JeffOps

The Scripting Dutchman

Data Deduplication in Windows Server 2012 explained from A to Z

25 Jan 2012 @ 9:44 AM

http://jeffwouters.nl/index.php/2012/01/disk-deduplication-in-windows-8-explained-from-a-to-z/

Last week I was at the Hyper-V.nu event at Microsoft Netherlands HQ in Amsterdam. Ronald Beekelaar (MVP Virtual Machine aka MVP Hyper-V) gave a Data Deduplication Deep Dive session.
This was a very good and highly technical session, which got me thinking… and I decided to write a little article about this new technology in Windows Server 2012 (Windows Server "8").

Introducing: Deduplication in Windows 8

With Windows Server 2012 Microsoft introduces a built-in software based data deduplication (dedupe) solution. Where several storage providers offer such solutions, Microsoft has taken another approach by providing a solution for duplicate data from an operating system level instead of a storage level. Where some deduplication solutions provide their services file-based, the deduplication offered in Windows Server 2012 does this block-based. More on that later on…

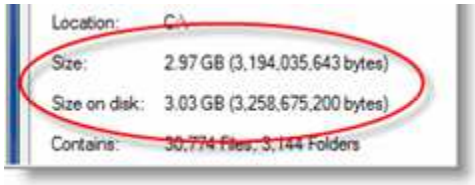
Now, let's take a few pointers before we start looking at dedupe in Windows Server 2012:

- Only available in Windows Server 2012.
- Is cluster aware.
- Based on a filter driver per volume.
- Not supported on boot- or system volumes, only intended for data storage volumes.
- Does not work on compressed or NTFS encrypted files.
- Dedupe requires an NTFS file system and is not supported for the new ReFS file system which is introduced in Windows Server 2012.
- Does not work with Cluster Shared Volumes.
- Does not work with encrypted files, files smaller than 64KB, re-parse points or files with extended attributes.
- Not configurable through Group Policy.
- It is a post-process deduplication process.
- Windows caching is dedupe aware.

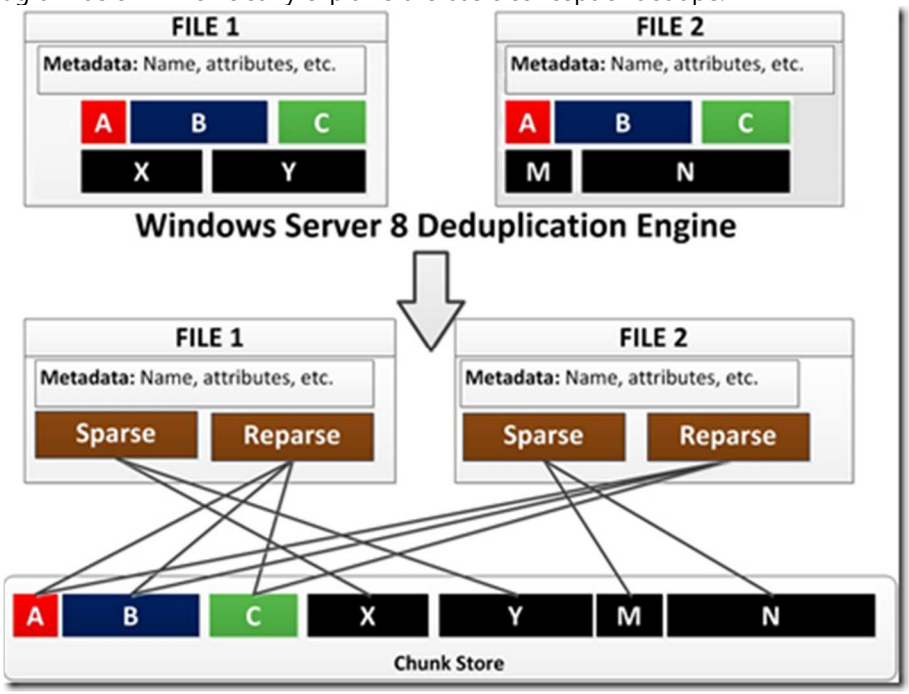
How does it work?

For me this is always the most fun question to ask… because when you know how it works, you can understand the use case and the possible gotcha's when designing an environment that makes use of this technology.
Dedupe looks at the storage from a block-based point of view and divides the storage on 'chunks' which are typically somewhere between 32 and 128 KB in size with an average of 80K, although smaller chunks are possible.
To understand dedupe in Windows Server 2012, we first have to understand the concept of 'hard links'.

When data is stored on a file system, the actual bits and bytes are stored on a single location. So, if some bits are the same… why save it multiple times? By using hard links you can refer to bits which can be used by multiple files.
Let's clarify that one a little… When you have hundreds of *.docs files created by your HR department, they probably use some templates. This means that a lot bits and bytes in the files is exactly the same!
Since dedupe views the storage in chunks, it will notice a lot of those being exactly the same. So, instead of saving the bits and bytes multiple times, it saves the chunk only one time and creates hard links on all locations so they refer to the same data.
When you view the properties of the Program Files folder, you will probably notice that the values behind "Size" and "Size on disk" differ from each other.



This is because some hard links are used for files in this folder. So "Size on disk" involves the accumulated amount of bits and bytes by the files in this folder and "Size" equals the accumulates amount of bits and bites on the disk minus the bits and bites that are replaced by hard links.
Note that the example provided above is file based, where dedupe is block based and provides a far better utilization of the available storage.
I found the diagram below which clearly explains the basic concept of dedupe.



As you can see, some chunks (A, B and C) are used by both files.
By using a technology similar to hard links, but on block-level, all files can access the correct bits and bytes where they only need to be stored once instead of multiple times.
The dedupe process works through scheduled tasks, but can be run interactively by using PowerShell. More about that command later on…

Why use data deduplication?
A valid question… what benefits does dedupe provide? A lot of my customers require massive amounts of storage.
The purpose of dedupe is the better utilize the storage capacity which is available to you.
Microsoft has done some research in their dedupe technology and come up with some numbers on the storage savings dedupe provided:

General	50-60% savings

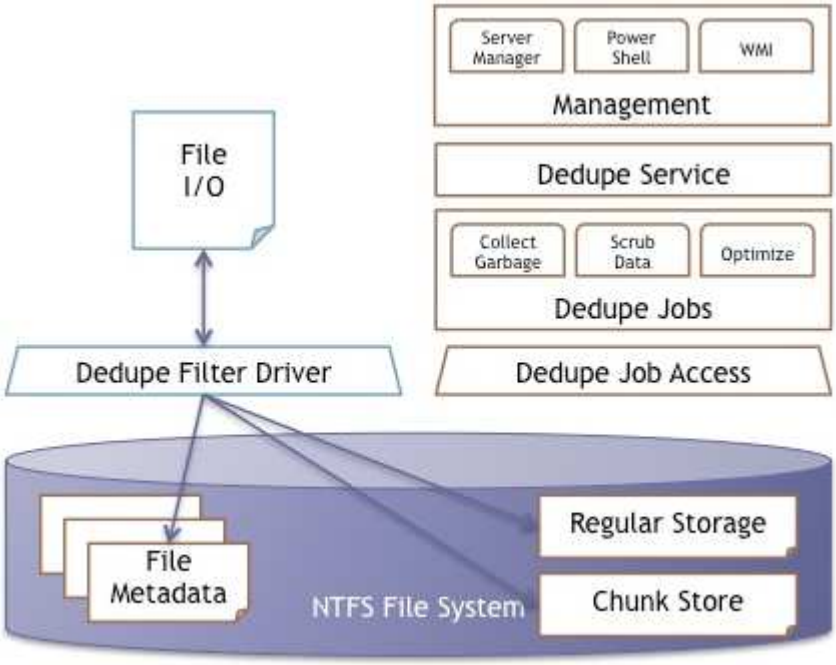
Documents	30-50% savings
Application Library	70-80% savings
VHD Library	80-95% savings

These numbers come straight from a vendor and these tests may be somewhat optimized for better results.
** As an economy teacher of mine always said:"You give me raw data and the results you want to come out of it, and I'll provide you a calculation that offers the results you want…".*
Nevertheless these are some pretty impressive numbers! I would love to test this in a production environment and hopefully see the eyes of some IT guys grow, aswell as the smiles of IT managers, when they see the storage savings in their environment

But what about the performance? Any dedupe technology causes some sort of a performance hit, right? Yeah, that's true… also with the dedupe in Windows Server 2012.
Microsoft has offered some information about this.
Write actions have no direct performance hit since the dedupe process is done in the background when the system is idle.
Read actions do have a performance hit, around 3% when the file is not in cache.

The components of deduplication

Drivers are always 'fun' to troubleshoot and since the entire technology of deduplication in Windows Server 2012 is based on the filter driver, some understanding of the thing may be useful.

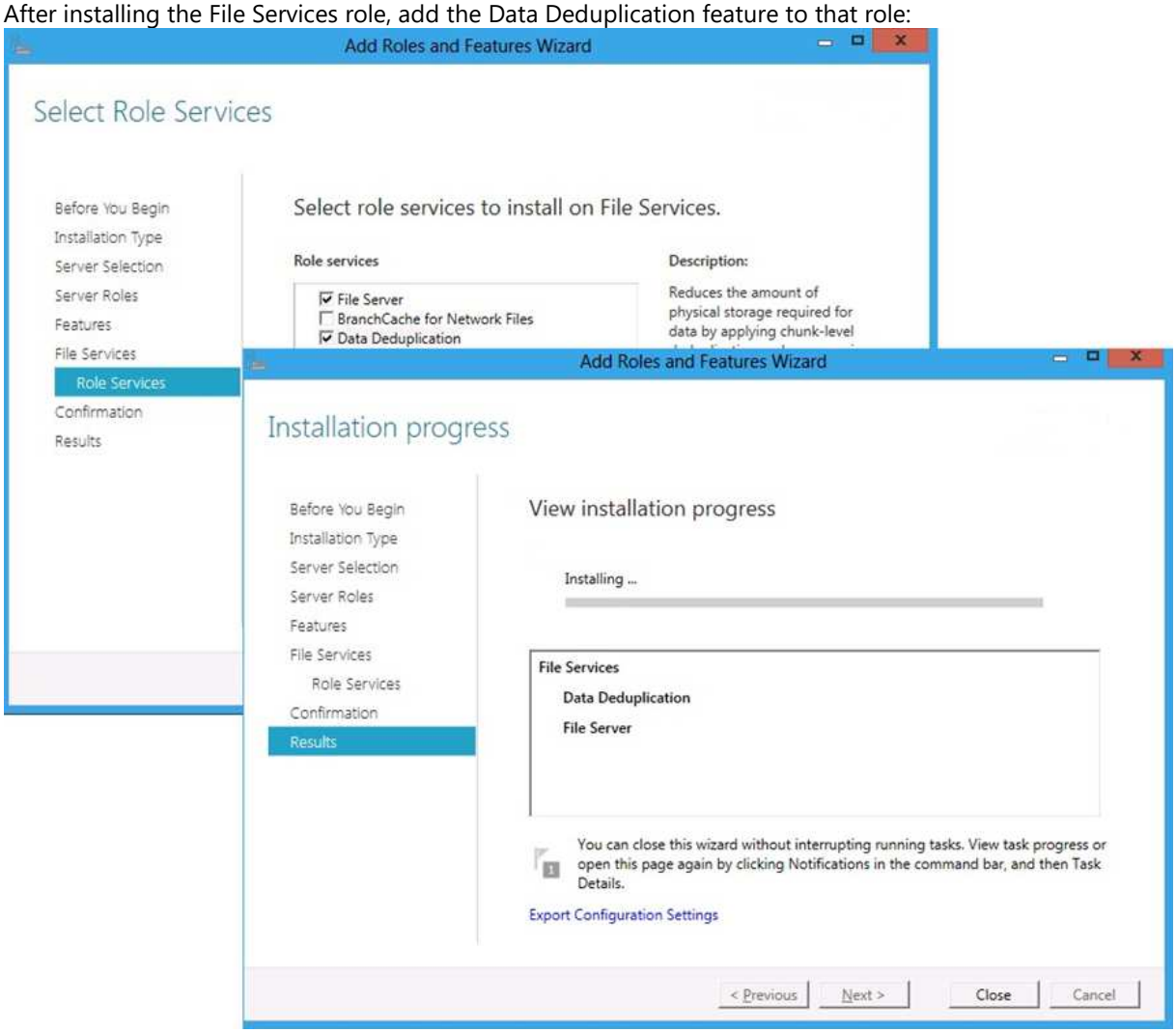


To do this, we have to look at the technology from an architectural view. Where the management of dedupe can be done by Server Manager, PowerShell and WMI, it only manages the dedupe service which in its turn manages the dedupe jobs.
Those dedupe jobs are the ones that talk to the dedupe filter driver that does the actual handling of the chunks of data on the file system. But when data is only stored once, the files will have to know where their data has gone to. That's where the metadata comes in to play since this is where the location of all the bits and bytes is stored.
With 'normal files the metadata will only have references to the regular storage. But when a file is affected by the dedupe process, the metadata will not only refer to the regular storage but also to some chunks in the chunk store.
The dedupe service can be scheduled or can run in a background mode while it waits for the system to enter an idle mode so that the system will not experience a negative performance in production hours. This is also called a post-process dedupe mode.

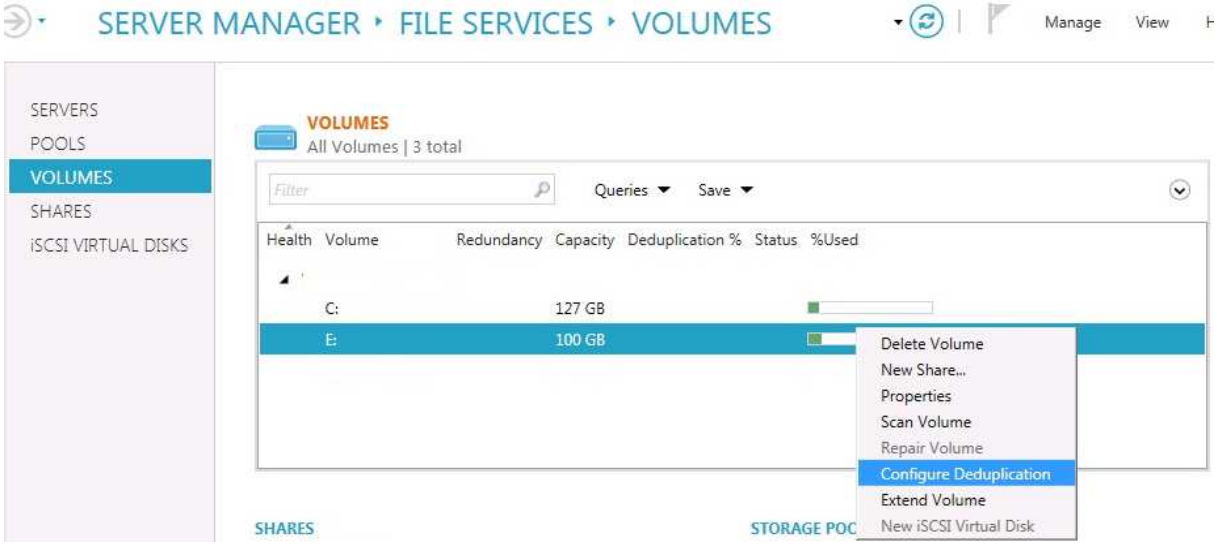
Dedupe and the GUI

The basic management features for dedupe are available in the GUI. Let's do a quick walkthrough for enabling and configuring the dedupe feature in Windows Server 2012.

After installing the File Services role, add the Data Deduplication feature to that role:

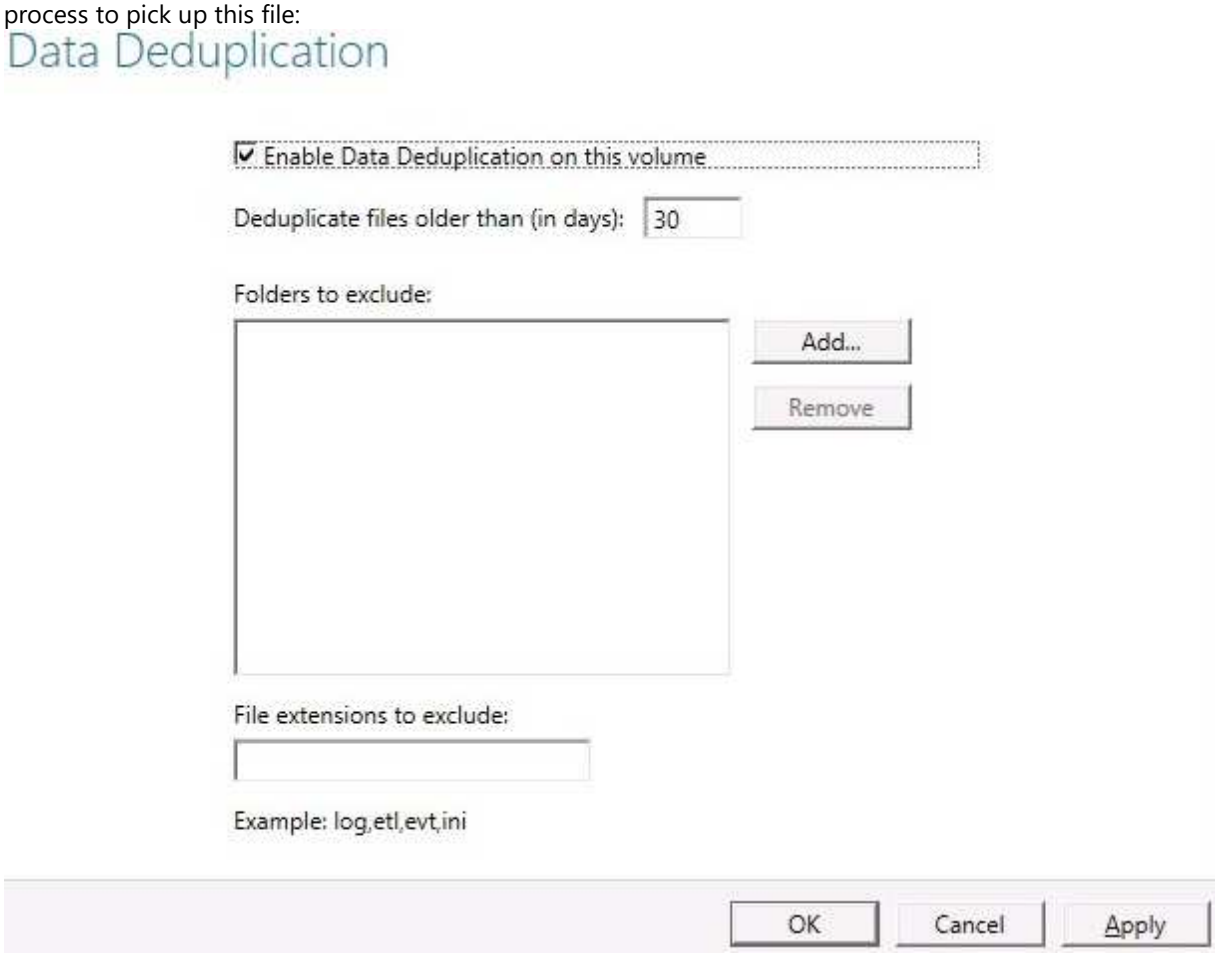


Next, you can configure dedupe on a volume:



Now we get the option to configure some settings in dedupe, such as files and folders to exclude… but more interesting is the setting for the minimum amount of days a file must not been changed for the dedup

process to pick up this file:



Dedupe and PowerShell

To enable dedupe we have to use my favorite tool: PowerShell.
The first task is add the deduplication feature which is part of the file system role. This can be done by using the Server Manager (GUI)… but where's the fun in that? You can't automate that… but by using PowerShell you can

To enable the deduplication feature by using (elevated) PowerShell commands:

```
Import-Module ServerManager
Add-WindowsFeature -name FS-Data-Deduplication
```

Now that the deduplication feature has been enabled, we can start configuring.
First, as with any other PowerShell module, we have to load the module. You can do this with the following command:

```
Import-Module Deduplication
```

To configure the dedupe feature on volume E on a device:

```
Enable-DedupVolume E:
```

Now that dedupe has been enabled and configured on a volume, we want to know some statistics such as what amount of storage we actually saved by using dedupe:

```
Get-DedupStatus
```

By default, the dedupe process will only affect files that have not been changed for 30 days. Especially in demo environments this can be a nasty gotcha… you probably don't want to wait 30+ days for dedupe to start doing its thing…
So, to change this value to 0 (process the file a.s.a.p.) you can use the following command:

```
Set-DedupVolume E: -MinimumFileAgeDays 0
```

Normally the dedupe process is done through scheduled tasks in the Windows operating system… but you can start this process manually with PowerShell:

```
Start-DedupJob E: –Type Optimization
```

However, this job runs in the background and may take some time. To view the status of that job, the following command can be used:

```
Get-DedupJob
```

HELP!!! I've done something wrong and I have to disable dedupe on this volume!!

Don't get your nickers in a twist… again, this can be done by using PowerShell  Use this command to un-dedupe the volume:

```
Start-DedupJob -Volume E: -Type Unoptimization
```

If you are as enthusiastic about this feature as I am you can read the help for the dedupe PowerShell cmdlets by using this command:

```
Help Dedup
```