Workflow performance analysis tests

Introduction
This document is intended to provide some analytical tests that help determine if the SharePoint workflow engine and Nintex databases are being forced to cope with a high amount of workflow related load. References to running through the tests in this document are often made by Nintex Support to determine next steps in improving workflow related performance. They may lead to recommendations such as reducing the size of the WorkflowProgress database table by way of purging data, modifying workflow designs, using more workflow task lists and other such techniques.

If requested, please run the tests below and return the results to Nintex Support.

Documents that may be referenced after analysis of these results include:

Reduce the size of the workflow progress database table
Splitting existing SharePoint and Nintex content databases
Nintex Workflow 2010 - NWAdmin.exe Operations

Please note: While the majority of references may be to the 2010 platform, the same concepts apply to the 2007 and 2013 platforms.

SharePoint timer service pending jobs
All delay actions utilize the SharePoint timer service. If there is a greater issue with this service it can affect any action that utilizes delays. The effect on these actions might be they do not continue at the expected time, or they never continue.

The following actions may be affected:

- Delay for
- Delay until
- Any workflow action that involves tasks
- Wait for item update
- Loop (when Safe Looping is enabled)
- State machine (when Safe looping is enabled)

To determine how much work the Timer Service has pending, go to the SharePoint Content Database (the default name will be something like “WSS_Content”), expand it, expand “Tables”, right-click the "dbo.ScheduledWorkItems" table and select “Properties”. Take note of the “Row count” (see the screenshot below).
A high number of items indicates a fair degree of backlog in the timer service. This could cause various behaviours such as:

- Workflows seeming to never wake up.
- Old workflows fire off after a restart of the SharePoint Timer service but then later they all stop again.
- Delays being late to execute, and possibly progressively getting worse (1 minute late, until they are days late).

If this is the case, you may have a workflow which is designed in such a way that lots of work items that require waking up in the Timer service are being created. A real-world example we're aware of is a workflow design with a loop that is polling - it adds a job for every instance of the workflow that runs. If hundreds of instances are running, each adding a job periodically, the number of delay jobs can get quite large.

To clear that backlog of work items, the only supported method is to cancel the workflow instances that are causing the creation of many batched work items.
Workflows potentially causing bottlenecks

The following query will help find workflow instances that are generating large amounts of database entries. It should be run against the Nintex Workflow content database (or all of them if you have more than one). By default, there is only one Nintex Workflow database (called NW2007DB, NW2010DB or NW2013DB) which combines a configuration and content database.

```sql
select I.WorkflowName, I.WorkflowInstanceId, I.SiteId, I.WebId, I.ListId, I.ItemId, I.WorkflowInitiator, I.WorkflowId, I.State, COUNT(P.WorkflowProgressId) as ActionCount
from WorkflowInstance I inner join WorkflowProgress P
on I.InstanceId = P.InstanceId
group by I.WorkflowName, I.WorkflowInstanceId, I.SiteId, I.WebId, I.ListId, I.ItemId, I.WorkflowInitiator, I.WorkflowId, I.State
order by COUNT(P.WorkflowProgressId) desc
```

The workflows listed at the top of the results will show workflows generating the most amount of entries. It might indicate that there’s a logic flaw causing a bottleneck, the workflow history needs purging or what is logged to the history list needs to be reduced.

Large workflow history causing performance issues

A large database of workflow history can cause performance issues. To determine how many workflow history records are in the database, open SQL Server Management studio, expand the Nintex Workflow 2010 database, expand the “tables” folder, right-click the “dbo.WorkflowProgress” table and select “Properties”. Select the “Storage” page and in the “General” section, note the “Row count” figure.
Depending on the results, using the “NWAdmin” application to reduce the workflow history via the purge commands may be necessary (http://connect.nintex.com/files/folders/technical_and_white_papers_nw2010/entry12004.aspx).

**General database size report**
We are interested in seeing the amount of data in your database tables which could likely cause the SQL timeout errors.

1. Open SQL Management Studio
2. Expand the Database node
3. Find the Nintex Workflow database (usually named NW2007DB or NW2010DB)
4. Right-click on the Nintex Workflow database and select Reports > Standard Reports > Disk Usage by Table

Please send us a screenshot of the report so we can see which tables are using the most space.