ReFS

Resilient File System (**ReFS**),[5] codenamed "Protogon",[6] is a Microsoftproprietary file system introduced with Windows Server 2012 with the intent of becoming the "next generation" file system after NTFS.

ReFS was designed to overcome problems that had become significant over the years since NTFS was conceived, which are related to how data storage requirements had changed. The key design advantages of ReFS include automatic integrity checking and data scrubbing, elimination of the need for running chkdsk, protection against data degradation, built-in handling of hard disk drive failure and redundancy, integration of RAIDfunctionality, a switch to copy/allocate on write for data and metadata updates, handling of very long paths and filenames, and storage virtualization and pooling, including almost arbitrarily sized logical volumes (unrelated to the physical sizes of the used drives).

These requirements arose from two major changes in storage systems and usage – the size of storage in use (large or massive arrays of multi-terabyte drives now being fairly common), and the need for continual reliability. As a result, the file system needs to be self-repairing (to prevent disk checking from being impractically slow or disruptive), along with abstraction or virtualization between physical disks and logical volumes.

ReFS was initially added to Windows Server 2012 only, with the aim of gradual migration to consumer systems in future versions; this was achieved as of Windows 8.1.[7] The initial versions removed some NTFS features, such as disk quotas, alternate data streams, and extended attributes. Some of these were re-implemented in later versions of ReFS.

In early versions (2012–2013), ReFS was similar to or slightly faster than NTFS in most tests,[8] but far slower when full integrity checking was enabled, a result attributed to the relative newness of ReFS.[9][10]

The ability to create ReFS volumes was removed in Windows 10's 2017 Fall Creators Update for all editions except Enterprise and Pro for Workstations.[4]

The cluster size of a ReFS volume is either 4 KiB or 64 KiB.[11]

ReFS

Developer(s)	Microsoft
Full name	Resilient File System
Introduced	1 August 2012[1]with Windows Server 2012
Structures	
Directory contents	B+ tree[2]
Limits	
Max. volume size	1 yobibyte (2^{80}bytes)
Max. file size	16 exbibytes (2^{64}–1 bytes)
Features	
Attributes	Yes
Transparent compression	No
Data deduplication	Yes, since v3.2 debuting in 2016 v1709[3]
Copy-on-write	Yes
Other	
Supported operating systems	Windows Server 2012 and later Windows 8.1 and later (ReFS volume creation ability removed in Windows 10's 2017 Fall Creators Update, except for Windows 10 Enterprise and Pro for Workstations[4])
Website	docs.microsoft.com/en-US/windows-server/storage/refs/refs-overview

Contents

Feature changes compared to NTFS

Major new features

Improved reliability for on-disk structures

ReFS uses B+ trees for all on-disk structures, including all metadata and file data.[2][12] Metadata and file data are organized into tables similar to a relational database. The file size, number of files in a folder, total volume size and number of folders in a volume are limited by 64-bit numbers; as a result, ReFS supports a maximum file size of 16 exbibytes (2^{64}–1 bytes), and a maximum volume size of 35 petabytes[7]

Built-in resilience

ReFS employs an allocation-on-write update strategy for metadata,[2] which allocates new chunks for every update transaction and uses large IO batches. All ReFS metadata have 64-bit checksums which are stored independently. The file data can have an optional checksum in a separate "integrity stream", in which case the file update strategy also implements allocation-on-write for file data; this is controlled by a new "integrity" attribute applicable to both files and directories. If file data or metadata become corrupt, the file can be deleted without taking the whole volume offline for maintenance, and then be restored from the backup. As a result of built-in resiliency, administrators do not need to periodically run error-checking tools such as CHKDSKwhen using ReFS.

Compatibility with existing APIs and technologies

ReFS supports only a subset of NTFS features – and only Win32 APIs that are "widely adopted" – but does not require new system APIs, and most file system filters continue to work with ReFS volumes.[2] ReFS supports many existing Windows and NTFS features such as BitLockerencryption, Access Control Lists, USN Journal, change notifications,[13] symbolic links, junction points, mount points, reparse points, volume snapshots, file IDs, and oplock. ReFS seamlessly integrates with Storage Spaces,[2] a storage virtualization layer that allows data mirroring and striping, as well as sharing storage pools between machines.[14] ReFS resiliency features enhance the mirroring feature provided by Storage Spaces and can detect whether any mirrored copies of files become corrupt using a data scrubbing process,[12] which periodically reads all mirror copies and verifies their checksums, then replaces bad copies with good ones.

Removed features

Some NTFS features are not implemented in ReFS. These include object IDs, 8.3 filename, NTFS compression, Encrypting File System (EFS), transactional NTFS, hard links, extended attributes, and disk quotas.[6][2][15] In addition, Windows cannot be booted from a ReFS volume.[2] Dynamic disks with mirrored or striped volumes are replaced with mirrored or striped storage pools provided by Storage Spaces; however, automated error-correction is only supported on mirrored spaces. Data deduplication was missing in early versions of ReFS.[2] It was implemented in v3.2, debuting in Windows Server v1709.[3]

Support for alternate data streams was initially not implemented in ReFS. In Windows 8.1 64-bit and Server 2012 R2 the file system reacquired support for alternate data streams, with lengths of up to 128K, and automatic correction of corruption when integrity streams are used on parity spaces.[16] ReFS had initially been unsuitable for Microsoft SQL Server instance allocation due to the absence of alternate data streams.[17]

Version history and compatibility

ReFS has some different versions, with various degrees of compatibility between operating system versions. Aside for development versions of the filesystem, usually, later operating system versions can mount filesystems created with earlier OS versions (backwards compatibility). Some features may not be compatible with the feature set of the OS. The version, cluster size and other features of the filesystem can be queried with the command *fsutil fsinfo refsinfo volumename*.

- **1.1**: The original version, formatted by Windows Server 2012.
- **1.2**: Default version if formatted by Windows 8.1, Windows 10 v1507 to v1607, Windows Server 2012 R2, and when specified ReFSv1 on Windows Server 2016. Can use alternate data streams under Windows Server 2012 R2.
- **2.2**: Default version formatted by Windows 10 Preview build 10049 or earlier. Could not be mounted in 10061 and later.
- **2.0**: Default version formatted by Windows Server 2016 TP2 and TP3. Could not be mounted in Windows 10 Build 10130 and later, or Windows Server 2016 TP4 and later.
- **3.0**: Default version formatted by Windows Server 2016 TP4 and TP5.
- **3.1**: Default version formatted by Windows Server 2016 RTM.
- **3.2**: Default version formatted by Windows 10 v1703 and Windows Server Insider Preview build 16237. Can be formatted with Windows 10 Insider Preview 15002 or later (though only became the default somewhere between 15002 and 15019). Supports deduplication in the server version.
- **3.3**: Default version formatted by Windows 10 Enterprise v1709 (ReFS volume creation ability removed from all editions except Enterprise and Pro for Workstations starting with build 16226; read/write ability remains[4]) and Windows Server version 1709 (starting with Windows 10 Enterprise Insider Preview build 16257 and Windows Server Insider Preview build 16257).
- **3.4**: Default version formatted by Windows 10 Pro for Workstations/Enterprise v1803 and newer, also server versions.
- **3.5**: Default version formatted by Windows 10 Enterprise Insider Preview (build 19536 or newer); adds support for hard links (only on fresh formatted volume; not supported on volumes upgraded from previous versions).[7]

Support for different ReFS versions, by each Windows version[18]

ReFS	Windows Server 2012	Windows 8.1, Server 2012 R2	Windows 10 v1507 – v1607	Windows Server 2016 TP2, TP3	Windows Server 2016 TP4, TP5	Windows Server 2016 RTM	Windows 10 v1703	Windows 10 v1709, Windows Server 1709[5]	Windows 10 v1803 – v1809, Windows Server 2019, 1803 – 1809[5]
1.1	**Default**	Yes[1]	Yes[1]	Yes[1]	Yes[1]	Yes[1]	Yes[1]	?	?
1.2	Yes	**Default**	**Default**	Yes	Yes	Yes	Yes	Yes	Yes
2.0	No	No	No	No	**Default**	No	No	No	No
3.0	No	No	No	No	No	Yes[2]	Yes[3]	Yes[4]	Yes[6]
3.1	No	No	No	No	No	**Default**	Yes[3]	Yes[4]	Yes[6]
3.2	No	No	No	No	No	No	**Default**	Yes[4]	Yes[6]
3.3	No	No	No	No	No	No	No	**Default**	Yes[6]
3.4	No	No	No	No	No	No	No	No	**Default**

Notes:

[1]: The following message is recorded to the event log: 'Volume "?:" was mounted in an older version of Windows. Some features may be lost.'
[2]: Windows upgrades it to 3.1 when the volume is mounted with write access.
[3]: Windows upgrades it to 3.2 when the volume is mounted with write access.
[4]: Windows upgrades it to 3.3 when the volume is mounted with write access.
[5]: ReFS volume creation ability removed in Windows 10 v1709 (2017's Fall Creators Update), except for Enterprise and Pro for Workstations editions.
[6]: Windows upgrades it to 3.4 when the volume is mounted with write access.

Stability and known problems

Issues identified or suggested for ReFS, when running on Storage Spaces, include:

- Adding thin-provisioned ReFS on top of Storage Spaces (according to a 2012 pre-release article) can fail in a non-graceful manner, in which the volume without warning becomes inaccessible or unmanageable.[19] This can happen, for example, if the physical disks underlying a storage space became too full. *Smallnetbuilder* comments that, in such cases, recovery could be "prohibitive" as a "breakthrough in theory" is needed to identify storage space layouts and recover them, which is required before any ReFS recovery of file system contents can be started; therefore it recommends using backups as well.[19]
- Thin-provisioned ReFS on top of Storage Spaces, occasionally if the ReFS partition is extended to the full size of the thin-volume. When later extending the thin-volume size the ReFS partition may fail to extend to the thin-volume size. Once it fails to extend the partition you can never again extend the partition, no matter how large you extend the thin-volume size too. The workaround is to never extend the ReFS partition to the full size of the thin-volume, always leave a few GB at the end of the volume unassigned. It is thought that during the extension of the partition, it writes values into the partition table that corrupt the table when using the full size of the thin-volume, preventing any further expansions. That data is still intact and the ReFS partition works normally. A

known solution is to create a new volume and ReFS partition and copy out of the old ReFS and into the new ReFS requiring double storage during the copy before the old volume can be deleted.
- Even when Storage Spaces is not thinly provisioned, ReFS may still be unable to dependably correct all file errors in some situations, because Storage Spaces operates on blocks and not files, and therefore some files may potentially lack necessary blocks or recovery data if part of the storage space is not working correctly. As a result, disk and data addition and removal may be impaired, and redundancy conversion becomes difficult or impossible.
- There are no tools to repair or recover a ReFS filesystem. Third party tools are dependent on reverse engineering the system and (as of 2014) few of these exist.[20]
- Windows Store cannot install apps on a ReFS volume.[21]
- If integrity streams are enabled and there is no data redundancy available (disk-level parity or file-level backup), ReFS will remove the file from the namespace and the entire file will be unrecoverable if even a single bit is incorrect.

Server 2016 updates

At the Storage Developer Conference 2015, a Microsoft developer presented enhancements of ReFS expected to be released with Windows Server 2016 and included in Technical Preview 4, titled "ReFS v2".[22] It highlighted that ReFS now included capabilities for very high speed moving, reordering, and cloning of blocks between files[23](which can be done for all blocks of a file). This is particularly needed for virtualization, and is stated to allow fast provisioning, diff merging, and tiering. Other enhancements cover the redo log (for synchronous disk writes), parallelization, efficient tracking of uninitialized sparse data and files, and efficient 4k I/O.[22] ReFS with File Integrity enabled also acts more like a log-structured file system, coalescing small random writes into large sequential ones for efficiency.[24]

Performance and competitor comparisons

Other operating systems have competing file systems to ReFS, of which the best known are ZFS and Btrfs, in the sense that all three are designed to integrate data protection, snapshots, and silent high-speed background healing of corruption and data errors.

In 2012, Phoronix wrote an analysis[25] of ReFS vs Btrfs, a copy-on-write file system for Linux. Their features are similar, with both supporting checksums, RAID-like use of multiple disks, and error detection/correction. However, ReFS lacks copy-on-write snapshots and compression, both found in Btrfs and ZFS.

In 2014, a review of ReFS and assessment of its readiness for production use concluded that ReFS had at least some advantages over two of its main file system competitors.

- ZFS (used in Solaris, illumos, FreeBSD and others) was widely criticized for its comparatively extreme memory requirements of many gigabytes of RAM for online deduplication. However, online deduplication is never enabled by default in ZFS and was not supported at the time by ReFS (it has since been added), so not enabling ZFS online deduplication yielded a more even comparison between the two file systems as ZFS then has a memory requirement of only a few hundred megabytes.[26]
- Offerings such as Drobo used proprietary methods which have no fallback if the company behind them fails.[27]

Reverse engineering and internals

As of November 2019, Microsoft has not published any specifications for ReFS, nor have any working open-source drivers been made. A third-party open-source project to document ReFS is on GitHub.[28][29]

Paragon Software Group provides a closed-source driver for Windows and Linux.

See also

- Comparison of file systems
- APFS
- WinFS

References

1. Snover, Jeffrey (1 August 2012). "Windows Server 2012 released to manufacturing!". *Windows Server Blog*. Microsoft Corporation. Retrieved 2 September 2014.
2. Verma, Surendra (16 January 2012). Sinofsky, Steven (ed.). "Building the next generation file system for Windows: ReFS". *Building Windows 8*. Microsoft. MSDN blogs. Archived from the original on 31 July 2012. Retrieved 20 January 2013.
3. Sarkar, Dona (13 July 2017). "Announcing Windows Server Insider Preview Build 16237". *Windows Experience Blog*. Microsoft. Retrieved 20 July 2017.
4. "Windows 10 features and functionality removed in Windows 10". *docs.microsoft.com*. 30 January 2020. Retrieved 23 February 2020. "Creation ability will be available in the following editions only: Windows 10 Enterprise and Windows 10 Pro for Workstations. Creation ability will be removed from all other editions. All other editions will have Read and Write ability."
5. Lucas, Martin (1 January 2013). "Windows Server 2012: Does ReFS replace NTFS? When should I use it?". *Ask Premier Field Engineering (PFE) Platforms*. Microsoft. TechNet blogs. Retrieved 20 January 2013.
6. Foley, Mary Jo (16 January 2012). "Microsoft goes public with plans for its new Windows 8 file system". *ZDNet*. CBS Interactive. Retrieved 21 January 2013.
7. "Resilient File System (ReFS) overview". *Microsoft*.
8. "ReFS vs NTFS Benchmark – First Ever – Exclusive Technology News". *FirstEver.eu*. Archived from the original on 8 July 2014.
9. "Relative Performance of Parity .vs. Mirror in Storage Spaces". *Baris Eris's Blog*.
10. Eckert, Jason (25 February 2014). "Microsoft ReFS vs Oracle ZFS – Fight!".
11. Storage at Microsoft: Cluster size recommendations for ReFS and NTFS
12. Meyer, David (17 January 2012). "Microsoft takes wraps off new Windows file system". *ZDNet*. CBS Interactive. Retrieved 20 January 2013.
13. "ReadDirectoryChangesW function". *MSDN Library*. Microsoft. 8 November 2012. Retrieved 21 January 2013.
14. Nagar, Rajeev (5 January 2012). Sinofsky, Steven (ed.). "Virtualizing storage for scale, resiliency, and efficiency". *Building Windows 8*. Microsoft. MSDN blogs. Archived from the original on 6 January 2012. Retrieved 18 January 2013.
15. Windows Server 2012: Does ReFS replace NTFS? When should I use it? – Martin Lucas, *TechNet*
16. "Resilient File System Overview". *TechNet Library*. Microsoft. 29 February 2012. Retrieved 25 December 2013.
17. Cherry, Denny (21 August 2012). "SQL Server and Windows Server 2012's ReFS File System | Troubleshooting SQL Server Storage Problems". *SQL Server Pro magazine*. Penton Media.
18. Windows ReFS versions.en.md

19. Pakhomova, Elena (8 August 2012). "Data Recovery Tales: When Windows Storage Spaces Go Bad".
20. "Reverse Engineered ReFS parser".
21. "What I learned after 30 minutes with ReFS". *Ctrl blog*. 18 May 2017.
22. J. R. Tipton. "ReFS v2 – Cloning, projecting, and moving data" (PDF).
23. "Block Cloning (Windows)". *MSDN*. Microsoft. Retrieved 2 December 2016.
24. Anton Kolomyeytsev (12 April 2016). "ReFS: Log-Structured".
25. "Microsoft's ReFS File-System: Competitor To Btrfs?". *phoronix.com*.
26. "How To Size Main Memory for ZFS Deduplication". *www.oracle.com*. Retrieved 5 July 2018.
27. "Windows Storage Spaces and ReFS: Is it time to ditch RAID for good?". *BetaNews*.
28. "libyal/libfsrefs". *GitHub*. 18 October 2019. Retrieved 11 November2019.
29. ReFS documentation project - PDF document of the ReFS filing system (direct link)

External links

- Analysis of detailed differences between NTFS and ReFS in Server 2012, and reasons for choosing one or the other
- ReFS documentation project - PDF document of the ReFS filing system