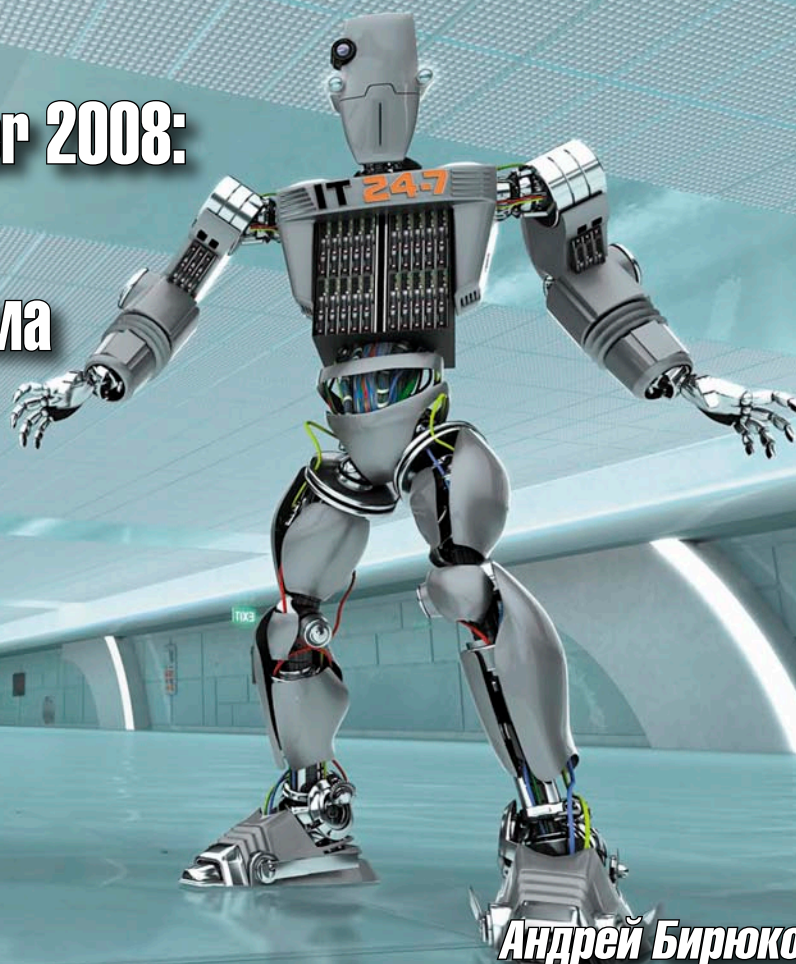


# Новшества в Windows Server 2008: транзакционная файловая система



**Андрей Бирюков**

**Появление в Windows Server 2008 транзакционной файловой системы является важным нововведением. Обсудим подробнее, что из себя представляет TxF.**

Вышедшая весной этого года операционная система Windows Server 2008 содержит целый ряд новых интересных функций, об одной из которых я хочу рассказать в этой статье. Файловая система является одним из ключевых элементов любой операционной системы, так как по сути это фундамент, на котором строится весь функционал ОС. В Windows Server 2008 таким фундаментом является транзакционная файловая система.

Транзакционная файловая система (TxF) – это расширение файловой системы NTFS, позволяющее выполнять файловые операции над томом файловой системы NTFS в рамках транзакций. Для читателей, не посвященных в специфику работы с базами данных, слово «транзакция» может показаться непонятным и требующим до-

полнительных пояснений. В рамках работы файловой системы транзакционность означает целостность выполнения операций с файлами. Другими словами, в любой момент времени состояние файловой системы всегда имеет целостную структуру, то есть даже при внезапном сбое питания на жестком диске не будет ошибок, связанных с нарушением целостности файлов, свойственных предыдущим версиям файловой системы. А, как известно, сбой при работе с системными файлами могли привести к очень печальным последствиям, вплоть до полной потери данных. Транзакционная файловая система использует для своей работы транзакции, то есть законченные операции с файлами, что позволяет добиться вышеупомянутой целостности файловой системы. Таким образом, функционал TxF позволяет

существенно увеличить стабильность работы как приложений, так и системы в целом. Еще одним дополнительным преимуществом TxF является возможность дальнейшей доработки и усовершенствования данной технологии, что позволит обеспечить ей дальнейшее развитие.

## Область применения

Приведа краткое пояснение о том, что же из себя представляет транзакционная файловая система, хотелось бы сказать несколько слов о том, для каких задач предназначена данная технология и, что тоже немаловажно, для чего она не подходит. TxF будет очень полезна при использовании на серверах, выполняющих частые операции по изменению файлов. Например, на серверах баз данных или файловых хранилищах. На таких

серверах частые обращения к файлам в многопользовательском режиме могут привести к потере данных при сбое электропитания, TxF поможет избежать данной проблемы.

Теперь о том, что не поддерживается. TxF не поддерживает сетевые диски (протоколы CIFS/SMB), также не поддерживается кэширование файловых операций на стороне клиента. Нельзя использовать TxF при работе с файловыми системами, отличными от NTFS (к FAT32 это тоже относится!). TxF не поддерживает операции над зашифрованной файловой системой (Encrypted File System, EFS) за исключением операций чтения (например, ReadEncryptedFileRaw).

## Технический функционал

Продолжая разговор о транзакционной файловой системе, немного углубимся в технические аспекты ее функционирования. Итак, реализация TxF стала возможной благодаря новой транзакционной инфраструктуре, реализованной на уровне ядра операционной системы, позволяющей сервисам ОС участвовать в транзакциях, используя новый компонент – менеджер транзакций Kernel Transaction Manager (KTM). Тут следует отметить, что KTM может взаимодействовать напрямую со службой Microsoft Distributed Transaction Coordinator (DTC), которая управляет транзакциями операционной системы и приложений. Это также позволяет существенно увеличить функционал TxF. Помимо этого в обеспечении функционирования транзакционной файловой системы задействована подсистема протоколирования Common Log File System (CLFS), впервые реализованная в Microsoft Windows Server 2003 R2.

За счет того, что TxF способна полностью взаимодействовать с MS Distributed Transaction Coordinator, она имеет возможность участвовать в транзакциях, использующих не только менеджеры ресурсов, предоставляемые Kernel Transaction Manager, но и другие менеджеры ресурсов, поддерживаемые на уровне DTC. Например, система документооборота может использовать эту возможность для работы как с файловой системой, так и с базой данных и все в рамках одной транзакции. Другой пример ис-

пользования транзакционной файловой системы – обновление файлов на группе компьютеров.

## Операции над файлами, поддерживаемые TxF

Сжатие. Если вы сжимаете файл, то операция по сжатию файла не является транзакционной. Но TxF будет работать со сжатым файлом, так же как и с обычным.

При создании новых файлов или каталогов доступ к создаваемому файловому ресурсу блокируется для других транзакций. И если в то же самое время другая транзакция попытается создать ресурс с таким же именем, то она получит сообщение о конфликте: ERROR\_TRANSACTIONAL\_CONFLICT, и завершится с ошибкой.

Работа с удаляемыми файлами и каталогами TxF осуществляется следующим образом. Файл будет удален только тогда, когда работу с ним завершат все транзакции. При удалении каталогов доступ к его содержимому блокируется для доступа всех транзакций.

А вот если вы вносите изменения в файл внутри каталога, то доступ к каталогу не блокируется для других транзакций, и они могут вносить любые изменения в каталог, но не в изменяемый файл, так как доступ к нему блокируется на время изменений.

Что касается административных задач, то на этом их описание завершается, оставшаяся часть статьи будет интересна преимущественно программистам и разработчикам, так как речь пойдет о программировании TxF.

Вот список функций используемых в TxF.

- CopyFileTransacted;
- CreateDirectoryTransacted;
- CreateFileTransacted;
- CreateHardLinkTransacted;
- CreateSymbolicLinkTransacted;
- DeleteFileTransacted;
- FindFirstFileNameTransactedW;
- FindFirstFileTransacted;
- FindFirstStreamTransactedW;
- GetCompressedFileSizeTransacted;
- GetFileAttributesTransacted;
- GetFullPathNameTransacted;
- GetLongPathNameTransacted;
- MoveFileTransacted;
- RemoveDirectoryTransacted;
- SetFileAttributesTransacted.

В качестве одного из параметров при вызове перечисленных выше функций указывается ссылка на транзакцию, в рамках которой выполняется данная операция. Транзакция может быть создана либо вызовом функции CreateTransaction при использовании Kernel Transaction Manager или функции GetKTMHandle при использовании DTC.

Появление транзакционной файловой системы привело к внесению ряда изменений в работу некоторых функций, например, CloseHandle, CreateFileMapping, FindNextFile и других. Последовательность действий при транзакционной работе с файловой системой может быть следующей:

- создание транзакций на уровне ядра:

```
IntPtr tx = CreateTransaction(
    IntPtr.Zero,
    IntPtr.Zero, 0, 0, 0, null);
```

- транзакционное удаление файла:

```
If (!DeleteFileTransactedW(
    file1, tx))
```

- завершение транзакции при успешном выполнении предыдущего пункта:

```
CommitTransaction(tx);
```

- откат транзакции при ошибке:

```
RollBackTransaction(tx);
```

- закрытие ссылки на транзакцию:

```
CloseHandle(tx);
```

На этом я завершаю свое краткое описание транзакционной файловой системы Transactional NTFS. Думаю, что появление новой файловой системы будет полезно не только разработчикам, но также и администраторам, так как позволит повысить надежность работы корпоративных ресурсов в целом. 🌐

1. Алексей Федоров «Microsoft Windows Server 2008. Краткий обзор ключевых новинок».
2. [http://msdn.microsoft.com/en-us/library/bb968806\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb968806(VS.85).aspx) – раздел MSDN, посвященный Transactional NTFS.