line.

this url - Full URL of list item.

Variable: variable - Output integer variable to store number of removed attachments.

10.22 Remove InfoPath Attachments

Remove Info Path attachment specified by XPATH of list item at this url.

This workflow action is used to remove InfoPath attachments from the list item specified by URL.

Parameters:

specified by XPATH - XPath query string. See "Copy InfoPath Attachments to Document Library" action examples section for details.

this url - Full URL of list item.

10.23 Update List Item at Url

Update list item with specified values at this list item url. Trigger events.

This workflow action is used to set values of list item fields. Values are specified as multi-line string of one from two formats.

First, the format in style of INI file. For example:

Position=Team Leader

First Name=John

Last Name=Smith

Age=23

Second, is XML format, like:

```
<item>

<field title="Position">Team Leader</Field>

<field title="First Name">John</Field>

<field title="Last Name">Smith</Field>

<field title="Age">23</Field>

</item>
```

See Also: "Create List Item" action.

Parameters:

<u>specified values</u> - Values for list item fields in INI or XML format.
--

<u>this</u> - Full URL of list item.

<u>Trigger</u> - Specifies whether to trigger an item update event.
--

10.24 Upload Files to List Item Attachments

Upload files from <u>those paths</u> to list item at <u>this url</u> . <u>Overwrite</u> existing items. Store number of uploaded files to <u>Variable: variable</u> .

This workflow action is used to upload files from local disk, network share, document library or web server to list item attachments.

Note: In order to specify file access credentials, add the "Set Network File Access" activity to the same workflow step first.

Parameters:

<u>those paths</u> - Multi-line string where each line contains a path with file name or URL. Local disk or network share paths can contain masks in file name (note, that recursive search in subfolders is not supported).

<u>this url</u> - Full URL of list item.

Overwrite - What to do with existing attachments with the same name. *Not overwrite* means skip existing attachments.

Variable: variable - Output integer variable with the number of uploaded files.

11. Network activities

11.1 Create Directory on FTP Server

Create specified directory in this folder on FTP server.

This workflow action is used to create directory on FTP server.

To specify server name, login and password place "Set FTP Server Options" activity the current workflow step first.

Parameters:

specified - Directory name on FTP server.

this - Folder name to create.

Examples:

See example in "Set FTP Server Options" activity description.

11.2 Fetch RSS Feed to Html

Fetch RSS from this url print by `<p>[DATE]: <a href="[URL]">[TITLE]</a></p><p>[DESCRIPTION]</p>` template and store in Variable: OutputHtml

This workflow action is used to fetch all items from RSS feed and print it by specified template.

Parameters:

this url - URL of RSS feed.

template - Printing template for RSS item. [DATE], [URL], [TITLE] and [DESCRIPTION] in template will be replaced by corresponding RSS item fields.

OutputHtml - Output string variable, will contains concatenated string with all feed items.

Examples:

Fetch RSS from http://www.zdnet.com/search t=7&mode=rss&tag=mantle_skin;content print by `<p>[DATE]: <a href="[URL]">[TITLE]</a></p><p>[DESCRIPTION]</p>` template and store in Variable: html

then Create document from `<html><body>[%Variable: html%]</body></html>` in <http://site/Shared Documents/news.html>. Overwrite existing document.

11.3 Fetch RSS Item

Fetch RSS from this url and store top item number 1's title in Variable: Title, link in Variable: Link, description Variable: Description and publication date in Variable: Date.

This workflow action is used to fetch top item from RSS feed.

Parameters:

this url - RSS feed address.

item number 1's - Number of item to fetch.

Variable: Title - Output string variable for RSS item title.

Variable: Link - Output string variable for RSS item link.

Variable: Description - Output string variable for RSS item text.

Variable: Date - Output date variable for RSS item publication date.

11.4 Send HTTP GET Request

Send HTTP request to this url using GET method with this headers. Store response in Variable: variable1 and status code in Variable: variable2.

This workflow action is used to send HTTP GET request.

Note: In order to specify HTTP request credentials, add the "*Set HTTP Request Options*" activity to the same workflow step first.

Parameters:

this url - Target URL with parameters.

this headers - Additional request headers each on new line. For example: *Accept-Language: fr*

Variable: variable1 - Output string variable to receive remote server response, usually contains requested web page body.

Variable: variable2 - Output integer variable for HTTP status code (example: 200).

Examples:

Encode *[%Current Item:Approver Comments%]* using *url encoder* and store in *Variable: param.*

then Send HTTP request to *http://mysite/ q=[%Variable: param%]* using *GET* method with *this* headers. Store response in *Variable: pageBody* and status code in *Variable: code.*

11.5 Send HTTP POST Request

Send HTTP request to this url using POST method with this headers and this request. Store response in Variable: variable1 and status code in Variable: variable2.

This workflow action is used to send HTTP POST request.

Note: In order to specify HTTP request credentials, add the "Set HTTP Request Options" activity to the same workflow step first.

Parameters:

this url - Target URL.

this headers - Additional request headers each on new line. For example: *Accept-Language: fr*

this request - The body of the request. For example: *name=john&age=23*

Variable: variable1 - Output string variable to receive remote server response, usually contains requested web page body.

Variable: variable2 - Output integer variable for HTTP status code (example: 200).

Examples:

Encode *[%Current Item:Approver Comments%]* using *url encoder* and store in *Variable: param.*

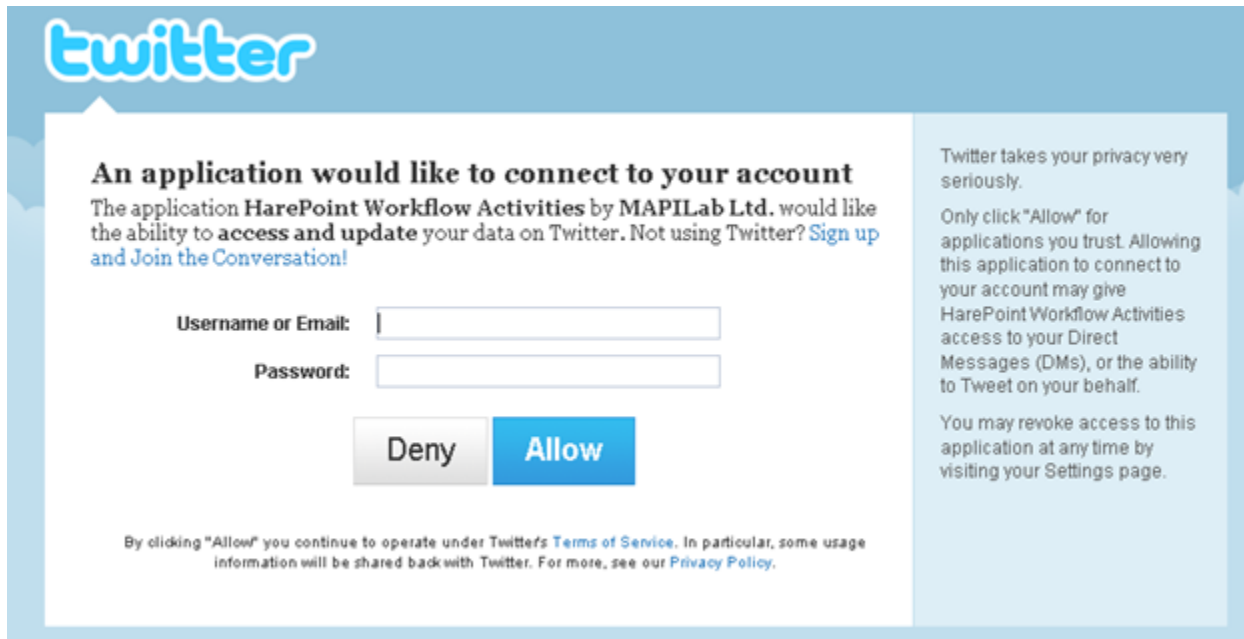
then Send HTTP request to *http://mysite/script.aspx* using *POST* method with *this* headers and *query=[%Variable: param%]* request. Store response in *Variable: pageBody* and status code in *Variable: code.*

11.6 Send Twitter Message

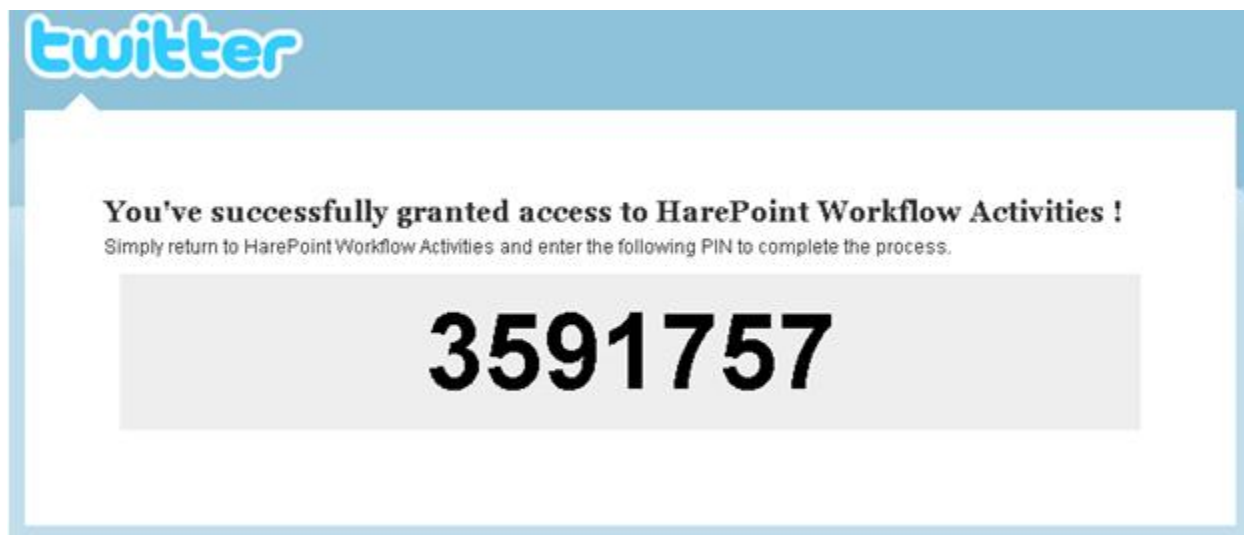
Send message with specified twitter authorization token.

This workflow action is used to send message to twitter. To set up this action, you must create authorization token with "Twitter Authorization Tool" placed in the program folder in Start menu.

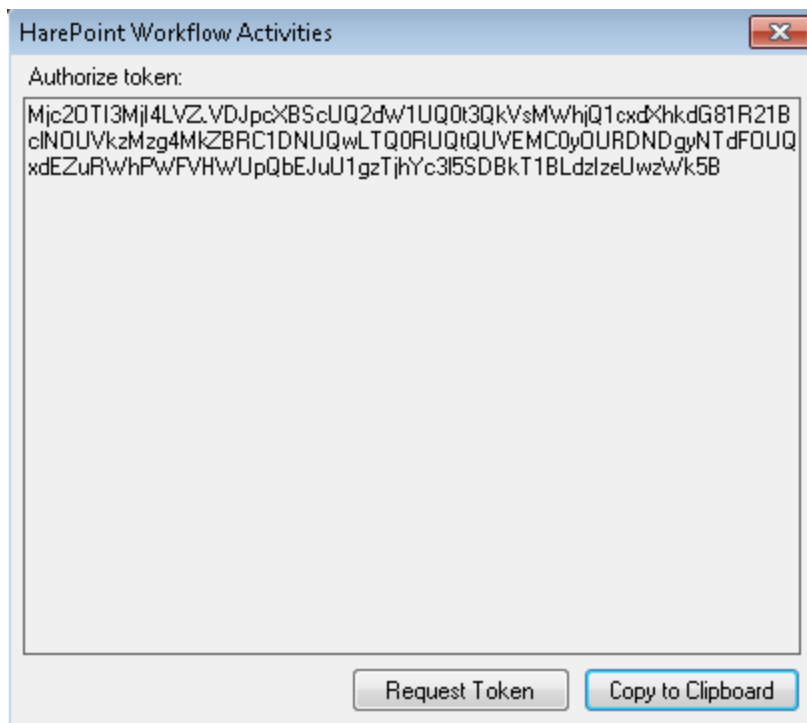
Run the tool, and click "Request Token" button. After that, Twitter web-page will be opened:

A screenshot of the Twitter authorization page. The Twitter logo is at the top left. The main heading is "An application would like to connect to your account". Below it, it says "The application HarePoint Workflow Activities by MAPILab Ltd. would like the ability to access and update your data on Twitter. Not using Twitter? [Sign up](#) and [Join the Conversation!](#)". There are two input fields: "Username or Email:" and "Password:". Below these are two buttons: "Deny" and "Allow". To the right, there is a privacy notice: "Twitter takes your privacy very seriously. Only click 'Allow' for applications you trust. Allowing this application to connect to your account may give HarePoint Workflow Activities access to your Direct Messages (DMs), or the ability to Tweet on your behalf. You may revoke access to this application at any time by visiting your Settings page." At the bottom, there is a small disclaimer: "By clicking 'Allow' you continue to operate under Twitter's [Terms of Service](#). In particular, some usage information will be shared back with Twitter. For more, see our [Privacy Policy](#)."

Enter your credentials and press Allow button on the web page to get the PIN number:

A screenshot of the Twitter PIN confirmation page. The Twitter logo is at the top left. The main heading is "You've successfully granted access to HarePoint Workflow Activities !". Below it, it says "Simply return to HarePoint Workflow Activities and enter the following PIN to complete the process." In the center, there is a large grey box containing the PIN number "3591757".

Enter the PIN number to "Twitter Authorization Tool" and get the authorization token:



Press "Copy to Clipboard" button and paste authorization token to the second parameter of this action.

Parameters:

message - Message text. Maximum length is 140 characters, longer messages will be cut off automatically.

specified - Twitter authorization token generated with "Twitter Authorization Tool".

11.7 Send XMPP Instant Message

Send XMPP instant Message to this JID by using JID: this JID, password: this password and optional server: this server.

This workflow action is used to sent instant message to XMPP-compatible messaging clients like Google Talk.

List of compatible clients is available here: <http://xmpp.org/xmpp-software/clients/>

Parameters:

Message - Message text

to this JID - Receiver's JID

using JID: **this JID** - Sender's JID

this password - Sender's password

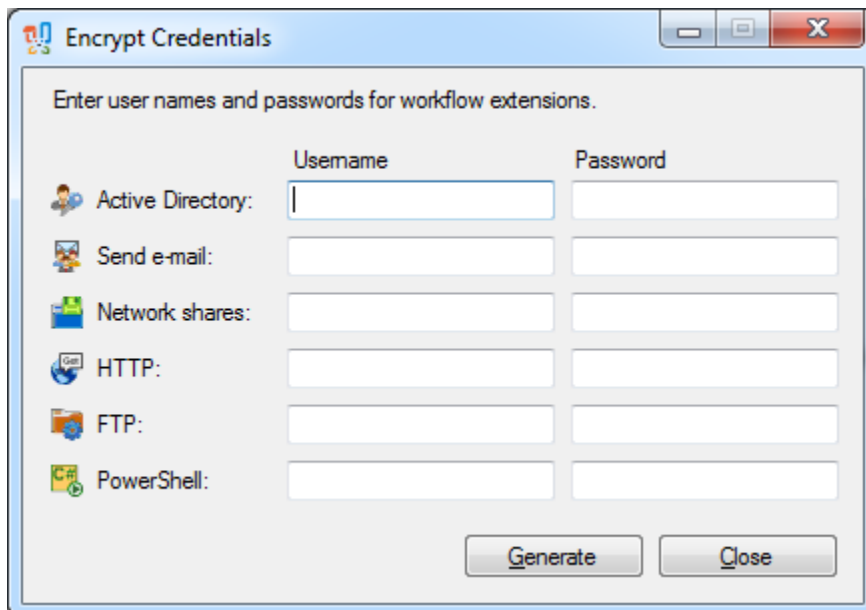
this server - Network host to logging on

11.8 Set Encrypted Passwords

Set encrypted passwords with **this** string.

This workflow action is used to setup authorization settings in encrypted form. This action can be used instead of any other action that sets the usernames and passwords.

To generate encrypted string for this action use "Encrypt Credentials" tool:



Parameters:

this - Base64 string generated by the "Encrypt Credentials" tool.

11.9 Set FTP Server Options

Use **specified** FTP server, **disable** SSL, login **this** user and **this** password.

This workflow action is used to set FTP server name, login, password and other options for other FTP-related activities like "Upload Document to FTP Server" or "Create Directory on FTP Server".

This activity must be placed above all other FTP-related activities in the step. You can use this activity several times in one workflow step to use different FTP servers.

Note: The password will be stored as a plain text in the workflow XOML file and may be accessed by the URL like `http://sharepoint/Workflows/workflowname/workflowname.xoml` or by SharePoint Designer.

Parameters:

specified - FTP server name or IP-address. By default port 21 is used, you can specify port with colon, like `ftp.company.com:121`.

disable - Disable or enable SSL. Note, that FTP servers with SSL usually listen for incoming connections at port 990.

login - FTP server login. If empty, *Anonymous* user name is used.

password - FTP server password. May be empty.

Examples:

Use `ftp.company.com:990` FTP server, enable SSL, login `root` user and `hU2BnkM13` password.

then Create mydir directory in web/temp folder on FTP server.

then Upload Current Item document to FTP server into web/temp/mydir folder. If file already exists upload with unique name.

then Use 10.1.1.1 FTP server, disable SSL, login this user and this password.

then Upload Tasks list item attachments to FTP server into incoming folder. If the file already exists do nothing.

In the example above, in the last activity, when choosing Tasks list item, specify searching 'Attachments=Yes' in the "Choose List Item" window. Unfortunately, SharePoint designer does not visualize searching options in workflow window.

11.10 Set HTTP Request Options

Use this user and this password to authenticate against website.

This workflow action is used to set HTTP login and password for other HTTP-related activities like "Send HTTP GET Request".

Note: The password will be stored as a plain text in the workflow XOML file and may be accessed by the URL like *http://sharepoint/Workflows/workflowname/workflowname.xoml* or by SharePoint Designer.

Parameters:

this user - User name that will be used for HTTP access.

this password - Password that will be used for Active Directory access (see Note).

11.11 Upload Document to FTP Server

Upload specified document to FTP server into this folder. If file already exists do nothing.

This workflow action is used to upload list item on FTP server.

To specify server name, login and password place "Set FTP Server Options" activity to current workflow step first.

Parameters:

specified - List item to upload.

this - FTP directory name.

do nothing - What to do if the file already exists. Four options are available: *do nothing*, *treat as an error* - generate error and stop workflow, *upload with unique name* - add unique identifier like {11fba226-59d6-4f31-b9c4-e43ddc67e048} to the end of file name, or *overwrite* existing file.

Examples:

See example in "Set FTP Server Options" activity description.

11.12 Upload List Item Attachments to FTP Server

Upload specified list item attachments to FTP server into this folder. If the file already exists do nothing.

This workflow action is used to upload list item attachments to FTP server directory.

To specify server name, login and password place "Set FTP Server Options" activity to current workflow step first.

Parameters:

specified - List item whose attachments will be uploaded.

this - Directory name on FTP server.

do nothing - What to do if the file already exists. Four options are available: *do nothing*, *treat as an error* - generate error and stop workflow, *upload with unique name* - add unique identifier like {11fba226-59d6-4f31-b9c4-e43ddc67e048} to the end of file name, or *overwrite* existing file.

Examples:

See example in "Set FTP Server Options" activity description.

12. Office activities

12.1 Find Text in Word Document

Find this text in this .docx document, match case: no. Store result in Variable: variable.

This workflow action is used to check presence of specified substring at Microsoft Word document.

In this version, only DOCX documents are supported.

Note: Word Automation Services are not required for this workflow action.

Parameters:

this text - Input string to find in Word document.

this - Reference to document, e.g. Current Item.

no - Is the search case-sensitive.

Variable: variable - Output boolean variable.

12.2 Get Cell Value

Get value of this cell of specified sheet in this document and store in Variable: CellValue.

This workflow action is used to get cell value at Microsoft Excel document at SharePoint server.

Note: This workflow action requires Excel Services installed on the server.

Parameters:

this - Cell address, e.g. A1.

specified - Name of the sheet, e.g. Sheet1.

this - Reference to XLS or XLSX document, e.g. Current Item.

Variable: CellValue - Output string variable.

12.3 Get Office Document Property

Get this property of this office document and store in Variable: variable

This workflow action is used to get the property of the document at SharePoint server. With this action, you can get the following properties:

- Title
- Author
- Keywords
- Subject
- Comments
- Last Modified By
- Category
- Manager
- Company

In the current version, only Open Office XML (Microsoft Office 2007/2010) formats are supported: DOCX, XLSX, PPTX.

See Also: Open Office XML at Wikipedia (http://en.wikipedia.org/wiki/Office_Open_XML).

Note: Word or Excel Automation Services are not required for this workflow action.

Parameters:

this property - Office document property name.

this - Reference to document, e.g. Current Item.

Variable: variable - Output string variable.

12.4 Read Word Document

Read this content control of this .docx document and store its value in Variable: variable

This workflow action is used to get text of named content control of Microsoft Word document. See 'Update Word Document' activity overview to find out how to place named content control to Microsoft Word document.

In this version, only DOCX documents are supported.

Note: Word Automation Services are not required for this workflow action.

Parameters:

this content control - Content control name, see overview section for details.

this - Reference to document, e.g. Current Item.

Variable: variable - Output string variable.

12.5 Read Word Document To Text

Read this .docx document and store its text in Variable: variable

This workflow action is used to get text of Microsoft Word document to string variable.

In this version, only DOCX documents are supported.

Note: Word Automation Services are not required for this workflow action.

Parameters:

this - Reference to document, e.g. Current Item.

Variable: variable - Output string variable.

12.6 Replace Text in Word Document

Replace all occurrences of this string in this document with this text, match case: no.

This workflow action is used to replace text in Microsoft Word document. This action can be used for generation of bills and invoices, filling forms and addresses on envelopes within workflow.

In this version, only DOCX documents are supported.

See Also: 'Update Word Document' action.

Note: Word Automation Services are not required for this workflow action.

Parameters:

<u>all occurrences</u> - With this action it is possible to replace <i>all occurrences, first occurrence or last occurrence</i> of specified pattern.
<u>this string</u> - String to replace.
<u>this</u> - Reference to document, e.g. Current Item.
<u>this text</u> - New string.
<u>no</u> - Case sensitive: <i>yes</i> or <i>no</i> .

12.7 Set Cell Value

Set value of this cell of specified sheet in this document to this value.

This workflow action is used to set cell value at Microsoft Excel document at SharePoint server.

Note: This workflow action requires Excel Services installed on the server.

Parameters:

<u>this</u> - Cell address, e.g. A1.
<u>specified</u> - Name of the sheet, e.g. Sheet1.
<u>this</u> - Reference to XLS or XLSX document, e.g. Current Item.
<u>this value</u> - String, Integer, Number, Date/Time or Boolean variable with new cell value.

12.8 Set Office Document Property

Set this property of this office document and to this value

This workflow action is used to set the value of document property at SharePoint server. With this action, you can set the value for the following properties:

- Title
- Author
- Keywords
- Subject
- Comments
- Last Modified By
- Category
- Manager
- Company

In the current version, Open Office XML (Microsoft Office 2007/2010) formats are supported only: DOCX, XLSX, PPTX.

See Also: Open Office XML at Wikipedia (http://en.wikipedia.org/wiki/Office_Open_XML).

Note: Word or Excel Automation Services are not required for this workflow action.

Parameters:

this property - Office document property name.

this - Reference to document, e.g. Current Item.

this value - String variable with new value.

12.9 Update Word Document

Set value of this content control of this .docx document to this value

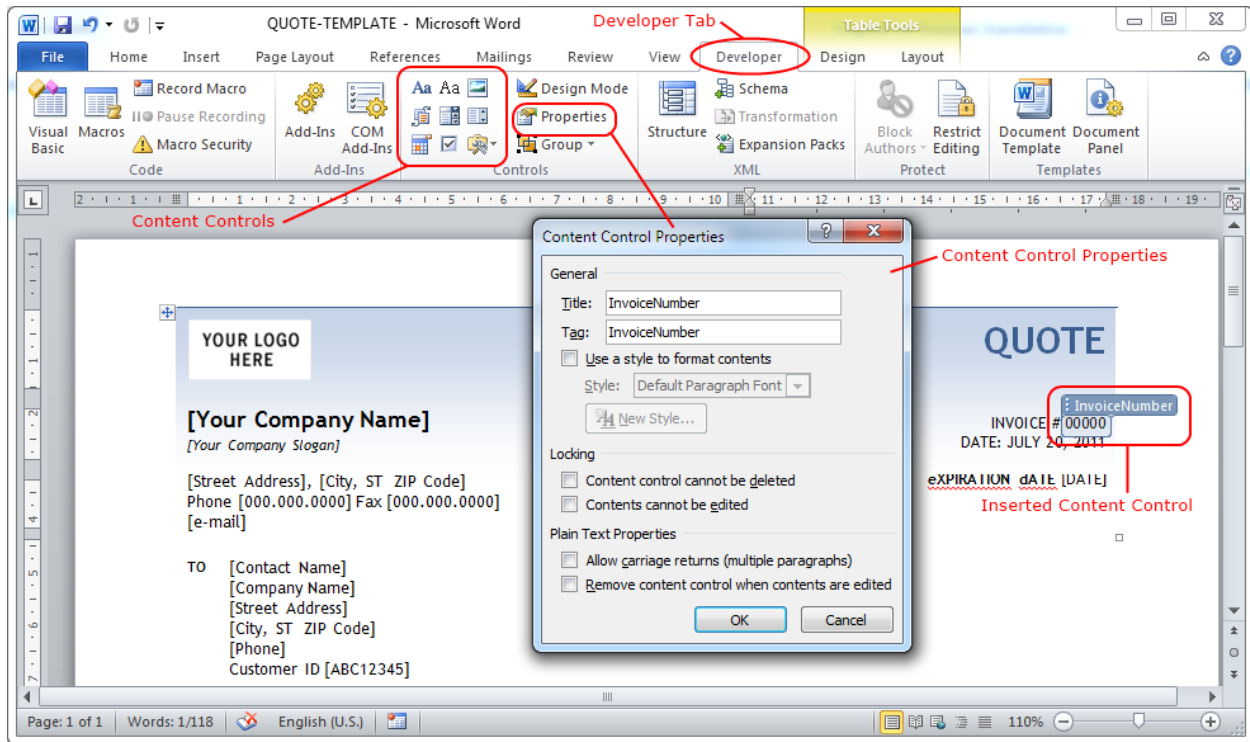
This workflow action is used to update text of named content control in Microsoft Word document. This action can be used for generation of bills and invoices, filling forms and addresses on envelopes within workflow.

To place content control, you should open Developer Tab in Microsoft Word:

1. Click the Microsoft Office Button , and then click Word Options.

2. Click Popular.
3. Select the Show Developer tab in the Ribbon check box, and then click OK.

On Developer Tab, you should select content control on the Ribbon, insert it into the document and assign name using Properties button:



See Also: 'Replace Text in Word Document' action.

Note: Word Automation Services are not required for this workflow action.

Parameters:

this content control - Content control name, see overview section for details.

this - Reference to document, e.g. Current Item.

this value - String with the new text for content control.

Examples:

Set InvoiceNumber of Current Item .docx document to 12345

This workflow replace content of "InvoiceNumber" content control with 12345. See overview sections for instructions how to place named content control to the Microsoft Word document.

13. Permissions activities

13.1 Change Permissions Inheritance at Url

Start inheriting permissions at this url

This workflow action is used to start or stop inheriting permissions for SharePoint site, list, list folder or list item specified by URL.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

Start/stop - Action to do with inheritance.

this url - Full URL of a SharePoint site, list, list folder or list item.

Examples:

Set Variable: item to [%Workflow Context:Current Item URL%].

then Stop inheriting permissions at Variable: item.

13.2 Change Permissions Inheritance for List

Start inheriting permissions for this list

This workflow action is used to start or stop inheriting of list's permissions.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

Start/stop - Action to do with inheritance.

this list - Target list.

13.3 Change Permissions Inheritance for List Item

Start inheriting permissions for this list item

This workflow action is used to start or stop inheriting of list's item permissions.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

Start/stop - Action to do with inheritance.

this list item - Target list item.

13.4 Delete All Permissions at Url

Delete all permissions at this url

This workflow action is used to delete all permissions from SharePoint site, list, list folder or list item specified by URL.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

this url - Full URL of a SharePoint site, list, list folder or list item.

13.5 Delete All Permissions from List

Delete all permissions from this list

This workflow action is used to delete all permissions from list.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

this list - Target list.

13.6 Delete All Permissions from List Item

Delete all permissions from this list item

This workflow action is used to delete all permissions from list item.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

this list item - Target list item.

13.7 Delete All User Permissions at Url

Delete all permissions at this url for this user.

This workflow action is used to delete all permissions for specified user from SharePoint site, list, list folder or list item specified by URL.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

this url - Full URL of a SharePoint site, list, list folder or list item.

this user - All permissions for this user or group will be removed.

13.8 Delete All User Permissions from List

Delete all permissions from this list for this user

This workflow action is used to delete all permissions for specified user from list.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

this list - Target list.

this user - All permissions for this user or group will be removed.

13.9 Delete All User Permissions from List Item

Delete all permissions from this list item for this user

This workflow action is used to delete all permissions for specified user from list item.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

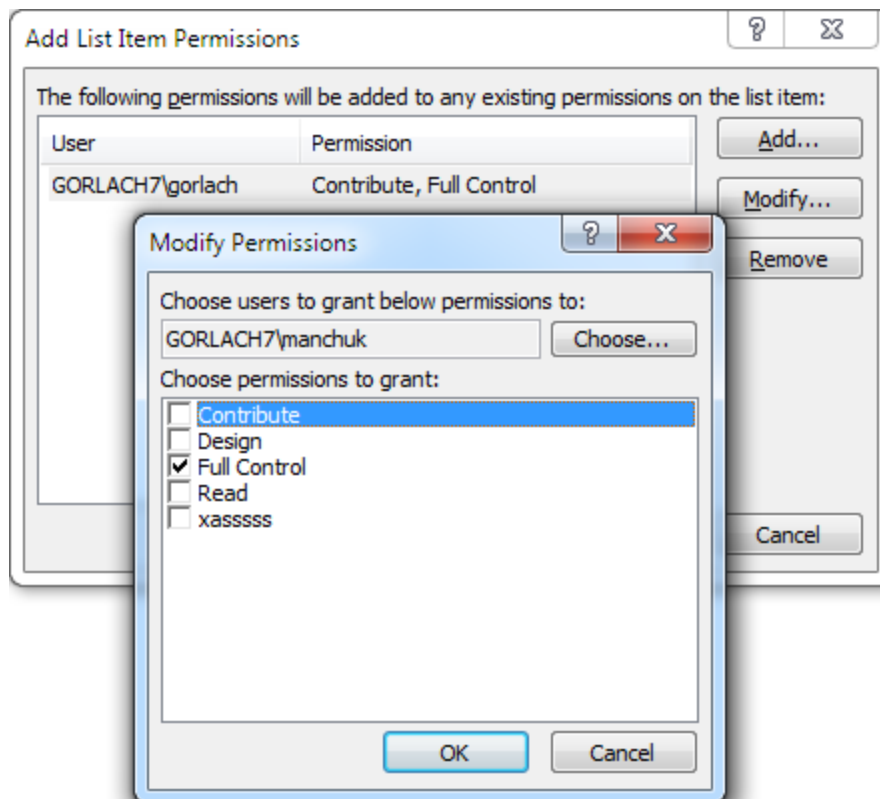
this list item - Target list item.

this user - All permissions for this user or group will be removed.

13.10 Delete Permissions at Url

Delete specified permissions at this url

This workflow actions is used to remove site users permissions from SharePoint site, list, list folder or list item specified by URL. In the Remove List Item Permissions window you can choose multiple users and select any permissions for them including Out Of The Box and custom permissions.



Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

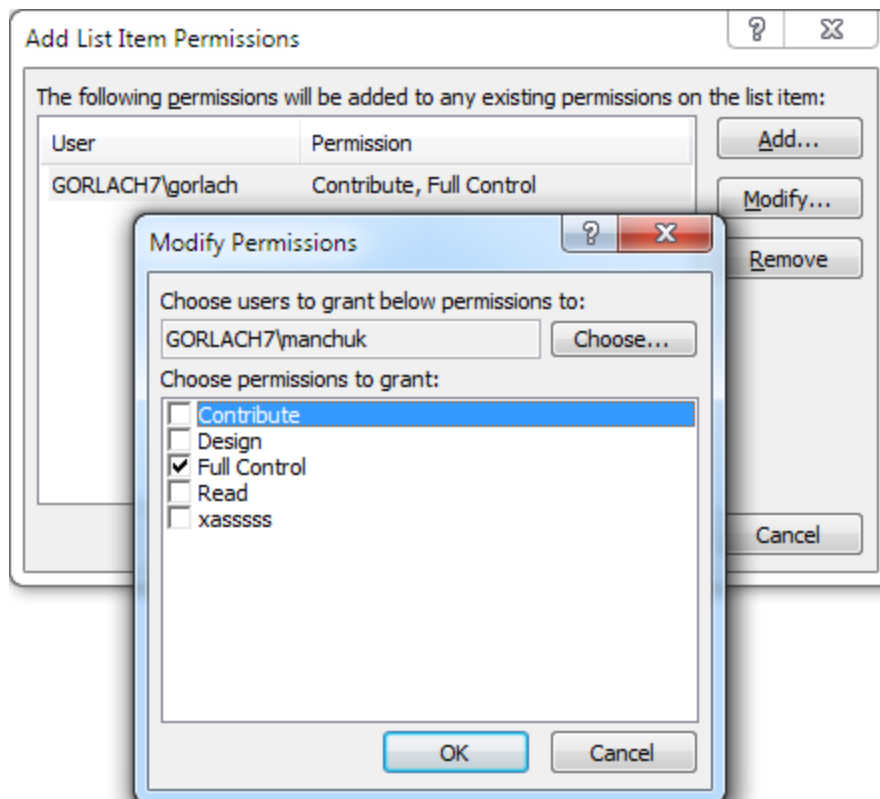
specified - Special-type array of the users and they permissions.

this url - Full URL of a SharePoint site, list, list folder or list item.

13.11 Delete Permissions from List

Delete specified permissions from this list

This workflow actions is used to remove permissions to site users for specified list. In the Remove List Item Permissions window you can choose multiple users and select any permissions for them including Out Of The Box and custom permissions.



Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

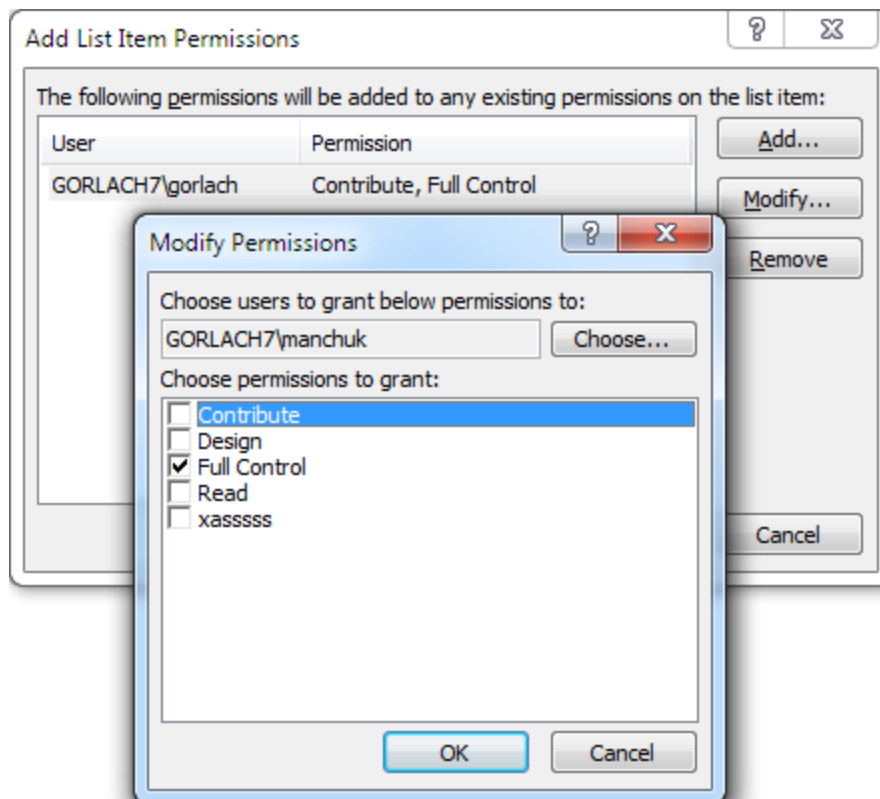
specified - Special-type array of the users and they permissions.

this list - Target list.

13.12 Delete Permissions from List Item

Delete specified permissions from this list item

This workflow actions is used to remove permissions to site users for specified list item. In the Remove List Item Permissions window you can choose multiple users and select any permissions for them including Out Of The Box and custom permissions.



Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

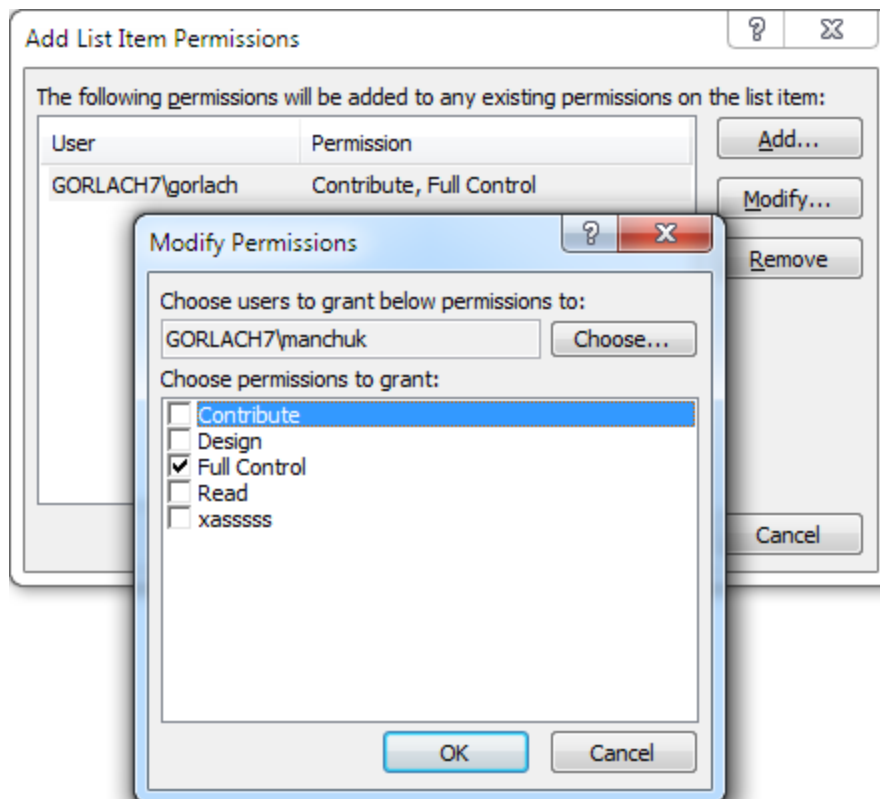
specified - Special-type array of the users and they permissions.

this list item - Target list item.

13.13 Grant Permissions at Url

Grant specified permissions at this url

This workflow actions is used to grant permissions to site users on SharePoint site, list, list folder or list item specified by URL. In the Add List Item Permissions window you can choose multiple users and add any permissions for them including Out Of The Box and custom permissions.



Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

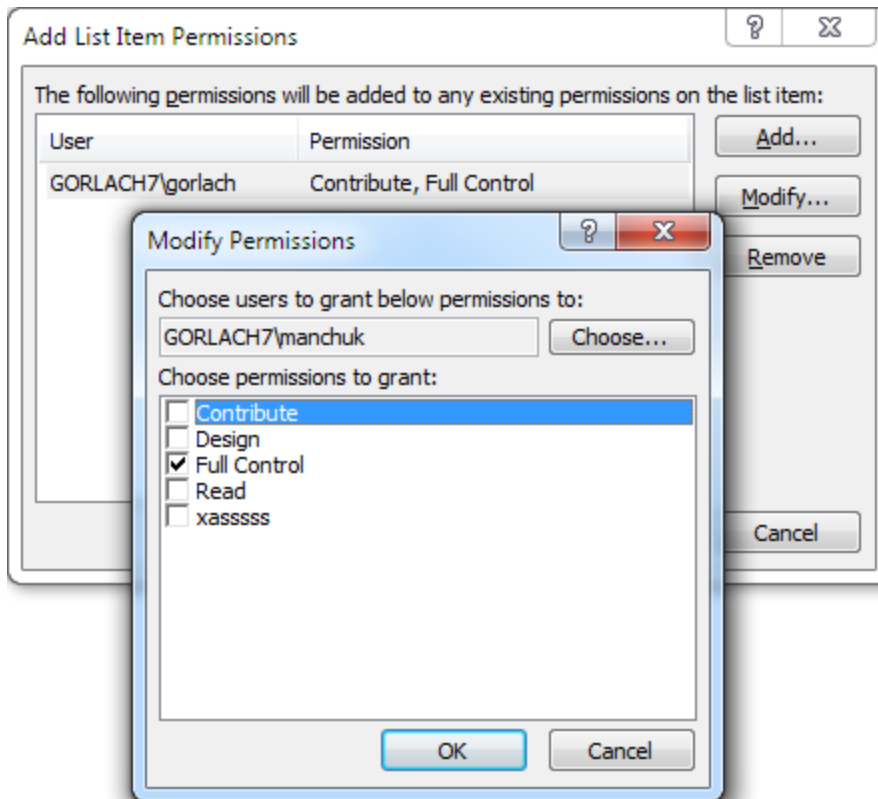
specified - Special-type array of the users and they permissions.

this url - Full URL of a SharePoint site, list, list folder or list item.

13.14 Grant Permissions on List

Grant specified permissions on this list

This workflow actions is used to grant permissions to site users for specified list. In the Add List Item Permissions window you can choose multiple users and add any permissions for them including Out Of The Box and custom permissions.



Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

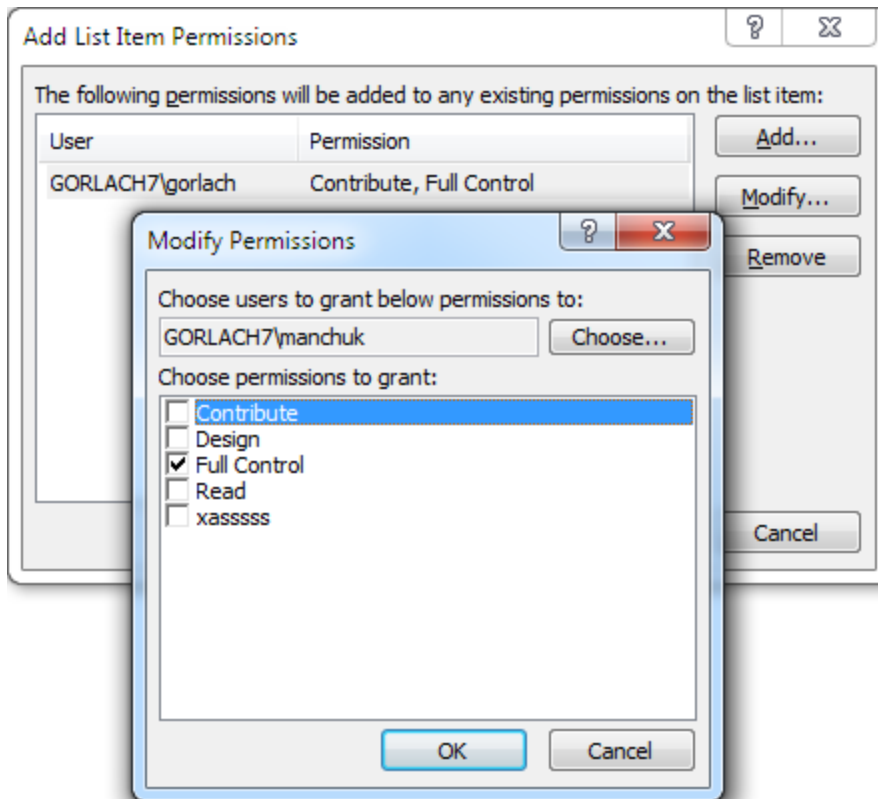
specified - Special-type array of the users and they permissions.

this list - Target list.

13.15 Grant Permissions on List Item

Grant specified permissions on this list item

This workflow actions is used to grant permissions to site users for specified list item. In the Add List Item Permissions window you can choose multiple users and add any permissions for them including Out Of The Box and custom permissions.



Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

specified - Special-type array of the users and they permissions.

this list item - Target list item.

14. Printing activities

14.1 Print Document

Print document at this url to default printer.

This workflow action is used to print specified document or image to specified printer. List of supported formats: Open XML Document (.docx, .docm, .dotx, .dotm); Word 97-2003 Document (.doc, .dot); Rich Text Format (.rtf); Web Page (.htm, .html, .mht, .mhtml); Text documents (.xml); Images (.jpg, .tif, .png, .gif, .bmp).

Note: To use this action you must install HarePoint Printing Service and configure HarePoint Printing Service Application. See product manual for details.

See Also: "Set Printing Options" action.

Parameters:

this url - Full URL of the document.

default - Network name of the printer.

Examples:

Print document

at <http://sp14.vlab.mapilab.local:133/Shared%20Documents/Test.docx> to <\\\\KOENIG\\OKIB410> printer.

14.2 Set Printing Options

Set number of copies: 1, print range: all and orientation: portrait.

This workflow action is used to specify page orientation, number of copies and pages to print.

Parameters:

1 - Specifies number of copies to print.

all - Specifies pages range to print, e.g. 1-5, 8, 11-13.

portrait - Specifies page layout.

15. Strings activities

15.1 Capture All Occurrences of Text

Capture regular expression in this text and store all occurrences in Variable: MatchResults

This workflow action is used to store all matches of regular expression in specified text to output array variable.

Regular expressions provide a powerful, flexible, and efficient method for processing text. To learn more about regular expressions, see: <http://msdn.microsoft.com/en-us/library/hs600312.aspx>

For regular expression language reference, see: <http://msdn.microsoft.com/en-us/library/az24scfc.aspx>

Note: To set additional options for this activity, like case sensitivity, use 'Set Regular Expression Options' activity.

Parameters:

regular expression - Regular expression pattern. For example, like '\b(\w+)\s\b'.
this text - Input text, expression, item field or variable.
Variable: MatchResults - Output array variable, each item contains one match of regular expression.

Examples:

Capture \b[A-Z0-9. %~]+@[A-Z0-9.~]+\.[A-Z]{2,4}\b in `xxx kkk@my.com owkk@ @kkk s@i12.com ddd` and store all occurrences in Variable: MatchResults2

then Join Variable: MatchResults2 at index 0 next -1 items using ; and store in Variable: emails

then Log Variable: emails to the workflow history list

Result in workflow history list:

kkk@my.com; s@i12.com

15.2 Capture Text

Capture regular expression in this text and store in Variable: FirstMatchResult

This workflow action is used to store first match of regular expression in specified text to output text variable.

Regular expressions provide a powerful, flexible, and efficient method for processing text. To learn more about regular expressions, see: <http://msdn.microsoft.com/en-us/library/hs600312.aspx>

For regular expression language reference, see: <http://msdn.microsoft.com/en-us/library/az24scfc.aspx>

Note: To set additional options for this activity, like case sensitivity, use 'Set Regular Expression Options' activity.

Parameters:

regular expression - Regular expression pattern. For example, like '\b(\w+)\s\b'.
this text - Input text, expression, item field or variable.
Variable: FirstMatchResult - Output string variable with the text of the first match of regular expression.

Examples:

Set regular expression options: ignore case - No, is pattern culture invariant - Yes, use multiline mode - No, use singleline mode - No, search right to left - No

then Capture .est in IT IS TEST and store in Variable: r1

then Set regular expression options: ignore case - Yes, is pattern culture invariant - Yes, use multiline mode - No, use singleline mode - No, search right to left - No

then Capture .est in IT IS TEST and store in Variable: r2

then Log <[%Variable: r1%]>, <[%Variable: r2%]> to the workflow history list

Result in workflow history list:

<>, <TEST>

15.3 Convert HTML To Text

Convert this HTML code to plain text and store result in Variable: variable.

This workflow action is used to strip HTML tags from specified string and return result in to output variable.

Parameters:

this - Text, expression, item field or variable to convert.

Variable: variable - Output string variable.

15.4 Convert Number To Text

Convert this number to text in English and store result in Variable: variable.

This workflow action is used to convert integer number into text using specified language. Supported languages: English, French, German, Russian.

Parameters:

this number - Input integer to convert.

English - One of the available languages.

Variable: variable - Output string variable.

Examples:

Convert Variable: totalPrice number to text in English and store result in Variable: totalPriceText.

15.5 Converts Base64 to Text

Converts specified Base64 text to text using default encoding and store result in Variable: ConvertedText.

This workflow activity will decode Base64 string, interpret resulting data as the string in one from five available encodings, and copy to output system string variable.

Available encodings:

- default - 8-bit encoding of the current operating system's ANSI code page.
- ASCII - 7-bit ASCII encoding.
- Unicode - also known as UTF16, represents each character as a sequence of one or two 16-bit integers.
- UTF8 - uses one byte for any ASCII characters, which have the same code values in both UTF-8 and ASCII encoding, and up to four bytes for other characters.
- UTF32 - uses exactly 32 bits for each character.

Because all Default encodings and ANSI encoding lose data (see Examples section), your workflow might use UTF8 instead.

Parameters:

specified Base64 text - Input text, expression, item field or variable.

default encoding - One from five available encodings.

Variable: ConvertedText - Output string variable.

Examples:

Let's encode German word 'Grüß' with different encodings with 'Converts Text to Base64' activity on the machine with Russian ANSI code page and decode results back with the same encodings with 'Converts Base64 to Text' activity.

- *default*: encoded string 'R3J1Pw==', decoded string 'Gru '
- *ASCII*: encoded string 'R3I/Pw==', decoded string 'Gr '
- *Unicode*: encoded string 'RwByAPwA3wA=', decoded string 'Grüß'
- *UTF8*: encoded string 'R3LDvMOF', decoded string 'Grüß'
- *UTF32*: encoded string 'RwAAAHIAAAD8AAAA3wAAAA==', decoded string 'Grüß'

As you can see, all three Unicode encodings save the German national characters unchanged. With 7-bit ASCII encoding, all national German characters were lost. In the case of default encoding, input string was initially mapped to Russian ANSI code page, where the German 'ü' was mapped to latin 'u' and the last 'ß' was lost, because it has no suitable mapping to Russian code page.

15.6 Converts Text to Base64

Converts specified text to Base64 using default encoding and store result in Variable: ConvertedText1.

This workflow activity will encode specified text with one from five available encodings and convert encoded text to Base64.

Available encodings:

- default - 8-bit encoding of the current operating system's ANSI code page.
- ASCII - 7-bit ASCII encoding.
- Unicode - also known as UTF16, represents each character as a sequence of one or two 16-bit integers.
- UTF8 - uses one byte for any ASCII characters, which have the same code values in both UTF-8 and ASCII encoding, and up to four bytes for other characters.
- UTF32 - uses exactly 32 bits for each character.

Because all Default encodings and ANSI encoding lose data (see Examples section), your workflow might use UTF8 instead.

Parameters:

specified text - Text, expression, item field or variable to encode.

default encoding - One from five available encodings.

Variable: ConvertedText1 - Output string variable.

Examples:

Let's encode German word 'Grüß' with different encodings with 'Converts Text to Base64' activity on the machine with Russian ANSI code page and decode results back with the same encodings with 'Converts Base64 to Text' activity.

- *default*: encoded string 'R3J1Pw==', decoded string 'Gru '
- *ASCII*: encoded string 'R3I/Pw==', decoded string 'Gr '
- *Unicode*: encoded string 'RwByAPwA3wA=', decoded string 'Grüß'
- *UTF8*: encoded string 'R3LDvMOF', decoded string 'Grüß'
- *UTF32*: encoded string 'RwAAAHIAAAD8AAAA3wAAAA==', decoded string 'Grüß'

As you can see, all three Unicode encodings save the German national characters unchanged. With 7-bit ASCII encoding, all national German characters were lost. In the case of default encoding, input string was initially mapped to Russian ANSI code page, where the German 'ü' was mapped to latin 'u' and the last 'ß' was lost, because it has no suitable mapping to Russian code page.

15.7 Decode Web Text

Decode specified text using url decoder and store in Variable: DecodedText.

This workflow action will decode text with one from two available decoding methods and return result in to output variable.

Available decode methods:

1. *url decoder* - Converts a URL-encoded string into a decoded string. If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3e, this method decodes its back.
2. *html decoder* - Converts a string that has been HTML-encoded for HTTP transmission into a decoded string. If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. HTML encoding converts characters that are not allowed in HTML into character-entity equivalents; HTML decoding reverses the encoding. For example, when embedded in a block of text, the characters < and > are encoded as < and > for HTTP transmission, this method decodes its back.

Parameters:

specified text - Text, expression, item field or variable to decode.

url decoder - One from two decoding methods: *url decoder* or *html decoder*.

Variable: DecodedText - Output string variable.

15.8 Encode Web Text

Encode specified text using url encoder and store in Variable: EncodedText.

Process text with one from four available encoding methods and returns result to output variable.

Available encoding methods:

1. *url encoder* - Can be used to encode the entire URL, including query-string values. If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents. For example, when the characters < and > are embedded in a block of text to be transmitted in a URL, they are encoded as %3c and %3e. Note, that this method converts each space character to a plus character (+).
2. *url path encoder* - The same as above, but converts each space character into the string "%20", which represents a space in hexadecimal notation. Use the this method when you encode the path portion of a URL in order to guarantee a consistent decoded URL, regardless of which platform or browser performs the decoding.
3. *html encoder* - Converts a string to an HTML-encoded string. If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. HTML encoding converts characters that are not allowed in HTML into character-entity equivalents. For example, when embedded in a block of text, the characters < and > are encoded as < and > for HTTP transmission.
4. *html attribute encoder* - Minimally converts a string to an HTML-encoded string. It converts only quotation marks ("), ampersands (&), and left angle brackets (<) to equivalent character entities. It is considerably faster than the *html encoder*. The result string should be used only for double-quoted (") attributes, security issues might arise when using result string with single-quoted (') attributes.

Parameters:

specified text - Text, expression, item field or variable to encode.

url encoder - One from four available encoding methods: *url encoder*, *url path encoder*, *html encoder*, *html attribute encoder*.

Variable: EncodedText - Output string variable.

Examples:

Encode [%Current Item:Approver Comments%] using url encoder and store in Variable: param.

then Send HTTP request to http://mysite/ q=[%Variable: param%] using GET method with this headers.
Store response in Variable: pageBody and status code in Variable: code.

15.9 Extract Substring

Extract substring from this text, starting at index 0 with -1 length and store in Variable: Substring.

Retrieves a substring from input text. The substring starts at a specified character position and has a specified length.

The main difference from standard SharePoint activity "Extract Substring from String from Index with Length" is following. In the case of exit out of bounds of string, standard activity will return empty string when this activity will decrease length parameter according with length of string.

Parameters:

this text - Input text, expression, item field or variable.

index 0 - Zero-based character position where substring starts.

-1 length - Substring length. If the length is out of string bounds, it will be decreased according with length of string. -1 is a special value which means 'to the end of input string'.

Variable: Substring - Output string variable.

Examples:

1. Example with this activity:

Extract substring from 123456, starting at index 3 with 2 length and store in Variable: var1.

then Extract substring from 123456, starting at index 1 with -1 length and store in Variable: var2.

then Extract substring from 123456, starting at index 1 with 100 length and store in Variable: var3.

Result: var1 is '45', var2 is '23456', var3 is '23456'.

2. Example with standard SharePoint activity "Extract Substring from String from Index with Length":

Copy from 123456, starting at 3 for 2 characters (Output to Variable: var1)

then Copy from 123456, starting at 1 for 100 characters (Output to Variable: var2)

Result: var1 is '45', var2 is ''.

15.10 Format Field of Current Item

Format this field with this format and store in Variable: FormattedText.

This workflow action is used to convert current item field to string with special formatting.

Format sting may have dozens of different specifiers. You can learn more about formatting is the following articles:

- Formatting Types: <http://msdn.microsoft.com/en-us/library/26etazsy.aspx#NetFormatting>
- Composite Formatting: <http://msdn.microsoft.com/en-us/library/txafckwd.aspx>
- Standard Date and Time Format Strings: <http://msdn.microsoft.com/en-us/library/az4se3k1.aspx>
- Standard Numeric Format Strings: <http://msdn.microsoft.com/en-us/library/dwhawy9k.aspx>
- Custom Numeric Format Strings: <http://msdn.microsoft.com/en-us/library/0c899ak8.aspx>

Parameters:

this field - Current item field.

this format - Format string.

Variable: FormattedText - Output string variable.

Examples:

Format Created field with {0:yyyy-MM-dd} and store in Variable: F2

then Format Modified By field with Item was modified by: {0} and store in Variable: F3

then Log [%Variable: F3%] at [%Variable: F2%] to the workflow history list

The result in workflow history list:

15.11 Get Random Number

Get random number within the range from 0 to 100 and store in Variable: RandomNumber1.

This workflow action is used to get the random number from specified range.

Note: The range of return values includes lower limit value but not higher limit value.

See Also: "Get Random String" activity.

Parameters:

0 - The lower limit of the range. May be negative.

100 - The higher limit of the range. May be negative.

Variable: RandomNumber1 - Output integer variable.

15.12 Get Random String

Get random string 16 characters long containing alphanumeric symbols and store in Variable:RandomString.

This workflow action is used to get the random string which contains characters from specified set.

By default, the following set is used: abcdefghijklmnopqrstuvwxyz0123456789

See Also: "Get Random Number" activity.

Parameters:

16 - Output string length.

alphanumeric - The set of characters for output string.

Variable:RandomString - Output string variable.

15.13 Is Value Matches Regular Expression (HarePoint)

If value matches regular expression (ignore case: Yes, is pattern culture invariant: Yes, use multiline mode: No, use singleline mode: No, search right to left: No)

This workflow condition returns True if the specified value matches with regular expression.

Regular expressions provide a powerful, flexible, and efficient method for processing text. To learn more about regular expressions, see: <http://msdn.microsoft.com/en-us/library/hs600312.aspx>

For regular expression language reference, see: <http://msdn.microsoft.com/en-us/library/az24scfc.aspx>

Parameters:

value - Text, expression or variable to check.

regular expression - Regular expression pattern. For example, like '\b(\w+)s\1\b'.

ignore case - Ignore case in regular expression and in the input string.

pattern is culture invariant - Specifies that cultural differences in language is ignored. Culture is used to obtain culture-specific information, such as casing rules and the alphabetical order of individual characters.

multiline mode - Multiline mode. Changes the meaning of ^ and \$ so they match at the beginning and end, respectively, of any line, and not just the beginning and end of the entire string.

singleline mode - Specifies single-line mode. Changes the meaning of the dot (.) so it matches every character (instead of every character except \n).

search right to left - Specifies that the search will be from right to left instead of from left to right.

15.14 Join Text

Joins string array at index 0 next -1 items using specified delimiter and store in Variable: JoinedText

This workflow action is used to join strings from array to one string using specified delimiter.

See Also: "Split Text" activity.

Parameters:

string array - Input strings array.

index - Zero-based index of the first array item to join.

next -1 - Zero-based index of the last array item to join. -1 means 'to the end of the array'.

specified delimiter - String placed between two array items.

Variable: JoinedText - Output string variable.

Examples:

Capture `\b[A-Z0-9. %-]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b` in `xxx kkk@my.com owkk@ @kkk s@i12.com ddd` and store all occurrences in **Variable: MatchResults2**

then Join **Variable: MatchResults2** at index `0` next `-1` items using `;` and store in **Variable: emails**

then Log **Variable: emails** to the workflow history list

Result in workflow history list:

kkk@my.com; s@i12.com

15.15 Replace Text

Find regular expression in this text, replace with this text and store in **Variable: ReplaceResult**.

This workflow action is used to store replace all matches of regular expression in specified text with specified string.

Regular expressions provide a powerful, flexible, and efficient method for processing text. To learn more about regular expressions, see: <http://msdn.microsoft.com/en-us/library/hs600312.aspx>

For regular expression language reference, see: <http://msdn.microsoft.com/en-us/library/az24scfc.aspx>

Note: To set additional options for this activity, like case sensitivity, use 'Set Regular Expression Options' activity.

Parameters:

regular expression - Regular expression pattern. For example, like `'\b(\w+ )\s\b'`.

this text - Input text, expression, item field or variable.

replace with this text - String to replace all matches.

Variable: ReplaceResult - Output string variable.

15.16 Set Regular Expression Options

Set regular expression options: ignore case - Yes, is pattern culture invariant - Yes, use multiline mode - No, use singleline mode - No, search right to left - No.

This workflow actions set options for regular expression activities like 'Capture Text' and 'Replace Text' in the current workflow step.

Parameters:

ignore case - Ignore case in regular expression and in the input string.

pattern is culture invariant - Specifies that cultural differences in language is ignored. Culture is used to obtain culture-specific information, such as casing rules and the alphabetical order of individual characters.

multiline mode - Multiline mode. Changes the meaning of ^ and \$ so they match at the beginning and end, respectively, of any line, and not just the beginning and end of the entire string.

singleline mode - Specifies single-line mode. Changes the meaning of the dot (.) so it matches every character (instead of every character except \n).

search right to left - Specifies that the search will be from right to left instead of from left to right.

Examples:

Set regular expression options: ignore case - No, is pattern culture invariant - Yes, use multiline mode - No, use singleline mode - No, search right to left - No

then Capture .est in IT IS TEST and store in Variable: r1

then Set regular expression options: ignore case - Yes, is pattern culture invariant - Yes, use multiline mode - No, use singleline mode - No, search right to left - No

then Capture .est in IT IS TEST and store in Variable: r2

then Log <[%Variable: r1%]>, <[%Variable: r2%]> to the workflow history list

Result in workflow history list:

<>, <TEST>

15.17 Split Text

Splits specified text using specified delimiters (strings, one per line) and store in Variable: StringsArray

This workflow action is used to split text and store results to strings array. You can specify string delimiters (one per line) or character delimiters (each character in specified string is delimiter).

See Also: "Join Text" activity.

Parameters:

<u>specified text</u> - Text, expression, item field or variable.
<u>specified delimiters</u> - String or multi-line string with delimiters.
<u>strings/characters</u> - Type of delimiters. You can choose string delimiters (one per line) or character delimiters (each character in specified string is delimiter).
<u>Variable: StringsArray</u> - Output strings array.

15.18 Trim Text

Removes leading and trailing white-space characters from this text and store in Variable: TrimmedText

Removes white-space characters from beginning and/or end of specified text.

Parameters:

<u>leading and trailing</u> - One from three options: <i>leading</i> , <i>trailing</i> , <i>leading and trailing</i> .
<u>this text</u> - Text, expression, item field or variable.
<u>Variable: TrimmedText</u> - Output variable.

16. Workflow activities

16.1 Get Workflow Association Data

Get this workflow default association data and store in Variable: variable

This workflow action is used to retrieve workflow association data.

See Also: "Start Workflow" activity.

Parameters:

this workflow - Workflow name.

Variable: variable - Output string variable with workflow association data in XML format.

16.2 Get Workflow Status

Get status of workflow with this name started on item at this url, store status in Variable: variable1, last status change time in Variable: variable2.

This workflow action is used to get site or list workflow status and the time of last status change.

Known workflow status strings are:

- Not Started
- Failed On Start
- In Progress
- Error Occurred
- Stopped By User
- Completed
- Canceled

This list may be extended by custom values, e.g. using 'Set Workflow Status' action.

Parameters:

with this workflow name - Workflow name, e.g. 'My workflow'.

item at this url - Optional parameter with item URL.

Variable: variable1 - Output string variable for status code, e.g. 'Completed'.

Variable: variable2 - Output Date/Time variable for last status change time.

16.3 Start Workflow

Start this workflow on the item with this url with default association data and store instance ID in Variable: variable.

This workflow action is used to start list or site workflow on SharePoint site. Due to the nature of Microsoft SharePoint Workflow Timer start a new workflow can take several seconds to several minutes, during which the workflow will have the status "Starting".

See Also: "Get Workflow Association Data" activity.

Parameters:

this workflow - Workflow name.
the item with this url - Optional parameter with item URL.
default - Workflow association data in XML format.
Variable: variable - Output string variable for workflow instance identifier.

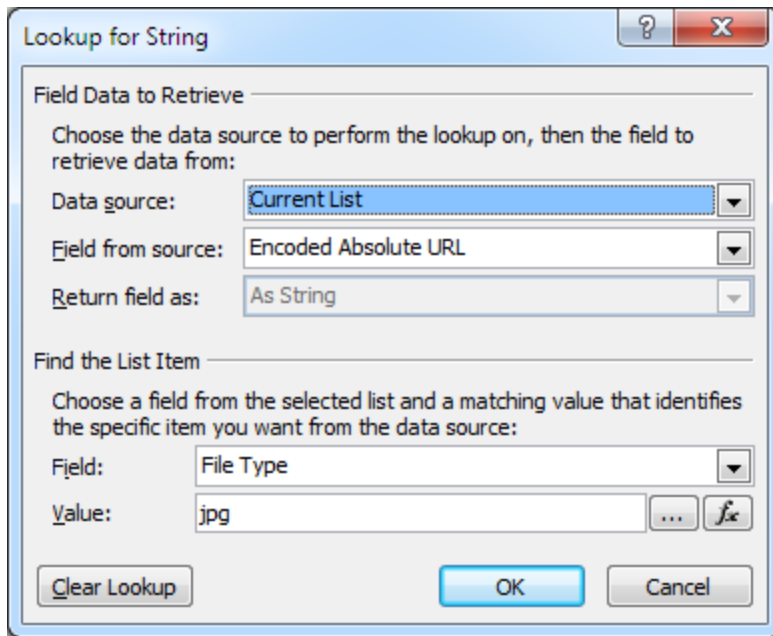
Examples:

Convert image at Current Item: Encoded Absolute URL to PNG format and store url to converted image in Variable: ConvertedImageUrl. Overwrite exisiting image.

then Delete item at Current Item: Encoded Absolute URL list item url.

then Start ConvertToPng on Current List: Encoded Absolute URL with default association data and store instance ID in Variable: ID.

This sample workflow convert all JPG images in document library to PNG. Use the following lookup settings for the second parameter of 'Start Workflow' action:



16.4 Terminate Workflow

Terminate workflow with this instance ID.

This workflow action is used to terminate running workflows started by 'Start Workflow' action.

Parameters:

with this instance ID - The ID of running workflow obtained from 'Start Workflow' action.

16.5 Wait Workflow To Complete

Wait until workflow with this instance ID is completed and store status of the workflow in Variable: variable.

This workflow action is used to wait until workflow started by 'Start Workflow' action is completed.

Known workflow status strings are:

- Not Started
- Failed On Start
- In Progress
- Error Occurred

- Stopped By User
- Completed
- Canceled

This list may be extended by custom values, e.g. using 'Set Workflow Status' action.

Parameters:

with this instance ID - The ID of running workflow obtained from 'Start Workflow' action.

Variable: variable - Output string variable for workflow status.

16.6 Wait Workflow To Complete By Name

Wait until workflow with this name started on item at this url is completed, store status of the workflow in Variable: variable1 and last status change time in Variable: variable2.

This workflow action is used to wait until site or list workflow with specified name will be completed.

Known workflow status strings are:

- Not Started
- Failed On Start
- In Progress
- Error Occurred
- Stopped By User
- Completed
- Canceled

This list may be extended by custom values, e.g. using 'Set Workflow Status' action.

Parameters:

with this name - Workflow name, e.g. 'My workflow'.

item at this url - Optional list item URL used for list workflows.

Variable: variable1 - Output string for workflow status.

Variable: variable2 - Output Date/Time variable for last status change time.

Appendix A. List of activities from Free Set

Activities from the below list are included to the free set. It means that after expiration of the Trial License, you still can use these activities to develop workflows and workflows created with its use will provide full functionality (see chapter 2 of Section II).

1. Add Array Item
2. Clear Array
3. Create Array
4. Get Array Item
5. Get Array Items Count
6. Get Index of Array Item
7. Insert Array Item
8. Remove Array Item
9. Remove Array Item at
10. Remove Duplicates from Array
11. Converts Date from Local Time to UTC
12. Converts Date from UTC to Local Time
13. Converts Date to Text
14. Get Date Property
15. Set DateTime Constant
16. Capture All Occurrences of Text
17. Capture Text
18. Convert HTML To Text
19. Convert Number To Text
20. Converts Base64 to Text
21. Converts Text to Base64
22. Decode Web Text
23. Encode Web Text
24. Extract Substring
25. Format Field of Current Item
26. Get Random Number

- 27. Get Random String
- 28. Is Value Matches Regular Expression (HarePoint)
- 29. Join Text
- 30. Replace Text
- 31. Set Regular Expression Options
- 32. Split Text
- 33. Trim Text