

# Master Data Services Capacity Guidelines

Yair Helman

## Quick Guide



**Microsoft**<sup>®</sup>

# Master Data Services Capacity Guidelines

Yair Helman

**Summary:** This document provides capacity planning guidelines for Microsoft SQL Server 2012 Master Data Services (MDS).

**Category:** Quick Guide

**Applies to:** SQL Server 2012

**Source:** TechNet Wiki ([link to source content](#))

**E-book publication date:** May 2012

6 pages

© 2012 by Microsoft Corporation

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Microsoft and the trademarks listed at

<http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx> are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

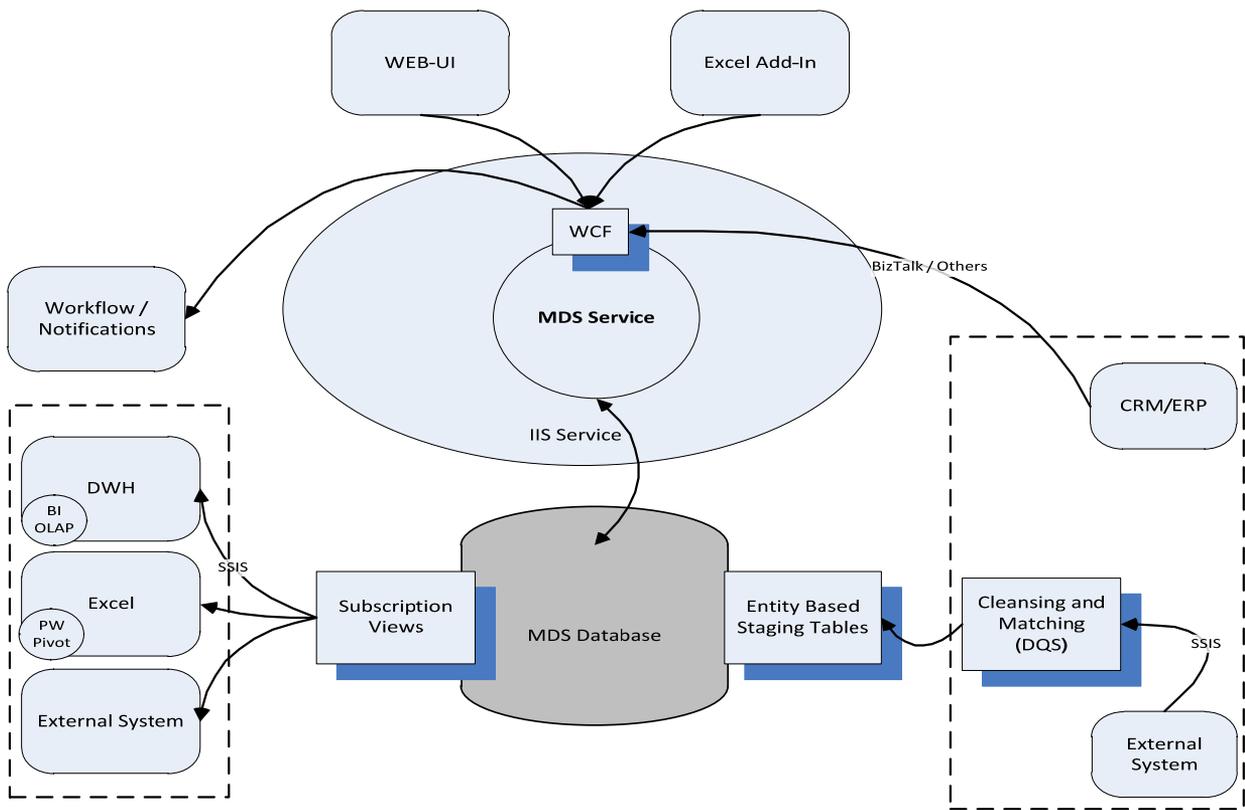
# Master Data Services Capacity Guidelines

This document provides capacity planning guidelines for Microsoft SQL Server 2012 Master Data Services (MDS). These guidelines include the following sections:

- [MDS Architecture Overview](#)
- [Capacity Considerations](#)
- [Capacity Models](#)
- [Impact of Row-Level Security on Performance](#)
- [Recommended Hardware](#)
- [Performance Considerations](#)

## MDS Architecture Overview

Master Data Services has a three-tiered architecture consisting of the database, service, and architecture layers, as shown in the following diagram.



MDS has two deployment configurations:

- Single-box (SQL Server + IIS on the same computer)
- Two-box (SQL Server on one computer, IIS on the second computer)

Most of the computational work for MDS is done in the database layer, so in most cases in order to increase capacity, a stronger SQL Server computer is required.

## Capacity Considerations

### Assumptions

Master Data Services was designed to work with data that is relatively slowly changing. Data that is involved in a high volume of transactions (that in some cases are stored in fact tables) should not be stored in MDS.

From a performance perspective, MDS will likely not process more than 50K distinct changes per day per model. By distinct is meant a separate call (either to entity-based staging or the WCF API). If significant changes are required, it is recommended that you perform them in batches either through entity-based staging or the WCF APIs. Both interfaces are tuned for batch operations and will generate much better performance than performing each change separately.

If you need to perform more than 50K calls per day per model, then it is recommended that you perform a proof of concept on the designated hardware.

### Direct Impact

The following factors will have a direct impact on MDS performance:

- Entity structure
  - Number of members in the larger entities
  - Number of attributes
  - Number of domain-based attributes
  - Number of versions
    - Note: Each version generates a copy of all the data in the entity, but all versions are still stored in the same table.
  - Number of business rules.
- Number of concurrent users
- Using row-level security.
  - Note: There is more information on row-level security in the “Impact of Row-Level Security on Performance” section later in this document.

### Slight Impact

The following factors will have a slight impact on MDS performance:

- Number of entities and number of models
- Using role-based and attribute-level security.

## Capacity Models

The following three models can be applied to capacity planning.

### Medium Capacity Model

In the medium capacity model, the MDS system has the following capacity:

- Entities with less than 500K members
- Number of attributes in entities is less than 100
- Number of domain-based attributes in an entity is less than 20
- 15 business rules per entity
- 5 concurrent users.

Medium-range hardware should easily handle this capacity. However, when using a Virtual Hard Drive (VHD), it is recommended that you perform a proof of concept. When using a VHD, it is recommended that you monitor the host server over time, and verify that there is sufficient spare capacity.

### Large Capacity Model

In the large capacity model, the MDS system has the following capacity:

- Entities with up to 10 million members have a maximum of 30 attributes
- Entities with up to 1 million members have 100 attributes or more (Note: for these entities it is recommended to use attribute groups)
- 15 or more business rules per entity.

For the large capacity model, see the recommendations below in the “Recommended Hardware” section. At this level, it is recommended that you perform a proof-of-concept.

### Extra Large Capacity Model

Any deployment that is larger than the large capacity model will likely require stronger hardware than that listed in the “Recommended Hardware” section. At this level, you should perform a proof-of-concept in the early phases of the development.

## Impact of Row-Level Security on Performance

The Microsoft Master Data Services engineering team cautions against adding row-level security to large datasets, because these permissions will generate significant permission data. The effects of hierarchy-member security on performance can be isolated by comparing performance of the model administrator to the secured users.

The following are tips for securing MDS using hierarchy-member security:

- Always secure MDS using Active Directory groups or local groups. This will limit individual permissions in the tables.
- Secure MDS using implicit denies as much as possible. Set permissions only for those areas that a user has access to.
- Do not grant users permissions to members at a higher level of the hierarchy while explicitly denying permissions at a lower level of the hierarchy.

## Recommended Hardware

The specification below focuses on the database computer and the one-box configuration option.

### Large-size Hardware Model

Use HW equivalent to the following configuration:

#### [HP Proliant DL360 G7](#)

2 Intel E5606 CPUs (quad-core 2.13GHz 80W)

24 GB (6 x 4GB PC3-10600E)

256MB Cache module for P410i

Enable HyperThreading and Turbo boost

Embedded P410i (SAS Array Controller) supports RAID 0 and 1 (part of the HW configuration)

8 x 10K RPM 2.5" HDD

\*Storing multiple models may require adding more RAM to the server.

### Mid-Size Hardware Model

The mid-size model has a similar configuration to the large-size model; however, you can reduce the amount of RAM to 12-16GB and have fewer disks in the raid.

### Impact of server virtualization

In general using server virtualization should have a minimal impact on performance.

The real challenge is to know how many IOPS (IO per Sec) you will get from your storage and what the physical storage configuration behind the virtual disk is.

Also make sure that you dedicate a sufficient amount of memory and CPU as described above.

## Performance Considerations

This section provides performance considerations for entity-based staging, user interface (UI) operations, and Excel operations.

### Performance Scenarios That Were Tested

The metrics provided below were tested by using the following two test models:

- Customer (Long) Model: 7 million members with 20 attributes in the main entity. Six attributes are domain-based attributes to other entities of varying sizes. Twelve business rules were created on this entity.
- Location (Wide) Model: 1 million members with 100 attributes in the main entity, 25 of which are domain-based.

The tests were performed with up to seven concurrent users where each user was creating members, updating members, or reading members at full speed.

All times listed below are in seconds.

### Entity-Based Staging Performance

Using the new entity-based staging interface introduced in SQL Server 2012 MDS, you can load 40K cells/sec.

It took about an hour to load the 7M model and approximately the same amount of time to load the 1M wide model.

### UI Operations Performance

Performance considerations for UI operations are as follows:

- Explorer Page Loads: While the initial page load takes longer (about 5 seconds), subsequent page loads take about a second. Skipping to the middle of the dataset takes about 3 seconds, and loading the last page directly takes a little over 5 seconds.
- Member actions: Selecting a member to edit takes 1 second no matter what page the member is on, and creating a new member completes in a little over 3 seconds.
- Filter: Simple filter operations (equals, <, >, or like (indexed)) return results in under 2 seconds. Complex operations such as non-indexed like operations or a match complete in less than 10 seconds on average.

### MDS Excel Add-in Operations

Performance considerations for Excel operations are as follows:

- Create Entity: Creating an entity from an Excel table is dependent on both the number of records and the number of columns, and appears to be linear in its progression. The number of attributes supported is based on SQL table limits, while the number of members will be constrained by Excel worksheet row limitations of 1M rows.

- Load Members: Members are loaded in a batch process and should increase in a linear progression.
- Open Filter Dialogue: The time required to open the filter dialog is directly related to the number of attributes on the selected entity. Please be patient for wide entities. Attribute groups do not affect this performance in Excel.

## Business Rules

The number of business rules and their types have a minor performance impact on a single-member edit. As a result, they will not have an effect on the user experience of adding/updating a small number of members.

The main performance impact in relation to business rules is in the case of first-time validation or after changing one or more business rules. In this case, the validation process needs to validate each member. There is no simple way to compute how each rule adds to the overall computation time because the computation is done in batches.

Note that later validation runs will only re-validate members that changed.

On the 7M customers long model, we had 12 different rules (default values, mandatory, uniqueness, concatenation, and more...), and it took about 2 hours to validate the entire entity.

## Model Deployment

Microsoft SQL Server 2012 introduced a new command-line tool (MDSModelDeploy.exe) to handle the export and import scenarios of models. The command-line tool enables the execution of creating and loading large models that take a long time.

The Cumulative Update #1 release of Microsoft SQL Server 2012 will include a set of performance improvements to these scenarios.

With those improvements, the performance results that were measured for a model with the same schema as the Customer (Long) model included the following:

- 100K members: 1 min. to create a package and 11 min. to load the package back to MDS
- 500K members: 18 min. to create the package and 60min. to load the package back to MDS.