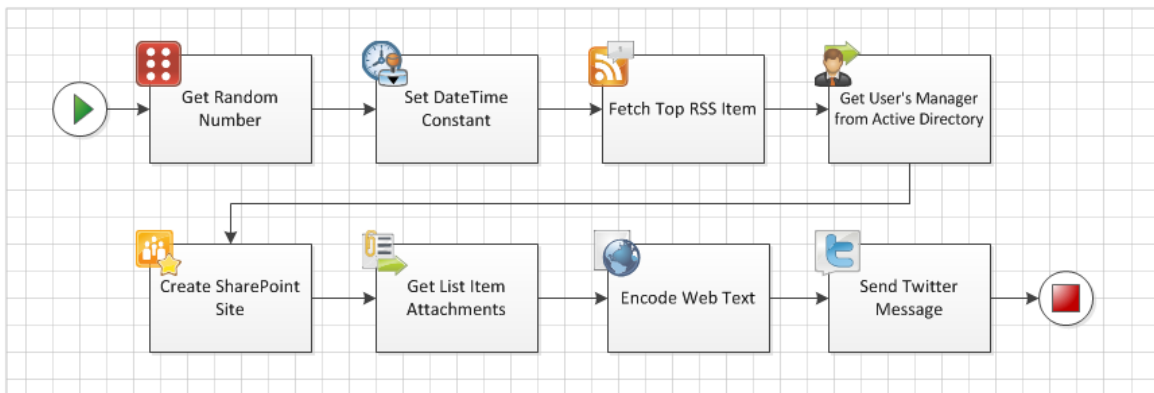


HarePoint Workflow Extensions Manual

For SharePoint Server Subscription Edition, SharePoint Server 2010/2013/2016/2019, SharePoint Foundation 2010/2013..



Product version 2.21

10/31/2023

(This Page Intentionally Left Blank)

Table of Contents

- Table of Contents 3
- I. Introduction 16
- II. Deployment..... 17
 - 1. System requirements..... 17
 - 2. Licensing and trial limitations 18
 - 3. Installing HarePoint Workflow Extensions..... 19
 - 4. Installing Client Applications 23
- III. How to upgrade, repair, or remove product 25
 - 1. Upgrading, repairing, or removing HarePoint Workflow Extensions 25
 - 2. Upgrading, repairing, or removing Client Applications..... 28
- IV. Product Overview 31
 - 1. Getting started 31
 - 2. Creating workflows in SharePoint Designer 32
 - 3. Encrypt Credentials tool..... 32
 - 4. Social Authorization tool..... 34
 - 5. Designing workflows in Microsoft Visio 36
 - 6. Creating workflows in Visual Studio 37
 - 7. Security concerns 38
 - 8. How to disable certain HarePoint activities..... 39
- V. HarePoint Activities and Conditions 41
 - 1. Active Directory activities 41
 - 1.1 Add Member to Active Directory Group..... 41
 - 1.2 Change Active Directory Account Settings..... 42
 - 1.3 Change Active Directory Group Settings..... 42
 - 1.4 Change Local User Account Settings on Server 43
 - 1.5 Create Group in Active Directory 44
 - 1.6 Create Local User Account on Server..... 44
 - 1.7 Create Mail Contact in Active Directory 45
 - 1.8 Create User in Active Directory..... 46

1.9 Delete Contact from Active Directory.....	47
1.10 Delete Group from Active Directory	47
1.11 Delete Local User Account from Server	47
1.12 Delete User from Active Directory.....	48
1.13 Find Group in Active Directory.....	48
1.14 Find Objects in Active Directory by Query	49
1.15 Find User in Active Directory	49
1.16 Find User in Active Directory by Query.....	50
1.17 Get Active Directory Groups Where User is Member.....	51
1.18 Get E-Mails of Active Directory Group Members	51
1.19 Get Object's Attribute from Active Directory	52
1.20 Get Object's Multivalued Attribute from Active Directory	52
1.21 Get User's Attribute from Active Directory	53
1.22 Get User's Manager from Active Directory.....	53
1.23 Get User's Multivalued Attribute from Active Directory	54
1.24 Is User a Member of an Active Directory Group (HarePoint)	54
1.25 Move User into Organizational Unit	55
1.26 Remove Member from Active Directory Group.....	55
1.27 Set Active Directory Authentication	56
1.28 Set Local User's Password on Server.....	57
1.29 Set Object's Attribute in Active Directory.....	58
1.30 Set Object's Multivalued Attribute in Active Directory	58
1.31 Set User's Attribute in Active Directory	59
1.32 Set User's Multivalued Attribute in Active Directory.....	59
1.33 Set User's Password in Active Directory	60
1.34 Set User's Photo in Active Directory	60
2. Administration activities	60
2.1 Activate Feature.....	60
2.2 Add Users to SharePoint Group	61
2.3 Add Users to Site.....	61
2.4 Change SharePoint Group Owner	61

2.5 Change SharePoint Group Settings.....	62
2.6 Change SharePoint List Settings.....	63
2.7 Create Host Named Site Collection.....	64
2.8 Create SharePoint Group	65
2.9 Create SharePoint Site	65
2.10 Create Site Collection.....	66
2.11 Deactivate Feature.....	67
2.12 Delete Entire SharePoint List at Url	68
2.13 Delete SharePoint Group	68
2.14 Delete SharePoint Site	68
2.15 Delete Site Collection.....	68
2.16 Extract Users from SharePoint Group to List.....	69
2.17 Get Multiple Site Users Property	70
2.18 Get Site Property.....	70
2.19 Get Site User Property	71
2.20 If SharePoint Group Exists (HarePoint)	72
2.21 If SharePoint Site Exists.....	72
2.22 Is Role Assigned to User (HarePoint)	73
2.23 Is User a Member of a SharePoint Group (HarePoint)	73
2.24 Remove User from SharePoint Group	73
2.25 Remove User from Site	74
2.26 Rename Site at Url	74
2.27 Set Master Page	74
2.28 Set SharePoint List Anonymous Access	75
2.29 Set SharePoint Site Anonymous Access.....	75
2.30 Set SharePoint Site Navigation Options.....	76
2.31 Set Site Property	76
2.32 Set Site User Property.....	77
2.33 Set User's Photo in SharePoint	78
3. Arrays activities.....	78
3.1 Add Array Item.....	78

3.2 Clear Array.....	78
3.3 Create Array	79
3.4 Get Array Item.....	79
3.5 Get Array Items Count	80
3.6 Get Index of Array Item.....	80
3.7 Insert Array Item	81
3.8 Remove Array Item	82
3.9 Remove Array Item at	82
3.10 Remove Duplicates from Array	82
4. Date/Time activities	82
4.1 Converts Date from Local Time to UTC.....	82
4.2 Converts Date from UTC to Local Time	83
4.3 Converts Date to Text	83
4.4 Evaluate Due Date.....	84
4.5 Get Date Property	85
4.6 Get Interval Between Dates	86
4.7 Get Working Time Between Dates.....	87
4.8 Set DateTime Constant	87
5. Development activities	88
5.1 Convert JSON to XML	88
5.2 Execute Custom Code	88
5.3 Execute PowerShell Script.....	90
5.4 Execute SQL Command	92
5.5 Execute SQL Command into List	95
5.6 Query XML.....	96
5.7 Transform XML.....	97
5.8 Update XML.....	100
6. Document Conversion activities	102
6.1 Convert Document	102
6.2 Convert Folder at Url.....	103
6.3 Convert Image to PDF	104

6.4 Convert InfoPath Form and Save to Document Library	105
6.5 Convert Library.....	106
6.6 Get Conversion Job Status	107
6.7 Wait Conversion Job	107
7. Document Library activities	108
7.1 Add Document to ZIP Archive	108
7.2 Check In Item at Url.....	109
7.3 Check Out Item at Url.....	109
7.4 Copy Document to List Item Attachment	109
7.5 Copy Folder Structure	109
7.6 Copy or Move Document	110
7.7 Copy or Move Document to Different Farm	111
7.8 Copy or Move Folder.....	112
7.9 Create Document from File.....	113
7.10 Create Document from Text	113
7.11 Create Document Set in the Document Library.....	114
7.12 Create Folder in the Document Library	114
7.13 Create Folder in the Document Library at Url.....	114
7.14 Create Link to a Document	115
7.15 Delete Folder in the Document Library.....	115
7.16 Delete Folder in the Document Library at Url.....	115
7.17 Discard Check Out Item at Url.....	116
7.18 Discard Publish Item	116
7.19 Download Documents to Folder	116
7.20 Get Document Content	117
7.21 Get Entry from Folder Property Bag	118
7.22 Is File Locked (HarePoint).....	118
7.23 Publish Item	119
7.24 Rename Document at Url	119
7.25 Save Document to File	119
7.26 Set Network File Access	120

7.27 Set Target Site Options	120
7.28 Upload Documents from Folder	121
7.29 Wait Until File Is Unlocked	122
8. E-Mail activities	122
8.1 Add Custom Headers to Email	122
8.2 Get E-mail Property	123
8.3 Receive All E-Mails to List from IMAP Server	124
8.4 Receive All E-Mails to List from POP3 Server	125
8.5 Receive E-Mail from IMAP Server	127
8.6 Receive E-Mail from POP3 Server	128
8.7 Save E-Mail Attachments to List Item	130
8.8 Send E-Mail with Document as Attachment	130
8.9 Send E-Mail with File Attachments	131
8.10 Send E-Mail with Http File Attachment	132
8.11 Send E-Mail with InfoPath Attachments	133
8.12 Send E-Mail with InfoPath Form	135
8.13 Send E-Mail with List Item Attachments	136
8.14 Send E-Mail with Multiple Attachments	137
8.15 Send E-mail with Word Document as a Body	138
8.16 Send E-Mail without Attachments	139
8.17 Send SMS Message	140
8.18 Set E-mail Delivery and Processing Options	140
8.19 Set SMTP Server Options	142
9. Exchange activities	143
9.1 Create Exchange Calendar Item	143
9.2 Delete Exchange Calendar Item	144
9.3 Respond to Exchange Calendar Item	144
9.4 Send Meeting Request	144
9.5 Set Exchange Web Service Options	145
9.6 Update Exchange Calendar Item	147
9.7 Update Exchange Calendar Item Advanced Properties	147

9.8 Update Exchange Calendar Item Attendees	148
9.9 Update Recurring Exchange Calendar Item	149
10. Images activities.....	152
10.1 Convert Image.....	152
10.2 Convert Images in Folder	153
10.3 Convert Images in Library	154
10.4 Flip Image	154
10.5 Get EXIF Property from Image	154
10.6 Get Image Size.....	156
10.7 Get XMP Property	158
10.8 Is Image Compressed (HarePoint).....	158
10.9 Is Image Type Supported (HarePoint).....	159
10.10 Resize Image	161
10.11 Resize Images in Folder.....	161
10.12 Resize Images in Library.....	162
10.13 Rotate Image.....	163
10.14 Set EXIF Property on Image.....	163
10.15 Set Transparent Color	164
10.16 Set XMP Property.....	164
11. Lists activities	165
11.1 Add Content Type to List.....	165
11.2 Add Existing Column to Content Type	165
11.3 Approve Item at Url	166
11.4 Build String from List.....	167
11.5 Calculate Aggregation Function	168
11.6 Copy InfoPath Attachments to Document Library.....	170
11.7 Copy List Item at Url.....	171
11.8 Copy List Item Attachments to Document Library.....	172
11.9 Copy List Item Attachments to Document Library (advanced).....	172
11.10 Copy List Item Attachments to List Item.....	173
11.11 Copy List Item to Different Farm.....	173

11.12 Create Content Type	174
11.13 Create List	174
11.14 Create List Item at Url	175
11.15 Create List Item on Different Farm	178
11.16 Create List Items from Array	179
11.17 Create Site Column	182
11.18 Delete All List Items	182
11.19 Delete Item at Url.....	184
11.20 Delete Item on Different Farm.....	184
11.21 Download InfoPath Attachments to Folder	184
11.22 Find List Item at Url.....	185
11.23 Find List Item by CAML.....	186
11.24 Find List Item by CAML on Different Farm.....	187
11.25 Find List Item on Different Farm	188
11.26 Get Field Value at Url	188
11.27 Get Field Value from Different Farm	189
11.28 Get List Item Attachment Content.....	190
11.29 Get List Item Attachments	190
11.30 Get List Item Url	191
11.31 Get List Url.....	193
11.32 Get Multi-Valued Field Value at Url	194
11.33 Get Special List Item Url at Url.....	194
11.34 Get the Number of Versions of a List Item	197
11.35 Get Workflow Task Property.....	197
11.36 If List Exists (HarePoint)	198
11.37 Is Current Item Checked Out (HarePoint)	198
11.38 Is Specified Item Checked Out (HarePoint).....	198
11.39 Loop Through and Update List Items.....	198
11.40 Move List Item to Subfolder	201
11.41 Reject Item at Url	201
11.42 Remove All Attachments.....	201

11.43 Remove Content Type from List	202
11.44 Remove InfoPath Attachments	202
11.45 Set Content Type Document Template.....	202
11.46 Set Default Content Type on List	203
11.47 Update List Item at Url.....	203
11.48 Update List Item on Different Farm	206
11.49 Update Workflow Task at Url.....	207
11.50 Upload Files to List Item Attachments	207
11.51 Wait for Field Change in Current Item Until Date.....	208
11.52 Wait for Field Change in Current Item with Timeout	208
12. Network activities	209
12.1 Create Directory on FTP Server	209
12.2 Delete Directory from FTP Server	209
12.3 Delete File from FTP Server	210
12.4 Download Document from FTP Server	210
12.5 Download Documents from FTP Folder	211
12.6 Fetch RSS Feed to Html.....	212
12.7 Fetch RSS Item	213
12.8 Get Formsite Form Results.....	213
12.9 Send HTTP GET Request.....	215
12.10 Send HTTP OAuth Request.....	216
12.11 Send HTTP POST Request.....	218
12.12 Send XMPP Instant Message.....	219
12.13 Set Encrypted Passwords	219
12.14 Set FTP Server Options.....	220
12.15 Set HTTP Request Options	221
12.16 Set OAuth Authentication	222
12.17 Set Web Service Options.....	223
12.18 Shorten Url.....	223
12.19 Upload Document to FTP Server.....	224
12.20 Upload Documents to FTP Folder	224

12.21 Upload List Item Attachments to FTP Server	226
13. Office activities.....	226
13.1 Accept Revision Changes in Word Document.....	226
13.2 Add Digital Signature	227
13.3 Add New Row into Excel Document	227
13.4 Add New Row into Word Document	228
13.5 Add Watermark To Word Document.....	229
13.6 Find Text in Word Document.....	229
13.7 Get Excel Cell Value.....	229
13.8 Get Office Document Property	230
13.9 Get Word Document Page Count	231
13.10 Read Word Document	231
13.11 Read Word Document to Text	232
13.12 Replace External Hyperlink in Word Document	232
13.13 Replace Text in Word Document	233
13.14 Set Excel Cell Value	234
13.15 Set Office Document Property.....	234
13.16 Sign PDF Document.....	235
13.17 Update Multiple Fields In Word Document.....	236
13.18 Update Picture in Word Document	237
13.19 Update Word Document.....	238
14. Permissions activities.....	240
14.1 Change Permissions Inheritance at Url.....	240
14.2 Change Permissions Inheritance for List.....	241
14.3 Change Permissions Inheritance for List Item	241
14.4 Delete All Permissions at Url.....	242
14.5 Delete All Permissions from List	242
14.6 Delete All Permissions from List Item	242
14.7 Delete All User Permissions at Url	243
14.8 Delete All User Permissions from List.....	243
14.9 Delete All User Permissions from List Item.....	243

14.10 Delete Permissions at Url.....	244
14.11 Delete Permissions from List	245
14.12 Delete Permissions from List Item	246
14.13 Grant Permissions at Url	247
14.14 Grant Permissions on List.....	248
14.15 Grant Permissions on List Item	249
14.16 Is Permission Inheritance Set (HarePoint)	250
14.17 Set Permissions at Url	250
15. Printing activities.....	250
15.1 Print Document.....	250
15.2 Set Printing Options	251
16. Social activities.....	252
16.1 Create Group in Yammer	252
16.2 Create MailChimp Campaign	252
16.3 Create User in Yammer	253
16.4 Delete MailChimp Campaign	254
16.5 Delete User from Yammer	254
16.6 Get User's Property from Yammer.....	255
16.7 Invite User to Yammer	255
16.8 Publish Message to Social Media via dlvr.it.....	256
16.9 Publish Message to Yammer	256
16.10 Replicate MailChimp Campaign	257
16.11 Schedule MailChimp Campaign	257
16.12 Send MailChimp Campaign	258
16.13 Send Twitter Message.....	258
16.14 Set User's Property in Yammer	260
16.15 Subscribe to MailChimp List.....	261
16.16 Unsubscribe from MailChimp List.....	261
16.17 Update Group in Yammer	262
17. Strings activities	262
17.1 Capture All Occurrences of Text	262

17.2 Capture Text.....	263
17.3 Convert HTML to Text	264
17.4 Convert Number to Text	264
17.5 Convert Text Case	265
17.6 Converts Base64 to Text	265
17.7 Converts Text to Base64	266
17.8 Decode Web Text.....	267
17.9 Encode Web Text	268
17.10 Evaluate Mathematical Expression.....	269
17.11 Extract Substring.....	270
17.12 Format Field of Current Item	271
17.13 Get Random Number	272
17.14 Get Random String.....	272
17.15 Get Text Length.....	273
17.16 Is Value Matches Regular Expression (HarePoint).....	273
17.17 Join Text	274
17.18 Replace Text.....	274
17.19 Set Regular Expression Options	275
17.20 Split Text.....	276
17.21 Trim Text	276
18. Taxonomy activities	277
18.1 Add Language to Term Store	277
18.2 Create Taxonomy Group.....	277
18.3 Create Taxonomy Term.....	277
18.4 Create Taxonomy Term Set.....	278
18.5 Delete Taxonomy Group	278
18.6 Delete Taxonomy Term.....	279
18.7 Delete Taxonomy Term Set.....	279
18.8 Update Taxonomy Group.....	280
18.9 Update Taxonomy Term	280
18.10 Update Taxonomy Term Set	281

19. Workflow activities	281
19.1 Get SharePoint 2013 Workflow Instance Id.....	281
19.2 Get SharePoint 2013 Workflow Status	282
19.3 Get Workflow Association Data	283
19.4 Get Workflow Instance Id	284
19.5 Get Workflow Status.....	285
19.6 Get Workflow Template ID	285
19.7 Start SharePoint 2013 Workflow	286
19.8 Start Workflow	287
19.9 Terminate SharePoint 2013 Workflow	288
19.10 Terminate Workflow	289
19.11 Wait SharePoint 2013 Workflow to Complete	289
19.12 Wait SharePoint 2013 Workflow to Complete by Name	290
19.13 Wait Workflow to Complete	290
19.14 Wait Workflow to Complete By Name.....	291
Appendix A. List of activities from Free Set	293

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, PowerShell cmdlets, 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 340 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 Visio¹, Microsoft Visual Studio;
- ✓ easy deployment, integration and minimum training costs.

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

¹ Current version supports Microsoft Visio 2010 only; Visio 2013 integration will be added in the nearest release

II. Deployment

1. System requirements

HarePoint Workflow Extensions is designed to operate in SharePoint Server Subscription Edition, SharePoint Server 2010/2013/2016/2019, SharePoint Foundation 2010/2013 (hereinafter referred to as SharePoint server) 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 server is deployed. The product requires 10 Mb of disk space on each SharePoint 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
Social Authorize Tool	Utility to generate authorization token for activities of Social category (see description of this activity in Section V).	Microsoft Internet Explorer 6.0 or higher
HarePoint Visio Add-In	Library of shapes and support module for HarePoint activities in Visio 2010 (see chapter 4 of Section IV).	Microsoft Visio Premium 2010 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 IV).	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 by a number of SharePoint servers where “**Microsoft SharePoint Foundation Workflow Timer Service**” is installed. The number of product licenses 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 are not accounted for the product licensing.

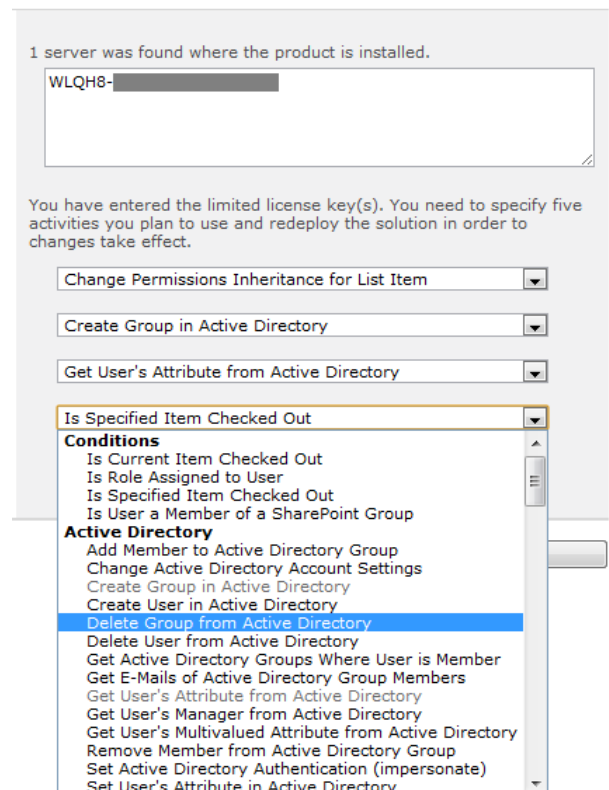
You may use the product on the basis of one of 4 licenses:

1. *Trial License*. Provided automatically for a 30-day period to use the product in organization’s environment without any limitation on number of servers and/or 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 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 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 server 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 a 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 an 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



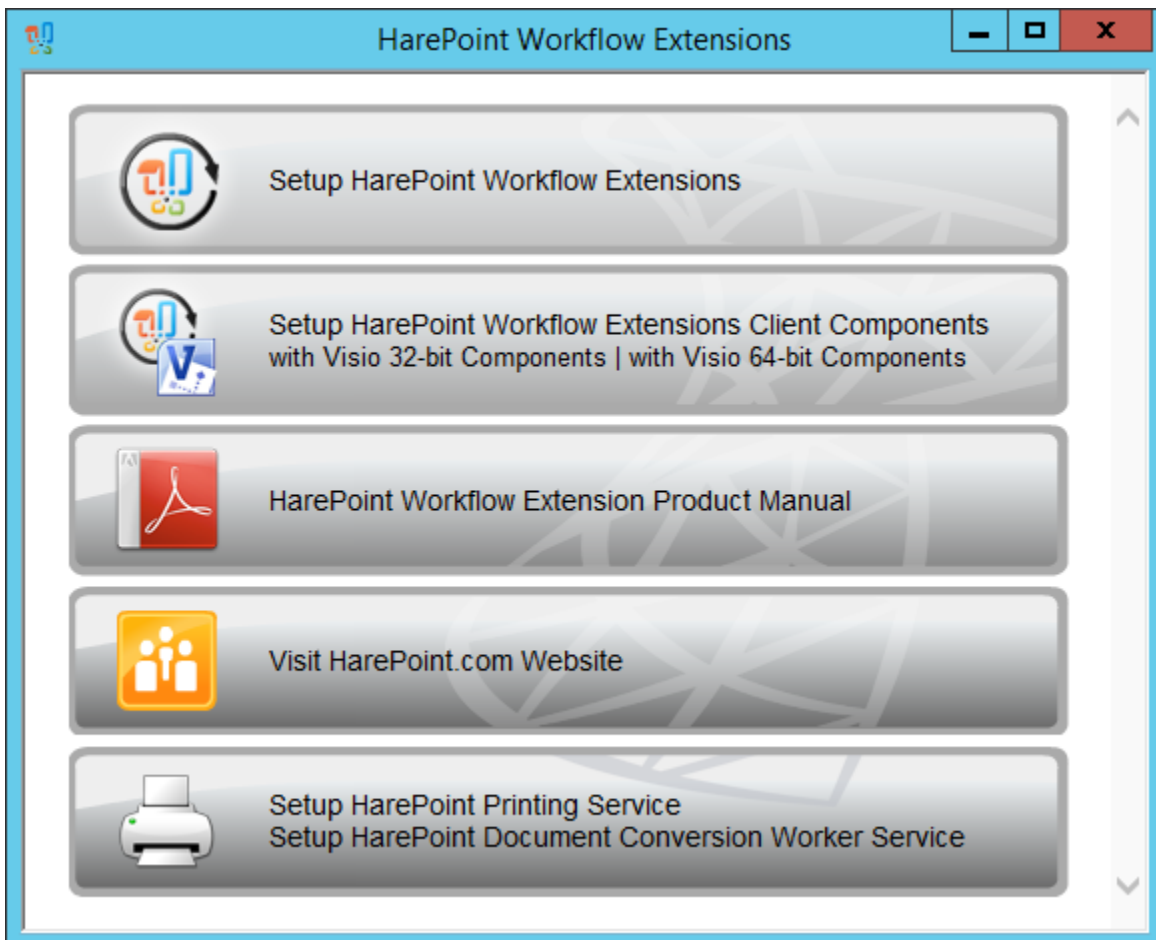
the paid list (in addition to activities from the free set). This selection can be changed later. The rest of paid activities will be inaccessible for creation of workflows, while execution of workflows created with their help will be terminated with error and record “This workflow action requires a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

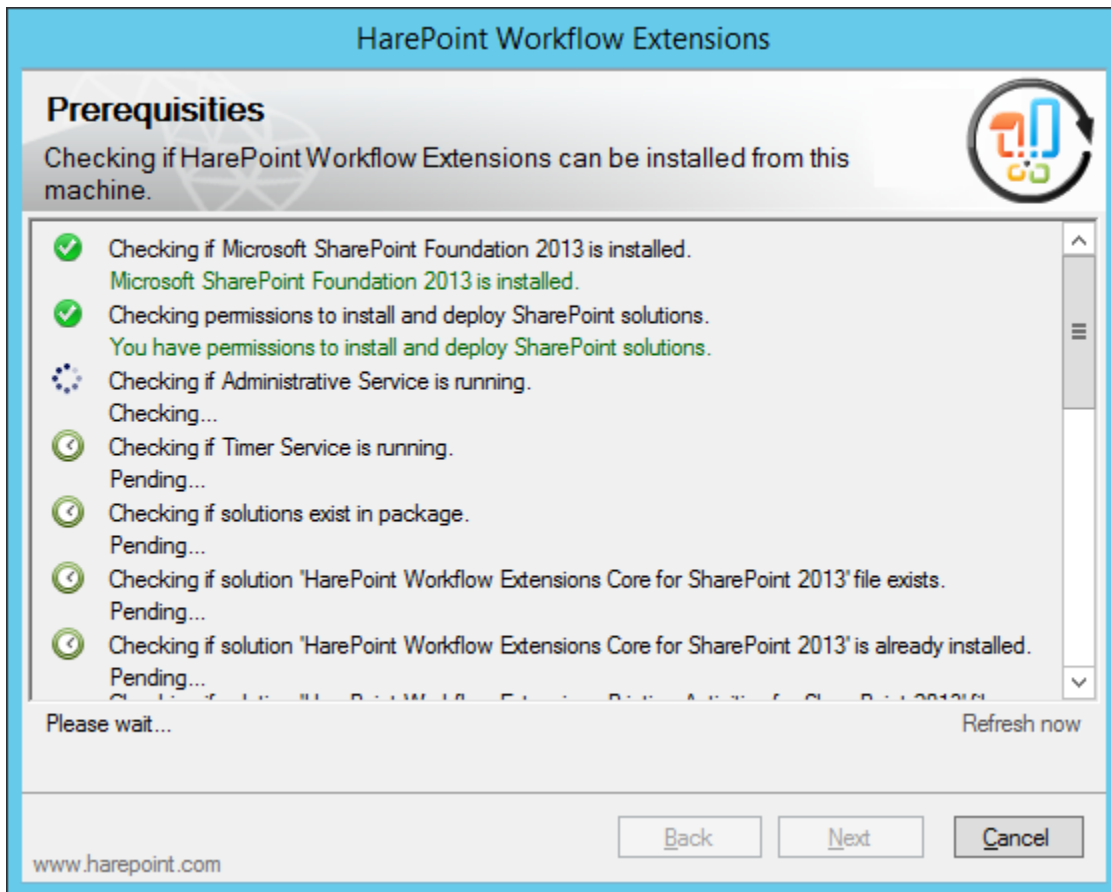
3. Installing HarePoint Workflow Extensions

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

Access any Web Front-End server of the SharePoint farm, unpack the product archive to a folder on a 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:



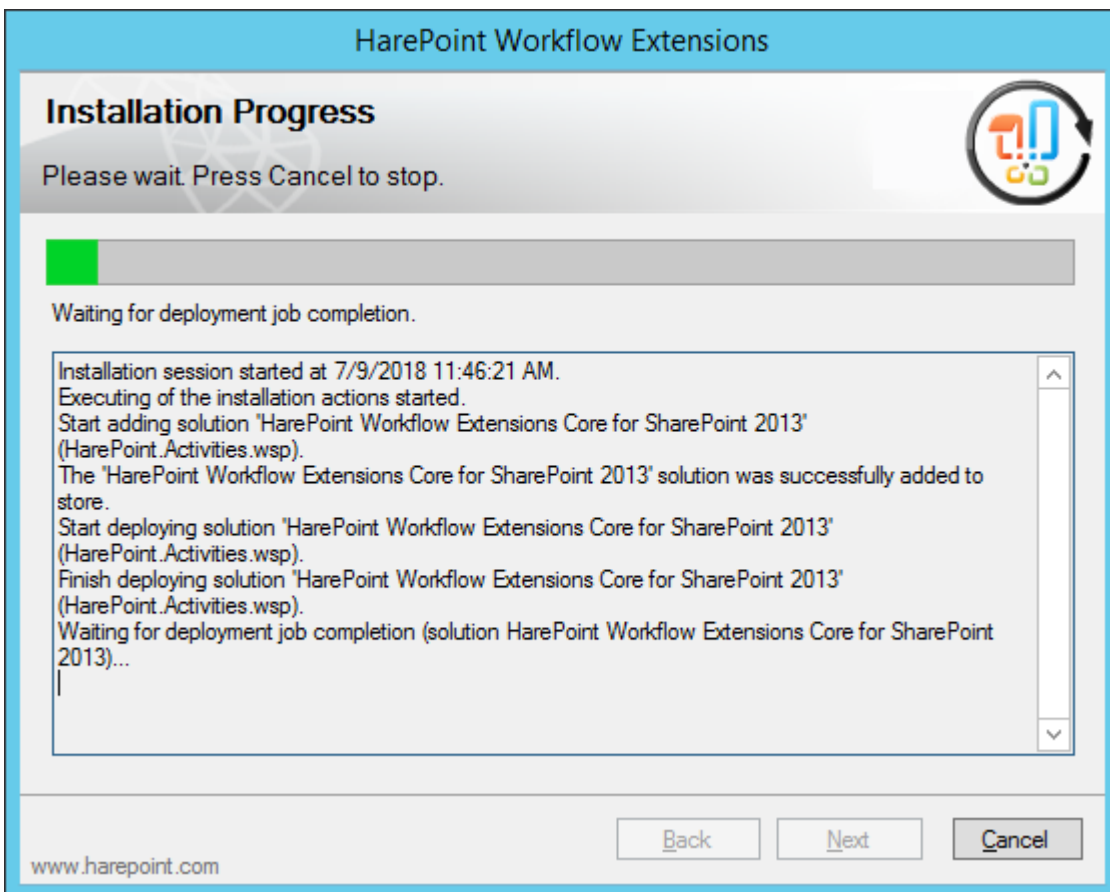
In case if any problems are detected, please fix them and restart the installation wizard.

At the **next step** of the Wizard, the solutions deployment will start.

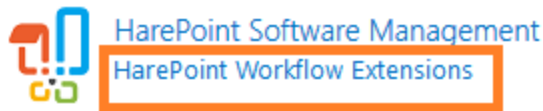
The following solutions will be installed:

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

- **HarePoint Printing Service Application** – service application that receives printing jobs from workflow action and transfer them to HarePoint Printing Service.
- **HarePoint Document Conversion Service Application** – service application that receives InfoPath to PDF conversion jobs from a workflow action and transfers them to HarePoint Document Conversion Worker Service.
- **HarePoint Workflow Extensions Printing Activities** – set of activities that allows printing from within a workflow.
- **HarePoint Workflow Extensions Document Conversion Activities** – “Convert InfoPath to PDF action” to convert InfoPath forms to PDF document.



When deployment is successfully completed, open Central Administration site.
Click the “**HarePoint Workflow Extensions**” link under **HarePoint Software Management** section:



If you have the permanent license key, please enter the it in the “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.

HarePoint Workflow Extensions Licensing

Licenses

Specify license keys for each SharePoint server where Microsoft SharePoint Foundation Workflow Timer Service is running.

Version 2.12.2743.1098

Trial version: 30 days left.

1 server detected where the program has to be licensed.

Note: deactivation of “HarePoint Workflow Extensions” in the “Manage Features” section of the Web Application Management in Central Administration **will not** result in deactivation of product features:

Name	Status
Apps that require accessible internet facing endpoints	Activate
Allow users to acquire apps that require internet facing endpoints from the SharePoint Store.	Activate
Document Sets metadata synchronization	Deactivate Active
Provides the infrastructure to synchronize metadata for Document Sets.	Deactivate Active
HarePoint Document Conversion Service Activities	Deactivate Active
HarePoint HelpDesk WebApplication Timer Jobs	Deactivate Active
HarePoint Printing Activities Feature	Deactivate Active
HarePoint Workflow Extensions	Deactivate Active
Install and activate HarePoint Workflow Extensions	Deactivate Active
SharePoint Server Enterprise Search	

The forced deactivation of full or partial functionality of the products is described in [chapter 8, Section IV](#) of this Guide.

4. Installing Client Applications

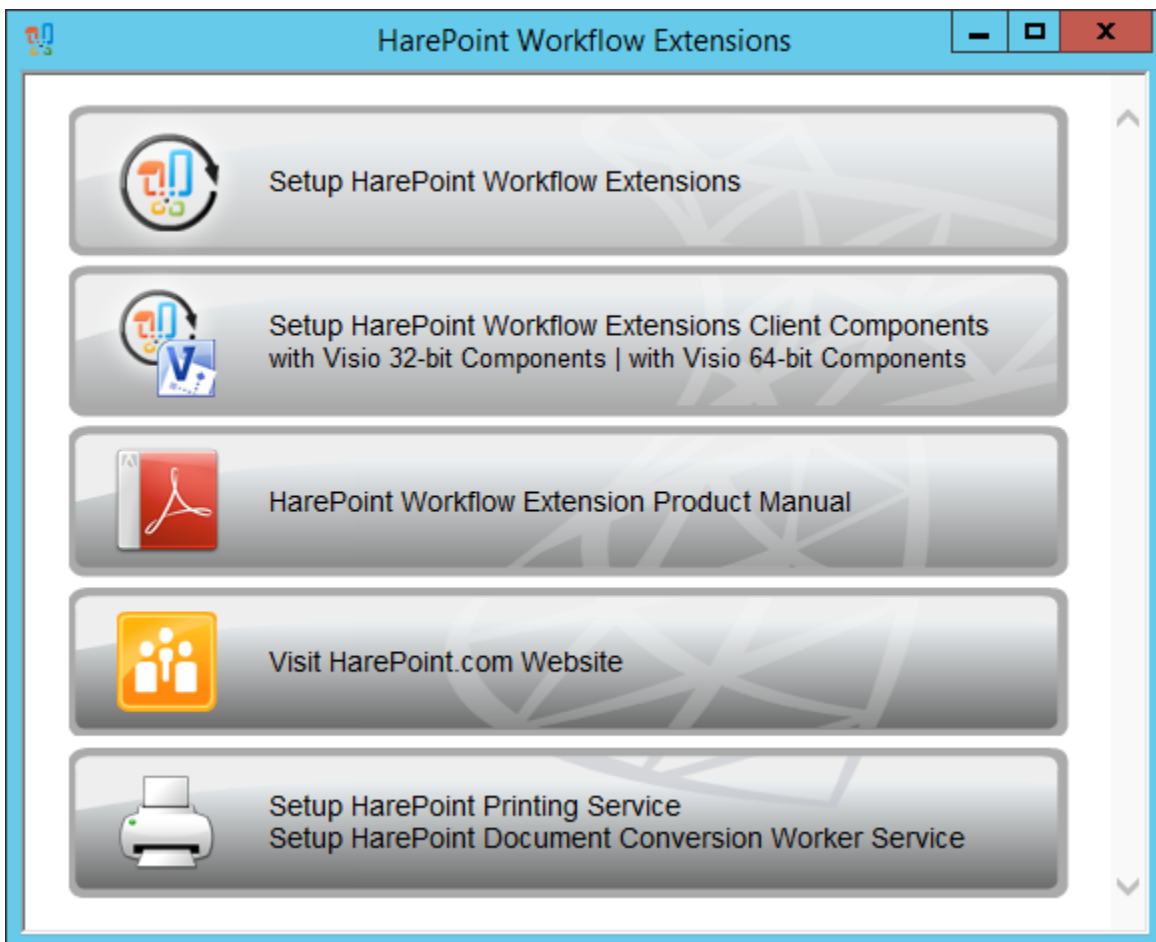
Note: Installation of Client Applications is **not** required to be able to use HarePoint Workflow actions in SharePoint Designer, since it is fully a server-side product.

Client Components are only auxiliary tools that may be needed but not required to build workflows using HarePoint Workflow Extensions.

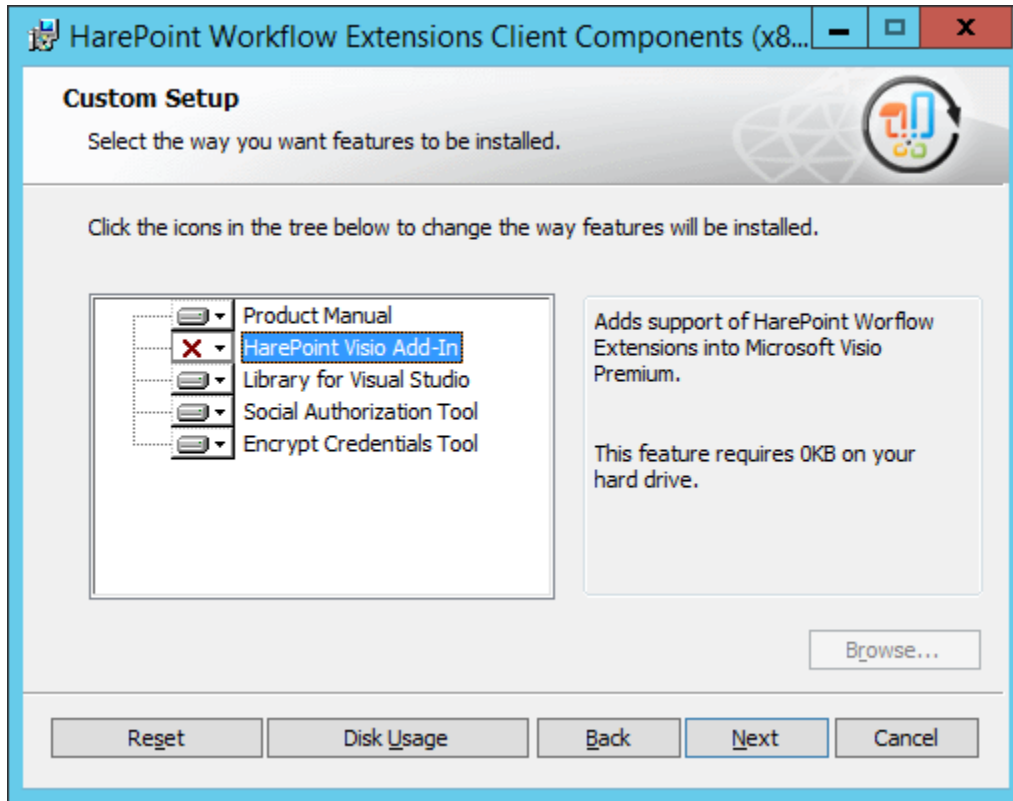
Client Extensions are normally installed not on a server, but on a workstation.

Log in to the Workstation; unpack the product archive to a folder on the local disk.

Run SETUP.EXE in the selected folder. The following window will be opened:

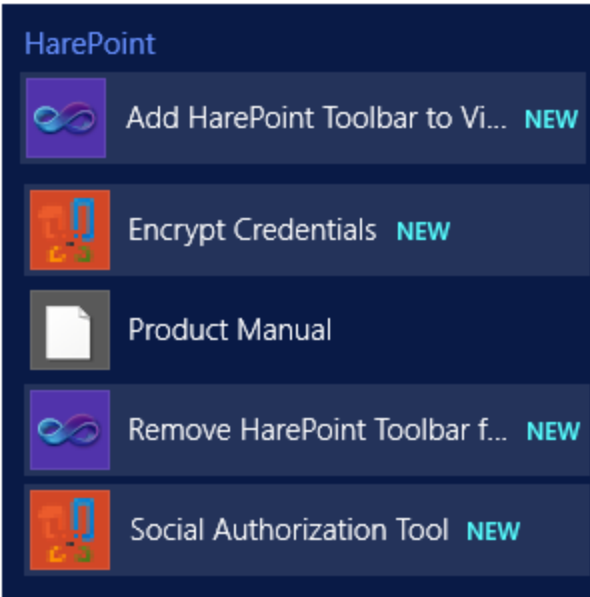


Click the “Setup HarePoint Workflow Extensions Client Components” item.
Installation Wizard will be launched.
Follow instructions of the Wizard to select the required Client Applications:



Then follow instructions of the Wizard and complete the installation.

The installed Client Applications are now available in **Start** menu - **HarePoint** section:



More details on how to use Client Applications are in [Section IV Product Overview](#) of this Guide.

III. How to upgrade, repair, or remove product

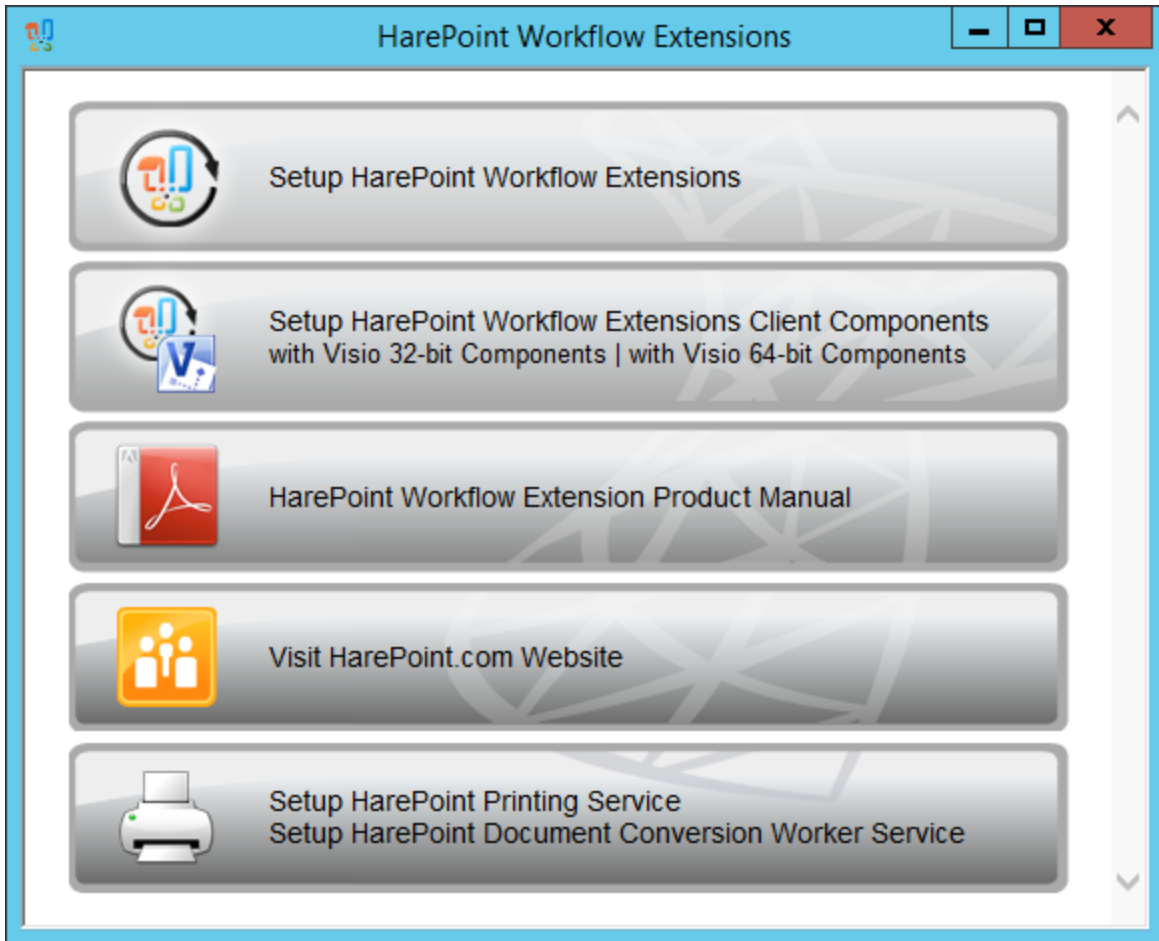
1. Upgrading, repairing, or removing HarePoint Workflow Extensions

To repair, upgrade or remove the product you will need SharePoint Farm Administrator permissions.

Login to any Web Front-End server of the SharePoint farm, unpack the product archive to a folder on the local disk.

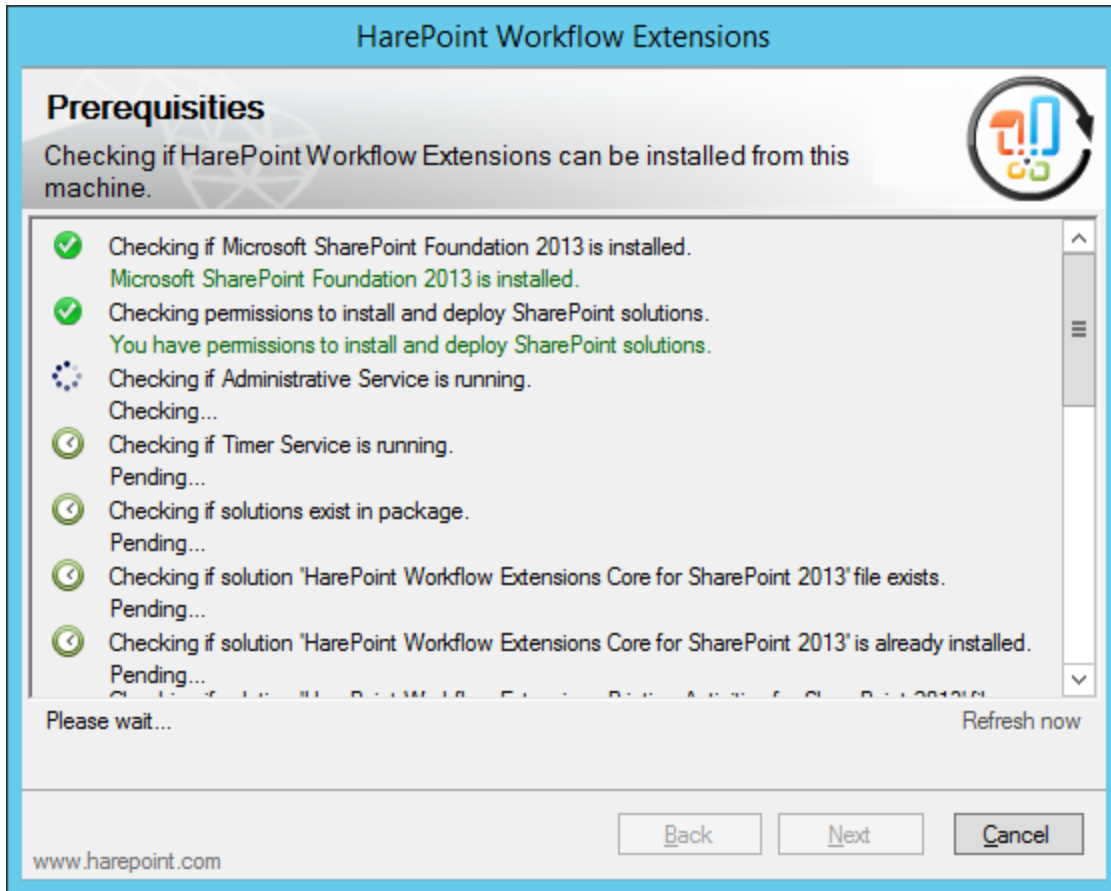
Run SETUP.EXE in the selected folder.

The following window will be opened:



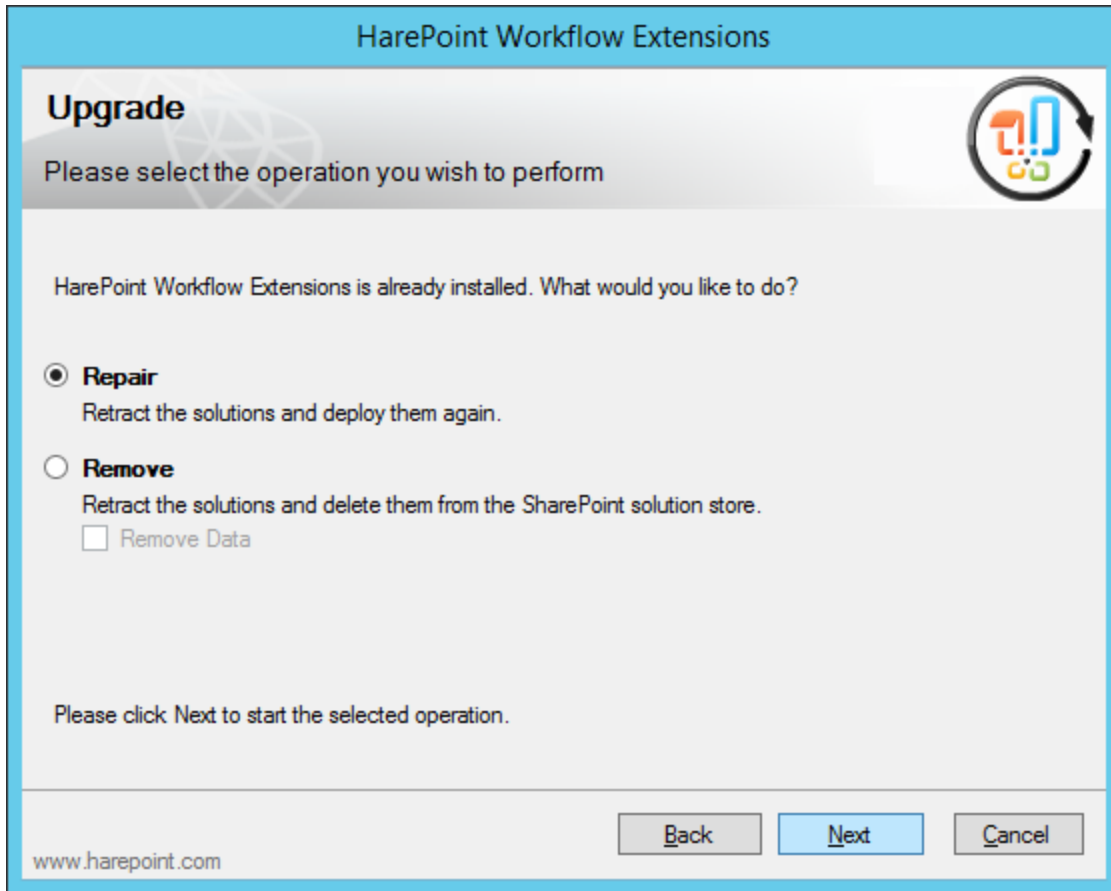
Select “**Setup HarePoint Workflow Extensions**” item. Installation Wizard will be launched.

Press **Next** button on the wizard’s welcome screen and prerequisites check will start:



In case if any problems are detected, please fix them and restart the installation wizard.

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

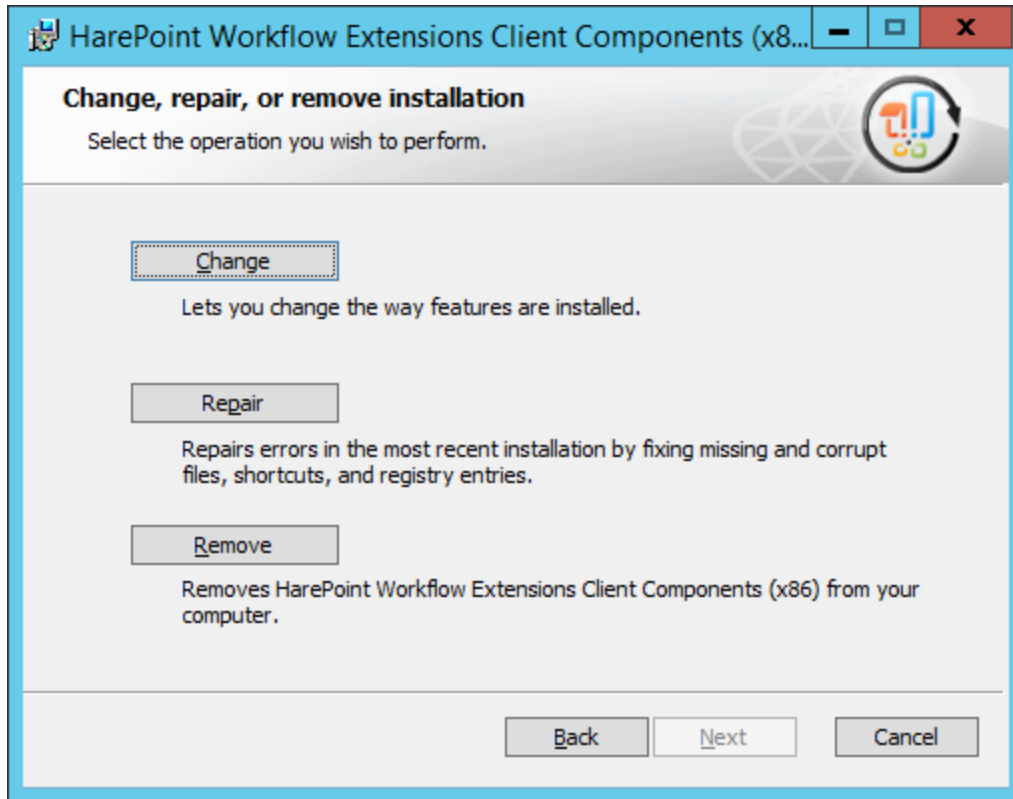


2. Upgrading, repairing, or removing Client Applications

To repair, upgrade or remove a Client Application, login to a machine where the Client Applications are installed, unpack the product archive to a folder on the local disk. Run SETUP.EXE in the selected folder. The following window will be opened:



Click “**Setup HarePoint Workflow Extensions Client Components**” item. Installation Wizard will be launched. Follow the instructions of the Wizard, select the required Client Applications and type of operation:



Follow the further instructions of the Wizard and complete the installation.

IV. 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.

SharePoint 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 consists of six 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 Visio². Installed on the workstation where it requires use of SharePoint Workflow Extensions activities in Microsoft Visio. Optional.
- The library for Visual Studio. Installed on the workstation where it requires use of SharePoint Workflow Extensions activities in Visual Studio. Optional.
- Utility to encrypt credentials. Used to provide logins and passwords not in clear text, but in a secure way, using the irreversible encryption algorithm. Optional.
- Utility for Social networks. Installed on the workstation, required for generation of an authorization token. Optional.

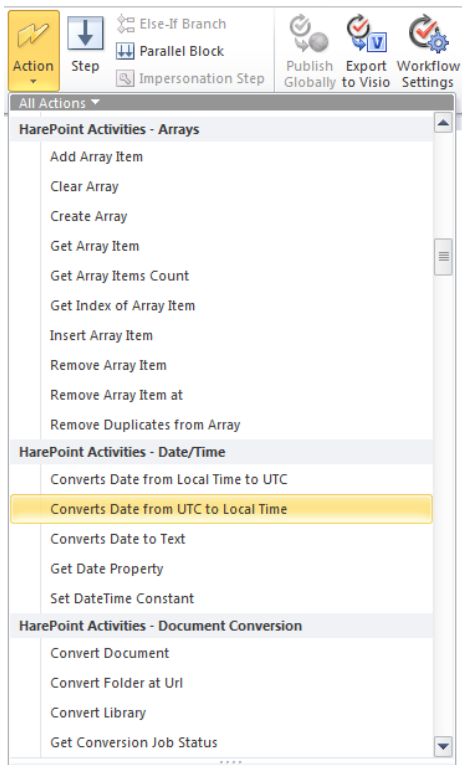
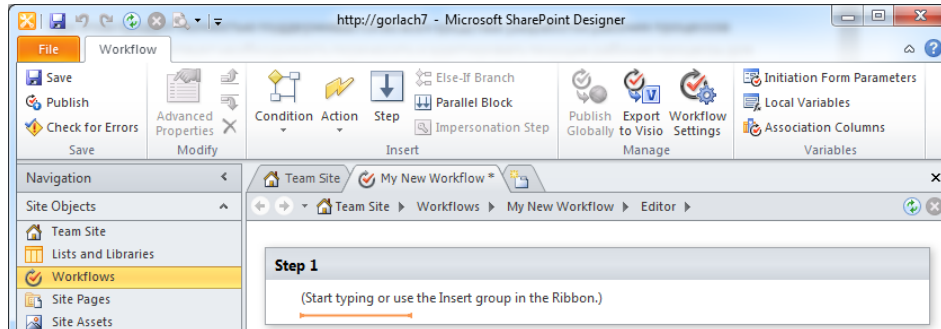
Note: For the basic functionality - to develop workflows in SharePoint Designer, only server-side components of SharePoint Workflow Extensions need to be installed; no client-side components are required to be installed on workstations for this.

² Current version supports Microsoft Visio 2010 only; Visio 2013 integration will be added in the nearest release

2. Creating workflows in SharePoint Designer

Open SharePoint Designer, and connect to the site.

Create a new workflow (for **2010 workflow platform**) or open an existing 2010 platform 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 V of this user manual.

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

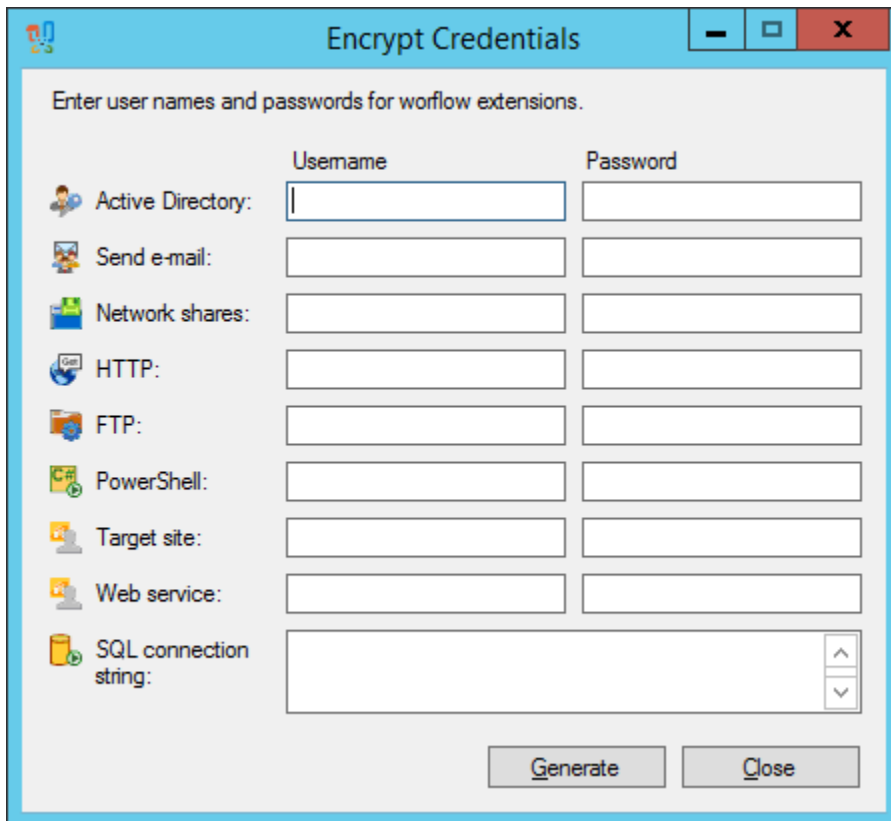
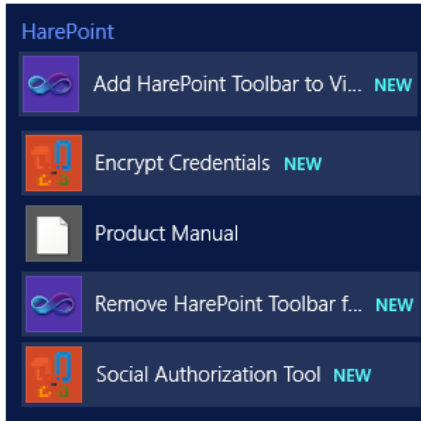
Please note that a number of activities and conditions are context-sensitive and sometimes you may not find them in the menu. For example, the **“Create Document from File”** activity is available only when creating a workflow for the Document Library and it won't be listed in the menu when a workflow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

3. Encrypt Credentials tool

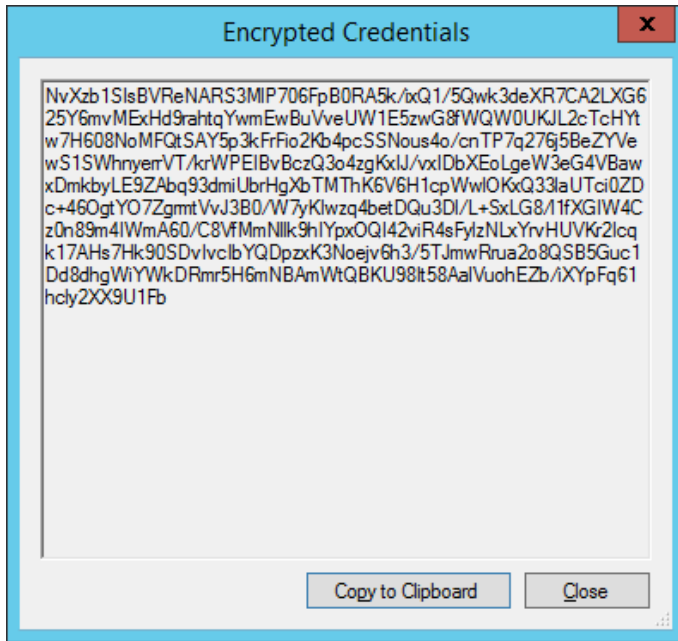
In order to securely provide credentials to various services in HarePoint workflow actions, you can encrypt them using the **Encrypt Credentials tool**. The tool will return a hash code (token) that you would

need to paste to **Set Encrypted Passwords** action that should be placed before any other actions that required authentication to services.

Encrypt Credentials tool is installed as a part of [Client Components](#), and available under Windows **Start** menu:



Put credentials or SQL connection string to corresponding fields, and then click **Generate**.



Click **Copy to Clipboard**.

Open your workflow in SharePoint Designer.

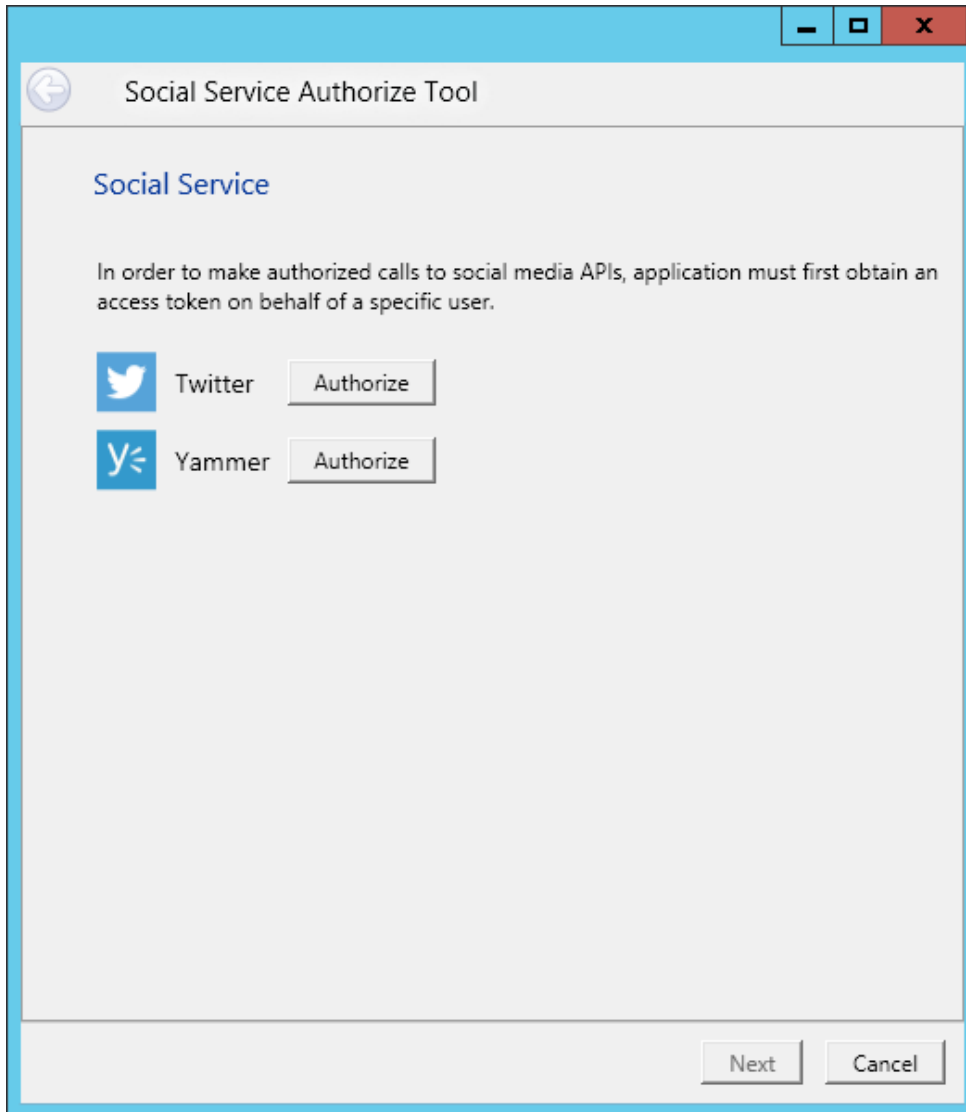
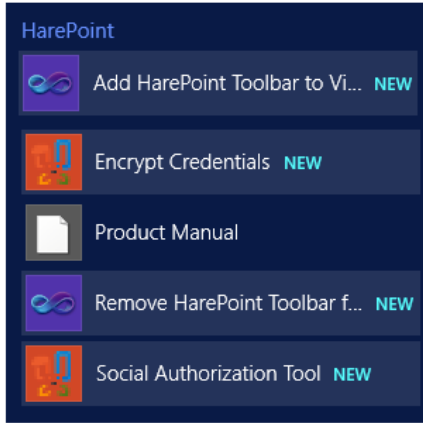
Add **Set Encrypted Passwords** action before the corresponding action that requires authentication and paste the hash code to its parameter:

Set encrypted passwords with NvXzb1SIsBVRenARS3MIP706FpB0RA5k/ixQ1... string.

4. Social Authorization tool

In order to be able to interact with social medias or services (such as **Twitter** and **Yammer**) from HarePoint actions, it is necessary to obtain tokens for these services. This can be done with **Social Service Authorize** tool.

Social Service Authorize tool is installed as a part of [Client Components](#), and available under Windows **Start** menu:



Click **Authorize** button next to the service or media you need to authenticate to. Follow the instructions.

Also, refer to **Send Twitter Message** and **Yammer** actions descriptions for details on obtaining and applying tokens to these actions.

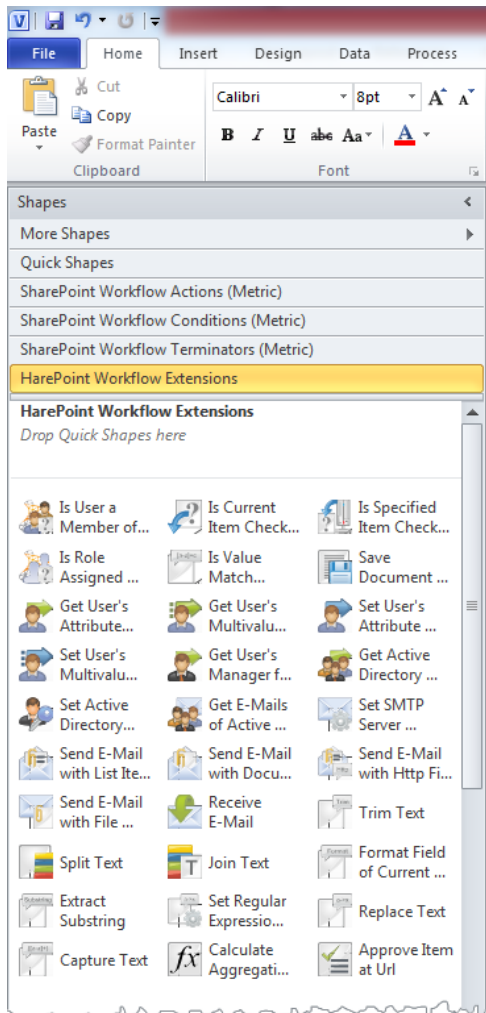
5. Designing workflows in Microsoft Visio

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

Note: current version supports Microsoft Visio 2010 only; Visio 2013 integration will be added in nearest releases.

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.

6. Creating workflows in Visual Studio

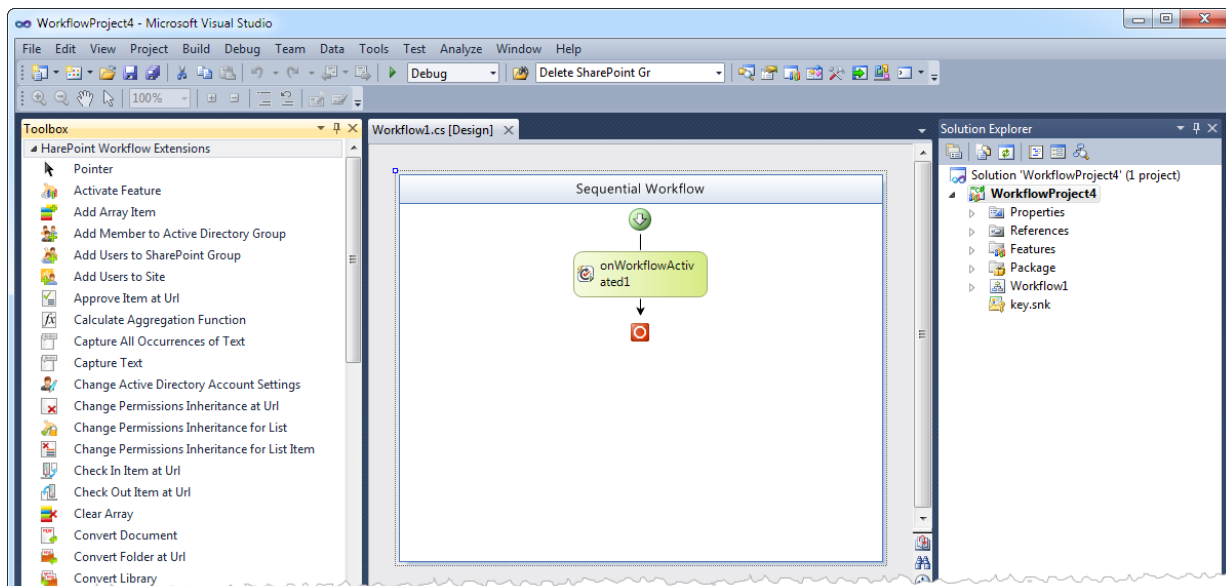
Creating workflows in Visual Studio is rather complex and requires deep knowledge of SharePoint architecture. Whenever possible, it is better to confine with “Execute Custom Code” activity or with development of a custom activity rather than create a workflow in Visual Studio.

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>

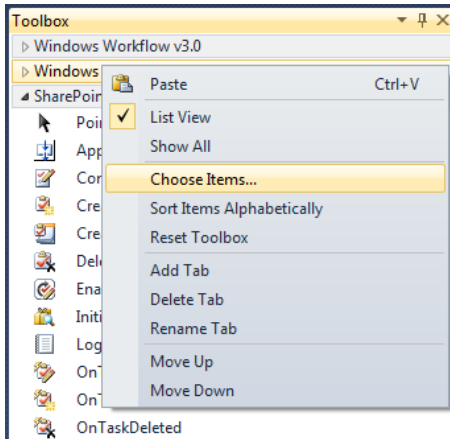
When developing a workflow in Visual Studio, you can use HarePoint activities and in-built SharePoint activities alike. For this, all components should be installed on the computer as required by the article “Requirements for Developing SharePoint Solutions” as well as HarePoint libraries for Visual 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

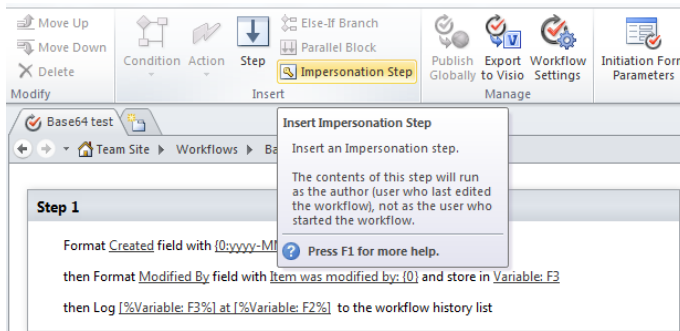
in SharePoint Designer and sorted by name by selecting the “Sort Items Alphabetically” item in the context menu. Please note that when components are added by a command “Add HarePoint toolbox to Visual Studio 2010”, they will be automatically renamed and sorted.

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

7. Security concerns

HarePoint activities are running in the context of the user who initiated a workflow. If you have created a workflow that starts automatically 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 8](#) of this Section.

8. 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.

V. HarePoint Activities and Conditions

This section contains the detailed description of actions and conditions of HarePoint Workflow Extensions. 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 is 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 one or several users or groups to specified Active Directory security or distribution group.

Mail contacts also can be added.

Note: In order to specify AD credentials, add the *Set Active Directory Authentication* activity to the same workflow step first. Otherwise, AD will be accessed as the user who started the workflow or as the workflow author in the case of an impersonation step.

Parameters:

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

To add **multiple users**, provide their logon names semicolon-delimited:
mapilab\test1;mapilab\test2;mapilab\test3

To add **mail contacts**, provide the e-mail address of a contact.

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, account expires: yes on this date.

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

See Also: *Create User 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 as the user who started the workflow or as the workflow author in the case of an 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.

account expires: yes - Select if the account expires: yes/no or keep existing settings. If Yes is selected, please specify the expiry date in the next parameter.

on this date - specify the expiry date of the account, if this account expires (previous parameter set to Yes).

1.3 Change Active Directory Group Settings

Change specified group settings, group type: keep existing, description: keep existing, email: keep existing.

This workflow action is used to change the Active Directory group settings.

Parameters:

specified group - Specify an AD group name that needs to be changed.

group type: keep existing - (optional) If a group type needs to be changed, select the new group type from a drop-down list.

The available options are:

- *universal security group*

- *global security group*
- *domain local security group*
- *universal distribution group*
- *global distribution group*
- *domain local distribution group*

description: keep existing - (optional) If a group description needs to be changed, specify the new group description.

email: keep existing - (optional) If a group email address needs to be changed, specify the new group email address.

1.4 Change Local User Account Settings on Server

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

This workflow action changes a local user account on a specified server.

Important Note: The *Set Active Directory authentication* action is required prior to using this action to specify the administrator account that will be used to change the properties of the local account.

Parameters:

logon name - Local account logon name.

this server - NetBIOS name, DNS name or IP address.

IPv4 and IPv6 addresses are supported.

Important note: if you need to change a user on a local machine, use *localhost* as a name!

enable - Enable or disable this local user account.

change password at next logon: keep existing

Options available: Yes, No, Keep existing.

password never expires: keep existing

Options available: Yes, No, Keep existing.

Examples:

Set Active Directory authentication (impersonate) using name COMPANY\Administrator and password 111.

Then Set Variable: logon name to test-n1

then Create new user on 192.168.2.16 server with Variable: logon name logon name, John Smith full name, Test user description, 123 password.

then Change Variable:logon name account settings on 192.168.2.16 : enable, change password at next logon: yes, password never expires: yes.

1.5 Create Group in Active Directory

Create new group in this container with specified group name.

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

See also: "Find Group in Active Directory" action can be used prior to this action to check if the AD group already exists.

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 as the user who started the workflow or as the workflow author in the case of an 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.

1.6 Create Local User Account on Server

Create new user on this server with specified logon name, this full name, this description, this password.

This workflow action creates a local user account on a specified server.

Important Note: *Set Active Directory authentication* is required prior to using this action to specify the administrator account that will be used to create the local account.

Parameters:

this server - NetBIOS name, DNS name or IP address.

IPv4 and IPv6 addresses are supported.

Important note: if you need to create a user on a local machine, use *localhost* as the name!

specified logon name -The logon name for the account.

this full name - (optional) The account full name.

this description - (optional) The description for the account.

this password - The password for account.

Examples:

Set Active Directory authentication (impersonate) using name COMPANY\Administrator and password 111.

Then Set Variable: logon name to test-n1

then Create new user on 192.168.2.16 server with Variable: logon name logon name, John Smith full name, Test user description, 123 password.

1.7 Create Mail Contact in Active Directory

Create new contact in this container with specified name, this display name, this alias, this external e-mail, mail enable: yes.

This workflow action creates a new mail contact with specified parameters.

Parameters:

this container - Specify the distinguished name or canonical name of the container where the contact will be created.

with specified name - Specify the name for the contact object in AD

this display name - Specify the name that will be displayed to the user.

this alias - (for mail-enabled contacts only) The internal e-mail address of the contact.

this external e-mail - External e-mail address (outside of organization).

mail enable: yes - Enable (or disable) receiving e-mails to this contact via Microsoft Outlook and Microsoft Exchange.

1.8 Create User in Active Directory

Create new user in **this** container with **specified** logon name, **this** first name, **this** last name, **this** password. Optional: use **this** as user principal name.

This workflow action creates a new user account in a specified Active Directory container. The created account will be enabled, and you can disable the account and change its settings with *Change Active Directory Account Settings*.

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 as the user who started the workflow or as the workflow author in the case of an 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).

Important note: Logon name is technically limited to **20 characters**. If you need to have a longer name, specify here only the first 20 characters of user logon name (or just any random characters) and use the optional parameter "user principal name" to specify a longer name, see below.

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* to ask the user to change the password at first logon.

use this as user principal name - (optional) Specify user principal name that can be used instead of user logon name.

Note: user principal name is technically limited to 64 characters.

1.9 Delete Contact from Active Directory

Delete contact with this e-mail address from Active Directory

This workflow action removes the contact from AD by specified e-mail address

Note: "Set Active Directory authentication" action needs to be placed in a workflow prior to this action.

Parameters:

this e-mail address - specify e-mail address of the contact to be removed (as corresponds to "mail" attribute)

1.10 Delete Group from Active Directory

Delete group with this group name from Active Directory.

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

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

Parameters:

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

1.11 Delete Local User Account from Server

Delete user with this logon name from this server.

This workflow action deletes the specified local user account from the server.

Important Note: *Set Active Directory authentication* is required prior using this action to specify the administrator account that will be used to create the local account.

Parameters:

this logon name - Local account logon name.

this server - NetBIOS name, DNS name or IP address.

IPv4 and IPv6 addresses are supported.

Important Note: if you need to delete a user on a local machine, use localhost as the name!

Examples:

Set Active Directory authentication (impersonate) using name COMPANY\Administrator and password 111.

Then Set Variable: logon name to test-n1

then Delete user with Variable: logon name logon name from 192.168.2.16

1.12 Delete User from Active Directory

Delete user with this logon name from Active Directory.

This workflow action will delete a 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 as the user who started the workflow or as the workflow author in the case of an impersonation step.

Parameters:

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

1.13 Find Group in Active Directory

Find this group in Active Directory and store logon name in Variable: variable.

This workflow action returns either the group logon name (if a group exists), or an empty string (if a group does not exist).

In particular, it can be used to find out if an AD group exists.

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 as the user who started the workflow or as the workflow author in the case of an impersonation step.

Parameters:

this group - Type in the AD group name.

store logon name in **Variable: variable** - Specify a variable to store the logon name (DOMAIN\group).

If an AD group does not exist, an empty string will be returned.

1.14 Find Objects in Active Directory by Query

Find objects in Active Directory by simple query using this query, search in entire forest and store distinguished names in **Variable: ArrayVariable**

This workflow action is used to search for various objects in AD.

To search for Active Directory users, refer to "Find User in Active Directory"

Note: In order to specify AD credentials, add the *Set Active Directory Authentication (impersonate)* activity to the same workflow step first.

Parameters:

simple query - One of the following query types:

- *simple query* (for .INI-style query)
- *LDAP filter* (for LDAP query)

this query - specify the search query

entire forest - choose to search in the *entire forest*, or only in *specified container*. If latter, specify the container by its distinguished name.

Variable: arrayVariable - specify an array variable that will contain the distinguished names of found objects.

1.15 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 the user's logon name in the output variable. The 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 as the user who started the workflow or as the workflow author in the case of an impersonation step.

Note: to search for any other objects in AD, refer to "Find Objects in Active Directory by Query" action.

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.

1.16 Find User in Active Directory by Query

Find user in Active Directory by simple query using this query and store logon name in Variable: variable.

This workflow action is used to find a user or group in Active Directory by specified query. You can use two type of queries.

First, the format in the style of ab INI file:

```
<AD_attribute_name1>=<value1>  
<AD_attribute_name2>=<value2>
```

For example:

```
displayName=John Smith
```

Second, in LDAP format. For example:

```
(&(objectCategory=user)(objectClass=user)(displayName=John Smith)(department=Marketing))
```

Parameters:

simple query - One of the following query types:

- *simple query* (for .INI-style query)
- *LDAP filter* (for LDAP query)

this query - String with the query.

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

1.17 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 as the user who started the workflow or as the workflow author in the case of an 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

1.18 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 returns the email addresses of an Active Directory group's members to an 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 as the user who started the workflow or as the workflow author in the case of an 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.

1.19 Get Object's Attribute from Active Directory

Get this attribute of object at this dn from Active Directory. Store attribute value in Variable variable .

This workflow action will return the value of an object's specified Active Directory attribute.

Note: In order to specify AD credentials, add the Set Active Directory Authentication (impersonate) activity to the same workflow step first.

See Also: *Get Object's Multivalued Attribute from Active Directory.*

Note: to get AD **user** attribute, refer to "Get User's Attribute from Active Directory" action

Parameters:

this attribute - specify the attribute name

object at this dn - specify the distinguished name of an object

Variable variable - specify the variable to store the attribute value

1.20 Get Object's Multivalued Attribute from Active Directory

Get multi-valued attribute this attribute of object at this dn from Active Directory. Store attribute value in Variable: ArrayVariable.

This workflow action will return the multivalued attribute of the specified object in Active Directory.

Note: In order to specify AD credentials, add the Set Active Directory Authentication (impersonate) activity to the same workflow step first.

See Also: *Get Object's Attribute from Active Directory.*

Note: to get AD **user** attribute, refer to "Get User's Multivalued Attribute from Active Directory" action

Parameters:

this attribute - specify the attribute name

object at this dn - specify the distinguished name of an object

Variable: ArrayVariable - Output array variable with attribute values.

1.21 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 the value of a user's specified Active Directory user 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 as the user who started the workflow or as the workflow author in the case of an impersonation step.

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

Note: to get attribute of any other AD object, refer to "Get Object's Attribute from Active Directory" action.

Parameters:

this attribute - Name of the attribute.

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

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

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

1.22 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 a specified Active Directory user or a 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 as the user who started the workflow or as the workflow author in the case of an impersonation step.

Parameters:

current user - The name of Active Directory user or a group whose manager's name is to be obtained (like DOMAIN\Account). Leave the field empty to use current user.

Variable: variable1 - Output variable for the obtained manager display name.

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

1.23 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 a multi-valued attribute of a specified Active Directory user.

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

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

Note: To get a mutli-valued attribute of any other object in AD, refer to "Get Object's Multivalued 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.

1.24 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 the specified user is a member of the specified Active Directory group.

You can specify optional credentials for Active Directory access. Otherwise, AD will be accessed as the user who started the workflow or as the workflow author in the case of an 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:

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.

1.25 Move User into Organizational Unit

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

This workflow action is used to move a specified AD user into a 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 as the user who started the workflow or as the workflow author in the case of an 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.

1.26 Remove Member from Active Directory Group

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

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

Mail contacts can also be removed.

Deleting multiple users is supported.

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 as the user who started the workflow or as the workflow author in the case of an impersonation step.

Parameters:

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

To delete **multiple** users, specify them delimiting by a semicolon (;).

Example: DOMAIN\user1; DOMAIN\user2; DOMAIN\user3

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

To remove a **mail contact**, specify its e-mail address.

Examples:

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

1.27 Set Active Directory Authentication

Set Active Directory authentication using user name this user and password this password, search in the current forest.

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 as the user who started the workflow or as the workflow author in the case of an 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:

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

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

current forest - Specify the search scope to find the necessary user. Possible values:

- current
- current and trusted
- custom - in this case, additionally specify the forest DNS name

1.28 Set Local User's Password on Server

Set logon name account password on this server to this password.

This workflow action provides the capability to change the local user account password on a specified server.

Important Note: *Set Active Directory authentication* is required prior to using this action to specify the administrator account that is used to create the local account.

Parameters:

logon name - Local account logon name.

this server - NetBIOS name, DNS name or IP address.

IPv4 and IPv6 addresses are supported.

Important Note: if you need to set a password for a user on a local machine, use *localhost* as the name!

this password - New password.

Examples:

Set Active Directory authentication (impersonate) using name COMPANY\Administrator and password 111.

then Set Variable: logon name to test-n1

then Create new user on 192.168.2.16 server with Variable: logon name logon name, John Smith full name, Test user description, 123 password.

then Set Variable: logon name account password on 192.168.2.16 to 321.

1.29 Set Object's Attribute in Active Directory

Set this attribute of object at this dn in Active Directory to this value

This workflow action is used to set an Active Directory attribute of a specified object.

Note: In order to specify AD credentials, add the Set Active Directory Authentication (impersonate) activity to the same workflow step first.

See Also: *Set Object's Multivalued Attribute in Active Directory.*

Note: In order to set AD user's attribute, refer to "Set User's Attribute in Active Directory" action

Parameters:

this attribute - specify the name of an attribute to be set

object at this dn - specify the object by its distinguished name

this value -specify the new value for the attribute

1.30 Set Object's Multivalued Attribute in Active Directory

Set multi-valued attribute this attribute of object at this dn in Active Directory to this value .

This workflow action is used to set a multi-valued attribute for a specified AD object.

Note: This action uses an array variable. Use *Create Array* or *Split Text* to create an array variable.

Tip: This action can be used to set AD attribute to empty (null). To do this, create an empty array using *Create Array*, then set the AD attribute to this array.

See Also: *Set Object's Attribute in Active Directory.*

Note: In order to set a mutli-valued attribute of an AD user, refer to "Set User's Multivalued Attribute in Active Directory" action.

Parameters:

this attribute - Name of attribute to be set

object at this dn - specify an AD object by its distinguished name

this value - specify an array variable with attribute values

1.31 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 set an Active Directory attribute of a specified 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 as the user who started the workflow or as the workflow author in the case of an impersonation step.

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

Note: In order to set an attribute for any other object in AD, refer to "Set Object'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 - Value for attribute.

1.32 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 set a multi-valued Active Directory attribute for a specified user.

Note: This action uses an array variable. Use *Create Array* or *Split Text* to create an array variable.

Tip: This action can be used to set AD attribute to empty (null). To do this, create an empty array using *Create Array*, then set the AD attribute to this array.

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

Note: In order to set multivalued attribute of any other object in AD, refer to "Set Object'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 - Array variable with values.

1.33 Set User's Password in Active Directory

Set logon name account password in Active Directory to this password.

This workflow action is used to set the password for a specified Active Directory user.

Parameters:

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

this password - String with new password.

Examples:

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

then Set MAPILAB\testuser account password in Active Directory to newpassword.

1.34 Set User's Photo in Active Directory

Set current user's photo in Active Directory to photo at this url.

This workflow action is used to set or remove a photo in the Active Directory user's profile (thumbnailPicture attribute). JPG image format is required.

Note: The size of photo is limited to 100KB.

Parameters:

current user - Leave intact if you want to change the photo of the current user or select a specific user.

this url - Full URL of photo. You can type "remove" or "delete" instead of URL to remove the current user's photo.

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 a feature in a SharePoint site or site collection.

Parameters:

this feature - Feature GUID or name, as 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 adds one or more users to one or more SharePoint groups.

Parameters:

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

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

This parameter also accepts **arrays of groups**. In this case, user(s) will be added to all groups from the array.

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 a SharePoint site.

Parameters:

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

this site - Full URL of the site.

2.4 Change SharePoint Group Owner

Change specified group owner at current site url to current user account.

This workflow action is used to change the owner of a SharePoint group.

Parameters:

specified group - Specify the SharePoint group name

current site url - (optional) Specify the address of the site where the group should be searched for.

current user - (optional) specify the name of the account or group that should be set as a group owner.

2.5 Change SharePoint Group Settings

Change specified group settings at current site url, about me: keep existing, view membership: keep existing, edit membership: keep existing, allow membership requests: keep existing, auto-accept requests: keep existing.

This workflow action is used to change the settings of a SharePoint group.

Parameters:

specified group - Type in the name of SharePoint group to change.

current site url - Keep it default to point to a current site, or specify the URL of a different site where the SharePoint group belongs to.

about me: keep existing - (optional) Specify the new text for group description.

view membership: keep existing - (optional) Specify who can view members of this group:

- *keep existing*
- *group members*
- *everyone*

edit membership: keep existing - (optional) Specify who can edit group membership:

- *keep existing*
- *group owner*
- *group members*

allow membership requests: keep existing - (optional) Specify if membership requests are allowed:

- *keep existing*

- *yes* - in this case additional parameter to this email appears
- *no*

auto-accept requests: keep existing - (optional) Specify if membership requests should be automatically accepted:

- *keep existing*
- *yes*
- *no*

2.6 Change SharePoint List Settings

Change list settings at this list url, require approval: keep existing, require checkout: keep existing, create versions: keep existing, enable content types: keep existing, open documents: keep existing

This workflow action is used to set the additional settings of a SharePoint list or a document library.

Parameters:

this list url - Specify the full path to the list or document library, which settings need to be changed.

require approval - Specify whether new items or changes to existing items should remain in a draft state until they have been approved.

Keep existing, yes, no

require checkout - Specify whether users must check out documents before making changes in this document library.

Keep existing, yes, no

create versions - Specify whether a version is created each time you edit a file in this document library.

Keep existing, major, major and minor

enable content types - Specify whether to allow the management of content types on this document library.

Keep existing, yes, no

open documents - Specify whether browser-enabled documents should be opened in the client or browser by default when a user clicks on them.

Keep existing, client application, browser, server default

2.7 Create Host Named Site Collection

Create new host named site collection with specified title and with this unique url at this web application url, using this template and specified description, compatibility level default. Site collection administrators: primary and secondary. Activate these features.

This workflow action is used to create a new host named site collection.

Note: You may need to place this action into an Impersonation Step. Otherwise, the action will be executed as the user who started the workflow.

Note: To create a regular site collection, use **Create Site Collection** action.

Parameters:

specified title - Title of new site collection.

with this unique url - The full absolute path to the new site collection.

Example:

http://blog.harepoint.com

this web application url - Specify the URL of the web application where the new host named site collection will be created.

this template - Specify a template for the new site. This can be

- the **name** of a site template (e.g. Team Site, Blank Site, Blog, etc.)
- the **ID** of a site template (e.g. STS#0, STS#1 and so on).

specified description - Type a description for the new host named site collection.

compatibility level default - (optional) Set the compatibility level for the site creation template.

Important note: this parameter works for SharePoint 2013/2016 only.

The list of available templates and their compatibility level can be obtained via **Get-SPWebTemplate** PowerShell cmdlet.

primary - Specify the Login name of administrator for this site collection. Only one user login can be provided. Security groups are not supported.

secondary - (optional) Specify a secondary site collection administrator. Only one user login can be provided. Security groups are not supported.

these features - (optional) Specify site collection features to be activated after applying a root site template.

To point to a feature, use the **feature ID**, e.g. {aebc918d-b20f-4a11-a1db-9ed84d79c87e}

2.8 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 creates a new SharePoint group. You can add members to the created group with *Add Users to SharePoint Group*.

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 listed in 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*.

2.9 Create SharePoint Site

Create new site with specified title at this url, using this template and this description, use unique permissions No. Store result in Variable: wasCreated.

This action creates a new site anywhere within the SharePoint environment.

You don't need to check in advance if the site exists, the action will check it automatically - see description for *Variable: wasCreated*.

You can however use a standalone condition to check if a SharePoint site exists - see also "**If SharePoint Site Exists**".

Parameters:

specified - Title of new site.
this url - Full URL for new site.
this template - Template for the new site. It might be the name of a site template, e.g. Team Site, Blank Site, Blog, etc., or it might be the ID of a site template, e.g. STS#0, STS#1 and so on.
Note: In case if the specified template is not found , the workflow will end with an error, and the workflow history log will contain a list of all available templates for troubleshooting purposes. Custom templates look like <i>{GUID}#Name</i> .
this description - New site description.
No - By default the new site inherits permissions from the parent Site or Site Collection. Unique permissions can also 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.
Variable: wasCreated - equals "True" if the site didn't exist and was created; equals "False" if the site already existed (no exception is thrown in this case).

2.10 Create Site Collection

Create new site collection with specified title at this url, using this template and specified description, compatibility level default. Site collection administrators: primary and secondary. Activate these features.

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

Note: You may need to place this action into an Impersonation Step. Otherwise, the action will be executed as the user who started the workflow.

See also: If you need to create a host named site collection, please use **Create Host Named Site Collection** action.

Parameters:

specified title - Title of new site collection.
--

this url - URL of new site collection.

this template - Template for the new site. This can be:

- the **name** of a site template (e.g. *Team Site, Blank Site, Blog, etc.*)
- the **ID** of a site template (e.g. *STS#0, STS#1* and so on)

specified description - Type the description for the new site collection.

compatibility level default - (optional) Set the compatibility level for the site creation template.

Important Note: This parameter only works for SharePoint 2013/2016 only.

The list of available templates and their compatibility level can be obtained via **Get-SPWebTemplate** PowerShell cmdlet.

primary - Specify the Login name of administrator for this site collection. Only one user login can be provided; security groups are not supported.

secondary - (optional) Specify a secondary site collection administrator. Only one user login can be provided; security groups are not supported.

these features - (optional) Specify site collection features to be activated after applying a root site template.

To point to a feature, use the **feature ID**, e.g. `{aebc918d-b20f-4a11-a1db-9ed84d79c87e}`

2.11 Deactivate Feature

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

This workflow action deactivates a feature on SharePoint site or site collection.

Parameters:

this feature - Feature GUID or name, as displayed in "Site Settings \ Manage site features" or in "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.

2.12 Delete Entire SharePoint List at Url

Delete the SharePoint list at this url.

This workflow action is used to delete the entire SharePoint list with all the list items completely.

Important note: The indicated SharePoint list will be deleted permanently and cannot be restored afterwards! If you need to remove just a single item in a list, use another action "Delete List Item at URL".

Parameters:

this url - Specify full URL of a SharePoint to be deleted.

2.13 Delete SharePoint Group

Delete specified group from this site url.

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

Parameters:

specified - Name of group.

site url - Full URL of the site.

2.14 Delete SharePoint Site

Delete site at this url.

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

Parameters:

this url - Full URL of site to delete.

2.15 Delete Site Collection

Delete site collection at this url.

This workflow action is used to delete a site collection.

Note: This action may need to be placed in an Impersonation Step. Otherwise, action will be executed as the user who started the workflow.

Parameters:

this url - Full URL to site collection.

2.16 Extract Users from SharePoint Group to List

Extract users from **this group** to the list at **this url**, use **specified** list column to user property mapping.

This workflow action is used to extract users from the given SharePoint Group and save them to a SharePoint list as individual list items.

You can customize the user properties that will be saved to a list - see third parameter description below.

Parameters:

this group - Specify the SharePoint Group name.

list at this url - Specify the full URL of the list where information for each extracted user will be saved as a separate list item.

specified "list column to user property" mapping - Specify how to map list columns to user properties.

The format:

list column=user property

If you need to obtain a property directly from User Information List (and not from User Profile), put the property name into **square brackets []**

You can use either **Internal name** or **Display name** to point to a property you need. Example: *SPS-JobTitle* or *Job Title*.

Example:

Title=UserName
Email=[Email]
Job Title=SPS-JobTitle
About=About me

for the list that contains *Title, Email, Job Title, About* fields

2.17 Get Multiple Site Users Property

Get (Custom) specified property for these users from user info list and store result in Variable:array.

This workflow action gets the values of one specified property from all specified users and saves them to an array variable.

Parameters:

(Custom) - Select the property from the drop-down list, or select (Custom) and specify the property display name in the next parameter.

specified - Specify the property internal name. The list of default properties names is available here: [http://technet.microsoft.com/en-us/library/hh147513\(v=office.14\).aspx](http://technet.microsoft.com/en-us/library/hh147513(v=office.14).aspx)

these users - Users whose property should be obtained.

user info list - Select the source: User Information List, or Profile Service (not available on SharePoint Foundation). Note that the workflow account should have permissions to read from User Profile Service as follows: Central Administration -> Application Management -> Manage Service Applications -> User Profile Service -> Administrators

Variable:array - Output array variable.

Examples:

Get Manager property for VM1\User1; VM1\User2 from user info list and store result in Variable:array.

2.18 Get Site Property

Get Title property for site at this url and store result in Variable: variable.

This workflow action is used to get one of the available site properties.

Parameters:

Title - Select which site property should be returned.

The available properties are:

- *Created* - date and time of site creation in UTC
- *Description* - site description
- *Last item modified* - date and time in UTC when anything on that site has been modified
- *Primary site collection administrator* - login name of a site collection administrator, as seen in Central Administration
- *Request access e-mail* - e-mail address used to request access to this site
Secondary site collection administrator - login name of a secondary site collection administrator, as seen in Central Administration
- *Site collection administrators* - list of login names (semicolon delimited (;)) of all site collection administrators, as seen in root site settings: Site Settings - Permissions - Site Collection Administrators
- *Title* - the site title
- *Web template* - the name of the template that was used to create the site

site at this url - Specify the full URL of the site.

Variable: variable - Specify the variable to store the output.

2.19 Get Site User Property

Get **(Custom)** specified property for **this user** from **user info list** and store result in **Variable: variable**.

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

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

Parameters:

(Custom) - Select the property from the drop-down list, or select *Custom* and specify it in the next parameter.

specified - Property internal name.

this user - User whose property should be obtained.

user info list - select the source: User Information List, or Profile Service (not available on SharePoint Foundation).

Note: Workflow account should have permissions to read from User Profile Service as follows:

Central Administration -> Application Management -> Manage Service Applications -> User Profile Service -> Administrators

Variable: variable - Output string variable.

Examples:

Get About me property for Current Item:Created By from user info list 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

2.20 If SharePoint Group Exists (HarePoint)

If this SharePoint group exists.

This workflow condition is used to check if the specified SharePoint group exists.

See also: "Find User in Active Directory", "Find Group in Active Directory".

Parameters:

this - Specify the name of the SharePoint group to be checked.

2.21 If SharePoint Site Exists

If SharePoint site at this url exists.

This workflow condition is used to check whether a SharePoint exists at an indicated URL.

Parameters:

this url - Specify the URL where you need check if a SharePoint site exists.

2.22 Is Role Assigned to User (HarePoint)

If role is assigned to this user at the current site.

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

Parameters:

role - User role (combination of permission levels).

this user - User logon name.

current site - (optional) Specify the URL of the site, list or list item.

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

If this user is a member of this SharePoint group.

This condition allows checks whether a user has membership in a given SharePoint group.

Parameters:

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

this - SharePoint group name.

2.24 Remove User from SharePoint Group

Remove user with this login name from specified group at this site.

Using this workflow action, you can remove either a specified user, or all users from a SharePoint group.

Parameters:

User - Choose to remove a single user, or all users:

- *user*
- *all users*

With this login name - Specify the login name of a user to be removed. This parameter is not displayed if you've chosen to remove all users.

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

this - Full URL of a site.

2.25 Remove User from Site

Remove **specified** user from **this** site.

This workflow action removes a user from a SharePoint site and from all site groups.

Parameters:

specified - Logon name of user.

this - Full URL of a site.

2.26 Rename Site at Url

Rename site at **this url** to **this title**.

This workflow action sets a new title for a site specified by URL anywhere within the SharePoint environment.

Parameters:

this url - Full URL of site.

this title - New title.

2.27 Set Master Page

Set **this** master page as default for **the current** site.

This workflow action sets the master page for the current or any other site.

Parameters:

this masterpage - Specify the name of the masterpage, or server-relative URL to it.

Examples:

- *v4.master*

- *default.master*
- */_catalogs/masterpage/v4.master*

the current site - (Optional) Specify the absolute URL of the site to set the masterpage to. Leave it *current site* to set the masterpage for the current site.

2.28 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 a 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.

2.29 Set SharePoint Site Anonymous Access

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

This workflow action is used to set anonymous access for a 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.

2.30 Set SharePoint Site Navigation Options

Set options for site at this url, show on parent's Quick Launch: yes , show on parent's top link bar: yes , inherit parent's top link bar: yes.

This workflow action sets the navigation options for a SharePoint site.

Parameters:

site at this url - Specify the site URL to set the options for.

show on parent's Quick Launch: yes - Specify if you need to display or hide this site from Quick Launch of the parent site.

Available values: *yes, no, don't change (to leave as it currently is)*.

show on parent's top link bar: yes - Specify if you need to display or hide this site from Top Link Bar of the parent site.

Available values: *yes, no, don't change (to leave as it currently is)*.

inherit parent's top link bar: yes - Choose to display the Top Link Bar from the parent site, or display the own Top Link Bar for this site.

Available values: *yes, no*.

2.31 Set Site Property

Set this property for site at this url to this value.

This workflow action is used to change the site properties, such as title, description and site collection administrators.

See also: "Get Site Property" action.

Parameters:

this property - Choose the property to set from the drop-down list:

- *Title*
- *Description*
- *Primary site collection administrator*
- *Request access e-mail*
- *Secondary site collection administrator*
- *Site collection administrators*

site at this url - Specify the full URL of a site.

this value - Specify the new value for a chosen property.

For *Site collection administrators* property, provide list of administrators' login names, delimited by a semicolon (;).

To **add a new administrator**, first read the list of current administrators using **Get Site Property** action, then append the new administrator's login name (by a semicolon), then update the property using **Set Site Property** action.

2.32 Set Site User Property

Set value to specified property for this user.

This workflow action sets a specified property value in a user's profile on the site.

You can get a list of all available properties in Central Administration \ Manage User Properties.

See Also: *Set User's Multivalued Attribute in Active Directory*

Parameters:

value - Text, expression, item field or variable.

specified - Property internal name.

this user - User, for whose profile the property value is set.

Examples:

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

If Current Item:File Type equals jpg

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

2.33 Set User's Photo in SharePoint

Set current user's photo in SharePoint profile to photo at this url. Copy file to User Information List.

This workflow action sets or removes a photo in the SharePoint user's profile. JPG image format is required. The size of photo is limited to 1MB.

Parameters:

current user - Do not change this value if you want to change the photo of the current user. Specify a user only if the workflow is created to change only that users photo.

this url - Full URL of the photo. You can type "remove" or "delete" instead of a URL to remove the current user's photo.

Copy file - Select to copy/do not copy this photo to the User Information List.

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 the 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 a 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 an 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 the 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 the item at a specified index of an array.

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

Negative index values indicate 'from the end of the 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 the zero-based index of a specified item in an array variable.

Result -1 means that the item is not found.

Parameters:

item - Item to get index of.

array variable - Array variable.

Variable: variable1 - 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 inserts a new item into an array variable at a specified position. It can also insert a source array specified in the first parameter to the specified position of target array indicated in the second parameter.

Parameters:

item - New item or source array to insert.

array variable - Target array variable.

0 - 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 deletes the first occurrence of a specified item from an 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 removes the item located at the specified zero-based index position in an 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 deletes duplicated values in an 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 a 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 a date/time variable to a string using a specified format.

Detailed descriptions of format string syntax are contained in the 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 a wrongly formatted string, the date will be converted with the default format and a warning message will be stored in the 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:

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

4.4 Evaluate Due Date

Evaluate due date from this time for this many hours, this many minutes and store result in Variable: variable.

This workflow action is used to calculate a due date by adding a specified time period to a starting date and excluding non-working hours.

Parameters:

this time - Starting date/time.

this many hours - Duration in hours.

this many minutes - Duration in minutes.

Variable: variable - Output date/time variable.

4.5 Get Date Property

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

This workflow action is used to get a specified numeric value (like day of week or year) from a date/time variable.

See the 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

4.6 Get Interval Between Dates

Get minutes between this date and this date, exclude holidays from iCalendar file at this url, store interval in Variable:interval.

This workflow action calculates the interval between two specified dates in specified units and saves the result to a variable.

Parameters:

get minutes - select the unit for the time interval. Possible values: *minutes, hours, days, weeks, months, years*.

between this date - The starting date.

and this date - The ending date.

iCalendar file at this url - (optional) Specify the iCalendar file (.ics) that contains holidays that need to be excluded from the calculation.

They can be downloaded from <http://www.officeholidays.com/ics/> or exported from SharePoint, Google or Outlook calendars.

The file can reside either in SharePoint document library, or on an external site, e.g.

http://www.officeholidays.com/ics/ics_country_code.php iso=US

Variable:interval - The Number variable where result will be stored.

4.7 Get Working Time Between Dates

Get working days between this date and this date, store interval in Variable: interval.

This workflow action is used to calculate the working days or working hours between the two dates.

The working days and hours are taken from **Regional Settings** of a SharePoint site.

Parameters:

days - Choose to calculate *days* or *hours*.

In *days* mode, the time is not taken into account; both start and end days are included.

In *hours* mode, the time is taken into account.

Example: "Monday 00:00 till Friday 00:00" will return 32 working hours. In order to get 40 working hours, specify "Monday 00:00 till Friday 23:59".

this date - Specify the start date.

this date - Specify the end date.

Variable: interval - Specify a variable to store the output.

4.8 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 the example section for details.

Parameters:

Now - One of the 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, Beginning of last week, End of last week, Beginning of last month, End of last month, Beginning of last year, End of last 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 Convert JSON to XML

Convert this JSON to XML and store result in Variable: XMLOutput

This workflow action is used to convert a JSON string to an XML document.

Parameters:

this JSON - specify the JSON string to be converted.

Note: The JSON string should contain the root element.

Example:

```
{"table": {"item":  
[{"Column1": "Value1", "Column2": "12.03.2022", "Column3": "User1"},  
{"Column1": "Value2", "Column2": "10.02.2022", "Column3": "User2"}]  
}}
```

Variable: XMLOutput - specify the string variable to store the XML output

5.2 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 a workflow.

In the code, an object named Context with an IExecuteCustomCodeLink interface is available. With this object, you can easily get access to the workflow context, output a Result variable and to any other workflow variable, activity execution context, or 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 presents security risks. If you need to disable it for 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 a *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# - Choice 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.

It is possible to specify assemblies from Global Assembly Cache. It is recommended to use a full name:

Multiplier35, Version=1.0.0.0, Culture=neutral, PublicKeyToken=83a9124cb39a6085

but shorter names are also allowed:

Multiplier35 or Multiplier35.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

5.3 Execute PowerShell Script

Execute this PowerShell script and store result in *Variable: results*. Run as *application pool* account.

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

You can also run **.ps1** files - see Examples section for more details.

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

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

See examples section below.

Note: This workflow action presents security risks. If you need to disable it for 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 in code as \$Context.Result.

Run as application pool account - specify the account that will be used to run the PowerShell script. You can use **Encrypt Credentials** tool and *Set Encrypted Passwords* to provide the credentials for an alternative account in a secure way; this tool is installed as a Client Component using the HarePoint Workflow Extensions installer.

Examples:

Example 1:

Set encrypted passwords with sfDdv1yb+3... string

Execute \$x = \$Context.GetWorkflowVariable("v... and store result in Variable: results. Run as encrypted password account.

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", "sample_value")
Write-Host $Context.WorkflowContext.Web.Title
Get-ExecutionPolicy -Scope Process
Get-ExecutionPolicy -Scope CurrentUser
Get-ExecutionPolicy -Scope LocalMachine
Get-ExecutionPolicy
```

Example 2:

Run a **.ps1** file using this workflow action:

```
C:\test\test.ps1 $Context
```

In this case, the **\$Context** object will be available in a PowerShell script.

To use **\$Context** within this **.ps1** file, add this line to the beginning of this file:

```
$context = $args[0]
```

You can then refer to workflow variables in **.ps1** file as follows:

```
$context.GetWorkflowVariable("v1")
```

Example 3:

If you need to transfer **multiple parameters** to a **.ps1** script, use **space** as delimiter:

```
C:\test\test.ps1 $Variable1 $Variable2
```

To use them in the **.ps1** file, add this line to the beginning of this file:

```
$variable1 = $args[0]
```

```
$variable2 = $args[1]
```

Note: If the parameter itself contains a space, enclose in in **single quotes**:

```
C:\test\test.ps1 $Variable1 'two words'
```

5.4 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 commands on Microsoft SQL Server, Oracle, ODBC or OLE DB data sources.

For an 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 the 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*

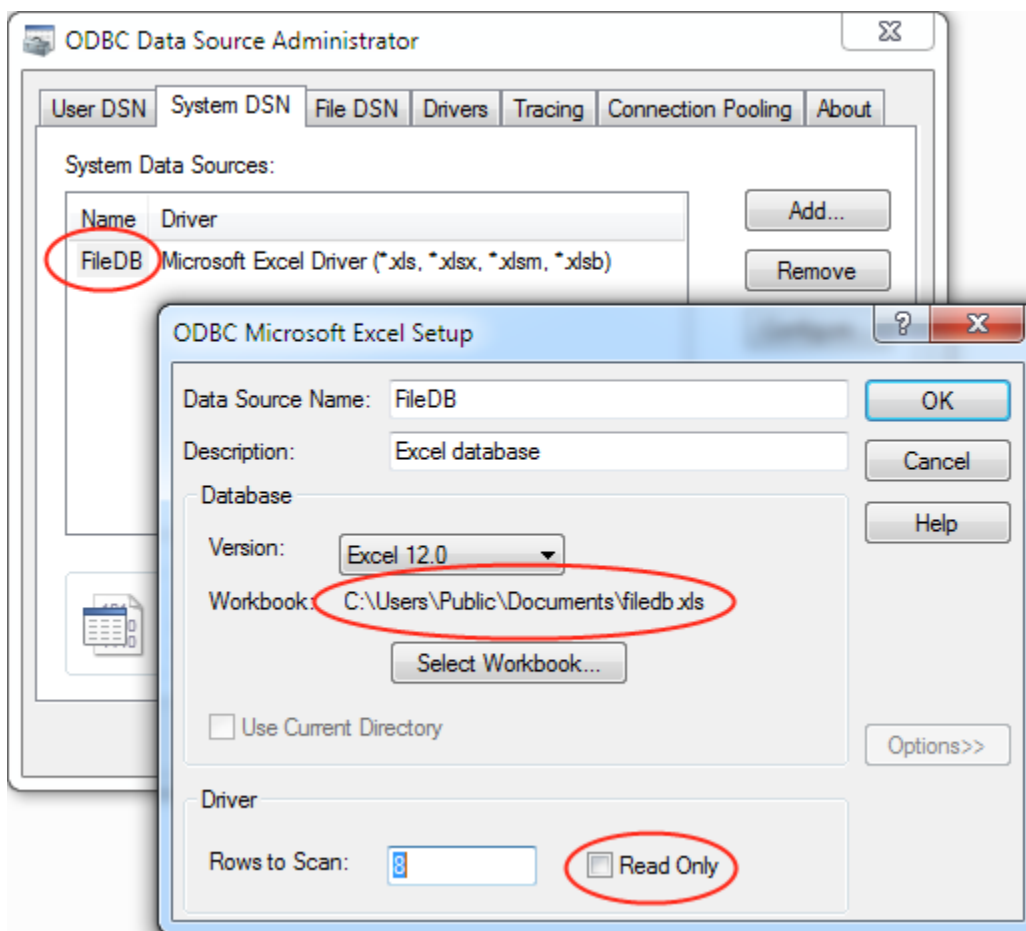
Get item at 1 index from *Variable: variable3* and store in *Variable: FirstModificationBy*

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

5.5 Execute SQL Command into List

Execute this SQL command using specified connection string for specified database and add results into list at this url using this column mapping, store number of records in Variable: recordsAffected.

This workflow action is used to execute SQL commands on Microsoft SQL Server, Oracle, ODBC or OLE DB data sources and output results to the specified SharePoint list.

Each row from the results represents a new list item.

You can specify an alternate column mapping in the fifth parameter.

For example, you have a list with two columns named *Title* and *Due Date*. And you have the following SQL query: *SELECT Name, Date from Table*. Then you can set the column mapping to the following:

Name=Title

Date=Due Date

Note: for SharePoint **lookup fields**, the data should be provided in the data source as '*ID;#Value*'

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.

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

this url - Full URL of the list where items should be added.

this column mapping - (optional) This parameter may be blank. Multiline string with alternate column mapping, each on new line, in the following format:

DB_Column_Name1=SP_Column_Name1

DB_Column_Name2=SP_Column_Name2

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

5.6 Query XML

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

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

See Also:

- XPath Language Reference: <http://www.w3.org/TR/xpath/>
- 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 of three available options:

- *text*
- *inner XML*
- *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 fetches the 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>
```

5.7 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: https://www.w3schools.com/xml/xsl_intro.asp
- Online XSLT Design
Tool: <http://www.w3schools.com/xml/tryxslt.asp> `xmlfile=cdcatalog&xsltfile=cdcatalog`

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:template>

</xsl:stylesheet>

```

Example of output HTML document:

European Central Bank: 2011-07-01

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

5.8 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 a node value or to append a new child node to an XML document using XPath query.

See Also:

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

Example 1: Update node

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 fetches an XML document with currency conversion rates from the European Central Bank site, increases the EUR/USD conversion rate by 10%, and generates an HTML document with currency rates in a document library.

Note that the type of EuroUsdNumber variable is Number. For the full text of the XSLT template and an example of output HTML, see the examples section of the *Transform XML* activity overview.

For an example of a source XML document, see the examples section of the *Query XML* activity overview.

Example 2. Append child node

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 example fetches an XML document with currency conversion rates from the European Central Bank site, adds a new node with a EUR/EUR rate of 1.0000, and generates an HTML document with currency rates in a document library.

For the full text of the XSLT template and an example of the output HTML, see the examples section of the *Transform XML* activity overview.

For an example of a source XML document, refer to examples section of the *Query XML* activity overview.

6. Document Conversion activities

6.1 Convert Document

Convert this document to PDF format, put it into the same library and store job ID in Variable: ConversionJobId. Overwrite existing document.

This workflow action converts a specified document to different format and stores the output file in the same document library.

This action processes files only of supported formats, files of other formats will be skipped. The list of supported input formats is:

- 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)

The list of supported output formats includes all input formats and two additional ones:

- Portable Document Format (.pdf)
- XML Paper Specification (.xps)

If the action returns an empty string in the ConversionJobId output variable, no files were found to convert.

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

See Also: *Wait Conversion Job* and *Get Conversion Job Status*.

Parameters:

this - Document to convert, e.g. Current Item.
PDF - Choose one of the available formats: <i>Word Document (docx)</i> , <i>Word 97-2003 Document (doc)</i> , <i>PDF</i> , <i>RTF</i> , <i>XPS</i> .
put into the same library - choose if an output document needs to be in the same library as the original document, or in a different one.
Variable: ConversionJobId - Output string variable with ID of conversion job. Action returns an empty string if no files were found for conversion.
Overwrite - If the folder already contains a file with the same name, then you can apply one of two options. <i>Overwrite</i> - replace the file with the new one, or <i>Do not overwrite</i> - skip this file.

Examples:

See example in *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 a document library specified by URL. Output files maybe stored in the same or in a different document library.

This action processes only files of supported formats, files of other formats will be skipped. The list of supported input formats includes:

- 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)

The list of supported output formats includes all input formats and two additional:

- Portable Document Format (.pdf)
- XML Paper Specification (.xps)

The action returns an empty string in ConversionJobId output variable if no files were found to convert.

Note: This action requires "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*.

Parameters:

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

PDF - Choose one of the 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 a file with the same name, then there are two options. *Overwrite* - replace file with new one. *Do not overwrite* - skip this file.

6.3 Convert Image to PDF

Convert image at specified url to PDF, put it into the same library and store url in Variable:PdfUrl. Overwrite existing file. Set margins to 3,2,2,2 cm.

This workflow action is used to convert an image to PDF file.

Multi page TIFF files are supported.

Parameters:

specified url - Specify the full URL of the picture to be converted

the same library - Choose if the output PDF should be placed to the same document library as the original picture, or specify a URL if a different target library is preferred.

Variable: PdfUrl - Specify the variable that will store the full URL of the output PDF document.

Overwrite - Choose to *Overwrite* or *Do not overwrite*, if the target document library already contains a PDF with the same name.

3,2,2,2 - Specify the margins, separated by a comma. The numbers stand for **left, top, right, and bottom** margins respectively.

If all margins are the same, you can specify only one number, e.g.:

... *Set margins to 2 cm*

6.4 Convert InfoPath Form and Save to Document Library

Convert InfoPath form at list item url to PDF and save to document library at library url. Default InfoPath form view will be converted.

This workflow action converts a completed InfoPath form in a specified Form Library item to PDF and saves to a specified Document Library.

Important note: This action requires additional setup before deploying. Please refer to the '*InfoPath to PDF Service Installation Guide*' and '*InfoPath to PDF How to use*' guide for detailed information. These guides can be found in **Docs** folder in the distributive package.

Optional: Depending on a computer configuration (where conversion service is running), the following registry change may need to be made to improve the conversion quality:

1. Login to machine that runs HarePoint Document Conversion Worker Service using the account this service runs (important)
2. Find or create the registry hive: HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_IVIEWOBJECTDRAW_DMLT9_WITH_GDI
3. Create there a new registry key DWORD, name InfoPath.exe, value 0.

Parameters:

list item url - Specify a list item url that should be processed.

PDF - Select the target format; currently *PDF* is the only available option.

library url - Specify the URL of the library where the converted InfoPath forms will be created.

Default - (optional) Type the name of the InfoPath form view to be converted. You can leave it set to *Default* - in that case all views will be converted.

Examples:

Convert InfoPath form at Current Item:Encoded Absolute URL to PDF and save to document library at http://labsp13/Converted%20Documents. Default InfoPath form view will be converted.

6.5 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 the 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 input 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)

List of supported output formats includes all input formats and two additional:

- Portable Document Format (.pdf)
- XML Paper Specification (.xps)

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

Note: This action requires "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.

6.6 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.

6.7 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 0 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 Add Document to ZIP Archive

Add document at this url to zip archive at this url. Overwrite existing archive. Set this password.

This workflow action is used to add a single or multiple documents to ZIP archive.

Parameters:

document at this url - specify the full URL of a document to be added to ZIP archive.

To add **multiple** documents, specify an array of URLs. Such array can be obtained, for example from **Find List Item at URL** or **Find List Item by CAML** actions.

archive at this url - specify the full URL of the target ZIP archive (including zip-file name and '.zip' extension).

Example: http://servername/site/library/archive.zip

Overwrite - choose the action for the case if specified zip file already exists: *Overwrite*, *Do not overwrite*.

this password - (optional) specify password for the ZIP archive.

7.2 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.

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.

7.3 Check Out Item at Url

Check out item at specified url.

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

Parameters:

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

7.4 Copy Document to List Item Attachment

Copy document at this url to list item at this url. Overwrite existing items.

This workflow action copies the specified document to a list item as an attachment.

Parameters:

Copy document at **this url** - the URL of the document to be copied.

to list item at **this url** - the URL of the target list item.

Overwrite - select to overwrite or not the existing items.

7.5 Copy Folder Structure

Copy folder structure of any content type from a folder at this url to the folder at this url.

This workflow action recreates a folder structure in a target folder, identical to the structure of the source folder. No content is copied, only empty folders are created.

Parameters:

any content type - Specify the content types that will be taken into account.

a folder at this url - Specify the full URL of the source folder.

to the folder at this url - Specify the full URL of the target folder.

7.6 Copy or Move Document

Copy document at this url to the folder at specified url and with the same name. Overwrite existing document, don't copy version history. Trigger events. Stop workflow on copy error.

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

It can also **copy** a document to **multiple folders**, if they are specified in an array.

Optionally, it can copy a document with its **version history**.

This action can be used to **copy .aspx pages** (these pages may contain web parts). **Important note:** copying .aspx pages requires "**Add and Customize Pages**" permission!

Parameters:

Copy - Choose to *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.

Arrays of URLs are also supported for **copying** a document to all folders mentioned in an array.

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.

Note: This action adds new file version when overwriting

don't copy version history - Choose if document history needs to be copied as well.

Trigger events - Select *Trigger* or *Do not trigger* events for this action.

Stop workflow on copy error - Choose what happens when an error during a copy occurs:

- *Stop workflow* (default) - the workflow stops
- *Skip document* - the current document will be skipped; copying will continue with the next document
- *Skip invalid metadata value* - only skips copying data for certain problematic columns, all other columns are copied

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.

7.7 Copy or Move Document to Different Farm

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

This workflow action is used to copy or move specified document to the document library located on a different SharePoint farm.

Important Note: the action '*Set Target Site Options*' always should be used before this action!

Additionally, you can use '*Set Encrypted Passwords*' action and *Encrypt Credentials* tool to specify the credentials in secure way.

See Also: '*Set Encrypted Passwords*' and '*Set Target Site Options*' actions.

Parameters:

Copy - Specifies whether to copy or move document.

this url - Full URL of the source document.

specified url - Full URL of the target document library on a different SharePoint farm.

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

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

Note: This action adds new file version when overwriting

7.8 Copy or Move Folder

Copy all documents of this content type from a folder at this url to the folder at this url. Overwrite existing documents. Include subfolders. Stop workflow on copy error.

This workflow action copies or moves files from the specified folder to another folder. Document sets are also supported.

It can optionally:

- include files in subfolders
- copy/move the folder/document set as well

Parameters:

Copy/Move - select to copy or move files.

all - select to copy/move only the files (*all*) or the folder with the files (*the folder and all*).

this content type - (Optional) Use this optional parameter to copy/move documents only with the indicated content type.

By default, all documents of all content types are copied/moved.

folder at this url - specify the URL of the source folder

to the folder at this url - specify the URL of the target folder

Overwrite/Do not overwrite - specify if the existing files with the same names in the target folder should be overwritten, or the workflow should stop with an error.

Include/Do not include subfolders - specify if the subfolders in the source folder should also be copied/moved.

Stop workflow on copy error - choose what happens when an error during a copy occurs:

- *Stop workflow* (default) - the workflow stops

- *Skip document* - the current document will be skipped; copying will continue with the next document
- *Skip invalid metadata value* - only skips copying data for certain problematic columns, all other columns are copied

7.9 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: 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.

7.10 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 text document or replace existing document by new file created from string variable.

Parameters:

this text - Document body.

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

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.

7.11 Create Document Set in the Document Library

Create this title document set with this description based on this content type in the folder at this url.

This workflow action creates a document set in a specified document library.

Note: Since the document set from technical point of view behaves the same way as a document library, you can use the actions for document libraries to manage the document set.

Parameters:

this title - The document set name.

this description - The text description of the document set.

this content type - The name of the document set content type that new document set should be based on.

folder at this url - The URL of the folder in a document library where the document set will be created.

7.12 Create Folder in the Document Library

Create specified folder in the current folder.

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

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.

7.13 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.

Parameters:

specified - Folder name.

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

7.14 Create Link to a Document

Create link to document with this url at this library url.

This workflow action creates a link in specified library to a document from another library.

Parameters:

this url - the url of the document for which the link will be created.

this library url - URL of the library where the link will be created.

7.15 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.

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.

7.16 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.

Parameters:

specified - Subfolder name to delete.

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

7.17 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.

Parameters:

specified url - Full list item URL including site name.

7.18 Discard Publish Item

Discard publish list item at **this url** with **specified comment**.

This workflow action is used to discard list item publishing.

Parameters:

this url - Full URL of target list item.

specified comment - Comment text.

7.19 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 either **all or only specific documents** from a document library or folder in document library to local disk or network share.

Necessary documents can be specified using a file mask, or CAML query.

With a special CAML query it is possible to copy only the **folders structure, without the documents** - see second parameter description for more details.

Note: In order to specify file access credentials, add the "Set Network File Access" action prior "Download Documents to Folder action".

Parameters:

Recursively - Two options available:

- *Recursively*: save the content of a document library or a folder including all subfolders; subfolders tree will be created in *specified path*
- *Non-recursively*: save files only from specified document library of folder.

all - (Optional) By default (*all* setting), **all documents** will be downloaded. If you need to download only **certain files**, you can use two methods to specify them:

1. File mask, for example *.jpg. You can specify only one file mask.
2. CAML query - it should start with <Where>, for example:

```
<Where>  
<BeginsWith>  
<FieldRef Name='FileLeafRef' />  
<Value Type='File'>test</Value>  
</BeginsWith>  
</Where>
```

If you need to copy only the **documents structure**, without the documents, use the CAML query below:

```
<Where><Eq><FieldRef Name='FSObjType' /><Value Type='Integer'>1</Value></Eq></Where>
```

this folder url - The full URL of a document library or a folder in a document library.

specified path - Local disk or network path where the downloaded files will be saved to.

Overwrite - If the folder already contains file with the same name, then you can apply one of 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.

7.20 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 the specified text document to output variable.

Note: this action only supports plain text documents. If you need to get the content of Word document (.docx file), use "Read Word Document to Text" action.

Parameters:

this url - URL of the document at SharePoint Server.
Variable: variable - Output variable.

7.21 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.

Parameters:

specified - Name of the entry in property bag, e.g. vti_timelastmodified.
this url - Full URL of list item.
Variable: variable - Output variable.

7.22 Is File Locked (HarePoint)

If file at this url is locked for editing.

This workflow condition is used to determine, if the specified file is locked for edit.

See also: "Wait Until File Is Unlocked" action.

Parameters:

this url - Specify the full URL of the file to check if it's locked for edit.
--

7.23 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.

Parameters:

this url - Full URL of target list item.

specified - Comment text.

7.24 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.

Parameters:

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

this file name - New file name without extension.

7.25 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: 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.

7.26 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.

7.27 Set Target Site Options

Use site at this url url as target site. Logon using this user user name and this password password.

This workflow action is only used to set target SharePoint site URL, login and password for actions that work with remote SharePoint farm like "Copy or Move Document to Different Farm" or "Copy List Item to Different Farm".

This action must be placed above all other activities that can use stored options and credentials. You can use this action several times in one workflow step to use different target SharePoint farms.

Note: This workflow action **will not** take effect on other actions, except actions that work with remote SharePoint farm like "Copy or Move Document to Different Farm" or "Copy List Item to Different Farm".

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 url - Full URL of the target site on the different SharePoint farm.

this user - User name that will be used to access remote SharePoint farm.

this password - Password that will be used to access remote SharePoint farm.

7.28 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: 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.

7.29 Wait Until File Is Unlocked

Wait until file at this url is unlocked.

This workflow action pauses the workflow until the specified document is unlocked.

Parameters:

this url - Specify the full URL of the file to wait until it's unlocked.

8. E-Mail activities

8.1 Add Custom Headers to Email

Add these custom headers to outgoing e-mail messages.

This workflow action adds the specified e-mail headers to the e-mails sent by the HarePoint *Send E-Mail ...* actions.

Note: This action must precede *Send E-Mail ...* actions in a workflow - see **Examples** section.

Parameters:

these custom headers - Specify the headers to be added. Multiple headers can be specified, delimited by a new line.

Header examples:

- *Return-Receipt-To: sender@example.com* - request a receipt

- *Importance: high* - set importance level as High
- *Sensitivity: company-confidential* - set the e-mail sensitivity as confidential

For more details, please refer to <https://tools.ietf.org/html/rfc4021>

Examples:

Example: Requesting a receipt for an outgoing e-mail

Add Return-Receipt-To: test@harepoint.com custom headers to outgoing e-mail messages.

then Send e-mail to John Smith with Normal importance, blind carbon copy to this address, use test@harepoint.com as sender

Note: e-mail addresses in both actions need to be the same!

8.2 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 from POP3 Server" and "Receive E-Mail from IMAP Server" actions.

Parameters:

HTML body - One of the available message properties.

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

You can obtain MIME by using:

- **Get Document Content** for the EML file
- **Receive E-Mail from POP3/IMAP Server** actions

Variable: variable - String value of the specified property.

Examples:

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

8.3 Receive All E-Mails to List from IMAP Server

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 or specified number of messages from the specified IMAP mailbox from the default folder (usually INBOX) to the prepared SharePoint list. The messages are marked as read after adding into SharePoint list.

The target SharePoint list should contain the 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

Later versions support **OAuth 2.0** to receive emails from office365.com. In this case, additional steps are needed, please follow [this guide](#) to obtain Client ID, Tenant ID, and client secret. Specify them in the parameters as described below.

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:

all - specify if all e-mails should be downloaded, or only a specific number of e-mails. *0 (zero)* also stands for "all e-mails".

this server - IMAP server name or IP-address.

By default, port 143 is used. You can specify port with colon, like `mail.company.com:1143`.

type **None** - Secure connection type.

Available values: *None, SSL*. Note, that IMAP servers with SSL usually listen for incoming connections at port 993.

this login - User name for the IMAP server.

When connecting to office365.com, specify full email address, Client ID and Tenant ID in the following format:

your_email@domain.com@ClientID@TenantID

this password - Password for the IMAP server.

When connecting to office365.com, specify client secret in this field.

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

this url - Full URL of the list.

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

Examples:

Download all e-mails from imap.mailserver.com:143, secure connection type None, 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.

8.4 Receive All E-Mails to List from POP3 Server

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 or specific number of messages from the specified POP3 mailbox to the prepared SharePoint list. The messages are deleted from the mailbox after adding into SharePoint list.

The target SharePoint list should contain the 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

Later versions support **OAuth 2.0** to receive emails from office365.com. In this case, additional steps are needed, please follow [this guide](#) to obtain Client ID, Tenant ID, and client secret. Specify them in the parameters as described below.

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:

all - Specify if all or only a specific number of e-mails needs to be downloaded. *0 (zero)* - also stands for "all e-mails".

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 servers with SSL usually listen for incoming connections at port 995.

this login - User name for the POP3 server.

When connecting to office365.com, specify full email address, Client ID and Tenant ID in the following format:

your_email@domain.com@ClientID@TenantID

this password - Password for the POP3 server.

When connecting to office365.com, specify client secret in this field.

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

this url - Full URL of the list.

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

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.

8.5 Receive E-Mail from IMAP Server

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 an e-mail message from IMAP mail server from a default folder (usually INBOX). Action gets the list of messages in mailbox, fetches the first one and marks the received message as read.

Later versions support **OAuth 2.0** to receive emails from office365.com. In this case, additional steps are needed, please follow [this guide](#) to obtain Client ID, Tenant ID, and client secret. Specify them in the parameters as described below.

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 - IMAP server name or IP-address.

By default, port 143 is used, you can specify port with colon, like mail.company.com:1430.

type **None** - Secure connection type.

Available values: *None*, *SSL*. Note, that IMAP server with SSL usually listen for incoming connections at port 993.

this login - User name for the IMAP server.

When connecting to office365.com, specify full email address, Client ID and Tenant ID in the following format:

your_email@domain.com@ClientID@TenantID

this password - Password for the IMAP server.

When connecting to office365.com, specify client secret in this field.

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 *imap.mailserver.com*, secure connection type *None*, login *harepoint@mailserver.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*.

8.6 Receive E-Mail from POP3 Server

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 an e-mail message from POP3 mail server. Action gets the list of messages in mailbox, receives the first one and deletes the received message from mailbox.

Later versions support **OAuth 2.0** to receive emails from office365.com. In this case, additional steps are needed, please follow [this guide](#) to obtain Client ID, Tenant ID, and client secret. Specify them in the parameters as described below.

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.

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.

When connecting to office365.com, specify full email address, Client ID and Tenant ID in the following format:

your_email@domain.com@ClientID@TenantID

this password - Password for the POP3 server.

When connecting to office365.com, specify client secret in this field.

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.

8.7 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 from POP3 Server" and "Receive E-Mail from IMAP Server" actions.

Parameters:

all - File masks, separated by semicolon. Leave this 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.

8.8 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.

A **receipt** can be requested for this outgoing e-mail - see "Add custom headers to e-mail" action for further details.

Parameters:

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

Images can also be inserted into a message by using [img][img] tags:

Example:

[img]http://servername/Shared%20Documents/sample.gif[/img]

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

blind carbon copy to **this address** - (optional) Blind carbon copy recipients, may be 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.

8.9 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.

A **receipt** can be requested for this outgoing e-mail - see "Add custom headers to e-mail" action for further details.

Parameters:

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

Images can also be inserted into a message by using [img][img] tags:

Example:

[img]http://servername/Shared%20Documents/sample.gif[/img]

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

blind carbon copy to **this address** - (optional) Blind carbon copy recipients, may be 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.

8.10 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.

A **receipt** can be requested for this outgoing e-mail - see "Add custom headers to e-mail" action for further details.

Parameters:

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

Images can also be inserted into a message by using [img][img] tags:

Example:

`[img]http://servername/Shared%20Documents/sample.gif[/img]`

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

blind carbon copy to **this address** - (optional) Blind carbon copy recipients, may be 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](#).*

8.11 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.

A **receipt** can be requested for this outgoing e-mail - see "Add custom headers to e-mail" action for further details.

Parameters:

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

Images can also be inserted into a message by using [img][/]img tags:

Example:

```
[img]http://servername/Shared%20Documents/sample.gif[/img]
```

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

blind carbon copy to **this address** - (optional) Blind carbon copy recipients, may be 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.

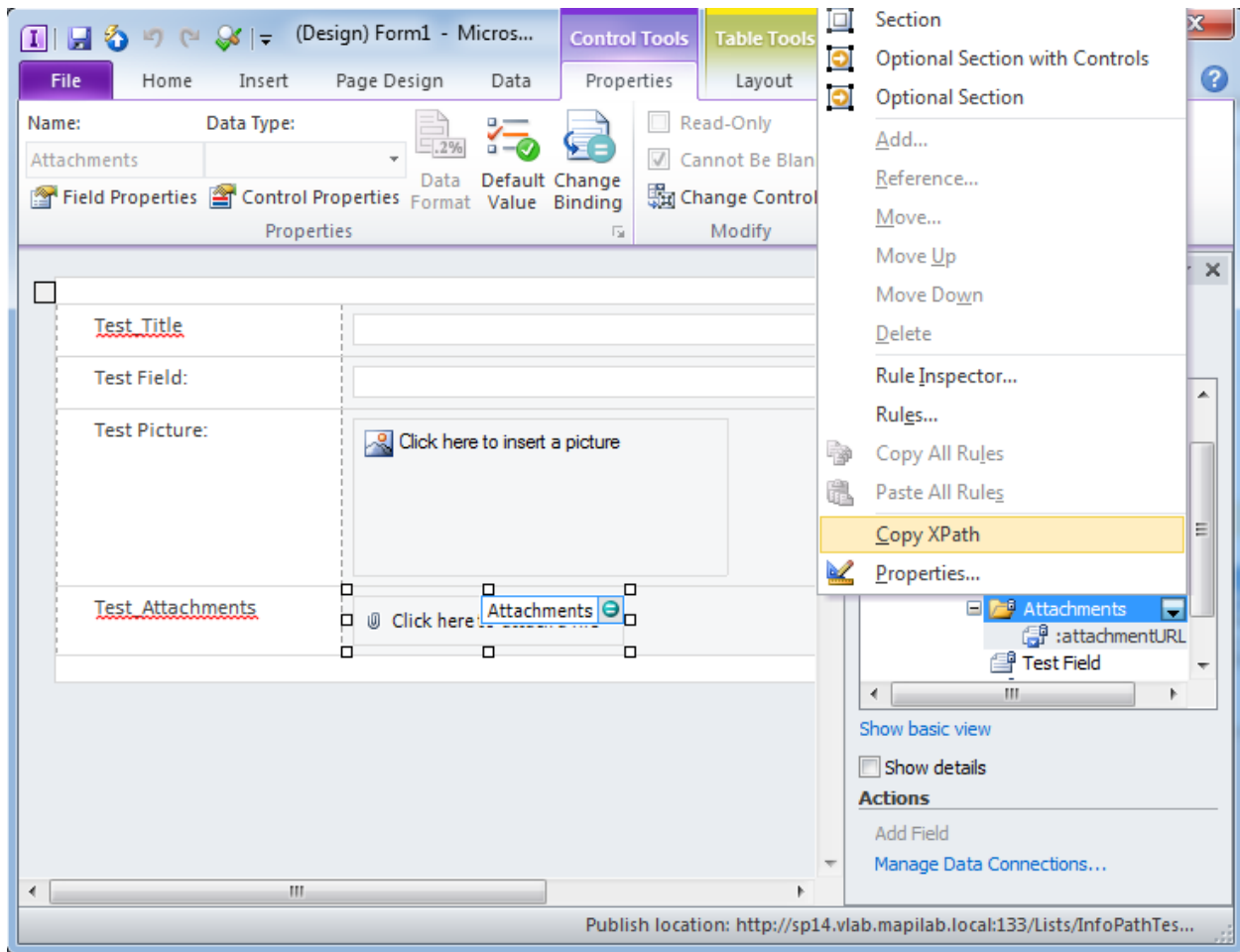
this item - InfoPath attachments of this list item will be attached to e-mail.

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:



8.12 Send E-Mail with InfoPath Form

Send e-mail to this address with Normal importance, blind carbon copy to this address, use this address as sender and attach InfoPath form at this url

This workflow activity sends an e-mail containing InfoPath form that will be displayed in Outlook, and also upon double-click it will be opened in InfoPath Form Filler.

Note: this action works for form libraries only.

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

See Also: "Set SMTP Server Options" action.

A **receipt** can be requested for this outgoing e-mail - see "Add custom headers to e-mail" action for further details.

Parameters:

this address - specify recipient's e-mail address
Normal importance - specify importance level. Available levels: Normal, High, Low
blind carbon copy to this address - (optional) specify blind carbon copy recipient's address
this address as sender - specify e-mail address that will appear as e-mail sender address
InfoPath form at this url - specify the URL of the InfoPath form to be sent

8.13 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.

A **receipt** can be requested for this outgoing e-mail - see "Add custom headers to e-mail" action for further details.

Parameters:

e-mail to this address - Click here to specify message recipients, carbon copy recipients, message subject and body.
Images can also be inserted into a message by using [img][img] tags:
Example:
<code>[img]http://servername/Shared%20Documents/sample.gif[/img]</code>
Normal - E-mail importance. Available options: <i>Normal, Low, High</i> .
blind carbon copy to this address - (optional) Blind carbon copy recipients, may be 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.

8.14 Send E-Mail with Multiple Attachments

Send e-mail to **this address** with **Normal** importance, blind carbon copy to **this address**, use **this address** as sender and add all files at **this urls** to attachments of e-mail.

This workflow action is used to send e-mail with multiple file attachment specified by array of URLs.

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

See Also: "Set SMTP Server Options" action.

A **receipt** can be requested for this outgoing e-mail - see "Add custom headers to e-mail" action for further details.

Parameters:

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

Images can also be inserted into a message by using [img][/]img tags:

Example:

```
[img]http://servername/Shared%20Documents/sample.gif[/img]
```

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

blind carbon copy to **this address** - (optional) Blind carbon copy recipients, may be 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 urls - Array of full URLs of files to be attached.

Examples:

Find item in the list at <http://spsite/Shared%20Documents> where item's Name is not empty value . Store result urls in Variable: array .

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

then Send e-mail to Sample User with Normal importance, blind carbon copy to this address , use test@mapilab.local as sender and add all files at Variable: array to attachments of e-mail.

8.15 Send E-mail with Word Document as a Body

Send e-mail to this address, CC to this address, BCC to this address with this subject and Word body at this url, use this address as sender and attach files at these urls.

This workflow action is used to send an e-mail with a Word document as a body. It is useful for sending invoices, surveys, and any other documents with complex formatting.

Parameters:

e-mail to this address - Specify the main recipients of an e-mail (TO field).

CC to this address - (optional) Specify carbon copy recipients (CC field).

BCC to this address - (optional) Specify blind carbon copy recipients (BCC field).

this subject - Specify the subject for an e-mail.

Word body at this url - Specify Encoded Absolute URL of a Word document that will be used as a body of an e-mail.

To get a proper URL, you can use the following:

- **Current Item: Encoded Absolute URL** lookup
- **Find list item at URL** action
- **Get List Item Url** action with *Native form* in the first parameter

this address as sender - (optional) Specify an alternative e-mail address that will be displayed as Sent From in the e-mail.

attach files at these urls - Specify an **array** of URLs of files that need to be attached.

Note: in order to get an array of URLs, you can use the following actions:

- **Find List Item at URL**

- **Get List Item Attachments**
- **Create Array** (*String* type)

Example of creating an array:

Create *String* array with <http://company.local/Documents/Sample.pdf> items and store in *Variable: Array*.

8.16 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.

A **receipt** can be requested for this outgoing e-mail - see "Add custom headers to e-mail" action for further details.

Parameters:

e-mail to **[this address](#)** - Click here to specify message recipients, carbon copy recipients, message subject and body.

Images can also be inserted into a message by using [img][img] tags:

Example:

```
[img]http://servername/Shared%20Documents/sample.gif[/img]
```

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

blind carbon copy to **[this address](#)** - (optional) Blind carbon copy recipients, may be blank.

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

8.17 Send SMS Message

Send SMS message to this phone numbers with this text.

This workflow action sends SMS messages to specified phone numbers with a specified text.

SMS settings should be set up at Central Administration -> System Settings -> Mobile Account

An example for RedOxygen provider:

<http://www.redoxygen.com/support/wiki/doku.php?id=sharepoint:integration>

Parameters:

this phone numbers - Text string with the phone numbers new line delimited, or the array. Should contain digits only, other symbols will be removed automatically.

this text - The text of the SMS message. The length is not limited, it will be sent as several SMS messages automatically if necessary.

Examples:

Send SMS Message to 446582513 with Hello! text

8.18 Set E-mail Delivery and Processing Options

Send regular e-mails. Expand macros in e-mail body and subject

This workflow action is used to send personalized e-mails.

This action needs to be placed prior to "Send E-Mail..." actions.

See also: "Set SMTP Server Options" action.

Parameters:

Regular - choose to send regular or personal e-mails:

- *regular*
- *personal*

Expand - Choose to expand or not expand macros in subject and body of an e-mail:

- *Expand*
- *Do not expand*

Is set to "Expand", the following Recipient's macros will be available in subject and body of an e-mail:

- %Name% - Recipient's display name. Example: John Smith.
- %FirstName% - Recipient's first name. Example: John.
- %LastName% - Recipient's last name. Example: Smith.
- %Department% - Recipient's department.
- %AccountName% - Recipient's account name. Example: HAREPOINT\jsmith.
- %JobTitle% - Recipient's job title.
- %Manager% - Recipient's manager account name.
- %MobilePhone% - Recipient's Mobile phone number.
- %WorkPhone% - Recipient's Work phone number.
- %WorkEmail% - Recipient's Work E-mail address.
- %Office% - Recipient's office.

Note: Recipient's macros are only available in personal e-mails.

The similar macros are available for the Sender:

- %SenderName% - Sender's display name. Example: John Smith.
- %SenderFirstName% - Sender's first name. Example: John.
- %SenderLastName% - Sender's last name. Example: Smith.
- %SenderDepartment% - Sender's department.
- %SenderAccountName% - Sender's account name. Example HAREPOINT\jsmith.
- %SenderJobTitle% - Sender's job title.
- %SenderManager% - Sender's manager account name.
- %SenderMobilePhone% - Sender's mobile phone number.
- %SenderWorkPhone% - Sender's work phone number.
- %SenderWorkEmail% - Sender's work E-mail address.

- %SenderOffice% - Sender's office.

Note: Sender's macros are available both in regular and in personal e-mails.

Note: macros only expand for **SharePoint users!** If the corresponding SharePoint has not been found, the workflow action will display an error. An e-mail will **not** be sent with macros that cannot be expanded.

8.19 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.

See also: "Set E-mail Delivery and Processing Options" action.

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. Exchange activities

9.1 Create Exchange Calendar Item

Create a new calendar item with this subject, with this body, event starts at this time, ends at this time, location at this location. Store item ID in Variable: itemID.

This workflow action is used to create a new calendar item in Exchange.

Note that this action only creates an event, but doesn't add attendees to it - this can be done by **Update Exchange Calendar Item Attendees** action.

The advanced parameters of an event (such as reminder interval, importance level, etc.) can be changed, if necessary, by **Update Exchange Calendar Item Advanced Properties** action.

Important note: **Set Exchange Web Service Options** action needs to be used prior to this action.

See also: **Send Meeting Request** action is equivalent to this action, but additionally sends meeting requests to attendees.

Parameters:

with this subject - Specify the subject of an event.

with this body - Specify the body of an event.

starts at this time - Specify when an event starts.

ends at this time - Specify when an event ends.

store item ID in Variable: itemID - Specify a variable to store the ID of an item.

This is a global ID and it is the same for all event attendees. It is used in other actions from "Exchange" category.

9.2 Delete Exchange Calendar Item

Delete calendar item with this id.

This workflow action is used to remove an event from user's calendar.

Important note: **Set Exchange Web Service Options** action needs to be used prior to this action.

Parameters:

with this id - Specify an ID of an event to be deleted. This ID can be obtained from "Create Exchange Calendar Item" action.

9.3 Respond to Exchange Calendar Item

Accept calendar item with this item id. Send updates: yes.

This workflow action is used to accept, decline or cancel an event.

Important note: **Set Exchange Web Service Options** action needs to be used prior to this action.

Parameters:

Accept calendar item - Select the response to this event, or cancel this event; the available options are:

- *Accept* - the given attendee will take part in this event
- *Tentatively accept* - the given attendee will likely take part in this event, but the situation may change; event will probably be accepted or declined later
- *Decline* - the given attendee won't be able to take part in this event
- *Cancel* - (only for event organizer) cancels the event for all attendees

with this item id - Specify an ID of an event to be updated. This ID can be obtained from "Create Exchange Calendar Item" action.

Send updates: yes - Specify if these updates should be reflected in calendars of the attendees.

9.4 Send Meeting Request

Send meeting request to this address, meeting starts at this time, ends at this time, location at this location, request responses: yes.

This workflow action is sending a meeting request.

Note: Prior to this action, **Set Exchange Web Service Options** action should be used.

See also: "Create Exchange Calendar Item" action is equivalent to this action, but does not assign attendees, it only creates an event.

Parameters:

this address - Recipient's e-mail address.
starts at this time - The meeting starting time.
ends at this time - The meeting ending time, leave blank for all day long event.
this location - Location where the meeting will be held.
request responses: yes - A value indicating whether responses are requested when invitations are sent for this meeting.

9.5 Set Exchange Web Service Options

Use Exchange Server at this url or e-mail address and this user and this password to authenticate against web service, Do not convert message body to plain text, use SharePoint computer time zone, impersonate as current user, use default as preferred culture.

This workflow action is setting Exchange Web Service options which are required to use actions from "Exchange" category.

This action needs to be put before any other actions from this category.

If you have several actions from "Exchange" category in a workflow, it is sufficient to have only one **Set Exchange Web Service Options** action in a workflow.

Parameters:

this url or e-mail address - Specify either URL of the EWS (e.g. https://exch.example.org/EWS/Exchange.asmx), or e-mail address of the user to determine this URL using Autodiscover Service (this option is triggered when '@' symbol is contained in the string).
Note: For Office 365 always use the following URL: https://outlook.office365.com/EWS/Exchange.asmx

this user - User name for authentication (should be left blank if credentials are provided via Encrypt Credentials tool).

Note: the meeting request will be sent on behalf of this account, unless *impersonate as current user* parameter is selected in this workflow action - see below for details.

this password - The password for authentication (should be left blank if credentials are provided via Encrypt Credentials tool).

Note: You can use the Encrypt Credentials tool to encrypt the password, in this case specify the Username and Password in Web Service field and use 'Set encrypted password' action.

Do not convert message body - Select to *Convert* message body to plain text or *Do not convert*.

Use **SharePoint computer** time - (optional) Specify the time zone for the meeting request. By default, SharePoint Computer time zone will be used.

If necessary, time zone can be specified explicitly. The available names for time zones can be viewed in the following system registry key:

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Time Zones

impersonate as current user - It is possible to impersonate as current user, so the meeting request will be sent on behalf of the user who launched the workflow.

Two conditions should be met for successful impersonation:

1. The user who will launch the workflow need to have a **valid e-mail address** in his profile on the Exchange server being used.

2. The user account (who will launch the workflow) needs to have **Send As permission**. Please refer to:

Exchange 2013 and above <https://docs.microsoft.com/en-us/exchange/client-developer/exchange-web-services/how-to-configure-impersonation>

Or grant a user that runs the workflow **ApplicationImpersonation** role in Exchange Admin Center

Exchange 2010, 2007 [http://msdn.microsoft.com/en-us/library/office/bb204095\(v=exchg.140\).aspx](http://msdn.microsoft.com/en-us/library/office/bb204095(v=exchg.140).aspx) for more details.

default as preferred culture - In most cases, keep this parameter set to default.

Examples:

Use Exchange Server at <https://exch.example.org/EWS/Exchange.asmx> and this user and this password to authenticate against web service, Do not convert message body to plain text, use SharePoint computer time zone, impersonate as current user.

9.6 Update Exchange Calendar Item

Update calendar item with this item id . Set this subject, this location and this message body. Set this time as start and this time as end, all-day event: yes. Send updates: yes.

This workflow action is used to change one or several parameters of an event.

Important note: **Set Exchange Web Service Options** action needs to be used prior to this action.

Parameters:

with this item id - Specify an ID of an event to be updated. This ID can be obtained from "Create Exchange Calendar Item" action.

this subject - (optional) Specify a new subject for an event.

this location - (optional) Specify a new location for an event.

this message body - (optional) Specify a new message body for an event.

this time as start - (optional) Specify a new starting time for an event.

this time as end - (optional) Specify a new ending time of an event.

all-day event: yes - Specify if the calendar event needs to last for entire day

Send updates: yes - Specify if these updates should be reflected in calendars of the attendees.

9.7 Update Exchange Calendar Item Advanced Properties

Update calendar item with this item id. Set fields: reminder 15 minutes, normal importance, normal sensitivity, response requested yes, allow new time proposal yes. Send updates: yes.

This workflow action is used to update an advanced properties of an event.

Important note: **Set Exchange Web Service Options** action needs to be used prior to this action.

Parameters:

with this item id - Specify an ID of an event to be updated. This ID can be obtained from "Create Exchange Calendar Item" action.

reminder 15 minutes - Set the reminder interval.

normal importance - Set the importance level.

normal sensitivity - Set the sensitivity for this item: *normal* or *private*.

response requested yes - Specify whether a response from attendees should be requested.

allow new time proposal yes - Specify if attendees can propose new time for an event.

Send updates: yes - Specify if these updates should be reflected in calendars of the attendees.

9.8 Update Exchange Calendar Item Attendees

Update calendar item with this item id, add these required attendees, add these optional attendees.
Send updates: yes.

This workflow action is used to add or change attendees for an event.

Important note: **Set Exchange Web Service Options** action needs to be used prior to this action.

Parameters:

with this item id - Specify the item ID of an event to be updated.

This ID can be obtained from **Create Exchange Calendar Item** action.

add these required attendees - Select to *add* or *set* the required attendees, and specify them.

Add option adds these attendees to existing ones; *set* option replaces the existing attendees with the specified ones.

add these optional attendees - Select to *add* or *set* the optional attendees, and specify them.

Add option adds these attendees to existing ones; *set* option replaces the existing attendees with the specified ones.

Send updates: yes - Specify if these updates should be reflected in calendars of the attendees.

9.9 Update Recurring Exchange Calendar Item

Update calendar item with this item id. Set this recurrence pattern. Send updates: yes.

This workflow action is used to update or create recurring events in Exchange calendar.

Parameters:

item with this item id - Specify an ID of a calendar item to be updated.

this recurrence pattern - Specify a recurrence pattern in XML format.

Note: XML format is case-sensitive!

The following patterns are available:

Daily pattern

A recurrence pattern where each occurrence happens a specific number of days after the previous one.

EndDate - The date after which the recurrence ends. Setting EndDate resets NumberOfOccurrences

Interval - Interval between occurrences in days

NumberOfOccurrences - The number of occurrences after which the recurrence ends. Setting NumberOfOccurrences resets EndDate

StartDate - The date and time when the recurrence start

Example:

```
<Daily>
<StartDate>2018-01-01</StartDate>
<Interval>1</Interval>
<EndDate>2018-01-31</EndDate>
</Daily>
```

Weekly pattern

A recurrence pattern in which each occurrence happens on specific days a specific number of weeks after the previous one.

DaysOfTheWeek - The comma separated list of days of the week on which the event recurs

Values: *Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday*

EndDate - The date after which the recurrence ends. Setting EndDate resets NumberOfOccurrences

Interval - Interval between occurrences in weeks

NumberOfOccurrences - The number of occurrences after which the recurrence ends. Setting NumberOfOccurrences resets EndDate

StartDate - The date and time when the recurrence start

Example:

The event recurs on Monday and Wednesday of every second week effective January 1, 2018, through March 31, 2018.

```
<Weekly>
<StartDate>2018-01-01</StartDate>
<DaysOfTheWeek>Monday,Wednesday</DaysOfTheWeek >
<Interval>2</Interval>
<EndDate>2018-03-31</EndDate>
</Weekly>
```

Monthly pattern

A recurrence pattern where each occurrence happens on a specific day a specific number of months after the previous one.

DayOfMonth - The day of the month when each occurrence happens. DayOfMonth must be between 1 and 31

EndDate - The date after which the recurrence ends. Setting EndDate resets NumberOfOccurrences

Interval - Interval between occurrences in months

NumberOfOccurrences - The number of occurrences after which the recurrence ends. Setting NumberOfOccurrences resets EndDate

StartDate - The date and time when the recurrence start

Example:

```
<Monthly>
<StartDate>2018-01-01</StartDate>
<DayOfMonth>15</DayOfMonth>
<Interval>1</Interval>
</Monthly>
```

RelativeMonthly pattern

A recurrence pattern where each occurrence happens on a relative day a specific number of months after the previous one.

DayOfTheWeek - The day of the week on which the event recurs

Values: *Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Day (any day of the week), Weekday, WeekendDay*

DayOfTheWeekIndex - The week of the month on which the event recurs

Values: *First, Second, Third, Fourth* or *Last*

EndDate - The date after which the recurrence ends. Setting EndDate resets NumberOfOccurrences

Interval - Interval between occurrences in months

NumberOfOccurrences - The number of occurrences after which the recurrence ends. Setting

NumberOfOccurrences resets EndDate

StartDate - The date and time when the recurrence start

Example:

```
<RelativeMonthly>  
<StartDate>2018-01-01</StartDate>  
<DayOfTheWeek>Monday</DayOfTheWeek>  
<DayOfTheWeekIndex>Third</DayOfTheWeekIndex>  
<Interval>1</Interval>  
</RelativeMonthly>
```

Yearly

A recurrence pattern where each occurrence happens on a specific day every year.

DayOfMonth - The day of the month when each occurrence happens. DayOfMonth must be between 1 and 31

Month - The month of the year when each occurrence happens.

Values: *January, February, March, April, May, June, July, August, September, October, November, December*

EndDate - The date after which the recurrence ends. Setting EndDate resets NumberOfOccurrences

NumberOfOccurrences - The number of occurrences after which the recurrence ends. Setting NumberOfOccurrences resets EndDate

StartDate - The date and time when the recurrence start

Example:

```
<Yearly>  
<StartDate>2018-01-01</StartDate>  
<DayOfMonth>15</DayOfMonth>  
<Month>November</Month>  
</Yearly>
```

Relative Yearly pattern

A recurrence pattern in which each occurrence happens on a relative day every year.

DayOfTheWeek - The day of the week on which the event recurs

Values: *Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Day (any day of the week), Weekday, WeekendDay*

DayOfTheWeekIndex - The week of the month on which the event recurs

Values: *First, Second, Third, Fourth or Last*

Month - The month of the year when each occurrence happens.

Values: *January, February, March, April, May, June, July, August, September, October, November,*

December

EndDate - The date after which the recurrence ends. Setting *EndDate* resets *NumberOfOccurrences*

NumberOfOccurrences - The number of occurrences after which the recurrence ends. Setting *NumberOfOccurrences* resets *EndDate*

StartDate - The date and time when the recurrence start

Example:

The appointment recurs on the first Monday in November, starting in November 2018.

```
<RelativeYearly>
<StartDate>2018-01-01</StartDate>
<DayOfTheWeek>Monday</DayOfTheWeek>
<DayOfTheWeekIndex>First</DayOfTheWeekIndex>
<Month>November</Month>
</RelativeYearly>
```

Send updates: yes - Specify if updates need to be sent.

10. Images activities

10.1 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 Absolute URL is supported

If image at Current Item:Encoded Absolute URL uses compression

Log Conversion is not required to the workflow history list

else

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

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

This workflow automatically converts uncompressed images to JPG format.

10.2 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.

10.3 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.

10.4 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.

10.5 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 the EXIF property from an image in one of 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:

Title - Name of the EXIF property (see list above).

this url - URL of the picture in Document Library.

Variable: variable - Output string variable. Empty string if the property is not found.

Variable: variable - Output string variable with the value in the form as it is stored in the image. Empty string if the property is not found.

10.6 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.

10.7 Get XMP Property

Get specified XMP metadata property from document at this url and store result in Variable: variable1.

This workflow action gets the specified XMP property from the specified document and saves it to a variable.

Parameters:

specified XMP metadata - Select the required XMP metadata from the drop-down list.

document at this url - Specify the full URL of the document.

Variable: variable1 - Specify the variable name to store the result.

Examples:

Get Dublin Core - Subject XMP metadata property from Current Item: Encoded Absolute URL and store in Variable:XMPSubject.

10.8 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 Absolute URL is supported

If image at Current Item:Encoded Absolute URL uses compression

Log Conversion is not required to the workflow history list

else

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

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

This workflow automatically converts uncompressed images to JPG format.

10.9 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 Absolute URL is supported

If image at Current Item:Encoded Absolute URL uses compression

Log Conversion is not required to the workflow history list

else

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

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

This workflow automatically converts uncompressed images to JPG format.

10.10 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.

10.11 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 - Two 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, 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.

10.12 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, 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.

10.13 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.

10.14 Set EXIF Property on Image

Set EXIF property Title on image at this url to variable value.

This workflow action sets the specified EXIF property for the specified image.

Parameters:

Title - The required EXIF property to be set can be selected from the drop-down list.

Available values: Title, Subject, Artist, Copyright, Date and Time, Software, GPS Latitude, GPS Longitude, GPS Altitude

this url - The URL of the image.

variable - The value of EXIF property to be set.

Examples:

Set EXIF property Title on image at Current Item:Encoded Absolute URL to Trafalgar Square value.

10.15 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.

10.16 Set XMP Property

Set specified XMP metadata property in document at this url to this value, trigger events.

This workflow action sets the specified XMP metadata property for specified document to specified value.

Parameters:

specified XMP metadata - Choose XMP metadata property from the drop-down list.

document at this url - Specify the full url of the document.

this value - Specify the value to be set.

In case if property field contains multiple properties, they should be delimited by semicolon (;)

Example:

Set Dublin Core - Subject XMP metadata property in url to construction; house; building.

trigger events - Select to trigger/do not trigger item change event.

Examples:

Set Dublin Core - Description XMP metadata property in Current Item: Encoded Absolute URL to Some description here, trigger events.

11. Lists activities

11.1 Add Content Type to List

Add this content type to list at this list url.

This workflow action adds specified content type to the list.

Parameters:

this - Name of the content type.

this list url - Full URL of the target list.

11.2 Add Existing Column to Content Type

Add this existing column to specified content type on the site at current site url.

This workflow action adds and existing column to a content type.

Parameters:

this existing column - Specify the name of the existing column to be added.

specified content type - Specify the content type name to which the column will be added.

current site - (optional) Specify the URL of the site.

Examples:

Step 1

Create a new site column on the site at [current site](#) url based on `<Field Type="Text" Name="Test Text Fi... CAML`.
then Create a new content type on the site at [current site](#) url with [Word Document](#) title based on [Document](#) , put into [this sorting group](#) .
then Add [Test Text Field](#) existing column to [Word Document](#) content type on the site at [current site](#) url.
then Set document template url for [Word Document](#) content type on the site at [current site](#) url to `Current Item:Encoded Absolute URL` , [copy](#) document to the resource folder.
then Add [Word Document](#) content type to list at `Current Item:Encoded Absolute URL` .

CAML: `<Field Type="Text" Name="Test_Text_Field" DisplayName="Test Text Field" />`

11.3 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`

11.4 Build String from List

Build string using this template and delimiter from list items at this list url by this CAML query. Store result in Variable: variable.

This workflow action is used to build string variable using values of specified fields of multiple list items.

In later versions, field values can also be formatted

Parameters:

this template - String containing one or more macros in format `<%FieldName%>`. Where FieldName is a display name or internal name of list column.

Example: *Item <%Title%> was created at <%Created%>.*

Optionally, you can specify the **formatting option** for field values: `<%FieldName:FormattingOption%>`.

For DateTime fields:

- `:date` - to display only the date part
- `:time` - to display only the time part

Example: *Creation date is <%Created:date%> and creation time is <%Created:time%>*

For other types of fields you can use:

- `:ToLower` - to convert the value to lower register
- `:ToUpper` - to convert the value to upper register
- `:Trim` - to remove leading and trailing space symbols

Example: *The lower case title is <%Title:ToLower%>*

delimiter - (optional) Delimiter between strings built from template for each list item. By default - new line (line feed).

this list url - Full URL of SharePoint list.

this CAML query - String with the CAML query to include the necessary list items.

To include all items in the list, you can use `<Where></Where>` query.

Variable: variable - Output string variable.

Examples:

Build string using *Item <%Title%> was created at <%Created%> and delimiter from list items at <http://sharepoint/Lists/Tasks/> by <Where> <IsNotNull> <Fi... . Store results in Variable: variable.*

the Log Variable: variable to the workflow history list

11.5 Calculate Aggregation Function

Calculate SUM on this field from all items recursively 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: 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.

Parameters:

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

this field - Field name. Versions v2.7.2250 and above of HarePoint Workflow Extensions support both **DisplayName** and **InternalName**.

all - (optional) Items filter. By default, all items are included.

This parameter is a 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 CAML 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/Software>

recursively - Count inside summary tasks which may also include child items.

Or select **non-recursively** to skip summary task child items.

this list - Specify the SharePoint list.

Variable:variable1 - Output variable to store result.

Note: The 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.

11.6 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. Store copied item url in Variable: variable.

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.

Note: in versions later than 2.12.2756 it is possible to use **Current Item: Encoded Absolute URL** lookup in this parameter

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*).

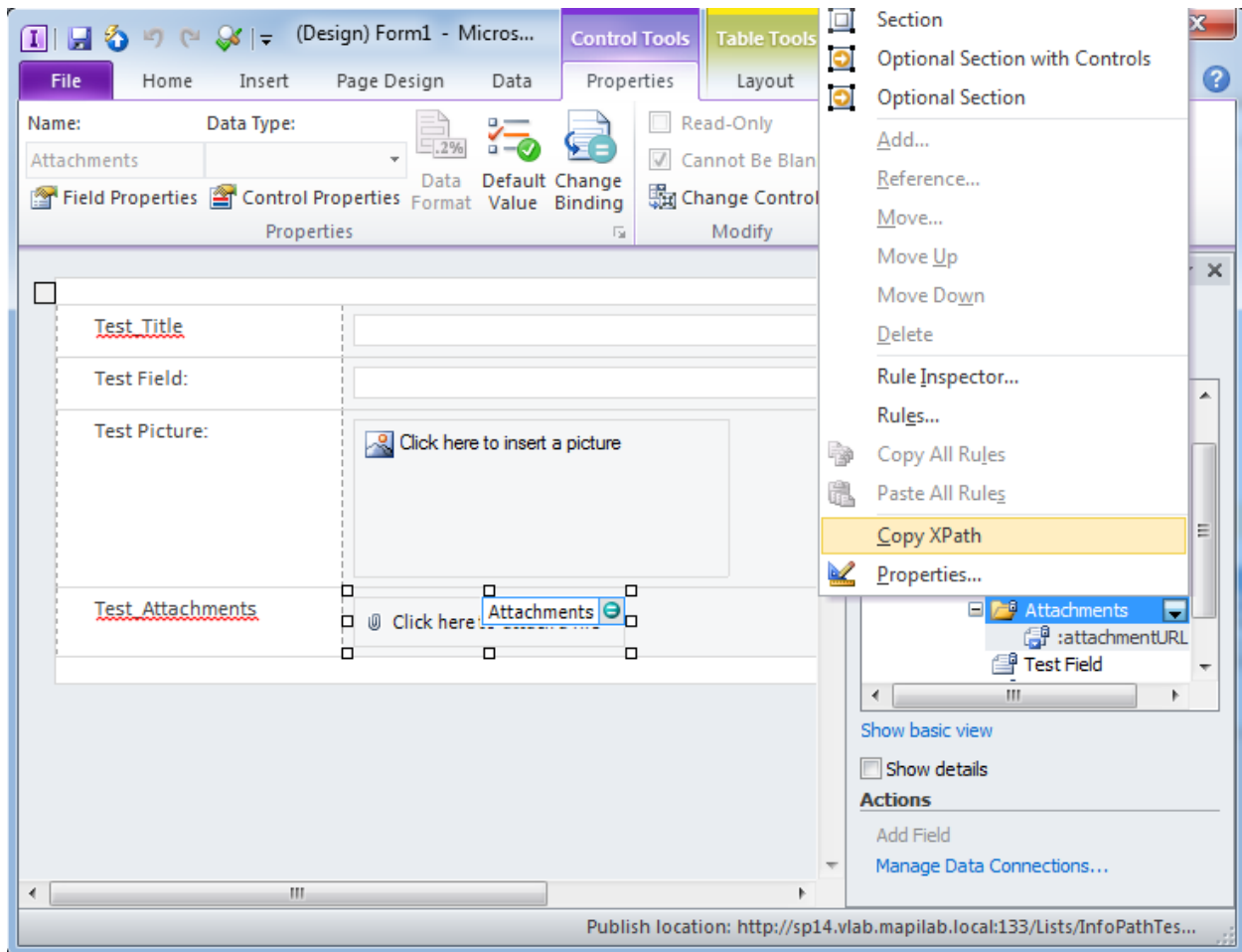
Variable: variable - Full URL of the new attachment.

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. Store copied item url in Variable: variable.

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:



11.7 Copy List Item at Url

Copy list item at [this url](#) to the list at [this list url](#). Copy version history: no. Store output item url in Variable: `variable1`. Trigger events.

This workflow action is used to copy list item from one list to another.

Note: This action can copy multiple list items, if an array of URLs is provided as the input parameter.

Parameters:

this url - Full URL of the source list item. Arrays of the URLs are also supported; in this case all items for this array will be copied.

this list url - Full URL of the target list.

Copy version history: no - specify if a version history for an item should be copied as well (where applicable). Default value is *no*.

Variable: variable1 - Output string variable for the URL of copied list item.

Trigger - Specifies whether to trigger an item create event.

11.8 Copy List Item Attachments to Document Library

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

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

See Also: "Copy List Item Attachments to Document Library (advanced)" - this action is identical, but allows you to save the URLs of the saved attachments in an array variable.

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, to skip this file (*Do not overwrite*), or to *Append unique ID* to the file name.

11.9 Copy List Item Attachments to Document Library (advanced)

Copy attachments of list item at this url to document library at specified url. Overwrite if items exist. Store urls of output items in Variable:UrlsArray.

This workflow action is used to copy attachments from specified list item to document library specified by URL, and save the URLs of the output items to an array.

See Also: "Copy List Item Attachments to Document Library" in case if you don't need to have the output URLs saved to a variable.

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, to skip this file (*Do not overwrite*), or to *Append unique ID* to the file name.

Variable:UrlsArray - The array variable to save the URLs of the items that have been copied to target library.

11.10 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.

11.11 Copy List Item to Different Farm

Copy list item at this url to the list at this list url. Store output item url in Variable: variable1.

This workflow action is used to copy list item from one list to another located on a different SharePoint farm.

Important Note: the action '*Set Target Site Options*' always should be used before this action!

Additionally, you can use '*Set Encrypted Passwords*' action and *Encrypt Credentials* tool to specify the credentials in secure way.

See Also: 'Set Encrypted Passwords' and 'Set Target Site Options' actions.

Parameters:

this url - Full URL of the source list item.

this list url - Full URL of the target list located on a different SharePoint farm.

Variable: variable1 - Output string variable for the URL of copied list item.

11.12 Create Content Type

Create a new content type on the site at current site url with this title based on this content type, put into this sorting group.

This workflow action creates a new content type.

Parameters:

current site - (optional) specify the URL of the site where the new content type needs to be created.

this title - the title of the new content type

based on this content type - the name of the parent content type

this sorting group - (optional) the name of the group where the new content type should be placed.

Examples:

Step 1

Create a new site column on the site at current site url based on `<Field Type="Text" Name="Test Text Fi... CAML`.
then Create a new content type on the site at current site url with Word Document title based on Document, put into this sorting group.
then Add Test Text Field existing column to Word Document content type on the site at current site url.
then Set document template url for Word Document content type on the site at current site url to Current Item:Encoded Absolute URL, copy document to the resource folder.
then Add Word Document content type to list at Current Item:Encoded Absolute URL.

CAML: `<Field Type="Text" Name="Test_Text_Field" DisplayName="Test Text Field" />`

11.13 Create List

Create list on the current site with this title and this description based on template with this name, show on Quick Launch: yes. Store list url to Variable: variable.

Creates new SharePoint list or document library using the specified template.

Also possible to create lists on other sites (not only on the current site).

Parameters:

on the current site - (Optional) Specify the URL of the site where the list needs to be created. Leave *current site* to create a list on a current site.

this title - Title of new list.

this description - Description of the new list.

this name - Name of the list or document library template.

Document Library - to create a document library

Custom List - to create a list.

Use the same name as seen in **Site Contents - Add new app**, or review the list of available templates here: https://docs.microsoft.com/en-us/openspecs/sharepoint_protocols/ms-wssts/8bf797af-288c-4a1d-a14b-cf5394e636cf

show on Quick Launch: yes - Specify whether to display this list on Quick Launch or not.

Variable: variable - This variable will contain the full URL of the created list or document library

11.14 Create List Item at Url

Create list item with specified values at this list url, use default content type. 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>
```

Note: XML format is case-sensitive. With XML format you can create multiple items by specifying several *<item></item>* sections, with parameters for each new item.

See Also: **Update List Item at Url** action.

Note: With this action you can also create other items in SharePoint that are based on List functionality, e.g. **tasks** (and **sub-tasks** in SharePoint 2013), **knowledgebase articles** (in SharePoint 2013), etc. - see **Examples** section below.

See Also: you can create multiple list items and fill them with values from an array using **Create List Items from Array** action.

Parameters:

specified values - Values for list item fields in INI or XML format.

a) For **Lookup** fields, specify only the ID of the lookup item:

Example:

Country=27

Multiselect Lookup fields are supported, values can be either comma (,) or semicolon (;) delimited.

See also: The IDs of the items can be obtained by using two actions: **Find List Item at URL** to find necessary item and **Get Field Value** to get its ID.

b) For **Hyperlink** type fields it is also possible to set description using *Url#description*:

Example:

Title=new task

Url=http://www.harepoint.com

Url#description=HarePoint website

c) **Multi-line text fields** - in order to transfer a multi-line text to a workflow properly, use only XML format in this action.

Example:

```
<field title="SomeLongText">Multi  
Line  
Text</field>
```

d) **Managed metadata** columns are supported, it is possible to specify term label names from Terms Set. Multiple labels should be delimited by a semicolon (;)

Example:

```
FieldName=Label1;Label2;Label3
```

e) **Choice** and **Checkboxes** types fields are supported. Specify the values delimited by a semicolon (;)

Example:

```
Title=New Order  
Order=Pizza;Pasta;Black Tea
```

f) You can use a **special (alternative) format** to get values of workflow variables. To use it, the first line in this parameter should be **UseCustomVars** - this keyword enables this functionality. In the next lines you can use any workflow variable by its name: %VariableName%. This also supports arrays - you can get array item values as %ArrayName[0]%, %ArrayName[1]%, and so on.

Example (for INI format):

```
UseCustomVars  
Title=%ResultArray[0]%  
Order=%ResultArray[1]%  
Size=%ResultArray[2]%  
Ingredients=%ResultArray[3]%
```

In this example the list item fields will get the values of array variable *ResultArray*.

this list url - Full URL of list.

default content type - (optional) specify the name of content type that should be used to create an item.

Variable: variable1 - Output string variable for the URL of created list item.

Trigger - Specifies whether to trigger an item create event.

Examples:

Example: Creating a sub-task (SharePoint 2013 only)

Create list item with *Task Name=...* at <http://myserver/Lists/MyTasks>, use *default* content type. Store new list item url in Variable: URL. *Trigger* events.

First parameter contains the following:

```
Task Name=Sub2
Due Date=11/29/2016
Assigned To=user1
Parent ID=1
```

Where *Parent ID* is the ID of the parent task. Without this line a regular task will be created.

11.15 Create List Item on Different Farm

Create list item with specified values at this list url. Store new list item url in Variable: variable.

This workflow action is used to create new list item at the different SharePoint farm 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>
```

Important Note: the action '*Set Target Site Options*' always should be used before this action!

Additionally, you can use 'Set Encrypted Passwords' action and *Encrypt Credentials* tool to specify the credentials in secure way.

See Also: 'Set Encrypted Passwords' and 'Set Target Site Options' actions.

Parameters:

specified values - Values for list item fields in INI or XML format.
this list url - Full URL of list on different farm.
Variable: variable - Output string variable for the URL of created list item.

Examples:

Example 1:

Use site at <http://tragetsite/one> url as target site. Logon using COMPANY\someuser username and somepassword password.

then Create list item with Somevalue=127 at <http://targetsites/one/Lists/Orders> ...

Example 2:

Set encrypted passwords with BjhqHUuj+jhucu... string

then Use site at <http://targetsites/one> url as target site. Logon using this user username and this password password.

then Create list item with Somevalue=127 at <http://targetsites/one/Lists/Orders> ...

11.16 Create List Items from Array

Create list items in a list at this url and fill specified field with items from this array, use specified field values. Trigger events.

This workflow action creates **several** items in a specified list, and fills a certain field in them with the corresponding values from array; also fills other fields in these items according to provided data in INI or XML format, same as "Create List Item at URL" action:

INI format:

Position=Team Leader

First Name=John

Last Name=Smith

Age=23

XML format:

```
<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>
```

Note: XML format is case-sensitive.

See also: This action can be used in combination with **Get Multi-Valued Field Value at Url** action.

See also: If you need to create only a **single** item, use **Create list item at URL** action.

Parameters:

list at this url - Specify the full URL of the list where items need to be created.

specified field - Type in the field name that should be filled with values from an array.

Note: Do not use lookups, simply type in the field name.

this array - Specify the array that contains values for the given field in the new items.

In particular, this array can be obtained from **Get Multi-Valued Field Value at Url** action.

specified field values - (optional) specify values for other fields of newly created list items. INI or XML format can be used.

a) For **Lookup** fields, specify only the ID of the lookup item:

Example:

Country=27

Multiselect Lookup fields are supported, values can be either comma (,) or semicolon (;) delimited.

See also: The IDs of the items can be obtained by using two actions: **Find List Item at URL** to find necessary item and **Get Field Value** to get its ID.

b) For **Hyperlink** type fields it is also possible to set description using *Url#description*:

Example:

```
Title=new task  
Url=http://www.harepoint.com  
Url#description=HarePoint website
```

c) **Multi-line text fields** - in order to transfer a multi-line text to a workflow properly, use only XML format in this action.

Example:

```
<field title="SomeLongText">Multi  
Line  
Text</field>
```

d) **Managed metadata** columns are supported, it is possible to specify term label names from Terms Set. Multiple labels should be delimited by a semicolon (;)

Example:

```
FieldName=Label1;Label2;Label3
```

e) **Choice** and **Checkboxes** types fields are supported. Specify the values delimited by a semicolon (;)

Example:

```
Title=New Order  
Order=Pizza;Pasta;Black Tea
```

f) You can use a **special (alternative) format** to get values of workflow variables. To use it, the first line in this parameter should be **UseCustomVars** - this keyword enables this functionality. In the next lines you can use any workflow variable by its name: %VariableName%. This also supports arrays - you can get array item values as %ArrayName[0]%, %ArrayName[1]%, and so on.

Example (for INI format):

```
UseCustomVars  
Title=%ResultArray[0]%  
Order=%ResultArray[1]%  
Size=%ResultArray[2]%  
Ingredients=%ResultArray[3]%
```

In this example the list item fields will get the values of array variable *ResultArray*.

Trigger - Select to *trigger* or *do not trigger* events when items are created in a list.

11.17 Create Site Column

Create a new site column on the site at current site url based on this CAML.

This workflow action creates a new site column on the site based on CAML query.

Parameters:

Current site - (optional) Specify the URL of the site where the column needs to be created.

this CAML - CAML query.

For details on CAML syntax please refer to [http://msdn.microsoft.com/en-us/library/office/ms437580\(v=office.15\).aspx](http://msdn.microsoft.com/en-us/library/office/ms437580(v=office.15).aspx)

Examples:

Step 1

Create a new site column on the site at current site url based on `<Field Type="Text" Name="Test Text Fi... CAML`.
then Create a new content type on the site at current site url with Word Document title based on Document, put into this sorting group.
then Add Test Text Field existing column to Word Document content type on the site at current site url.
then Set document template url for Word Document content type on the site at current site url to Current Item:Encoded Absolute URL, copy document to the resource folder.
then Add Word Document content type to list at Current Item:Encoded Absolute URL.

CAML: `<Field Type="Text" Name="Test_Text_Field" DisplayName="Test Text Field" />`

11.18 Delete All List Items

Delete all list items in current list. 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/Software/>

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.

current list - (Optional) Specify the URL of the list where the items should be deleted. By default, current list is used.

Variable: variable - Output integer variable to store number of deleted items.

11.19 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.

Note: the array of full URLs of list items is also supported.

11.20 Delete Item on Different Farm

Delete item at specified list item url on different farm.

This workflow action is used to delete list item at different SharePoint farm.

Important Note: the action '*Set Target Site Options*' always should be used before this action!

Additionally, you can use '*Set Encrypted Passwords*' action and Encrypt Credentials tool to specify the credentials in secure way.

See Also: '*Set Encrypted Passwords*' and '*Set Target Site Options*' actions.

Parameters:

specified - Full URL of list item at different SharePoint farm.

11.21 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*).

11.22 Find List Item at Url

Find item in the list at this url where item's field equals value. Store top 1000 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.
top 1000 - specify number of results to be stored
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.

11.23 Find List Item by CAML

Find item in the list at this url by this CAML query, look for all files recursively. Store top 1000 result urls in Variable: array.

This workflow action is used to find list items using a 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.
all files recursively - specify the search scope. The following scopes are available:
<ul style="list-style-type: none"><i>only files</i> - finds files only in the specified list/library or specified folder<i>files and subfolders</i> - finds files and subfolders only in the specified list/library or specified folder<i>all files recursively</i> - (default) finds files in the current folder and all its subfolders

- *all files and subfolders recursively* - finds files and subfolders in the current folder and all its subfolders

top 1000 - Specify number of results to be stored.

Variable: array - Output array variable with URLs of list items.

Examples:

Find item in the list at Variable: listUrl by <Where><And><Eq><FieldRef>..., look for all files recursively. Store result urls in Variable: array.

11.24 Find List Item by CAML on Different Farm

Find item in the list at this url by this CAML query. Store top 1000 result urls in Variable: array.

This workflow action is used to find list items at the different SharePoint farm using CAML query.

See "Calculate Aggregation Function" activity description for examples and references on CAML queries.

Important Note: the action '*Set Target Site Options*' always should be used before this action!

Additionally, you can use '*Set Encrypted Passwords*' action and Encrypt Credentials tool to specify the credentials in secure way.

See Also: '*Set Encrypted Passwords*' and '*Set Target Site Options*' actions.

See Also: For the complete list of supported operators and types see 'Find List Item at Url' action.

Parameters:

this url - Full URL of list at the different SharePoint farm.

this CAML query - String with the CAML query.

top 1000 - Specify number of results to be stored.

Variable: array - Output array variable with URLs of list items.

11.25 Find List Item on Different Farm

Find item in the list at this url where item's field equals value. Store top 1000 result urls in Variable: array.

This workflow action is used to find list items on different SharePoint farm by the value of the specified field.

On SharePoint 2013 and above, this workflow action can also be used to interact with **SharePoint Online**.

Important Note: the action '*Set Target Site Options*' always should be used before this action!

Additionally, you can use '*Set Encrypted Passwords*' action and Encrypt Credentials tool to specify the credentials in secure way.

See Also: '*Set Encrypted Passwords*' and '*Set Target Site Options*' actions.

See Also: For the complete list of supported operators and types see 'Find List Item at Url' action.

Parameters:

this url - Full URL of list at the different SharePoint farm.

field - The name of the field to search for.

equals - Comparison operator.

value - Value to search for.

top 1000 - Specify number of results to be stored.

Variable: array - Output array variable with URLs of list items.

11.26 Get Field Value at Url

Get value from this field of list item at this url (current version) and store result in Variable: variable.

This workflow action is used to get a field value from list item specified by URL.

See also: you can get multi-valued field values and store them to an array using **Get Multi-Valued Field Value at Url** action.

Parameters:

this field - Field name.

this url - Full URL of list item.

current version - Specify the version of the list item.

If the versioning is enabled for the list, you can get the data from the previous versions of the item.

Two formats can be used:

- -1, -2, -3, ... - to specify previous versions relatively to the current version
- 1.0, 2.0, 3.0 - to specify the required version number directly

Note: To make sure if an item has older versions available, you can use **Get the Number of Versions of a List Item** action before **Get Field Value at Url** action.

Variable: **variable** - Output variable for the field value.

11.27 Get Field Value from Different Farm

Get value from this field of list item at this url and store result in variable.

This workflow action allows getting values of list item's fields, for lists located on different farms.

On SharePoint 2013 and above, this workflow action can also be used to interact with **SharePoint Online**.

Important note: In order to login to a remote farm, use **Set Target Site Options** before this workflow action.

Parameters:

this field - Type in the field name.

Note: simply type in a field name as a plain text; do not use lookup in SharePoint Designer.

this url - Full path to the list item on a remote site.

Note: we recommend using **Find List Item on Different Farm** or **Find List Item by CAML on Different Farm** action to obtain a proper list item URL.

Variable - Specify a variable that will store the value of a list item field.

11.28 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.

11.29 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

11.30 Get List Item Url

Get Display Form url for list item this list item (current version) and store result in Variable:variable.

This workflow action is used to get special URLs for the specified list item, such as Edit Form URL, Display Form URL, Full path, and more.

See the detailed description below.

In order to point to a list item, a lookup dialog is used.

Note: due to SharePoint Designer limitations, with this lookup dialog you can only point to list items in lists from current site only.

See also: **Get Special List Item Url at Url** action - it has the same features, but it uses list item URL instead of lookup dialog, allowing to point to list items in lists from other sites as well.

Parameters:

Display Form - Choose one of six options:

- *Display Form*
- *Edit Form*
- *Native Form*
- *Full Path*
- *Path without Filename*
- *Parent Folder Name*

In the case of documents library, *Native Form* is the URL of file.

Full path and *Path without Filename* are displayed relative to the current document library - see **Examples** section.

Note: HarePoint Workflow Extensions list item actions use *Native Form* as **list item URL**.

this list item - Select a list item from a list.

current version - Specify the version of the list item.

If the versioning is enabled for the list, you can get the data from the previous versions of the item.

Two formats can be used:

- -1, -2, -3, ... - to specify previous versions relatively to the current version
- 1.0, 2.0, 3.0 - to specify the required version number directly

Variable:variable- Output string variable.

Examples:

Example 1

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

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

Get Native Form url for list item Current Item (current version) 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 in 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

Example 2

a) If document location is *Library1/filename.txt*,

Full Path will return *filename.txt*

Path without filename will return empty string

Parent folder will return empty string

b) If document location is *Library1/folder/filename.txt*,

Full Path will return *folder/filename.txt*

Path without filename will return *folder*

Parent folder will return *folder*

c) If document location is *Library1/folder1/folder2/filename.txt*,

Full Path will return *folder1/folder2/filename.txt*

Path without filename will return *folder1/folder2*

Parent folder will return *folder2*

11.31 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.

11.32 Get Multi-Valued Field Value at Url

Get value from this field of list item at this url (current version) and store result in Variable: ArrayVariable1.

This workflow action gets the **multi-valued** field values of specified list item and saves them to array.

See also: in particular, this action can be used in combination with **Create list items from Array** action.

See also: if you need to get a **single-value** field value of a list item, use **Get Field Value at Url** action instead.

Parameters:

this field - Specify the multi-valued list item field to take data from.

this url - Specify the full url of the list item. Use **Current Item: Encoded Absolute URL** lookup, or **Find List Item at URL** action to obtain the proper URL for this field.

current version - (optional) Specify the version of a list item, or leave it *current* for the current version.

If the versioning is enabled for the list, you can get the data from the previous versions of the item.

Two formats can be used:

- -1, -2, -3, ... - to specify previous versions relatively to the current version
- 1.0, 2.0, 3.0 - to specify the required version number directly

Variable: ArrayVariable1 - Specify the array variable to store the output.

11.33 Get Special List Item Url at Url

Get Display Form url for list item this url (current version) and store result in Variable:variable.

This workflow action is used to get special URLs for the specified list item, such as Edit Form URL, Display Form URL, Full path, and more.

See detailed description below.

In order to point to a list item, a full URL of an item needs to be specified. With this action you can point to items in lists from another sites.

See also: "Get List Item Url" action

Parameters:

Display Form - Choose one of six options:

- *Display Form*
- *Edit Form*
- *Native Form*
- *Full Path*
- *Path without Filename*
- *Parent Folder Name*

In the case of documents library, *Native Form* is the URL of file.

Full path and *Path without Filename* are displayed relative to the current document library - see **Examples** section.

Note: HarePoint Workflow Extensions list item actions use *Native Form* as **list item URL**.

this url - Specify the full URL of a list item.

Full URL of an item can be obtained by:

- **Current Item: Encoded Absolute URL** lookup
- **Find List Item at URL** action
- **Create List Item at URL** and **Update List Item at URL** actions

current version - Specify the version of the list item.

If the versioning is enabled for the list, you can get the data from the previous versions of the item.

Two formats can be used:

- -1, -2, -3, ... - to specify previous versions relatively to the current version
- 1.0, 2.0, 3.0 - to specify the required version number directly

Variable:variable- Output string variable.

Examples:

Example 1

Get Display Form url for list item Current Item: Encoded Absolute URL (current version) and store result in Variable: F1

Get Edit Form url for list item Current Item: Encoded Absolute URL (current version) and store result in Variable: F2

Get Native Form url for list item Current Item: Encoded Absolute URL (current version) 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 in 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

Example 2

a) If document location is *Library1/filename.txt*,

Full Path will return *filename.txt*

Path without filename will return empty string

Parent folder will return empty string

b) If document location is *Library1/folder/filename.txt*,

Full Path will return *folder/filename.txt*

Path without filename will return *folder*

Parent folder will return *folder*

c) If document location is *Library1/folder1/folder2/filename.txt*,

Full Path will return *folder1/folder2/filename.txt*

Path without filename will return *folder1/folder2*

Parent folder will return *folder2*

11.34 Get the Number of Versions of a List Item

Get the number of versions of list item at this url and store in Variable: numberOfVersions.

This workflow action is used to get how many versions the specified list item has.

In particular, this action is helpful to make sure there is more than one list item version before using "Get Field Value at URL" action when it is needed to retrieve older versions of a field value.

Parameters:

list item at this url - Specify the URL of a list item.

To get a proper list item URL, you can use the following:

- **Current Item: Encoded Absolute URL** lookup
- **Find list item at URL** action
- **Get List Item Url with Native form** in the first parameter

Variable: numberOfVersions - Specify the variable to store the number of versions.

11.35 Get Workflow Task Property

Get this property value from workflow task at this url and store result in variable.

This workflow action allows to get values of extended properties of a workflow task.

Parameters:

this property - The property name. Can be left blank, in this case it will log the names of all available properties.

this url - Specify the URL of the workflow task item.

variable - Specify the variable to store the property value.

If property is not found, *Null* will be stored.

11.36 If List Exists (HarePoint)

If list with this name exists on the site at this url.

This workflow condition is used to determine if the specified list exists on an indicated site.

Parameters:

list with this name - Specify the List name.

at this url - Specify the site URL where to look for the required List.

11.37 Is Current Item Checked Out (HarePoint)

If current item is checked out.

This workflow condition returns true if the current item is checked out.

11.38 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.

11.39 Loop Through and Update List Items

Loop through items by this CAML query in list at this URL and update specified values. Trigger events, create new version of list item.

This workflow action is used to update multiple items in a list with the same data (set certain column(s) of those items to a certain value(s)). The necessary items are selected by a CAML query.

Parameters:

this CAML query - Specify a CAML query to select items that need to be updated.

Example: all list items where the values of **Title** column begin with **test1**

```
<Where>
<BeginsWith>
<FieldRef Name="Title" />
<Value Type="Text">test1</Value>
</BeginsWith>
</Where>
```

list at this URL - specify URL of a list where items need to be updated.

specified values - specify new values for the columns that need to be changed for the selected items, in INI or XML format:

First one is the format in style of **INI** file:

Position=Team Leader

First Name=John

Last Name=Smith

Age=23

Second one is **XML** format:

```
<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>
```

Various field types are supported:

a) For **Lookup** fields, specify only the ID of the lookup item:

Example:

Country=27

Multiselect Lookup fields are supported, values can be either comma (,) or semicolon (;) delimited.

See also: The IDs of the items can be obtained by using two actions: **Find List Item at URL** to find

necessary item and **Get Field Value** to get its ID.

b) For **Hyperlink** type fields it is also possible to update description using *Url#description*:

Example:

```
Title=new task
Url=http://www.harepoint.com
Url#description=HarePoint website
```

c) **Managed metadata** columns are supported, it is possible to specify term label names from Terms Set. Multiple labels should be delimited by a semicolon (;)

Example:

```
FieldName=Label1;Label2;Label3
```

d) **Choice** and **Checkboxes** types fields are supported. Specify the values delimited by a semicolon (;)

Example:

```
Title=New Order
Order=Pizza;Pasta;Black Tea
```

e) You can use a **special (alternative) format** to get values of workflow variables. To use it, the first line in this parameter should be **UseCustomVars** - this keyword enables this functionality. In the next lines you can use any workflow variable by its name: %VariableName%. This also supports arrays - you can get array item values as %ArrayName[0]%, %ArrayName[1]%, and so on.

Example (for INI format):

```
UseCustomVars
Title=%ResultArray[0]%
Order=%ResultArray[1]%
Size=%ResultArray[2]%
Ingredients=%ResultArray[3]%
```

In this example the list item fields will get the values of array variable *ResultArray*.

Trigger events - Choose to *Trigger* or *Do not trigger* item update events.

When set to *Trigger*, the corresponding list workflows will launch upon item change in this list.

Create - Choose to *Create* or *Do not create* a new version of list items upon an update.

11.40 Move List Item to Subfolder

Move list item at this url to this subfolder. Store output item url in Variable: outputVariable .

See also: to copy or move list item(s) to a different list, use "Copy List item at URL" and "Delete Item at URL" actions.

This workflow action is used to move one or multiple list items to a subfolder in the same list. Version history and attachments for the list item are retained.

Parameters:

item at this url - specify the URL of a list item to move.

For current item, use **Current Item: Encoded Absolute URL** lookup.

To move multiple items, specify **array** of list item URLs. To obtain this array, use **Find List Item at Url** or **Find List Item by CAML** actions.

this subfolder - type in the name of a subfolder in the same list as the list item.

Variable: outputVariable - specify the string variable to store the URL of the list item in the new location.

11.41 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.

11.42 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.

11.43 Remove Content Type from List

Remove this content type from list at this list url.

This workflow action removes specified content type from the list.

Parameters:

this - Name of the content type.

this list url - Full URL of the target list.

11.44 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.

11.45 Set Content Type Document Template

Set document template url for this content type on the site at current site url to this url, copy document to the resource folder.

This workflow action sets a document template for a content type.

Parameters:

this content type - specify the content type name

current site - (optional) specify the URL of the site

to this url - specify the URL of the document that will be used as a template

copy - select to copy/do not copy the document template to the resource folder.

Examples:

Step 1

Create a new site column on the site at [current site](#) url based on `<Field Type="Text" Name="Test_Text_Fi... CAML`.

then Create a new content type on the site at [current site](#) url with [Word Document](#) title based on [Document](#) , put into [this sorting group](#) .

then Add [Test Text Field](#) existing column to [Word Document](#) content type on the site at [current site](#) url.

then Set document template url for [Word Document](#) content type on the site at [current site](#) url to `Current Item:Encoded Absolute URL` , [copy](#) document to the resource folder.

then Add [Word Document](#) content type to list at `Current Item:Encoded Absolute URL` .

CAML: `<Field Type="Text" Name="Test_Text_Field" DisplayName="Test Text Field" />`

11.46 Set Default Content Type on List

Set [this content type](#) as default content type on the list at [this list url](#).

This workflow action sets specified content type as default for the specified list. If list does not contain specified content type, then it will be added first.

Parameters:

this content type - Name of the content type.

this list url - Full URL of the target list.

11.47 Update List Item at Url

Update list item with [specified values](#) at [this list item url](#). [Trigger](#) events, [create](#) new version of list item.

This workflow action is used to set values of list item fields. It is possible to update **multiple** list items in a single action by providing an **array** of URLs as an input parameter.

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>
```

Note: XML format is case-sensitive.

See Also: "Create List Item at URL" action.

Note: With this action you can also update other items in SharePoint that are based on List functionality, e.g. **tasks** (and **sub-tasks** in SharePoint 2013), **knowledgebase articles** (in SharePoint 2013), etc. See "Create List Item at URL" action for examples.

Parameters:

specified values - Values for list item fields in INI or XML format.

a) For **Lookup** fields, specify only the ID of the lookup item:

Example:

Country=27

Multiselect Lookup fields are supported, values can be either comma (,) or semicolon (;) delimited.

See also: The IDs of the items can be obtained by using two actions: **Find List Item at URL** to find necessary item and **Get Field Value** to get its ID.

b) For **Hyperlink** type fields it is also possible to update description using *Url#description*:

Example:

Title=new task

Url=http://www.harepoint.com

Url#description=HarePoint website

c) **Multi-line text fields** - in order to transfer a multi-line text to a workflow properly, use only XML format in this action.

Example:

<field title="SomeLongText">Multi

Line

Text</field>

d) **Managed metadata** columns are supported, it is possible to specify term label names from Terms Set. Multiple labels should be delimited by a semicolon (;)

Example:

FieldName=Label1;Label2;Label3

e) **Choice** and **Checkboxes** types fields are supported. Specify the values delimited by a semicolon (;)

Example:

Title=New Order

Order=Pizza;Pasta;Black Tea

f) You can use a **special (alternative) format** to get values of workflow variables. To use it, the first line in this parameter should be **UseCustomVars** - this keyword enables this functionality. In the next lines you can use any workflow variable by its name: *%VariableName%*. This also supports arrays - you can get array item values as *%ArrayName[0]%*, *%ArrayName[1]%*, and so on.

Example (for INI format):

UseCustomVars

Title=%ResultArray[0]%

Order=%ResultArray[1]%

Size=%ResultArray[2]%

Ingredients=%ResultArray[3]%

In this example the list item fields will get the values of array variable *ResultArray*.

this - Full URL of list item. **Arrays** of full URLs are also supported, in this case **multiple** items will be updated.

Trigger - Specifies whether to trigger an item update event.

create new version - Specify to create or do not create a new version of a list item when it has been updated by this workflow action.

Examples:

Example 1: Updating a single item

Update list item with *Title=[%Current Item:Title%] (updated)* at *Current Item:Encoded Absolute URL list item url*. Trigger events, create new version.

Example 2: Updating multiple items

Find item in the list at *http://sitename.local/Lists/Tasks* where item's *Title equals test*. Store top 100 result urls in Variable: array.

then Update list item with *Title=test (updated)* at Variable: array list item url. Trigger events, create new version.

11.48 Update List Item on Different Farm

Update list item with specified values at this list item url.

This workflow action is used to set values of list item fields on different SharePoint farm.

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>
```

Important Note: the action '*Set Target Site Options*' always should be used before this action!

Additionally, you can use '*Set Encrypted Passwords*' action and Encrypt Credentials tool to specify the credentials in secure way.

See Also: '*Set Encrypted Passwords*' and '*Set Target Site Options*' actions.

Parameters:

specified values - Values for list item fields in INI or XML format.

this - Full URL of list item at different SharePoint farm.

11.49 Update Workflow Task at Url

Update workflow task with specified values at this url.

This workflow action allows to set the values of the workflow task properties, including extended properties.

One of the scenarios how this action can be used, is to complete a workflow task from a workflow:

<http://blogs.msdn.com/b/malag/archive/2009/05/20/how-to-programmatically-complete-approval-workflow-task.aspx>

Parameters:

specified values - Specify values workflow task properties in the following format:

```
<PropertyName1>=<value1>
```

```
<PropertyName2>=<value2>
```

...

this url - Specify the URL of the workflow task item.

11.50 Upload Files to List Item Attachments

Upload files from those paths to list item at this url. Overwrite existing items. Store number of uploaded files to Variable: variable.

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:

those paths - 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).

this url - 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.

11.51 Wait for Field Change in Current Item Until Date

Wait for field to equal value until date this time. Store timeout flag to Variable: isTimeOut.

This workflow action is waiting till field value satisfies the specified logical expression. If the specified date has reached, time out is triggered.

Parameters:

field - Specify the field to be checked.

to equal - Logical operand. Available values: To equal / To not equal.

value - Specify the target value.

this time - Specify the date when timeout will occur.

Variable: isTimeOut - 'False' when no timeout occurred, and 'true' when timeout has occurred.

11.52 Wait for Field Change in Current Item with Timeout

Wait for field to equal value with timeout 0 days, 0 hours, 5 minutes. Store timeout flag to Variable: isTimeOut.

This workflow action is waiting till field value satisfies the specified logical expression. If the specified time limit has been exceeded, time out is triggered.

Parameters:

field - Specify the field to be checked.
to equal - Logical operand. Available values: To equal / To not equal.
value - Specify the target value.
0 days - Days for timeout.
0 hours - Hours for timeout.
5 minutes - Minutes for timeout.
Variable: isTimeOut - 'False' when no timeout occurred, and 'true' when timeout has occurred.

12. Network activities

12.1 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.

12.2 Delete Directory from FTP Server

Delete this directory from FTP server. Do not delete directory if directory is not empty.

This workflow action is used to delete specified directory on the FTP server.

To specify server name, login and password place "Set FTP Server Options" activity the current workflow step first.

Parameters:

this - Directory name to delete.

Do not delete directory - Specifies whether to delete child items or not.

12.3 Delete File from FTP Server

Delete this file from this directory on FTP server.

This workflow action is used to delete one or several files (using wildcards) from the FTP server.

To specify the server name, login, and password, place "Set FTP Server Options" action to the beginning of a workflow.

Parameters:

this file - name of the file to delete.

To delete several files, you can use wildcards.

Example: *.docx; *.pdf

this directory - Directory name on the FTP server.

12.4 Download Document from FTP Server

Download this file from FTP server into document library at this url with the same name. Do not overwrite if the file already exists.

This workflow action is used to download a document from FTP server to a SharePoint document library.

Note: "Set FTP Server Option" action needs to be used before this workflow action to specify the parameters of the FTP server.

Parameters:

this file - specify the server-relative URL of the file on FTP server. (The FTP server URL is specified in "Set FTP Server Options" action earlier).

document library at this url - Encoded absolute URL of the document library, or a folder in it, where the downloaded document should be saved.

the same name - (optional) specify the new document name to be used. By default, the original name will be used.

Do not overwrite - specify if the file should be overwritten if it already exists: *Do not overwrite*, *overwrite*.

12.5 Download Documents from FTP Folder

Recursively download **all** files from **FTP folder** to **this folder url**. **Overwrite** existing documents. **Trigger** events

This workflow action is used to save either **all or only specific documents** from an FTP folder to a local disk or network share.

Necessary documents can be specified using a file mask, or CAML query.

With a special CAML query it is possible to copy only the **folders structure, without the documents** - see second parameter description for more details.

Note: "Set FTP Server Option" action needs to be used before this workflow action to specify the parameters of the FTP server.

Parameters:

Recursively - Two options available:

- *Recursively*: save the content of a document library or a folder including all subfolders; subfolders tree will be created in *specified path*
- *Non-recursively*: save files only from specified document library of folder.

all - (Optional) By default (*all* setting), **all documents** will be downloaded. If you need to download only **certain files**, you can use two methods to specify them:

1. File mask, for example *.jpg. You can specify only one file mask.
2. CAML query - it should start with <Where>, for example:

```
<Where>  
<BeginsWith>  
<FieldRef Name='FileLeafRef' />
```

```
<Value Type='File'>test</Value>
</BeginsWith>
</Where>
```

If you need to copy only the **documents structure**, without the documents, use the CAML query below:

```
<Where><Eq><FieldRef Name='FSObjType' /><Value Type='Integer'>1</Value></Eq></Where>
```

FTP folder - specify full URL of the source FTP folder

this folder url - The full URL of a document library or a folder in a document library to store the downloaded documents

Overwrite - If the folder already contains file with the same name, then you can apply one of 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 events - specify where to *Trigger* or *Do not trigger* an item creation/update event in a target document library

12.6 Fetch RSS Feed to Html

Fetch RSS from this url print by `<p>[DATE]: [TITLE]</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]: [TITLE]</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.

12.7 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.

12.8 Get Formsite Form Results

Get Formsite form results from this form api url starting from this result id, use this api key. Store results in Variable: ResultsArray and result id in Variable: ResultId.

This workflow action is used to obtain the Formsite form results and save them to array variable.

Parameters:

this form api url - Specify the form URL from the page Forms - Integrations - API.

<https://fs7.formsite.com/bld/FormSite FormId=LoadApi>

Usually looks as follows:

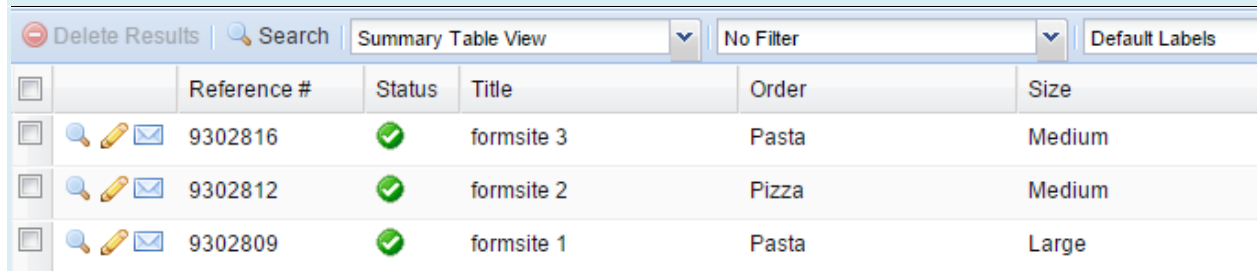
<https://fs7.formsite.com/api/users/<username>forms/form1/>

starting from this result id - (optional) Specify the result ID that you need to get. If there's no such ID found, the nearest next result ID will be used.

The result ID that has actually been used to retrieve data will be saved to a variable - see last parameter in this action.

Example:

The result IDs are called Reference # in this example:



	Reference #	Status	Title	Order	Size
	9302816		formsite 3	Pasta	Medium
	9302812		formsite 2	Pizza	Medium
	9302809		formsite 1	Pasta	Large

this api key - Specify the API key from Forms - Integrations - API.

<https://fs7.formsite.com/bld/FormSite FormId=LoadApi>

Variable: ResultsArray - Specify the array variable to store the result.

The order of the data in this output array is pre-defined, you can check it on Forms - Integrations - API, Forms Items section.

Items:

ID	Text
0	Title
1	Order
2	Size
3	Ingredients
4	How many?
5	Address
7	Comments

The array will contain the following:

ResultArray[0] - Title

ResultArray[1] - Order

ResultArray[2] - Size
ResultsArray[3] - Ingredients
etc.

Variable: ResultID - The ID of the result that has actually been obtained.

In case if the 2nd parameter in this action contains ID that has not been found, the nearest next result ID will be used, and its ID will be stored in this variable.

12.9 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.

Additional information: If the web server returns an error (e.g. HTTP 401 or HTTP 500) the workflow will be interrupted. It is possible to set the number of retries for sending an HTTP request; to do this, right-click the action in SharePoint Designer and set *NumberOfRetries* parameter to values 5...10.

Parameters:

this url - Target URL with parameters.

this headers - (Optional) Additional request headers, each on new line.

For example:

Accept-Language: fr

Accept: application/json

User-Agent: workflow extensions app

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=\[%Variable: param%\]](http://mysite/q=[%Variable: param%]) using GET method with this headers. Store response in Variable: pageBody and status code in Variable: code.

12.10 Send HTTP OAuth Request

Send HTTP OAuth request to this url using GET method with these headers and this request. Store response in variable.

This workflow action is used to send HTTP OAuth request. Various HTTP methods available.

Note: Set **OAuth Authentication** action needs to be used prior this action.

See also: **Set OAuth Authentication** action.

Parameters:

this url - This is the API endpoint, the URL of the web-resource to send the request to.

Please refer to "Developers API" documentation for details.

using GET method - Select the HTTP method. Possible values:

- GET
- POST
- PUT
- DELETE

Please refer to "Developers API" documentation to find out which method should be used.

these headers - (Optional) Specify the additional headers for this request, if necessary.

this request - (Optional) Specify the request body, if necessary.

Variable - Specify the variable where the server response will be saved to.

Examples:

How to work with xero.com: Creating a new contact

(see also **Set OAuth Authentication** action)

Step 1. Obtain the base64-encoded private key. It can be generated using the command line utility *openssl*:

`openssl gensra -out privatekey.pem 1024`

More details here: <http://developer.xero.com/documentation/advanced-docs/public-private-keypair/>

Step 2. Obtain consumer key and consumer secret here: <https://app.xero.com/Application/List>

Step 3. Set up a workflow (this is the example to **create a new Contact**):

- Set xero.com OAuth authentication. Use LLWHH7WS505 key, PRNCI5KNCHU76NSP secret, sign request with RSA-SHA1 MIICWwIBljchPkmJKwcOL..., request token at [this url](#), get access token at [this url](#)
- then Send HTTP OAuth request to <https://api.xero.com/api.xro/2.0/cont...> using PUT method with [these headers](#) and `<Contact> <Name>ABCDEFG</Name> <F... request`. Store response in Variable:variable
- then Log Variable:variable to the workflow history list

Comments for workflow action 1:

- The consumer key and consumer secret from **Step 2** are used in this action in respective parameters
- The base64-encoded private key from **Step 1** is used in the additional parameter after RSA-SHA1

Comments for workflow action 2:

According to the xero.com API documentation, the parameters are filled as follows:

- request URL is <https://api.xero.com/api.xro/2.0/Contacts>
- HTTP method that is used is PUT
- request body is:

```
<Contact>
<Name>ABCDEFG</Name>
<FirstName>John</FirstName>
<LastName>Smith</LastName>
<EmailAddress>john.smith@gmail.com</EmailAddress>
<Addresses>
<Address>
<AddressType>POBOX</AddressType>
<AddressLine1>P O Box 123</AddressLine1>
<City>Wellington</City>
<PostalCode>6011</PostalCode>
```

```
</Address>
</Addresses>
<BankAccountDetails>01-0123-0123456-00</BankAccountDetails>
<TaxNumber>12-345-678</TaxNumber>
<AccountsReceivableTaxType>OUTPUT</AccountsReceivableTaxType>
<AccountsPayableTaxType>INPUT</AccountsPayableTaxType>
<DefaultCurrency>NZD</DefaultCurrency>
</Contact>
```

12.11 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 - (Optional) Additional request headers, each on new line.

For example:

Accept-Language: fr

Accept: application/json

User-Agent: workflow extensions app

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.

12.12 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 send 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

Examples:

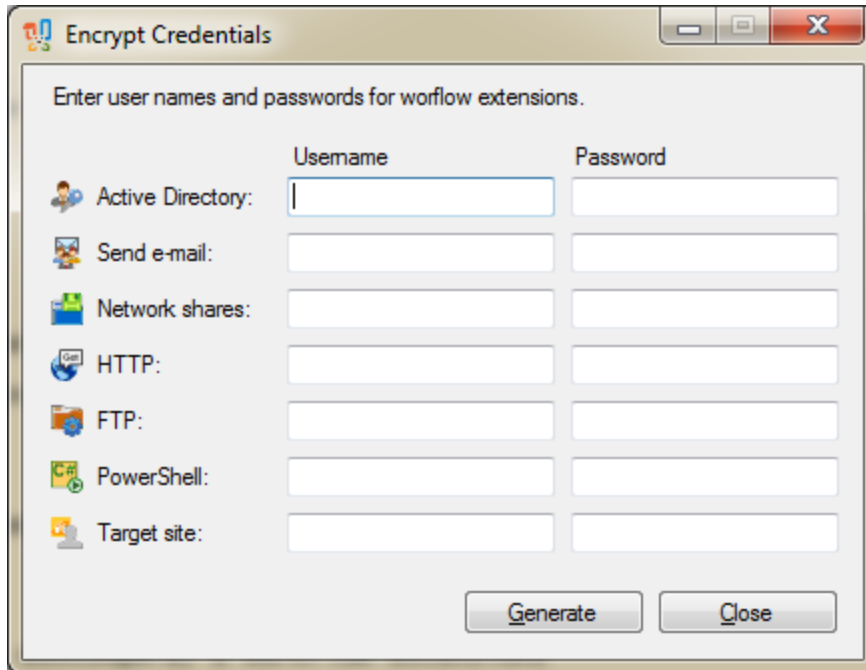
Send XMPP message *Hello!* to *myfriend@gmail.com* by using JID: *myaccount@gmail.com*, password: *mypassword* and optional server: *talk.google.com*.

12.13 Set Encrypted Passwords

Set encrypted passwords with this string.

This workflow action is used to setup authorization settings in an 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, which is installed as a part of **Client Components** by the product Installer:



Parameters:

this - Base64 string generated by the "Encrypt Credentials" tool.

12.14 Set FTP Server Options

Use specified FTP server and specified protocol, login as this user and this password.

This workflow action is used to set FTP server name, protocol, 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.

Later versions also support SFTP (SSH) 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.

You can specify port with a colon, like `ftp.company.com:121`.

specified protocol - specify the protocol for the connection. Available options:

- *FTP (no SSL)* - simple FTP connection without encryption. Default port is usually 21.
- *FTPS (FTP over SSL)* - FTP connection using SSL. Default port is usually 990.
- *SFTP (SSH)* - Transferring files over SSH. Default port is usually 22.

Note: SFTP is only supported in versions v2.19.4163 and later.

login - FTP server login. If empty, *Anonymous* user name is used.

Note: for SFTP mode, login and password are required.

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.

12.15 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 authentication.

this password - Password that will be used for HTTP authentication.

12.16 Set OAuth Authentication

Set [xero.com](#) OAuth authentication. Use [this consumer key](#), [this consumer secret](#), sign request with [default](#), request token at [this url](#), get access token at [this url](#).

This workflow action is used to set OAuth authentication.

See also: **Send HTTP OAuth Request** action

Parameters:

xero.com - select the site to authenticate against. Possible values:

- xero.com
- general

this consumer key - Specify the consumer key used for authentication. You can obtain it from the respective web site, usually in "Developers API" section.

this consumer secret - Specify the consumer secret used for authentication. You can obtain it from the respective web site, usually in "Developers API" section.

sign request with default - Select the method to sign the requests to the server.

Possible values:

- default
- plain text
- hmac-sha1
- rsa-sha1 (**required for xero.com**)

Please refer to "Developers API" section for more details.

When **rsa-sha1** is selected, an **additional parameter** would need to be filled:

RSA key - Paste here the base64-encoded private key from **.pem** file. (The instructions how to obtain .pem file are available in the description on the respective site).

request token at this url - (Optional) Specify the URL to obtain the request token. It is used during the authentication process, e.g. for digital signature.

access token at this url - (Optional) Specify the URL to exchange request token with access token. It is used to access API functionality.

Examples:

See **Send HTTP OAuth Request** action for an example on how to use these actions for **xero.com**.

12.17 Set Web Service Options

Use default web service base url and this username and this password.

This workflow action is setting the Web Service options which are required to use 'Get Excel Cell Value' and 'Set Excel Cell Value' actions when older format (.xls) Excel documents are processed.

Note: This action is not required for newer format (.xlsx) Excel documents.

Parameters:

default web service - (optional) You can specify URL for Excel Web Services in case if it's not detected properly with the default setting.

Note: Site address URL should be used.

this username - Specify the username.

this password - Specify the password.

Note: You can use the Encrypt Credentials tool to encrypt the password, in this case specify the Username and Password in Web Service field and use 'Set encrypted password' action.

12.18 Shorten Url

Shorten this url with goo.gl using this login and this password and store result in Variable:variable.

This workflow action generates the short URL using online providers for short URLs.

Parameters:

this url - Specify URL to be shortened.

with goo.gl - Select the provider

Currently two provides available:

- Goo.gl - does not require registration, so login and password are not required
- Bit.ly - requires registration on Bitly.com, login and password should be specified in action parameters.

this login - Specify the login name (only for bit.ly).

this password - Specify the password (only for bit.ly).

store result in Variable:variable - The shortened URL is stored in this variable.

12.19 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.

12.20 Upload Documents to FTP Folder

Recursively upload all documents from this folder url to FTP folder. Overwrite existing files.

This workflow action is used to upload either **all or only specific documents** from a folder in a document library to an FTP folder.

Necessary documents can be specified using a file mask, or CAML query.

With a special CAML query it is possible to copy only the **folders structure, without the documents** - see second parameter description for more details.

Note: "Set FTP Server Option" action needs to be used before this workflow action to specify the parameters of the FTP server.

Parameters:

Recursively - Two options available:

- *Recursively:* save the content of a document library or a folder including all subfolders; subfolders tree will be created in *specified path*
- *Non-recursively:* save files only from specified document library or folder.

all - (Optional) By default (*all* setting), **all documents** will be downloaded. If you need to download only **certain files**, you can use two methods to specify them:

1. File mask, for example *.jpg. You can specify only one file mask.
2. CAML query - it should start with <Where>, for example:

```
<Where>  
<BeginsWith>  
<FieldRef Name='FileLeafRef' />  
<Value Type='File'>test</Value>  
</BeginsWith>  
</Where>
```

If you need to copy only the **documents structure**, without the documents, use the CAML query below:

```
<Where><Eq><FieldRef Name='FSObjType' /><Value Type='Integer'>1</Value></Eq></Where>
```

this folder url - the full URL of a source folder

FTP folder - specify the URL of the target FTP folder

Overwrite - If the folder already contains file with the same name, then you can apply one of 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.

12.21 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.

13. Office activities

13.1 Accept Revision Changes in Word Document

Accept all revision changes in Word document at this url.

This workflow action accepts all revision changes in the specified Word document.

Note: To update the fields in the document, conversion *docx to docx* is required. Please refer to the following screenshot for proper usage of this action:

Step 1

Accept all revision changes in Word document at Current Item:Encoded Absolute URL .

then Convert Current Item document to Word Document (.docx) format and store job ID in Variable: ConversionJobId . Overwrite existing document.

then Wait while conversion job Variable: ConversionJobId is running.

then Copy document at Current Item:Encoded Absolute URL to the folder at http://wss14-2.vlab.mapilab.local:133... and with the same name. Overwrite existing document.

Parameters:

this url - The URL of the Word document.

Examples:

Accept all revision changes in Word document at Current Item:Encoded Absolute URL.

13.2 Add Digital Signature

Add digital signature to the document at this url using certificate at this url protected with this password

This workflow action is used to add a digital signature to the specified document. PFX certificate file and password are needed for this procedure.

Parameters:

document at this url - specify the full URL to a document that needs to be signed.

To point to a current document (that workflow has triggered for), you can use Current Item: Encoded Absolute URL lookup.

certificate at this url - specify the full URL to a .pfx certificate file.

this password - specify the password for the PFX certificate provided

13.3 Add New Row into Excel Document

Add new row from this array into specified sheet in this document. Start looking for the last row from this cell.

This workflow action adds a new row to Excel document. It can automatically detect the nearest empty row to paste the values.

Note: Only .xlsx files are supported.

Parameters:

this array - The array of values that will be added. The array can be created using "Create Array" or "Split Text" actions.

specified sheet - The sheet name in Excel document.

this document - The full URL to Excel file.

this cell - If this cell is empty, the values from the array will be added to this cell and following cells in the same row.

If this cell is not empty, this workflow action will first check next cells in the same column if they are empty. As soon as empty cell is found, the values from the array will be added to this cell and following cells in the same row.

13.4 Add New Row into Word Document

Add new row into this repeating section content control of this .docx document.

This workflow action adds a copy of the last row in the Repeating Section Content Control. After this has been added it can be modified using 'Update Word Document' or 'Update Multiple Fields in Word Document' actions.

To create a Repeating Section Content Control, highlight the necessary table row in Word Document, and on the ribbon on Developer Tab click 'Add Repeating Section Content Control' button.

Parameters:

this repeating section - The name of the repeating content control.

this .docx document - The URL of the word document to be modified.

Examples:

Add new row into Repeating content control of Current Item document

An article with detailed example can be found here:

<https://www.harepoint.com/Blog/sharepoint-workflow-repeating-content/>

13.5 Add Watermark To Word Document

Add image at this url as a watermark to the Word document at this url

This workflow action is used to add watermarks to Word documents.

Note: Only **DOCX** format (OpenXML) of Word documents is supported.

See also: "Add watermark to PDF document" and "Set PDF Watermark Options" actions to add watermarks to PDF documents.

Parameters:

image at this url - specify the full URL of an image to be used as a watermark

Word document at this url - specify the full URL of a Word document to add a watermark to.

13.6 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.

13.7 Get Excel 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 on a SharePoint server.

Note: For older format (.xls) Excel documents, this workflow action requires Excel Services installed on the server, as well as "Set Web Services options" action need to be placed prior this action.

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.

13.8 Get Office Document Property

Get Custom property name 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
- Time Created
- Time Modified
- Custom*

*Note that you can also get custom properties by selecting *Custom*. In that case, an additional parameter appears where you can type the name of the custom property.

In the current version, only Open Office XML (Microsoft Office 2007-2013) 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:

Custom - Select the property from the list of available properties, or select *Custom* to get the custom property. In that case, an additional parameter appears where you can type the name of the custom property.

property name - (only visible when previous parameter is set to Custom) Type the custom property name.

this - Reference to document, e.g. Current Item.

Variable: variable - Output string variable.

13.9 Get Word Document Page Count

Get the page count of this document and save to Variable: variable

This workflow action is used to get the page count of a Word document. Only .docx (Word 2007 and later) version is supported.

Note: the .docx file only stores the static information about the page count, which is updated when the document is rendered by MS Word application. That is, this count **may be incorrect** if the document was closed before rendering was complete, or if specific views (such as "Web document" view) have been used.

Parameters:

this document - select the document using the lookup controls.

save to Variable: variable - specify the output variable, to store the page count.

13.10 Read Word Document

Read this content control of this .docx document and store its value in Variable: variable. If the control wasn't found, stop workflow.

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.

stop workflow - Choose if the workflow should stop with an error (*stop workflow*), or continue running (*continue workflow*).

13.11 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.

13.12 Replace External Hyperlink in Word Document

Replace all occurrences of this hyperlink with this hyperlink in this document, apply changes to: both URL and text , match case: no , increment version number: yes .

This workflow action is used to replace the URLs and/or text labels in external hyperlinks in Word document.

Note: only DOCX (OpenXML) format documents are supported.

Parameters:

all occurrences - choose which of the matching hyperlinks should be replaced:

- *all occurrences*
- *first occurrence*
- *last occurrence*

this hyperlink - specify the text string to search in hyperlink URL and/or text label

this hyperlink - specify the replacement text string.

Note: When replacing URLs, make sure that resulting string after replacement will be a valid URL.

this document - choose the document that should be changed.

both URL and text - choose if *only text labels/only URLs/both* should be replaced

match case: **no** - choose if search and replacement should be case sensitive, or not.

increment version number: **yes** - choose to create a new version of a document, or keep changes within the current version.

13.13 Replace Text in Word Document

Replace **all occurrences** of **this string** in **this document** with **this text**, match case: **no**, increment version number: **yes**.

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 to update content control fields, 'Replace external hyperlinks in Word document' action to replace hyperlink URLs and/or text labels.

Note: Word Automation Services are not required for this workflow action.

Parameters:

all occurrences - With this action it is possible to replace *all occurrences*, *first occurrence* or *last occurrence* of specified pattern.

this string - String to replace.

this - Reference to document, e.g. Current Item.

this text - Specify the new string.

no - Case sensitive: *yes* or *no*.

yes - If the document library has versioning enabled, you can choose, if document version number should be increased (*yes*, default) or not (*no*).

If *no* is selected, the last version of a document will be overwritten; **modified date** and **modified by** fields will be updated and set to the current date and current user respectively. The version number will not be changed.

13.14 Set Excel 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 on a SharePoint server.

Note: For older format (.xls) Excel documents, this workflow action requires Excel Services installed on the server, as well as "Set Web Services options" action need to be placed prior this action.

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.

this value - String, Integer, Number, Date/Time or Boolean variable with a new cell value.

13.15 Set Office Document Property

Set this property of this office document 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
- Time Created
- Time Modified

In the current version, Open Office XML (Microsoft Office 2007-2013) 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.

13.16 Sign PDF Document

Sign the document at this url using image at this url and put it on page 1 at top left corner, indent left: 1 in., top: 1 in

This workflow action adds the specified picture to a PDF document.

Note: To add picture to Word document, refer to "Update Picture in Word Document" action.

Parameters:

document at <u>this url</u> - specify the full URL of a PDF document to add a picture.
image at <u>this url</u> - specify the full URL of a picture to be added.
Supported picture formats: gif, png, jpg, bmp, tiff, wmf, emf, ico.
page <u>1</u> - specify the page number in PDF document to add the picture to.
You can use '-1' to point to the last page in a document.
top left corner - specify where the picture should be located on the page:
<i>top left, top center, top right, middle left, middle center, middle right, bottom left, bottom center, bottom right</i>
left: <u>1 in</u> - specify the picture indent from the left side, in inches
top: <u>1 in</u> - specify the picture indent from the top, in inches

13.17 Update Multiple Fields In Word Document

Set values of the following content controls to the specified values (all occurrences) for this document, text format: plain text.

This workflow action is used to update multiple content control fields in Word document in a single run, which is more convenient and efficient than using "Update Word Document" action several times in a workflow.

See also: "Update Word document" action to update a single content control field.

Note: only DOCX (OpenXML) Word files are supported.

Parameters:

the specified values - specify the content control field names and their new values. Two formats are supported:
INI-style:
control name1=value1 control name2=value2 ...

XML-format:

```
<document>
<field title="control name1">value1</field>
<field title="control name2">value2</field>
<field title="control name3">value3</field>
...
</document>
```

Note: New line breaks are only possible with Rich Text content control fields, **
** tag and **HTML** format selected - see below.

all occurrences - specify which content control fields should be updated if the document has several content control fields with the same name:

- *all occurrences*
- *last occurrence (repeating content control)*

this document - select the Word document stored in a document library to be updated.

You can use the lookup "Name (for use in forms) - documentname.docx"

text format - select the format for **all** content control fields that will be updated. Two options available:

- *plain text*
- *HTML* - supports tags for text formatting, including **
** tag for new line break

Note: If you need to update content controls of different formats, either group them by the same formats and update them in two "Update Multiple Fields In Word Document" actions, or use "Update Word Document" action for each content control individually.

13.18 Update Picture in Word Document

Set this picture content control of this document to picture at this url.

This workflow action sets the picture located at specified URL to picture content control of the specified Word document.

See Also: 'Update Word Document' action to update text in Word document; 'Sign PDF Document' action to add picture to PDF document.

Parameters:

this picture content control - The picture content control to be set.

this document - The URL of the document to be modified.

this url - The direct URL to the picture within this SharePoint farm.

Examples:

Set *logo of Invoices* document to picture at <https://spsite/Shared%20Documents/newpic.jpg>

13.19 Update Word Document

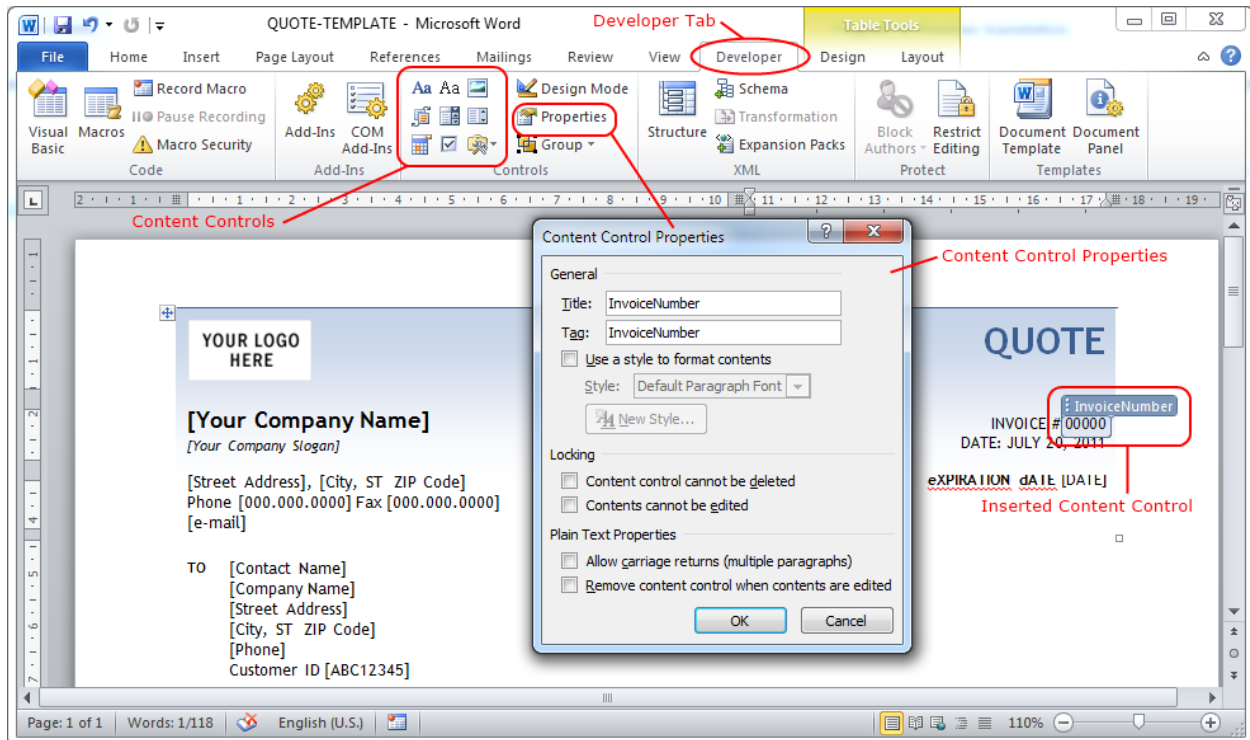
Set value of this content control (all occurrences) of this .docx document to this value, text format: plain text.

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 a content control, you should first enable **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; 'Update Multiple Fields In Word Document' action to update multiple content control fields in one run; 'Replace external hyperlinks in Word document' action to replace URLs and/or text labels.

Note: Word Automation Services are not required for this workflow action.

Parameters:

this content control - Content control name, see overview section for details.

all occurrences - specify which content controls with the specified name should be updated, if the document has several content controls with the same name:

- *all occurrences*
- *last occurrence (repeating content control)*

this .docx document - specify DOCX document to update

this value - String with the new text for content control.

Checkbox type content controls are also supported, in this case you need to specify *true* value to set the checkbox as checked, and *false* value to set the checkbox as unchecked. **Text format** parameter needs to be set to *plain text*.

plain text - Specify the format of the text. Available values: *plain text*, *HTML*.

For **checkbox** type content controls, set this parameter to *plain text*.

Examples:

Example 1: text content controls

Set value of *InvoiceNumber (all occurrences)* of *Current Item .docx* document to 12345, text format:HTML

This workflow action replaces the content of "InvoiceNumber" content control with 12345.

Example 2: checkbox content controls

Set value of *Checkbox1 (all occurrences)* of *Current Item .docx* document to true, text format: plain text

This workflow action sets the "Checkbox1" content control as checked.

See also: overview section for instructions on how to place a named content control to a Microsoft Word document.

14. Permissions activities

14.1 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.

The action can **process multiple items** if an array of URLs is provided.

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.

Arrays of URLs are also supported. In this case all items from array will be processed.

Examples:

Set *Variable: item* to *[%Workflow Context:Current Item URL%]*.

then *Stop inheriting permissions* at *Variable: item*.

14.2 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.

14.3 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.

14.4 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.

Permissions for **multiple items** can be deleted with this action if an array of URLs is provided.

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.

Arrays of URLs are also supported. In this case all items from array will be processed.

14.5 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.

14.6 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.

14.7 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.

Permissions for **multiple items** can be deleted with this action if an array of URLs is provided.

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.

Arrays of URLs are also supported. In this case all items from array will be processed.

this user - All permissions for this user or group will be removed.

14.8 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.

14.9 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:

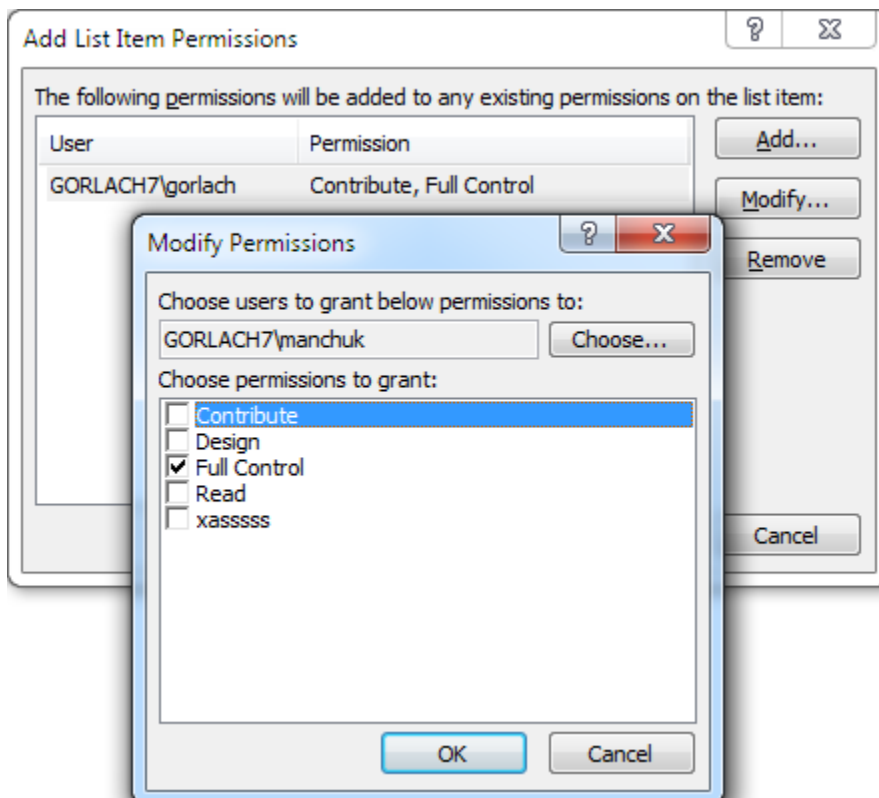
- this list item** - Target list item.
- this user** - All permissions for this user or group will be removed.

14.10 Delete Permissions at Url

Delete specified permissions at this url.

This workflow action 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.



Permissions for multiple items can be deleted with this action if an array of URLs is provided.

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.

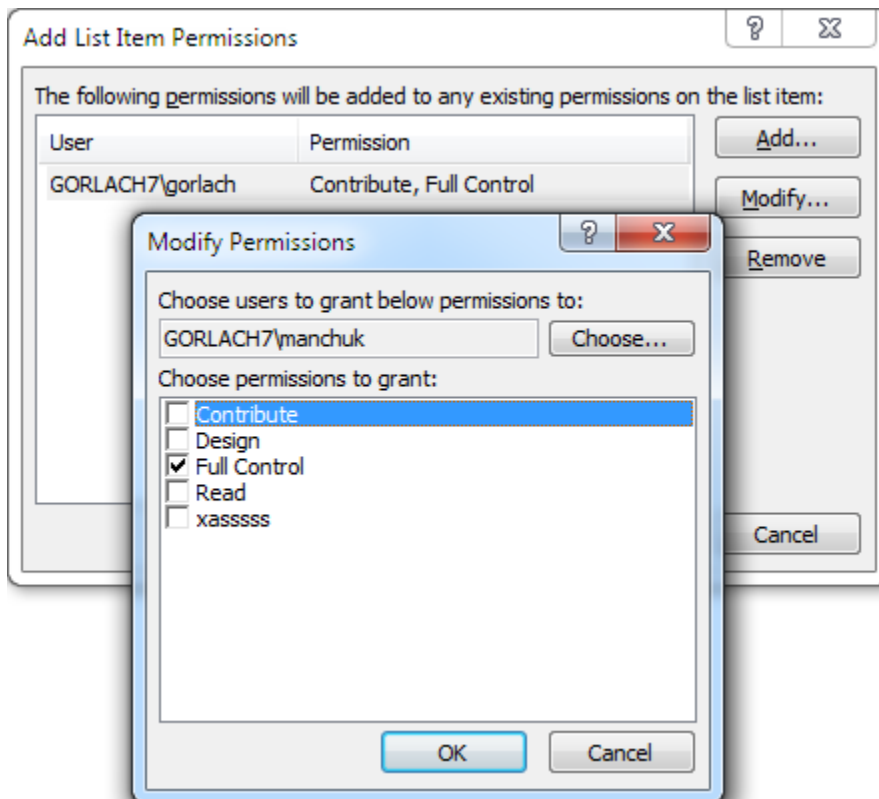
Arrays of URLs are also supported. In this case all items from array will be processed.

14.11 Delete Permissions from List

Delete specified permissions from this list.

This workflow action 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.

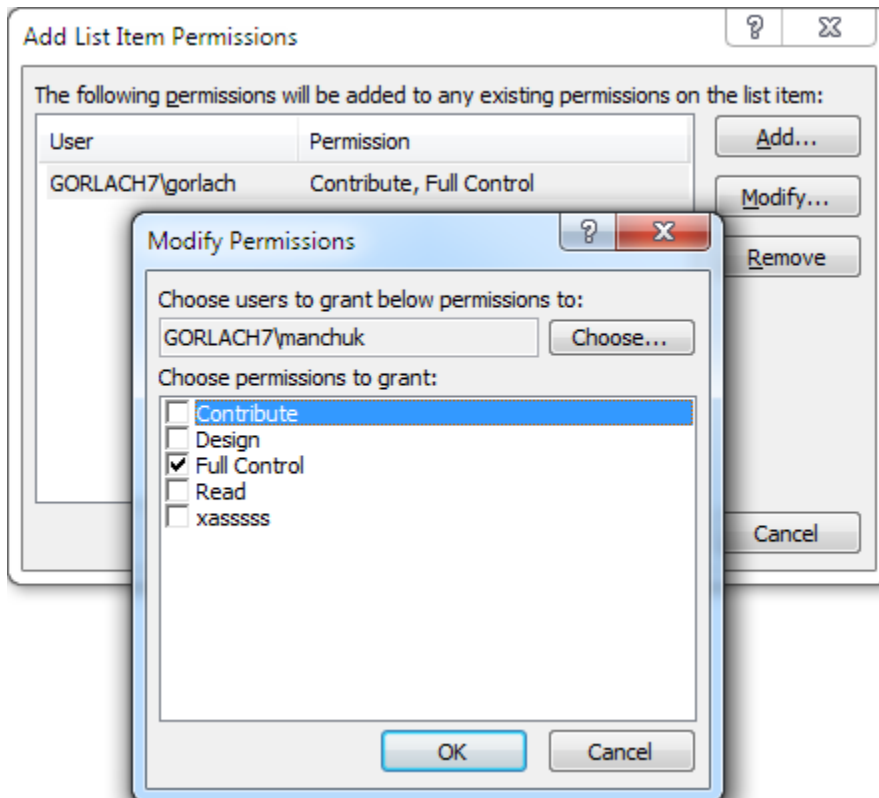
this list - Target list.

14.12 Delete Permissions from List Item

Delete **specified** permissions from **this list item**.

This workflow action 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.

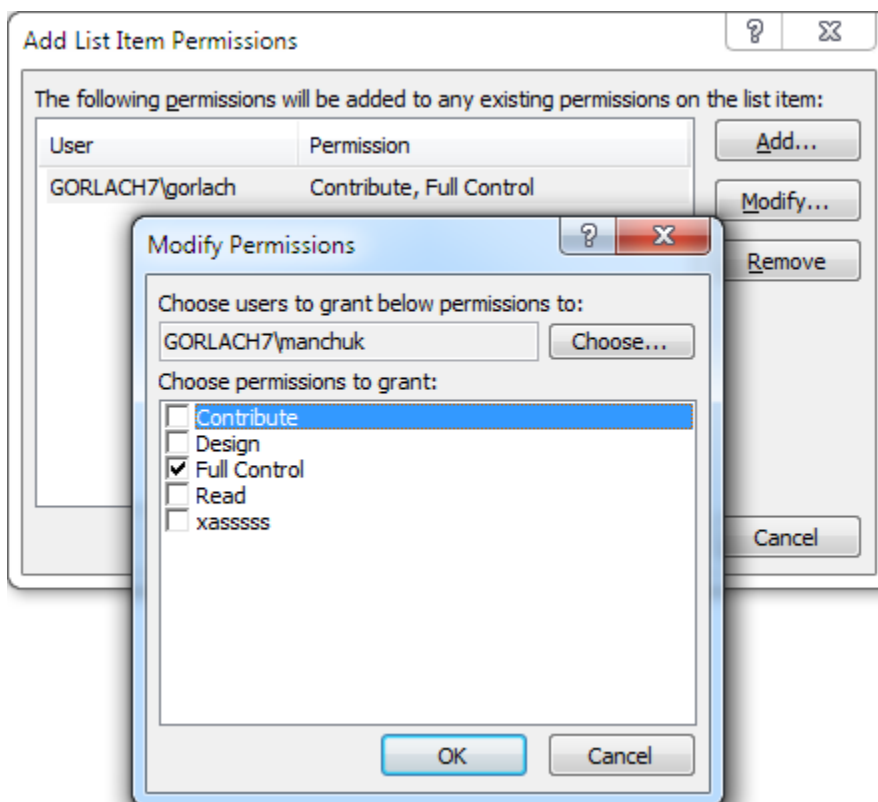
this list item - Target list item.

14.13 Grant Permissions at Url

Grant **specified** permissions at **this url**.

This workflow action is used to grant permissions for site users to a 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.

Note: The specified permissions will be added to existing permissions (if any) for an item. In case, if you need to be sure that only the specified (and no other) permissions are granted, use "**Set permissions**" action instead.

Parameters:

specified - Special-type array of the users and their permissions.

this url - Full URL of a SharePoint site, list, list folder or list item.

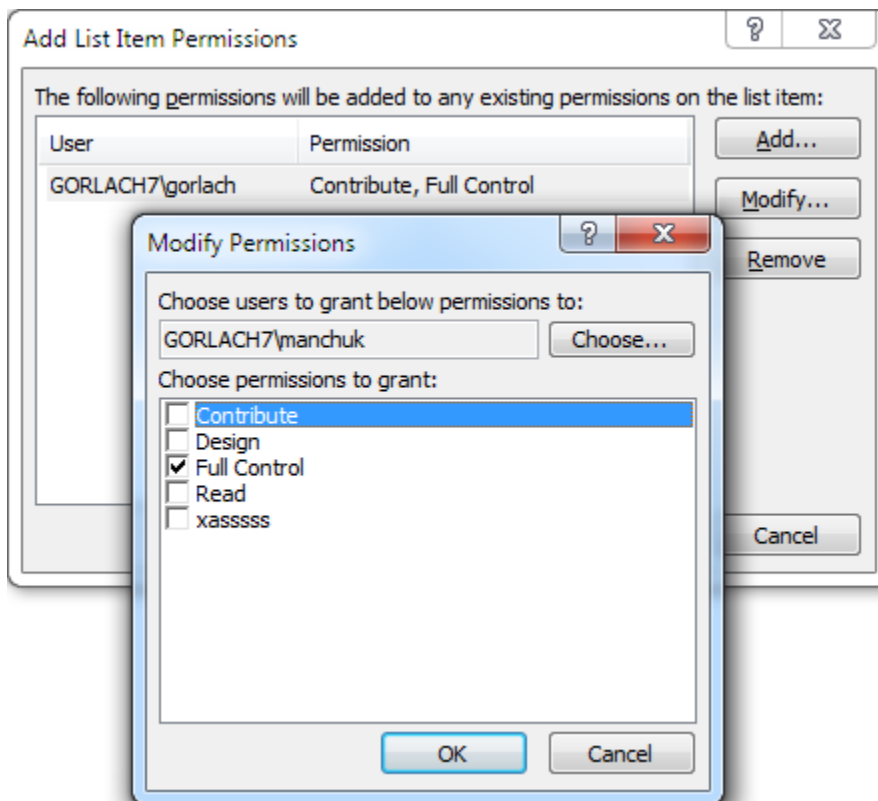
Arrays of URLs are also supported. In this case all items from array will be processed.

14.14 Grant Permissions on List

Grant **specified** permissions on **this list**.

This workflow action 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.

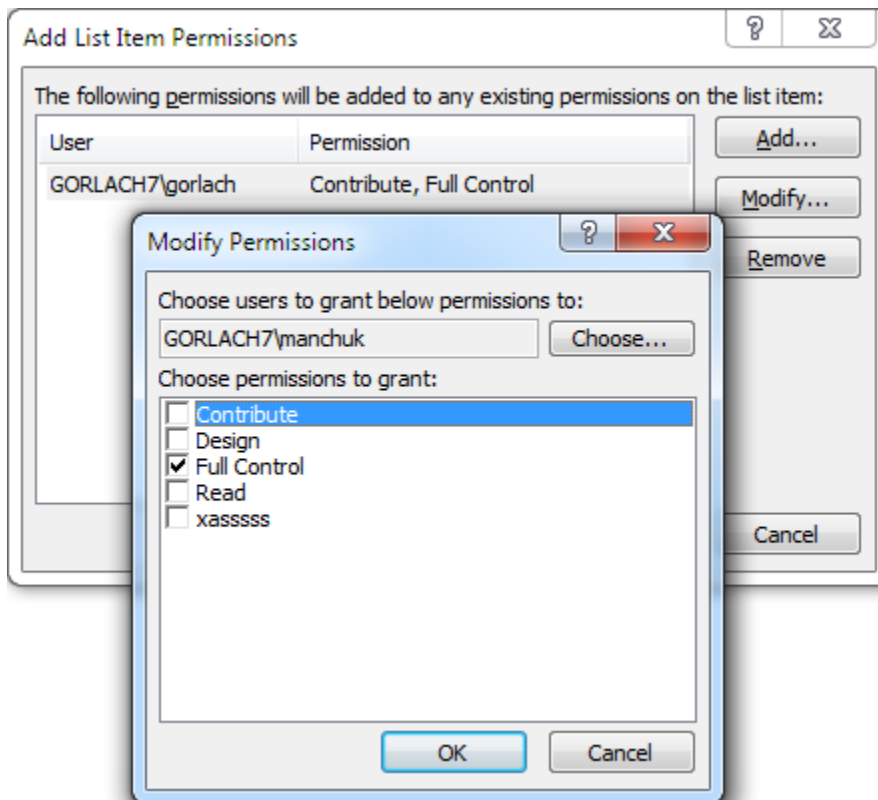
this list - Target list.

14.15 Grant Permissions on List Item

Grant specified permissions on this list item.

This workflow action 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.

14.16 Is Permission Inheritance Set (HarePoint)

If permission inheritance set at this url.

This workflow condition allows to find out if permission inheritance is set for an object at the specified URL, or if permission inheritance has been broken.

Parameters:

this url - Specify the URL of the object to check its permission inheritance status.

14.17 Set Permissions at Url

Remove all existing permissions and set specified permissions at this url.

This workflow action is used to set the permissions as specified in this action.

Note: It removes any other permissions granted earlier to ensure that only indicated permissions are applied.

If you only need to add specific permissions without removing the existing ones, use "Grant permissions" action instead.

Parameters:

specified permissions - Specify permission(s) this item or document needs to have. Note that any other permissions will be removed.

this url - Specify the URL of a document or a list item.

The proper URL can be obtained by using one of the following:

- **Current Item: Encoded Absolute URL** lookup
- **Find list item at URL** action
- **Get List Item Url** action with **Native form** in the first parameter

15. Printing activities

15.1 Print Document

Print document at this url to default printer.

This workflow action is used to print specified document or image to a specified printer.

List of supported formats:

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

An **array** of document URLs is also supported; in this case, action will print each document from an array.

Be sure to send to printing the reasonable amount of documents.

Array can be obtained e.g. from "Find List Item at URL" action.

default - (optional) Network name of the printer (UNC path), e.g. `\\Printserver\Laser01`

Leave it as *Default* to print documents to a printer which is set as a default printer for the account that runs HarePoint Printing Service.

Examples:

Print document

at <http://sp14.vlab.mapilab.local:133/Shared%20Documents/Test.docx> to `\\KOENIG\OKIB410` printer.

15.2 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.

16. Social activities

16.1 Create Group in Yammer

Create new public group in Yammer with this title, this description, use this access code.

This workflow action creates a new group in Yammer with specified title and description. The group can be public or private.

Parameters:

public group - Choose if the new group should be *public* or *private*.

this title - Specify the title for the group.

this description - Specify description for the group.

use **this access code** - Specify the access code:

There are two ways to receive it:

- Using **Social Authorize Tool** from **Client Components** of HarePoint Workflow Extensions (use Installer to install Client Components)
- Using the following link: <http://auth.mapilab.com/Yammer/RequestToken>

16.2 Create MailChimp Campaign

Create a new MailChimp campaign from html with this subject and this body to recipient name and recipient emails list. Use this api key, store campaign id in Variable: variable.

This workflow action is used to create a new MailChimp campaign.

Note: this action only creates a campaign, but doesn't send it.

See also: "Send MailChimp campaign" action.

Parameters:

html/plain text - Select if e-mail body should be in HTML or Plain Text.

this subject - The e-mail subject, and also campaign name on MailChimp site.

this body - The e-mail body.

recipient name - (Optional) The recipient name for "To:" field.

Macros are supported:

- E-mail address *|EMAIL|*
- First name *|FNAME|*
- Last name *|LNAME|*
- Unsubscribe link *|UNSUB|*
- Forward to a Friend Link *|FORWARD|*
- Update Profile Link *|UPDATE_PROFILE|*
- MonkeyRewards Affiliate Link *|REWARDS|*

recipient emails list - the name of the recipient e-mails list from MailChimp site. This list contains e-mails of the recipients for this campaign.

this api key - The authorization key.

See: <http://kb.mailchimp.com/accounts/management/about-api-keys>

store campaign id in Variable:variable - Specify the variable where an ID for this campaign will be stored.

16.3 Create User in Yammer

Create user in Yammer with this e-mail, this full name, this job title, this department, use this access code.

This workflow action creates a new user in Yammer network with specified parameters.

Parameters:

this e-mail - Specify the e-mail address for the new user.

this full name - Specify the full name for the new user.

this job title - Specify the job title for the new user.

this department - Specify the department for the new user.

use **this access code** - Specify the access code.

There are two ways to receive it:

- Using **Social Authorize Tool** from **Client Components** of HarePoint Workflow Extensions (use Installer to install Client Components)
- Using the following link: <http://auth.mapilab.com/Yammer/RequestToken>

16.4 Delete MailChimp Campaign

Delete **this** MailChimp campaign permanently. Use **this** api key.

This workflow action is used to delete **MailChimp** campaign permanently.

Important note: the deleted campaign cannot be restored!

Parameters:

this MailChimp campaign - specify either *ID* or *name* of the campaign to be deleted.

Note: If the name is specified and there are multiple campaigns with this name, the latest created campaign will be removed.

this api key - the authorization key. See <http://kb.mailchimp.com/accounts/management/about-api-keys>

16.5 Delete User from Yammer

Delete user with **this** e-mail from Yammer, use **this** access code.

This workflow action deletes the specified user from Yammer.

Parameters:

user with this e-mail - Specify the e-mail address of the user.

use **this access code** - Specify the access code.

There are two ways to receive it:

- Using **Social Authorize Tool** from **Client Components** of HarePoint Workflow Extensions (use Installer to install Client Components)
- Using the following link: <http://auth.mapilab.com/Yammer/RequestToken>

16.6 Get User's Property from Yammer

Get specified property of Yammer user with this e-mail address, use this access code, store result in Variable: variable.

This workflow action gets the specified property from the profile of the specified user in Yammer.

Parameters:

specified property - Specify the profile property to get.

user with this e-mail address - Specify the user's e-mail address.

use **this access code** - Specify the access code:

There are two ways to receive it:

- Using **Social Authorize Tool** from **Client Components** of HarePoint Workflow Extensions (use Installer to install Client Components)
- Using the following link: <http://auth.mapilab.com/Yammer/RequestToken>

store result in **Variable: variable** - Specify the variable to store the output.

16.7 Invite User to Yammer

Send Yammer invitation e-mail to this e-mail address, use this access code.

This workflow action initiates sending an invitation e-mail from Yammer network to specified e-mail address.

Parameters:

this e-mail address - Specify the recipient's e-mail address.

use **this access code** - specify the access code:

There are two ways to receive it:

- Using **Social Authorize Tool** from **Client Components** of HarePoint Workflow Extensions (use Installer to install Client Components)
- Using the following link: <http://auth.mapilab.com/Yammer/RequestToken>

16.8 Publish Message to Social Media via dlvr.it

Publish message with this text, this title and attach image at this url using this dlvr.it API key and this route.

This workflow action allows posting to Facebook, Twitter, Google+ and other social networks using dlvr.it.

Note: This action requires registration on dlvr.it.

Free subscription allows posting to 3 networks.

Parameters:

this text - Message text.

this title - Message title (optional).

attach image at this url - Path to an image to be attached to a post (optional).

this API key - User's secret key to access dlvr.it API.

API key can be checked here: <https://app.dlvr.it/settings/account>

this route - Route name for dlvr.it that will be used to post message.

The route is set up in dlvr.it (<https://app.dlvr.it/deliveries>) right after you log into it. Add there RSS, then add those social networks that you wish to repost to.

Specify the route name in this parameter.

16.9 Publish Message to Yammer

Publish this message to All Company Yammer group, attach file at this url, use this access code.

This workflow action posts a message to any group of Yammer corporate network; a file can be attached from the specified document library.

Parameters:

this message - a text of the message to be posted

to **All Company** Yammer group - select the Yammer group where this message should be posted

attach **file at this url** - specify the URL of the file from Document Library in SharePoint

use **this access code** - specify the access code:

There are two ways to receive it:

- Using **Social Authorize Tool** from **Client Components** of HarePoint Workflow Extensions (use Installer to install Client Components)
- Using the following link: <http://auth.mapilab.com/Yammer/RequestToken>

16.10 Replicate MailChimp Campaign

Replicate **this** MailChimp campaign. Use **this** api key. Store replicated campaign id in **Variable: variable**.

This workflow action is used to replicate (create a copy of) an existing **MailChimp** campaign.

Parameters:

this MailChimp campaign - specify either *ID* or *name* of the campaign to be replicated.

Note: If the name is specified and there are multiple campaigns with this name, the latest created campaign will be replicated.

this api key - the authorization key. See <http://kb.mailchimp.com/accounts/management/about-api-keys>

store campaign id in Variable:variable - specify the variable where an ID for the new campaign will be stored.

16.11 Schedule MailChimp Campaign

Schedule **this** MailChimp campaign for **this** time. Use **this** api key.

This workflow action is used to schedule the **MailChimp** campaign to be sent on a certain date at a certain time.

Note: A campaign can also be sent immediately, see "Send MailChimp Campaign" action.

Parameters:

this MailChimp campaign - specify either *ID* or *name* of the campaign to be scheduled.

Note: If the name is specified and there are multiple campaigns with this name, the latest created campaign will be scheduled.

this time - specify the time in GMT (Greenwich Mean Time) time zone, in 24h format as follows:

yyyy-mm-dd hh:mm:ss

Example:

2013-12-30 20:30:00

this api key - the authorization key. See <http://kb.mailchimp.com/accounts/management/about-api-keys>

16.12 Send MailChimp Campaign

Send this MailChimp campaign immediately. Use this api key.

This workflow action is used to initiate sending a MailChimp campaign immediately.

Note: The campaign sending can also be scheduled - see "Schedule MailChimp Campaign" action.

Parameters:

this MailChimp campaign - specify either *ID* or *name* of the campaign to be sent.

Note: If the name is specified and there are multiple campaigns with this name, the latest created campaign will be sent.

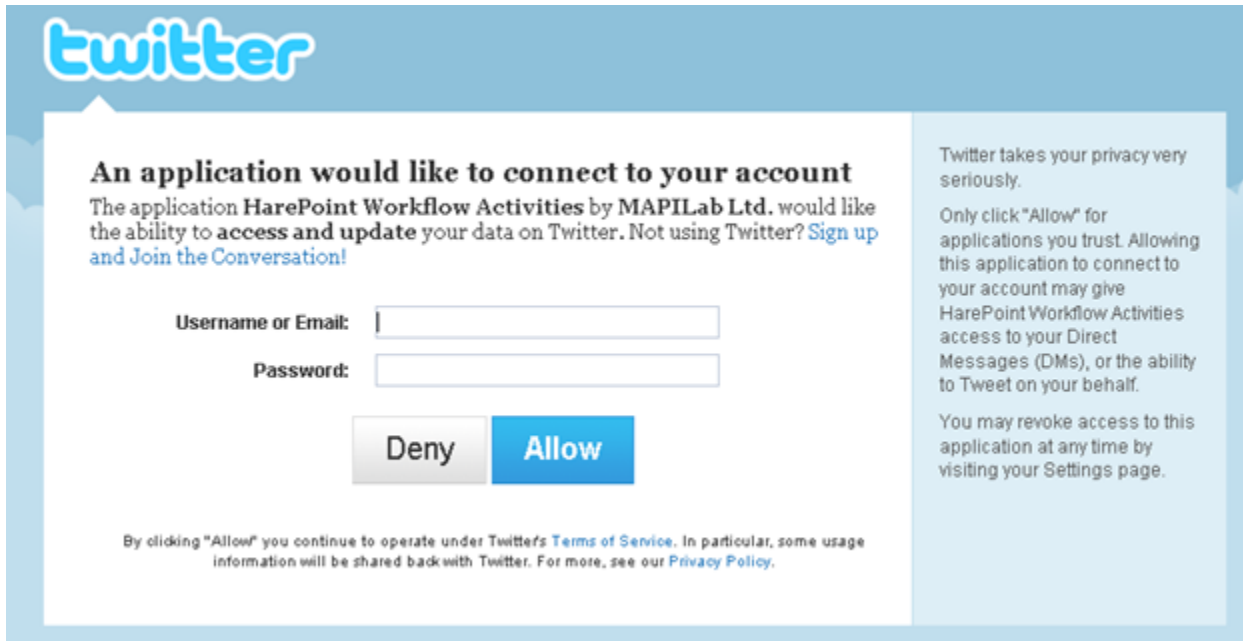
this api key - the authorization key. See <http://kb.mailchimp.com/accounts/management/about-api-keys>

16.13 Send Twitter Message

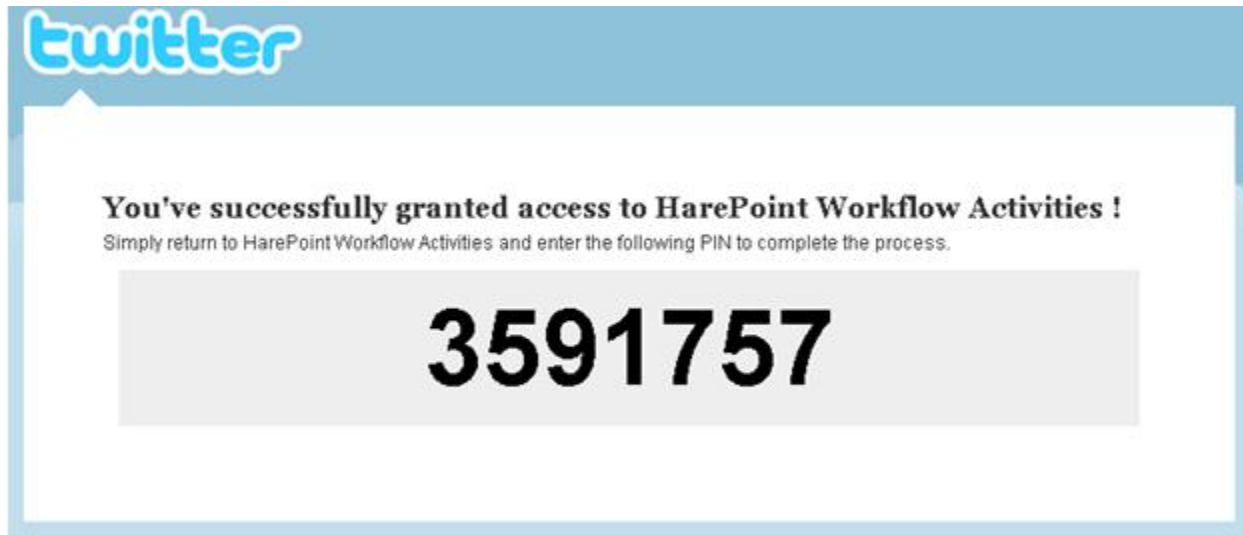
Send message with specified twitter authorization token.

This workflow action is used to send message to twitter.

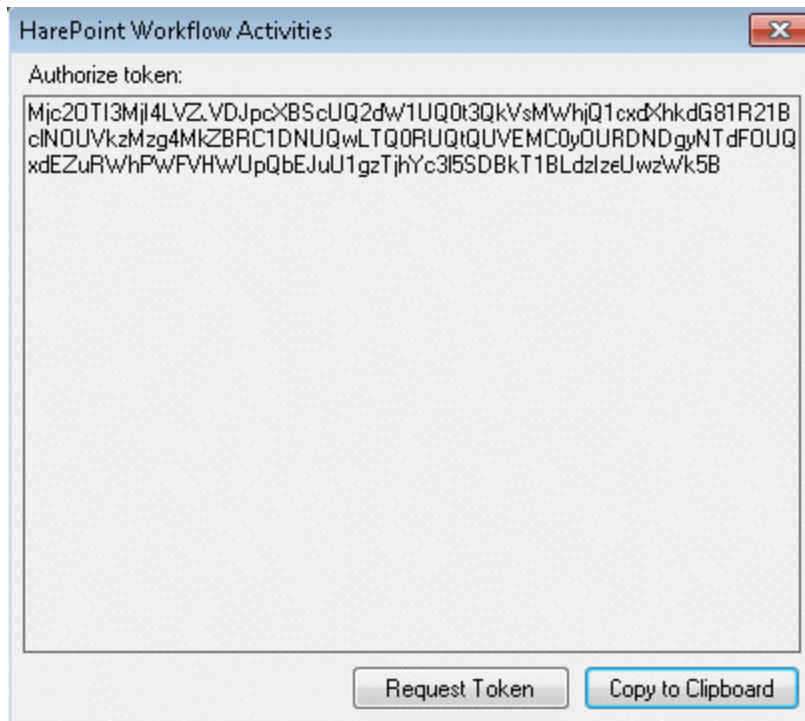
To set up this action, you must create authorization token with "Social Authorization Tool" placed in the program folder in **Start** menu. Run the tool, and click "Request Token" button next to Twitter icon. After that, Twitter web-page will be opened:



Enter your credentials and press Allow button on the web page to get the PIN number:



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".

16.14 Set User's Property in Yammer

Set specified property of Yammer user with this e-mail address to this value, use this access code.

This workflow action sets the specified property to the profile of the specified user in Yammer.

Parameters:

- specified property** - Select the property to set.
- user with this e-mail address** - Specify the user's e-mail address.
- this value** - Specify the new value for this property.
- use **this access code** - Specify the access code:

There are two ways to receive it:

- Using **Social Authorize Tool** from **Client Components** of HarePoint Workflow Extensions (use Installer to install Client Components)
- Using the following link: <http://auth.mapilab.com/Yammer/RequestToken>

16.15 Subscribe to MailChimp List

Subscribe this e-mail address to this MailChimp list. Use this api key.

This workflow action is used to add a new subscriber to a MailChimp list.

Parameters:

this e-mail address - specify the e-mail address of a new subscriber.

this MailChimp list - specify the name of the MailChimp list where the new subscriber needs to be added.

this api key - the authorization key. See <http://kb.mailchimp.com/accounts/management/about-api-keys>

16.16 Unsubscribe from MailChimp List

Unsubscribe this e-mail address from this MailChimp list. Use this api key, delete completely: no, send goodbye: yes, notify: yes .

This workflow action is used to unsubscribe people from your MailChimp list.

Parameters:

this e-mail address - Specify e-mail address of a person that needs to be unsubscribed.

this MailChimp list - Specify the name of the MailChimp list from which the subscriber should be removed.

this api key - The authorization key.

See <http://kb.mailchimp.com/accounts/management/about-api-keys>

delete completely: no/yes - Select if the subscriber should be removed completely from the list, or only marked as unsubscribed.

send goodbye: yes/no - Specify if an e-mail should be sent to a person, informing that he or she is now unsubscribed.

notify: yes/no - Specify if an e-mail should be sent to the administrator of the MailChimp list informing that a user has been unsubscribed.

16.17 Update Group in Yammer

Update this group in Yammer, set this title, this description, make it public, use this access code.

This workflow action modifies the parameters of the existing group in Yammer.

Parameters:

this group - Specify the group to be updated.

this title - Specify the new title for the group.

this description - Specify the new description for the group.

make it public - Choose to make this group *public* or *private*.

use **this access code** - Specify the access code:

There are two ways to receive it:

- Using **Social Authorize Tool** from **Client Components** of HarePoint Workflow Extensions (use Installer to install Client Components)
- Using the following link:<http://auth.mapilab.com/Yammer/RequestToken>

17. Strings activities

17.1 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\1\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

17.2 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\1\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>

17.3 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.

17.4 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*.

17.5 Convert Text Case

Convert this text to uppercase and store in Variable: variable.

Converts specified text to UPPER CASE, lower case, Title Case or Sentence case.

Parameters:

this text - Text, expression, item field or variable to convert.

uppercase - One of the following conversion types: *uppercase, lowercase, title case, sentence case*.

Variable: variable - Output string variable.

17.6 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.

17.7 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.

17.8 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.

17.9 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 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.

17.10 Evaluate Mathematical Expression

Evaluate this mathematical expression and store result in variable.

This workflow action is used to evaluate the specified mathematical expression. In particular, this action can be used to add up values from multiple list columns.

Parameters:

this - Specify the mathematical expression.

Supported operators: + - * /

Note: there are more supported operators and functions, for more details, please refer to <https://ncalc.codeplex.com/>

variable - Specify the variable to keep the result.

Examples:

Evaluate *[%Current Item: File Size%] + 100* mathematical expression and store result in Variable: test integer

17.11 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 ''.

17.12 Format Field of Current Item

Format this field with this format and store in Variable: FormattedText.

This workflow action is used to convert the current item field to string with special formatting.

Parameters:

this field - Select the field of a Current Item.

Important note: This action only supports the following field types:

- **DateTime**
- **Guid**
- **Number**

Text fields are not supported.

this format - Format string.

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>

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

Note: In this example, *Created* field needs to be **DateTime** type.

The result in workflow history list:

Item was modified by: 1;#GORLACH7\gorlach at 2011-04-30

17.13 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.

17.14 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.

17.15 Get Text Length

Get the number of characters in this text and store in Variable: text length.

This workflow action is used to calculate the length of a text string.

Parameters:

this text - Input a text string.

Variable: text length - Specify a variable to store the text length.

17.16 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 is 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.

17.17 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 0 - 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

17.18 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.

17.19 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>

17.20 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:

specified text - Text, expression, item field or variable.

specified delimiters - String or multi-line string with delimiters.

strings/characters - Type of delimiters. You can choose string delimiters (one per line) or character delimiters (each character in specified string is delimiter).

Variable: StringsArray - Output strings array.

17.21 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:

leading and trailing - One from three options: *leading*, *trailing*, *leading and trailing*.

this text - Text, expression, item field or variable.

Variable: TrimmedText - Output variable.

18. Taxonomy activities

18.1 Add Language to Term Store

Add English language to Default term store.

This workflow action is used to add a language to a specified term store.

Parameters:

English language - Select the language to be added from the drop-down list.

Default term store - (optional) Type the name of the term store where the new language needs to be added.

18.2 Create Taxonomy Group

Create taxonomy group with this name in Default term store. Store guid to Variable: groupGuid.

This workflow action is used to create a new taxonomy group in a specified term store.

Parameters:

this name - type the name for the new taxonomy group.

Default term store - type the name of the term store where the new group needs to be created

Variable: groupGuid - select a variable where the GUID of the new group will be stored.

18.3 Create Taxonomy Term

Create taxonomy term with this name in Default term store, this group and this term set. Use English language. Store guid to Variable: termGuid.

This workflow action is used to create a new taxonomy term in a specified term store and a specified term set.

Parameters:

this name - Type the name for the new taxonomy term.
Slash symbols can be used to indicate the new taxonomy term hierarchy, for example: <i>term1\subterm\new term name</i>
Default term store - (optional) Type the term store name.
this group - Type the name of the group.
this term set - Type the name of the term set.
English language - select the language for the new taxonomy term. If necessary, more languages can be added to term store using <i>Add Language to Term Store</i> .
Variable: termGuid - select the variable to store the GUID of a new term.

18.4 Create Taxonomy Term Set

Create taxonomy term set with this name in Default term store and group with this name. Use English language. Store guid to Variable: termSetGuid.

This workflow action creates a new taxonomy term set in a specified term store.

Parameters:

this name - type the name for the new taxonomy term.
Default term store - (optional) type the term store name.
this group - type the name of the group.
English language - select the language for the new taxonomy term. If necessary, more languages can be added to the term store using <i>Add Language to Term Store</i> .
Variable: termSetGuid - select a variable to store the GUID of a new term set.

18.5 Delete Taxonomy Group

Delete taxonomy group with this name in Default term store.

This workflow action deletes a specified taxonomy group.

Parameters:

this name - type the name of the taxonomy group to be deleted.

Default term store - (optional) type the name of the term store where a new group needs to be deleted.

18.6 Delete Taxonomy Term

Delete taxonomy term with this name in Default term store, this group and this term set. Use English language.

This workflow action is used to delete a specified taxonomy term from a specified term set, term group and term store, for a specified language.

Parameters:

this name - type the name of the term to be deleted.

Slash symbols can be used to indicate the new taxonomy term hierarchy, for example:

term1\subterm\new term name

Default term store - (optional) type the name of the term store.

this group - type the name of the group

this term set - type the name of a term set.

English language - select the language for the taxonomy term.

18.7 Delete Taxonomy Term Set

Delete taxonomy term set with this name in Default term store and group with this name. Use English language.

This workflow action is used to delete a specified term set from a specified taxonomy term store and group.

Parameters:

this name - enter the name of the term set to be deleted.

Default term store - (optional) the name of the term set.

this name - the name of the group.

English language - select the language.

18.8 Update Taxonomy Group

Update taxonomy group with **this** name in **Default** term store. Set **this** as new name and **this** as description.

This workflow action is used to update the name and the description of a taxonomy group.

Parameters:

this name - Enter the name of the taxonomy group to be updated.

Default term store - (optional) The name of the term store.

this as new name - (optional) The new name for the taxonomy group.

this as description - (optional) The new description for the taxonomy group.

18.9 Update Taxonomy Term

Update taxonomy term with **this** name in **Default** term store, **this** group and **this** term set. Set **this** as new term name, **this** as description, **this** as default label and **Yes** term is available for tagging. Use **English** language.

This workflow action is used to update the parameters of a taxonomy term.

Parameters:

this name - Enter the name of a term to be updated.

Default term store - (optional) The name of the term store where the term is located.

this group - The name of the group where the term is located

this term set - Enter the name of the term set where this term belongs.

this as new term name - (optional) Enter the new name for this term.

this as description - (optional) Enter the new description for this term.

this as a default label - The new default label for this term.

Yes - Is the term is available for tagging, *yes* or *no*.

18.10 Update Taxonomy Term Set

Update taxonomy term set with this name in Default term store and group with this name. Set this as new name, this as description, this as contact and Yes term set is open for term creation. Use English language.

This workflow action is used to update the parameters of a taxonomy term set.

Parameters:

this name - Enter the name of the term set to be updated.

Default term store - (optional) The name of the term store.

group with this name - The name of the group.

this as new name - (optional) The new name for this term set.

this as description - (optional) Enter the new description for this term set.

this as contact - (optional) the new contact for this term set.

Yes - Specify whether the term is open for new term creation, *yes* or *no*.

English language - Select the language for the new taxonomy term.

19. Workflow activities

19.1 Get SharePoint 2013 Workflow Instance Id

Get this SharePoint 2013 workflow instance on list item at this url and store id in Variable: variable.

This workflow action is used to obtain the Instance ID of a 2013 platform type workflow.

List workflows, reusable workflows and site workflows are supported.

Note: to get instance ID of 2010 platform type workflow, use "Get Workflow Instance Id" action.

Parameters:

this SharePoint 2013 workflow - Type in the name of the workflow whose instance ID you need to obtain.

list item - Specify the type of a workflow:

- *list item* - for list workflows and reusable workflows
- *current site* - for site workflows

at this url - (only displayed if "list item" is selected in previous parameter) Specify the full URL of a list item, for which the list workflow was running or was completed.

To get a proper list item URL, you can use the following:

- **Current Item: Encoded Absolute URL** lookup
- **Find list item at URL** action
- **Get List Item Url with Native form** in the first parameter

Variable: variable - Specify a variable to store the Instance ID of a workflow.

19.2 Get SharePoint 2013 Workflow Status

Get status of workflow with this name started on item at this url, store status in Variable: variable1 and last status change time in Variable: variable2.

This workflow action is used to get status of a 2013 Platform Type workflow.

Important note: site level feature "**Workflows can use app permissions**" needs to be activated on a corresponding site for this action to run properly.

Note: To get a status of a 2010 Platform Type workflow, use **Get Workflow Status** action instead.

Parameters:

with this name - Type in the name of 2013 Platform Type workflow which status you need to get.

item at this url - Specify the Encoded Absolute URL of an item that this **list workflow** is associated with.

Note: We recommend using **Find List Item at URL** or **Find List Item by CAML** actions to obtain a proper list item URL.

Leave this parameter empty for **site workflows**.

store status in **Variable:variable1** - Specify a variable to store the workflow status.

The following values are possible:

- *NotStarted* - The workflow instance has not started.
- *Started* - The workflow instance has started and is running.
- *Suspended* - Execution of the workflow has been stopped, but may be resumed.
- *Canceling* - The workflow instance has received a cancel command and is responding to the request.
- *Canceled* - Execution of the specified workflow instance is canceled but executes the workflow's cancelation scope.
- *Terminated* - Terminates the running workflow instance and raises the Completed event in the host. Once the workflow is terminated, it cannot be resumed.
- *Completed* - The workflow instance has finished running.
- *NotSpecified* - No status has been specified.
- *Invalid* - The workflow instance is in an invalid state.

last status change time in **Variable: variable2** - Specify a variable to store a last status change time.

19.3 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.

19.4 Get Workflow Instance Id

Get this workflow instance running on list item at this url and store id in Variable: variable.

This workflow action is used to obtain the Instance ID of a running or stopped 2010 platform type workflow.

List workflows, reusable workflows and site workflows are supported.

Note: to get instance ID of 2013 platform type workflow, use "Get SharePoint 2013 Workflow Instance Id" action.

Parameters:

this workflow - Type in the name of the workflow whose instance ID you need to obtain.

Note: Only 2010 platform type workflows are supported. For 2013 platform type workflows, use "Get SharePoint 2013 Workflow Instance Id" action.

running - Specify the status of the workflow, whose instance ID you need to obtain. Two options available:

- *running*
- *completed*

list item - Specify the type of a workflow:

- *list item* - for list workflows and reusable workflows
- *current site* - for site workflows

at this url - (only displayed if list item is selected in previous parameter) Specify the full URL of a list item, for which the list workflow was running or was completed.

To get a proper list item URL, you can use the following:

- **Current Item: Encoded Absolute URL** lookup
- **Find list item at URL** action
- **Get List Item Url with Native form** in the first parameter

Variable: variable - Specify a variable to store the Instance ID of a workflow.

19.5 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 a site or a list workflow status and the time of last status change.

Known workflow statuses 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.

Note: this action only supports 2010 Platform Type workflows. There is a similar action for 2013 Platform Type workflows: **Get SharePoint 2013 Workflow Status**

Parameters:

with this workflow name - Workflow name, e.g. 'My workflow'.

item at this url - (optional) only for list workflows specify the list item URL that this workflow runs on.

Variable: variable1 - Output string variable for status code, e.g. 'Completed'.

Variable: variable2 - Output Date/Time variable for last status change time.

19.6 Get Workflow Template ID

Get template id of this workflow and store in Variable: variable.

This workflow action is used to obtain the template ID of a workflow by its name (as seen in SharePoint Designer).

Parameters:

this workflow - Type the workflow name, as seen in SharePoint Designer.

Variable: variable - Specify a variable to store the template ID.

19.7 Start SharePoint 2013 Workflow

Start **this workflow** on the item with **this url** with **default payload** and store instance ID in **Variable: variable**.

This workflow action is used to start a 2013 Platform Type workflow.

Important note: site level feature "**Workflows can use app permissions**" needs to be activated on a corresponding site for this action to run properly.

Note: To start a 2010 Platform Type workflow, use **Start Workflow** action instead.

Parameters:

this workflow - Type in the name of 2013 platform workflow that needs to be started.

Note: In case of a List Workflow, be sure to specify the workflow that is associated with the list item from the next parameter.

the item with this url - Specify the Encoded Absolute URL of a list item for which the **list workflow** should be running.

Note: We recommend using **Find List Item at URL** or **Find List Item by CAML** actions to obtain a proper list item URL.

The parameter should be left empty if starting a **site workflow**.

default payload - (optional) Specify initiation parameters in the following format:

parameter1=value1

parameter1=value2

...

Variable: variable - Specify a variable that will store workflow instance ID.

This instance ID can be used in other workflow-related actions in HarePoint Workflow Extensions.

19.8 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 a list or a site workflow. Due to the nature of Microsoft SharePoint Workflow Timer starting a new workflow can take from several seconds to several minutes, during which the workflow will have the status "Starting".

Note: this action only supports 2010 Platform Type workflows. There is a similar action for 2013 Platform Type workflows: **Start SharePoint 2013 Workflow**.

See Also: "Get Workflow Association Data" activity.

Note: This action can be used to start a workflow on **multiple** list items. See *Example 2* for more details.

Parameters:

this workflow - Type in the name of the 2010 platform workflow that needs to be started.

Note: In case of a List Workflow, be sure to specify the workflow that is associated with the list item from the next parameter.

the item with this url - (for List Workflows) specify the Encoded Absolute URL of an item.

Arrays of list items URLs are supported, in this case the workflow will start on all list items from the array.

default - Workflow association data in XML format.

Variable: variable - Output string variable for workflow instance identifier.

In case if the action starts multiple workflows, this variable will contain multiple Instance IDs delimited by a semicolon (;).

Example: 1044a581-f7c0-404f-89d5-ae849e39910f;11842b5d-a9a6-4a4b-ac79-d070aaacb96b

Examples:

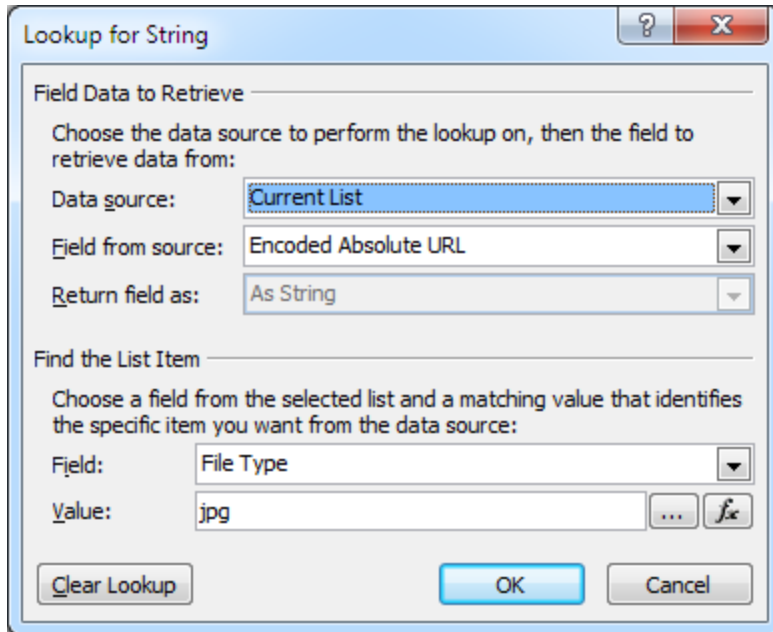
Example 1: Converting all JPG images in document library to PNG

Convert image at Current Item: Encoded Absolute URL to PNG format and store url to converted image in Variable: ConvertedImageUrl. Overwrite existing 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 converts all JPG images in document library to PNG. Use the following lookup settings for the second parameter of 'Start Workflow' action:



Example 2: Starting a workflow on multiple list items

Find item in the list at Current Item: Encoded Absolute URL where item's Title begins with abc . Store top 100 result urls in Variable: array .

then Start compute discount on Variable: array with default association data and store instance ID in Variable: variable.

19.9 Terminate SharePoint 2013 Workflow

Terminate workflow with this instance ID.

This workflow action is used to terminate a 2013 Platform Type workflow instance by its instance ID.

Important note: site level feature "**Workflows can use app permissions**" needs to be activated on a corresponding site for this action to run properly.

Note: To terminate a 2010 Platform Type workflow instance use **Terminate Workflow** action instead.

Parameters:

with this instance ID - Specify the instance ID of a workflow to be terminated.

In particular, instance ID can be obtained from **Get SharePoint 2013 Workflow Instance Id** or **Start SharePoint 2013 Workflow** actions.

19.10 Terminate Workflow

Terminate workflow with this instance ID.

This workflow action is used to terminate running workflows started by '**Start Workflow**' action.

Note: this action only supports 2010 Platform Type workflows. There is a similar action for 2013 Platform Type workflows: **Terminate SharePoint 2013 Workflow**.

Parameters:

with this instance ID - Specify the Instance ID of a running workflow.

It can be obtained from '**Start Workflow**' action, or from '**Get Workflow Instance Id**' action.

19.11 Wait SharePoint 2013 Workflow to Complete

Wait until workflow with this instance ID is completed and store status of the workflow in Variable: variable1.

This workflow action pauses the workflow till the specified instance of a 2013 Platform Type workflow enters the **Completed** status.

Important note: site level feature "**Workflows can use app permissions**" needs to be activated on a corresponding site for this action to run properly.

Note: To wait for a 2010 Platform Type workflow instance to complete, use **Wait Workflow to Complete** action instead.

Parameters:

with this instance ID - Specify the instance ID of a workflow to wait for.

The instance ID, in particular, can be obtained from **Start SharePoint 2013 Workflow** action.

Alternatively, you can use **Wait SharePoint 2013 Workflow to Complete by Name** action.

Variable: variable1 - Specify a variable to store the status of a workflow.

19.12 Wait SharePoint 2013 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 pauses the workflow till the specified instance of a 2013 Platform Type workflow enters the **Completed** status.

Important note: site level feature "**Workflows can use app permissions**" needs to be activated on a corresponding site for this action to run properly.

Note: To wait for a 2010 Platform Type workflow instance to complete, use **Wait Workflow to Complete by Name** action instead.

Parameters:

with this name - type in the name of a workflow to wait for.

item at this url - if this is a list workflow, specify the URL of a list item for which that workflow is running.

store status of the workflow in Variable: variable1 - specify a variable to store the status of a workflow

last status change time in Variable: variable2 - specify a variable to store the last status change time.

19.13 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.

Note: this action only supports 2010 Platform Type workflows. There is a similar action for 2013 Platform Type workflows: **Wait SharePoint 2013 Workflow to Complete**

Known workflow statuses 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 a running workflow obtained from 'Start Workflow' action.
Variable: variable - Output string variable for workflow status.

19.14 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.

Note: this action only supports 2010 Platform Type workflows. There is a similar action for 2013 Platform Type workflows: **Wait SharePoint 2013 Workflow to Complete by Name**

Known workflow statuses 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 - only 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. Evaluate Due Date
15. Get Date Property
16. Get Interval Between Dates
17. Get Working Time Between Dates
18. Set DateTime Constant
19. Capture All Occurrences of Text
20. Capture Text
21. Convert HTML to Text
22. Convert Number to Text
23. Convert Text Case
24. Converts Base64 to Text
25. Converts Text to Base64
26. Decode Web Text

27. Encode Web Text
28. Evaluate Mathematical Expression
29. Extract Substring
30. Format Field of Current Item
31. Get Random Number
32. Get Random String
33. Get Text Length
34. Is Value Matches Regular Expression (HarePoint)
35. Join Text
36. Replace Text
37. Set Regular Expression Options
38. Split Text
39. Trim Text