ImageNow for Microsoft SQL Server

Best Practices Guide

ImageNow Version: 6.7.x

perceptivesoftware

Written by: Product Documentation, R&D

Date: September 2016

Table of Contents

Managing the ImageNow Database on SQL Server

This document provides best practices for managing the Microsoft SQL database server. In this document, you will find best practices defined for database configuration and maintenance, as well as improving system performance.

Choosing the right version of SQL Server

ImageNow supports the 64-bit, Standard and Enterprise versions of Microsoft SQL Server for 2008 SP1 to R2, and Microsoft SQL Server 2012 is supported when you run ImageNow Server patch 6.7.0.2717 or higher. You should use the Enterprise version of SQL Server if one or more of the following criteria apply to your organization:

- Your organization requires 24/7 server availability, so maintenance tasks must be performed on your server while it is online.
- Your organization has over 200 users.
- Your organization requires high availability options.

What is the INOW6 database structure?

Perceptive Software, Inc., structures the ImageNow 6 database, INOW6, to provide the greatest flexibility and growth for your data. To ensure the integrity and performance of the database, your Database Administrator (DBA) must perform the same general maintenance tasks for the INOW6 database as required for any Microsoft SQL Server database. These tasks include checking database integrity, backing up the database and log files, and defragmenting indexes. In SQL Server, you select the scheduling and procedures for these tasks in your database maintenance plan.

What are the most dynamic tables?

The INOW6 database contains several database tables used to store critical data within ImageNow. It is important to consider the most critical and dynamic information in the database when you plan maintenance because you may choose to schedule more frequent backups for this type of information. You set the backup schedule in your database maintenance plan. The following tables contain the most activity (inserts, deletes, and updates) in the INOW6 database:

- The physical objects table, in_phsob, contains information that relates documents to objects in the Object Storage Manager (OSM).
- The logical objects table, in_logob, links pages with documents and with their associated objects in the OSM.
- The document table, in_doc, contains index (DocKey) information for each document stored within ImageNow.
- The version table, in_version, contains document metadata, such as the time the document was created, modified, and last viewed, and the users who performed those actions.
- The subobject tables, in_subob and in_subob_annot, contain information about subobjects, such as annotations.
- When you implement ImageNow workflow, the queue tables, in_wf_item, in_wf_item_arch, in_wf_item_hist, in_wf_item_hist_arch, in_wf_item_queue_hist, and in_wf_item_queue_hist_arch, contain current and historical information about items in workflow.

- If you use the Content Server, the content table, in_content, contains dynamic information for content searches.
- The in_instance and in_instance_prop tables contain information about custom properties.

Although the timeframe you use for database tasks is dependent on your standards, you should review larger or more dynamic tables periodically to track performance. To decrease the amount of time required to perform maintenance tasks, you can perform selective index reorganization and re-indexing based on percent level of fragmentation.

Another consideration for the tables is stale statistics. Generally, statistics have become stale when a 10-percent greater change is required for the data set within the given table. You can create scripts that will gather just the stale information, which you can use in addition to selective index reorganization and re-indexing. If your organization does not operate on a 24/7 basis, you should apply the maintenance plan in this document.

About defragmenting indexes

All indexes become fragmented over time. Dynamic tables are more prone to fragmented indexes because records in these tables are frequently updated, deleted, and inserted. When an index becomes fragmented, it requires more input/output, (I/O), time when you select records which, in turn, diminishes response time. The best way to fix index fragmentation is, depending on the level of fragmentation, to reorganize or rebuild indexes on a regular basis.

Configuring your database system

The following recommendations are steps you can do while configuring your SQL Server system.

- Locate the transaction log on a separate drive to avoid disk contention.
- Make sure your virus scanner excludes the SQL Server \Data directory.
- Make sure you apply the latest service packs supported by your installed ImageNow version to your SQL Server version. Service pack releases generally provide you with the latest fixes and diagnostics.
- Ensure the MSSQL install directory is located on a separate drive from your operating system.

Maintaining the database system

Your DBA must perform maintenance tasks to manage any database within a DBMS like SQL Server. These tasks ensure the integrity, and most importantly, the performance of your data. When you store the INOW6 database in SQL Server, make sure you define the following tasks in your database maintenance plan:

- Reorganize data and index pages
- Perform the update statistics maintenance task
- Check database integrity including indexes
- Establish your database backup procedures
- Establish your transaction log backup procedures
- Rebuild your indexes as recommended in the steps below by reorganizing the indexes weekly and rebuilding them at least once a year.

Before creating your maintenance plan, increase the allotted tempdb space and the transaction log size to the recommended amounts. Then, when you create your database maintenance plan, make sure that you plan and schedule backups to meet or exceed the recommended backup schedule. The following sections provide the recommended backup schedule, as well as the steps you need to perform to increase tempdb space and transaction log size, and create a database maintenance plan.

Recommended backup schedule

The following list contains the minimum recommended backup requirements for ImageNow. As needed, tailor these recommendations to your specific environment.

- Back up the INOW6 database daily. To do this, you can perform a weekly full backup and daily incremental backups.
- Back up the transaction log more frequently than daily, depending on your transaction volume. If you have a high transaction volume, set the transaction log backup more frequently. Some customers with large transaction volume back up the transaction log every five minutes, while others choose to do it hourly. If you have a low volume environment, set the transaction log to back up less frequently, such as twice daily.

Increase temporary database space

Increase the space for tempdb to 250 MB before creating your maintenance plan.

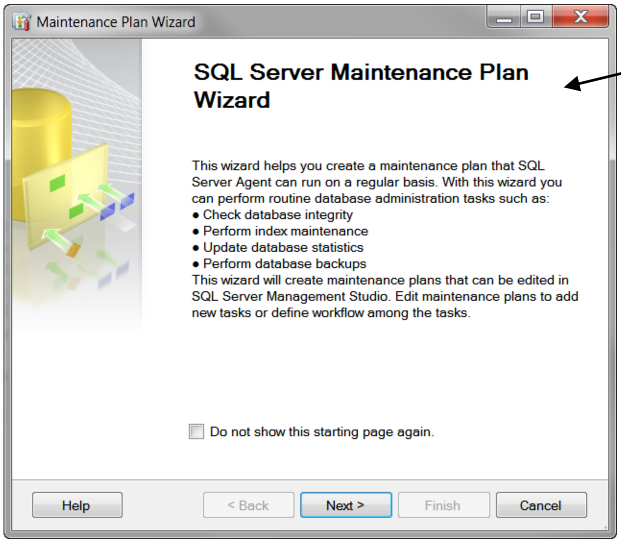
1. In **SQL Server Enterprise Manager**, in the **Console Root** window, open **Microsoft SQL Servers**, **SQL Server Group**, your server group, and then **Databases**.
2. Right-click **tempdb** and then select **Properties**.
3. In the **tempdb Properties** dialog box, on the **Data Files** tab, in the **tempdev** row, change **Space allocated** to **250**, and then click **OK**.

Schedule optimization

When you set the task to update database statistics in your Database Maintenance Plan, schedule it to run at a time when the server is not heavily used. Updating statistics of large tables or databases can take a lot of database server resources, which noticeably decreases performance. In most environments, SQL Server can gather statistics on a continuous basis.

Create a database maintenance plan

You can select and schedule database maintenance tasks in SQL Server using the Database Maintenance Plan wizard. If you have already created a Database Maintenance Plan, compare your settings to those shown in the "Use the Database Maintenance Plan wizard for SQL Server 2008" section of this document.



Maintenance Plan Wizard

SQL Server Maintenance Plan Wizard

This wizard helps you create a maintenance plan that SQL Server Agent can run on a regular basis. With this wizard you can perform routine database administration tasks such as:
- Check database integrity
- Perform index maintenance
- Update database statistics
- Perform database backups

This wizard will create maintenance plans that can be edited in SQL Server Management Studio. Edit maintenance plans to add new tasks or define workflow among the tasks.

□ Do not show this starting page again.

This wizard provides the steps you need to develop a plan to maintain any SQL Server database, including INOW6.

You can also write a script to perform database management tasks. However, scripts are beyond the scope of this document. For information about the commands to use in your scripts, consult the SQL Server help files.

Use the Database Maintenance Plan wizard for SQL Server 2008

Perform the following steps to create a new database maintenance plan for the INOW6 database using SQL Server 2008.

Note The following steps will vary if you are using a different service pack version. For information about increasing the performance of your database maintenance plan, see the Improving system performance section in this document.

1. Open **Microsoft SQL Server Management Studio (SSMS)**.
2. In the **Object Explorer** pane, open **Management**, right-click **Maintenance Plans**, and then select **Maintenance Plan Wizard**.
3. In the **Select Plan Properties** page, in **Name**, type a name for your INOW6 maintenance plan, click **Separate schedules for each task**, and then click **Next**.
4. In the **Select Maintenance Tasks** page, select every task except for **Define Reorganize Index**, **Shrink Database**, **Create Maintenance Cleanup Task**, and **Execute SQL Server Agent Job**, and then click **Next**.
5. In the **Select Maintenance Task Order** page, accept the default settings, and then click **Next**.

6. In the **Define Database Check Integrity Task** page, create the maintenance task to check the integrity of the INOW database, which examines the allocation and structural integrity of the tables and indexes in the database. To do this, complete the following steps.
   1. Click **Databases**, select **These databases**, and then select **INOW6**.
   2. Click **OK**.
   3. Verify that **Include indexes** is selected.
   4. Under **Schedule**, click **Change**.
   5. In the **Job Schedule Properties** dialog box, verify that the **Schedule type** is set to **Recurring**, and under **Frequency**, set **Occurs** to **Daily**.
   6. Under **Daily frequency**, specify an off-peak time, and then click **OK**.
   7. In the **Define Database Check Integrity Task** page, click **Next**.
7. In the **Define Rebuild Index Task** page, create the task to rebuild the indexes for all tables and views in INOW6 by setting the following options.
   1. Click **Databases**, select **These databases**, and then click **INOW6**.
   2. Click **OK**.
   3. Under **Schedule**, click **Change**.
   4. In the **Job Schedule Properties** dialog box, verify that the **Schedule type** is set to **Recurring**, and under **Frequency**, specify when you want to perform this task, such as daily, weekly, or monthly, and then click **OK**.
   5. In the **Define Rebuild Index Task** page, click **Next**.
8. In the **Define Update Statistics Task** page, create the task to automatically update statistics, which optimize performance, by setting the following options.
   1. Click **Databases**, select **These databases**, and then click **INOW6**.
   2. Click **OK**.
   3. Under **Schedule**, click **Change**.
   4. In the J**ob Schedule Properties** dialog box, verify that the **Schedule type** is set to **Recurring**, and under **Frequency**, set **Occurs** to **Daily**.
   5. Under **Daily Frequency**, specify an off-peak time, and then click **OK**.
   6. In the **Define Update Statistics Task** page, click **Next**.
9. In the **Define History Cleanup Task** page, create a task to clear old task history by setting the following options.
   1. Under **Select the historical data to delete**, verify that all three options are selected.
   2. Under **Remove historical data older than**, select the time span that meets your needs.
   3. Under **Schedule**, click **Change**.
   4. In the **Job Schedule Properties** dialog box, verify that the **Schedule type** is set to **Recurring**, and under **Frequency**, set **Occurs** to **Daily**.
   5. Under Daily Frequency, specify an off-peak time, and then click **OK**.
   6. In the **Define History Cleanup Task** page, click **Next**.

10. In the **Define Back Up Database (Full) Task** page, create a task to perform a full backup of the INOW6 database by setting the following options.
    1. Click **Databases**, select **These databases**, and then click **INOW6**.
    2. Click **OK**.
    3. In **Back up to**, select your format.
    4. Verify that **Create a backup file for every database** is selected and make any changes to the default backup path.
    5. Verify the **Backup file extension** is **BAK**. The default for full backups is **BAK**.
    6. Make sure that **Verify backup integrity** is selected.
    7. Under **Schedule**, click **Change**.
    8. In the J**ob Schedule Properties** dialog box, verify that the **Schedule type** is set to **Recurring**, and under **Frequency**, set **Occurs** to **Weekly**.
    9. Specify an off-peak day, and then click **OK**.
    10. In the **Define Back Up Database (Full) Task** page, click **Next**.
11. In the **Define Back Up Database (Differential) Task** page, create the differential backup that occurs between full backups by setting the following option.
    1. Click **Databases**, select **These databases**, and then click **INOW6**.
    2. Click **OK**.
    3. In **Back up to**, select your format.
    4. Verify that **Create a backup file for every database** is selected and make any changes to the default backup path.
    5. Verify the **Backup file extension** is **DIFF**. The default for differential backups is **BAK**, but it is better to use different extensions for full and differential backups.
    6. Make sure that **Verify backup integrity** is selected.
    7. Under **Schedule**, click **Change**.
    8. In the J**ob Schedule Properties** dialog box, verify that the **Schedule type** is set to **Recurring**, and under **Frequency**, set **Occurs** to **Daily**.
    9. Under **Daily frequency**, specify an off-peak time, and then click **OK**.
    10. In the **Define Back Up Database (Differential) Task** page, click **Next**.
12. In the **Define Back Up Database (Transaction Log) Tasks** page, you create the transaction log backup task that provides you with the ability to restore your database with the most recent backup if needed. To do this, complete the following steps.
    1. Click **Databases**, select **These databases**, and then click **INOW6**.
    2. Click **OK**.
    3. In **Back up to**, select your format.
    4. Verify that **Create a backup file for every database** is selected and make any changes to the default backup path.
    5. Verify the **Backup file extension** is **TRN**. The default for transaction log backups is **TRN**.

   6. Make sure that **Verify backup integrity** is selected.
   7. Under **Schedule**, click **Change**.
   8. In the **Job Schedule Properties** dialog box, verify that the **Schedule type** is set to **Recurring**, and under **Frequency**, set **Occurs** to **Daily** or whenever you need the transactions backed up.
   9. Under **Daily frequency**, specify your required frequency, and then click **OK**.
   10. In the **Define Back Up Database (Transaction Log) Tasks** page, click **Next**.

13. In the **Select Report Options** page, select the options you want for the reporting maintenance plan actions, either **Write a report to a text file**, or **E-mail report** then click **Next**.
    - If you selected Write a report to a text file, in **Folder location**, leave the default value, or search for another location.
    - If you selected E-mail report, in **E-mail report**, enter an email address.
14. In the **Complete the Wizard** page, verify your choices, and then click **Finish**.
15. After the wizard completes, click **Close**.
16. Modify the newly created **Maintenance Plan** and add a **Maintenance Cleanup Task** for the **Full Backup**, **Differential Backup** and **Transaction Log Backup** jobs.
    1. Right-click the **Maintenance Plan** and select **Modify**.
    2. Click the **Backup Subplan**.
    3. From the **Toolbox** pane, click **Maintenance Cleanup Task** and drag and drop it into the design window below the **Back Up** task.
    4. Click the **Back Up** task and an arrow displays on the bottom of the task box.
    5. Click the arrow and drag it down to connect with the **Maintenance Cleanup Task**.
    6. Double-click the **Maintenance Cleanup Task** and fill out the following parameters.
       1. Under **Delete files of the following type**, select **Backup files**.
       2. Click **Search folder and delete files based on an extension** and in **Folder**, click the ellipse and select the backup directory.
       3. In **File Extension**, enter the file extension that you specified for the backup, such as **BAK**, **DIFF**, or **TRN**.
       4. Under **File age**, click **Delete files based on the age of the file at task run time**, and then set the time span you want.
       5. Click **OK**.
    7. **Save** the **Maintenance Plan** and close the **Maintenance Plan Designer**.

    Note Monitor the maintenance plan jobs over the first few weeks to ensure the backups are taking place as scheduled and that the old backups are being purged according to the retention period defined in the maintenance cleanup tasks. Also, make sure the backup location is large enough to hold all backups and allow for growth based on how fast the database is expected to grow.

Tune the Microsoft SQL Server for the ImageNow Server on Windows

After you install ImageNow and any additional ImageNow agents you want to use, like ImageNow Content Server or ImageNow Fax Agent, you can optionally tune your SQL Server settings to make your database run more efficiently. For some SQL Server installations, the default number of processes is not set high enough to cover the number of users accessing the database through ImageNow. To assess the number of default max worker threads set in SQL Server, do the following:

- Open the SQL Server Management Studio, and run the following statement:

```
sp_configure 'max worker threads';
```

This statement returns a config_value, which is a configurable number of processes that can simultaneously access the database server. You can increase this number to cover the number of users accessing the SQL Server through ImageNow using the following procedure:

1. In the **SQL Server Management Studio**, in the left pane, open your SQL Server group.
2. In the right pane, right-click your server, and then select **Properties**.
3. In the **SQL Server Properties** dialog box, click the **Processors** tab.
4. In **Maximum worker threads**, type a new number for the threads that can accommodate the number of simultaneous ImageNow users.
5. Click **OK**, and then stop and restart the database server as needed.

If you add an agent or increase the num.workers setting in the inserver.ini file or any of the ImageNow agent .ini files, you must re-evaluate the max worker threads setting in SQL Server.

Improving system performance

The following recommendations can increase performance after you have ImageNow running in production. In addition, as your user base gets larger, you can make the following changes to maximize the performance of your system.

You can improve performance of the database maintenance process by streamlining indexes and testing paging allocations used by the SQL Server tempdb database. Other performance-enhancing tasks include defragmenting files and optimizing the scheduling of maintenance tasks. Your performance needs drive the options you select for these settings.

After developing a database maintenance plan, you may find it necessary to update the plan to meet your needs. The following sections provide suggestions to improve your database maintenance processes. For more information about improving your database server, see your SQL Server documentation.

To ensure the integrity of your data, set up a maintenance plan using the recommendations in this document, which is available on the ImageNow Customer Portal. In addition, to avoid disk contention, if you can do so, move the database transaction log to a different drive than the drive where the data files and operating system are located.

In your anti-virus application, disable on-access scanning for the \inserver6 directory, including all subdirectories, and your database. When you use on-access scanning, your virus scanner continually examines the ImageNow Server memory and file system, which can decrease performance. If you move any directory outside of \inserver6, make sure you disable on-access scanning in the new location. You can verify the location of ImageNow directories through the \inserver6\etc\inow.ini file.

Verify that the num.workers setting in the inserver.ini file is set to reflect an accurate number of users. The suggested ratio is to set one thread for every ten users. If you change this setting, make sure you re-evaluate the max worker threads setting in SQL Server, using the information in the "Tune the Microsoft SQL Server for the ImageNow Server on Windows" section of this document.

Managing index integrity

If you find that your database integrity check takes more time than is available during daily, weekly, or monthly execution, you may find it more efficient to select **Exclude indexes** in the **Database Integrity Check** page of the Database Maintenance Plan wizard rather than the suggested selection of **Include indexes**. When you select **Exclude indexes**, the maintenance plan performs the same integrity checks but does not check non-clustered indexes.

A table can contain only one clustered index because the clustered index dictates the order of the physical storage of the data in the table. The primary key in each INOW6 table is clustered. Checking non-clustered indexes is one of the most time-consuming jobs. By setting **Exclude indexes**, you can greatly reduce the time it takes the **Check database integrity** process to run. Although you might lose some integrity-checking benefits with this selection, you can easily rebuild non-clustered indexes if necessary.

Integrity checks can spool to the tempdb when they are running. You can move the tempdb database to its own dedicated disk array so that it can work as efficiently as possible, which increases I/O for tempdb and speeds up the integrity checking process. You can also improve the performance of integrity checks by adding more RAM to your server. The additional RAM increases SQL Server's data cache and can improve overall performance.

Estimate space required on SQL Server for temporary database

Depending on the size of the database being analyzed, the process of checking database integrity takes a lot of time and resources. If this check is required on a large database on SQL Server, and it has not previously been executed, first consider running the following command from the SQL Query Analyzer, by using the ESTIMATEONLY option:

```
DBCC CHECKDB(database name) with ESTIMATEONLY
```

This command estimates how much tempdb space that SQL Server requires for the integrity operation; however, it does not actually run the integrity check. Also, verify that the drive containing your tempdb database provides enough expansion room to cover the entire space needed to run the integrity operation. If the tempdb database is forced to grow and execute the integrity operation at the same time, it slows down your database server. If the tempdb database is not set to auto-grow, you must expand it manually before checking database integrity to avoid running out of space before the integrity check finishes.

Test data and index page allocations

You can test the data and index page allocations in your databases for errors when you select **Check database integrity** and **Include indexes** in the **Database Integrity Check** page. Weigh the benefits of this information against the fact that these checks are resource-intensive and can lower the performance of your database server during the tests. Schedule these checks to run when your database server is less busy.

If you choose the **Perform these tests before backing up the database or transaction log option** in the **Database Integrity Check** page, consider that you will run these tests as a prerequisite step each time before the maintenance plan performs a database or transaction log backup. If your database is large or if you perform frequent database or transaction log backups, running these tests each time might slow database server performance. In general, schedule these integrity tests to run independently from the backups on a weekly or monthly basis.

Defragmenting files

You can speed up most maintenance processes by ensuring that your SQL Server database and log files are defragmented before running a re-index operation. When you keep your database and log files contiguous, the task of rebuilding indexes is faster and requires less I/O resources, resulting in increased overall performance.

Defragmentation is an I/O-intensive process that can negatively impact a busy SQL Server, so schedule this task during off-peak hours. Choose a defragmentation tool to meet your needs. For example, Microsoft Windows provides a defragmentation utility that performs simple, manual defragmentation, but it is not particularly suited for production environments. Instead, consider a third-party tool that offers additional features such as scheduling.

Fragmentation reduces performance in range and table scans because of the amount of data that is generally read during these scans. For example, if the drive head has to search the entire disk to locate the various pages to scan, it can radically slow query response time. In addition, fragmentation significantly reduces performance on RAID devices.

To clarify, rebuilding indexes is not the same function as defragmenting SQL Server’s data files. The indexes are rebuilt within the SQL Server data files, so SQL Server’s physical files are not affected. When you evaluate defragmentation software, assess its capabilities to defragment SQL Server database and log files online. If online defragmentation is not supported, you must stop SQL Server services before running the defragmentation software