

# Nintex Workflow 2007: NWAdmin operations

## Table of Contents

Introduction .....	3
AddAction .....	4
AddCustomDataItem .....	5
AddInlineFunction .....	6
AddZoneSetting .....	7
CheckDatabaseVersion .....	9
CleanTaskRedirects .....	10
ConfigureSettings .....	11
DelegateAllTasks .....	12
DeployWorkflow .....	13
DepreciateActivity .....	14
EnumCustomDataItems .....	15
EnumSafeActions .....	16
EnumSchedules .....	17
EnumInlineFunctions .....	18
EnumZoneSettings .....	19
ExportWorkflow .....	20
ExportWorkflowConstants .....	21
FindWorkflows .....	22
FixSiteAfterImport .....	23
GenerateServerNameHash .....	24
ImportWorkflowConstants .....	25
InstallTimerJob .....	26
MessageTemplateUsage .....	27
MigrateUser .....	28
MoveData .....	29
PrepareSiteForExport .....	30

PurgeWorkflowData .....	31
PurgeHistoryListData .....	33
RemoveAction .....	35
RemoveCustomDataItems .....	36
RemoveSafeActions .....	37
RemoveInlineFunction .....	38
RemoveZoneSetting .....	39
ResolveSharePointIds .....	40
SetCallWebServiceTimeout .....	41
SetTaskForm .....	42
SyncTerminatedWorkflows .....	43
UninstallTimerJob .....	44
UpdateContentDatabase .....	45
UpgradeDatabases .....	46

## Introduction

NWAdmin is a command line tool that ships with Nintex Workflow 2007. It is used to perform various administration operations.

The NWAdmin.exe tool is found in the directory where Nintex Workflow 2007 was installed on the server where the MSI was run.

The default install directory for Nintex Workflow 2007 is:

[on a 32bit environment]

c:\program files\nintex\nintex workflow 2007

[on a 64bit environment]

c:\program files (x86)\nintex\nintex workflow 2007

This document will always reflect the copy of NWAdmin released with the most recent build of Nintex Workflow 2007. If you find a documented operation is missing, contact [support@nintex.com](mailto:support@nintex.com) for information on updating your build of Nintex Workflow 2007.

*This document is current for Nintex Workflow 2007 Build 11200*

## 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 -nwaFilepathToNWAFile [-updateIfAdapterExists] [-activateForFarm] [-verifyTypes]

### Parameters

Name	Description
-nwaFile	The location on the computer of the .nwa file containing the action details to import.
-updateIfAdapterExists	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	Specifies that the action should be activated on the 'manage allowed actions' page for the entire farm when this parameter is provided.
-verifyTypes	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 install custom 'Common' data items that can be inserted into text fields when configuring workflow action. See the SDK for details on creating a custom data item.

### Usage

NWAdmin.exe -o AddCustomDataItem -type typename -assembly assemblyname

### Parameters

Name	Description
-type	The namespace and class name of the custom data item class.
-assembly	The fully qualified assembly name of the assembly containing the custom data item. This assembly must be installed into the Global Assembly Cache on each front end server.

## AddInlineFunction

### Description

This operation is used to register a new inline function so that it can be used in the workflow designer. To add multiple aliases for a function, run this command for each one, specifying a different alias parameter.

For more information on the requirements for a valid inline function, see the Nintex Workflow 2007 SDK (<http://connect.nintex.com/files/folders/sdk/entry122.aspx>).

### Usage

```
NWAdmin.exe -o AddInlineFunction -functionalias FunctionAlias -assembly AssemblyName -namespace Namespace -typename TypeName -method MethodName
```

### Parameters

Name	Description
-functionalias	The name that the function will be referenced with when designing workflows. A single function can have multiple aliases, but AddInlineFunction can only add one function at a time.
-assembly	The fully qualified name of the assembly containing the code for the function. This assembly must be installed into the GAC.
-namespace	The namespace in the assembly that contains the type with the method that will be invoked
-typename	The name of the class that contains the method that will be invoked. This parameter should include the class name only, without the namespace.
-method	The name of the method in the specified type that should be invoked when the function alias is used in a workflow. Note, the workflow engine will automatically determine the parameters required for the method.

## 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: <ul style="list-style-type: none"> <li>• WebApplication</li> <li>• SiteCollection</li> <li>• TeamSite</li> <li>• ExplicitList</li> <li>• ExplicitWorkflow</li> </ul>
-list	The name of the list to define the scope of the type entry. Required for

	<p>types:</p> <ul style="list-style-type: none"> <li>• AnyList</li> <li>• ExplicitList</li> <li>• ExplicitWorkflow</li> </ul>
-workflow	<p>The name of the workflow to define the scope of the type entry. Required for types:</p> <ul style="list-style-type: none"> <li>• AnyWorkflow</li> <li>• ExplicitWorkflow</li> </ul>
-zone	<p>The url zone to use when a workflow matches this entry. See the Alternate Access Mappings screen in SharePoint Central Administration for more information.</p>



## 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

When the NWAdmin -o MoveData command is used to move data for a site collection to a new content database, workflow task redirect records are created to point to the relocated task records in the new database. This operation is used to clean up any workflow task redirect records that are no longer required.

The command will prompt the user to specify which database they wish to analyse and remove records from.

### Usage

NWAdmin.exe -o CleanTaskRedirects [-test]

### Parameters

Name	Description
-test	Perform a test run (won't delete 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 2007 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

## 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 that 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-username username -password password [-domain domain] [-siteUrl urlToASharePointSite] [-comments comments] [-sendNotification]
```

### 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.
-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.
-password	A valid password for -username
-domain	A valid domain for -username
-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.
-comments	Optionally add any comments to explain the delegation.
-sendNotification	Provide this parameter to send a notification to the new assignee informing them that the task is now assigned to them.

## DeployWorkflow

### Description

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

Note that this operation uses the Nintex Workflow web service, and requires credentials to connect to the web service (windows login user will be used if credentials are not provided).

### Usage

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

### 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.
-username	A valid SharePoint user's username with permissions to publish a workflow. If no username is specified, the credentials of the user running the command will be used.
-password	A valid password for -username
-domain	A valid domain for -username
-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.

## DepreciateActivity

### Description

This operation is used to automatically upgrade one workflow action with another.

When an action is depreciated, the activity will still be used and loaded when a workflow is opened, but once that workflow is published again, the activity marked as the replacement will automatically be added to the workflow instead of the original activity.

### Usage

```
NWAdmin.exe -o DepreciateActivity -oldActivityType Namespace.Type -oldActivityAssembly FullyQualifiedAssemblyName -oldAdapterType Namespace.Type -oldAdapterAssembly FullyQualifiedAssemblyName -newActivityType Namespace.Type -newActivityAssembly FullyQualifiedAssemblyName -newAdapterType Namespace.Type -newAdapterAssembly FullyQualifiedAssemblyName
```

### Parameters

Name	Description
-oldActivityType	The class namespace and type name of the activity to be replaced.
-oldActivityAssembly	The four-part assembly name of the activity to be replaced.
-oldAdapterType	The class namespace and type name of the adapter representing the activity to be replaced.
-oldAdapterAssembly	The four-part assembly name of the adapter representing the activity to be replaced.
-newActivityType	The class namespace and type name of the new activity.
-newActivityAssembly	The four-part assembly name of the new activity.
-newAdapterType	The class namespace and type name of the adapter representing the new activity.
-newAdapterAssembly	The four-part assembly name of the adapter representing the new activity.

## EnumCustomDataItems

### Description

This operation is used to display all the custom 'Common' insertable data items registered with Nintex Workflow 2007. See the SDK for more information on custom data items. Each data item's name and type information will be displayed.

### Usage

NWAdmin.exe -o EnumCustomDataItems

### Parameters

This operation has no parameters.

## 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 2007 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

### Parameters

This operation has no parameters.

## EnumInlineFunctions

### Description

This operation is used to display all the string functions that are registered with Nintex Workflow 2007.

### Usage

NWAdmin.exe -o EnumInlineFunctions

### Parameters

This operation has no parameters.

## **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 that this operation uses the Nintex Workflow web service, and requires credentials to connect to the web service.

### Usage

```
NWAdmin.exe -o ExportWorkflow -siteUrl targetSiteUrl -list targetList -workflowName workflowName -filename fileToSave -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.
-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.
-username	A valid SharePoint user's username with permissions to publish a workflow.
-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 environments. 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 2007 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 2007 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.

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, so make take some time, and 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

This operation is used to generate a hashed version of a machine name for licensing purposes.

### Usage

```
NWAdmin.exe -o GenerateServerNameHash [-servername servername]
```

### Parameters

Name	Description
-servername	Optional. The name of the server to hash. If this parameter is not specified the name of the computer on which the command is being run will be hashed.



## 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

Note that 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 2007 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. It is only installed in environments that were updated from a build 10829 or earlier. It is used to process task responses where a) the SharePoint infrastructure updates are installed and b) the workflow containing the task was published in a build previous to 10830.

### 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; as 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).

After the data is moved, the source content database cannot be removed, even if there is no site collection still using that particular database. The reason is that some functionality will still look for data in the original database. The MoveData tool will add redirect records in this database for this functionality, so any process checking for data in this database will be redirected to look in the new database. Therefore, the database must still exist.

### Usage

NWAdmin.exe -o MoveData -url siteCollectionUrl

### Parameters

Name	Description
-url	The url to the top level site of the site collection that data will be moved for.

## 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.

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.

The FixSiteAfterImport command must be used immediately after the PrepareSiteForExport command. Please see the FixSiteAfterImport command for details.

### Usage

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

### Parameters

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

## PurgeWorkflowData

### Description

*WARNING: Performance degrades when querying a history list containing 15,000 or more items.*

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

1. Removing old data to decrease the size of the Nintex Workflow content databases.
2. 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.

Data Purged:

- Workflow instance.
- Every task associated with the workflow instance.
- Delegation History.
- Which actions ran in the workflow.

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 | -  
lastActivityBeforeLocal lastActivityBeforeDateLocal] [-listId listId] [-itemId itemId] [-instanceId  
workflowInstanceId] [-initiator loginname] [-purgeAll] [-deletedSites] [-timeout seconds] [-silent]
```

## Parameters

Name	Description
-workflowName	Remove historic data for workflows with this name.
-state	Remove historic data for workflows in a specific state.
-url	Remove historic data for workflows in a site.
-listName	Remove historic data for workflows associated with lists with this name.
-siteCollectionUrl	Remove historic data for workflows in a site collection.
-deletedLists	When specified, any historic data associated with a list or library that has been deleted will be removed. When the –url parameter is specified, only workflow data from lists deleted from the url site will be removed. When –url 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.
-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.
-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.
-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.



## 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 2000.

### 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.

## 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 previously registered custom 'Common' insertable data item. See the SDK for more information on custom data items.

### Usage

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

### Parameters

Name	Description
-typename	The namespace and class of the custom data item to remove.
-assembly	Optional. The full assembly name of the assembly containing the custom data item type. Use this is multiple custom data items have matching namespaces and class names.

## 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.

## RemoveInlineFunction

### Description

This operation is used to remove a registered string function so it can no longer be used in a workflow. If the function is already in uses in a workflow, it will no longer be evaluated and will be treated as ordinary text. If a function is also registered under an alternative alias, the alternative alias will still be usable.

### Usage

NWAdmin.exe -o RemoveInlineFunction -functionalias FunctionAlias

### Parameters

Name	Description
-functionalias	The name of the function to remove.

## 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.  <b>Note: Security can cause this command to treat an item as deleted.</b> <b>Note: Items in the site recycle bin will be treated as deleted.</b>

## UninstallTimerJob

### Description

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

Nintex Workflow 2007 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. It is only installed in environments that were updated from a build 10829 or earlier. It is used to process task responses where a) the SharePoint infrastructure updates are installed and b) the workflow containing the task was published in a build previous to 10830.

### 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] [-
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.
-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.

## 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 [-force]

### Parameters

Name	Description
-force	Optional. If this option is specified the upgrade script will be run even if an upgrade is not required.