



NWAdmin Operations: Nintex Workflow 2013

Last updated Sunday, October 23, 2016

Legal notices

For the latest information, please see <http://en-us.nintex.com/company/legal>.

Contents

NWAdmin Guide	1
Introduction	1
Operations	1
AddAction	3
AddCustomDataItem	4
AddEventReceiver	5
AddInlineFunction	6
AddZoneSetting	8
AttachDatabase	10
CheckDatabaseVersion	11
CleanTaskRedirects	12
ConfigureSettings	13
ConvertToClaims	14
DelegateAllTasks	15
DeployWorkflow	16
DetachDatabase	17
EnumCustomDataItems	18
EnumEventReceivers	19
EnumInlineFunctions	20
EnumSafeActions	21
EnumSchedules	22
EnumZoneSettings	23
ExportWorkflow	24
ExportWorkflowConstants	25
FindWorkflows	26
FixSiteAfterImport	27
GenerateServerNameHash	28
ImportWorkflowConstants	29
InstallTimerJob	30
MessageTemplateUsage	31
MigrateUser	32
MoveData	33
PrepareSiteForExport	35
PurgeHistoryListData	36

PurgeWorkflowData	37
RemoveAction	39
RemoveCustomDataItems	40
RemoveEventReceiver	41
RemoveInlineFunction	42
RemoveSafeActions	43
RemoveZoneSetting	44
ResolveSharePointIds	45
SetCallWebServiceTimeout	46
SetTaskForm	47
SyncTerminatedWorkflows	48
UninstallTimerJob	49
UpdateContentDatabase	50
UpdateContentDatabaseParameters	51
UpgradeDatabases	52

NWAdmin Guide

This document describes the operations available with the NWAdmin.exe tool.

Introduction

NWAdmin is a command line tool that ships with Nintex Workflow. It is used to perform various administration operations.

The NWAdmin.exe tool is automatically deployed into the SharePoint '15' hive and can be accessed from the SharePoint 2013 Management Shell.

Run NWAdmin operations on servers that are running the Web Application Service; that is, run the operations on web application servers.

This document will always reflect the copy of NWAdmin which is released with the most recent build of Nintex Workflow 2013. If you find a documented operation is missing, contact support@nintex.com for information on updating your build of Nintex Workflow.

This document is current for Nintex Workflow 2013 Version 3.0.0.0.

Operations

This section lists operations available with NWAdmin.

NWAdmin Guide	1
Introduction	1
Operations	1
AddAction	3
AddCustomDataItem	4
AddEventReceiver	5
AddInlineFunction	6
AddZoneSetting	8
AttachDatabase	10
CheckDatabaseVersion	11
CleanTaskRedirects	12
ConfigureSettings	13
ConvertToClaims	14
DelegateAllTasks	15
DeployWorkflow	16
DetachDatabase	17
EnumCustomDataItems	18
EnumEventReceivers	19
EnumInlineFunctions	20
EnumSafeActions	21
EnumSchedules	22
EnumZoneSettings	23
ExportWorkflow	24
ExportWorkflowConstants	25
FindWorkflows	26

FixSiteAfterImport	27
GenerateServerNameHash	28
ImportWorkflowConstants	29
InstallTimerJob	30
MessageTemplateUsage	31
MigrateUser	32
MoveData	33
PrepareSiteForExport	35
PurgeHistoryListData	36
PurgeWorkflowData	37
RemoveAction	39
RemoveCustomDataItems	40
RemoveEventReceiver	41
RemoveInlineFunction	42
RemoveSafeActions	43
RemoveZoneSetting	44
ResolveSharePointIds	45
SetCallWebServiceTimeout	46
SetTaskForm	47
SyncTerminatedWorkflows	48
UninstallTimerJob	49
UpdateContentDatabase	50
UpdateContentDatabaseParameters	51
UpgradeDatabases	52

AddAction

Description

This operation is used to add a custom workflow action into Nintex Workflow. The workflow action must be defined in a XML file with an extension of `nwa`. The action will be registered in each content web application `web.config` file as an authorized workflow type.

Usage

NWAdmin.exe -o AddAction -nwaFile pathToNWAFile [-updateIfAdapterExists] [-activateForFarm] [-verifyTypes]

Parameters

Name	Description
-nwaFile	Required. The location on the computer of the .nwa file containing the action details to import.
-updateIfAdapterExists	Optional. If the adapter is already registered, specifying this argument causes the values for this adapter to be overwritten with the values in the nwa file. If updateIfAdapterExists is not specified and the adapter does already exist, an error will be returned.
-activateForFarm	Optional. Specifies that the action should be activated on the `manage allowed actions` page for the entire farm when this parameter is provided.
-verifyTypes	Optional. When specified, the classes referenced in the nwa are loaded, causing an error if they cannot be found on the server.

AddCustomDataItem

Description

This operation is used to register a custom workflow context property, so that it can be used in the Workflow designer. Custom workflow context properties are displayed in the Common tab of the Insert Reference dialog. For more information about creating custom workflow context properties, see the Nintex Workflow 2013 SDK.

Usage

```
NWAdmin.exe -o AddCustomDataItem -type typename -assembly assemblyname [-displayname defaultdisplayname] [-description defaultdescription] [-data data]
```

Parameters

Name	Description
-type	Required. The fully qualified type name, including namespace, of the class for the custom workflow context property. The maximum length of this parameter is 255 characters.
-assembly	Required. The fully-qualified four-part name of the assembly that contains the type for the custom workflow context property. The maximum length of this parameter is 255 characters.
-displayname	Optional. The default display name for the custom workflow context property. If this value is not specified, the value of the DisplayName property from the class for the custom workflow context property is used. The maximum length of this parameter is 255 characters.
-description	Optional. The default description for the custom workflow context property. The maximum length of this parameter is 1024 characters.
-data	Optional. The string data, if applicable, specified for the custom workflow context property.

AddEventReceiver

Description

Adds an event receiver listening to a Nintex Workflowevent.

Usage

Intended for use when instructed by Nintex customer service.

AddInlineFunction

Description

This operation is used to register a new function alias for an inline function, so that it can be used in the Workflow designer.

Usage

NWAdmin.exe -o AddInlineFunction -functionalias functionAlias -assembly AssemblyName -namespace Namespace -typename TypeName -method MethodName [-usage Usage] [-description Description] [-lcid Lcid] [-hidden]

Parameters

Name	Description
-functionalias	<p>Required. A string that represents the name of the function alias for the inline function. The name of the function alias is used to invoke the inline function in workflow actions. The maximum length for this parameter is 512 characters.</p> <p>Note: Nintex suggests that function aliases for inline functions use 15 characters or less, if possible.</p>
-assembly	<p>Required. A string that represents the name of the assembly that contains the type and method for the inline function. If the type and method is not contained in the .NET Framework core library (mscorlib), use the fully-qualified four-part name of the assembly; otherwise, specify mscorlib.</p> <p>The maximum length of this parameter is 512 characters.</p>
-namespace	<p>Required. A string that represents the fully-qualified name of the namespace, in the assembly specified in <i>assembly</i>, that contains the type and method for the inline function.</p> <p>The maximum length for this parameter is 512 characters.</p>
-typename	<p>Required. A string that represents the unqualified type name of the static class that contains the method for the inline function.</p> <p>The maximum length for this parameter is 512 characters.</p>
-method	<p>Required. A string that represents the unqualified member name of the public static method, in the type specified in <i>type</i>, for the inline function.</p> <p>The maximum length for this parameter is 512 characters.</p> <p>Note: Do not specify additional syntax elements, such as parameters. Nintex Workflow 2013 uses reflection to identify additional syntax elements for the member specified in this parameter.</p>
-usage	<p>Optional. A string that represents the description of the syntax information for the function alias. The maximum length for this parameter is 500 characters.</p>
-description	<p>Optional. A string that represents the description of the function alias. The maximum length for this parameter is 500 characters.</p>
-lcid	<p>Optional. An integer that represents the locale identifier of the function alias. The default value is null.</p> <p>Note: Duplicate function aliases are not allowed, even if the locale identifiers are different.</p>

Name	Description
-hidden	Optional. Indicates whether the function alias is displayed in Workflow designer. If specified, the function alias is not displayed in Workflow designer; otherwise, the function alias is displayed.

Note: This parameter does not affect whether the function alias is displayed in the output of the **EnumInlineFunctions** operation. For more information, see "[EnumInlineFunctions](#)" on page 20.

Remarks

You can add multiple aliases for a single inline function by running this operation for each new function alias, specifying a different value for the *functionalias* parameter for each one. Duplicate alias values are not allowed, however, even if different locale identifiers are specified.

For more information about inline functions, including how to develop, define, and invoke custom inline functions, see the Nintex Workflow SDK. To access Nintex SDKs, go to help.nintex.com.

AddZoneSetting

Description

This operation is used to add a record to define which alternative access mapping zone URL workflows in the specified scope should use when generating urls (e.g. for the approval url used in approval notifications). By default, workflows will use the 'default' zone url regardless of the zone they were started in. This command can overwrite this behaviour.

When a workflow runs, it will query for all the zones settings where the scope includes the workflow. The zone setting with the highest precedence will be used. The order of precedence is as follows, with the top most taking highest priority:

1. ExplicitWorkflow
2. ExplicitList
3. AnyWorkflow
4. AnyList
5. TeamSite
6. SiteCollection
7. WebApplication
8. Global

If no zone rule is matched (and no Global rule was added) the 'Default' zone url will be used.

For example, if a workflow called "Finance Approval" is running in a site collection at <http://sharepoint/sites/internalservices> and an individual zone setting has been added for this site collection and this workflow, the workflow rule will be used to generate urls.

AnyWorkflow and AnyList will match any workflow or list in the farm with the same name, while ExplicitWorkflow and ExplicitList will only match a specific list or workflow.

Usage

NWAdmin.exe -o AddZoneSetting -type Global | WebApplication | SiteCollection | TeamSite | AnyList | AnyWorkflow | ExplicitList | ExplicitWorkflow [-url url] [-list listName] [-workflow workflowName] -zone Default | Intranet | Internet | Custom | Extranet

Parameters

Name	Description
-type	The scope of the entry for this url zone entry.
-url	The URL to define the scope of the type entry. Required for types: WebApplication SiteCollection TeamSite ExplicitList ExplicitWorkflow
-list	The name of the list to define the scope of the type entry. Required for types: AnyList ExplicitList ExplicitWorkflow

Name	Description
- workflow	The name of the workflow to define the scope of the type entry. Required for types: AnyWorkflow ExplicitWorkflow
-zone	The URL zone to use when a workflow matches this entry. See the Alternate Access Mappings screen in SharePointCentral Administration for more information.

AttachDatabase

Description

This operation attaches an existing content database that was detached with the DetachDatabase command into the current environment. The attach process performs the following steps:

- Upgrades the database to the version required in the target environment.
- Maps any site collections stored in the attached database to use the correct database.
- Assigns web application permissions to the database.
- Recovers any pending Nintex Live service requests stored from the previous environment.
- Adds Nintex Live service version information from the previous environment.

This command should be run after the timer service account has been granted owner rights to the content database.

The command should be run after the related SharePoint content database and web application have been attached to the target environment.

The command should be run by an account with read and write access to the Nintex Workflow databases.

The command first performs some initial validation and prints out any errors or warnings. It will only continue to attach the database if no errors or warnings occur. This behaviour can be modified by using the `-validateOnly` and `-skipValidation` switches.

It is recommended that the "SharePoint 2013Timer" service is stopped while running this command.

Usage

```
NWAdmin.exe -o AttachDatabase -serverName serverName -databaseName contentDatabase [-integrated | -username username -password password] [-validateOnly | -skipValidation]
```

Parameters

Name	Description
-serverName	The server name of the content database to attach.
-databaseName	The database name of the content database to attach.
-integrated	Optional, configure the content database connection to use integrated security. Recommended.
-username	Optional, configure the content database to use database authentication with this username. Use in conjunction with -password.
-password	Optional, configure the content database to use database authentication with this password. Use in conjunction with -username.
-validateOnly	Modifies the operation to only run the initial validation steps and not actually attach the database, even if all validation steps pass.
-skipValidation	Modifies the operation to not run the initial validation steps and attach the database. Use this to attach the database regardless of any errors or warnings.

CheckDatabaseVersion

Description

This operation is used to list the Nintex Workflow databases, their version, and if they require an upgrade.

Usage

NWAdmin.exe -o CheckDatabaseVersion

Parameters

This operation has no parameters.

CleanTaskRedirects

Description

This operation is used to check if a content database is still in use after a MoveData command has been used to remove data from it.

The command will prompt the user for which database to analyze.

When MoveData is used to move data out of a database, it may leave 'redirect' records to support LazyApproval in workflows that are currently running. If the database is disconnected while a running workflow needs these records, LazyApproval for these tasks will not work.

This command will allow an administrator to know if a database can be safely disconnected without affecting running workflows. It will check if a redirect record is still required, and if not, delete it. When there are no redirect records remaining, the database can be deleted.

Usage

```
NWAdmin.exe -o CleanTaskRedirects [-test]
```

Parameters

Name	Description
-test	Simulates running the command without deleting any redirect records.

ConfigureSettings

Description

This operation is used to automatically perform the configuration steps usually carried out in the configuration pages in Nintex Workflow. This can be used as part of an automated install process.

The values to set are defined in an XML file. The schema file "NintexWorkflowSetup.xsd" exists in the Nintex Workflow install directory and must be used to create the xml file.

Usage

NWAdmin.exe -o ConfigureSettings -file pathToXMLFile

Parameters

Name	Description
-file	The location of the xml file containing the configuration settings

Examples

The format of the ActionActivations:

```
<ActionActivations ActivateAllAcrossFarm="true">
<ActivationRecord AdapterAssembly="" AdapterType="" Scope="Farm" Url="" />
</ActionActivations>
```

The format of the RegisteredActions:

```
<RegisteredActions>
<AdditionalActions>
<NwaPath Filename="" UpdateIfExists="" />
</AdditionalActions>
<RemoveStandardActions>
<Remove AdapterType="" AdapterAssembly="" />
</RemoveStandardActions>
</RegisteredActions>
```

An example of part of the GlobalSettings:

```
<GlobalSettings>
<SmtpServer></SmtpServer>
<SmtpFromAddress></SmtpFromAddress>
<SmtpReplyToAddress></SmtpReplyToAddress>
<EnforceSafeLooping>true</EnforceSafeLooping>
<DefaultTaskListName>Workflow Tasks</DefaultTaskListName>
</GlobalSettings>
```

ConvertToClaims

Description

This operation is used to convert all Nintex Workflow user data for a web application from classic authentication to claims.

The operation will search all workflow databases for records associated with the specified web application, and update the user and group values to be the claims equivalent. The specified web application should already have been converted to claims from classic authentication.

Note: In normal circumstances, this command does not need to be executed. Conversion of the user data will happen automatically when the web application is converted from classic to claims.

Usage

NWAdmin.exe -o ConvertToClaims -webApplication url

Parameters

Name	Description
- webApplication	The URL of the claims web application to update Nintex Workflow data for.

DelegateAllTasks

Description

This operation is used to reassign all tasks from one user to another. It can be scoped to a single site, or the entire environment. Note that tasks can only be delegated to individual users, not groups.

Note: This operation uses the Nintex Workflow web service, and requires credentials to connect to the web service.

Usage

NWAdmin.exe -o DelegateAllTasks -currentUser domain\username -newUser domain\username [-siteUrl urlToSharePointSite] [-sendNotification] [-comments] [-username username] [-password password] [-domain domain]

Parameters

Name	Description
-currentUser	The domain\username of the user whose tasks will be reassigned.
-newUser	The domain\username of the user who the tasks will be assigned to.
-siteUrl	A URL to a SharePoint site containing tasks to delegate. If this parameter is not provided, tasks in the whole environment will be delegated.
-sendNotification	Provide this parameter to send a notification to the new assignee informing them that the task is now assigned to them.
-comments	Optionally add any comments to explain the delegation.
-username	A valid SharePoint user's username. If no siteUrl is specified, the user must be a farm administrator. If a siteUrl is specified, the user must be an administrator of the site. If no username is provided, the credentials of the user running the command will be used.
-password	A valid password for -username
-domain	A valid domain for -username

DeployWorkflow

Description

This operation is used to publish or save a workflow that is saved as an exported NWF file.

Note: This operation uses the Nintex Workflow web service, and requires credentials to connect to the web service.

Usage

NWAdmin.exe -o DeployWorkflow -workflowName workflowName -nwfFile pathToNwfFile -siteUrl targetSiteUrl [-targetList targetList] [-saveOnly] [-saveIfCannotPublish] [-overwrite] [-skipValidation] [-username username] [-password password] [-domain domain]

Parameters

Name	Description
-workflowName	The name to call the workflow in the target location.
-nwfFile	The path to the nwf file to deploy.
-siteUrl	The URL to the site to deploy the workflow to.
-targetList	The name of the list or library on the target site to deploy the workflow to. Do not use this parameter if it is a site workflow being deployed.
-saveOnly	Specify to save the workflow and not publish it. If not specified, the workflow will publish.
-saveIfCannotPublish	Specify to fall back to a save operation if the workflow cannot be published. If not specified, no changes will be made if the workflow cannot publish.
-overwrite	Specify to overwrite a workflow that already exists on the site.
-skipValidation	Advanced. Specify to skip the validation of a workflow. This option has no effect if -saveOnly is specified.
-username	A valid SharePoint user's username with permissions to publish a workflow. If no username is provided, the credentials of the user running the command will be used.
-password	A valid password for -username
-domain	A valid domain for -username

DetachDatabase

Description

This operation detaches a content database from the environment. The detach process performs the following steps:

- Prepares any pending Live requests so they will not be processed in the current environment.
- Stores Live service information that will be required in the target environment.
- Unregisters the database from the environment and removes any site collection mappings that use it.

Running this command will cause workflow history to be lost from the environment. The command is intended to be run as part of the process of migrating a SharePoint database to another environment or in other scenarios where the data stored in it will no longer be required by the environment.

The command should be run by an account with read and write access to the Nintex Workflow databases.

The command does not delete the database from the database server.

It is recommended that the "SharePoint 2013Timer" service is stopped while running this command.

The detached database can be reattached to an environment using the AttachDatabase command.

Usage

```
NWAdmin.exe -o DetachDatabase -serverName serverName [-databaseName contentDatabase]
```

Parameters

Name	Description
-serverName	The server name of the content database to detach.
-databaseName	Optional, the database name of the content database to detach. Used to differentiate two databases on the same server.

EnumCustomDataItems

Description

This operation is used to display a list of available custom workflow context properties. Custom workflow context properties are displayed in the Common tab of the Insert Reference dialog. For more information about creating custom workflow context properties, see the Nintex Workflow 2013 SDK.

Usage

```
NWAdmin.exe -o EnumCustomDataItems
```

Parameters

This operation has no parameters.

Remarks

When invoked, this operation lists the display name, type name, and assembly name of each custom workflow context property, delimited with space characters.

EnumEventReceivers

Description

Lists the event receivers listening to Nintex Workflow events.

Usage

Intended for use when instructed by Nintex customer service.

EnumInlineFunctions

Description

This operation is used to display all the function aliases registered for inline functions in Nintex Workflow.

Usage

NWAdmin.exe -o EnumInlineFunctions

Parameters

This operation has no parameters.

Remarks

This operation returns all function aliases, regardless of the current locale, defined in the Nintex configuration database. The output of this operation contains the following five columns, separated by tab characters:

Column name	Description
functionAlias	The name of the function alias for the inline function, formatted as a 15-character padded column.
MethodName	The name of the public static method for the inline function, formatted as a 15-character padded column.
AssemblyName	The name of the assembly, formatted as a 13-character padded column.
Namespace	The name of the namespace for the inline function, formatted as a 13-character padded column.
TypeName	The name of the type for the inline function.

EnumSafeActions

Description

This operation is used to display all the actions that have been registered in the web.config file by Nintex Workflow. This includes the entries required for the core operation of Nintex Workflow and any entries added for custom workflow actions.

These entries can be viewed manually in the web.config file for the web application. The entries are listed in the

<System.Workflow.ComponentModel.WorkflowCompiler><authorizedTypes> section of web.config.

Usage

NWAdmin.exe -o EnumSafeActions [-url urlToWebApplication] [-showFullXml]

Parameters

Name	Description
-url	The URL to the web application to display safe actions for. If this is not specified, entries for all web applications will be displayed
-showFullXml	When specified, the XML fragment that is displayed in web.config will be shown. If not specified, only the namespace and type names of each entry will be displayed without the surrounding XML.

EnumSchedules

Description

This operation is used to display all the workflow schedules created in the farm.

Usage

NWAdmin.exe -o EnumSchedules [-siteCollection SiteCollectionUrl]

Parameters

Name	Description
- siteCollection	Optional. The URL to the site collection to display safe actions for. If this is not specified, entries for all site collections will be displayed

EnumZoneSettings

Description

This operation is used to display all the registered zone URL configuration rules that have been added with the AddZoneSetting operation.

Usage

NWAdmin.exe -o EnumZoneSettings

Parameters

This operation has no parameters.

ExportWorkflow

Description

This operation is used to export a workflow to an NWF file.

Note: This operation uses the Nintex Workflow web service, and requires credentials to connect to the web service.

Usage

NWAdmin.exe -o ExportWorkflow -siteUrl targetSiteUrl -workflowName workflowName -filename fileToSave [-workflowType reusable|globallyreusable|site|list|workflowapproval|userdefinedaction -list listName] [-username username] [-password password] [-domain domain]

Parameters

Name	Description
-siteUrl	The url to the site that contains the workflow to export.
-list	The list that contains the workflow to export. Do not use this parameter if it is a site workflow being deployed.
-workflowName	The name of the workflow to export.
-filename	The path and filename to save the export file as. It should end with a .nwf extension.
-workflowtype	The type of the workflow to export. <div data-bbox="469 1003 1292 1111" style="background-color: #4F81BD; color: white; padding: 5px; border: 1px solid #4F81BD;"> Note: If the value of this parameter is set to list, the -list parameter is required. </div>
-list	The name of the list from which to export the workflow.
-username	A valid SharePoint user's username with permissions to publish a workflow. If no username is provided, the credentials of the user running the command will be used.
-password	A valid password for -username
-domain	A valid domain for -username

ExportWorkflowConstants

Description

This operation is used to export the workflow constants configured for an environment. The details of each workflow constant for the specified scopes will be saved to an XML file.

At least one of 'includeSite', 'includeSiteCollection' or 'includeFarm' must be specified.

Use in conjunction with the ImportWorkflowConstants operation to transfer constants across environments.

Usage

NWAdmin.exe -o ExportWorkflowConstants -siteUrl siteUrl -outputFile pathToFile [-includeSite] [-includeSiteCollection] [-includeFarm]

Parameters

Name	Description
-siteUrl	The url to the team site for which workflow constants will be exported.
-outputFile	The path and filename to store the workflow constant information in.
-includeSite	Optional. When specified, the workflow constants created at the site scope will be included in the export file.
-includeSiteCollection	Optional. When specified, the workflow constants created at the parent site collection scope will be included in the export file.
-includeFarm	Optional. When specified, the workflow constants created at the parent farm scope will be included in the export file.

FindWorkflows

Description

This operation is used to list all sites where Nintex Workflow is activated and all the workflows that have been published. Workflows are grouped by list.

Usage

NWAdmin.exe -o FindWorkflows [-activationsOnly]

Parameters

Name	Description
- activationsOnly	When specified, only the sites where Nintex Workflow is activated is listed. The workflows on each site are not displayed.

FixSiteAfterImport

Description

This operation is used to reconfigure the workflows in a site that was imported with the stsadm -o import command.

This command must be used in conjunction with the PrepareSiteForExport command.

Caution: This command should be run immediately after the site was imported. If it is not run, the process will run automatically when the workflow gallery or manage workflows page is accessed in the site. The process involves publishing all workflows on the site, which will take some time. It is therefore ideal to run this command so a delay is not noticed in the SharePoint site.

The command makes use of metadata in a file named SiteMetaData.xml stored in the hidden NintexWorkflows library on the site, generated with PrepareSiteForExport.

See also the PrepareSiteForExport command.

Usage

```
NWAdmin.exe -o FixSiteAfterImport -siteUrl urlToSiteThatWasImported [-metaDataFile pathToMetadataFile]
```

Parameters

Name	Description
-siteUrl	The URL to the site that has been imported with stsadm.exe.
-metaDataFile	Optional file path to an alternative SiteMetaData.xml file.

GenerateServerNameHash

Description

Creates a hash of a server name for licensing purposes when a server name must remain confidential.

Usage

Nintex internal use only.

ImportWorkflowConstants

Description

This operation is used to import the workflow constants from a file into an environment. At least one of 'includeSite', 'includeSiteCollection' or 'includeFarm' must be specified.

Use in conjunction with the ExportWorkflowConstants operation to transfer constants across environments

How to handle conflicts (where a constant of the same name already exists in the target scope) is determined with the -handleExisting parameter. The options are:

- Skip: The constant from the file is not imported, and the existing constant is unaffected
- Overwrite: The constant from the file is imported and used to update the details of the existing constant
- Abort: The import process will stop. Constants that have already been imported before the conflict was encountered will remain imported

A conflict is only encountered if the constant name and the scope is the same.

For example, if the import file contains a constant called "Test" from a **team site**, and the target environment contains a constant called "Test" in the **site collection**, they are treated as two separate constants and not recognized as a conflict. The result will be that the site collection and site both contain a constant called "Test".

Usage

```
NWAdmin.exe -o ImportWorkflowConstants -siteUrl siteUrl -inputFile pathToFile -handleExisting Skip|Overwrite|Abort [-includeSite] [-includeSiteCollection] [-includeFarm]
```

Parameters

Name	Description
-siteUrl	The URL to the team site to where the workflow constants will be imported.
-inputFile	The path and filename of a workflow constant file generated by the ExportWorkflowConstants command.
-handleExisting	One of Skip, Overwrite or Abort. Determines who to handle a constant of the same name already existing in the target.
-includeSite	Optional. When specified, the workflow constants in the file with a scope of site will be imported to the target team site.
-includeSiteCollection	Optional. When specified, the workflow constants in the file with a scope of site collection will be imported to the target parent site collection.
-includeFarm	Optional. When specified, the workflow constants in the file with a scope of farm will be imported to the target parent farm.

InstallTimerJob

Description

This operation is used to install a Nintex Workflow timer job into an environment.

Nintex Workflow uses two timer jobs.

- **ScheduledWorkflows**
This job is used to run workflows that have been set to run to a schedule. This job is installed in all environments.
- **TaskProcessing**
This job is to process some task responses.

Usage

NWAdmin.exe -o InstallTimerJob -job ScheduledWorkflows|TaskProcessing [-url webApplicationUrl] [-minutes minutesBetweenExecution]

Parameters

Name	Description
-job	Specifies which timer job to install.
-url	[Advanced] A URL to a web application. If not specified, the central administration web application is used.
-minutes	The number of minutes between each execution of the timer job. Default is 5 minutes.

MessageTemplateUsage

Description

This operation is used to list all the message templates that have been defined in an environment. It will list the site and site collections that have a custom message template defined. In multi-language environments, each language specific template will also be displayed.

Usage

NWAdmin.exe -o MessageTemplateUsage

Parameters

This operation has no parameters.

MigrateUser

Description

This operation is used to change all references to a username to a new username. Changes are only made to historic workflow data and user settings. Workflow definitions and running workflows are not changed.

Usage

NWAdmin.exe -o MigrateUser -oldUser originalUsername -newUser newUsername

Parameters

Name	Description
-oldUser	The 'before' domain\login of the user.
-newUser	The 'after' domain\login of the user.

MoveData

Description

This operation is used to move all workflow progress data for a site collection to a selected workflow content database.

This command should be run as the farm account, as this identity will have access to each workflow database.

The utility will begin by warning the user to pause all services that can run or interact with workflow. This is important; if the workflow data changes during the migration process it may become corrupted. Once the migration is complete, the tool will attempt to detect any changes to the data and handle this by warning the user and moving the changed data, but it is recommended to avoid this situation.

The tool will list the workflow content databases that are configured with numeric identifiers and prompt the user to choose which of these databases the site collection workflow data should be moved in to.

The data for each workflow instance is moved one by one. If any fails to move, the tool will display error details and prompts the user either to retry moving the instance, skip the instance or abort the process, rolling back any changes.

Once the operation is complete, restart all services to continue workflow operation. From this point on, workflows in the site collection will use the new database. If any services were not stopped during the migration process, they will contain cached references to use the old database so it is critical they are restarted at this point (which will cause them to clear the cached database pointer and use the new database).

Retaining the source content database

After the data is moved, the source content database must remain in place until the redirects added by the MoveData operation have been cleaned up. To clean up redirects, use [the CleanTaskDirects operation](#). After redirects are cleaned up and you are certain that no site collection is using the source content database, you can remove the source content database.

Use in migration scenarios and external databases

In some scenarios, it will be useful to move data to a Nintex Workflow database not connected to the SharePoint farm. For example if a site collection backup and restore is performed to move an entire site collection to a different farm, it is useful to be able to push Nintex Workflow data directly into the target farm. Likewise, there will be scenarios where it is useful to pull Nintex Workflow data from another farm into the current farm.

In this scenario, the details of an external database can be provided via the command line options. The user running the command must be in the WSS_Content_Application_Pools group on the external database to have permissions to read or write the data.

A limitation with external databases is workflows in progress at the time of the MoveData command cannot accept a LazyApproval response. Workflow tasks can still be completed via the SharePoint UI. New task actions that assign their tasks after the move data command can still use LazyApproval.

Note: It is recommended the workflow job in Central Administration is stopped in the target environment when a site collection migration with workflow is performed. If a workflow wakes up from a delay before the data has been imported, that workflow will fail. Stopping the workflow timer job will prevent workflows from executing.

When an external database is used, the notes under "Retaining the source database" above do not apply as the source database is not connected to the target farm.

When data has been pushed to an external database, the 'Storage' table in the Nintex Workflow configuration database for the target farm will need to be manually updated to link the target site collection with the correct content database. The 'Storage' table lists site collection IDs and the Nintex Workflow database ID to determine the database that the site collection will use. The IDs for the databases connected to the environment can be retrieved from 'Databases' table in the Nintex Workflow configuration database.

Changing the Site Collection ID

In migration scenarios, the ID of the site collection can change. For example, if a site collection is backed up from one farm and restored to another, it will be assigned a new site collection ID. Nintex Workflow application pages and workflow instances use the site collection ID to find data in the Nintex Workflow content database. If the site collection ID changes, and the records in the database are using an out of date ID, the data for a workflow will not be found.

In this case, the MoveData command allows the new site collection ID to be provided as part of the command. When data is written into the target database, it can be written with a new site collection ID, allowing the product to find the data when accessed from the restored site collection.

Usage

```
NWAdmin.exe -o MoveData [-Url <siteCollectionUrl> | -SiteID <site collection GUID>] [-SourceDatabase <connection string>] [-TargetDatabase <connection string>] [-NewSiteID <target site collection GUID>] [-RetainSourceData]
```

Parameters

Name	Description
-Url	The url to the top level site of the site collection that data will be moved for.
-SiteID	The ID of the site collection to move data for. Use when retrieving data from an external database.
-SourceDatabase	A database connection string to the Nintex Workflow database to extract data from. Use when retrieving data from an external database not connected to the target environment.
-TargetDatabase	A database connection string to the Nintex Workflow database to move data to. If a value is not provided the console UI will prompt for the target database.
-NewSiteID	A GUID to override the site collection ID data when the data is written in to the target database. Used in migration scenarios where the site collection ID of the site in SharePoint is different between the source environment and the target environment. If not provided, no change will be made to the site collection ID.
-RetainSourceData	Option to not delete data from the source database when the operation has finished inserting data into the target database. When not provided, data inserted into the target database will be deleted from the source database.

PrepareSiteForExport

Description

This operation is used to prepare a site that will be exported with stsadm.exe -o export so workflows will not need to be manually reconfigured in the site that is imported from this file.

Note: This command should be run immediately **before** stsadm.exe -o export so all information is up to date.

The command adds metadata to the site in the form of a file named SiteMetaData.xml stored in the hidden NintexWorkflows library on the site. This file is used when the site is imported to update the list ids used in the workflow.

See also the [FixSiteAfterImport](#) command.

Usage

```
NWAdmin.exe -o PrepareSiteForExport -siteUrl urlToSiteToPrepare
```

Parameters

Name	Description
-siteUrl	The URL to the site that will be exported with stsadm.exe.

PurgeHistoryListData

Description

This operation is used to remove workflow history entries from the workflow history list on a site. This operation can be used for maintenance on the history list to prevent the number of history items exceeding 5000.

Usage

```
NWAdmin.exe -o PurgeHistoryListData -siteUrl urlToSite [-workflowName workflowName] [-days daysSinceLastActivity] [-lastActivityBefore datetime DateFormat]] [-state All|Running|Completed|Cancelled|Error] [-deletedLists] [-clearAll [-workflowItemId id -workflowListName "list name"]] [-verbose] [-reportOnly] [-batchSize numberDefaultIs500] [-pauseAfterBatch] [-maxItemsToDelete number] [-silent]
```

Parameters

Name	Description
-siteUrl	The URL of the site to delete history list items from.
-workflowName	Specify to remove history list items for workflows with a specific name.
-days	Specify to remove history list items for workflows that have had no activity since this number of days.
-lastActivityBefore	Specify to remove history list items for workflows that have had no activity since this date. Run "NWAdmin -o PurgeHistoryListData" to see the format to specify the data and time value in.
-state	Specify to remove history list items for workflows that are either Running, Completed, Cancelled, Error or any state. The default is Completed.
-deletedLists	Specify to remove history list items for any workflow that as associated to a list that is now deleted.
-clearAll	Removes all data from the history list, regardless of what other parameters are chosen except "-workflowItemId" and "-workflowListName". To remove history list entries for workflows that are not recorded in the Nintex Workflow database, this option must be used.
-workflowItemId	Used in conjunction with the -clearAll and -workflowListName command. Used to clear history list items for workflows associated with a particular item.
-workflowListName	Used in conjunction with the -clearAll and -workflowItemId command. Used to clear history list items for workflows associated with a particular item.
-verbose	Writes additional information to the screen.
-reportOnly	Runs through the operation without making any changes to data.
-batchSize	The number of items in the history list to process at once.
-pauseAfterBatch	After each batch is run, the user is prompted whether or not to process the next batch. Allows review of the changes in SharePoint before continuing.
-maxItemsToDelete	An integer determining the maximum number of history items that the process is allowed to delete. Once this limit is reached, the process will abort.
-silent	Specifies the command should run without prompting for confirmation.

PurgeWorkflowData

Description

This operation is used to remove historic data from the Nintex Workflow database. Reasons that data may need to be removed include:

- Removing old data to decrease the size of the Nintex Workflow content databases.
- Removing data added during testing of a workflow.

The operation accepts a number of arguments to use in combination to build up a filter of which data to remove. Before executing the operation, a summary of the selected filters will be displayed for confirmation. Only database entries matching all entered criteria will be purged.

Caution: Once data has been purged, it cannot be recovered without restoring a backed up version of the database.

Consequences

After this command has run, Nintex Workflow will no longer have any record that the affected workflows were executed. The affected workflow instances will:

- No longer be listed as having run on the 'view workflow history' page.
- No longer be able to have their graphical status of detailed history viewed.
- No longer affect statistics or reports.
- No longer appear in web parts.
- No longer have viewable tasks (the task item will exist, but clicking on it will receive an error about not being able to find the ID in the database).
- No longer be able to have history list items associated with them purged using the 'PurgeHistoryListData' command unless the `-clearall` switch is used.

SharePoint views, such as accessing the 'workflows' link of a list item, will still display the workflow.

Usage

```
NWAdmin.exe -o PurgeWorkflowData [-workflowName workflowName] [-state [All|Running|Completed|Cancelled|Error]] [-url teamsiteUrl [-listName listName]] [-siteCollectionUrl siteCollectionUrl] [-deletedLists] [-lastActivityBeforeUTC lastActivityBeforeDateUTC (" + DateFormat + ")] [-lastActivityBeforeLocal lastActivityBeforeDateLocal (" + DateFormat + ")] [-listId listId] [-itemId itemId] [-instanceId workflowInstanceId] [-initiator loginname] [-timeout seconds] [-silent] [-purgeAll] [-deletedSites]
```

Parameters

Name	Description
<code>-workflowName</code>	Remove historic data for workflows with this name.
<code>-state</code>	Remove historic data for workflows in a specific state.
<code>-url</code>	Remove historic data for workflows in a site.
<code>-listName</code>	Remove historic data for workflows associated with lists with this name.
<code>-siteCollectionUrl</code>	Remove historic data for workflows in a site collection.
<code>-deletedLists</code>	When specified, any historic data associated with a list or library that has been deleted will be removed. When the <code>-url</code> or <code>-siteCollectionUrl</code> parameter is specified, only workflow data from lists deleted from the URL site will be removed. When <code>-url</code> is not specified, the operation is global for all sites. If a site has been deleted, the list is also considered deleted and the workflow data will be purged.

Name	Description
-lastActivityBeforeUTC	Remove historic data for workflows where the last action was before a specific time, specified in UTC format. Viewing the usage in stsadm will display the expected format to specify the date.
-lastActivityBeforeLocal	Remove historic data for workflows where the last action was before a specific time, specified in server local time. Viewing the usage in stsadm will display the expected format to specify the date.
-listId	Remove historic data for workflows associated with a specific list that has this internal GUID.
-itemId	Remove historic data for workflows that have run on list items with this integer ID.
-instanceId	Remove historic data for a specific workflow instance where the instance ID matches this GUID.
-initiator	Remove historic data for workflows where this is the user who started the workflow.
-timeout	The number of seconds before the operations displays an error due to a time out. Default is 600. Purging a large number of items can take some time.
-silent	Specifies the command should run without prompting for confirmation.
-purgeAll	When specified, all historic workflow data will be removed. Used without any additional parameters.
-deletedSites	When specified, any historic workflow data associated with a site that has been deleted will be removed. Used without any additional parameters.

RemoveAction

Description

This operation is used to remove a workflow action from the Nintex Workflow database. The action will no longer be able to be used in the Workflow designer or displayed in existing workflows.

Usage

NWAdmin.exe -o RemoveAction -adapterType <namespace.class of adapter> | -id <id of action>

Parameters

Name	Description
-adapterType	The namespace and class name of the adapter used to handle the action that is to be removed. Not required if the -id parameter is specified.
-id	The numeric ID of the action. No required if the -adapterType parameter is specified.

RemoveCustomDataItems

Description

This operation is used to remove a custom workflow context property, so that it can no longer be used in the Workflow designer. Custom workflow context properties are displayed in the Common tab of the Insert Reference dialog. For more information about creating custom workflow context properties, see the Nintex Workflow 2013 SDK.

Usage

```
NWAdmin.exe -o RemoveCustomDataItem -type typename [-assembly assemblyname]
```

Parameters

Name	Description
- typename	Required. The fully qualified type name, including namespace, of the class for the custom workflow context property. The maximum length of this parameter is 255 characters.
- assembly	Optional. The fully-qualified four-part name of the assembly that contains the type for the custom workflow context property. Specify a value for this property if two or more custom workflow context properties share the same type name, to uniquely identify the custom workflow context property to be removed. The maximum length of this parameter is 255 characters.

Remarks

This operation only removes the relevant references to the custom workflow context property from the configuration database for Nintex Workflow 2013. This operation does not remove the assembly containing the custom workflow context property from the Global Assembly Cache (GAC).

RemoveEventReceiver

Description

Removes an event receiver listening to a Nintex Workflow event.

Usage

Intended for use when instructed by Nintex customer service. Removing event receivers may break functionality.

RemoveInlineFunction

Description

This operation is used to remove a registered function alias for an inline function, so the function alias can no longer be used in a workflow.

Usage

NWAdmin.exe -o RemoveInlineFunction -functionalias FunctionAlias

Parameters

Name	Description
-	Required. A string that represents the name of the function alias to functionalias remove for the inline function.

Remarks

This operation only removes the specified function alias from the Nintex configuration database. It does not remove the corresponding assembly from the Global Assembly Cache (GAC).

If the function alias is already in use in a workflow, the function alias will no longer be evaluated and is instead treated as ordinary text. If an inline function has multiple registered function aliases, only the function alias removed by this operation is no longer available. The other function aliases remain useable. If you want to completely remove an inline function from Nintex Workflow, you must remove all registered function aliases for that inline function.

Note: Use this operation carefully. This operation can remove any function alias, including function aliases defined by Nintex Workflow.

RemoveSafeActions

Description

This operation is used to remove an authorized workflow action type that was previously registered in the web.config file.

Note, once an action has been unregistered, workflows containing this action will not run. Unregistering the safe action entry will not remove the code from the server Global Assembly Cache or remove the entries in the Nintex workflow configuration database.

Usage

NWAdmin.exe -o RemoveSafeActions -namespace <namespace> -type <type name> -assembly <assembly name> [-url urlToWebApplication] [-verbose] [-confirmAll]

Parameters

Name	Description
- namespace	The namespace of authorised type entries to remove.
-type	The class type of authorised type entries to remove.
-assembly	The assembly of authorised type entries to remove.
-url	Optional. The URL of the web application to remove entries from. If a URL is not specified, entries are removed from all web applications.
-verbose	If specified each entry that will be removed is displayed.
-confirmAll	If specified, the user is prompted to confirm that each entry should be removed.

RemoveZoneSetting

Description

This operation is used to remove a configure zone URL setting.

Usage

NWAdmin.exe -o RemoveZoneSetting -id ZoneConfigurationId

Parameters

Name	Description
-id	The numeric ID representing the rule to remove. Use EnumZoneSettings to determine the ID of each rule.

ResolveSharePointIds

Description

This operation is used to convert SharePoint internal Ids to URLs.

Usage

NWAdmin.exe -o ResolveSharePointIds -siteId guid [-webId guid] [-listId guid]

Parameters

Name	Description
-siteId	The GUID of the site collection.
-webId	The GUID of the team site. Requires that the -siteId parameter is also provided.
-listId	The GUID of the list. Requires that the -webId and -siteId parameters are also provided.

SetCallWebServiceTimeout

Description

This operation is used to specify how long every call web service action should wait before causing an error due to a timeout.

Note that having many workflows that remain processing in memory for a long time (for example, while waiting for a web service response) is not recommended.

Usage

NWAdmin.exe -o SetCallWebServiceTimeout -milliseconds numberOfMilliSeconds

Parameters

Name	Description
- milliseconds	The number of milliseconds the call web service action should wait before timing out. The default is 100,000 milliseconds (100 seconds).

SetTaskForm

Description

This operation is used to set the task response form for a task content type. This determines the page that is displayed when a user responds to a task generated by a workflow. See the SDK for more information on custom task forms.

Usage

```
NWAdmin.exe -o SetTaskForm -siteUrl teamsiteurl -contentType contenttypename -taskformUrl taskformurl [-formType Edit|View|All] [-updatechildren Yes|No]
```

Parameters

Name	Description
-siteUrl	The URL to the site containing the content type to modify.
-contentType	The name of the content type to modify.
-taskformUrl	The URL to the aspx page to use as the task form. If the page is deployed to layouts, the URL will be in the format of “_layouts/folder/page.aspx”
-formType	Specifies whether the edit form, display form or both forms should be set to this URL. Default value is 'All'
-updateChildren	Replicate the change to content types that inherit from the specified content type. If the content type has already been associated with a list, 'yes' must be specified to update the URL for list items. Default value is 'No'

SyncTerminatedWorkflows

Description

This operation is used to find workflows where the status stored in the Nintex Workflow does not match SharePoint's representation of the status. The Nintex Workflow status is updated to match the SharePoint status.

Usage

NWAdmin.exe -o SyncTerminatedWorkflows -url teamsiteurl [-preScan] [-verbose] [-confirm] [-showMissingItems] [-terminateDeletedItems]

Parameters

Name	Description
-url	The url to the site to search for out of sync workflows.
-preScan	When specified, out of sync workflows are displayed but no changes are made.
-verbose	When specified, detailed information of the actions taken by the operation are displayed.
-confirm	When specified, a prompt is displayed to confirm before each action.
-showMissingItems	When specified, additional output is displayed when a workflow instance is stored for an item that no longer exists in SharePoint.
- terminateDeletedItems	When specified, if a list item is not found, the workflow is treated as if it was cancelled. SharePoint does not cancel a workflow when an item is deleted, so the workflow can still display in reports. This option will tell Nintex Workflow to treat the workflow as cancelled.

Note: Security can cause this command to treat an item as deleted.

Note: Items in the site recycle bin will be treated as deleted.

UninstallTimerJob

Description

This operation is used to uninstall a Nintex Workflow timer job from an environment.

Nintex Workflow uses two timer jobs.

- **ScheduledWorkflows**
This job is used to run workflows that have been set to run to a schedule. This job is installed in all environments.
- **TaskProcessing**
This job is to process some task responses.

Usage

NWAdmin.exe -o UninstallTimerJob -job ScheduledWorkflows|TaskProcessing [-url webApplicationUrl]

Parameters

Name	Description
-job	Specifies which timer job to uninstall.
-url	[Advanced] A URL to a web application. If not specified, the central administration web application is used.

UpdateContentDatabase

Description

This operation is used to update a reference to a Nintex Workflow content database.

When migrating a configuration database to another server, the internal reference to itself as a content database must be updated with this command.

Usage

NWAdmin.exe -o UpdateContentDatabase -serverName serverName [-databaseName databaseName] [-newServerName newServerName] [-newDatabaseName newDatabaseName] [-newFailoverPartner newFailoverPartner] [-integrated] [-username username -password password]

Parameters

Name	Description
-serverName	The existing server name of the content database to update.
-databaseName	Optional, the existing database name of the content database to update. Used to differentiate two databases on the same server.
-newServerName	Optional, the server name to update the content database record with.
-newDatabaseName	Optional, the database name to update the content database record with.
-newFailoverPartner	Optional, the failover database server for the content database.
-integrated	Optional, change the content database record to use integrated security.
-username	Optional, change the content database record to use database authentication with this username. Use in conjunction with password.
-password	Optional, the password to use if setting the content database record to use database authentication.

UpdateContentDatabaseParameters

Description

This operation is used to update parameters for Nintex Workflow content databases, including setting MultiSubnetFailover to True to make content databases "AlwaysOn."

For more information on "AlwaysOn" databases, see the Microsoft article [SqlClient Support for High Availability, Disaster Recovery](#).

Note: Run the IISreset command to implement the changes from this operation. Parameters for databases are listed on the Database management page. For more information on this page, see the product help. When migrating a configuration database to another server, the internal reference to itself as a content database must be updated with this command.

Usage

```
NWAdmin.exe -o UpdateContentDatabaseParameters -serverName serverName -databaseName databaseName -newParameters "<Parameters>"
```

Example updating the parameters MultiSubnetFailover, ApplicationIntent, and Encrypt for the content database "NintexWorkflowDB-Content" on the database server "MyServer":

```
NWAdmin.exe -o UpdateContentDatabaseParameters -serverName MyServer -databaseName NintexWorkflowDB-Content -newParameters "MultiSubnetFailover=True;ApplicationIntent=ReadOnly;Encrypt=False"
```

Parameters

Name	Description
-serverName	The name of the database server for the content database parameter update.
-databaseName	The name of the content database for the parameter update.
-newParameters	The database parameters to update. For a list of parameters, see https://msdn.microsoft.com/en-us/library/ms378988%28v=sql.110%29.aspx ; at the time of writing, all except the following parameters were supported. Unsupported parameters: databaseName, password, serverName, userName, integratedSecurity.

UpgradeDatabases

Description

This operation is used to upgrade each Nintex Workflow database. This is the same as clicking the 'upgrade now' link in the Nintex Workflow database management page in SharePoint Central Administration.

Usage

NWAdmin.exe -o UpgradeDatabases [-database <database connection string> -isConfig]

Parameters

Name	Description
- database	The connection string to the database to be upgraded.
-isConfig	If specified, indicates that the database to be upgraded is a configuration database; otherwise, the database to be upgraded is a content database.