

I D E R A

SQL Defrag Manager

Version 3.5



Table of Contents

Table of Contents	2
Find and fix SQL fragmentation hot spots fast	6
Release notes	7
New features	8
3.5 New features.....	8
3.4 New features.....	8
3.3 New features.....	8
Fixed issues.....	10
Fixed issues in SQL Defrag Manager 3.5.....	10
Fixed issues in SQL Defrag Manager 3.4.....	10
Fixed issues in SQL Defrag Manager 3.3.....	10
Known issues.....	12
Known issues in version 3.5	12
Welcome to SQL Defrag Manager	14
What is fragmentation?.....	15
Understanding the different types of fragmentation.....	15
Fragmentation examples.....	16
How does SQL Defrag Manager defragment indexes?	16
Defragmenting an index example	17
How SQL Defrag Manager compacts data.....	18
About interleaved indexes.....	18
What is SQL Defrag Manager?.....	19
How SQL Defrag Manager helps.....	20
Centralized Management	20
Automated Analysis and Increased DBA Productivity.....	20
Continuous Improvement	20
Comprehensive Reporting.....	21

Find answers	22
Document conventions	23
Using SQL Defrag Manager Help	24
Get started	25
Upgrade to this release.....	26
Upgrade checklist	26
Available upgrade paths	26
Perform a typical upgrade	28
Perform a staged upgrade.....	29
Upgrade SQL Defrag Manager in non-trusted domains	30
Copy of Perform a typical upgrade.....	31
Installation and deployment	32
Product components and architecture	33
Product requirements.....	35
Supported installation scenarios	38
How to install SQL Defrag Manager	39
Copy of Product requirements.....	44
Configure your deployment	47
Configure Management Console preferences	48
Configure Email settings for alert notifications.....	49
Recent activity configuration.....	50
Connect the Console to the Repository	52
Register an instance	53
Modify your SQL Server instances list.....	57
Support partitioned indexes	58
Adjust fill factor	59
Manage your licenses.....	60
Cloud for Azure and AW EC2	61
Automate defragmentation using policies.....	62
Accessing policies	62

Creating a policy.....	62
Create a policy.....	63
To get started with the Policy wizard:	63
Policy Wizard - Select the indexes you want to evaluate	64
Policy Wizard - Select the index thresholds and analysis frequency.....	65
Policy Wizard - Select index filter options.....	67
Policy Wizard - Select policy response.....	68
Policy Wizard - Schedule when and how often you want the policy to run.....	70
Policy Wizard - Select which resource checks you want to run before running the policy	72
Policy Wizard - Select what action you want to occur when certain conditions are met.....	74
Policy Wizard - Name the policy.....	75
Edit a policy.....	76
Import a policy	77
Export a policy.....	78
View To Do tasks based on policy results	79
Access the To Do view	79
How partitioned indexes are addressed in the To Do view.....	79
How grouping information helps.....	80
Perform an action listed on this window	80
Perform a manual defragmentation.....	81
Analyze database fragmentation.....	82
Access the Analyze Fragmentation window	82
Selecting the instances and databases to analyze	82
Difference between fast and thorough analysis.....	82
2-Steps analysis.....	83
Canceling an activity.....	83
View analysis results.....	84
Access the Fragmentation Explorer.....	84
About the icons and fragmentation statistics	84

About the Show/Hide filter	85
Use the contextual menu filter option	85
Rediscover Objects	86
Performing actions from the Explorer navigation tree	86
Performing actions from the objects view.....	87
Canceling an activity	88
Optimize performance and space usage.....	89
Access the Defragment and Optimization window	89
Selecting instances and databases to defragment.....	89
Difference between reorganize and rebuild	89
2-Steps defragmentation	90
Canceling an activity	90
Using the Command Line	91
View activity and results.....	92
View alert history	93
Filtering the alert data	93
Grouping information	93
View recent activity	94
Displaying details for an activity	94
About the details.....	94
Grouping information	95
Using this window.....	95
View reports	97
SQL Defrag Manager PDF	98

Find and fix SQL fragmentation hot spots fast

- Automate identification of index fragmentation "hot spots"
- Schedule index defragmentation jobs
- Avoid database contention with system resource pre-checks
- Reduce page splits with index fill factor settings control
- Centrally manage, report and notify instantly

Release notes

SQL Defrag Manager is a powerful solution that automates and optimizes your SQL Server database defragmentation. SQL Defrag Manager improves server performance by detecting fragmentation hot spots and automatically defragmenting them. The automation built into SQL Defrag Manager enables you to implement a "set it and forget it" approach to managing fragmentation on your production SQL Servers.

To get a quick glimpse into the newest features, fixed issues, and known issues in this release of SQL Defrag Manager, review the following sections of the Release Notes:

- [Learn about key new features in this release](#)
- [Review issues fixed by this release](#)
- [See known issues](#)

New features

SQL Defrag Manager provides the following new features:

3.5 New features

- **Support for SQL 2017**
SQL Defrag Manager now supports SQL Server 2017.
- **Auto-discovery**
A new option has been added to policies allowing users to configure auto-discovery of server objects when the policy runs
- **Show a table's schema**
Hovering over a table in the **Targets** section of the **Policy wizard** displays a table's schema as a tool tip.
- **Recent Activity Configuration**
Configure the maximum number of displayed items on the Recent Activity pane. You can find more information [here](#).

3.4 New features

3.4 New features

- **This release does not add extra functionality**

3.3 New features

3.3 New features

- **Import policies**
Allows you to import policies from XML files. [Click here](#) to learn more.
- **Export policies**
Allows you to export policies to XML files. [Click here](#) to learn more.
- **Navigation tree filtering option**
Allows you to filter the items displayed in the SQL Defrag Manager Explorer by specific name or by keywords (using wildcards). [Click here](#) to learn more.
- **"Rediscover Objects" action for databases and tables**
Allows you to quickly Rediscover Objects within a specific database or table. [Click here](#) to learn more.

3.2 New features

- **Supports SQL Server 2014**
SQL Defrag Manager supports the use of SQL Server 2014.

- **Includes a new CLI option that initiates a defragmentation from the command line**

SQL defrag manager 3.2 includes a new command line option that initiates a defragmentation. Use the following CLI option:

```
sqldefragc [[[-rebuild [online]] | -reorg] -instance <instance name>  
-database <database name> | -policy <policy name>]
```

You have two options when using this code. You can:

- Define the defragmentation as follows: specify "rebuild" or "reorg" where "online" is optional for "rebuild"; specify the "instance name"; and specify the "database name".
- Define the defragmentation based on an existing policy by specifying the "policy name".

Fixed issues

Fixed issues in SQL Defrag Manager 3.5

Configuration

SQLDEFR-737 Adjusting the fill factor of an index will now identify the correct schema name to find the index properties. .

SQLDEFR-1025 Unable to Run Autodiscover in SQL Defrag Manager.

SQLDEFR-659 Issues with SQL Defrag Manager sending emails to a list of addresses.

SQLDEFR-1036 Alerts failing to fire.

General

SQLDEFR-731 Databases are no longer listed with an "Unknown" status after a policy has completed successfully.

SQLDEFR-1030 Tables list scrolling back to top after selecting an action.

SQLDEFR-1051 Application closes after clicking **Find fragmented hot spots**.

SQLDEFR-1057 The SQL Defrag Management Console crashes when exporting the results of **Recent Activity**.

SQLDEFR-1058 Issues with columns in exported HTML files, columns names are incomplete and the data in the cells are cut off.

Previous Fixed Issued

Fixed issues in SQL Defrag Manager 3.4

Licensing

- Updated licensing files to resolve a bug related to trial license keys.

Fixed issues in SQL Defrag Manager 3.3

Installation

- The issue causing SQLAuthentication to fail while installing SQL Defrag Manager no longer occurs.

Known issues

IDERA strives to ensure our products provide quality solutions for your SQL Server needs. The following known issues are described in this section. If you need further assistance with any issue, please contact [Support](http://www.idera.com/support) (www.idera.com/support).

Known issues in version 3.5

There are no new known issues in this release

Previous known issues

Phase out SQL Defrag Manager Itanium support

IDERA is phasing out all Itanium support in SQL Defrag Manager 3.0 and all subsequent versions. SQL Defrag Manager 3.0 does not support the Itanium processor architecture and limited technical support is available through November 2012. For more information, see the product requirements.

SQL Server 2000 does not support Thorough analysis

If you select a SQL Server 2000 database and select the **Thorough** analysis type, SQL Defrag Manager performs a Fast analysis for that database.

Small indexes can cause errors in the Recent Activity tab

Small indexes may not meet your fragmentation threshold levels and can cause SQL Defrag Manager to display errors on the Recent Activity tab. When SQL Defrag Manager analyzes and defragments a database, it displays an error message on the Recent Activity tab for each index that it cannot sufficiently defragment. Small indexes often have fragmentation levels that cannot be reduced. By default, SQL Defrag Manager excludes small indexes from the Fragmentation Explorer tab. You can also modify your automation policies to ignore the fragmentation level of small indexes.

Fragmentation level may not be updated

If an analysis process fails, SQL Defrag Manager does not update the fragmentation information displayed on the Fragmentation Explorer tab. SQL Defrag Manager displays that fragmentation information from the last successful analysis.

HTML column headings may be truncated

When you export information from a tab, such as the Scheduled Activity tab, to an HTML file, the column headings may be truncated for narrow columns. To avoid this issue, export the information to an XML file.

SQL Server may not list all instances

When you register instances on a server, SQL Defrag Manager may not list all instances on that server. You can add missing instances by name to register those instances.

Requirements for 64-bit support

SQL Defrag Manager supports the following 64-bit software versions:

- Microsoft Windows Server 2003 SP1
- Microsoft SQL Server 2005 Standard and Enterprise Editions

Disabling OLE can cause a resource check error

A policy configured to use the **CPU Load Percentage (Total Server)** resource check does not work correctly if OLE is disabled on the server.

Non-clustered indexes automatically rebuilt with clustered indexes for SQL Server 2000 users

When a user with SQL Server 2000 performs a rebuild of a clustered index, SQL Server also rebuilds all non-clustered indexes. This issue occurs during all clustered index rebuilds, including when the user accepts a change to the fill factor settings as that feature requires a rebuild of the clustered index.

Partitions on partitioned indexes are not automatically recognized after an upgrade

Users with partitioned indexes and who upgrade from a previous version must use the **Rediscover server objects** feature for SQL Defrag Manager to recognize partitions. To rediscover these objects, right-click the appropriate server name while in Fragmentation Explorer, and then select **Rediscover server objects**.

Welcome to SQL Defrag Manager

IDERA SQL Defrag Manager is a powerful solution that automates and optimizes your SQL Server database defragmentation. SQL Defrag Manager improves server performance and disk I/O by detecting fragmentation hot spots and defragmenting them using the optimal method and schedule.

This wiki helps you understand SQL Defrag Manager characteristics and provides step-by-step help on how to use all its features.

Need help using SQL Defrag Manager? See the following sections:

- [Installation and deployment](#)
- [Configure your deployment](#)
- [Automate defragmentation using policies](#)
- [Perform a manual defragmentation](#)
- [View activity and results](#)

What is fragmentation?

As data is modified in a database, the database and its indexes become fragmented. As indexes become fragmented, ordered data retrieval becomes less efficient and reduces database performance.

Understanding the different types of fragmentation

There are several types of fragmentation that can occur and impact SQL Server performance and space usage. Note that logical order and page density issues exist on tables and indexes within SQL Server. These issues cannot be resolved by operating system level defragmentation tools because the fragmentation exists within the files, rather than at the file level itself.

File fragmentation at the operating system level

When deletes and inserts are performed over time, pages become fragmented as the physical sequence of data pages no longer matches their logical order. This fragmentation happens at the file allocation level and can be addressed with system tools. On larger systems, such as a storage area network (SAN), the disk subsystem automatically maintains low fragmentation levels. If you have a small to medium size system and you do not have a SAN, you should run a system defragmentation tool before addressing logical order and page density fragmentation within SQL Server.

Logical order fragmentation

This issue, also known as external fragmentation within SQL Server, is similar to file fragmentation at the operating system level. When data is deleted, inserted, and modified over time, an index can cause pages to be out of order, where the next logical page is not the same as the next physical page.

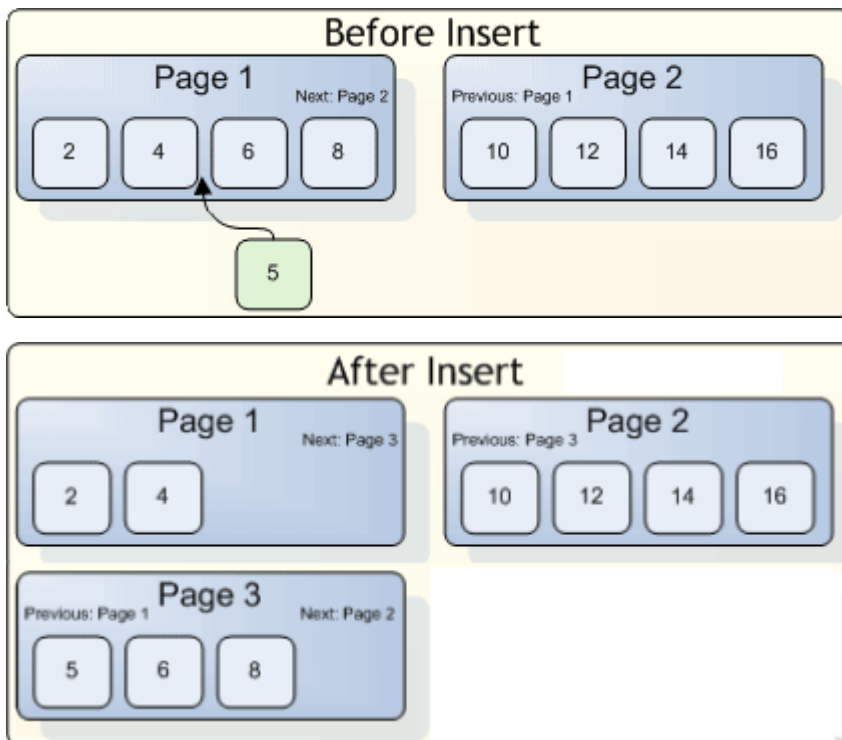
Page density fragmentation

This issue, also known as internal fragmentation, occurs as pages split to make room for information added to a page, there may be excessive free space left on the pages. This extra space can cause SQL Server to read more pages than necessary to perform certain tasks.

SQL Defrag Manager defragments the leaf level of an index so the physical order of the pages matches the left-to-right logical order of the leaf pages. The leaf pages of a clustered index contain the table data. This process improves index scanning performance and all data retrieval activities.

Fragmentation examples

Imagine there are two data pages for a table with a clustered index. The data is ordered and the pages are full as shown in the following figure. A new row with a primary key of "5" needs to be inserted, and since it is a clustered index, the new row is inserted in order. Because the target page is full enough that the new row does not fit, SQL Server splits the page roughly in half and inserts the new data on the new page, as shown in the following figure. Now, the logical order of the index does not match the physical order, and the index has become fragmented.



How does SQL Defrag Manager defragment indexes?

Depending on the policy settings you select, SQL Defrag Manager defragments tables and indexes in one of the following ways:

Rebuild

To rebuild the indexes on tables, the rebuild defragmentation option uses the `ALTER INDEX REBUILD` command for SQL Server 2005 and newer versions. For SQL Server 2000 it uses `DBCC DBREINDEX`. The rebuild operation creates new, contiguous pages. SQL Server 2005/2008 allows the option to Rebuild Online, which allows access to the tables before the operation is finished. However, choosing to rebuild online requires more resources (disk space, CPU, memory), and may slow performance.

Reorganize

To reorder the leaf pages of the index in-place, the reorganize defragmentation option uses the

```
ALTER INDEX REORGANIZE
```

command for SQL Server 2005 and newer versions. For SQL Server 2000 it uses

```
DBCC INDEXDEFRAG
```

. This process is similar to a bubble sort. Although the pages are physically reordered, they may not be contiguous within the data file. This issue can cause interleaved indexes, which need to be rebuilt to store them in contiguous pages.

ⓘ Online index operations are available only in SQL Server Data center, Enterprise, Developer, and Evaluation editions.

Defragmenting an index example

Consider a simplified example of pages after many inserts, updates, and deletes, as shown in the following figure. The page numbering represents the logical sequence of the pages. However, the physical sequence, as shown in the figure from left to right, does not match the logical sequence.

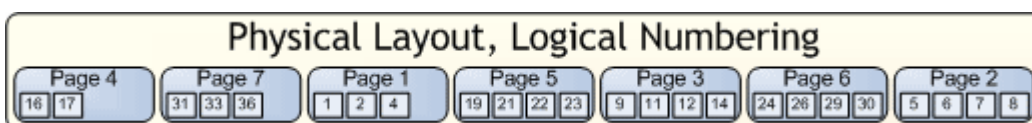
The following figure illustrates multiple passes during the Reorganize defragmentation process, which causes the physical pages to be reordered by having the first logical page swapped with the first physical page, and then the second logical page swapped with the second physical page, and so on.

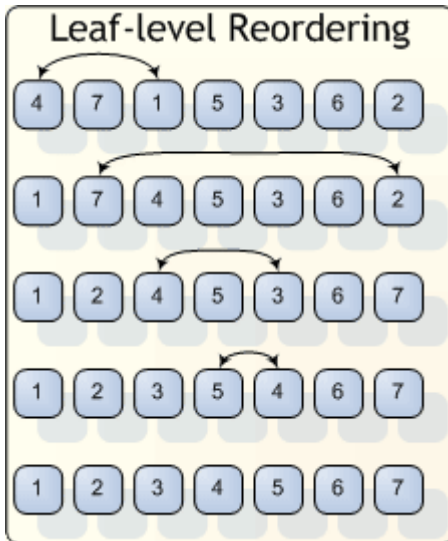
On the first pass, SQL Server finds the first physical page (4) and the first logical page (1). SQL Server then swaps these pages in a discrete transaction.

On the second pass, SQL Server swaps the next physical page (7) with the next logical page (2).

On the third pass, SQL Server swaps the next physical page (4) with the next logical page (3).

On the fourth pass, SQL Server swaps the next physical page (5) with the next logical page (4). Sorting is now complete, as all the remaining physical pages match their logical positions.





How SQL Defrag Manager compacts data

In addition to reordering the leaf pages of the index, SQL Defrag Manager compacts the data in the pages using the original fill factor value specified for the table and then removes any empty pages. Consider the following conditions related to this compaction phase:

- Compaction is completely skipped if the Inhibit Page Locks property is set for the index.
- There are various algorithms built into the compaction phase to stop unnecessary work. For example, if the first page in the index is empty and all the other pages are full, SQL Server does not repeatedly move all the data forward one page.
- SQL Server compacts pages back to the fill factor value defined for the index. Make sure this value is not set too high. For more information, see the SQL Server documentation.
- If a lock cannot be obtained on a page during the compaction phase of DBCC INDEXDEFRAG, SQL Server skips that page.

About interleaved indexes

Interleaving occurs when an index extent, which is a group of eight index pages, is not physically contiguous because an extent for another index is intermingled with it. This condition can happen even when there is no logical fragmentation in the index. Although the pages may be physically and logically ordered, they are not necessarily contiguous. Switching between extents can impact performance as data access is inefficient. To resolve this issue, use SQL Defrag Manager to rebuild the indexes to store them in contiguous pages and reduce the need to switch between extents.

What is SQL Defrag Manager?

IDERA SQL Defrag Manager is a powerful solution that automates and optimizes your SQL Server database defragmentation. SQL Defrag Manager improves server performance and disk I/O by detecting fragmentation hot spots and defragmenting them using the optimal method and schedule.

SQL Defrag Manager provides comprehensive database fragmentation management, analysis, and resolution through the following critical features:

- Central enterprise console with three-tiered agentless architecture
- Automatic, semi-automatic, and manual management modes
- Detailed scheduling and management at multiple object levels
- Automated analysis and remediation
- Comprehensive reporting and notification
- Easy installation and use

How SQL Defrag Manager helps

SQL Defrag Manager helps you obtain unparalleled speed and optimized performance across your SQL Server environment. Performing proper defragmentation is time consuming, often risky, and requires complex analysis and in-depth understanding. SQL Defrag Manager solves these issues by providing a proactive solution that safely defragments your databases and continuously optimizes performance, avoiding future issues related to database delays.

SQL Defrag Manager remediates both logical order and page density fragmentation. Both types of fragmentation cause slow response times and can degrade performance up to the point that SQL Server discards its indexes and users table scans for data retrieval.

Diligent DBAs often employ mass defragmentation on their SQL Servers through self-made scripts or maintenance plans. Unfortunately, this approach falls short because no two tables or indexes have identical tuning requirements. Using this mass approach can also be problematic because it often acquires unnecessary table and index locks during defragmentation, which can cause deadlocks.

SQL Defrag Manager incorporates proven DBA experience and knowledge in a comprehensive, automated, and effective solution. SQL Defrag Manager resolves and comprehensively optimizes your SQL Server performance by managing your fragmentation issues.

Centralized Management

The SQL Defrag Manager Console provides a real-time window into fragmentation levels, and the ability to manage and control automated analysis and defragmentation activity for databases across your enterprise.

Automated Analysis and Increased DBA Productivity

Fragmentation analysis and defragmentation are complicated, time-consuming tasks. SQL Defrag Manager improves productivity for DBAs of all experience levels. Automated detection, analysis, and display of fragmentation hot spots allow you to assess many key items to pinpoint highly fragmented areas that severely hinder performance.

Continuous Improvement

Increased fragmentation of databases over time translates into slower response times, degraded performance, and excessive resource consumption. SQL Defrag Manager keeps your databases running at peak performance and eliminates costly application downtime by automatically correcting problems due to fragmentation.

Comprehensive Reporting

Comprehensive reporting of analysis and defragmentation activity provides both DBAs and managers with the information they need to ensure database performance is continuously optimized.

Find answers

This documentation set includes a comprehensive online Help system as well as additional resources that support you as you install and use the product. You can also search the IDERA Solutions knowledge base, available at the [IDERA Customer Service Portal](#).

Document conventions

IDERA documentation uses consistent conventions to help you identify items throughout the printed online library.

Convention	Specifying
Bold	Window items
<i>Italics</i>	Book and CD titles Variable names New terms
Fixed Font	File and directory names Commands and code examples Text typed by you
Straight brackets, as in [value]	Optional command parameters
Curly braces, as in {value}	Required command parameters
Logical OR, as in value 1 value 2	Exclusively command parameters where only one of the options can be specified

Using SQL Defrag Manager Help

The IDERA wiki includes a comprehensive online Help system as well as additional resources that support you as you install and use IDERA products. You can also search multiple IDERA support solutions, available at www.idera.com/support/faq

Additionally, IDERA helps you by providing:

- 24/7 technical support for critical issues.
- Availability to report cases and access a web-based customer portal for update status.
- Access to our [Knowledge center](#) where you can find FAQs, How To's, Best Practices, and Webcasts.

This wiki includes the following Web browser minimum requirements:

- Internet Explorer 8.0
- Mozilla Firefox 4
- Google Chrome 6

You can access the IDERA SQL Defrag Manager Help system through the **Help** icon on the top right section of your window or by pressing F1 on the section where you need more information.

You can print a help topic from the wiki using the Print function in your browser.

Get started

SQL Defrag Manager is a unique SQL Server index defragmentation solution that automates the time-consuming process of finding fragmented indexes based on parameters that you define within a policy for the targeted database. It gives you the flexibility to run the defragmentation utility immediately or during off-peak hours, all through an easy-to-use graphical user interface. There are no manual scripts required. SQL Defrag Manager improves server performance by ensuring that indexes are defragmented, which helps SQL server applications run faster and frees DBAs to perform other tasks.

Use the following checklist to get started using SQL Defrag Manager.

<input checked="" type="checkbox"/>	Follow these steps ...
<input type="checkbox"/>	Register the SQL Server instances you want to defragment.
<input type="checkbox"/>	Analyze your SQL Servers to find fragmented hot spots.
<input type="checkbox"/>	Perform a manual defragmentation of any databases needing reorganization or rebuilding.
<input type="checkbox"/>	Create policies to automate your defragmentation.

Upgrade to this release

Upgrading SQL Defrag Manager allows you to take advantage of the [new features](#) available in the latest version.

Upgrade checklist

To successfully upgrade your Microsoft SQL Server environment to this build, complete the procedures outlined in the following table. These procedures support upgrades from SQL Defrag Manager versions 2.0 or later.

<input checked="" type="checkbox"/>	Follow these steps
<input type="checkbox"/>	Ensure the computers on which you want to upgrade SQL Defrag Manager meet or exceed the product requirements for SQL Defrag Manager. For example, ensure .NET Framework 4.0 or later is running on the target computer.
<input type="checkbox"/>	Ensure your Windows logon account has local administrator permissions on the computers you intend to upgrade.
<input type="checkbox"/>	Review the product architecture .
<input type="checkbox"/>	Review the supported installation scenarios and identify the appropriate upgrade path for your environment.
<input type="checkbox"/>	Close all open applications on the computers you intend to upgrade.
<input type="checkbox"/>	Upgrade your SQL Defrag Manager installation.

Available upgrade paths

Because you can install each component separately on different computers, the type of upgrade you need to perform depends on your environment. The following table describes the conditions under which you would follow a typical or staged upgrade path.

Environment description	Recommended path	Why
Simple environment, where the SQL Defrag Manager Console, the Repository, and the Management Service all reside on the same computer	Typical	A simple environment can be upgraded through the setup program.
Distributed environment, where each SQL Defrag Manager component resides on a different computer.	Staged	A distributed environment requires a staged upgrade to maintain continuity while each component is upgraded.
Multiple Management Consoles deployed to different computers.	Staged	A SQL Defrag Manager installation with multiple Management Consoles requires a staged upgrade to maintain continuity while each component is upgraded.
Change control policies that require multi-phased upgrades.	Staged	An environment with stringent change control policies requires a multi-phased upgrade in order to test each updated component thoroughly before moving on to the next step.

Perform a typical upgrade

Use the Typical install to upgrade your SQL Defrag Manager installation if you have centralized SQL Defrag Manager deployment or are upgrading from a trial installation.

Understanding typical environments

In a typical environment, the SQL Defrag Manager management components are installed on the same physical computer. These components include:

- Repository database
- Management Service
- Management Console

Performing a typical upgrade

You can easily complete a typical upgrade during off-hours.

To perform a typical upgrade:

1. Run the installation file.
2. The installer detects any previous version of SQL Defrag Manager and prompts a confirmation dialog window to perform the upgrade. Click **Yes** to launch the upgrade wizard.
3. To start the upgrade process click **Next**
4. Specify an authentication method to upgrade the SQL Defrag Manager Repository on the selected instance. Type in the the Login ID and password if you want to use Microsoft SQL Server Authentication as the connection credentials to upgrade the SQL Defrag Manager Repository.
5. Click **Next** to perform the upgrade.

i If you use remote consoles, you must upgrade the Management Service and Repository before you upgrade the remote consoles. If you use policies to automate your defragmentation operation, start the SQL Defrag Manager Console to synchronize the jobs associated with each policy. This synchronization should happen automatically.

Perform a staged upgrade

Use a staged process to upgrade your SQL Defrag Manager installation if you have a distributed environment or you need to adhere to change control policies.

Understanding distributed environments

A distributed environment consists of the SQL Defrag Manager management components running on different physical computers. The management components are:

- Repository database
- Management Service
- Management Console

The time scale over which you choose to perform a staged upgrade depends on the size of your SQL Server environment and your corporate change control policies. For example, you may choose to perform one of the following steps per day or per week; however, the order in which you perform them should remain the same.

Performing a staged upgrade

Use this process if your SQL Defrag Manager components are installed on different computers.

To perform a staged upgrade:

1. Upgrade the Management Service and Repository database to the latest version by using the setup program to perform a Custom install.
2. Upgrade your Management Console installations to the latest version.
3. If you use policies to automate your defragmentation operation, start the SQL Defrag Manager Console to synchronize the jobs associated with each policy. This synchronization should happen automatically.

Upgrade SQL Defrag Manager in non-trusted domains

SQL Defrag Manager 2.0 and later support seamless integration and communications between trusted and non-trusted domains. If you want to take advantage of this feature, use the following instructions to upgrade your SQL Defrag Manager 2.0 or earlier environment.

i To upgrade SQL Defrag Manager 2.0 or later to the current release version, follow the standard [Typical](#) or [Staged](#) upgrade instructions.

Upgrading an island deployment

This deployment consists of:

- A complete SQL Defrag Manager installation in each non-trusted domain
- A single centralized installation that manages SQL Server instances in your trusted domain

If you use remote consoles, you must upgrade the Management Service and Repository before you upgrade the remote consoles.

To upgrade this deployment:

1. Decide how many SQL Defrag Manager Consoles you need to keep in each domain. Maintaining a Management Console in a non-trusted domain is not required.
2. Identify which domain you want to host your new centralized SQL Defrag Manager deployment.
3. Ensure your Management Service account has the [appropriate permissions](#).
4. Uninstall the SQL Defrag Manager components from the domains you do not intend to use to host SQL Defrag Manager. Remember to preserve the Management Consoles you identified in Step 1.
5. [Upgrade the SQL Defrag Manager management components](#) that were previously deployed to the selected domain.
6. Run the setup program to upgrade all remaining Management Consoles.
7. If you use policies to automate your defragmentation *operation*, start the SQL Defrag Manager Console to synchronize the jobs associated with each policy. This synchronization should happen automatically.
8. In each Management Console, verify that:
 - It can connect to the upgraded SQL Defrag Manager Repository
 - It is using the correct Management Service

Copy of Perform a typical upgrade

Use the Typical install to upgrade your SQL Defrag Manager installation if you have centralized SQL Defrag Manager deployment or are upgrading from a trial installation.

Understanding typical environments

In a typical environment, the SQL Defrag Manager management components are installed on the same physical computer. These components include:

- Repository database
- Management Service
- Management Console

Performing a typical upgrade

You can easily complete a typical upgrade during off-hours.

To perform a typical upgrade:

1. Run the installation file.
2. The installer detects any previous version of SQL Defrag Manager and prompts a confirmation dialog window to perform the upgrade. Click **Yes** to launch the upgrade wizard.
3. To start the upgrade process click **Next**
4. Specify an authentication method to upgrade the SQL Defrag Manager Repository on the selected instance. Type in the the Login ID and password if you want to use Microsoft SQL Server Authentication as the connection credentials to upgrade the SQL Defrag Manager Repository.
5. Click **Next** to perform the upgrade.

i If you use remote consoles, you must upgrade the Management Service and Repository before you upgrade the remote consoles. If you use policies to automate your defragmentation operation, start the SQL Defrag Manager Console to synchronize the jobs associated with each policy. This synchronization should happen automatically.

Installation and deployment

You can install and deploy SQL Defrag Manager in any sized environment.

- Learn about the [product components and architecture](#).
- Review the [hardware, software, and permissions requirements](#).
- Check the [supported installation scenarios](#).
- View trial [installation instructions](#).

Product components and architecture

SQL Defrag Manager provides a robust, easy-to-use SQL Server solution. Behind a friendly user interface, SQL Defrag Manager offers a unique architecture that is both flexible and extremely powerful. SQL Defrag Manager fits your environment, no matter how simple or complex.

Product components

SQL Defrag Manager consists of a light, unobtrusive architecture that easily runs in your SQL Server environment with minimal configuration. All SQL Defrag Manager components run outside and separate from SQL Server processes. SQL Defrag Manager does not add to or modify any of your native SQL Server files or services.

SQL Defrag Manager Console

The SQL Defrag Manager Console is the interface you use to configure SQL Defrag Manager, view analysis and defragmentation results, schedule automation activities, and perform other related functions.

SQL Defrag Manager Repository

The SQL Defrag Manager Repository is a central storage area for configuration information, as well as fragmentation data and statistics collected by the Management Service.

Management Service

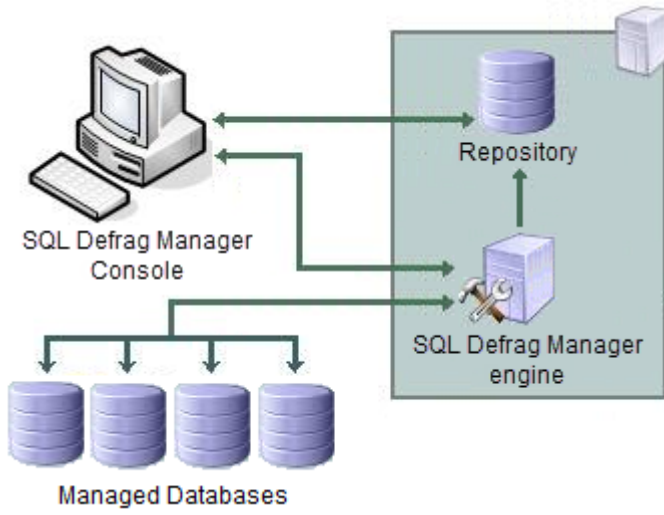
The SQL Defrag Manager Management Service gathers fragmentation data from your managed SQL Server instances and stores the information in the SQL Defrag Manager Repository database. The Management Service also controls scheduling activities.

Managed instances

The managed instances are the SQL Server instances you registered with SQL Defrag Manager. SQL Defrag Manager periodically analyzes and optimizes these instances based on the management settings you specified. SQL Defrag Manager stores this collected data in the SQL Defrag Manager Repository and displays the data in the SQL Defrag Manager Console.

Product architecture

The following diagram illustrates the components of the SQL Defrag Manager architecture.



Product requirements

You can easily and quickly install SQL Defrag Manager on any computer that meets or exceeds the following hardware, software, and permission requirements. Consider the following requirements when installing SQL Defrag Manager in a typical environment.

General product requirements

All SQL Defrag Manager components require the following software:


- Microsoft .NET Framework version 4.0
- MDAC 2.8 or later
- Windows Installer 3.1 or later

Typical installation

A typical installation sets up all SQL Defrag Manager components on the same computer. The following table lists the requirements for a typical installation.

Type	Requirement
CPU	2 GHz
Memory	2 GB
Hard disk space	30 GB for the Repository
Operating system	One of the following operating systems: <ul style="list-style-type: none"> • Windows Server 2008 R2 • Windows Server 2012 • Windows Server 2012 R2 • Windows Server 2016 • Windows 7 • Windows 8 • Windows 10

Type	Requirement
Microsoft SQL Server (Repository)	One of the following versions of Microsoft SQL Server: <ul style="list-style-type: none"> • SQL Server 2005 Standard or Enterprise • SQL Server 2008 R2 Standard or Enterprise • SQL Server 2012 Standard or Enterprise • SQL Server 2014 Standard or Enterprise • SQL Server 2016 Standard or Enterprise • SQL Server 2017 Standard or Enterprise for Windows and Linux.
Microsoft SQL Server (Managed Server)	One of the following versions of Microsoft SQL Server: <ul style="list-style-type: none"> • SQL Server 2000 Standard or Enterprise • SQL Server 2005 Standard or Enterprise • SQL Server 2008 R2 Standard or Enterprise • SQL Server 2012 Standard or Enterprise • SQL Server 2014 Standard or Enterprise • SQL Server 2016 Standard or Enterprise • SQL Server 2017 Standard or Enterprise for Windows and Linux.



Console-only installation

A Console-only installation installs the SQL Defrag Manager Console. The Console-only installation assumes that a full installation already resides on another computer. The following table lists the requirements for a Console-only installation.

Type	Requirement
CPU	2 GHz
Memory	2 GB

Type	Requirement
Operating system	One of the following operating systems: <ul style="list-style-type: none"> • Windows Server 2008 R2 • Windows Server 2012 • Windows Server 2012 R2 • Windows Server 2016 • Windows 7 • Windows 8 • Windows 10

Management Service permission requirements

The Windows user account used by the Management Service requires the following permissions to access the SQL Server instances you want to manage.

Type	Requirement
Windows permissions	Local Administrator permissions on the computer hosting the Management Service.
SQL Server privileges	A member of the sysadmin fixed server role in the SQL Server instance hosting the Repository and on each managed SQL Server instance.
Ports	SQL Defrag Manager service uses port TCP 5280.

Supported installation scenarios

You can install and deploy SQL Defrag Manager to meet your unique defragmentation and SQL Server environment needs.

Typical environment

A typical SQL Defrag Manager implementation scenario includes the following installations:

- Management Console on your workstation
- Repository and Management Service on a SQL Server instance

Non-trusted environment

You can install and configure SQL Defrag Manager to defragment SQL Server databases running in non-trusted domains or workgroups.

This configuration includes the following installations:

- Management Console on your workstation in a trusted or non-trusted domain.
- Repository and Management Service on a SQL Server instance in a trusted or non-trusted domain.

i When deploying SQL Defrag Manager to a non-trusted domain, specify an account with sysadmin fixed role rights for the Management Service account and ensure that SQL Authentication is enabled on each SQL Server instance where a SQL Defrag Manager component is installed.

How to install SQL Defrag Manager

This procedure guides you through a typical install of SQL Defrag Manager. This process installs all SQL Defrag Manager components on one system. Use this procedure for first-time installations and evaluation installations.

Before you begin the installation process, ensure you review:

- [Product components and architecture](#)
- [Product requirements](#)

Check Ports Availability

Make sure SQL Defrag Manager service can use port TCP 5280 for listening.

Start the setup program

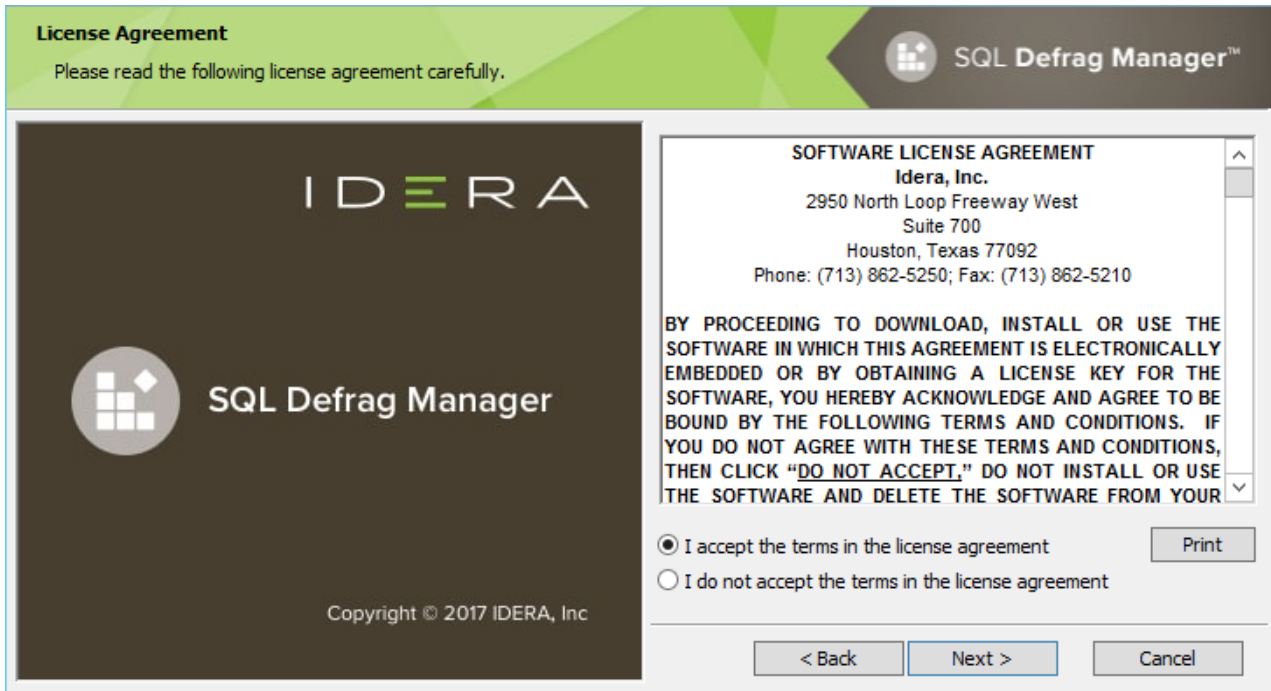
Before you install IDERA SQL Defrag Manager, be sure to have:

- A system that meets or exceeds the [product requirements](#).
- A local administrator account on the system you are installing the product.

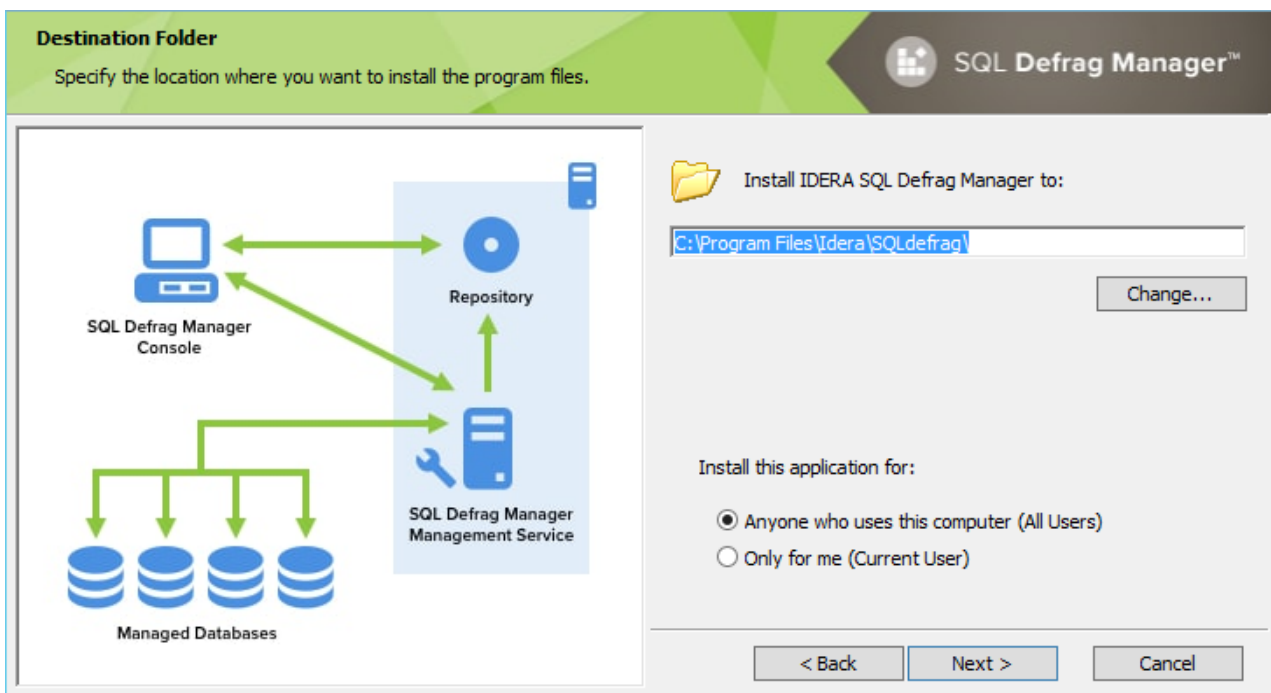
1. On the IDERA SQL Defrag Manager setup window, click **Next** to continue with the installation.



2. Read the License Agreement. If you agree, click **I accept the Terms and Conditions** and then click **Next**.

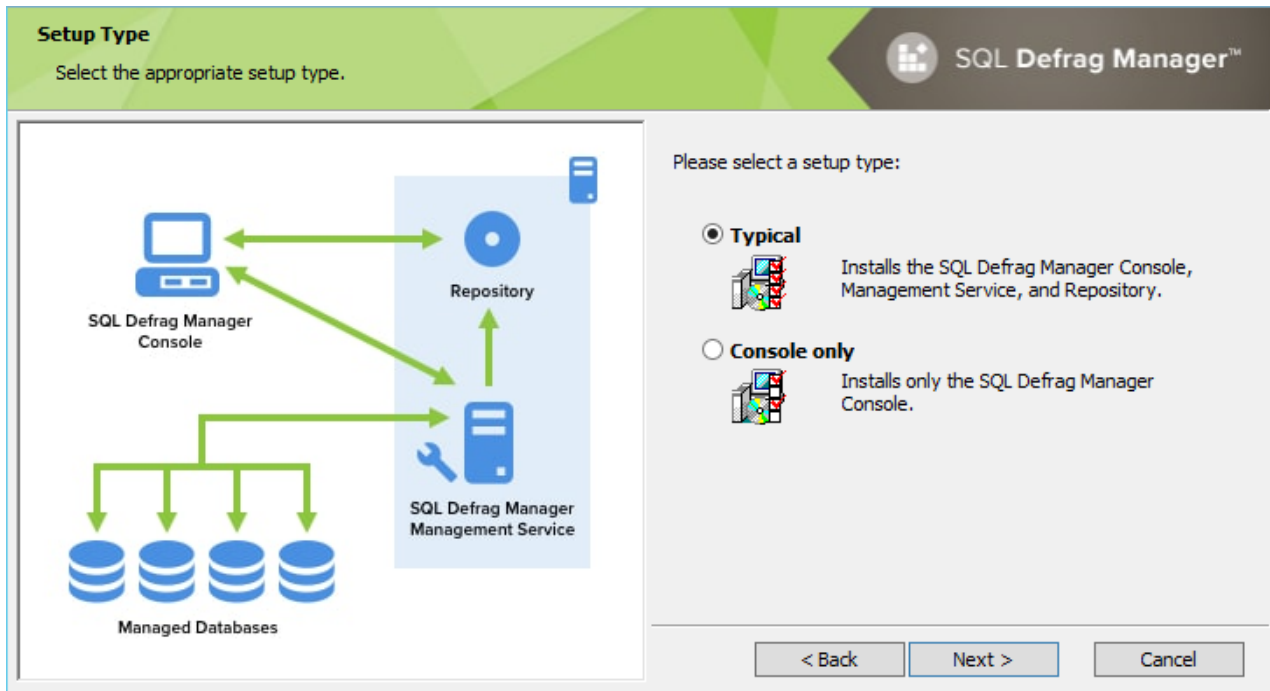


3. To use the default installation settings, click **Next**. To use a different path, click **Change**.



4. Choose a setup type. For your first install, we recommend using the **Typical** setup. This type ensures that you install and configure all required IDERA SQL Defrag

Manager components, so you can immediately begin using the product in your environment.



5. To select an instance for the SQL Defrag Manager Repository, type in the name of the instance or use the drop-down option to select an instance from the available local instances and click **Next**.

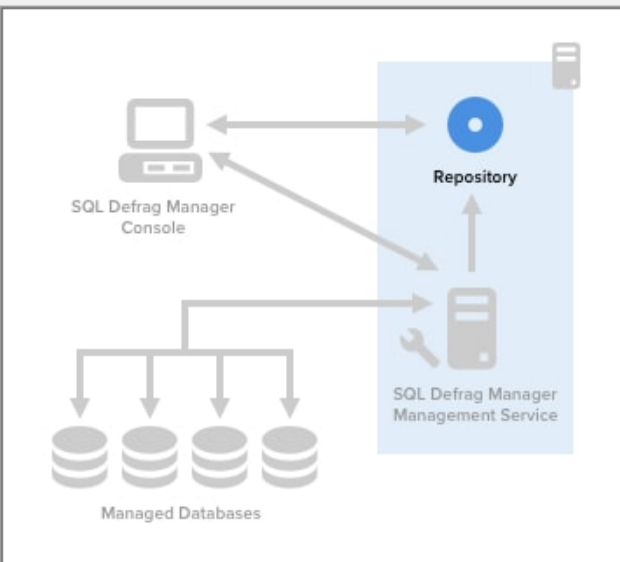
i The SQL Defrag Manager Repository stores all collected activity and defragmentation information, which you can then access in the SQL Defrag Manager Console. SQL Defrag Manager can connect to the selected SQL Server instance using the credentials of your current Windows login account or a SQL Server login. For your first install, we recommend using your current login account credentials.

To specify a SQL Server login:

- a. Select **Use Microsoft SQL Server authentication** .
- b. Specify the credentials of a login with sysadmin privileges on that instance, and then click **OK** .
- c. Click **Next** .

Install the SQL Defrag Manager Repository

Specify the installation options for the SQL Defrag Manager Repository



Specify the SQL Server instance that will host the SQL Defrag Manager Repository. Type in the name of the instance in the field below or use the drop-down option to select an instance from the available local instances.

SQL Server instance:

(local) ▾

Connection Credentials:
You can use SQL Server authentication to create the Repository. By default, the Setup Wizard uses your Windows logon credentials.

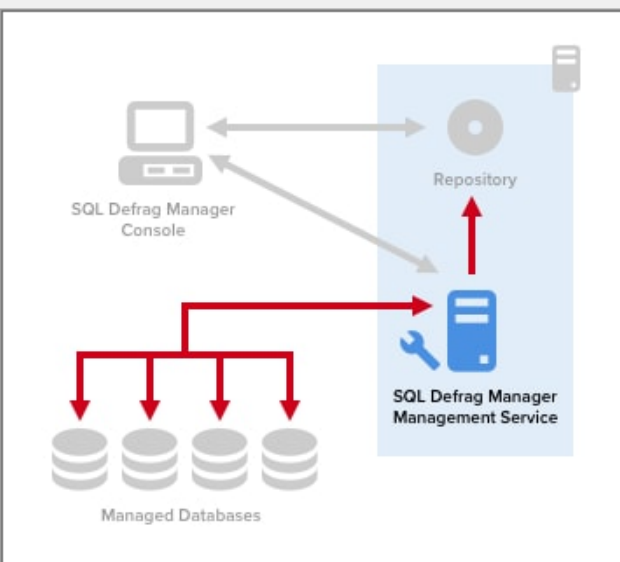
Use Microsoft SQL Server Authentication Change

< Back Next > Cancel

6. The SQL Defrag Manager service uses a Windows account to connect to the SQL Defrag Manager Repository. For your first install, we recommend using local system account for the Management Service account. Type in the appropriate credentials in the requested format (Domain\Username and password) and click **Next**.

Management Service Account

Specify the account the Management Service will run under.



The Management Service runs as the Windows credentials you specify. SQL Defrag Manager uses this account to connect to the SQL Defrag Manager Repository and all managed instances.

Specify the service account:

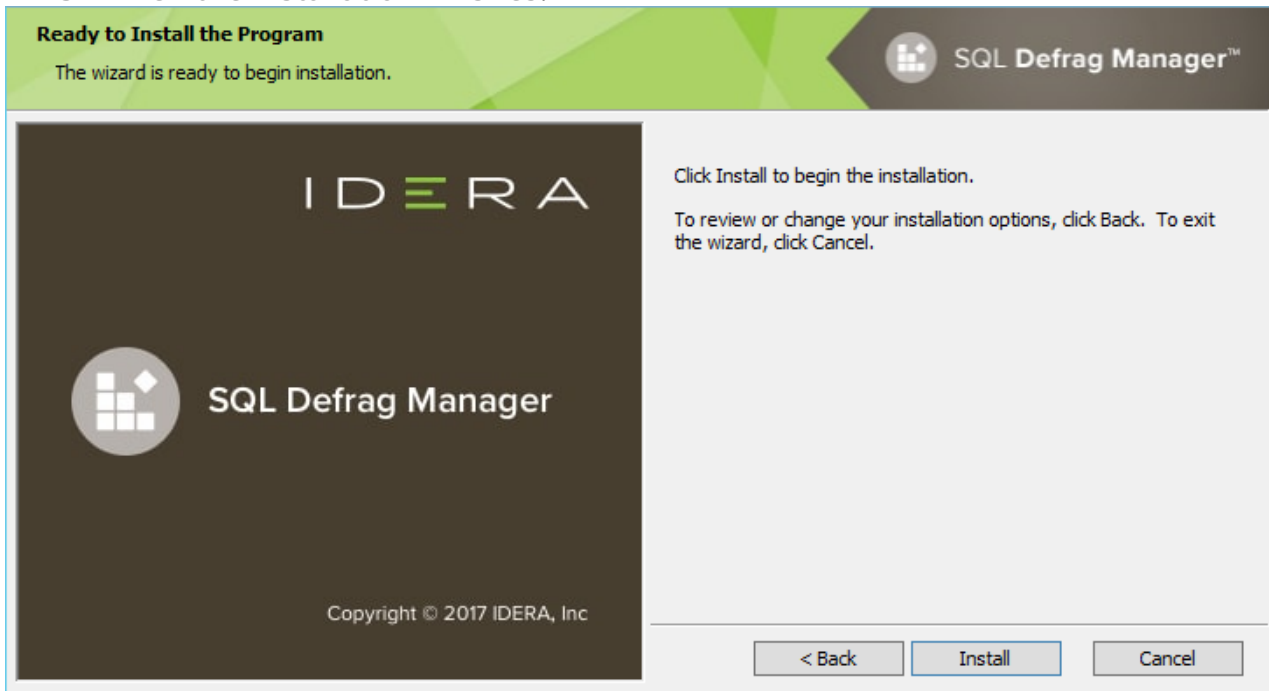
Domain\Username:

Password:

The Logon as a Service right will be granted to this account.

< Back Next > Cancel

7. Click **Install** to perform the installation, click **Back** to review your settings. Click **Finish** when the installation finishes.



Copy of Product requirements

You can easily and quickly install SQL Defrag Manager on any computer that meets or exceeds the following hardware, software, and permission requirements. Consider the following requirements when installing SQL Defrag Manager in a typical environment.

General product requirements

All SQL Defrag Manager components require the following software:

- Microsoft .NET Framework version 4.0
- MDAC 2.8 or later
- Windows Installer 3.1 or later

Typical installation

A typical installation sets up all SQL Defrag Manager components on the same computer. The following table lists the requirements for a typical installation.

Type	Requirement
CPU	2 GHz
Memory	2 GB
Hard disk space	30 GB for the Repository
Operating system	One of the following operating systems: <ul style="list-style-type: none"> • Windows Server 2008 R2 • Windows Server 2012 • Windows Server 2012 R2 • Windows Server 2016 • Windows 7 • Windows 8 • Windows 10

Type	Requirement
Microsoft SQL Server (Repository)	One of the following versions of Microsoft SQL Server: <ul style="list-style-type: none"> • SQL Server 2005 Standard or Enterprise • SQL Server 2008 R2 Standard or Enterprise • SQL Server 2012 Standard or Enterprise • SQL Server 2014 Standard or Enterprise • SQL Server 2016 Standard or Enterprise • SQL Server 2017 Standard or Enterprise for Windows and Linux.
Microsoft SQL Server (Managed Server)	One of the following versions of Microsoft SQL Server: <ul style="list-style-type: none"> • SQL Server 2000 Standard or Enterprise • SQL Server 2005 Standard or Enterprise • SQL Server 2008 R2 Standard or Enterprise • SQL Server 2012 Standard or Enterprise • SQL Server 2014 Standard or Enterprise • SQL Server 2016 Standard or Enterprise • SQL Server 2017 Standard or Enterprise for Windows and Linux.

Console-only installation

A Console-only installation installs the SQL Defrag Manager Console. The Console-only installation assumes that a full installation already resides on another computer. The following table lists the requirements for a Console-only installation.

Type	Requirement
CPU	2 GHz
Memory	2 GB

Type	Requirement
Operating system	One of the following operating systems: <ul style="list-style-type: none"> • Windows Server 2008 R2 • Windows Server 2012 • Windows Server 2012 R2 • Windows Server 2016 • Windows 7 • Windows 8 • Windows 10

Management Service permission requirements

The Windows user account used by the Management Service requires the following permissions to access the SQL Server instances you want to manage.

Type	Requirement
Windows permissions	Local Administrator permissions on the computer hosting the Management Service.
SQL Server privileges	A member of the sysadmin fixed server role in the SQL Server instance hosting the Repository and on each managed SQL Server instance.
Ports	SQL Defrag Manager service uses port TCP 5280.

Configure your deployment

After initially installing and setting up SQL Defrag Manager, there are several tasks you might want to do in order to further customize and streamline your install.

- [Configure Management Console preferences](#)
- [Configure Email settings for alert notifications](#)
- [Connect the Console to the Repository](#)
- [Register an instance](#)
- [Modify your SQL Server instances list](#)
- [Manage your licenses](#)

Configure Management Console preferences

SQL Defrag Manager allows you to modify many of the default settings of the application, and you can change your Console preferences at any time. Reconfiguring these preferences allows you to modify settings in the following categories:

- Alerts
- Recent activity
- Repository
- User Experience

Configuring Email alert notifications

You can [configure email settings](#) in SQL Defrag Manager to send Email alert notifications regarding fragmentation using the Default Email Configuration window. To open the Default Email Configuration window, click **Email Configuration** on the SQL Defrag Manager command menu of the SQL Defrag Manager Console.

Configuring recent activity

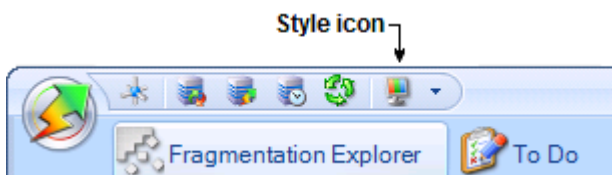
By default, SQL Defrag Manager only displays the last 1000 actions on the Recent Activity pane. Starting with version 3.5 users can [configure this value](#) in the SQL Defrag Manager Console or through the command line.

Changing the SQL Server Repository

You can [change the SQL Server repository](#) using the SQL Defrag Manager Console. To open the Connect to Repository window, click the **Connect** icon at the top left edge of the SQL Defrag Manager Console.

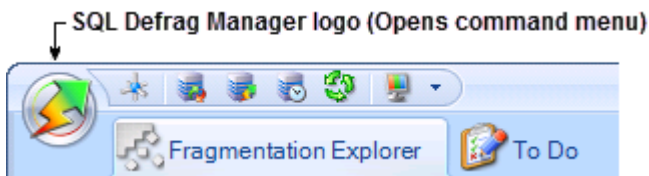
Changing user experience settings

The SQL Defrag Manager Console provides multiple styles that change the appearance of the Console. You can choose which style you prefer to use. To select a style, click the **Style** icon at the top left edge of the SQL Defrag Manager Console, as shown in the following image.



Configure Email settings for alert notifications

You can enable SQL Defrag Manager to send Email alert notifications regarding fragmentation. To access the Default Email Configuration window, click **Email Configuration** on the SQL Defrag Manager command menu of the SQL Defrag Manager Console. To display the command menu, click the SQL Defrag Manager icon in the top left corner of the SQL Defrag Manager Console, as shown in the following image:



If you enable Email alert notifications, you can configure how the Email appears in your Inbox.

SMTP Server Configuration

Enter the address and the port number of your SMTP server. Select if you want to send Email using HTML formatting or not.

Logon Information

Select if your server requires authentication to send Email messages and enter a valid **User Name** and **Password**.

Sender Information

Enter the name and Email address you want to appear in the **From** field of the Email alert notifications.

Default Email Information

Enter one or more Email addresses to send alert notifications to.

✓ Click **To...** to select from a list of previously registered Email addresses.

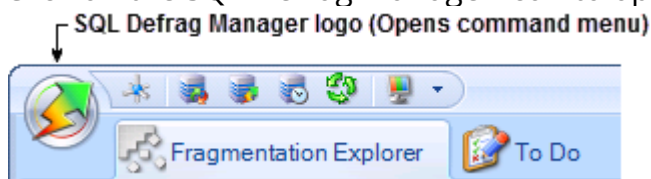
Recent activity configuration

By default, SQL Defrag Manager only displays the last 1000 actions on the Recent Activity pane. Starting with version 3.5 users can change this value in the SQL Defrag Manager Console or through the command line.

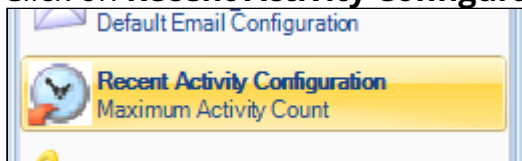
Changing the value on the SQL Defrag Manager Console

To access the **Recent Activity Configuration** dialog follow these steps:

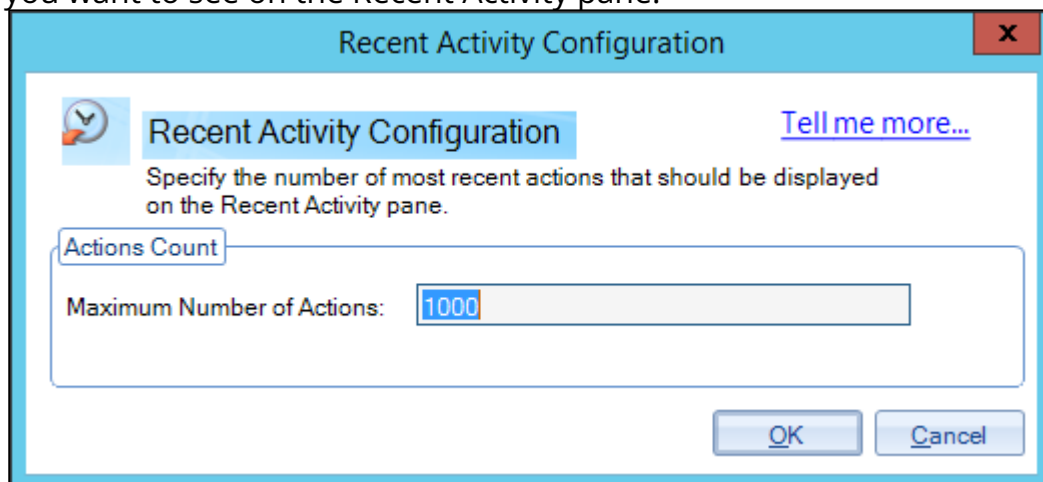
1. Click on the SQL Defrag Manager icon to open the **Command Menu**.



2. Click on **Recent Activity Configuration**.



3. On the **Recent Activity Configuration** dialog set the number of recent actions you want to see on the Recent Activity pane.



Changing the value on the Command line

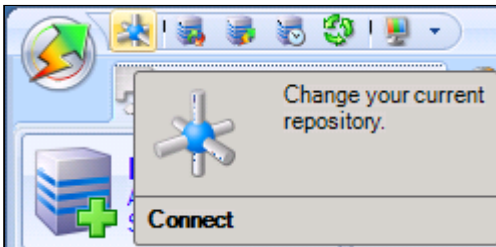
Open the command line and use the following command.

```
sqldefragc.exe -set -maxIndexes <value>
```

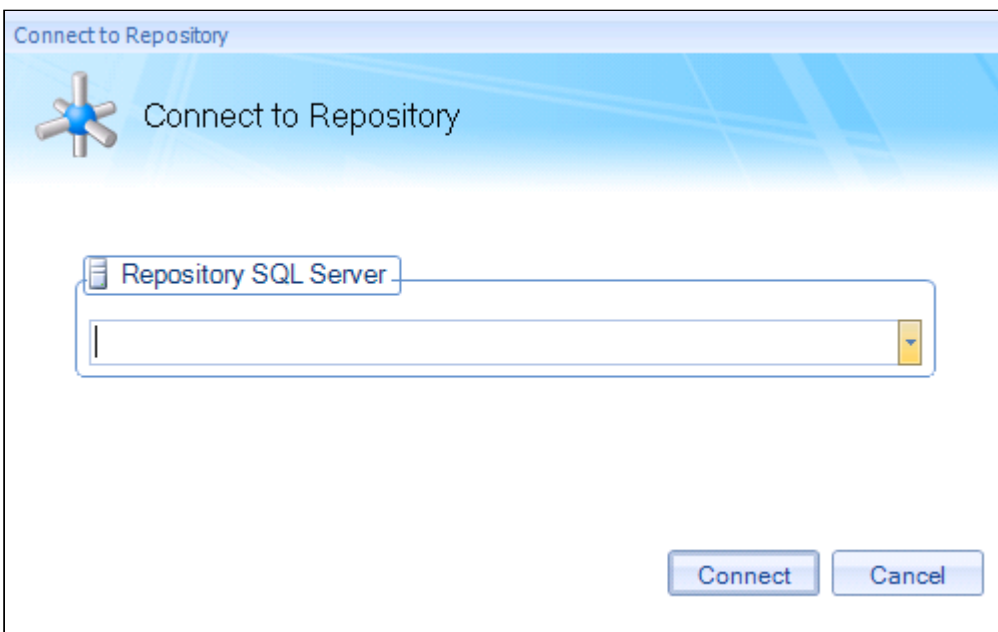
Connect the Console to the Repository

Access the Connect to Repository window

To open the Connect to Repository window, click the **Connect** icon along the top left edge of the SQL Defrag Manager Console, as shown in the following image.



SQL Defrag Manager displays the **Connect to Repository** configuration window. You can type in the name of the SQL Server instance you want to host the SQL Defrag Manager Repository or select one from the drop-down list of identified instances.



The Connect to Repository window allows you to specify the SQL Server instance where the SQL Defrag Manager Repository resides. Specify the instance in the following format:

```
server-name\instance-name
```

Register an instance

To begin managing your SQL Server instances, you must first register each instance with SQL Defrag Manager. When you start the SQL Defrag Manager Console the first time, the Welcome window prompts you to select and register one or more instances.

To register a SQL Server instance with SQL Defrag Manager:

1. Click **Register** on the Fragmentation Explorer tab of the SQL Defrag Manager Console or click **Next** on the Welcome window.
2. Follow the instructions until you finish registering the SQL Server instances you want to manage. This wizard allows you to select the:
 - [Instances to register](#)
 - [Authentication type](#)
 - [Policy](#)

Access the Register SQL Servers window

To display the Register SQL Servers window, click **Register** on the Fragmentation Explorer tab of the SQL Defrag Manager Console.

This wizard makes the SQL Server instances you register available in the SQL Defrag Manager Console. You must register a SQL Server instance before you can manage, analyze, or defragment the databases, tables, and indexes in that instance.

The Register SQL Servers wizard prompts you for the information it needs to register the selected instances and databases with SQL Defrag Manager. Once you have registered an instance with SQL Defrag Manager, you can view the instance on the Fragmentation Explorer tab, analyze and defragment the databases in that instance, and customize the management settings. SQL Defrag Manager registers all databases, tables, and indexes.

Select instances to register

The Select SQL Servers to Register window lets you select the SQL Server instances to manage. Select the instances you want to register, and then click **Next**.

Access the Select SQL Servers to Register window

To display the Select SQL Servers to Register window, click **Register** on the Fragmentation Explorer tab of the SQL Defrag Manager Console.

Adding a specific instance to the list

To list the available instances on your network, click **Retrieve SQL Servers**.

If the instance you are looking for is not included in the list, you can add the information manually. To add an instance to the list by name, click **Add By Name**.

If you added the instance, but it still does not appear in the list, check your network access and the availability of the instance on the network. Then, click **Retrieve SQL Servers** to list the available instances on your network. If the instance is still not listed, contact your network administrator. When you add an instance by name, the instance must be available on the network for SQL Defrag Manager to add it to the list of managed instances.

Select authentication type

Select which credentials SQL Defrag Manager uses to connect to registered SQL Servers. If you select SQL Server authentication, you must provide a user name and password.

Select policies

The Policy window lets you select policies to apply to the instance you are registering. Select the policies you want to apply, and then click **Finish**. The following sample policies are available:

Rebuild Online Daily Top 10

Rebuild Online if available on a daily schedule the top 10 most fragmented indexes over 70%

Rebuild Weekly All

Rebuild indexes where fragmentation is greater than 70% and page count is greater than 1,000

Reorganize Daily All (Nightly Window)

Reorganize indexes where fragmentation is greater than 30% and page count is greater than 1,000

Reorganize Interval Top 10%

Reorganize the top 10% of the indexes on a one-hour interval that are greater than 30% fragmented

Reorganize Interval Top 10

Reorganize the top 10 indexes on a one-hour interval that are greater than 30% fragmented

Reorganize Large Indexes (Nightly Window)

Reorganize the largest indexes (greater than 10,000 pages) with fragmentation greater than 70% only during the nightly maintenance window (12 am to 4 am)

To Do List Reorganize Large Indexes

Run a policy once a week to create a To Do list of large indexes (greater than 10,000 pages) that are candidates for reorganization

To Do List Reorganize Medium Indexes

Run a policy daily to create a To Do list of medium indexes (greater than 1,000 but less than 10,000 pages) that are candidates for reorganization.

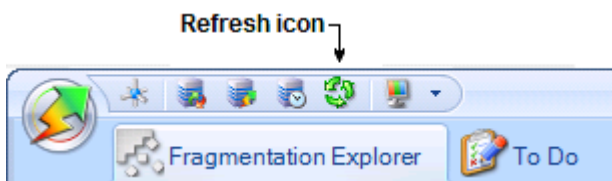
Modify your SQL Server instances list

SQL Defrag Manager provides the ability to add, remove, and edit SQL Server instances from within the tree pane.

To add a SQL Server instance while on the Fragmentation Explorer tab, click **Register**. You can also right-click in the tree pane, and then select **Add another server**. To remove SQL Server instances, right-click the server instance in the tree pane, and then select **Remove this server**. To edit an existing instance, right-click the server instance in the tree pane, and then select **Edit configuration for this server**. Make your changes, and then click **Save settings**.

To improve performance, each SQL Defrag Manager Console caches some information about objects in the database. SQL Defrag Manager maintains and updates this cached information based on activities in the Console.

When multiple SQL Defrag Manager Consoles connect to the same Repository, some changes made in one Console may not be displayed in the other Consoles. For example, if one Console registers a new instance, that instance may not immediately appear in other open Consoles. To refresh the information displayed in the Console, click the **Refresh** icon at the top left edge of the SQL Defrag Manager Console, as shown in the following figure.



To refresh all the cached database information in a Console, display the Fragmentation Explorer tab and then click the **Refresh** icon in that Console.

Support partitioned indexes

SQL Defrag Manager provides partitioned index support through the Fragmentation Explorer. Partitions appear as a child of the index node and include the same pie chart and statistic data layout as an index. You can analyze and defragment a partition just as easily as an index. Any activity that occurs on a parent node affects the child nodes, including partition nodes.

For each partition, SQL Defrag Manager includes a pie chart, average fragmentation percentage, the number of pages included, number of pages per fragment, and the page density.

Analyzing or defragmenting a partition

To [analyze](#) a partition, right-click the appropriate partition, and then click **Analyze this partition**. From there, you can select from a fast or thorough analysis. To [defragment](#) a partition, right-click the appropriate partition, and then click **Defragment this partition**. From there, you can select whether you want to reorganize or rebuild the selected partition.

SQL Defrag Manager displays all available index partitions through auto discovery. If your partitions are not displayed in the available list, you can [register an instance](#) with SQL Defrag Manager by clicking **Cancel**, and then clicking **Master Control > Register**.

Adjust fill factor

SQL Defrag Manager allows you to adjust the fill factor and padding for your indexes. The Fill Factor Settings window provides a recent analysis and fragmentation results for the selected index. Use this information to decide whether to adjust your fill factor. The **New Values** area allows you to enter your new values in the **Fill Factor** and **Pad Index** fields. The **Current Values** area displays your current fill factor and index padding.

Access the Fill Factor Settings window

To display the Fill Factor Settings window, right-click the index for which you want to adjust the fill factor, and then select **Adjust Fill Factor**. SQL Defrag Manager displays this option only when you select an index.

Differences when rebuilding the index online or offline

The Fill Factor Settings window allows you to rebuild the index online or offline based on the [optimize performance for your environment](#). Online rebuilds use more memory because SQL Server builds a new copy of the index in memory while providing access to the index and the underlying table for the duration of the rebuild. Offline rebuilds use less memory because the rebuild occurs on the index itself, which requires that the index is locked and inaccessible for the entire time necessary for the rebuild.

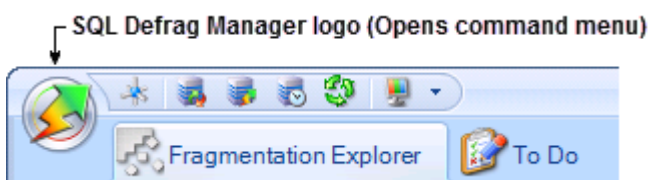
Note that if you perform an offline rebuild on a clustered index, then the entire table is inaccessible until the rebuild completes. This functionality occurs because the leaf-level pages of the clustered index are the physical data pages of the table.

Manage your licenses

The License Key Manager provides an intuitive, simple-to-use interface for SQL Defrag Manager license key management. You can view the status of your license keys and add licenses to manager additional instances. Each license allows you to manage a predetermined number of SQL Server instances. When you reach your limit, SQL Defrag Manager does not allow you to add new instances.

Access the Manage Licenses window

To display the Manage Licenses window, click **Manage Licenses** on the SQL Defrag Manager command menu of the SQL Defrag Manager Console. To display the command menu, click the SQL Defrag Manager logo in the top left corner of the SQL Defrag Manager Console, as shown in the following figure.



Managing licenses

You may need to add a license if you exhaust your trial license.

To upgrade a trial license to a permanent license:

1. On the SQL Defrag Manager command menu, click **Manage Licenses**.
2. On the Manage Licenses window, click **Add**
3. On the license key window, enter the license key.
4. Click **OK**.

License terms

By default, SQL Defrag Manager installs with a limited time, unlimited instance trial license key. After you install the SQL Defrag Manager components using the Typical or Custom setups, the Management Console lists your trial license in the Manage Licenses window. You cannot manage the default trial license. This license key is stored in the SQL Defrag Manager Repository.

SQL Defrag Manager licenses are issued per SQL Server instance and for a specific time period. You can manage these licenses with the License Key Manager.

Cloud for Azure and AW EC2

Follow these instructions to setup cloud environment for SQL Defrag to test on Azure and AWS EC2:

1. Create Virtual Machines in same virtual network and network security group
2. While creating a VM, create a virtual network and specify the virtual network for the virtual machine
3. Edit windows firewalls to create an inbound rule to allow traffic on port 1433

Automate defragmentation using policies

You can use SQL Defrag Manager to analyze database index fragmentation across your enterprise. Analyzing indexes allows you to find and defragment hot spots to improve performance. SQL Defrag Manager uses policies to automate this process and maintain your databases with minimal time and effort.

Accessing policies

To view information about any policy, click the **Automation Policies** tab. You can view information about all your policies (per type) or view the status of an individual policy. You can also create new policies, edit, delete, enable, or disable existing policies, as well as to import and export policies.

Creating a policy

Before creating an automated defragmentation policy, review the required information so that you can [create an efficient and effective policy](#).

Create a policy

You can use SQL Defrag Manager to analyze database index fragmentation across your enterprise. You can then defragment hot spots to improve performance. SQL Defrag Manager provides policies to automate this process and maintain your databases with minimal time and effort. From the **Automation Policies** tab, click the **Create** option to access the Policy Configuration wizard.

To get started with the Policy wizard:

1. [Select the indexes you want to evaluate.](#)
2. [Select the index thresholds and analysis frequency.](#)
3. [Select index filter options.](#)
4. [Select policy response](#)
5. [Schedule when and how often you want the policy to run](#)
6. [Select which resource checks you want to run before running the policy.](#)
7. [Select what action you want to occur when certain conditions are met.](#)
8. [Name the policy.](#)

If you want to perform manual defragmentation activities, you do not need to define a policy. The policies define how SQL Defrag Manager handles automated analysis activities and the defragmentation activities related to the results of those automated analyses.

If you want to define custom settings for specific database objects, such as creating To Do items for fragmentation issues identified in a very large index instead of attempting to automatically defragment that index, you should define the policy for those specific database objects.

Policy Wizard - Select the indexes you want to evaluate

The **Targets** tab allows you to select the indexes you want this policy to evaluate. Your selection in the **Index Types** drop-down list determines whether SQL Defrag Manager displays all indexes, or only clustered or non-clustered indexes.

 Changing the **Index Type** clears the selected targets.

If your indexes are not displayed in the available list, you can [register an instance](#) with SQL Defrag Manager by clicking **Cancel**, and then opening the command menu and clicking **Register**.

Indexing individual partitions

SQL Defrag Manager evaluates all partitions in an index based on the settings in the **Thresholds** tab and reacts according to the settings in the **Response** tab. The **Thresholds** tab allows you to set your fragmentation and scan density rules, and your analysis frequency. The **Response** tab lets you choose how you want the policy to respond in a certain situation.

Policy Wizard - Select the index thresholds and analysis frequency

Select the index thresholds and analysis frequency

The **Thresholds** tab allows you to adjust the rules for Fragmentation and Scan Density, and set the frequency of analysis.

This window provides a slider control that allows you to adjust the threshold value that triggers a response for this policy. To change the current threshold value, click the divider button between two levels and drag it left or right to decrease or increase that threshold value.

Recommended fragmentation and scan density ranges

The fragmentation level identifies the amount of fragmentation of a database object. The fragmentation range allows you to define what range of fragmentation levels are displayed as Acceptable, Caution, and Critical.

Indexes typically perform well when fragmentation is less than 30% and scan density is greater than 80%. Select the ranges that are appropriate for your selected target indexes. If you select Fragmentation and Scan Density, both conditions must be met for an index to comply with this policy. Once you specify these threshold levels, SQL Defrag Manager uses these values to determine how fragmentation levels are displayed in the Console. SQL Defrag Manager also uses these values when performing automated responses based on fragmentation levels found during scheduled analysis activities.

How SQL Defrag Manager uses these values

SQL Defrag Manager displays objects with an Acceptable fragmentation level as green, objects with a Caution fragmentation level as yellow, and objects with a Critical fragmentation level as red. On the **Summary** tab, you can define indexes that are not compliant with this policy as either Caution or Critical.

You can also assign a response for indexes which are not in compliance with this policy. These responses range from creating a To Do item on the SQL Defrag Manager Console To Do tab to automatically rebuilding indexes with the associated fragmentation level.

You can specify custom threshold levels for different database objects. If you have a table that impacts productivity with low levels of fragmentation, you may want to lower the threshold levels for that table. Although you can manage multiple threshold levels across many objects, use standard fragmentation levels across most objects to reduce complexity and to present data consistently in the Console.

Select a frequent analysis renewal for indexes that fragment rapidly due to updates, deletions, and insertions; select less frequency for indexes that are more static.

Policy Wizard - Select index filter options

The **Filters** tab allows you to select additional criteria for this policy to evaluate the selected indexes.

You can choose from the following additional filters for your policy:

Page Count

Set the minimum and maximum ranges to dictate the size of the indexes selected. Indexes with small page counts do not benefit as much from defragmentation because query optimizers often use table scans instead of indexes.

Operation Order

Choose the order in which the optimization occurs for the qualified indexes.

Qualifying Indexes

Select a top number or percentage of qualified indexes to decrease the number of indexes optimized.


Policy Wizard - Select policy response

Select policy response

The **Response** tab allows you to select how the policy responds when the specified conditions are met.

Advantages of each optimize response

Reorganize does not hold long-term table locks and has less impact to availability. However, it results in less reduction of fragmentation levels. **Rebuild** has a higher impact on your system availability, but provides the best results in the reduction of fragmentation. If availability during the rebuild is important, select **ONLINE (SQL 2005/2008 only)**, but note that this option requires additional operation time and resource consumption. Use `SORT_IN_TEMPDB` to direct SQL Server to use tempdb for the intermediate sort results used to build the index. For more information, see Microsoft Books Online.

 If you select **Rebuild**, the **ONLINE (SQL 2005/2008 only)** option applies to the Enterprise versions of SQL Server 2005, 2008, and 2008 R2.

Adding qualifying index defragmentation to the To Do list

You can select whether you want SQL Defrag Manager to automatically defragment qualifying indexes or to add them to the To Do list. These responses define what SQL Defrag Manager does when it identifies a fragmented index during an automated, scheduled analysis. For example, if you select **Yes, Automatically Defrag**, when SQL Defrag Manager performs a scheduled analysis activity, it attempts to defragment each index it finds that is not compliant with policy, using the selected reorganize or rebuild method. If you select **No, Add To Do item**, when SQL Defrag Manager performs a scheduled analysis activity, SQL Defrag Manager creates a To Do item on the **To Do** view of the SQL Defrag Manager Console for each index it finds that is not compliant with policy.

Available options for updating the statistics

The SQL Defrag Manager **Response** tab provides three options for updating your statistics:

- **Before Analysis Renewal:** Updates the statistics for each object covered by this policy each time an analysis is performed.
- **Before Optimization:** Updates the statistics before reorganization or rebuild occurs so the query optimizer has the most current information for building the execution plan for the operation.

- **Never:** Prevents any statistics updates regardless of what other activities occurred.

These options can be applied using a **Full Scan**, **Sample Percent**, **Sample Rows**, or **Resample** method.

Select the **Use NORECOMPUTE** check box, and SQL Server does not maintain the statistics after you perform a manual update. Leave this check box clear and SQL Server maintains statistics after a manual update.

Policy Wizard - Schedule when and how often you want the policy to run

The **Schedule** tab allows you to select when and how often the policy runs. Select the **Type** to change the Schedule options displayed by SQL Defrag Manager. Use **Restrictions** to create a maintenance window for the time frame in which the policy runs. You can also set the policy to stop execution once this window passes.

Automation types

Based on your specified policy settings, SQL Defrag Manager analyzes registered instances and can also defragment the hot spots it finds. The schedule you specify in this window defines how often the policy analyzes the selected instances and databases. You can use SQL Defrag Manager in several ways:

- You can schedule SQL Defrag Manager to analyze your registered instances on a regular basis and automatically defragment database objects as needed. SQL Defrag Manager uses your defined responses and [threshold levels](#) to determine what type of defragmentation tasks to perform.
- You can schedule SQL Defrag Manager to analyze your registered instances on a regular basis. You can define the policies so SQL Defrag Manager creates To Do items when it finds Caution or Critical levels of fragmentation. Then, you can review the To Do items on a regular basis and run the defragmentation tasks when appropriate.
- You can analyze the database objects when you want updated analysis information. Then, you decide what to do based on the data collected and displayed. You can review the results and [manually defragment](#) objects that have high levels of fragmentation.

Considerations when defining a schedule

SQL Defrag Manager allows you to define a custom schedule for database fragmentation analysis. You can define the exact schedule you want to use when analyzing the associated database objects. Scheduling is ideal when you have specific time constraints for analysis and defragmentation tasks, or when there are only certain hot spot areas you want to analyze and defragment on a regular basis. For example, since a thorough analysis can use system resources needed for other database activity, you may want to schedule this analysis during periods of low database activity or demand. You can schedule a custom analysis on a single date and time, or at specified intervals, such as daily or weekly.

The **Interval** option defines the period of time to wait from the end of a rebuild before running the next analysis. For example, if you select the **Interval** option and specify 60 minutes as the repeat interval, SQL Defrag Manager waits 60 minutes from the end of a rebuild before running the next policy.

If you select the **Interval** or **Daily** option, you can specify the time period during which the policy can start. SQL Defrag Manager runs the policy and the associated

defragmentation activities based on the defined automated responses. You can set SQL Defrag Manager to stop these automated analysis and defragmentation activities when the end time specified in the **Restrictions** for the **Execute only between** option is reached.

If the policy you are creating/editing includes server objects that change very frequently, you may consider enabling the **Rediscover server objects** option. By enabling this option, SQL Defrag Manager will refresh the list of server objects prior to running the policy to ensure it has an updated list of server objects. Enabling this options, however, will impact the performance of the managed instance(s).

 **Note:**

By default, SQL Defrag Manager automatically rediscovers server objects once a day at 3:00 AM.

Policy Wizard - Select which resource checks you want to run before running the policy

The **Resource Check** tab allows you to select which checks SQL Defrag Manager performs before running a policy. These checks define the conditions that allow a policy to execute. If the conditions are met, SQL Defrag Manager runs the policy.

The following columns of information are available on the **Resource Check** tab.

Name

Select the check box next to the name of the resource check to apply it to this policy.

Comparison

Select how the resource check evaluates the value for this type:

- Numeric comparisons
 - is equal to
 - is not equal to
 - is greater than
 - is greater than or equal to
 - is less than
 - is less than or equal to
- Alphanumeric comparisons
 - is equal to
 - is not equal to
 - contains
 - does not contain
 - starts with
 - does not start with

Value

Type the value to which the resource check compares data.

- ✔ Leave the **Executing Job Name** resource check value empty to prevent the policy from running when any other job is running.

Details

Describes the resource you want SQL Defrag Manager to check.

Resource Check Options

Allows you to **Verify Server Support** for the Resource Checks previously established to run the policy. You can also set the policy to cancel automatically if the resource checks prevent it from running for a defined period of time.

Policy Wizard - Select what action you want to occur when certain conditions are met

The **Alerts** tab allows you to select what action you want SQL Defrag Manager to perform when certain events occur.


 You can modify the value for the **Policy has not completed successfully in ... days.**

Alerts

Select to notify by Email or create an event log when certain conditions for the associated policy are met.

Email Options

Select the appropriate checkbox to send Email alerts to the default or specified recipients.

 Click **To...** to choose from a list of previously registered Email addresses.

Setting default email configuration

Click **Configure** next to the **Send email alerts to the default email recipient** option to open the Default Email Configuration window. [Click here](#) to learn more about the Default Email Configuration settings.

Policy Wizard - Name the policy

The **Summary** tab allows you to include a name and description for the policy. Check **Enable Policy** to activate this policy.

SQL Defrag Manager uses the Caution and Critical fragmentation level options to perform automated responses as defined in the **Response** tab.

You are required to enter a unique name for each policy.

Both the name and description appear in the status messages for your policies. Using a meaningful name and description allows you to easily identify problems when they occur. For example, consider specifying a description that help you later choose the correct backup to restore during a disaster recovery situation.

Edit a policy


To edit an automation policy, choose a policy from the **Policy Scheduled Activity** list displayed in the **Automation Policies** tab and click the **Edit** option or double-click the policy you want to edit.

You can modify the following settings, according to your requirements:

- **Targets:** Allows you to choose different or additional indexes to evaluate using the selected policy.
- **Thresholds:** Allows you to adjust the threshold values for fragmentation and scan density, as well as setting the analysis frequency.
- **Filters:** Allows you to modify the **Page Count**, **Operation Order**, and **Qualifying Indexes** filters.
- **Response:** Enables you to define a different response when the specified conditions are met.
- **Schedule:** Allows you to change when and how often the policy runs, as well as to modify or establish time restrictions if necessary.
- **Resource Check:** Allows you to modify the resource parameters that need to be verified and met to execute the policy.
- **Alerts:** Enables you to change the alerts generated by the policy when certain events occur.
- **Summary:** Allows you to modify the name and description of the policy.


Import a policy

To import a policy, click the **Import** option displayed in the **Automation Policies** tab. SQL Defrag Manager displays a window where you can enter the policy details such as **Policy Name** and **Description** to help identify the policy, as well as locating the file you want to import. Click **OK** to import the policy.

 The file needs to be in XML format and include settings for Thresholds, Filters, Response, Schedule, Resource Check, and Alerts.

Export a policy

To export an existing policy, in the **Automation Policies** tab, select a policy from the **Policy Scheduled Activity** list and click the **Export** option. SQL Defrag Manager displays a window where you can select the path to save the file and enter the file name. Click **Save** to export the policy.

 The policy will be saved to an XML file.

View To Do tasks based on policy results

The **To Do** tab of the SQL Defrag Manager Console lists the indexes and partitions that require your attention based on automated policies. When you create policies, you specify threshold levels and automated responses for these fragmentation levels. Then, when SQL Defrag Manager performs a scheduled analysis, it can create To Do items for each analyzed index based on your policies.


You can review the displayed activities and choose which activities to perform. These activities help you defragment the objects that have serious or critical levels of fragmentation.

Access the To Do view

To display the To Do view, click the **To Do** tab of the SQL Defrag Manager Console.

How partitioned indexes are addressed in the To Do view

SQL Defrag Manager supports partitioned indexes similarly to non-partitioned indexes. A row exists for each partition that triggers an action on the To Do view. Each partition in an index is identified by a number, which SQL Defrag Manager displays in the Partition column. Select the row for the partition and SQL Defrag Manager displays additional information about that partition in the Details section of the window.

 SQL Defrag Manager displays "1" in the **Partition** column for a non-partitioned index.

The data in the Details section of the window provides complete information about the item selected in the main section of the window.


Item	Description
Server Name	Specifies the name of the instance that contains the database.
Database Name	Specifies the name of the database that contains the table.
Table Name	Specifies the name of the table that has the identified index to defragment.
Index Name	Specifies the name of the index to defragment.
Partition Number	Specifies the number of the partition to defragment.

Item	Description
Clustered Index	Indicates whether the index is a clustered index.
Detected On	Identifies when the analysis that created this To Do item occurred.
Problem	Describes the issue identified by SQL Defrag Manager.
Recommendation	Provides a recommendation for how to resolve the identified fragmentation issue.

How grouping information helps

You can group activities by the policy that generated the To Do activity, the identified problem or related recommendation, by the name of the instance, database, table, or index, or the partition number. To group the displayed activities by a column value, drag that column heading to the gray bar above the column headings.

Grouping To Do activities allows you to quickly scan and evaluate the recommended activities. For example, you can group and view activities for a specific database, table, or index, which lets you focus on the recommended activities for that database object to make sure they are the activities you want to run.

 Grouping on a partitioned index column causes SQL Defrag Manager to display a single row for that index including totals of all partition statistics.

Perform an action listed on this window

Select the item, and then click **Auto Fix** near the top of the window. Auto Fix performs either an analysis or defragmentation based on the recommended action. To analyze the object again, select the To Do item for that object, and then click **Analyze**.

To sort the activities by a column value, click the heading of that column. You can also group activities by a column value by dragging that column heading to the gray bar just above the column headings.

You can remove an action by selecting the item, and then clicking **Clear** near the top of the window.

Perform a manual defragmentation

When you initially use SQL Defrag Manager, you should first analyze your databases to collect updated fragmentation information. Then, review that information and defragment the highly-fragmented areas. As you become familiar with SQL Defrag Manager, you may choose to automate this process. This section describes how to manually analyze, view, and correct fragmentation issues on tables and indexes. Use the following topics to guide you through the manual defragmentation process.

- [Analyze database fragmentation](#)
- [View analysis results](#)
- [Optimize performance and space usage](#)
- [View recent activity](#)
- [Using the Command Line](#)

Analyze database fragmentation

The SQL Defrag Manager Analyze Fragmentation window allows you to investigate the [fragmentation](#) levels of tables and indexes in the registered databases and instances.

Access the Analyze Fragmentation window

To open the Analyze Fragmentation window, click **Analyze** on the Fragmentation Explorer tab of the SQL Defrag Manager Console.

Selecting the instances and databases to analyze

The SQL Servers and Databases field lists all the registered instances and databases. When you select an instance, SQL Defrag Manager selects all the databases in that instance. You can manually select specific databases to analyze tables and indexes in those individual databases.

If the instance or database you want to analyze is not listed in the SQL Servers and Databases field, you may not have registered it with SQL Defrag Manager. To register an instance with SQL Defrag Manager, click **Register** on the Fragmentation Explorer tab of the SQL Defrag Manager Console.

Difference between fast and thorough analysis

Fast analysis provides base-level analysis information, including statistics about logical order fragmentation. Logical order fragmentation occurs when the physical order of pages does not match the logical order of the pages. A fast analysis runs five to eight times quicker than a thorough analysis because SQL Defrag Manager does not scan the leaf pages of an index. Therefore, SQL Defrag Manager does not gather page density information during a fast analysis. You can perform a fast analysis on a database that is in use because it consumes less time and resources than a thorough analysis.

Thorough analysis provides comprehensive analysis information, including page density statistics. This analysis type requires more time and resources than a fast analysis and may block active processes while it collects analysis information. Therefore, you should run a thorough analysis only during times when the databases are not needed.

The time required to perform an analysis depends on the number of pages used by the databases, the activity level on those databases, the performance of the computer where the database resides, and the type of analysis you choose to perform. In general, an analysis can require from one minute to several hours.

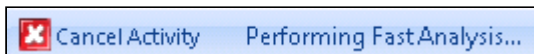
Automation allows you to define a custom schedule for analyzing and potentially defragmenting databases during periods of low activity. You can also set SQL Defrag

Manager to analyze databases on a regular schedule, such as daily or weekly. Since an analysis can use system resources needed for other common database processes, you may want to schedule this analysis during periods of low database activity or demand.

2-Steps analysis

The **Fragmentation Explorer** tab allows you to easily analyze individual objects from the navigation tree, such as a specific instance, database, table, or index, as well as the entire SQL Defrag Manager environment. Right-click the object you want to analyze to open the contextual menu, click the **Analyze all/entire...** option and select the type of analysis you want to perform. [Click here](#) to learn more about actions you can perform from the contextual menu.

Canceling an activity



When an activity is running, SQL Defrag Manager displays a **Cancel Activity** option in the lower left-hand corner of the status bar. Click the option to stop the activity from running.

View analysis results

Once SQL Defrag Manager analyzes the registered instances and databases, you can use the **Fragmentation Explorer** tab to view the latest analysis information. SQL Defrag Manager displays the analysis details for the objects in the selected database object. For example, if you select a table, SQL Defrag Manager displays the analysis details for each index on that table.

You should regularly monitor fragmentation levels on indexes and schedule SQL Defrag Manager to analyze database fragmentation. Use the scheduled analysis to identify specific fragmentation levels before [defining your policies to automatically reorganize or rebuild](#) indexes.

Access the Fragmentation Explorer

To display the Fragmentation Explorer, click the **Fragmentation Explorer** tab of the SQL Defrag Manager Console.

About the icons and fragmentation statistics

You can mouse over any icon to display a tip that identifies what that icon represents. The fragmentation statistics include:

Average Fragmentation

Displays the total fragmentation of all indexes on the associated object and the objects contained divided by the total number of indexes on those objects. This calculation is weighted based on the size of each index. For example, if you have three similar-sized indexes with 10%, 10%, and 70% fragmentation levels respectively, this value is 90% divided by three, which is 30%. The logical fragmentation level for each index is the percentage of out-of-order pages in that index when the index was last analyzed.

The Average Fragmentation value can help identify critical fragmentation levels as you drill-down to details on individual tables and indexes. You can compare logical fragmentation levels across indexes to identify hot spots. You may see performance gains after defragmenting indexes with as little as 10% fragmentation. In general, consider defragmenting indexes with 20% or more fragmentation.

Pages Read

Displays the total number of pages used by the associated object and the objects it contains. For example, when you view this value for a database, it displays the total number of pages used by the database, including all pages used by all the tables and indexes in that database.

Fragmentation impacts disk I/O. Since larger indexes are less likely to be cached by SQL Server, fragmentation impacts the performance of large indexes more than small indexes. In general, focus on indexes with 1,000 pages or more.

Pages Per Fragments

Displays the average number of pages per fragment. This value is the total number of pages identified by the Pages Read value divided by the number of those pages that are not in logical order.

Page Density

Displays how much data is stored in the pages, identified by the Pages Read value, compared to the maximum amount of data that can be stored in those pages. Only a thorough analysis collects or updates this information. If no page density information has been collected for these pages, this value is set to N/A (not available).

Low page density can cause SQL Server to read more pages than necessary when using an index, such as when performing a query. When data is initially stored on the pages, SQL Server uses the fill factor value to determine how much data to store on each page. At this point, page density is relatively high, but the page density becomes lower as data is inserted and pages are split to make room for the additional data. SQL Defrag Manager can reorganize or rebuild indexes to increase page density and reduce the number of reads when using the index.

Indexes

Displays the total number of indexes on the associated object and the objects it contains.

About the Show/Hide filter

The Show/Hide filter options allow you to refine which objects SQL Defrag Manager displays on the Fragmentation Explorer tab. When you clear the check box for a group of objects, such as the **Excluded** check box, SQL Defrag Manager does not display those objects and updates the displayed fragmentation statistics, such as Average Fragmentation and Relative Fragmentation to not include those objects.


The Criteria option defines which objects are represented by the Excluded group. If you clear the Excluded check box, SQL Defrag Manager does not display objects that match the specified criteria.

Use the contextual menu filter option

You can apply filters to either the complete SQL Defrag Manager environment (all registered instances), an instance, a database, or a table. From the navigation tree, right-click the object you want to filter and select **Filter** to open the dialog window.

Use the text field in the dialog to enter the exact name of the instance, database, or table you are searching. You can also use '%' as a wildcard to filter the results by keywords. Use a comma to separate multiple filtering criteria.

Click **Find** to apply the filter. SQL Defrag Manager displays the results in the Fragmentation Explorer.

To remove the applied filters, right-click the instance, database, or table you applied the filters to and choose **Clear filter**, or click the **Clear Filters** icon  above the navigation tree.

Rediscover Objects

Use this feature to update the view of objects found on a specific server, database, or table. Right-click the specific server, database, or table and select the **Rediscover Objects** option.

Performing actions from the Explorer navigation tree

You can perform several actions on either the entire SQL Defrag Manager environment, to a specific instance, database, or table from the navigation tree. Right-click an object to open the contextual menu options. The actions available depend on the object selected.

Actions on the entire SQL Defrag Manager environment

- **Analyze all registered SQL Servers:** Allows you to select the type of analysis - Fast, Thorough-Sampled, or Thorough-Detailed
- **Defragment all registered SQL Servers:** Allows you to select the type of defragmentation - Reorganize, Rebuild, or Rebuild (online)
- **Add another SQL Server:** Opens the Register SQL Servers configuration window
- **Filter:** Allows you to filter the objects view by name or keywords

Actions on a specific instance

- **Analyze entire server:** Allows you to select the type of analysis - Fast, Thorough-Sampled, or Thorough-Detailed
- **Defragment entire server:** Allows you to select the type of defragmentation - Reorganize, Rebuild, or Rebuild (online)
- **Rediscover server objects:** Updates the view of objects found on the server
- **Edit configuration for this server:** Allows you to modify authentication settings
- **Remove this server:** Allows you to remove the instance from SQL Defrag Manager

- **Add another server:** Allows you to register an instance
- **Filter:** Allows you to filter the objects view by name or keywords

Actions on a specific database

- **Analyze entire database:** Allows you to select the type of analysis - Fast, Thorough-Sampled, or Thorough-Detailed
- **Defragment entire database:** Allows you to select the type of defragmentation - Reorganize, Rebuild, or Rebuild (online)
- **Rediscover database objects:** Updates the view of objects found on the database
- **Filter:** Allows you to filter the objects view by name or keywords

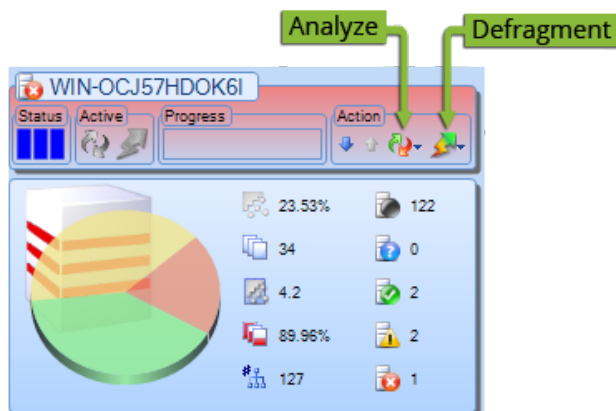
Actions on a specific table

- **Analyze entire table:** Allows you to select the type of analysis - Fast, Thorough-Sampled, or Thorough-Detailed
- **Defragment entire table:** Allows you to select the type of defragmentation - Reorganize, Rebuild, or Rebuild (online)
- **Rediscover table objects:** Updates the view of objects found on the table
- **Filter:** Allows you to filter the objects view by name or keywords

Performing actions from the objects view

To analyze or defragment a specific table:

1. In the navigation tree, click the database that contains the table you want to analyze or defragment.
2. In the right pane, click the Analyze or Defragment option in the panel for the table you want to analyze or defragment.



3. Select the analysis or defragmentation type you want to perform.

Canceling an activity

When an activity is running, SQL Defrag Manager displays a **Cancel Activity** option in the lower left-hand corner of the status bar. Click the option to stop the activity from running.



Optimize performance and space usage

The **Defragment and Optimization** window allows you to defragment one or more instances and databases. The SQL Servers and Databases field lists all the registered instances and databases. You can select the databases you want to defragment, and then select the type of defragmentation approach you want SQL Defrag Manager to use. SQL Defrag Manager can simply reorganize indexes for optimal performance, or defragment indexes by completely rebuilding them.

Access the Defragment and Optimization window

To display the Defragment and Optimization window, click **Defragment** on the Fragmentation Explorer tab of the SQL Defrag Manager Console.

Selecting instances and databases to defragment

The **SQL Servers and Databases** field lists all the registered instances and databases. You can select an instance, which then selects all the databases in that instance. You can also select specific databases to defragment indexes in those individual databases.

If the instance or database you want to defragment is not listed in the SQL Servers and Databases field, you may not have registered it with SQL Defrag Manager. To register an instance with SQL Defrag Manager, click **Register** on the Fragmentation Explorer tab of the SQL Defrag Manager Console.

Difference between reorganize and rebuild

The type of defragmentation you choose depends on many considerations:

- Database availability
- Logical fragmentation level
- Parallel processor availability
- Page compaction
- Can you stop and restart the process without losing data?
- Can you untangle interleaved indexes?
- Is additional disk space available in the data file?
- Is the index large?
- Do you need to rebuild statistics?
- Should busy pages be skipped?

Reorganize: Directs SQL Defrag Manager to defragment the tables and indexes online. This defragmentation can be stopped without losing progress. The amount of log space used varies with the amount of work completed. During reorganization, SQL Defrag Manager skips pages that are in use. In general, select

this defragmentation type when the instance or database must be available during defragmentation or when you defragment a small index.

Rebuild (SORT_IN_TEMP): Directs SQL Defrag Manager to defragment the tables and indexes offline by completely rebuilding the indexes. This defragmentation type requires sufficient free space on your server and the amount of log space used can be extensive in full recovery mode. On large, fragmented indexes, this defragmentation type is significantly faster than a reorganization. In general, select Rebuild when it is acceptable that the database is unavailable during the time that the indexes are rebuilt.

Rebuild - ONLINE (SQL 2005 Enterprise and later): Directs SQL Defrag Manager to defragment the tables and indexes online by completely rebuilding the indexes. This defragmentation type requires greater resources such as disk space, memory, and processing. Because of the additional operation time and resource consumption, rebuilding online is only recommended when availability is important.

The time required to defragment a database depends on the number of pages used by the database, the activity level on that database, the performance of the computer where the database resides, and the type of defragmentation you choose to perform.

Automation allows you to define a custom schedule for analyzing and potentially defragmenting databases during periods of low activity. You can also set SQL Defrag Manager to analyze databases on a regular schedule, such as daily or weekly. Because the database is unavailable while SQL Defrag Manager rebuilds one or more indexes in the database, you should run this defragmentation during periods when the database can be offline and unavailable. [Click here](#) to learn more about Automation.

2-Steps defragmentation

The **Fragmentation Explorer** tab allows you to easily defragment individual objects from the navigation tree, such as a specific instance, database, table, or index, as well as the entire SQL Defrag Manager environment. Right-click the object you want to defragment to open the contextual menu, click the **Defragment all/entire...** option and select the type of defragmentation you want to perform. [Click here](#) to learn more about actions you can perform from the contextual menu.

Canceling an activity

When an activity is running, SQL Defrag Manager displays a **Cancel Activity** option in the lower left-hand corner of the status bar. Click the option to stop the activity from running.



Using the Command Line

You can perform a manual defragmentation using the command line utility `sqldefragc`. This utility is included with the SQL Defrag Manager installation.

Note: You need to run the command line console as an Administrator.

```
sqldefragc [[[-rebuild [online]] | -reorg] -instance <instance name> -  
database <database name> | -policy <policy name>]
```

There are two ways to use `sqldefragc`

1. Specify **rebuild** or **reorg** where **online** is optional for **rebuild**, specify the **instance name**, and specify the **database name**.

```
sqldefragc [[-rebuild [online]] | -reorg] -instance <instance name> -database  
<database name>
```

2. Define the defragmentation based on an existing policy by specifying the **policy name**.

```
sqldefragc -policy <policy name>
```

View activity and results

After an action occurs, you can view alert history (if configured), the recent activities occurring in your environment, and run reports to compile and display the available information. All of this data helps you verify that your policies are effective and that you are using SQL Defrag Manager to its maximum effectiveness. Use the following topics to help you decide what information you want to view.

- [View alert history](#)
- [View recent activity](#)
- [View reports](#)

View alert history


If you configured policies to send alerts, SQL Defrag Manager displays a record of those alerts on the Alerts History tab.

Click **Export** to export the data in html, xml, or txt format. Click **Print** to send all the data to a printer. Select an alert and click **Clear** to remove it from the list. Click **Clear All** to erase the existing alert history.

Filtering the alert data

The **Filters** box allows you to narrow the results based on specified criteria. Select **Type** to show only alerts of a single severity level. Select **From** and **To** to show only alerts generated during a specified time frame.


SQL Defrag Manager supports partitioned indexes similarly to non-partitioned indexes. A row exists for each partition that triggers an alert. Each partition in an index is identified by a number, which SQL Defrag Manager displays in the **Partition** column.

 SQL Defrag Manager displays "1" in the **Partition** column for a non-partitioned index.

Grouping information

You can group activities by the description that generated the alert, the identified problem or action, by the name of the policy, instance, database, table, or index, the partition number, or the date and time the alert occurred. To group the displayed activities by a column value, drag that column heading to the gray bar above the column headings.

Grouping alerts allows you to quickly scan and evaluate the health of your activities. For example, you can group and view alerts for a specific database, table, index, or partition, which lets you focus on the alerts for that database object to make sure they are the alerts you want to receive and what action may be necessary on that object.

 Grouping on a partitioned index column causes SQL Defrag Manager to display a single row for that index including totals of all partition statistics.

View recent activity

The **Recent Activity** tab of the SQL Defrag Manager Console displays a list of the analysis and defragmentation tasks performed. You can view the improvements accomplished during each defragmentation task.

SQL Defrag Manager supports partitioned indexes similarly to non-partitioned indexes. A row exists for each partition having a recent activity. SQL Defrag Manager displays the total number of partitions analyzed or defragmented in the Partition column. Select that row and SQL Defrag Manager displays additional information about the partitions in the Details section of the window.

Displaying details for an activity

To view the details of an activity, click the plus sign icon to the left of the activity. SQL Defrag Manager lists expanded information about each index or partition. You can select an item in the list and then review the information displayed in the **Details** section of the window.

About the details

The information in the **Details** section of the window provides complete information about the recent activity selected in the main section of the window.

Item	Description
Action	Specifies whether the action was an analysis or a defragmentation, and the type of analysis (fast or thorough), or the type of defragmentation (reorganize or rebuild) that occurred.
Job Type	Identifies whether the action was scheduled or run manually.
Process Time	Identifies the amount of time required for the action to finish.
Result	Specifies the final result of the action, such as whether it was completed.
Error Detail	Identifies any error that occurred during the action.
SQL Server Name	Specifies the name of the instance that contains the database.

Item	Description
Database Name	Specifies the name of the database that was analyzed or defragmented. If you select an activity for more than one database, this field identifies the total number of databases included in the recent activity.
Table Name	Specifies the name of the table that was analyzed or defragmented. If you select an activity for more than one table, this field identifies the total number of tables included in the recent activity.
Index Name	Specifies the name of the index that was analyzed or defragmented. If you select an activity for more than one index, this field identifies the total number of indexes included in the recent activity.
Partition Number	Specifies the number of the partition that was analyzed or defragmented. If you select an activity for more than one partition, this field identifies the total number of partitions included in the recent activity.

A Before/After comparison between Fragmentation and Pages statistics view is available only from the expanded view of an activity (click the plus sign icon to the left of an activity) after you select an individual index.

When SQL Defrag Manager finishes performing an analysis or defragmentation activity, it adds that activity to the **Recent Activity** tab of the SQL Defrag Manager Console.

Grouping information

You can group activities by when they occurred, the action that occurred, or by the name of the instance, database, table, index, or partition. To group the displayed activities by a column value, drag that column heading to the gray bar above the column headings.

Grouping recent activities allows you to quickly scan and evaluate your management activities. For example, you can group and view activities for a specific database, table, index, or partition, which lets you focus on the activities for that database object to make sure they are the activities you need to run.

Using this window

To sort the activities by a column value, click the heading of that column.

To remove an activity listed in the **Recent Activity** tab, select the activity, and then click **Clear**. You can also select **Clear All** to remove all activities listed in this window.

To save the information displayed in the **Recent Activity** tab, click **Export** near the top of the window to save the information as an HTML, XML, or TXT file.

To print the list of activities displayed in the **Recent Activity** tab, click **Print** near the top of the window.

You can also filter the list of activities displayed by **Result**, **Action**, and **Type**.

View reports

Once you have configured SQL Defrag Manager and assigned policies, you can run reports to compile and display the information available. You can access these reports by clicking the **Reports** tab.

You can select from the following list of reports:

- **All Registered Objects**
Shows all registered objects in SQL Defrag Manager (servers, databases, tables, indexes, and partitions), whether they are scheduled for automated analysis or defragmentation.
- **Index Counts**
Shows index counts by server, database, and table registered with SQL Defrag Manager.
- **Page Density Change Over Time**
Shows page density changes as a result of index and partition operations on a specific time frame, filtered by server, database, and minimum page count.
- **Resources Reclaimed Over Time**
Shows the number of pages reclaimed on a specific time frame, filtered by server and minimum page count.
- **Top 10 Fragmented Clustered Indexes**
Shows a list of the top 10 clustered indexes with the greatest amount of fragmentation, filtered by server and minimum page count.
- **Top 10 Fragmented Indexes**
Shows a list of the top 10 indexes with the greatest amount of fragmentation, filtered by server, minimum page count, and the option to run the report on clustered indexes only.
- **Top 10 Largest Databases**
Shows a list of the top 10 largest databases ordered by the sum of the last analyzed index page counts.
- **Top 10 Largest Indexes**
Shows a list of the top 10 largest indexes ordered by the last analyzed index page counts. You can also choose to run the report on clustered indexes only.
- **Top 10 Largest Tables**
Shows a list of the top 10 largest tables ordered by the sum of the last analyzed index page counts.

All reporting options require you select a server from the drop-down to run the Report from. Click **Run Report** to run the Report. When SQL Defrag Manager displays the Report you can print it or export it to an Excel or PDF file.

SQL Defrag Manager PDF

This page contains a direct link to the SQL Defrag Manager help in PDF format. This format is suitable for printing and for saving on your local PC for further reference. The PDF includes all pages from the relevant help published on wiki.idera.com.

- [IDERA SQL Defrag Manager 3.4](#)