



Database Design Guide: Nintex Workflow

Last updated: Monday, December 1, 2014

Legal Notices

For the latest information, please see

<https://www.nintex.com/en-US/Pages/TermsAndConditions.aspx>

Contents

Legal Notices	i
Introduction	1
Database descriptions	1
Database growth	1
Plan Nintex Workflow content databases and their mapping	2
Fewer than 25 SharePoint content databases	2
More than 25 SharePoint content databases	3
Plan maintenance	5
Database backup	5
Workflow data maintenance	7
Monitor workflow data	7
Purge workflow data	7
Rebuild or reset indexes	8
Split existing databases	9
Troubleshoot maintenance	10
Recover accidental deletion of workflow history	10
Diagnose causes of slow performance	11

Introduction

This document provides guidance on planning database mapping and maintenance for Nintex Workflow.

Every deployment of Nintex Workflow involves a configuration database by default. If no Nintex Workflow content databases are added, then the default configuration database also acts as a content database. In most cases, however, one or more Nintex Workflow content databases are also added and mapped to SharePoint content databases.

Database descriptions

The following databases are involved in setting up Nintex Workflow.

- **Default configuration database.** The default configuration database contains database mappings, LazyApproval settings, and other configuration information for Nintex. If no content databases are created during the installation process, then the default configuration database also acts as a content database, serving as repository for workflow history and task information for Nintex workflows on the linked site collection.
- **Nintex Workflow content databases.** Nintex Workflow content databases are typically (but not always) added during the installation process to store workflow history and task information for the linked site collections.
- **SharePoint content databases.** SharePoint content databases are mapped to Nintex Workflow content databases. In the context of database design, the important thing to remember about SharePoint content databases is that they need to be in sync with their mapped Nintex Workflow content databases; backup and restore activities should be synchronized for mapped databases.

Database growth

When workflows run, rows are added to the tables in the Nintex Workflow content databases, which contain workflow history and task information. If not monitored, these tables can grow large enough to cause performance issues.

Plan Nintex Workflow content databases and their mapping

Nintex Workflow content databases are needed for any site collections that use Nintex workflows, except for the unusual case when the default configuration database suffices.

Review your environment to determine how many SharePoint content databases correspond to site collections that use Nintex Workflow.

To plan database mapping for your environment

1. Identify the site collections that use Nintex Workflow.
2. Count the SharePoint content databases corresponding to these site collections.
3. Refer to the related section below.

Fewer than 25 SharePoint content databases

Use 1:1 mapping if your environment has fewer than 25 SharePoint content databases corresponding to site collections that use Nintex Workflow. Map each SharePoint content database to a Nintex Workflow content database.

Note: This practice is most helpful when workflows are of limited complexity and low to medium traffic. If workflows in your environment are more complex or tend toward high traffic, see ["More than 25 SharePoint content databases" on the next page](#).

In scenario A, a company uses two SharePoint content databases, WSS_Content_01 and WSS_Content_02, for leave requests and other HR workflows. Only one site collection is active per database. Each database is mapped to a Nintex Workflow content database (either NW2010DB_Content01 or NW2010DB_Content02).

Note: The illustration references typical database names for Nintex Workflow 2010; however, the illustration applies to all versions of Nintex Workflow for SharePoint.

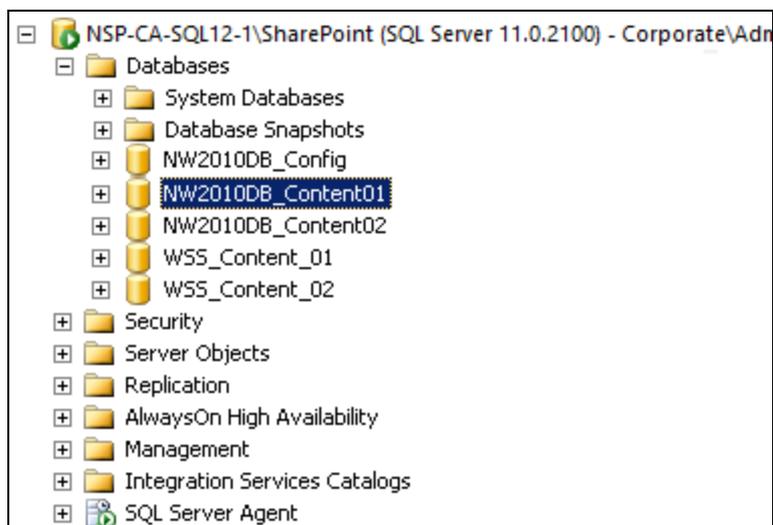


Figure 1-1 In this scenario, WSS_Content_01 is mapped to NW2010DB_Content01 and WSS_Content_02 is mapped to NW2010DB_Content02.

In scenario B, a company uses 10 SharePoint content databases for HR and Finance workflows: again, with only one site collection per database, and with each database mapped to a Nintex Workflow content database. For example, WSS_Content_10 is mapped to NW2010DB_Content10.



Figure 1-2 In this scenario, each of the SharePoint databases is mapped to a Nintex database.

More than 25 SharePoint content databases

Group Nintex Workflow content databases by web application or department if your environment has 25 or more SharePoint content databases, or when your environment contains complex or high-traffic workflows. This practice helps you avoid an increased database maintenance burden by limiting the number of Nintex Workflow content databases per SharePoint content database.

In scenario C, IT uses 10 SharePoint content databases, HR uses 5, Finance uses 15, and Sales uses 12. Each department's set of SharePoint content databases is mapped to a single Nintex Workflow content database.

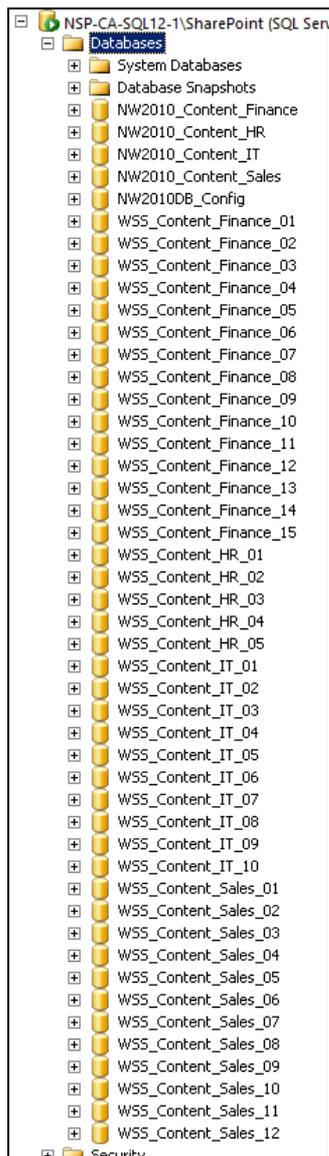


Figure 1-3 In this scenario, each department's set of SharePoint content databases is mapped to a single Nintex content database. For example, the 15 SharePoint content databases owned by Finance are mapped to a single Nintex content database: `NW2010_Content_Finance`.

Note: The illustration references typical database names for Nintex Workflow 2010; however, the illustration applies to all versions of Nintex Workflow for SharePoint.

Plan maintenance

To keep database size optimal for performance, maintain all content databases used with Nintex Workflow: Nintex Workflow content databases as well as their mapped SharePoint content databases. Maintenance involves database backup and workflow data maintenance.

Database backup

Developing a database backup strategy involves synchronizing backup and restore activities for mapped databases. For full recovery model, the backup strategy also involves scheduling routine backups of transaction logs.

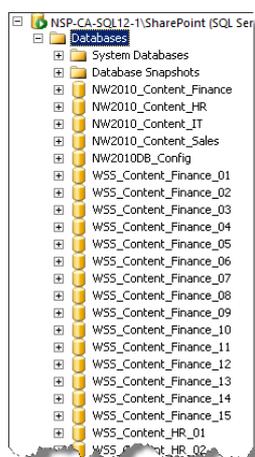
Synchronize backup and restore activities for mapped databases to prevent out-of-sync data.

For example, if ten SharePoint content databases are mapped to a single Nintex Workflow content database, then all 11 databases must be backed up at the same time. Restore of these databases also must happen at the same time.

Because of this requirement to synchronize backup and restore, take special care when setting up incremental (transaction log) backups. Be sure to configure the sequence of backup so that mapped databases are backed up together.

Note: The Nintex Workflow configuration database can be backed up at any time, as long as it is not acting as a content database.

For example, consider the earlier scenario in which content databases are organized by department.



With the default backup sequence, all Nintex databases ("NW" database names) are backed up before SharePoint databases ("WSS" database names). A typical fifteen-minute period for incremental backups could cause the Nintex Workflow content database NW2010_Content_Sales to become out of sync with its mapped SharePoint content databases (those database names beginning with WSS_Content_Sales_).

With the configured backup sequence, mapped databases are less likely to go out of sync. For example, NW2010_Content_Sales is backed up with its 12 mapped SharePoint databases (those database names beginning with WSS_Content_Sales_).

You can configure backup sequences using scripts, SQL Server agents, or third-party backup tools.

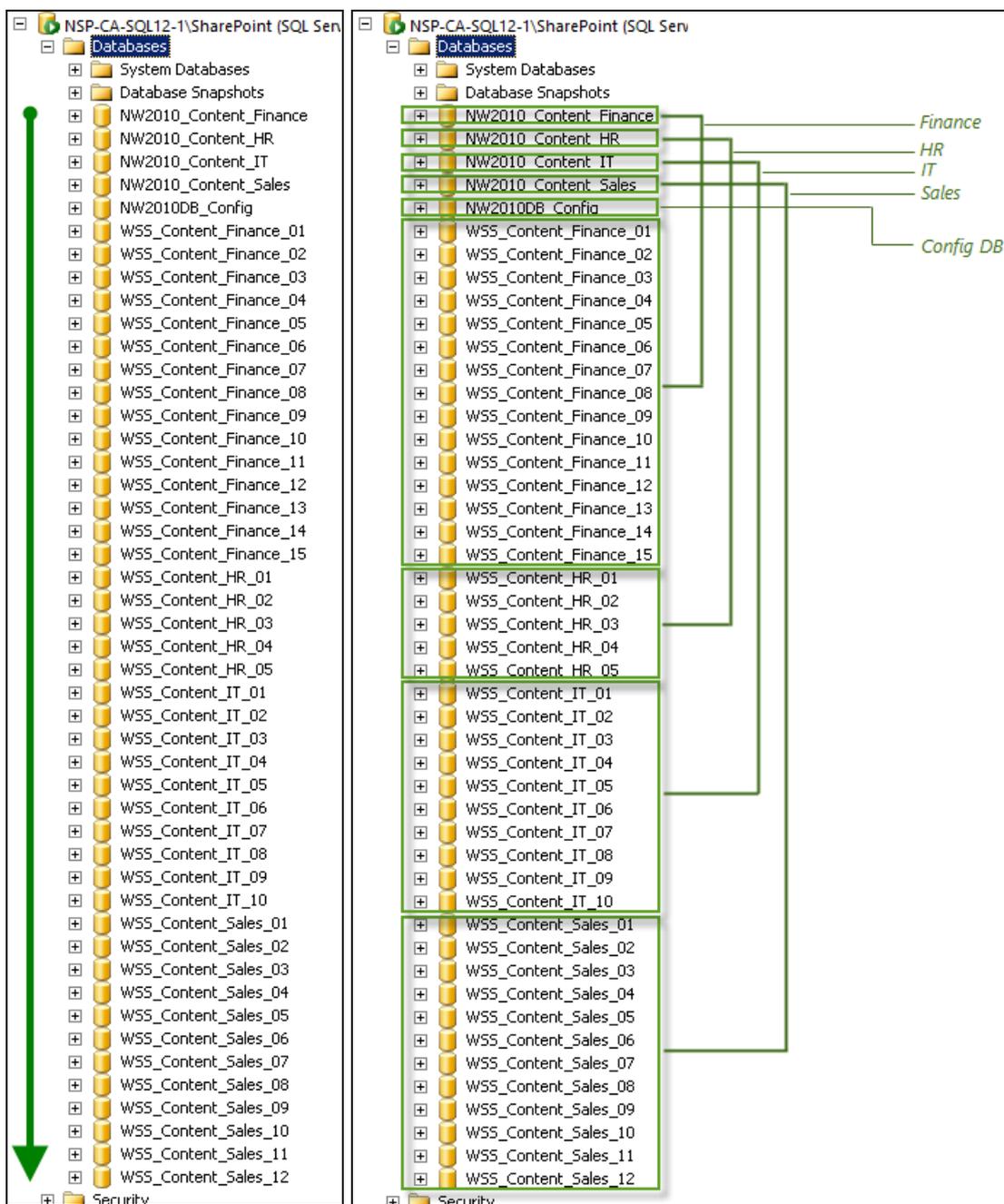


Figure 1-4 Default backup sequence.

Figure 1-5 Configured backup sequence to back up mapped databases together.

If your databases use the full recovery model, then make sure you also back up the transaction log to keep its file size under control. For more information on transaction log growth and truncation, see the following TechNet article: [Transaction Log Truncation](#).

For detailed instructions on backup and restore practices for Nintex Workflow, see the following Nintex article: [Nintex Workflow Backup and Restore](#). For more information about database backup with SharePoint and SQL Server, see the following TechNet articles: [Backup and Recovery \(SharePoint Foundation 2010\)](#) and [Backup Under the Full Recovery Model](#).

Workflow data maintenance

To prevent errors and slow workflows from overloaded content databases, we recommend regular maintenance of workflow data. Maintenance involves monitoring workflow data, purging select workflow data when thresholds are exceeded, and rebuilding or resetting indexes as needed.

Monitor workflow data

Monitoring workflow data involves defining a record-count threshold that is in line with your business systems; enterprise systems can typically continue performing with higher record counts than small businesses.

You can use a monitoring system such as System Center Operations Manager (SCOM) or you can create a SQL Server Agent job to run queries on a regular basis and send email if the specified threshold is exceeded. You can also manually run reports on a routine basis.

To run a report for record counts

- In Microsoft SQL Server Management Studio, right-click each Nintex Workflow content database and select **Reports > Standard Reports > Disk Usage by Top Tables**.

In the following report, the table `dbo.WorkflowProgress` contains over 10 million records. This large number of records may adversely affect performance in some environments. If errors or slow workflows were also occurring in this scenario, we would recommend purging select workflow data to reduce the record count of this table to below 10 million records if possible.

Table Name	# Records	Reserved (KB)	Data (KB)	Indexes (KB)	Unused (KB)
<code>dbo.WorkflowLog</code>	2,499,860	445,208,304	445,122,320	272	85,712
<code>dbo.WorkflowProgress</code>	10,775,207	1,096,424	846,656	249,696	72
<code>dbo.HumanWorkflowApprovers</code>	40,516	861,576	812,424	1,936	47,216
<code>dbo.WorkflowInstance</code>	87,669	73,304	53,368	15,088	4,848
<code>dbo.HumanWorkflow</code>	38,394	4,024	3,056	824	144

Note: Record counts that cause performance issues in some environments do not cause issues in others. Many environments with 100 million or more records run without issue. We recommend regularly monitoring your record count and then comparing the record count to your historical results if you start to notice performance issues. If you notice that your record count increased more than expected, it may be time to purge workflow data from the database to improve overall performance of workflows. For more information about purging workflow data, see ["Purge workflow data" below](#).

Purge workflow data

Purge workflow data to reduce the size of overloaded tables in Nintex Workflow content databases. Workflow data includes entries in the workflow history list and historic data stored in the database. When you purge data from Nintex Workflow content databases, the mapped SharePoint content databases are automatically updated.

Target deleted sites, workflows with errors, and workflow instances that have created surplus records when deciding which workflow data to purge.

Note: When purging workflow data, purge the workflow history list before purging historic data. If you purge historic data first, then the workflow history list items cannot be purged.

Although SharePoint offers a method for deleting items from the workflow history list on each site, the NWAdmin operation is preferred.

For information on the NWAdmin operations PurgeHistoryListData and PurgeWorkflowData, see the "NWAdmin Operations" guide for your version of Nintex Workflow on [the Nintex Connect site](#).

Rebuild or reset indexes

After purging workflow data, assess index fragmentation to determine if the indexes should be rebuilt or reset.

For more information, see [the "Measure and reduce index fragmentation" section](#) of the following TechNet article: [Database maintenance for SharePoint 2010 Products](#).

Split existing databases

Sometimes the existing database organization needs splitting up to improve performance.

For example, a complex, heavily-used workflow may cause performance issues for other workflows and items on the database. In this case, you may decide to create a separate database for the high-traffic workflow.

For instructions on splitting existing databases, see the following article on Nintex Connect: [Splitting existing SharePoint and Nintex Content Databases](#).

Troubleshoot maintenance

This section covers issues that may occur during backup, recovery, and maintenance.

Recover accidental deletion of workflow history

Symptom

The wrong workflow history was purged using the NWAdmin operation PurgeHistoryListData.

Cause

Dates or other criteria were incorrectly entered.

Note: For more information about purging workflow history, see "[Workflow data maintenance](#)" on page 7.

Resolution

You can recover accidentally purged workflow history by restoring the databases, adding the restored databases to Nintex Workflow, and mapping the restored databases.

To recover purged workflow history

1. Using SQL Server Management Studio, restore the following databases:
 - The backed up Nintex content databases that contained the accidentally purged workflow history.
 - The SharePoint content databases that were mapped to these Nintex content databases.
2. Add the restored Nintex content database to Nintex Workflow.
 - a. On the Central Administration Home page, click **Nintex Workflow Management**.
 - b. Click **Database setup**.
 - c. Click **Add content database**.
 - d. Under Database Name, enter the name of the restored Nintex content database and then select the check box for **Connect to an existing database**.
 - e. Edit other settings as needed and then click **OK**.

3. Map the restored Nintex content database to the restored SharePoint content database.
 - a. On the Central Administration Home page, click **Nintex Workflow Management**.
 - b. Click **Database setup**.
 - c. Click **Manage database mappings**.
 - d. In the row for the restored SharePoint content database, under **Map to Workflow Content Database**, select the restored Nintex content database.
 - e. Click **OK**.

These changes can take up to 15 minutes to take effect.

- f. To make changes take effect more quickly, do one of the following:
 - Run the **IISreset** command to reset IIS.
 - Deactivate and reactivate the Nintex Workflow feature for the site collection: From Settings, click Site collection features (under Site Collection Administration) and then click **Deactivate**; when the deactivation is complete, click **Activate**.

This action forces SharePoint to look at the mapping table in SQL Server.

Note: Changes may take effect in SQL Server before taking effect in SharePoint. To check status in SQL Server, look for the mapping in SQL Server Management Studio: Find the primary ID of the new Nintex database in `dbo.Database`, then search for that ID in `ContentDbMapping` (which maps SharePoint content databases to Nintex content databases).

Diagnose causes of slow performance

If you notice issues with performance such as slow starting workflows, workflow history taking a long time to load, or other types of performance-related behavior, we recommend evaluating the size of following items. Excessive sizes for these items can cause performance issues.

- Workflow history lists. For more information, see the following Nintex article: [Defensive Workflow Design Part 1: Workflow History Lists](#).
- Workflow data in Nintex Workflow content databases. In particular, review the size of the following tables.
 - `dbo.WorkflowLog`: Verbose log information for Nintex workflows. This table grows only when verbose logging is enabled in Central Administration > Nintex Workflow Management > Global Settings.
 - `dbo.WorkflowProgress`: Action history for Nintex workflows that are run.