

# Миграция приложений, предназначенных для обработки данных, в Windows Azure

Кан Чен Селчин Тёркаслан Ноберто Гарсия  
Стив Ховорд Шон Тинлайн-Джонс  
Средхар Пеллюру Сильвано Кориани  
Джейми Алва Браво

## Справочник\руководство



Microsoft

# Миграция приложений, предназначенных для обработки данных, в Windows Azure

Кан Чен, Селчин Тёркаслан, Ноберто Гарсия, Стив Ховорд, Шон Тинлайн-Джонс, Средхар Пеллюру, Сильвано Кориани, Джейми Алва Браво

**Обзор:** При переносе приложения с локальной платформы в облако следует решить несколько задач. В этой книге описаны рекомендации по миграции локальных приложений, в основе которых лежит работа с данными. В частности, в книге описывается вопрос планирования миграции, например, способы снижения времени простоя при переходе; рассматриваются различные способы миграции: на основе виртуальных машин Windows Azure (IaaS-ориентированный подход) или облачных служб (PaaS-ориентированный подход); приводятся рекомендации по переносу кода с SQL Server на SQL Database и повышению производительности базы данных SQL Database, а так же обеспечение высокой доступности решения.

**Категория:** Справочник\руководство

**Аудитория:** Windows Azure

**Источник:** [MSDN Library](#)

**Дата публикации электронной книги:** Сентябрь 2012

*This page intentionally left blank*

Все права защищены. 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.

Миграция приложений, предназначенных для обработки данных, в Windows Azure .....	10
В этом разделе .....	10
См. также .....	11
Общие сведения о жизненном цикле миграции.....	11
Анализ .....	12
Миграция приложений .....	14
Перенос данных .....	14
Тестирование и оптимизация .....	15
Работа и управление .....	15
См. также .....	15
Планирование миграции .....	16
Планирование затрат .....	16
Определение основных бизнес-требований и технических требований, которые могут быть удовлетворены с помощью Windows Azure .....	17
Выполнение анализа и проектирования .....	18
Планирование временной шкалы .....	22
Создание плана промежуточных действий.....	22
Создание плана тестирования .....	23
Определение необходимых ресурсов.....	23
Планирование управления приложениями в Windows Azure .....	24
См. также .....	24
Реализация плана миграции.....	25
Настройка проверочных тестов .....	25
Синхронизация баз данных для уменьшения времени перехода.....	27
Синхронизация данных Windows Azure SQL .....	27
Репликация, зеркальное отображение и доставка журналов.....	28
Настраиваемые операции извлечения, преобразования и загрузки данных (ETL) .....	29
Экспорт приложения уровня данных (DAC) .....	29
Резервное копирование и восстановление.....	29
Переход на Windows Azure .....	30
См. также .....	31
Миграция с помощью виртуальных машин Windows Azure .....	31
В этом разделе .....	31
См. также .....	32
Общие сведения о виртуальных машинах Windows Azure .....	32
Windows Azure и виртуальные машины .....	32
Образы виртуальных машин, поставляемые с Windows Azure.....	34
Список важных понятий.....	36

Высокий уровень доступности и аварийное восстановление при использовании виртуальных машин Windows Azure.....	37
Общие шаблоны приложений при использовании виртуальных машин Windows Azure ..	37
Поддержка интерфейсов API для виртуальных машин Windows Azure .....	38
См. также .....	38
Миграция с помощью SQL Server в виртуальные машины Windows Azure .....	38
Когда следует использовать SQL Server на виртуальных машинах Windows Azure .....	39
Выбор платформы: SQL Server в виртуальной машине Windows Azure или база данных SQL Windows Azure.....	40
Когда следует использовать на виртуальных машинах Windows Azure перенесенные в SQL Server схемы базы данных и данные.....	41
Как подготовить локальные схему и данные и передать их в экземпляр SQL Server на виртуальной машине. ....	42
Вариант 1. Использование файлов BACPAC или DACPAC приложения уровня данных ..	42
Вариант 2. Резервное копирование и восстановление .....	43
Вариант 3. Использование команд Attach и Detach.....	44
Вариант 4. Использование других функций SQL Server .....	45
Перемещение схемы базы данных и файла данных на виртуальную машину Windows Azure.....	46
См. также .....	48
Миграция с помощью облачных служб Windows Azure.....	48
В этом разделе .....	48
См. также .....	49
Рекомендации по разработке для облачных служб Windows Azure .....	50
Создание облачной службы в Windows Azure.....	51
Разработка приложений в Windows Azure .....	52
Регистрация, тестирование, диагностика и отладка приложений Windows Azure .....	53
Возможности подключения к облачной сети в Windows Azure.....	55
Развертывание приложений и управление приложениями в Windows Azure.....	57
См. также .....	57
Общие сведения о службах управления данными в Windows Azure.....	58
Служба таблиц Windows Azure .....	59
Служба больших двоичных объектов Windows Azure.....	59
Диск Windows Azure.....	59
База данных SQL Windows Azure.....	59
Сравнение базы данных SQL Windows Azure и табличного хранилища.....	60
См. также .....	64
Миграция баз данных SQL Server в базу данных SQL Windows Azure .....	65
Общие сведения о процессе миграции.....	65

В этом разделе .....	66
Управление проектом миграции базы данных SQL Windows Azure .....	67
Введение .....	68
Анализ .....	69
Бизнес-требования .....	69
Определите цель подпроекта базы данных .....	70
Оценка сложности проекта.....	70
Зависимости от функций, не поддерживаемых базами данных SQL Windows Azure ..	70
Масштаб изменений приложений, обусловленных изменениями в базе данных .....	71
План и проект .....	72
Разработка.....	75
Тест .....	77
Стабилизация .....	79
Развертывание .....	79
Вопросы производительности базы данных SQL Windows Azure .....	80
Управление подключениями в База данных SQL Windows Azure.....	80
Варианты обработки управления соединениями в коде .....	82
Краткое рассмотрения способа использования программного блока обработки	
временных сбоев.....	82
Сетевая задержка в База данных SQL Windows Azure.....	89
Минимизируйте расстояние между данными и приложениями.....	90
Уменьшение сетевых циклов приема-передачи .....	90
Соблюдайте общие рекомендации для локальных баз данных .....	92
Рекомендации для ODBC и JDBC .....	93
Рекомендации для ADO.NET .....	94
Рекомендации для Entity Framework.....	94
Рекомендации по асинхронному программированию .....	95
Вопросы высокого уровня доступности и аварийного восстановления с помощью базы	
данных SQL Windows Azure.....	96
Как защитить базы данных от сбоя отдельных серверов, устройств или потери связи с	
сетью .....	96
Как защитить свою базу данных от нежелательного удаления или изменений.....	97
Защита базы данных от масштабных разрушений в центре обработки данных.....	99
Выбор средств для миграции базы данных в базу данных SQL Windows Azure.....	100
Выбор средств миграции .....	100
Средства для миграции базы данных, отличной от SQL Server .....	100
Средства миграции между службами базы данных SQL .....	101
Средства для миграции базы данных SQL Server.....	101
Сравнение средств миграции SQL Server .....	105

Как использовать мастер миграции SQL Azure .....	107
До начала работы.....	108
Рекомендации .....	108
Ограничения .....	109
Предварительные требования .....	110
Перенос базы данных с помощью мастера миграции базы данных SQL.....	111
Анализ базы данных с помощью мастера миграции баз данных SQL .....	111
Анализ файла Transact-SQL с помощью мастера миграции базы данных SQL.....	111
Анализ файла трассировки с помощью мастера миграции базы данных SQL.....	112
Ресурсы .....	112
Как использовать SQL Server Data Tools для миграции базы данных в базу данных SQL	
Windows Azure.....	112
До начала работы.....	112
Рекомендации .....	113
Ограничения .....	114
Предварительные требования .....	114
Использование средств SSDT с пакетами приложения уровня данных .....	114
Использование средств SSDT с прямым соединением.....	115
См. также .....	115
Как использовать пакет DAC BACPAC для миграции базы данных в базу данных SQL	
Windows Azure.....	116
До начала работы.....	116
Рекомендации .....	116
Ограничения .....	117
Предварительные требования .....	117
Перенос файла DAC BACPAC .....	118
Ресурсы .....	119
Как использовать пакет приложения уровня данных для миграции базы данных в базу	
данных SQL Windows Azure.....	119
До начала работы.....	120
Рекомендации .....	120
Ограничения .....	120
Предварительные требования .....	121
Миграция пакета приложения уровня данных .....	121
Ресурсы .....	123
Как создать скрипты для миграции базы данных в базу данных SQL Windows Azure.....	123
До начала работы.....	124
Рекомендации .....	124
Предварительные требования .....	124



Миграция базы данных с помощью мастера формирования скриптов .....	124
Ресурсы .....	125
Как использовать bsr для миграции базы данных в базу данных SQL Windows Azure.....	125
До начала работы.....	126
Рекомендации .....	126
Ограничения .....	127
Предварительные требования .....	127
Использование программы bsr для переноса данных .....	128
Ресурсы .....	129
Как использовать службы Integration Services для миграции базы данных в базу данных SQL Windows Azure.....	129
До начала работы.....	130
Рекомендации .....	130
Ограничения .....	131
Предварительные требования .....	131
Миграция базы данных с помощью служб SQL Server Integration Services .....	131
Ресурсы .....	132
Как использовать мастер импорта и экспорта для миграции базы данных в базу данных SQL Windows Azure.....	132
Мастер импорта и экспорта SQL Server .....	133
Рекомендации .....	133
Ограничения .....	133
Предварительные требования .....	134
Перенос данных с помощью мастера импорта таблиц .....	134
Ресурсы .....	136
Как использовать помощник миграции SQL Server с базой данных SQL Windows Azure ....	136
До начала работы.....	136
Предварительные требования .....	136
Перенос базы данных с использованием помощника миграции SQL Server.....	137
Ресурсы .....	137
Вопросы переноса секционированных данных в базу данных SQL Windows Azure.....	138
Перенос существующих локальных функций в базу данных SQL .....	139
Репликация .....	139
Секционирование таблиц.....	140
Секционированные представления .....	141
Распределенные секционированные представления .....	141
Стратегии файловых групп.....	141
Межбазовые запросы.....	142

Перенос данных в другие службы управления данными в Windows Azure .....	142
В этом разделе .....	150
Перенос данных в хранилище таблиц .....	150
Вопросы миграции .....	151
Вопросы данных .....	151
Сравнение хранилища таблиц и базы данных SQL Windows Azure .....	152
Вопросы доступа к данным .....	153
Преимущества использования хранилища таблиц .....	153
Отправка существующих данных в хранилище таблиц Azure .....	154
См. также .....	154
Перенос данных в хранилище больших двоичных объектов .....	154
Сравнение блочных и страничных больших двоичных объектов .....	156
Вопросы миграции .....	156
Вопросы данных .....	156
Сравнение хранилища больших двоичных объектов и базы данных SQL Windows Azure .....	157
Вопросы доступа к данным .....	158
Преимущества использования хранилища больших двоичных объектов .....	159
Перенос существующих данных в хранилище больших двоичных объектов Azure .....	160
Перенос данных на диски .....	160
Вопросы миграции .....	161
Передача данных в диск Windows Azure .....	162
См. также .....	162
Вопросы миграции в службу кэша Windows Azure .....	162
Вопросы миграции .....	163
Windows Azure Caching (Preview) .....	166
См. также .....	168
Миграция приложений, которые используют технологии обмена сообщениями .....	168
Миграция MSMQ .....	170
Шина обслуживания Windows Azure .....	170
Очередь Windows Azure .....	171
Рабочая роль Windows Azure .....	171
Миграция приложений для использования локального хранилища .....	171
Вопросы миграции .....	172
См. также .....	172

# Миграция приложений, предназначенных для обработки данных, в Windows Azure

---

При переносе приложения с локальной платформы в облако следует решить несколько задач. В этом подразделе описаны лучшие рекомендации и руководства при миграции локальных приложений и баз данных в Windows Azure.

## В этом разделе

### [Общие сведения о жизненном цикле миграции](#)

Даются пошаговые инструкции, позволяющие выполнить миграцию приложений и данных в Windows Azure.

### [Планирование миграции](#)

Рассматриваются несколько вопросов и действия, которые следует иметь в виду при планировании миграции в Windows Azure.

### [Реализация плана миграции](#)

Дается руководство по выполнению заключительных шагов миграции в Windows Azure.

### [Миграция с помощью виртуальных машин Windows Azure](#)

Даются общие сведения о виртуальных машинах (ВМ) Windows Azure и руководство по переносу существующих баз данных SQL Server на такой же сервер, но на виртуальных машинах Windows Azure.

### [Миграция с помощью облачных служб Windows Azure](#)

Даются общие сведения об облачных службах Windows Azure и руководство по переносу существующих приложений и баз данных в Windows Azure с помощью функций платформы облачных служб Windows Azure.

## См. также

[Planning and Designing Windows Azure Applications](#)

[Developing Windows Azure Applications](#)

[Testing, Managing, Monitoring and Optimizing Windows Azure Applications](#)

[Node.js Developer Guidance \(Windows Azure\)](#)

[PHP Developer Guidance \(Windows Azure\)](#)

[Руководство для Java-разработчиков \(Windows Azure\)](#)

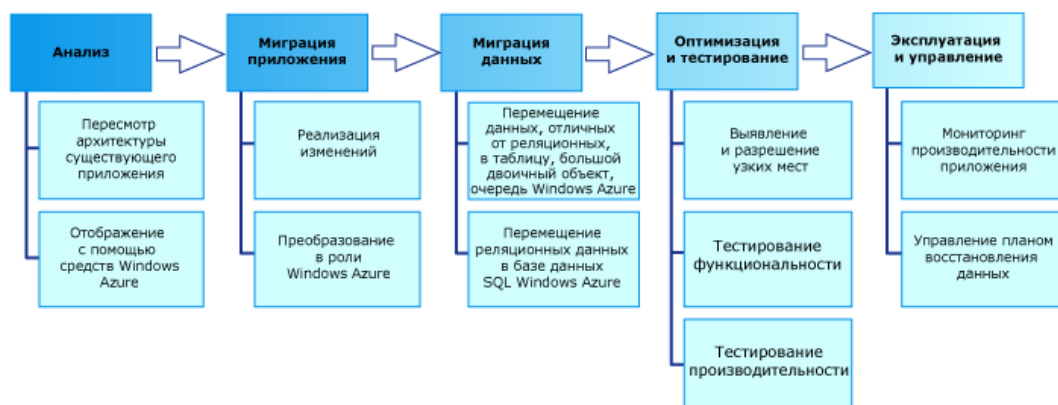
[Other Third Party Software on Windows Azure](#)

## Общие сведения о жизненном цикле миграции

**Авторы:** Кун Чен (Kun Cheng), Селсин Туркарслан (Selcin Turkarslan), Норберто Гарсиа (Norberto Garcia)

**Рецензенты:** Паоло Сальватори (Paolo Salvatori), Стив Говард (Steve Howard), Стюарт Озер (Stuart Ozer)

Жизненный цикл миграции — это стандартная методология, которая обеспечивает пошаговые инструкции для миграции приложений и переноса данных в Windows Azure. Основные шаги миграции включают анализ, миграцию приложений, перенос данных, тестирование и оптимизацию, работу и управление, как показано на приведенной далее диаграмме.



В этом разделе подробно объясняется каждый этап и приводятся ссылки на дополнительную информацию.

## Анализ

Цель этого этапа — понять бизнес-потребности, которые требуют решения Windows Azure. После определения бизнес-целей изучите существующую архитектуру приложения и выявите основные отличия между Windows Azure и локальными решениями. Затем решите, нужно ли изменять существующее локальное приложение для обеспечения бизнес-потребностей решения Windows Azure. Следующие задачи и вопросы помогут создать план облачной миграции.

- **Определение бизнес-требований.** Существует несколько потенциальных вопросов бизнес-сценария, если приложение выполняется в Windows Azure.

Ориентировано ли решение по разворачиванию Windows Azure на новых клиентов и пользователей?

Потребуется ли мультитенантность для обеспечения поддержки нескольких клиентов?

Отвечает ли приложение нормативным правилам при размещении данных в дата-центре корпорации Майкрософт, а не на клиентских сайтах?

Какие приложения больше подходят для облака с точки зрения архитектуры и стратегии?

Какая служба лучше всего подходит для приложений и баз данных: облачные службы Windows Azure или виртуальные машины Windows Azure?

Ответы на эти вопросы влияют на разработку приложения и его поведение на платформе Windows Azure.

- **Определение функциональных отличий.** Можно ли запустить существующее приложение в облаке без каких-либо изменений? Например, база данных SQL Windows Azure (база данных SQL) не поддерживает все функции, которые поддерживаются локальной системой SQL Server. Если необходимо переместить локальное приложение, использующее среду CLR, в базу данных SQL, потребуется изменить приложение и перенести логику среды CLR из SQL Server на уровень приложений или переписать логику среды CLR с помощью инструкций Transact-SQL, которые поддерживаются базой данных SQL. Обратите внимание, что база данных SQL в настоящее время не поддерживает SQL CLR.

Начиная с CTP-версии Windows Azure 2012, в Windows Azure были добавлены новые возможности виртуальных машин. Виртуальные машины Windows Azure позволяют

выполнить миграцию существующих приложений SQL Server, построенных на платформе Windows Server, на платформу Windows Azure с минимальными изменениями кода либо без изменений. Используя SQL Server на виртуальной машине Windows Azure, администраторы и разработчики по-прежнему могут использовать те же доступные локально средства разработки и администрирования. Производительность реляционной базы данных на виртуальной машине Windows Azure зависит от многих факторов, включая размер виртуальной памяти, количество и конфигурации дисков, сети, конфигурации программного обеспечения базы данных и рабочей нагрузки приложения. Рекомендуется, чтобы разработчики провели оценку размеров виртуальной машины и конфигурации хранения и выбрали наиболее подходящий вариант. Дополнительные сведения см. в разделе [Миграция с помощью SQL Server в виртуальные машины Windows Azure](#).

- **Подготовка плана производительности и масштабируемости.** Многие прежние версии приложений были разработаны для тесной интеграции между логикой приложений и компонентами доступа к данным. Для прежних версий приложений имеет смысл разъединить компоненты приложения для лучшей работы и увеличения масштабируемости в Windows Azure. Если приложение отправляет слишком много запросов, изучите возможность использования [службы кэширования Windows Azure](#) или реализуйте собственный механизм кэширования для пакетного доступа к данным и уменьшения обращений приложения к данным. Если приложение, для которого нужно выполнить перенос, работает с большими базами данных или большим объемом транзакций, для перенесения в базу данных SQL, скорее всего, потребуется изменение модели базы данных. Это вызвано тем, что один экземпляр базы данных SQL может управлять ограниченным числом транзакций в секунду и имеет ограниченный размер базы данных. При работе с большими базами данных или большим объемом транзакций изучите возможность реализации горизонтально масштабируемой архитектуры с помощью нескольких баз данных в базе данных SQL или начните использовать федерации баз данных SQL вместо дорогих локальных систем вертикального масштабирования.
- **Подготовка плана для управления жизненным циклом приложений.** Важно рассмотреть версии приложений и сценарии обновления на Windows Azure. В зависимости от соглашения об уровне обслуживания может потребоваться поддержка нескольких версий приложения и поддержка различных уровней клиентов. Также можно свести к минимуму время простоя при обновлении приложения на платформе Windows Azure. Рекомендуется должным образом поддерживать промежуточную и рабочую среды на Windows Azure. Убедитесь, что существует возможность отката обновлений в случае возникновения проблем с совместимостью. План отката обновлений должен сначала устранить проблемы в работе приложения, затем в работе базы данных.

После этого этапа рекомендуется построить пилотный проект, поскольку он даст четкое понимание служб и средств платформы Windows Azure.

# Миграция приложений

После принятия решения выполнить миграцию приложения в облако начните с версии пилотного проекта приложения с минимальными данными с целью сборки экспериментальной версии. Во-первых, реализуйте необходимые изменения в коде приложения с целью выполнения целей развертывания Windows Azure в отношении бизнес- и технических требований. После этого скомпилируйте и выполните развертывание кода приложения в соответствующие роли в Windows Azure.

В общем, большинство существующих локальных приложений можно запустить в облачных службах Windows Azure с минимальными изменениями или без изменений, но это может создать проблемы производительности, масштабируемости и безопасности. Для оптимизации производительности и обеспечения возможности будущей масштабируемости рекомендуется рассмотреть вариант изменения приложения с помощью нескольких ролей перед миграцией в облачные службы Windows Azure.

Дополнительные сведения см. в разделе [Рекомендации по разработке для облачных служб Windows Azure](#). Рекомендуется сначала переместить в облачные службы Windows Azure приложение, а затем данные. Из соображений безопасности, производительности или по другим причинам некоторым частям приложения может потребоваться локальное размещение. Для этого требуется гибридное решение. Дополнительные сведения см. в разделе [Построение гибридных решений с помощью Windows Azure](#).

Если принято решение использовать SQL Server на виртуальной машине Windows Azure, измените существующие приложения SQL Server так, чтобы они подключались к базе данных SQL Server виртуальной машины Windows Azure. Дополнительные сведения о миграции существующих баз данных SQL Server в SQL Server виртуальной машины Windows Azure см. в разделе [Миграция с помощью SQL Server в виртуальные машины Windows Azure](#).

## Перенос данных

Если используются облачные службы Windows Azure, переместите реляционные данные из локальной системы SQL Server в базу данных SQL, а неструктурированные данные — в хранилище Windows Azure, такое как хранилище больших двоичных объектов, хранилище таблиц или диски Windows Azure. Дополнительные сведения см. в разделах [Перенос данных в другие службы управления данными в Windows Azure](#) и [Миграция баз данных SQL Server в базу данных SQL Windows Azure](#).

Если принято решение использовать SQL Server в виртуальных машинах Windows Azure, см. раздел [Миграция с помощью SQL Server в виртуальные машины Windows Azure](#), чтобы узнать, как

выполнить миграцию существующей базы данных SQL Server на виртуальную машину SQL Server в Windows Azure.

## Тестирование и оптимизация

После миграции приложения и данных в облако выполняйте функциональный тест и тест производительности. На этом этапе протестируйте приложение в облаке и убедитесь, что оно работает надлежащим образом. Затем сравните результаты производительности между локальным экземпляром и Windows Azure. Затем устраните проблемы, связанные с функциональностью, производительностью или масштабируемостью облачного приложения. Дополнительные сведения см. в разделе [Реализация плана миграции](#).

## Работа и управление

После тестирования и оптимизации создайте и реализуйте мониторинг приложения и трассировку с помощью службы диагностики Windows Azure. Служба диагностики Windows Azure позволяет собирать диагностические данные из приложения, запущенного в Windows Azure. Диагностические данные можно использовать для отладки и устранения неполадок, измерения производительности, наблюдения за использованием ресурсов, анализа трафика, планирования вычислительной мощности и для аудита. Дополнительные сведения см. в разделе [Diagnostics and Debugging in Windows Azure](#) в библиотеке MSDN.

Если нужно синхронизировать данные между локальным экземпляром и базой данных SQL или между различными серверами баз данных SQL, установите и настройте службу **SQL Data Sync**. Кроме того, рекомендуется установить и настроить план восстановления данных в случае ошибки пользователя или стихийных бедствий. Дополнительные сведения см. в разделе [Вопросы высокого уровня доступности и аварийного восстановления с помощью базы данных SQL Windows Azure](#).

## См. также

[Миграция приложений, предназначенных для обработки данных, в Windows Azure](#)



# Планирование миграции

---

**Авторы:** Стив Ховард (Steve Howard)

**Рецензенты:** Джеймс Подгорски (James Podgorski), Паоло Сальватори (Paolo Salvatori), Сельцин Туркаслан (Selcin Turkarslan), Стюард Озер (Stuart Ozer)

В начале планирования миграции необходимо рассмотреть несколько ключевых факторов, таких как затраты, бизнес-требования и технические требования, временная шкала и тестирование, необходимое во время процесса миграции. В этом разделе подробно рассматриваются несколько вопросов и действия, которые должны быть предусмотрены при планировании миграции в Windows Azure.

[Планирование затрат](#)

[Определение основных бизнес-требований и технических требований, которые могут быть удовлетворены с помощью Windows Azure](#)

[Выполнение анализа и проектирования](#)

[Планирование временной шкалы](#)

[Создание плана промежуточных действий](#)

[Создание плана тестирования](#)

[Определение необходимых ресурсов](#)

[Планирование управления приложениями в Windows Azure](#)

## Планирование затрат

При рассмотрении миграции локального приложения в Windows Azure одним из самых больших, требующих разрешения вопросов является вопрос затрат, поэтому ответ на этот вопрос рекомендуется сделать на ранней стадии процесса принятия решений и планирования. Определение цены приложения для Windows Azure зависит от многих факторов, например: уровня трафика в сети, характеристик ввода-вывода приложения и объема данных, обрабатываемых приложением. Расчет цены выходит за рамки этого раздела. В начале планирования миграции для оценки стоимости рекомендуется использовать калькулятор затрат Windows Azure. Калькулятор затрат Windows Azure находится [здесь](#).

При расчете затрат в организации необходимо включить прямые расходы для Windows Azure во время разработки и тестирования. В локальном проекте разработки нужно

платить за серверы разработки и тестирования. Аналогично в среде Windows Azure нужно платить за ресурсы, используемые во время разработки и тестирования. Кроме того, необходимо рассчитать затраты на обучение и затраты, связанные с переносом приложения в Windows Azure. Чтобы заранее оценить, какая производительная мощность должна быть у приложения, рекомендуется произвести тестирование производительности и планирование мощности. Чтобы понять затраты для типичного приложения Windows Azure, ознакомьтесь со статьей [Windows Azure Cost Assessment](#).

## Определение основных бизнес-требований и технических требований, которые могут быть удовлетворены с помощью Windows Azure

Платформа Windows Azure может очень хорошо обеспечить некоторые бизнес- и технические требования. Хотя приводимый далее перечень далеко не полон, приложения с такими характеристиками хорошо подходят для миграции в Windows Azure.

- **Распределенная база пользователей.** Центры обработки данных Windows Azure находятся на нескольких континентах. Взаимодействие центров обработки данных позволяет выполнять при необходимости высокопроизводительное распределение данных. Компоненты Windows Azure, такие как сеть доставки контента (CDN) и службы синхронизации данных, обеспечивают предоставление релевантных или интенсивно используемых данных путем распределения в ближайших к пользователю центрах обработки данных. Если пользователи связываются с центрами обработки данных, расположенными недалеко от их географического положения, длина каналов для обращения и обратного отклика уменьшается, тем самым оптимизируется взаимодействие пользователей.
- **Переменная нагрузка.** Чтобы локальные приложения справлялись с пиковыми нагрузками, необходимо приобрести дополнительное оборудование. Например, чтобы розничный магазин справлялся с повышенной нагрузкой в предпраздничные периоды, характерные большим объемом продаж, обычно приобретаются серверы. Аналогично, чтобы бухгалтерский отдел мог справляться с пиковыми нагрузками в конце месяца или года, может быть запланирована дополнительная инфраструктура. В остальное время серверы, предназначенные для обработки таких пиковых нагрузок, недогружены. С другой стороны, приложения, архитектура которых обеспечивает гибкое масштабирование, могут получить преимущество от Windows Azure благодаря подключению новых экземпляров приложений в сети в пиковые периоды и переходу к более низкому уровню мощности обработки в периоды меньшей нагрузки. Таким образом, при наличии приложений с соответствующей архитектурой и с должным уровнем управления Windows Azure позволяет платить только за необходимое.

- **Мультитенантность.** Для поставщиков услуг Windows Azure предоставляет несколько способов предоставления служб приложения любому количеству клиентов по одной инфраструктуре, уменьшая тем самым эксплуатационные расходы.
- **Необходимость концентрироваться на приложениях.** Поставщики услуг, в частности, стремятся направить свои усилия в основном на разработку приложений и функций, а не на поддержание инфраструктуры. Windows Azure освобождает вас от административных издержек, связанных с инфраструктурой, размещаемой в локальных приложениях или в традиционных размещаемых серверных приложениях. Эта платформа позволяет направить ресурсы на разработку приложений и функций, не отвлекаясь на другие задачи.
- **Уменьшение требований к инфраструктурным ресурсам.** При разработке приложений, которые могут использовать способность гибкого масштабирования, которую предоставляет Windows Azure, экземпляры ролей и ресурсы могут выделяться по потребности. Нет необходимости тратить на интерфейсное оборудование, не нужно поддерживать серверы для пиковых нагрузок в периоды небольшого использования.

Наряду с преимуществами платформы, ориентированной на использование служб, Windows Azure может размещать виртуальные машины. Эти виртуальные машины могут работать под управлением любой операционной системой, поддерживаемой платформой Windows Azure, и приложения могут выполняться в них так же, как они выполняются в локальных компьютерах. Список поддерживаемых операционных систем см. в разделе [Общие сведения о виртуальных машинах Windows Azure](#). Эти виртуальные машины могут быть также частью архитектуры более крупного приложения, которое может включать экземпляры веб-ролей и рабочих ролей, а также другие компоненты Windows Azure. Виртуальные машины предоставляют способ переноса некоторых служб или частей приложений, которые в ином случае не могли бы так несложно переноситься в Azure. Дополнительные сведения см. в разделе [Миграция с помощью виртуальных машин Windows Azure](#).

## Выполнение анализа и проектирования

На этапе анализа и проектирования необходимо определить приложения, которые планируется переносить в Windows Azure. Затем начните проектирование реализации Windows Azure и создайте план реализации. На этом этапе необходимо спланировать общий вид архитектуры и временной шкалы.

Далее приводятся некоторые ключевые элементы планирования.

- **Определение текущих задач.** В следующем списке приведены некоторые примеры задач, которые должны быть определены при планировании для повторного проектирования архитектуры.

- **Компоненты приложения, которые не работают согласно стандарту при текущих нагрузках на текущей архитектуре.** Например, если SQL-запрос выполняется неудовлетворительно, его необходимо уточнить перед миграцией или последующим проектированием. Необходимо также выполнить повторное проектирование и горизонтальное масштабирование компонентов на уровне приложений.
- **Определение требований к гибкому масштабированию.** Необходимо определить, как приложение может быть декомпозировано на функциональные, независимо масштабируемые подразделения, которые могут выполняться независимо друг от друга.
- **Шаблоны неравномерной нагрузки.** Необходимо определить схемы неравномерной нагрузки и спроектировать приложение с возможностью горизонтального масштабирования для обработки пиковых периодов. Необходимо составить планы, как управлять уровнем горизонтального масштабирования в периоды низкой потребности.
- **Увеличение проекций.** Часто увеличение проекций является первым предупреждением для ИТ-отдела о том, что необходимо изменение подхода. Решите, где горизонтальное масштабирование может быть решением проблемы увеличения проекций. Увеличение проекций может быть также индикатором того, что нужно рассмотреть смену подхода, например нужно перейти на подход анализа большого объема данных в некоторых приложениях с хранилищем данных. На этапе планирования необходимо обсудить эти возможности. Помните, что заранее, до процесса проектирования и реализации, решение может быть неизвестно. Необходимо составить список таких ситуаций и определяющих факторов, чтобы оценить их в подходящее время, например во время первоначальной миграции или позже.
- **Определение технических требований.** Необходимо определить требования для каждого компонента приложения в периоды пиковых и минимальных нагрузок. Затем спланируйте горизонтальное масштабирование каждого компонента. Для каждого компонента могут быть разные возможности и механизмы масштабирования. Технические требования могут относиться не только к производительности. Например, при планировании миграции необходимо определить и сравнить с возможностями Windows Azure требования для высокого уровня доступности и аварийного восстановления или требования максимальных задержек в сети. Далее перечисляются некоторые примеры технических требований.
- **Использование реляционного хранилища.** Проверьте данные, хранящиеся в реляционных базах данных. В реляционном хранилище должны находиться транзакционные и реляционные по своей природе данные или данные, для которых требуется истинно транзакционная обработка. Для хранения таких типов данных можно использовать базу данных SQL Windows Azure (базу данных SQL) или систему SQL Server, выполняющуюся в виртуальной машине. Другие типы данных можно хранить в таблицах Windows Azure, в хранилище больших двоичных

объектов Windows Azure или на дисках Windows Azure. Рекомендуется определить тип хранилища, необходимого для каждой части данных.

- **Выбор хранилища реляционных данных.** Выбор базы данных SQL или системы SQL Server, выполняющейся в виртуальной машине Windows Azure, зависит от нескольких факторов. Если нужно предотвратить административные издержки, связанные с высоким уровнем доступности, балансировкой нагрузки и с прозрачной обработкой отказа, наилучшим вариантом является база данных SQL. Но для промежуточного состояния во время переноса приложения или в специальных случаях, в которых функции в базе данных SQL недоступны, наилучшим решением может быть система SQL Server, выполняющаяся в виртуальных машинах Windows Azure. Ответы на эти вопросы зависят от ситуации и решения. Далее обсуждаются следующие вопросы.
- **Размер базы данных.** Базы данных SQL Windows Azure в настоящее время ограничены объемом 5 ГБ для базы данных версии Web Edition, а для базы данных версии Business Edition существует ограничение 150 ГБ. Для масштабирования базы данных сверх этого размера необходимо использовать федерации или сегментирование. Рекомендации и ограничения для федераций в базе данных SQL см. в разделе **Federation Guidelines and Limitations**. Этим обеспечивается большая вычислительная мощность при извлечении данных, но одновременно вводится ограничение на соединения и статистическую обработку, связанную с размещением данных. Подробные сведения о федерациях см. в разделе **Federations in SQL Database (SQL Database)**. Наиболее актуальные сведения о доступных версиях и размерах баз данных см. в разделе **Accounts and Billing in SQL Database**.
- **Количество баз данных.** По умолчанию база данных SQL поддерживает до шести серверов на одну подписку и 150 баз данных на каждом сервере баз данных SQL, включая базу данных master. Возможно расширение этого предела. Для получения дополнительных сведений обратитесь к представителю службы технической поддержки пользователей на пользовательском портале услуг Microsoft в сети.
- **Межбазовые запросы.** В настоящий момент база данных SQL не поддерживает межбазовые соединения или другие межбазовые запросы к базе данных. Если имеются объединения или соединения, которые требуют данных из нескольких баз данных в базе данных SQL, эту логику необходимо выполнить на уровне приложений вашего приложения.
- **Объекты среды CLR.** База данных SQL в настоящее время не поддерживает хранимые процедуры CLR, агрегаты, триггеры или функции. Необходимо перенести хранимые процедуры, триггеры или функции в Transact-SQL для их выполнения в базе данных SQL. Сложная логика или такие операции, как статистическая обработка, которые не выполняются в Transact-SQL на уровне базы данных, должны быть перенесены на уровень приложений. Для этого можно использовать рабочую роль.

- **Типы данных.** База данных SQL не поддерживает некоторые типы данных системы SQL Server. Наиболее актуальные сведения см. в статье **Data Types (SQL Database)** в библиотеке SQL MSDN.
- **Репликация.** Типы репликации, такие как репликация транзакций или репликация слиянием, недоступны в базе данных SQL. Их можно установить и запустить в системе SQL Server, выполняющейся в виртуальных машинах Windows Azure. Синхронизацию данных SQL можно использовать для синхронизации данных между экземплярами базы данных SQL. Но если требуется согласованность транзакций или устранение сложных конфликтов, служба синхронизации данных SQL может работать неудовлетворительно.  
**Предупреждение.** В настоящее время доступна только предварительная версия службы синхронизации данных SQL, предназначенная для составления отзывов для будущих версий. Использовать эту предварительную версию в рабочих средах не рекомендуется.
- **Полнотекстовый поиск.** База данных SQL Windows Azure в настоящий момент не поддерживает полнотекстовый поиск. Если приложение выполняет полнотекстовые запросы символьных данных в таблицах SQL Server, может понадобиться рассмотреть миграцию базы данных в SQL Server в виртуальной машине Windows Azure. Дополнительные сведения о предварительной версии SQL Server в виртуальной машине Windows Azure см. в разделе [Миграция с помощью SQL Server в виртуальные машины Windows Azure](#).
- **Лицензирование.** Каждый месяц взимается плата за базу данных SQL выбранного размера. Для SQL Server требуется лицензия при выполнении системы в виртуальной машине Windows Azure.
- **Вход и безопасность.** Аутентификация Windows (встроенная безопасность) в базе данных SQL не поддерживается, но доступна для системы SQL Server, выполняющейся в виртуальных машинах Windows Azure. Дополнительные рекомендации по безопасности и ограничения базы данных SQL см. в разделе **Security Guidelines and Limitations (SQL Database)**.
- **Сходные функции.** Дополнительные сведения о подоби и различиях SQL Server и базы данных SQL см. в разделе **SQL Database Overview**.
- **Вход и безопасность пользователя.** Благодаря новым сетевым улучшениям в Windows Azure домен Active Directory локальной сети может быть расширен до Windows Azure. Дополнительные сведения см. в разделе [Миграция с помощью виртуальных машин Windows Azure](#). Подробные сведения об администрировании безопасности базы данных SQL см. в разделе **Managing Databases and Logins in SQL Database**.
- **Функциональная декомпозиция приложения.** Определите, каким образом приложение может быть разбито на функциональные подразделения таким образом, чтобы оно могло выполняться в отдельных ролях или виртуальных машинах Windows Azure. Это можно сделать для создания гибкого масштабирования, а также для того,

чтобы разрешить гибридные приложения, если некоторые приложения не очень подходят для облачных вычислений.

- **Индустрия платежных карточек (PCI) и другие нормативные требования.** Перед перемещением приложения или компонента в Windows Azure проверьте текущее состояние необходимой сертификации или нормативных требований. При наличии требований совместимости с PCI может потребоваться удалить некоторые части приложения и базы данных, предназначенных для переноса в Windows Azure, и запустить гибридное приложение. Это позволяет сохранить преимущества Windows Azure и облачных вычислений в большинстве компонентов, обеспечивая при этом жесткий контроль нормативных органов и совместимость частей данных и приложения, для которого это необходимо.
- **Ключевые компоненты, которые не могут быть размещены на платформе Windows Azure.** Вследствие некоторых других проблем может быть невозможным разместить некоторые компоненты или некоторые типы данных в общедоступном облаке. Определите эти компоненты и рассмотрите использование гибридного приложения. С помощью гибридной архитектуры можно разместить некоторые компоненты в Windows Azure, чтобы реализовать все преимущества Windows Azure и облачных вычислений. В то же время обеспечивается жесткий контроль нормативных органов и совместимость частей данных и приложения, для которого это необходимо.

## Планирование временной шкалы

После определения объектов миграции становится ясным объем работы на каждом этапе плана миграции. Рассмотрите каждый компонент приложения и данных и оцените время и ресурсы, необходимые для разработки, тестирования и миграции. При функциональной декомпозиции приложения необходимо разработать разбиваемые компоненты и одновременно создать масштабируемые компоненты.

В плане миграции задайте основные этапы проекта, такие как тестирование производительности, а также даты версий. Миграция может производиться в несколько этапов и итераций, как только различные компоненты становятся готовыми для Azure или как только компоненты становятся готовыми для миграции в веб-роли и рабочие роли Windows Azure.

## Создание плана промежуточных действий

После задания временной шкалы создайте план роста в данный период времени и решите, что должно быть сделано в существующем приложении и инфраструктуре. Способ планирования позволяет управлять существующей системой до завершения миграции. При формировании этого промежуточного плана определите точки текущего плана и установите, что должно быть сделано для обеспечения непрерывных операций, определите, какие операции масштабирования могут продолжаться в промежуточной



инфраструктуре. Кроме того, определите шаги, которые могут быть необходимы для обеспечения непрерывных операций. Такие шаги часто могут быть очень простыми, например уточнение SQL-запроса или добавление веб-сервера в зависимости от характеристик определенного приложения. Определите планы на непредвиденные случаи при более быстром, чем предполагалось, росте или неожиданном внезапном росте. При составлении планов на непредвиденные случаи рассмотрите, можно ли управлять ростом путем горизонтального масштабирования в виртуальных машинах Windows Azure, поскольку это может позволить справляться с такими ситуациями без приобретения дополнительного оборудования.

## Создание плана тестирования

План миграции должен включать планы всестороннего тестирования функциональных возможностей и тестирования загрузки. Описание методологий тестирования выходит за рамки данной статьи. Далее перечисляются некоторые критически важные моменты, которые нужно помнить при тестировании.

- Автоматизируйте скрипты теста
- Протестируйте все уровни и компоненты приложения
- Протестируйте соотношения действий, которые представляют реальные соотношения в системах
- Выполняйте тесты вплоть до максимально ожидаемой степени использования или более

Рекомендуется отводить время на построение и выполнение тестов, а также на устранение проблем, найденных при тестировании.

## Определение необходимых ресурсов

При определении бизнес- и технических требований необходимо определить ресурсы, необходимые для успешного выполнения миграции. Необходимо их добавить для миграции. При определении ресурсов необходимо рассмотреть три основные области.

- **Персонал.** Для успешного выполнения миграции может понадобиться добавить дополнительных сотрудников с соответствующими навыками. Кроме того, после миграции роли технического персонала могут измениться, а для навыков может потребоваться обновление. Например, для управления входом, доступом и уровнями масштабирования рассмотрите роли администратора учетной записи и администратора служб.



- **Средства.** Определите средства, необходимые для разработки, теста и развертывания приложения Windows Azure. Дополнительные сведения см. в разделах [Средства Windows Azure для Microsoft Visual Studio](#) и **Tools and Utilities Support (SQL Database)**.
- **Консультирование.** Для миграции приложения может понадобиться консультация у специалиста по миграции. Помощь специалиста по миграции может сэкономить значительное время и финансовые ресурсы и решить общие проблемы.

## Планирование управления приложениями в Windows Azure

Для небольших приложений портал управления Windows Azure может быть достаточен для управления развертыванием Windows Azure. Портал управления Windows Azure позволяет войти в него и управлять развертыванием и приложениями, включая изменение количества ролей экземпляра, а также управлять экземплярами базы данных SQL. Однако для сложных приложений и приложений, предоставляющих услуги клиентам, портал управления Windows Azure может быть недостаточен.

Windows Azure предоставляет API-интерфейс REST, позволяющий программным путем управлять приложениями и виртуальными машинами, размещенными в Windows Azure. Кроме того, он обеспечивает провизионирование и использование хранилища Windows Azure. Для ведения масштабирования и наблюдением за средой Windows Azure можно самостоятельно написать интерфейс управления. План миграции должен включать план управления приложением после миграции, особенно если такое управление должно включать настраиваемый интерфейс или автоматизацию.

Дополнительные сведения об API-интерфейсе для управления развертыванием Windows Azure см. в разделе [Программные интерфейсы для Windows Azure](#).

## См. также

[Миграция приложений, предназначенных для обработки данных, в Windows Azure](#)

[Моделирование реальной нагрузки приложения Windows Azure](#)

[Использование нагрузочных тестов Visual Studio в ролях Windows Azure](#)

# Реализация плана миграции

---

**Авторы:** Кун Чен (Kun Cheng), Стив Говард (Steve Howard)

**Соавторы:** Селсин Туркарслан (Selcin Turkarslan)

Windows Azure — это интернет-платформа масштаба для вычислений и служб, размещенная в центрах обработки данных Майкрософт. С Windows Azure разработчикам и администраторам не требуется реализовывать базовую инфраструктуру программного обеспечения и оборудования, так как все ресурсы операционной системы, оборудования, сети, хранилища и обновления платформы автоматически обрабатываются Майкрософт.

Настоятельно рекомендуется после переноса приложения в облако выполнить функциональные тесты и тест производительности для приложения так же, как и для любого нового развернутого приложения. Поскольку Windows Azure отличается от локальной платформы, при реализации миграции необходимо учитывать следующие важные вопросы:

[Настройка проверочных тестов](#)

[Синхронизация баз данных для уменьшения времени перехода](#)

[Резервное копирование и восстановление](#)

[Переход на Windows Azure](#)

Обратите внимание, что этот раздел в основном посвящен облачным службам Windows Azure. Предварительные инструкции по миграции с SQL Server на виртуальных машинах Windows Azure см. в разделе [Миграция с помощью виртуальных машин Windows Azure](#).

## Настройка проверочных тестов

При переносе приложений в облако следует знать, как выполнять тест и отладку приложения, чтобы оно правильно работало в облаке. Ниже приведен список методов, которые можно использовать для теста приложения.

**Средства Windows Azure для Microsoft Visual Studio.** Можно создать приложение, а затем запустить и отладить его локально с помощью эмуляторов вычислений и хранилища, которые предоставляются как часть средств Windows Azure. Это позволяет разрабатывать приложения локально перед публикацией в Windows Azure. Средства Windows Azure для Microsoft Visual Studio расширяют возможности Visual Studio 2010 и позволяют тестировать приложение с помощью эмуляторов вычислений и хранилища, которые обеспечивают большую часть функций Windows Azure. Рекомендуется

выполнить этот тип проверки на ранних стадиях функционального тестирования.

Дополнительные сведения см. в разделе [Средства Windows Azure для Microsoft Visual Studio](#).

**SQL Server Data Tools.** Средства SSDT предоставляют интегрированную среду в Visual Studio 2010, которую можно использовать для разработки баз данных, создания или изменения объектов баз данных и данных или выполнения запросов для всех поддерживаемых платформ SQL, в том числе облачной базы данных SQL Windows Azure и локальной базы данных Microsoft SQL Server 2012. Это позволяет проверить решения проекта базы данных с помощью местной базы данных по умолчанию или базы данных SQL Windows Azure путем изучения части приложения, отвечающей за доступ к реляционным данным. Дополнительные сведения см. в разделе [SQL Server Data Tools](#). **Примечание.** Средства Windows Azure для Microsoft Visual Studio и SSDT позволяют выполнять основные тесты функциональности и совместимости приложения с источниками данных в сети и вне сети. Чтобы протестировать все аспекты реального облачного приложения с точки зрения функциональности, производительности и масштабируемости, необходимо провести тестирование с эмуляцией на платформе Windows Azure, где приложение развертывается и работает.

**Автоматизированная платформа тестирования.** Многие существующие приложения уже имеют автоматизированную платформу тестирования, которую можно использовать для проверки работы всех компонентов или функций приложения. Когда приложение работает в Windows Azure, автоматическая платформа теста может работать или нет в зависимости от того, как она была спроектирована. Если автоматическая платформа теста должна запускаться локально, но может подключиться к приложению в Windows Azure, используя определенные конечные точки, она может работать. В противном случае рекомендуется размещать автоматическую платформу теста и ваше приложение на платформе Windows Azure для смягчения потенциальных проблем потери соединения и задержек.

**Тестирование нагрузки Visual Studio.** Если у приложения нет автоматической платформы теста, рекомендуется создать новую платформу тестирования и использовать средства тестирования нагрузки Visual Studio для моделирования с несколькими параллельно работающими пользователями. Дополнительные сведения см. в разделе [Использование нагрузочных тестов Visual Studio в полях Windows Azure](#).

# Синхронизация баз данных для уменьшения времени перехода

Между тестированием, промежуточным хранением и производственным использованием следует попытаться свести к минимуму время перехода на новую платформу. На копирование большой базы данных из локальной системы в Windows Azure может уйти несколько часов или дней. Кроме того, сбой приложения не является желательным в течение полного периода, необходимого для полного переноса существующих данных. Поэтому следует разработать план для минимизации времени простоя из-за перехода. Обратите внимание, что под временем перехода подразумевается время, необходимое для окончательного перемещения в Windows Azure. Перед этим изучите таблицы, выясните, какие таблицы содержат статические данные, а какие таблицы содержат данные, которые могут измениться во время перехода. Статические данные не нужно перемещать во время перехода. Однако, если существует сомнение относительно возможных изменений данных в определенной таблице в течение перехода, следует включить в систему логику последующей миграции всех изменений. Также рекомендуется выяснить, следует ли переносить все данные из локальных систем в облако, прежде чем приложение станет активным в Windows Azure. Если приложение готово к запуску и позволяет обновить данные позднее, вы можете устранить любое время простоя.

Однако, если данные в Windows Azure должны соответствовать локальным данным перед запуском в Windows Azure, попробуйте минимизировать объем переносимых данных при переходе, так как это помогает сократить время простоя, необходимое для фактического перехода. В некоторых случаях может быть целесообразным переместить некоторые данные до перехода, а затем переместить оставшиеся данные. В таких случаях план миграции должен четко определить данные, которые нужно перенести первыми, а также оставшиеся данные, которые могут быть перенесены после перехода. Это позволяет запустить приложение в Windows Azure с меньшим временем простоя, так как оно может работать, пока выполняется перенос оставшихся данных. Можно использовать следующие методы синхронизации данных для переноса данных до перехода:

## Синхронизация данных Windows Azure SQL

Служба синхронизации данных Windows Azure SQL предоставляет возможности для синхронизации данных для баз данных SQL Windows Azure. В настоящее время служба обеспечивает две основные функции:

синхронизацию данных между локальными базами данных SQL Server и экземплярами баз данных SQL Windows Azure, обеспечивающую возможность использования одних и тех же данных как локальными, так и облачными приложениями;

синхронизацию данных двух и более баз данных SQL Windows Azure, которые могут находиться в одном центре обработки данных, в разных центрах обработки данных или в разных регионах.

Служба синхронизации данных Windows SQL Azure — это хороший вариант для синхронизации локальных баз данных и экземпляров баз данных SQL Windows Azure в следующих ситуациях.

Тестировать приложения следует параллельно.

Приложение следует запускать параллельно до окончательного перемещения всех локальных операций с данными в Windows Azure.

Во время миграции на Windows Azure запустите приложение локально, не забывая о необходимости минимизировать время простоя.

Выполняйте непрерывную синхронизацию данных как гибридного решения между локальной средой и приложением Windows Azure.

Обратите внимание, что для того, чтобы отслеживать добавочные изменения данных, служба синхронизации данных SQL добавляет таблицу отслеживания изменений для каждой синхронизируемой таблицы при настройке. При использовании службы синхронизации данных SQL необходимо оставить место для таблицы отслеживания изменений, чтобы обеспечить синхронизацию данных. Кроме того, не следует вносить изменения в структуры таблиц или первичные ключи таблиц, которые синхронизируются после настройки синхронизации, если группа синхронизации не инициализируется повторно. Служба синхронизации данных SQL не является оптимальным выбором для ситуаций, когда требуется промежуточная или постоянная синхронизация данных. Дополнительные сведения см. в разделе **SQL Data Sync. Предупреждение**. В настоящее время доступна только предварительная версия службы синхронизации данных SQL, предназначенная для составления отзывов для будущих версий. Использовать эту предварительную версию в рабочих средах не рекомендуется.

## **Репликация, зеркальное отображение и доставка журналов**

Репликация, зеркальное отображение или доставка журналов используются для перемещения данных из локального сервера SQL Server на другие локальные серверы SQL Server или на экземпляр SQL Server, работающий на виртуальной машине Windows Azure. Однако их нельзя использовать для перемещения данных в базу данных SQL Windows Azure или из нее. Дополнительные сведения см. в разделах [Репликация и доставка журналов](#) и [Зеркальное отображение базы данных и доставка журналов](#).

## Настраиваемые операции извлечения, преобразования и загрузки данных (ETL)

Для того чтобы свести к минимуму время, необходимое для передачи данных во время перехода, следует переместить как можно больше данных в Windows Azure до фактического перехода. Можно создать настраиваемое ETL-задание для перемещения измененных данных из локальной системы в среду Windows Azure. При миграции из локальной системы SQL Server 2008 или более поздней версии рекомендуется использовать средства отслеживания измененных данных и отслеживания изменений, чтобы все измененные данные, и только измененные данные перемещались из локальной базы данных в экземпляр базы данных SQL Windows Azure. Дополнительные сведения об этих двух функциях см. в разделе [Отслеживание измененных данных](#) в документации по SQL Server в сети. Для баз данных, не использующих отслеживание измененных данных и отслеживание изменений, необходимо создать систему отслеживания изменений и перенесенных данных. Во всех случаях минимальный объем данных для перемещения на момент перехода — это ключевой момент для минимизации времени простоя.

## Экспорт приложения уровня данных (DAC)

С помощью приложения уровня данных вы можете экспортировать данные из экземпляра SQL Server и помещать их в хранилище больших двоичных объектов Windows Azure, где их можно импортировать в базу данных SQL Windows Azure. С помощью приложений уровня данных можно настроить фильтры для импорта и экспорта таблиц. Настроить фильтры на уровне строк нельзя. Поэтому приложение уровня данных используется, когда таблицы целиком помещаются в одной базе данных, но этот метод не подходит для федеративных баз данных. Приложение уровня данных не является оптимальным выбором для переноса данных приложений, когда требуется постоянная синхронизация данных. Дополнительные сведения см. в разделе [Экспорт приложения уровня данных](#) в документации по SQL Server в сети.

## Резервное копирование и восстановление

Целью создания резервных копий баз данных является возможность восстановления при потере данных, вызванной административными ошибками, ошибками приложения, или полной потере центра обработки данных. Создание резервных копий данных и их восстановление из копии в базе данных SQL Windows Azure отличается от таких же процессов в локальном SQL Server и должно использовать имеющиеся ресурсы и средства. Таким образом, для надежной работы операций резервирования и восстановления необходима стратегия резервирования и восстановления базы данных SQL Windows Azure.

Существует три общих категории проблем, при которых может потребоваться восстановить экземпляр базы данных SQL Windows Azure.

**Сбой инфраструктуры или оборудования.** В центре обработки данных может произойти сбой оборудования. Например, может произойти сбой физического узла, на котором работает экземпляр базы данных SQL Windows Azure.

**Ошибки или сбои, вызванные приложением или пользователем.** Пользователи или приложения могут внести нежелательные изменения в данные. Для этого могут потребоваться обратные операции. Например, пользователь может изменить какие-то данные, относящиеся не к тому клиенту, и так далее.

**Потеря центров обработки данных.** В текущем соглашении об уровне обслуживания базы данных SQL Windows Azure указано исключение для факторов, находящихся вне разумного контроля корпорации Майкрософт, таких как стихийные бедствия. При стихийном бедствии центр обработки данных может быть поврежден таким образом, что базы данных нельзя будет восстановить из реплик или локальных резервных копий.

В конечном счете нужно решить, какой уровень риска является оптимальным в отношении данных, хранящихся в базах данных SQL Windows Azure в центрах обработки данных. Подробную информацию о доступных инструментах резервного копирования и восстановления и о том, как создать стратегии аварийного восстановления на их основе, см. в статье **Business Continuity in SQL Database** в библиотеке MSDN.

## Переход на Windows Azure

При выполнении фактического переноса приложения в Windows Azure используются следующие подходы:

**Параллельный запуск.** С этим подходом приложение выполняется параллельно как локально, так и в Windows Azure. Это позволяет выполнить практические тесты приложения в Windows Azure перед тем, как приложение станет полностью зависеть от облака. Тесты должны включать в том числе следующие проверки: функциональное тестирование, тестирование производительности и масштабируемости. После завершения тестирования новой системы на платформе Windows Azure выполните окончательный перенос данных и выключите локальную систему.

**Приостановка и переход.** Этот подход используется, когда все данные нужно синхронизировать перед полным запуском в Windows Azure. При использовании этого подхода сначала необходимо выполнить все функциональные тесты и проверки

производительности в Windows Azure. Затем настройте репликацию данных в среду Windows Azure с помощью одного из указанных выше способов синхронизации данных. Рекомендуется обеспечивать состояние данных, наиболее близкое к синхронизированному, сокращая время, необходимое для последней синхронизации или ETL-операции до окончательного перехода. Когда придет время для перехода на Windows Azure, отключите локальную систему, выполните последнюю синхронизацию данных и включите приложение в Windows Azure.

## См. также

[Миграция приложений, предназначенных для обработки данных, в Windows Azure](#)

# Миграция с помощью виртуальных машин Windows Azure

[Эта документация предназначена исключительно для предварительного ознакомления и в следующих версиях может быть изменена. Пустые разделы включены в документацию в качестве заполнителей.]

В этом разделе даются общие сведения о виртуальных машинах Windows Azure и сопутствующих технологиях и функциях. Также приводятся рекомендации по миграции существующих баз данных SQL Server на платформу Windows Azure, использующую виртуальные машины Windows Azure.

## В этом разделе

Раздел	Описание
<a href="#">Общие сведения о виртуальных машинах Windows Azure</a>	Представлены общие сведения о виртуальных машинах Windows Azure.
<a href="#">Миграция с помощью SQL Server в виртуальные машины Windows Azure</a>	Предоставлен путь миграции для существующих баз данных SQL Server в Windows Azure с использованием возможности виртуальных машин Windows Azure.



## См. также

[Миграция приложений, предназначенных для обработки данных, в Windows Azure](#)

# Общие сведения о виртуальных машинах Windows Azure

---

**Авторы:** Сельцин Туркаслан (Selcin Turkarslan)

**Рецензенты:** Дрю Мак-Дэниэл (Drew McDaniel), Джейсон Чен (Jason Chen), Ганеш Сринивасан (Ganesh Srinivasan), Линдси Аллен (Lindsey Allen), Стив Говард (Steve Howard)

[Эта документация предназначена исключительно для предварительного ознакомления и в следующих версиях может быть изменена. Пустые разделы включены в документацию в качестве заполнителей.]

Начиная с версии Windows Azure 2012 Preview, в Windows Azure были добавлены новые возможности виртуальных машин. В рамках этого улучшения Windows Azure была выпущена новая версия портала управления Windows Azure и расширены существующие предложения и возможности.

В этом разделе представлены общие сведения о виртуальных машинах Windows Azure. Более подробную информацию о пути миграции для существующих баз данных SQL Server с помощью виртуальных машин Windows Azure см. в разделе [Миграция с помощью SQL Server в виртуальные машины Windows Azure](#).

## Windows Azure и виртуальные машины

Начиная с предварительной версии, Windows Azure предоставляет следующие возможности.

**Развертывание виртуальной машины:** вы можете создать собственную виртуальную машину непосредственно в облаке, используя образ, представленный в коллекции образов портала управления Windows Azure, без загрузки образа Windows Server или Linux, созданного локально. Для этого можно использовать портал управления, PowerShell, программируемый интерфейс API (REST) или средства командной строки для настольных компьютеров Mac и Linux. После создания собственного диска можно управлять им, получая доступ к виртуальной машине. Для виртуальной машины под управлением операционной системой Windows Server можно использовать кнопку «Подключить» на портале управления для запуска соединения с удаленным рабочим столом. Для виртуальной машины под управлением операционной системой Linux для

входа в систему используется клиент Secure Shell (SSH). Подробную информацию о создании, развертывании виртуальных машин и управлении ими на настольных компьютерах Windows, Mac и Linux см. в разделе [Загрузки](#) на сайте WindowsAzure.com.

**Используйте собственную виртуальную машину:** новый портал управления Windows Azure и пакет средств разработки программного обеспечения Windows Azure позволяют вам переносить собственные виртуальные машины в облако. Например, на диске может быть уже установлен SQL Server. В этом случае передайте образ в учетную запись хранилища больших двоичных объектов и используйте этот образ для создания экземпляра новой виртуальной машины.

Кроме того, Windows Azure предоставляет новый набор возможностей для виртуализации сети и подключения сайтов на основе VPN как часть нового компонента виртуальной сети Windows Azure. Виртуальная сеть Windows Azure предоставляет следующие возможности.

Корпоративную сеть в платформе Windows Azure можно расширить с помощью виртуальных сетей. Кроме того, функция виртуальных сетей позволяет облачным приложениям подключаться друг к другу, разместив их в одной виртуальной сети.

Для виртуальных машин, размещенных в Windows Azure, можно использовать подмножество корпоративного адресного пространства IPv4.

Можно использовать стабильные IPv4-адреса для ваших виртуальных машин.

Windows Azure поддерживает шлюз размещенной виртуальной частной сети (VPN), который обеспечивает соединение между облаком и локальной системой с помощью безопасного подключения IPSec.

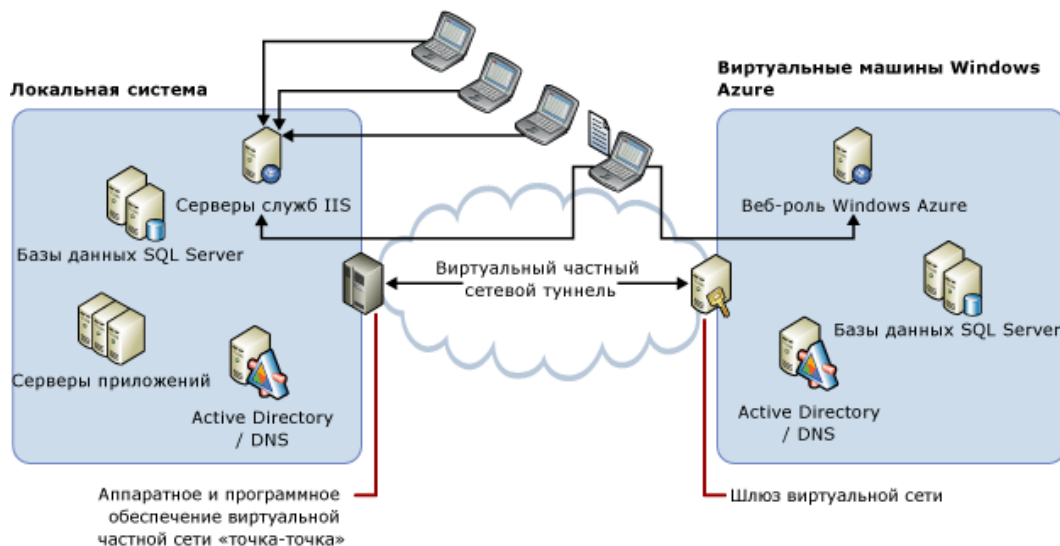
Можно подготовить виртуальные сети и управлять ими, используя портал управления Windows Azure. Виртуальные машины можно подготовить и управлять ими, используя портал управления Windows Azure.

Виртуальная сеть Windows Azure поддерживает стандартные устройства виртуальных частных сетей (VPN).

В облаке можно запустить локальную службу Active Directory или DNS-серверы. Виртуальные машины Windows Azure позволяют присоединять виртуальные машины в Windows Azure к корпоративным локальным доменам с помощью локальных служб Active Directory.

Модель виртуальных машин Windows Azure позволяет присоединять виртуальные машины в Windows Azure к корпоративным локальным доменам с помощью локальных служб Active Directory.

На следующей диаграмме показано, что виртуальные машины Windows Azure позволяют клиентам легко расширить их корпоративные сети для работы в Windows Azure. Это дает большое преимущество для переноса существующих приложений в Windows Azure. Можно легко поддерживать гибридные приложения, которые охватывают облачные и локальные системы. Можно управлять виртуальными сетями в Windows Azure и использовать размещенный VPN-шлюз для установки подключения между локальной и облачной средой. Можно присоединить виртуальные машины в Windows Azure к локальным корпоративным доменам.



Подробную информацию об управлении виртуальными машинами в Windows Azure см. в [центре управления](#) и в разделе [Загрузки](#) на сайте WindowsAzure.com. Этот сайт содержит ссылки на практические руководства, командлеты PowerShell и средства командной строки, которые можно использовать для платформ Windows, Mac и Linux.

## Образы виртуальных машин, поставляемые с Windows Azure

Начиная с предварительной версии, Windows Azure позволяет клиентам развертывать виртуальные машины с помощью поставляемых образов Windows Azure. При выборе панели виртуальных машин на портале можно увидеть несколько образов платформы, доступных по умолчанию.

После выбора требуемого образа платформы можно создать и затем подключиться к виртуальной машине на Windows Azure. После этого создания экземпляра образа

платформы пользователь несет ответственность за его обслуживание. Периодически Windows Azure обновляет образы, предоставленные с Windows Azure. Но Windows Azure не выполняет принудительных обновлений дисков операционной системы, уже развернутых клиентами. Аналогично партнеры Linux будут периодически обновлять базовые образы Linux. Подробную информацию об управлении виртуальными машинами Windows и Linux и входе на них см. в [центре управления](#) на сайте WindowsAzure.com.

Ниже приведен список приложений, которые поддерживаются в виртуальных машинах, работающих на платформе Windows Azure.

Приложения	Сведения
Microsoft SQL Server	<p>Поддерживаемые версии при использовании собственной виртуальной машины: SQL Server 2008, SQL Server 2008 R2 и все версии SQL Server 2012.</p> <p>Поддерживаемые версии в коллекции образов: SQL Server 2012 Evaluation.</p> <p>Поддерживаемые приложения: СУБД базы данных SQL Server, службы аналитики SQL Server, службы SQL Server Reporting Services, службы SQL Server Integration Services, средства управления SQL Server, пакет SQL Connectivity SDK, средства настройки, обновления и миграции SQL Server, такие как приложения уровня данных (DAC), резервного копирования, восстановления, присоединения и отсоединения. Обратите внимание, что службы Master Data Services в настоящее время не поддерживаются.</p>
Windows Server Active Directory	Поддерживаемые версии: Windows Server 2008 R2

Приложения	Сведения
Microsoft SharePoint	Поддерживаемые версии: все версии SharePoint 2010
Поддержка Linux	Можно передать виртуальный жесткий диск (VHD-файл) Linux для запуска в Windows Azure. Самую последнюю информацию о поддерживаемых версиях см. на портале управления Windows Azure.

Windows Azure использует средство подготовки системы (Sysprep) для захвата работающих виртуальных машин Windows Server для повторного развертывания. Пользователь может хранить все развернутые диски операционных систем и диски данных в собственной учетной записи хранилища больших двоичных объектов Windows Azure как стандартные VHD-файлы.

## Список важных понятий

**Образ** операционной системы представляет собой файл виртуального жесткого диска, который можно использовать в качестве шаблона для создания новой виртуальной машины. Образ — это шаблон, так как у него нет конкретных параметров, как у настроенной виртуальной машины, таких как параметры учетной записи пользователя и имя компьютера.

**Диск** виртуальной машины — это виртуальный жесткий диск, который может быть загружен и подключен как работающая версия экземпляра операционной системы.

**Диск** также может быть присоединен к запущенному экземпляру как диск данных отдельно от диска операционной системы.

Работающую виртуальную машину можно захватить как образ. Однако эта операция не позволяет записать подключенные диски. Полученную VM можно использовать для создания нескольких VM. Конечным результатом является новый файл образа в той же учетной записи хранения, в которой расположен диск ОС полученной виртуальной машины.

У приложения Windows Azure может быть несколько виртуальных машин. Все виртуальные машины в облачной службе могут разрешить IP-адрес других виртуальных машин, предоставленных Windows Azure, с помощью DNS-имени. Доступом к сети можно управлять с помощью параметров межсетевого экрана.

## Высокий уровень доступности и аварийное восстановление при использовании виртуальных машин Windows Azure

Для обеспечения аварийного восстановления данных и дисков Windows Azure использует недавно представленную возможность географической репликации хранилища Windows Azure. Все изменения, внесенные приложением или заказчиком на дисках операционной системы или дисках данных клиента, даже в случае сбоя оборудования сохраняются с помощью хранилища больших двоичных объектов Windows Azure. Как описано в записи блога [Введение в географическую репликацию данных в хранилище Windows Azure](#), большие двоичные объекты и таблицы Windows Azure географически реплицируются между двумя центрами обработки данных, расположенными на определенном расстоянии друг от друга на одном континенте, обеспечивая дополнительную долговечность данных в случае крупной катастрофы без дополнительных затрат. При запуске виртуальной машины служба географической репликации хранилища Windows Azure реплицирует диски операционной системы и данных во второй географический регион по умолчанию.

В предварительной версии Windows Azure для приложений SQL Server поддерживаются такие возможности обеспечения высокого уровня доступности, как зеркальное отображение базы данных и доставка журналов. В настоящее время возможности **Windows Server Failover Clustering (WSFC) with SQL Server** и **SQL Server 2012 AlwaysOn** не поддерживаются. При сбое оборудования, на котором размещается ваша виртуальная машина, Windows Azure восстанавливает ту же виртуальную машину на другом компьютере. Для предотвращения потерь данных и конфигурации из-за ошибки пользователей и приложений рекомендуется периодически создавать резервные копии данных на виртуальных машинах.

## Общие шаблоны приложений при использовании виртуальных машин Windows Azure

Следующие шаблоны приложений представляют собой примеры, которые могут использовать преимущества новой модели виртуальных машин Windows Azure:

Существующие некритические приложения баз данных

Новые приложения баз данных, которые необходимо развернуть в SQL Server на виртуальной машине Windows Azure, если база данных SQL Windows Azure не поддерживает все необходимые функции

Быстро и легко создавайте и тестируйте среду для новых приложений, которые будут развернуты локально

Решение для резервного копирования для локальных приложений баз данных

Решение, которое можно быстро и легко масштабировать в часы пик

Решение, которое может преодолеть недостатки локальной платформы виртуализации

## Поддержка интерфейсов API для виртуальных машин Windows Azure

Windows Azure позволяет управлять приложениями и виртуальными машинами с помощью API-интерфейса REST и командлетов PowerShell. Дополнительные сведения см. в статье [Справочник по интерфейсам API для Windows Azure](#) в библиотеке MSDN.

### См. также

[Миграция приложений, предназначенных для обработки данных, в Windows Azure](#)

[Миграция с помощью виртуальных машин Windows Azure](#)

[Виртуальные машины](#)

## Миграция с помощью SQL Server в виртуальные машины Windows Azure

---

**Авторы:** Сельцин Туркаслан (Selcin Turkarslan)

**Рецензенты:** Евгений Кривошеев (Evgeny Krivosheev), Авилай Парех (Avilay Parekh), Линдси Аллен (Lindsey Allen), Стив Ховард (Steve Howard)

[Эта документация предназначена исключительно для предварительного ознакомления и в следующих версиях может быть изменена. Пустые разделы включены в документацию в качестве заполнителей.]

Начиная с предварительной версии Windows Azure 2012, вы можете легко переносить существующие приложения SQL Server, разработанные для платформы Windows Server, на

виртуальные машины Windows Azure. В данном разделе даются общие сведения об этой новой возможности и процесс принятия решения о том, когда выбирать SQL Server на виртуальной машине в базе данных SQL Windows Azure (база данных SQL).

С помощью SQL Server на ВМ можно сократить общие расходы стоимости развертывания, а также затраты на управление и поддержку приложений уровня организации, перенеся эти приложения в общедоступное облако. При переносе существующих приложений SQL Server на виртуальные машины Windows Azure требуются минимальные изменения в коде. SQL Server в ВМ позволяет администраторам и разработчикам использовать те же средства разработки и администрирования, что и при работе с локальными серверами.

## Когда следует использовать SQL Server на виртуальных машинах Windows Azure

Использование SQL Server в Windows Azure Virtual Machine позволяет реализовать множество локальных сценариев и в облаке.

- **Быстрая разработка и тестирование приложения.** Разрабатывается приложение базы данных, для которого требуется некоторая проверка в среде, схожей с рабочей. Вместо приобретения нового оборудования для выполнения проверки нового приложения используйте SQL Server на ВМ. Создайте ВМ с помощью поставляемого с платформой SQL Server на образе Windows Server или передайте в облако собственный образ. Затем подключитесь к новой машине, переместите нужные данные, настройте приложение, выполните тесты и внесите исправления.
- **Общая стоимость владения на платформе виртуализации.** Перемещение существующей платформы виртуализации в общедоступное облако. Вместо приобретения нового оборудования для поддержки увеличивающегося числа приложений уровня организации на своей собственной локальной платформе виртуализации переместите их в общедоступное облако, воспользовавшись функциями SQL Server infrastructure в Windows Azure Virtual Machine.
- **Резервное копирование данных приложения.** Создание резервных копий базы данных приложения организации с использованием инфраструктуры SQL Server infrastructure в Windows Azure Virtual Machine. Вместо выделения локальных ресурсов для вычисления и хранения для резервных копий приложений баз данных на виртуальных машинах используйте SQL Server. Определите, какие приложения базы данных подходят для подобной задачи в облаке. После перемещения приложений и данных в облако используйте Windows Azure Virtual Network для установления соединения между локальными серверами и облаком и настройте задания и предупреждения по расписанию для создания резервных копий на диске виртуальной машины.



- **Быстрое масштабирование по надобности.** Выделение дополнительных ресурсов вычисления, хранения и сети для обслуживания сезонных всплесков нагрузки на приложение. Вместо приобретения дополнительного оборудования, нужного лишь на определенный период времени, используйте инфраструктуру SQL Server в Windows Azure Virtual Machines.
- **Доступность и мобильность данных.** Размещение баз данных SQL Server databases в Windows Azure Virtual Machines позволяет использовать их и в локальных и в облачных приложениях.

## Выбор платформы: SQL Server в виртуальной машине Windows Azure или база данных SQL Windows Azure

Следующий рабочий процесс решения позволяет понять, когда следует использовать SQL Server в ВМ или базу данных SQL Windows Azure.

Для новых приложений баз данных можно использовать либо базу данных SQL Windows Azure (SQL Database), либо SQL Server на ВМ в Windows Azure.

- Если база данных SQL поддерживает все требуемые функции, выделите новую базу данных SQL в Windows Azure. Разработайте новое приложение баз данных с помощью Windows Azure SDK, модулей для Visual Studio 2010, Java, PHP или Node.js. Разверните приложения в Windows Azure и создайте таблицы в базе данных SQL.
- Если база данных SQL не поддерживает все требуемые функции и вы не желаете вкладывать средства в перепроектирование базы данных приложения, выделите новую виртуальную машину Windows Azure Virtual Machine с образом платформы SQL Server на новом портале управления. Создайте пакет развертывания базы данных с помощью SQL Server Data Tools, разверните пакет на SQL Server в Windows Azure Virtual Machine. С помощью традиционных средств администрирования, например среды SQL Server Management Studio, можно обновлять и отслеживать подобную базу данных и управлять ею.

Сначала определите, какие из существующих баз данных приложения вы хотите перенести в SQL Server на виртуальные машины Windows Azure. Затем выберите один из следующих вариантов.

Преобразуйте физические или виртуальные машины в виртуальные жесткие диски Hyper-V с помощью мастеров System Center 2012 Virtual Machine. Передайте виртуальные машины в хранилище Windows Azure Storage с помощью инструмента командной строки [Csupload](#). Новую виртуальную машину также можно развернуть с помощью загруженной ранее. С помощью традиционных средств

администрирования, например среды SQL Server Management Studio, можно обновлять и отслеживать подобную базу данных и управлять ею.

Выделите новую виртуальную машину Windows Azure с поставляемым с платформой SQL Server образом и требуемые облачные ресурсы на новом портале управления Windows Azure. Создайте пакет развертывания базы данных с помощью SQL Server Data Tools и среды SQL Server Management Studio. Перенесите существующий проект приложения в Windows Azure Project с помощью Windows Azure SDK, модулей для Visual Studio 2010, Java, PHP или Node.js. Разверните проект приложение в Windows Azure для получения доступа к данным в облаке.

## **Когда следует использовать на виртуальных машинах Windows Azure перенесенные в SQL Server схемы базы данных и данные**

При переносе базы данных в SQL Server на виртуальных машинах Windows Azure следуйте следующим шагам в указанном порядке.

1. Подготовьте на локальных машинах схему базы данных и файлы данных с помощью средств DAC, резервного копирования или отсоединения. Дополнительные сведения см. ниже в подразделе [Как подготовить локальные схему и данные и передать их в экземпляр SQL Server на виртуальной машине..](#)
2. Дополнительно вы можете сжать и зашифровать файлы перед их передачей в Windows Azure.
3. Переместите схему базы данных и файлы журналов в Windows Azure. Если используется средство CsUpload, сначала разместите файлы в виртуальном жестком диске (VHD) и затем передайте его в Windows Azure. Дополнительные сведения см. в подразделе ниже [Перемещение схемы базы данных и файла данных на виртуальную машину Windows Azure.](#)
4. Загрузите схему базы данных и файл данных на виртуальную машину Windows Azure. Дополнительные сведения см. ниже в подразделе [Как подготовить локальные схему и данные и передать их в экземпляр SQL Server на виртуальной машине..](#)
5. Воссоздайте на виртуальной машине Windows Azure все метаданные, которые не могли быть созданы средствами миграции на SQL Server.

# Как подготовить локальные схему и данные и передать их в экземпляр SQL Server на виртуальной машине.

В данном подразделе описывается, как подготовить на локальных машинах схему базы данных и файлы данных. В зависимости от конкретных условий может быть применен один из следующих вариантов выполнения данного шага.

[Вариант 1. Использование файлов BACPAC или DACPAC приложения уровня данных](#)

[Вариант 2. Резервное копирование и восстановление](#)

[Вариант 3. Использование команд Attach и Detach](#)

[Вариант 4. Использование других функций SQL Server](#)

## Вариант 1. Использование файлов BACPAC или DACPAC приложения уровня данных

Для подготовки к передаче схемы базы данных и файлов данных с локального компьютера в облако можно использовать приложение уровня данных.

- Файл DACPAC: файл, включающий все определения объектов SQL Server, таких как таблицы, представления и объекты экземпляров, в том числе имена входа, связанные с базой данных пользователя. С помощью пакета DACPAC осуществляется захват и развертывание схемы базы данных, включая обновление существующей базы данных. Для получения дополнительных сведений о том, как извлечь пакет приложения уровня данных из существующей базы данных SQL Server, см. раздел [Мастер извлечения пакета приложения уровня данных из базы данных](#).
- Файл BACPAC содержит схему базы данных, а также данные, хранимые в ней. Пакет BACPAC в основном предназначен для хранения схемы и данных. Это логический эквивалент резервной копии базы данных, он не может использоваться для обновления существующих баз данных. Дополнительные сведения о создании файла BACPAC см. в разделе [Экспорт приложения уровня данных](#).

Схему и хранимые в базе данные можно экспортировать в файл BACPAC. Затем схему и данные можно импортировать в новую базу данных на основном сервере. Обе эти возможности поддерживаются средствами управления базами данных: среда SQL Server Management Studio и DACFx API. Для получения доп. сведений см. разделы [Импорт файла BACPAC File для создания новой пользовательской базы данных](#) и [Пространство имен Microsoft.SqlServer.Dac](#) в библиотеке MSDN.

**Примечания.**

При выполнении DAC-операций файлы BACPAC и DACPAC не шифруются автоматически.

Следует убедиться в том, что канал связи между локальным сервером и облаком достаточно безопасен. Также файлы можно зашифровать отдельно, чтобы получить дополнительный уровень защиты, пока он хранится в хранилище больших двоичных объектов Windows Azure или на локальном диске.

DAC не поддерживает полнотекстовые каталоги.

Для повышения безопасности имена входа, использующие аутентификацию SQL Server, хранятся в пакете приложения уровня данных без пароля. При развертывании или обновлении пакета имя входа создается как отключенное имя входа с созданным паролем. Чтобы включить имена входа, войдите в систему под учетной записью, имеющей разрешение ALTER ANY LOGIN и с помощью команды ALTER LOGIN включите имя входа и присвойте ему новый пароль, который можно передать пользователю. Это не требуется для имен входа, использующих аутентификацию Windows, поскольку SQL Server не управляет их паролями.

## **Вариант 2. Резервное копирование и восстановление**

Для переноса базы данных на другой экземпляр SQL Server или на другой сервер применяются операции резервного копирования и восстановления. Если SQL Server на локальной машине и на виртуальной машине имеют одинаковую версию, то на виртуальную машину можно скопировать файл резервной копии базы данных и затем восстановить базу. Дополнительные сведения о создании резервной копии и восстановлении баз данных см. в разделе [Резервное копирование и восстановление баз данных SQL Server](#).

### **Примечания.**

Резервное копирование и восстановление выполняются быстрее, чем операции с приложением уровня данных.

Можно создать сжатую копию данных. Дополнительные сведения о сжатии см. в разделе [Сжатие резервной копии](#).

Можно использовать существующие и знакомые средства, например мастер резервного копирования среды SQL Server Management Studio и другие сторонние инструменты.

При перемещении базы данных для приложения на другой экземпляр сервера необходимо повторно создать все метаданные подчиненных сущностей и объектов в

базах данных master и msdb на экземпляре целевого сервера. Дополнительные сведения см. в разделе [Управление метаданными при обеспечении доступности базы данных на другом экземпляре сервера](#).

### **Вариант 3. Использование команд Attach и Detach**

Для переноса базы данных на другой экземпляр SQL Server или на другой сервер применяются операции присоединения и отсоединения. Скопируйте файлы данных (MDF, NDF) и журналов (LDF) в локальную папку виртуальной машины и затем подсоедините базу данных. Дополнительные сведения см. в разделе [Перенос базы данных с помощью команд подсоединения и отсоединения](#).

#### **Примечания.**

Отсоединение базы данных означает удаление ее с экземпляра SQL Server, но сама база данных остается неповрежденной со всеми своими файлами данных и журналов транзакций. Для этого необходим перевод базы данных-источника в режим «вне сети». Это наилучшим образом подходит для обновления баз данных или переноса особо больших баз данных.

Базу данных невозможно отсоединить в следующих случаях.

База данных реплицируется и публикуется.

Имеется моментальный снимок базы данных.

База данных находится в сеансе зеркального отображения базы данных.

База данных помечена как подозрительная.

База данных является системной базой данных.

Рекомендуется выполнить полное резервное копирование и перезапустить создание разностных резервных копий до отсоединения.

Отсоединенные файлы можно сжать с помощью отдельного средства.

При присоединении базы данных к другому экземпляру сервера для обеспечения ее согласованного функционирования для пользователей и приложений может потребоваться повторное создание некоторых или всех метаданных базы данных, например имен входа и заданий на другом экземпляре сервера. Дополнительные

сведения см. в разделе [Управление метаданными при обеспечении доступности базы данных на другом экземпляре сервера](#).

## Вариант 4. Использование других функций SQL Server

Для копирования или перемещения баз данных между серверами воспользуйтесь одним из следующих способов.

Мастер копирования баз данных в среде SQL Server Management Studio можно использовать для копирования или перемещения баз данных между серверами, а также для обновления базы данных до более новой версии. Дополнительные сведения см. в разделе [Использование мастера копирования баз данных](#).

Мастер импорта и экспорта SQL Server предоставляет самый простой способ копирования данных между источниками и создания основных пакетов. Дополнительные сведения о мастере см. в разделе [Мастер импорта и экспорта SQL Server](#). Назначение мастера импорта и экспорта SQL Server заключается в копировании данных из исходного расположения в целевое. Этот мастер может также создать целевую базу данных и целевые таблицы. Однако, если нужно скопировать несколько баз данных, таблиц или других объектов базы данных, следует использовать мастер копирования баз данных.

С помощью мастера формирования и публикации скриптов можно создавать скрипты для переноса баз данных между экземплярами компонента SQL Server Database Engine. Сформированные скрипты могут выполняться на другом экземпляре компонента Database Engine. С помощью мастера также можно публиковать содержимое базы данных непосредственно в веб-службе, созданной с помощью проекта служб Database Publishing Services. Создать скрипты можно как для всей базы данных, так и ограничить их определенными данными или объектами. Дополнительные сведения см. в разделе [Использование мастера формирования и публикации скриптов](#).

Можно использовать класс `Transfer` из библиотеки управляющих объектов SQL Server (SMO). Дополнительные сведения см. в разделе [Передача данных](#). При использовании библиотеки SMO исходная и целевая базы данных могут оставаться в сети, также не требуется перемещать файлы из хранилища больших двоичных объектов Windows Azure в качестве отдельного шага. Недостаток при использовании библиотеки SMO — клиентское подключение к базе данных и протокол потока табличных данных, которые могут быть неэффективны для больших наборов данных.

Задача [«Передача базы данных» копирует или перемещает базу данных](#) может переместить или скопировать базу данных между двумя экземплярами SQL Server. База данных может

быть передана в режимах «в сети» или «вне сети». В режиме «в сети» база данных остается присоединенной и передается с помощью SQL Management Objects (SMO) для копирования объектов. В режиме «вне сети» база данных отсоединяется, файлы базы данных перемещаются или копируются, база данных прикрепляется на целевом объекте после того, как передача успешно завершена.

## Перемещение схемы базы данных и файла данных на виртуальную машину Windows Azure

Маленькие файлы (копии базы данных, BACPAC или DACPAC) можно скопировать на виртуальную машину, используя операции копирования и вставки с помощью удаленного рабочего стола. Для передачи больших файлов выберите один из следующих способов.

Используйте средство командной строки CSUpload (CSUpload.exe) для передачи VHD-файлов в Windows Azure. VHD-файл может включать базу данных. Дополнительные сведения см. в разделе [Как передать файл VHD в Windows Azure](#).

Передайте файл в хранилище больших двоичных объектов в том же центре данных, где размещена виртуальная машина, затем выполните удаленное подключение к рабочему столу на виртуальной машине и загрузите файл из хранилища.

Дополнительные сведения см. в разделе [Основные сведения об облачном хранилище](#).

Скопируйте напрямую схему и файлы данных в общую папку виртуальной машины.

Передача файла по FTP. Дополнительные сведения о FTP см. в разделе [Служба публикации FTP](#).

Используйте веб-браузер для загрузки базы данных из Интернета. Например, можно загрузить базу данных AdventureWorks со страницы [codeplex](#)

В следующей таблице описаны некоторые распространенные методы передачи файлов на виртуальную машину Windows Azure. Там же приведены плюсы и минусы каждого метода.

Метод передачи	Преимущества	Недостатки
Загрузка копии в виде VHD-файла в хранилище больших двоичных объектов Windows Azure с помощью инструмента	Быстро, оптимизировано для Windows Azure  Средства могут работать с ненадежными	Программу командной строки CSUpload предоставляет корпорация Microsoft. Если требуется графический

Метод передачи	Преимущества	Недостатки
Csupload	соединениями. Безопасная передача	пользовательский интерфейс, воспользуйтесь сторонними средствами.  Прежде чем воспользоваться программой CSUpload для загрузки VHD в Windows Azure, следует подготовить сам файл, а также создать и передать на портал сертификат управления.  Подключите загруженный файл VHD к виртуальной машине Windows Azure.
Копирование файлов в общий ресурс виртуальной машины	<ul style="list-style-type: none"> <li>• Простой в использовании</li> <li>• Доступно множество клиентских средств</li> <li>• Файл размещается напрямую на виртуальной машине</li> </ul>	<ul style="list-style-type: none"> <li>• Требуется VPN-соединение.</li> <li>• Только несколько средств копирования файлов имеют функцию восстановления и перезапуска копирования.</li> </ul>
Передача файлов по FTP	<ul style="list-style-type: none"> <li>• Простой в использовании</li> <li>• Доступно множество клиентских средств</li> <li>• Файл размещается напрямую на виртуальной машине</li> </ul>	Тяжело добиться безопасности при передаче данных.

Дополнительные сведения о настройке, конфигурировании и развертывании виртуальной машины SQL Server в Windows Azure см. в разделе [Учебник. Управление виртуальной машиной SQL Server в Windows Azure](#). В этом учебнике описывается, как использовать портал



управления Windows Azure для выбора и установки стандартной виртуальной машины. Также объясняется удаленное подключение к виртуальной машине с использованием удаленного рабочего стола и подключение к SQL Server на виртуальной машине с помощью среды SQL Server Management Studio.

Дополнительные сведения см. в разделах [Приступая к работе с SQL Server на виртуальной машине Windows Azure](#) и [Создание и отправка виртуального жесткого диска с операционной системой Windows Server](#).

## См. также

[Общие сведения о виртуальных машинах Windows Azure](#)

# Миграция с помощью облачных служб Windows Azure

---

В этом разделе описываются облачные службы Windows Azure и сопутствующие технологии и функции. Кроме того, в данном разделе представлены рекомендации по использованию этих технологий и функций при переносе приложений и баз данных в среду облачных служб Windows Azure. В этом разделе приводятся детальные графики сравнения использования различных вариантов хранилища и средств миграции.

## В этом разделе

Раздел	Описание
<a href="#">Рекомендации по разработке для облачных служб Windows Azure</a>	Вводные сведения об облачных службах Windows Azure, а также основных поддерживаемых технологиях, которые необходимо использовать при разработке или переносе приложений в службы Windows Azure.
<a href="#">Общие сведения о службах управления данными в Windows Azure</a>	Описание предложений по управлению данными, предоставленными Windows Azure, такими как таблицы Windows Azure, большие двоичные объекты, база данных

Раздел	Описание
	SQL Windows Azure и диски Windows Azure.
<a href="#">Миграция баз данных SQL Server в базу данных SQL Windows Azure</a>	Инструкции по переносу базы данных с сервера SQL Server в базу данных SQL Windows Azure, в том числе для миграции определений объектов данных в схемах и данных в таблицах.
<a href="#">Перенос данных в другие службы управления данными в Windows Azure</a>	Руководство по миграции приложений для использования предложения по управлению данными Windows Azure: таблицами, большими двоичными объектами, диском Windows Azure (при поддержке больших двоичных объектов страницы).
<a href="#">Вопросы миграции в службу кэша Windows Azure</a>	Руководство по использованию службы кэширования Windows Azure во время переноса данных в Windows Azure.
<a href="#">Миграция приложений, которые используют технологии обмена сообщениями</a>	Руководство по миграции приложений, которые используют технологии обмена сообщениями, особенно службу очередей сообщений (Майкрософт).
<a href="#">Миграция приложений для использования локального хранилища</a>	Руководство по использованию локального хранилища на виртуальной машине, на которой выполняется экземпляр вашего приложения.

## См. также

[Миграция приложений, предназначенных для обработки данных, в Windows Azure](#)

# Рекомендации по разработке для облачных служб Windows Azure

---

**Авторы:** Сельцин Туркаслан (Selcin Turkarslan)

**Рецензенты:** Валерий Мизонов (Valery Mizonov), Авилай Парех (Avilay Parekh), Паоло Сальватори (Paolo Salvatori), Стив Ховард (Steve Howard)

[Эта документация предназначена исключительно для предварительного ознакомления и в следующих версиях может быть изменена. Пустые разделы включены в документацию в качестве заполнителей.]

При рассмотрении миграции приложений в облачные службы Windows Azure рекомендуется вначале изучить облачные службы Windows Azure. В статье [Знакомство с Windows Azure](#) даются сведения о компонентах Windows Azure, управлении данными и поддерживаемых пакетах средств разработки программного обеспечения (пакетах SDK).

В этом разделе предоставлены общие сведения о реализации приложения Windows Azure. Из-за наличия бесконечного количества разных сценариев миграции разработчикам рекомендуется выбрать наиболее подходящую технику и решения для своих приложений и пользователей.

Миграция существующего приложения в Windows Azure включает следующие действия.

Добавление конкретной конфигурации Windows Azure и некоторого нового пользовательского кода.

Повторная упаковка существующего приложения как приложения Windows Azure.

Развертывание приложения как облачной службы, выполняющейся в виртуальной машине Windows Azure.

Облачная служба Windows Azure включает код приложения и параметры конфигурации Windows Azure. При разработке приложения для облака применимы такие общие архитектурные шаблоны, что разработчики должны сконструировать в распределенной среде архитектуру приложений для обработки доступности, масштабируемости, надежности и безопасности. Кроме того, разработчики должны рассмотреть соглашения об уровне обслуживания, планирование ресурсных затрат, выставление счетов клиентам, аудит, наблюдение за приложением, анализ трафика и управление затратами, а также определение моментов для вертикального и горизонтального масштабирования.

# Создание облачной службы в Windows Azure

В традиционной среде частных центров обработки данных для выполнения услуг необходимо приобрести, установить и поддерживать оборудование. С помощью Windows Azure вы можете проектировать и строить приложения, которые могут масштабироваться в зависимости от потребности, выделяя виртуализованные ресурсы. Хотя некоторые локальные приложения могут выполняться в Windows Azure при минимальных изменениях или вообще без изменений, в большинстве приложений будут получены преимущества от проектирования и изменения архитектуры для работы в облачной среде. Для получения всех преимуществ от Windows Azure рекомендуется перед миграцией в Windows Azure изменить приложение с использованием множества ролей.

Например, веб-службы, размещенные в прежних центрах обработки данных, часто объединяют несколько функций в одно приложение, которое не может хорошо масштабироваться. Оно также хранит состояние приложения на локальном диске, что не работает в среде облачных служб Windows Azure. При выполнении миграции из существующего веб-приложения в Windows Azure рекомендуется преобразовать код устаревшего приложения в веб-роли и рабочие роли Windows Azure.

В облачных службах Windows Azure каждое приложение реализуется как одна или несколько ролей. Каждая роль содержит код и данные конфигурации, необходимые для выполнения некоторой части функции приложения. Веб-роль предназначена для использования службами веб-интерфейса, а код, который непосредственно взаимодействует с веб-браузерами или другими клиентами HTTP, выполняется в службах IIS и веб-сервере Microsoft. Рабочая роль обычно используется для выполнения фоновой обработки и задач поддержки. Она наиболее подходит для служб среднего уровня. Каждая роль может иметь несколько экземпляров. В каждом экземпляре выполняется один и тот же код, который был написан для роли, но каждый экземпляр роли размещается в отдельной виртуальной машине в центре обработки данных Windows Azure. Для каждой роли можно указать желательный размер виртуальной машины, которая должна использовать эту роль. Дополнительные сведения см. в разделе [Настройка размера виртуальных машин](#) в библиотеке MSDN. В дополнение к функциональному различию ролей каждая роль служит для приложения в качестве единицы масштабирования. Другими словами, можно иметь 20 экземпляров веб-роли для обслуживания дополнительного трафика и только 5 экземпляров рабочей роли для асинхронной обработки запросов веб-роли. Если нужно создать только простое приложение ASP.NET, PHP, Node.js или, например, службу WCF, можно использовать только веб-роль. Рекомендуется выполнить детальные функциональные тесты и тесты производительности по облачным службам, чтобы определить оптимальное количество экземпляров ролей и размер виртуальной машины перед отправкой приложения в

производственную среду. Дополнительные сведения об основных понятиях облачных служб см. в разделе [Центр разработчиков Windows Azure](#).

Рекомендуется рассмотреть использование преимуществ доступных вариантов хранения, которые предоставляет Windows Azure. Это может упростить логику приложения и увеличить производительность облачной службы. Именно поэтому важно понимать ограничения каждого варианта хранения данных при планировании миграции приложения и четко понимать, какой вариант хранения данных нужно использовать для облачной службы. Дополнительные сведения о доступных вариантах хранения в Windows Azure см. в разделе [Общие сведения о службах управления данными в Windows Azure](#) в руководстве по Windows Azure. Кроме того, для предоставления размещаемого в памяти высокоскоростного хранилища для приложений Windows Azure может потребоваться [служба кэширования Windows Azure](#). Кэширование повышает производительность благодаря временному хранению данных из других источников на сервере и может снизить расходы, связанные с транзакциями доступа к месту хранения базы данных в облаке. Если планируется миграция приложения, имеющего сильные зависимости в базовой файловой системе, ознакомьтесь с подробной рекомендацией в разделе [Перенос данных на диски](#). Использование облачного диска Windows Azure дает возможность выполнять миграцию приложения, которое должно сохранять свое состояние в традиционной файловой системе NTFS. Для получения лучших результатов производительности выполните развертывание приложения в центрах обработки данных, наиболее близко расположенных к большинству клиентов, и в центрах обработки данных, в которых размещена учетная запись хранилища или экземпляры базы данных SQL Windows Azure. Но если имеются проблемы юрисдикции или правовые проблемы, связанные с данными и с тем, где они находятся, можно выбрать центр обработки данных, близкий к компании или к данным. Дополнительные сведения см. в разделе [Вопросы производительности базы данных SQL Windows Azure](#).

## Разработка приложений в Windows Azure

Перед началом реализации облачной службы Windows Azure необходимо вначале получить подписку на Windows Azure. Дополнительные сведения см. на сайте [WindowsAzure.com](#). Теперь нужно подготовить среду разработки. Корпорация Майкрософт в настоящее время предоставляет зависящие от языка программирования пакеты SDK для .NET, Java, PHP, Node.js и Python. Наиболее актуальные сведения о поддерживаемых платформах и языках программирования см. на сайтах [WindowsAzure.com](#). Кроме того, для доступа ко всем клиентским библиотекам, пакетам SDK и средствам командной строки можно использовать [центр загрузки Windows Azure](#).

Чтобы клиентские приложения могли работать на разных платформах для связи с облаком, в качестве стандартного API-интерфейса для приложений собственных клиентов, соединяющихся с базой данных SQL Windows Azure (базой данных SQL), корпорация Майкрософт выбрала ODBC. Поставщик OLE DB для собственного клиента SQL Server будет в последний раз поставляться в SQL Server 2012 и не поддерживается в базе данных SQL. При написании приложений для Windows или Windows Azure необходимо использовать драйвер поставщика ODBC для собственного клиента SQL Server, который поставляется вместе с SQL Server 2008 R2 или более поздней версии. При разработке новых или будущих версий приложения рекомендуется адаптировать ODBC. Для существующих приложений, которые используют OLE DB, рекомендуется рассмотреть миграцию приложений в ODBC как часть будущей стратегии. Дополнительные сведения о преобразовании приложения OLE DB в приложение ODBC см. в техническом документе [Руководство по преобразованию OLE DB в ODBC](#). Начиная с предварительной версии Windows Azure, Windows Azure предоставляет поставщикам служб новую службу: **Windows Azure Web Sites**. Эта новая служба дает разработчикам возможность быстро создавать и развертывать веб-сайт в Windows Azure. С помощью этой новой службы можно разрабатывать и публиковать веб-сайты непосредственно на портале Windows Azure. Дополнительные сведения см. в разделе [Веб-сайты Windows Azure](#) на сайте WindowsAzure.com.

Как и при разработке других приложений, необходимо сконструировать архитектуру приложения для обработки доступности, аварийного восстановления и безопасности в мультитенантной, распределенной облачной среде. Дополнительные сведения см. в разделах [Вопросы высокого уровня доступности и аварийного восстановления с помощью базы данных SQL Windows Azure](#) в документации по Windows Azure и [Ресурсы безопасности для Windows Azure](#). Требования безопасности рекомендуется обеспечить в начале процесса разработки, перед размещением приложения в Windows Azure. Дополнительные сведения о разработке приложений в Windows Azure см. в разделе [Developing Windows Azure Applications](#).

## Регистрация, тестирование, диагностика и отладка приложений Windows Azure

Поскольку Windows Azure является мультитенантной, динамически масштабируемой платформой в облачной среде, при конструировании приложения нужно рассмотреть специфичную для облака технику наблюдения и диагностики. Наблюдение и диагностику рекомендуется рассмотреть в начале жизненного цикла разработки приложения. Рекомендуется оценить, будет ли предоставлять существующая техника наблюдения и диагностики правильные результаты после развертывания приложения в облаке.

Например, для приложения, которое формирует большие файлы журнала, может понадобиться уточнение параметров диагностики и ведения журнала. В результате приложение станет создавать небольшое количество файлов журнала, которые можно быстро просмотреть или загрузить в локальную среду для последующего анализа.

Далее перечисляются некоторые доступные средства поиска и устранения неполадок и разные виды техники для Windows Azure.

**Средство диагностики Windows Azure (WAD).** Это средство выполняет сбор данных операций и диагностических данных службы Windows Azure. Средство диагностики Windows Azure заносит в журнал данные диагностики из различных источников данных, таких как журналы служб IIS 7.0, журналы инфраструктуры диагностики Windows, журналы событий Windows, счетчики производительности, аварийные дампы и журналы пользовательских ошибок. Записи в журнале производятся через регулярные настраиваемые интервалы времени, собранные данные сохраняются для анализа в таблице Windows Azure и в хранилище больших двоичных объектов. Дополнительные сведения см. в разделе [Общие сведения о диагностике Windows Azure](#).

**Пакет управления System Center Windows Azure (MP).** Пакет управления наблюдением Windows Azure позволяет осуществлять наблюдение доступности и производительности приложения, выполняющегося в Windows Azure. Дополнительные сведения см. в разделе [Руководство по пакету наблюдения для приложений Windows Azure](#). Для эффективного наблюдения доступности и производительности приложений Windows Azure рекомендуется использовать и диагностику Windows Azure, и пакет управления System Center Windows Azure. Дополнительный видеоролик см. в разделе [System Center 2012: управление приложениями в частном и общедоступном облаках](#).

**Аналитика хранилища Windows Azure.** Производит регистрацию и предоставляет данные учетной записи хранилища. Эти данные можно использовать для трассировки запросов, анализа трендов использования и для диагностики проблем с помощью учетной записи хранилища. Дополнительные сведения см. в разделе [Storage Analytics](#) в библиотеке MSDN.

**Управление подключением к базе данных SQL.** Обрабатывает коды ошибок путем реализации в приложении логики повторных попыток. Дополнительные сведения см. в разделе [Управление подключениями к базе данных SQL](#) на вики-сайте TechNet.

**Командлеты Windows Azure PowerShell.** Командлеты Windows Azure PowerShell позволяют просматривать и настраивать облако и службы управления данными Windows Azure, а также управлять ими непосредственно из PowerShell. Эти средства могут быть полезны

при разработке и тестировании приложений, которые используют службы Windows Azure. Дополнительные сведения см. в разделе [Командлеты Windows Azure PowerShell](#).

Как и для любого другого приложения, для облачной службы Windows Azure также требуется детальное тестирование перед отправкой в рабочую среду. Должны быть протестированы функциональные возможности, проверено сквозное исполнение, производительность, масштабируемость и пр.

Подробные конкретные рекомендации см. в разделе [Рекомендации по устранению неполадок в приложениях Windows Azure](#). Дополнительные сведения см. в разделах [Размещенные службы мониторинга и данные журналов в Windows Azure](#) и [Testing, Managing, Monitoring and Optimizing Windows Azure Applications](#) в библиотеке MSDN.

## Возможности подключения к облачной сети в Windows Azure

Windows Azure предлагает целый ряд возможностей подключения к сети для интеграции существующих приложений с облаком и для управления сетевым трафиком. Далее перечислены основные компоненты подключения к сети в Windows Azure.

**Шина обслуживания Windows Azure.** Для связи службы со службой в пределах Windows Azure, а также для обеспечения интеграции между локальными серверами и Windows Azure рекомендуется использовать шину обслуживания Windows Azure. Шина обслуживания предоставляет безопасные возможности обмена сообщениями и ретрансляции на уровне приложений. Шина обслуживания Windows Azure предлагает облачную инфраструктуру связи, которая поддерживает общую безопасную службу обмена сообщениями по общественной сети с простым унифицированным пространством имен, например <https://myhostname.servicebus.windows.net>. Шина обслуживания поддерживает следующие программные модели: .NET API, REST API и WCF.

**Служба управления доступом Windows Azure.** Для предоставления федеративного управления на основе утверждений для размещенных в облаке веб-служб WCF и REST и для приложений конечных пользователей рекомендуется использовать управление доступом Windows Azure. Служба управления доступом является службой Windows Azure, которая предоставляет несложный способ аутентификации пользователей, веб-приложениям и службам которых нужен доступ без включения в код сложной логики аутентификации. Служба обеспечивает также интеграцию с [Windows Identity Foundation \(WIF\)](#). Дополнительные сведения о ACS см. в разделе [Как проверять подлинность веб-пользователей с помощью службы управления доступом Windows Azure](#).



**Windows Azure Connect.** Чтобы разрешить безопасное подключение между локальными серверами и Windows Azure, рекомендуется использовать Windows Azure Connect. Благодаря этому предоставляется путь миграции в Windows Azure для существующих приложений путем включения сетевого соединения на основе IP с существующими локальными службами и с инфраструктурой. Кроме того, Windows Azure Connect упрощает прямое подключение к размещенным в облаке виртуальным машинам, разрешая тем самым выполнение удаленного администрирования и устранения неполадок посредством средств, используемых для локальным приложений. Дополнительные сведения см. в разделе [Соединение локальных компьютеров с ролями Windows Azure](#).

**Диспетчер трафика Windows Azure.** Диспетчер трафика Windows Azure позволяет производить распределение нагрузки по входящему трафику между несколькими размещенными в Windows Azure службами вне зависимости от того, выполняются они в одном центре обработки данных или в нескольких центрах в различных частях мира. Эффективное управление трафиком позволяет обеспечить высокую производительность, доступность и эластичность приложений. Диспетчер трафика Windows Azure в настоящее время доступен бесплатно в CTP-версии.

**Сеть доставки содержимого Windows Azure.** Сеть доставки содержимого Windows Azure (CDN) предлагает разработчикам глобальное решение в ближайших к клиентам или пользователям местоположениях, чтобы обеспечить лучшие условия работы приложения. CDN позволяет кэшировать данные больших двоичных объектов Windows Azure и выходные данные вычислительных экземпляров в стратегических местоположениях для доставки содержимого пользователям с максимально доступной пропускной способностью. Доставку CDN для поставщиков содержимого можно разрешить с использованием портала управления платформой Windows Azure. Дополнительные сведения см. в разделе [Общие сведения о CDN Windows Azure](#).

**Виртуальная сеть Windows Azure.** Начиная с CTP-версии Windows Azure, Windows Azure поддерживает эту новую службу для предоставления подключений типа «сеть-сеть» между локальными и облачными приложениями. Дополнительные сведения см. в разделе [Общие сведения о виртуальных машинах Windows Azure](#) в библиотеке MSDN.

**Синхронизация данных SQL Windows Azure SQL (синхронизация данных SQL).** В службе в настоящий момент имеются две основные возможности. Она позволяет синхронизировать данные между локальными базами данных SQL Server и базами данных в базе данных SQL Windows Azure, позволяя тем самым локальным и облачным приложениям использовать одни и те же данные. Кроме того, можно

синхронизировать данные между несколькими экземплярами базы данных SQL; базы данных могут быть в одном центре обработки данных, в разных центрах обработки данных или в разных регионах. Служба синхронизации данных SQL часто используется вместе с диспетчером трафика Windows Azure. **Предупреждение.** В настоящее время доступна только предварительная версия службы синхронизации данных SQL, предназначенная для составления отзывов для будущих версий. Использовать эту предварительную версию в рабочих средах не рекомендуется.

Дополнительные сведения см. в разделе [Сети и кэширование в Windows Azure](#).

## Развертывание приложений и управление приложениями в Windows Azure

После завершения разработки облачной службы или приложения скомпилируйте или передайте его в Windows Azure. Имеются два сценария развертывания.

- Новое развертывание
- Изменения конфигурации
- Добавочное обновление кода
- Существенное обновление

Windows Azure позволяет настроить несколько подписок с одной учетной записью Windows Azure. Это позволяет создать отдельные среды разработки и тестирования для службы. В этом случае рассмотрите развертывание службы сначала в тестовой подписке, а затем в производственной подписке. После развертывания облачной службы в производственной подписке можно использовать промежуточную среду для непрерывного тестирования. Как только служба будет готова к работе, ее можно перенести в рабочую среду.

### См. также

[Планирование и проектирование приложений для Windows Azure](#)

[Перемещение приложений в облако](#)

[Средства Windows Azure для Microsoft Visual Studio](#)

[Миграция приложений в Windows Azure](#)

# Общие сведения о службах управления данными в Windows Azure

**Авторы:** Сридхар Пеллур (Sreedhar Pelluru)

**Соавторы:** Джеймс Подгорски (James Podgorski), Сильвано Кориани (Silvano Coriani)

**Рецензенты:** Кристиан Мартинес (Christian Martinez), Стив Говард (Steve Howard), Кун Чен (Kun Cheng), Паоло Сальватори (Paolo Salvatori), Шон Эрнан (Shawn Hernan)

Платформа Windows Azure обеспечивает работу следующих служб управления данными.

Служба управления данными	Назначение
Служба таблиц Windows Azure	Обеспечивает долговременное хранилище для структурированных данных.
Служба больших двоичных объектов Windows Azure	Обеспечивает долговременное хранилище больших двоичных объектов, таких как видео или аудио.
База данных SQL Windows Azure	Система управления реляционными базами данных.

Эти предложения размещаются в дата-центрах Windows Azure и доступны для приложений независимо от того, запущены они локально, размещаются в дата-центре Windows Azure или в конкурирующей облачной службе. Предложения по хранению данных обеспечивают множество преимуществ, таких как высокий уровень доступности, масштабируемость, хорошая управляемость, неограниченное хранилище и безопасность. Дополнительные сведения см. в разделе [Управление данными Windows Azure](#).

Если код приложения запускается в дата-центре Windows Azure, виртуальная машина, на которой размещается приложение, обеспечивает дополнительный вариант хранения — диски Windows Azure. Windows Azure предоставляет долговременный диск, являющийся страничным большим двоичным объектом. Кроме того, на виртуальной машине можно использовать локальное хранилище, которое обеспечивает временное хранилище для экземпляра приложения.

## Служба таблиц Windows Azure

Служба таблиц предоставляет в облаке нереляционное структурированное хранилище с высокой степенью масштабируемости. Служба обеспечивает нереляционную коллекцию ключей и контейнеров свойств, которая может быть полезной для хранения табличных данных, таких как сведения о клиентах, заказы, новостные каналы и результаты игр. Если структурированные данные в настоящий момент хранятся в базе данных SQL Server или другом хранилище данных и не требуют вычислений на стороне сервера, таких как соединение, сортировка, представления и отсортированные процедуры, рассмотрите возможность хранения данных в таблицах Windows Azure. Дополнительные сведения см. в разделе [Перенос данных в хранилище таблиц](#).

## Служба больших двоичных объектов Windows Azure

Служба больших двоичных объектов позволяет хранить большое количество неструктурированных текстовых или двоичных данных, таких как видео, аудио- и видеофайлы. Если приложение хранит большие двоичные объекты в базе данных SQL Server или хранит большое количество неструктурированных данных в файловой системе, рассмотрите возможность использования службы больших двоичных объектов Azure. Дополнительные сведения см. в разделе [Перенос данных в хранилище больших двоичных объектов](#).

## Диск Windows Azure

Диск Windows Azure является страничным большим двоичным объектом, который содержит виртуальный жесткий диск (VHD) с форматом NTFS. Он позволяет существующим приложениям, которые используют файловую систему для хранения данных, работать на Windows Azure с минимальными изменениями кода. Если приложение хранит данные в файловой системе и нельзя перенести данные в службу таблиц или службу больших двоичных объектов Windows Azure, воспользуйтесь локальным хранилищем для непостоянного хранения или диском Windows Azure для постоянного хранения. Дополнительные сведения см. в разделе [Перенос данных на диски](#).

## База данных SQL Windows Azure

База данных SQL Windows Azure обеспечивает систему управления реляционными базами данных для платформы Windows Azure и основывается на технологии SQL Server. База данных SQL Windows Azure обеспечивает интерфейс с потоком табличных данных и Transact-SQL (T-SQL), так что многие средства и приложения, работающие с SQL Server, также будут работать с базой данных SQL Windows Azure. Приложения, написанные с

использованием существующих технологий, таких как ADO.NET и ODBC, для взаимодействия с SQL Server, можно обновить для доступа к экземплярам базы данных SQL Windows Azure с минимальными изменениями кода. База данных SQL Windows Azure также предоставляет стандартные функции SQL Server, такие как хранимые процедуры, представления, множественные индексы, соединения и агрегаты.

Если приложение использует базу данных SQL Server, можно легко выполнить миграцию базы данных в экземпляр базы данных SQL Windows Azure на платформе Windows Azure. Но если приложение использует функции SQL Server, которые база данных SQL Windows Azure не поддерживает, потребуется изменить решение базы данных. Подробные сведения см. в разделе [Миграция баз данных SQL Server в базу данных SQL Windows Azure](#).

## Сравнение базы данных SQL Windows Azure и табличного хранилища

Табличное хранилище хранит структурированные данные, так же как и база данных SQL. Поэтому при миграции приложений с локальной платформы на платформу Windows Azure часто возникает вопрос, использовать хранилище таблиц или базу данных SQL.

Основное различие между базой данных SQL и хранилищем таблиц заключается в следующем: база данных SQL — это система управления реляционными базами данных, которая обеспечивает возможности обработки данных через запросы, транзакции и хранимые процедуры, выполняемые на стороне сервера. В то время как хранилище таблиц не предоставляет хранилища реляционных данных и возможностей обработки данных, поддерживаемых базой данных SQL. Таким образом, если приложение хранит и извлекает большие наборы данных, но не требует обработки данных на стороне сервера, таблица Windows Azure является лучшим вариантом. Если приложению требуется обработка больших наборов данных, то база данных SQL подойдет лучше.

Есть несколько других факторов, которые необходимо рассмотреть перед выбором между базой данных SQL и хранилищем таблиц Azure. В следующей таблице сравниваются возможности хранилища таблиц Azure и базы данных SQL.

Критерии сравнения	Хранилище таблиц	База данных SQL
Максимальный размер сущности	Сущности в хранилище таблиц могут иметь размер не более 1 МБ и не более 255 свойств, которые включают три обязательных свойства: PartitionKey,	Размер строк может быть до 8 МБ с числом столбцов 1024.

Критерии сравнения	Хранилище таблиц	База данных SQL
	RowKey и Timestamp.	
Связи данных	Хранилище таблиц не предоставляет способа представления связей между данными.	Да. База данных SQL позволяет определить связи между данными, которые хранятся в различных таблицах с помощью внешних ключей.
Обработка на стороне сервера	Хранилище таблиц поддерживает базовые операции, такие как вставка, обновление, удаление и выбор. Оно не поддерживает соединения, хранимые процедуры, триггеры или обработку на стороне подсистемы хранилища, в отличие от базы данных SQL.	База данных SQL предоставляет стандартные функции SQL Server, такие как хранимые процедуры, представления, множественные индексы, соединения и статистические обработки.
Поддержка транзакций	Ограниченная Хранилище таблиц поддерживает транзакции для сущностей в той же таблице и в той же секции. В одной транзакции поддерживается до 100 операций. Хранилище таблиц поддерживает оптимистичный параллелизм.  Дополнительные сведения см. в разделе <a href="#">Групповые транзакции сущностей</a> .	Да. База данных SQL поддерживает типичные ACID-транзакции в той же базе данных. Распределенные транзакции в базах данных не поддерживаются. База данных SQL также поддерживает оптимистичный параллелизм.
Высокий уровень доступности	Да. Таблицы, хранимые на Windows Azure,	Да. Три копии экземпляров базы данных SQL хранятся в

Критерии сравнения	Хранилище таблиц	База данных SQL
и отказоустойчивость	реплицируются в три местоположения одного центра обработки данных для обеспечения устойчивости против сбоев оборудования.	выбранном центре обработки данных.
Георепликация	Да. Таблицы Windows Azure также реплицируются между двумя географически разделенными центрами обработки данных на одном континенте, обеспечивая таким образом дополнительную долговечность данных в случае серьезной аварии.	Нет. Экземпляр базы данных SQL по умолчанию не реплицируется в другие подчиненные регионы.
Максимальный размер данных	100 ТБ для каждой учетной записи хранилища. Учетная запись хранилища (вместе таблицы, большие двоичные объекты и очереди) может хранить до 100 ТБ данных. Таким образом, максимальный размер таблицы Azure равен 100 ТБ.	150 ГБ для каждой базы данных. Максимально допустимый размер базы данных в базе данных SQL в настоящее время составляет 150 ГБ. Для хранения данных большего объема воспользуйтесь <a href="#">федерациями</a> .
Протокол и средства управления	REST-интерфейс по протоколу HTTPS. Можно использовать <a href="#">обозреватель хранилищ Azure на сайте CodePlex</a> или другие средства сторонних производителей, включая <a href="#">Cloud Storage Studio</a> .	REST-интерфейс по протоколу HTTPS (или) TDS-интерфейс по протоколу SSL. Можно использовать портал управления Azure или среду SQL Server Management Studio для управления экземпляром базы данных

Критерии сравнения	Хранилище таблиц	База данных SQL
		SQL. Эти средства используют протокол TDS (поток табличных данных) по SSL-протоколу для соединения с базой данных SQL.
Доступ к данным	Данные, хранящиеся в хранилище таблиц, можно получить с помощью HTTP (S) API-интерфейса REST или клиентской библиотеки .net для службы данных WCF. См. раздел <a href="#">Как использовать хранилище таблиц</a> .	<p>Приложения, написанные с помощью существующих технологий, таких как ADO.NET и ODBC, для взаимодействия с SQL Server, можно использовать для доступа к экземплярам базы данных SQL с минимальными изменениями кода.</p> <p>Экземпляр базы данных SQL доступен для приложений, которые выполняются в Windows Azure, локально или в облачных платформах, отличных от Azure.</p>
Схема для таблицы	Нет фиксированной схемы. Каждая сущность (строка) может иметь различные свойства. Например, можно хранить сведения о заказе в одной строке и информацию о клиенте в другой строке той же таблицы.	Фиксированная схема для уже определенной таблицы. Все строки должны соответствовать правилам схемы.
Поддерживаемые типы данных	Массив байтов, Boolean, DateTime, Double, GUID,	См. раздел <a href="#">Типы данных, поддерживаемые базой</a>



Критерии сравнения	Хранилище таблиц	База данных SQL
	Int32, Int64, String	<a href="#">данных SQL</a> .
Стоимость	См. раздел <a href="#">Стоимость Windows Azure</a> .	См. раздел <a href="#">Стоимость Windows Azure</a> .
Поддержка Java API	Да	Да
Поддержка API-интерфейса Node.js	Да	Нет. Не поддерживается в текущей версии.
Аутентификация	Для проверки подлинности пользователей используется 256-разрядный симметричный ключ.	Для проверки подлинности пользователей, обращающихся к экземпляру базы данных SQL, используется аутентификация SQL.  Портал управления платформой Windows Azure для проверки подлинности пользователей использует Windows Live ID.
Подобно существующим локальным хранилищам данных.	Нет.	Аналогично SQL Server при некоторых <a href="#">ограничениях</a> .
Доступно из локальных приложений или приложений, размещенных на платформе Windows Azure.	Да	Да

## См. также

- [Службы управления данными Windows Azure](#)
- [Перенос данных в другие службы управления данными в Windows Azure](#)

- [Миграция баз данных SQL Server в базу данных SQL Windows Azure](#)

# Миграция баз данных SQL Server в базу данных SQL Windows Azure

---

**Автор:** Шон Тинлайн Джонс (Shaun Tinline-Jones)

**Рецензент:** Шон Хернан (Shawn Hernan)

В этом разделе описано, как выполнить миграцию локальной реляционной базы данных в База данных SQL Windows Azure. Описывается, как перенести и определения объектов данных в схемах, и данные в таблицах. Также рассказывается, как определить, какие объекты базы данных не поддерживаются в База данных SQL Windows Azure и какие изменения потребуется внести в приложение, чтобы использовать базу данных в База данных SQL Windows Azure.

## Общие сведения о процессе миграции

База данных SQL Windows Azure действует как служба Windows Azure, размещенная в центрах обработки данных Microsoft, поэтому она имеет операционную среду, отличную от среды экземпляра компонента SQL Server Database Engine, выполняющегося на локальном сервере. Хотя есть большое сходство между компонентом SQL Server Database Engine и База данных SQL Windows Azure, имеются также и определенные различия. Эти различия означают, что, если производится перемещение базы данных из экземпляра компонента Database Engine в База данных SQL Windows Azure, данный проект является проектом миграции, а не просто проектом перенесения базы данных от одного экземпляра в другой. Даже если база данных использует только объекты, поддерживаемые База данных SQL Windows Azure, могут потребоваться изменения, чтобы приложения, использующие базу данных, продолжали нормально работать с веб-службой.

При миграции должны быть рассмотрены следующие изменения.

1. Удалите зависимости базы данных от функций SQL Server, например репликация, которых нет в База данных SQL Windows Azure.
2. Удалите зависимости базы данных от типов объектов базы или синтаксиса Transact-SQL, например использование распределенных запросов, которые не поддерживаются в База данных SQL Windows Azure.
3. Если планируется использовать базу данных только в База данных SQL Windows Azure, при необходимости можно включить поддержку уникальных возможностей База данных SQL Windows Azure, например федерацию базы данных, чтобы использовать

возможности эластичного горизонтального масштабирования Windows Azure. Если планируется развернуть разные копии базы данных на локальных экземплярах SQL Server или База данных SQL Windows Azure, то используйте только функции и объекты, поддерживаемые в обеих средах.



### **Важно!**

Добавление поддержки уникальных возможностей База данных SQL Windows Azure может увеличить сложность процесса миграции. Рассмотрите возможность добавления этой поддержки в последующем проекте, если эта функция не является необходимой для размещения базы данных в База данных SQL Windows Azure.

4. Внесите необходимые изменения в приложения, использующие базу данных. Они делятся на три категории.
  - a. Измените код приложения, который зависит от каких-либо функций, которые были изменены в базе данных или удалены из базы данных.
  - b. Добавьте код приложения, необходимый для полного использования функций База данных SQL Windows Azure, добавленных в базу данных, например федерации.
  - c. Внесите в приложение изменения, необходимые для эффективной работы с базой данных, размещенной в среде База данных SQL Windows Azure. Например, перемещение базы данных с локального сервера в центр обработки данных Windows Azure может повлиять на задержку в сети, поэтому для приложения станет важным уменьшить объем данных, передаваемых по сети.
5. Продумайте процесс миграции, постройте пакеты, необходимые для использования этого пакета, затем запустите этот процесс.

Внесение обширных изменений в базу данных и приложения часто является наиболее большой расходной частью проекта миграции. Бизнес-требования к базе данных также должны хорошо соответствовать возможностям База данных SQL Windows Azure. Дополнительные сведения об определении того, является ли база данных хорошим кандидатом для миграции, см. в разделе [Планирование миграции](#).

В дополнение к миграции базы данных с локального экземпляра SQL Server Database Engine можно использовать помощник миграции SQL Server для переноса базы данных Oracle, MySQL или Access в База данных SQL Windows Azure.

## **В этом разделе**

В темах этого раздела даются подробные рекомендации по аспектам миграции базы данных в База данных SQL Windows Azure.

Описание	Раздел
Планирование и выполнение проекта по миграции База данных SQL Windows Azure, включая рекомендации по определению области изменений в базе данных и связанных с ней приложениях.	<a href="#">Управление проектом миграции базы данных SQL Windows Azure</a>
Обзор изменений в приложении, которые могут потребоваться для поддержки хорошего уровня производительности при перенесении базы данных в База данных SQL Windows Azure.	<a href="#">Вопросы производительности базы данных SQL Windows Azure</a>
Даются рекомендации по стратегиям обеспечения высокого уровня доступности и аварийного восстановления, чтобы защитить данные от ошибок пользователей, ошибок приложений, аппаратного сбоя, завершения работы центра обработки данных из-за стихийных бедствий и пр.	<a href="#">Вопросы высокого уровня доступности и аварийного восстановления с помощью базы данных SQL Windows Azure</a>
Выбор средств миграции и процессов, наиболее подходящих для конкретного проекта. Описывает шаги использования инструментов и процессов.	<a href="#">Выбор средств для миграции базы данных в базу данных SQL Windows Azure</a>

## Управление проектом миграции базы данных SQL Windows Azure

**Автор:** Шон Тинлайн-Джонс (Shaun Tinline-Jones)

**Соавтор:** Стив Ховард (Steve Howard)

**Рецензент:** Шон Хернан (Shawn Hernan)

В этом разделе описываются рекомендации по планированию и миграции базы данных на База данных SQL Windows Azure в составе проекта по миграции в Windows Azure.

Описывается оценка сложности миграции базы данных, задание целей проекта для миграции баз данных и управление подпроектом на этапах проектирования, разработки, теста, стабилизации и реализации.

## Введение

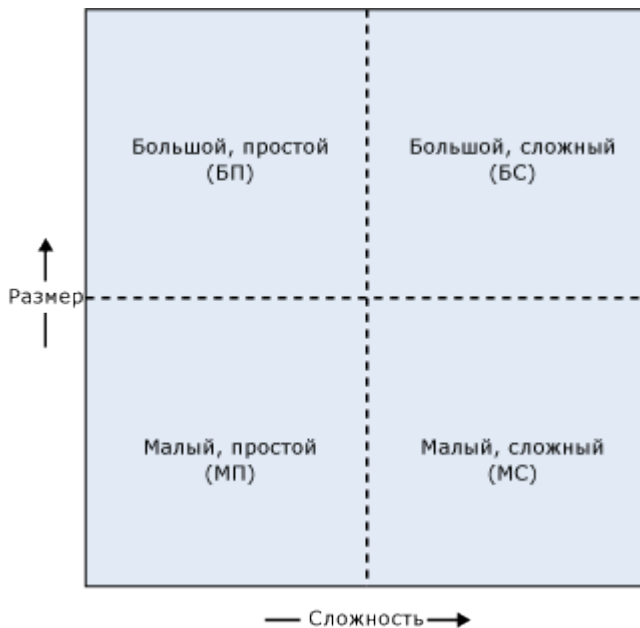
Миграция базы данных является подпроектом более крупного проекта миграции решения. Между проектами миграции базы данных и приложения обычно существуют точки интеграции и зависимостей. Миграция базы данных обычно может выполняться параллельно, хотя существует несколько узких мест.

При реализации проекта миграции База данных SQL следует иметь в виду три момента.

- Жизненный цикл проекта должен быть гибким и итеративным. Создайте первоначальный план на основании ранних исследований. В ходе планирования новой итерации уточните план на основании исследований, проведенных в предыдущих итерациях.
- Размер и сложность базы данных и связанных с ней приложений влияют на многие факторы в проекте миграции.
  - Сложность базы данных влияет на инженерные исследования, необходимые для миграции базы данных.
  - Размер базы данных, объем данных, которые она содержит, влияют на то, сколько потребуется времени для заполнения новой базы данных и переключения с локальной базы данных на базу данных, размещенную в База данных SQL.
- При миграции базы данных на новую платформу часто требуются изменения в базе данных, которые влияют на другие уровни решений, использующих базу данных. Проект миграции должен координировать разработку на всех затронутых уровнях и обеспечивать унифицированное развертывание всех измененных компонентов.

Каждую перенесенную базу данных можно отнести к одному из четырех квадрантов, определяемых размером и сложностью. Квадрант, в который попадает база данных, помогает оценить масштаб проекта, необходимого для миграции базы данных на База данных SQL, и выбрать хороший механизм для миграции. Квадранты могут быть следующими:

## Размер проекта и квадранты сложности



Большой базе данных потребуется больше времени для переключения на База данных SQL, поскольку для передачи данных через Интернет потребуется больше времени. Дополнительная сложность базы данных означает большую вероятность того, что потребуется внести изменения, а это потребует большего объема инженерных работ.

Понимание размера и сложности базы данных-источника является ключевым аспектом задания цели для проекта миграции.

## Анализ

На этапе анализа устанавливаются цель и видение проекта. Общие цели проекта должны включать цели для перенесенной базы данных.

## Бизнес-требования

Все базы данных должны соответствовать бизнес-требованиям, таким как доступность, возможность восстановления, время ответа и соответствие правилам безопасности и конфиденциальности. При переносе базы данных на База данных SQL, настройте службу базы данных таким образом, чтобы она соответствовала бизнес-требованиям, или обсудите новый набор требований, которые могут подходить для База данных SQL. Возможно, придется изменить процессы администрирования. Например, если в данный момент резервные копии базы создают данные каждую ночь, может потребоваться

перейти на использование функций копирования базы данных или экспорта приложений уровня данных, поддерживаемых База данных SQL.

Бизнес-требования нельзя определить путем анализа существующей базы данных и кода приложения. Необходимо собрать требования от заинтересованных лиц и администраторов и проанализировать технологические документы, такие как соглашения об уровне обслуживания.

## **Определите цель подпроекта базы данных**

Цели проекта миграции Windows Azure, связанные с миграцией баз данных, должны соответствовать бизнес-требованиям для базы данных и отражать размер и сложность базы данных. Сложные базы данных требуют больших инженерных усилий для переноса, чем простые базы данных. Проект миграции сложной базы данных может снизить риск путем ограничения первоначального проекта только миграцией функций, используемых в локальной базе данных. Включение функций, уникальных для База данных SQL, например федераций, можно выполнить в последующих проектах.

Этап анализа обеспечивает руководство более высокого уровня для этапа планирования и проектирования. Во время этапа анализа важно рассмотреть весь набор вопросов, которые могут повлиять на миграцию, но не слишком углубляться в детали. Затем в ходе первой итерации этапов планирования и проектирования необходимо более глубоко рассмотреть эти вопросы для формирования более детализированных проектов и планов. Реализуйте процесс обратной связи для регулировки видения и определения документов с результатами работы раннего планирования и проектирования.

## **Оценка сложности проекта**

Оценка сложности проекта миграции База данных SQL означает определение объема изменений, необходимых для выполнения успешной миграции. Различные этапы в проекте по миграции База данных SQL требуют все более точных оценок масштабов инженерных изменений, обусловленных миграцией. Изначальная общая оценка должна быть учтена в определении цели проекта и решении запуска проекта. Она также образует основу ранних усилий по планированию и проектированию проекта. Результаты более углубленных исследований, проведенных позднее в проекте, должны быть отражены в более подробных планах проекта и, возможно, во внесении изменений в цели проекта.

## **Зависимости от функций, не поддерживаемых базами данных SQL Windows Azure**

Все зависимости от функций, не поддерживаемых База данных SQL, следует учитывать в рамках проекта миграции. Первичное определение этих зависимостей можно выполнить

без необходимости в доступе к рабочей системе. Это делается путем сравнения существующей документации о функциях, поддерживаемых База данных SQL, с проектной документацией базы данных или спецификацией приложения. Документацию также могут рассмотреть люди, знакомые с конструкциями базы данных и приложения. Позднее некоторые виды зависимостей можно будет подтвердить с помощью таких инструментов, как мастер миграции База данных SQL.

Многие локальные базы данных имеют зависимости от служб вне базы данных. Примеры включают участие в топологии репликации, процесс извлечения служб Integration Services SQL Server или повторяющиеся задачи обслуживания, управляемые агентом SQL Server. Проект миграции должен включать затраты и время разработки, необходимое для изменения зависимостей от таких служб. Устраните зависимости от любых служб, которые не поддерживают База данных SQL. Возможно, придется внести изменения в другие системы, которые поддерживают База данных SQL из-за архитектурных различий между SQL Server и База данных SQL.

Кроме того, локальные базы данных могут иметь объекты Transact-SQL, которые не поддерживаются в База данных SQL. Приложения, осуществляющие доступ к базе данных и коду в таких объектах базы данных, как хранимые процедуры или триггеры, также могут использовать элементы синтаксиса, не поддерживаемые База данных SQL.

Первоначальную оценку сложности базы данных можно выполнить, рассмотрев проекты или код базы данных и приложения на предмет наличия проблем, обсуждаемых в следующих статьях:

- [Общие рекомендации и ограничения \(база данных SQL Windows Azure\)](#)
- [Безопасность: рекомендации и ограничения \(база данных SQL Windows Azure\)](#)
- [Ограничения функций SQL Server \(база данных SQL Windows Azure\)](#)
- [Поддержка средств и программ \(база данных SQL Windows Azure\)](#)
- [Неподдерживаемые инструкции Transact-SQL \(база данных SQL Windows Azure\)](#)
- [Частично поддерживаемые инструкции Transact-SQL \(база данных SQL Windows Azure\)](#)
- [Поддерживаемые инструкции Transact-SQL \(база данных SQL Windows Azure\)](#)

## **Масштаб изменений приложений, обусловленных изменениями в базе данных**

При миграции базы данных на База данных SQL часто требуется внести изменения в приложения и системы, которые используют базу данных.



Сначала необходимо сделать изменения, необходимые для эффективной работы приложений в среде База данных SQL. База данных SQL — это веб-служба, размещаемая вне вашего центра обработки данных. Это означает, что некоторые рекомендации для баз данных, которые имеют малое значение при размещении сервера базы данных в одной стойке с сервером приложений, становятся более важными, когда база данных переносится в База данных SQL. Каждая база данных, размещенная в База данных SQL, кластеризована на нескольких серверах для повышения общей доступности. Однако некоторые операции или сбой на вашем текущем сервере могут вызвать событие временной отработки отказа, которая закроет все открытые соединения и выполнит откат их активных транзакций. Это делает важным наличие в ваших приложениях надежной логики повторных попыток, перезапускающей транзакции, когда приложение получает ошибку отключения.

Дополнительные сведения об изменениях в приложении, необходимых для поддержки высокой производительности, см. в разделе [Вопросы производительности базы данных SQL Windows Azure](#).

Дополнительные изменения в приложениях, необходимые для эффективной работы с База данных SQL, обсуждаются в следующих документах:

- [Руководство по доставке базы данных SQL Windows Azure](#)
- [Обработка транзакций в базе данных SQL Windows Azure](#)

Также необходимо сделать изменения в приложениях, необходимые для подстройки ко всем изменениям, внесенным в базу данных. Например, если определяемый пользователем агрегат удаляется из базы данных, необходимо изменить любое приложение, которое выполняет инструкции Transact-SQL, которые ссылаются на агрегат.

## План и проект

Работа по планированию и проектированию должна начаться в ходе оценки сложности. Важнейшей частью этого этапа является выполнение все более подробных исследований вопросов, рассмотренных в этих двух разделах:

- [Зависимости от функций, не поддерживаемых базами данных SQL Windows Azure](#)
- [Масштаб изменений в приложениях, обусловленных изменениями в базе данных](#)

По мере определения функций, которые должны быть изменены, сформулируйте первоначальные оценки работы, необходимой для внесения этих изменений на ранних итерациях проекта. Каждая последующая итерация должна выполнить более всеобъемлющий обзор базы данных для выявления всех изменений и сформулировать

более детальные определения масштаба требуемых изменений. Настройте процесс обратной связи, в котором цели и масштаб проекта могут корректироваться с учетом результатов более подробных исследований, проведенных в ходе итераций планирования и проектирования. Первой итерации не требуется доступ к рабочей среде, но нужно достаточно точное отражение производственной нагрузки.

Можно реализовать это первоначальное представление планирования, используя мастер миграции База данных SQL и наращивая различия между локальной версией SQL Server и База данных SQL. Они обсуждаются более подробно в соответствующих рубриках. SQL Server Data Tools (SSDT) является потенциально полезным средством на данном этапе, особенно если управление жизненным циклом приложения (ALM) на уровне данных включает в себя использование этого средства или скриптов объектов базы данных. Объем необходимых усилий не слишком велик — он немногим больше, чем необходим для мастера миграции База данных SQL, и не является слишком обременительным. Необходимо отметить элегантность обычных функций производительности Visual Studio, например при двойном щелчке предупреждения выполняется переход непосредственно к строке с ошибкой.

Оценку различных вопросов, таких как поддержка Transact-SQL, можно проверить с использованием ряда инструментов.

- Можно подключить мастер миграции База данных SQL к тестовой или производственной копии локальной базы данных и сформировать отчет об объектах, которые следует изменить для работы на База данных SQL. Дополнительные сведения см. в разделе [Как использовать мастер миграции SQL Azure](#).
- Если все объекты в локальной базе данных поддерживаются приложением уровня данных (DAC), можно извлечь пакет приложения уровня данных и выполнить одно из следующих действий.
  - Запустить пакет приложения уровня данных посредством службы оценки совместимости База данных SQL для отчета о требуемых изменениях (в настоящее время в состоянии бета-версии).
  - Импортировать пакет приложения уровня данных для создания проекта базы данных в SQL Server Data Tools и задать База данных SQL в качестве цели проекта.
  - Дополнительные сведения см. в разделе [Как использовать пакет приложения уровня данных для миграции базы данных в базу данных SQL Windows Azure](#).

Хотя инструменты оценки могут помочь определить функции, которые не поддерживаются в База данных SQL, они необязательно определяют альтернативные функции, которые можно использовать для обеспечения тех же функциональных возможностей. Планировщики проекта должны понимать коммерческое применение функциональных

возможностей и конструктивных альтернативных возможностей, которые также будут поддерживать такое применение. Решение о продолжении дальнейших работ должно быть принято на основе оценки стоимости разработки и внедрения альтернатив. Не следует просто отказываться от использования База данных SQL ввиду отсутствия поддержки функций.

Избегайте включения задач, не связанных с миграцией, особенно для сложных миграций. Добавление сложности является часто встречающейся причиной неудачи проектов миграции. Часто имеется желание использовать масштабируемую модель баз данных. Делать это рекомендуется только при необходимости, например в следующих случаях.

- Выполняется миграция базы данных, которая больше, чем максимальный размер, поддерживаемый База данных SQL.
- Экономическая модель жизнеспособна только в случае, если мультитенантность реализуется изначально.
- Требования к вычислительным ресурсам базы данных превышают возможности одной базы данных База данных SQL.

Для масштабной федерации требуется изменение модели данных, что влияет на все уровни решения. Эластичное масштабирование или мультитенантность часто являются допустимыми для облака, и их изначальная реализация может быть заманчивой. Но попытка реализации такого уровня изменений часто изменяет характер проекта с миграции на реализацию новых функций, что увеличивает степень риска.

Планирование и проектирование должны включать соображения стоимости. Стоимостные факторы необходимо оценивать в течение каждого этапа и с каждой итерацией. Они должны составлять важную часть каждого решения о том, продолжать ли работу. Этот элемент может быть исключен из планирования и проектирования, поскольку вероятность того, что этот риск станет проблемным, низка, однако последствия недооценки расходов могут быть довольно тяжелыми.

Ключевым фактором успеха является выявление рисков и назначение им вероятностей и серьезности. Перечислите все риски, даже те, которые могут показаться тривиальными. Убедитесь, что все заинтересованные лица согласны с тем, что более вероятные риски были надлежащим образом оценены. Убедитесь, что для каждого возможного риска существует план по минимизации ущерба с приемлемым уровнем планирования на случай, если такая проблема возникнет. Создайте процесс добавления планов для новых рисков, найденных во время всех этапов миграции. Проблемы, найденные и решенные на этапе окончательного развертывания, должны быть зарегистрированы как риски для

будущих проектов. Также важно оценить риски в среде облака, а не только в более привычной локальной среде.

Важно, чтобы проект имел всеобъемлющий обзор всех функций, используемых базой данных, в сравнении с набором функций, поддерживаемых в База данных SQL, в сочетании с оценками затрат преобразования и издержек ведения базы данных в База данных SQL. Этот обзор должен выполняться на достаточно ранних итерациях, что позволило бы отменить проект в случае, если затраты превышают выгоды. Один проект миграции База данных SQL столкнулся с трудностями относительно поздно, потому что планировщики проекта не знали, что сжатие данных, используемое в локальной базе данных, не поддерживается в База данных SQL. Это значительно повлияло на прогнозируемые издержки ведения базы данных База данных SQL. Такой тип проблемы следует определять на ранних стадиях проекта.

Каждая итерация этапов планирования и проектирования должна включать более глубокую оценку изменений на уровне данных. Например, вторая итерация может включать создание трассировки профилировщика среды функционального тестирования и его запуск посредством мастера миграции SQL Azure. Третья итерация может перейти в среду тестирования производительности, в которой средства наблюдения производительности включены в область инструментов для определения возможных областей элементов, готовых к миграции.

База данных SQL не поддерживает функции SQL Server 2000 и SQL Server 2005, которые были удалены из SQL Server 2008 и более поздних версий. Например, База данных SQL не поддерживает синтаксис \*= и \*= для указания соединений. Таким образом, миграция этих баз данных на База данных SQL должна также решать многие из вопросов, возникших при обновлении с SQL Server 2000 или SQL Server 2005. Можно использовать инструменты, такие как счетчики системного монитора, XEvents и помощники по обновлению SQL Server, чтобы найти эти зависимости. Дополнительные сведения об исследованиях этих вопросов см. в статье [Обновление компонента Database Engine](#).

## Разработка

Разработка — это отдельная операция для выполнения задач, созданных на этапе планирования и проектирования. Людям, которым назначено выполнение задач разработки, не должны назначаться задачи других этапов проекта миграции.

При перенесении большинства баз данных в База данных SQL необходимо вносить изменения, влияющие на уровни приложений. При изменении таких вещей, как типы данных, число возвращаемых столбцов, динамический Transact-SQL или входные параметры, появляется необходимость внесения изменений на уровне данных для кода

приложения. Даже если объекты базы данных не изменяются в процессе миграции, архитектура База данных SQL обязывает внести изменения в приложения, например для реализации надежной логики повторных попыток и обработки ошибок. Короче говоря, разработка баз данных должна быть интегрирована с разработкой на уровне приложений.

Разработка базы данных может осуществляться с использованием любых средств разработки, которые поддерживают базы данных SQL Server. Преимуществом использования таких инструментов, как SSDT, содержащих логику для База данных SQL, является то, что можно установить База данных SQL в качестве цели построения проекта базы данных, что позволит SSDT определять несовместимый синтаксис по мере написания кода. Дополнительные сведения см. на странице [Microsoft SQL Server Data Tools](#). До версии SSDT разработчики использовали комплект эмуляции Windows Azure и подключались к SQL Server Express. Это обеспечивает удобство и дает некоторое ощущение разработки вне сети, но это все равно набор локальных функций, поэтому может быть непонятно, возможно ли это в База данных SQL. Более продуктивным и эффективным подходом к выявлению проблем является их выявление как можно ближе ко времени планирования. Они должны быть уже выявлены ко времени написания кода. Если для разработки используется инструмент, который содержит логику для База данных SQL, то можно быстро начать разработку в База данных SQL. Средства разработки вне сети, такие как SSDT, обеспечивают преимущество параллельной работы нескольких разработчиков над одним проектом. SSDT также интегрируется с исходными функциями сборок и системы управления версиями, такими как доступные в Team Foundation Server. Если миграция влияет и на код приложения, и на код базы данных, проект базы данных может быть интегрирован в ту же среду сборки. Если миграция влияет и на приложения, и на базы данных, проект SSDT может быть интегрирован в то же решение, что и процессы построения проектов приложений. При разработке непосредственно для База данных SQL эти преимущества доступны в дополнение к решению очевидных задач соединения.

Если миграция требует относительно мало изменений в модели данных, импортируйте эту модель данных в SSDT, а затем приступите к применению этих изменений. Полезным в данном случае является то, что индивидуальные разработчики могут работать в соответствующих областях специализации и интегрировать их во время этапа сборки. Если решение База данных SQL требует наличия федеративной модели данных, то это определяет количество объектов базы данных, которые нужно будет изменить, но по большей части создать с нуля.

Разработка федеративных решений все еще является довольно сложной задачей, поскольку ни одно из средств разработки не обрабатывает их должным образом. Также встречаются случаи, когда инструменты не обнаруживают проблему, поэтому рекомендуется строить и развертывать последние изменения в База данных SQL в рамках

ежедневной рутины построения. Отказ от развертывания может быть привлекательным, поскольку это можно считать «тестированием развертывания», а не «тестированием сборки», однако, хотя средства в данный момент еще дорабатываются, продемонстрировать готовность Windows Azure можно несколькими простыми шагами развертывания, просто создав базу данных в База данных SQL. Есть многие другие риски, которые делают тестирование развертывания необходимой и важной частью тестирования.

В каждой итерации проекта команда разработчиков должна давать команде планирования регулярные отзывы и активно реагировать на циклы планирования. Необходимо признать, что менее рискованно выполнять проверки регулярно, а затем при неудаче быстро восстановиться, чем долгое время писать код без проверок. Это не новая парадигма. Нежелание придерживаться ей при разработке облачных решений может обойтись очень дорого.

## Тест

Подобно другим действиям в жизненном цикле приложения проекта миграции, существуют действия и цели, которые остаются постоянными независимо от того, выполняется ли миграция на База данных SQL. Следующие области тестирования являются важными для База данных SQL, однако часто не рассматриваются должным образом планировщиками и разработчиками.

- Обработка ошибок
- Логика повторных попыток
- Действия при регулировании
- Развертывание изменений
- Парадигма масштабирования
- Эффекты задержки в сети
- Безопасность
- Ведение журналов

Обратите внимание на эти области, поскольку они обеспечивают основу, из которой могут быть получены варианты использования для теста функциональности и производительности. Существует много инструментов для тестирования, важно, чтобы в них была возможность изолировать действия базы данных. Функциональное тестирование увеличит производительность, смягчив незрелость средств разработки для точного

отслеживания функциональности SQL Server, недоступной в База данных SQL. В зависимости от средств разработки и механизмов сборки функциональное тестирование может не выявить проблем и просто послужить в качестве дополнительной страховки перед выполнением более длительного и дорогостоящего теста производительности.

Тестирование перенесенного решения должно рассматривать новые случаи использования, которые имели незначительное влияние для локальных реализаций. Создавайте тесты, принудительно реализующие ошибки, которые требуют повторного выполнения транзакций, и проверьте влияние максимальных и пиковых нагрузок. Хорошие решения на основе облака помогают уменьшить время устранения неполадок. Это достигается путем разработки логики приложения, которая регистрирует все действия и регулярно анализирует текущее состояние системы. Поскольку ведение журнала действий в базе данных может быть дорогостоящим и ограничивается записью в другую таблицу, код приложения, который выполняет вызовы базы данных, должен выполнять протоколирование ошибок, предупреждений и длительности. Например, клиент провел несколько часов, устраняя неполадки в нескольких серверах, имевших низкую производительность. Он пытался использовать многие инструменты для определения причины проблемы с целью ее разрешения. После нескольких часов серверы внезапно начали работать должным образом. Проблема не имела никакого отношения к их приложению, просто выполнялось обновление платформы. Усилия, затраченные на разрешение проблемы, можно было бы значительно сократить, если бы приложение было лучше спроектировано и протоколировало бы точки данных, а администраторы регулярно выполняли анализ состояния решения. Они могли бы предоставить инженеру службы поддержки более качественные данные, которые облегчили бы определение причины проблемы. Конечно, в будущем данные этого типа могут использоваться для перенаправления объектов-получателей в другой центр данных.

Область, о которой можно легко забыть, — это выполнение развертывания. Тестирование должно учитывать выполнение развертывания и обеспечивать понимание и уверенность в том, что последующие изменения можно будет применить к существующей среде. Процесс развертывания новой базы данных, ее заполнения и подготовки к производственному использованию довольно сильно отличается от процесса развертывания изменений в существующей рабочей среде. Это ничем не отличается от вопросов в локальной среде, но упоминается по двум причинам.

- Действия развертывания в локальной среде, могут не работать для облачного решения.
- Для включения этих действий в план теста требуется заметный объем работ, однако немедленная выгода практически не обеспечивается.

Тестирование также должно быть включено в итеративную модель, в которой сообщения о проблемах передаются обратно командам разработки и планирования. В начале испытания могут быть довольно простыми и очень сходными с тестом в локальной среде. На каждой итерации включайте все большее число проверочных вариантов, учитывающих архитектуру облака. Проверочные варианты должны охватывать проблемы, определенные в документации, упоминаемой в вышеуказанном разделе «Оценка сложности».

Сложность миграции может вызываться ограничением времени простоя, что увеличивает приоритет тестирования предлагаемого плана развертывания.

## Стабилизация

Цель данного этапа ничем не отличается от его цели в обычном жизненном цикле разработки программного обеспечения. Для проекта миграции главной задачей является достижение точки, где разработчики работают только над проблемами, выявленными в тестах, а не выполняют кодирование первой перенесенной версии базы данных.

## Развертывание

Подобно стабилизации, этап развертывания для миграции в облако ничем не отличается от локальных проектов миграции. Хорошее тестирование увеличивает вероятность успеха на этом этапе.

Хорошая связь — это ключевой фактор успеха для миграции. Результаты регулярного обновления состояния с соответствующим уровнем детализации, поставляемые объекту-получателю, имеют жизненно важное значение для хорошей работы или по крайней мере для того, чтобы постоянно знать о широких бизнес-целях миграции в облако.

Успех на этапе развертывания зависит от качества работы, проделанной в ходе предыдущих этапов. Плохо проведенные исследования, планирование, разработка и тестирование значительно увеличивают риск возникновения проблем во время развертывания. Иногда проблемы развертывания достаточно серьезны, чтобы остановить проект и вызвать откат к локальным системам. В некоторых случаях они заставляли организации ставить под сомнение собственную способность использования облачных систем. Хорошие исследования, планирование, разработка и тестирование обычно приводят к успешному развертыванию, выполняемому в соответствии с планом. Даже если возникают проблемы, при хорошем планировании часто предусмотрены встроенные меры по устранению проблем, уменьшающие последствия от них.



# Вопросы производительности базы данных SQL Windows Azure

---

**Авторы:** Сильвано Кориани (Silvano Coriani), Стив Говард (Steve Howard)

**Рецензенты:** Марк Симмс (Mark Simms), Валерий Мизонов (Valery Mizonov), Кун Чен (Kun Cheng), Паоло Сальватори (Paolo Salvatori), Хайме Альва Браво (Jaime Alva Bravo)

Эта статья охватывает рекомендации для приложений, которые позволяют повысить производительность приложений, использующих базу данных, перенесенную в База данных SQL Windows Azure. Если сервер базы данных смонтирован в той стойке, что и сервер приложений, эти рекомендации могут не иметь большого значения. Но если сервер базы данных размещен в удаленном центре обработки данных, эти рекомендации могут иметь решающее значение для обеспечения высокой производительности. Кроме того, Windows Azure является общей средой, в которой все ресурсы используются совместно с другими приложениями, ролями и базами данных. Посредством использования компонентов сетевой балансировки нагрузки и шлюза Windows Azure обеспечивает экономическую эффективность масштабирования и обеспечивает эффективное использование вычислительных и сетевых ресурсов для окружающей среды в целом. Эти факторы следует принимать во внимание при разработке и развертывании приложений в Windows Azure.

Эта статья описывает разработку и реализацию рекомендаций для оптимизации производительности в среде База данных SQL Windows Azure. В частности, в этой статье рассматриваются рекомендации для решения проблем в двух областях, которые могут вызвать проблемы с производительностью при перенесении локальных баз данных в База данных SQL Windows Azure:

- Управление подключениями
- Задержки в сети между уровнями приложений и баз данных

## Управление подключениями в База данных SQL Windows Azure

Подключения к базе данных могут завершаться более часто, если база данных размещается в База данных SQL Windows Azure, а не в локальной среде. Пользователи могут воспринимать эти завершения как проблемы производительности, если приложения не могут быстро обнаруживать потери соединения и восстанавливать их в случае временных ошибок. Как поставщик службы мультитенантных баз данных на общих ресурсах, База данных SQL Windows Azure кластеризует каждую базу данных на трех узлах

и распределяет ресурсы между узлами кластера для обеспечения эффективной работы для всех пользователей. Примером переходной ошибки является обнаружение База данных SQL Windows Azure интенсивного использования сервера, на котором размещается несколько баз данных. В этом случае База данных SQL Windows Azure может перенести одну из баз данных на вторичный узел кластера, который имеет более низкую рабочую нагрузку. Отработка отказа закрывает все открытые соединения с базой данных и выполняет откат необработанных транзакций. Приложения должны быстро обнаружить ошибку, повторно подключиться к базе данных и повторить свою последнюю транзакцию.

Далее описываются некоторые из причин, по которым База данных SQL Windows Azure может завершать соединения.

- Общая топология сети База данных SQL Windows Azure включает межсетевые экраны, подсистемы балансировки загрузки и шлюзы потока табличных данных (TDS). Каждый из этих компонентов топологии добавляет слой между кодом доступа к данным и узлом базы данных. Ошибки в этих дополнительных слоях могут завершить соединения.
- База данных SQL Windows Azure непрерывно собирает и анализирует статистические данные использования базы данных. На основании этих статистических данных База данных SQL Windows Azure может завершать соединения при необходимости сохранить службу в работоспособном состоянии.
- В результате атаки типа «отказ в обслуживании» База данных SQL Windows Azure может блокировать соединения с определенным IP-адресом на определенный период времени.
- Некоторые события отработки отказа могут вынудить База данных SQL Windows Azure внезапно завершить сеанс. Обратите внимание, что все соединения, открытые на узле до события отработки отказа не будут доступны на новом узле после отработки отказа.

Здесь описаны лишь некоторые из причин завершения соединений. Дополнительные сведения об ошибках подключения и сведения о том, как управлять соединениями в База данных SQL Windows Azure, см. в следующих документах:

- [Управление подключениями к базе данных SQL Windows Azure](#) — в этой статье на вики-сайте TechNet предоставляется полный список ошибок соединения, советов по устранению неполадок и рекомендаций.
- [Ошибки потери соединения](#) — в этом разделе темы о сообщениях об ошибках в библиотеке MSDN предоставляется обновленный список большинства ошибок соединения.
- [Регулирование базы данных SQL Windows Azure и расшифровка кодов ошибок](#) — в этом разделе темы о сообщениях об ошибках в библиотеке MSDN объясняется, как декодировать коды ошибок, появляющиеся в результате регулирования ресурсов.

- [Руководство по обеспечению производительности и эластичности базы данных SQL Windows Azure](#) — эта статья на вики-сайте TechNet дает дополнительную информацию о вопросах производительности.

## Варианты обработки управления соединениями в коде

Чтобы помочь в разрешении проблем с соединениями в База данных SQL Windows Azure, в корпорации Майкрософт разработали следующие функциональные возможности:

- Последовательный подход к указанию основных сведений, таких как имена серверов и учетные данные безопасности или полная точность с использованием таких инструментов, как программа массового копирования (bpc.exe). Например, начиная с версии 11 собственного клиента SQL Server, JDBC версии 4.0 и поставщика данных платформы .NET Framework для SQL Server (**System.Data.SqlClient**) с версии 4.0 платформы .NET Framework, уже не нужно указывать имя\_пользователя@сервер при доступе к База данных SQL Windows Azure.
- Акцент на обеспечение возможностей поддержания соединения для всех технологий, даже в периоды простоя. Например, в отличие от Windows платформа Java не может изначально управлять интервалами проверки активности для подключения к базе данных. Поэтому компоненты JDBC, устанавливающие подключение с База данных SQL Windows Azure, требуют некоторых изменений параметра реестра для обеспечения того, чтобы неактивные соединения не удалялись.  
Дополнительные сведения см. в разделе библиотеки MSDN [Соединение с базой данных SQL Windows Azure](#).

Кроме того, корпорация Майкрософт неизменно разворачивает обновления в наиболее распространенных библиотеках доступа к данным и в наборах исправлений База данных SQL Windows Azure. Возможно, наиболее важным из этих обновлений является [Программный блок обработки временных сбоев](#) — библиотека приложений, которая обеспечивает надежную логику обработки временных ошибок. Временные сбои — это ошибки, которые возникают из-за некоторых временных условий, например из-за проблем с сетевым подключением или недоступностью службы. В следующем разделе приводится рассмотрение того, как можно применить программный блок обработки временных сбоев к приложению.

## Краткое рассмотрение способа использования программного блока обработки временных сбоев

Программный блок обработки временных сбоев инкапсулирует сведения о временных ошибках, которые могут возникнуть при использовании следующих служб Windows Azure в приложении:

- База данных SQL Windows Azure

- Шина обслуживания Windows Azure
- Хранилище Windows Azure
- Служба кэширования Windows Azure

Каждая из этих служб может иметь различные временные сбои. В связи с этим программный блок обработки временных сбоев использует отдельные политики обнаружения ошибок для каждой службы. Аналогичным образом для различных приложений требуется использование различных стратегий обработки ошибок. Для обработки этих различий в программном блоке обработки временных сбоев доступны различные подходы к логике повторных попыток, которые используются для различных сценариев временных ошибок. Эти готовые политики можно расширить путем создания пользовательских классов, которые реализуют хорошо определенные интерфейсы.

Использование программного блока обработки временных сбоев внутри существующих приложений имеет очень незначительное воздействие на производительность. Вследствие своей конструкции программный блок обработки временных сбоев предлагает несколько классов и методов расширения, которые имитируют поведение типичного слоя доступа к данным ADO.NET.

Чтобы продемонстрировать легкость использования этой библиотеки приложений, в следующих нескольких пунктах показывается, как применить программный блок обработки временных сбоев к существующему коду. В следующем примере показан простой метод, в котором запрашивается база данных и потребляющий результирующий набор.

```
public static void ReadFromDB()
{
    using (SqlConnection conn = new SqlConnection(connString))
    {
        try
        {
            conn.Open();

            SqlCommand selectCommand =
new SqlCommand(@"SELECT SOH.SalesOrderID
                FROM SalesLT.SalesOrderHeader SOH
```

```

        JOIN SalesLT.SalesOrderDetail SOD ON SOH.SalesOrderID =
SOD.SalesOrderID

        JOIN SalesLT.Product P ON SOD.ProductID = P.ProductID

        JOIN SalesLT.Customer C ON SOH.CustomerID = C.CustomerID

        JOIN SalesLT.CustomerAddress CA on C.CustomerID = CA.CustomerID

        JOIN SalesLT.Address A on CA.AddressID = A.AddressID

        WHERE A.City=@City", conn);

        selectCommand.Parameters.Add(new SqlParameter("@City",
SqlDbType.VarChar, 20, ParameterDirection.Input, false, 0, 0, "", DataRowVersion.Current,
"London"));

        selectCommand.CommandType = CommandType.Text;

        IDataReader dataReader = selectCommand.ExecuteReader();

        while (dataReader.Read())
        {
            Console.WriteLine("OrderID: {0}", dataReader["SalesOrderID"]);
        }
    }
    catch (Exception e)
    {
        Console.WriteLine("Exception: {0}", e.Message);
    }
}

```

Чтобы сделать этот код более надежным, сначала определите подходящую стратегию повторных попыток. Лучше всего использовать добавочную стратегию повторных попыток. Для реализации этой стратегии сначала используйте класс `Microsoft.Practices.EnterpriseLibrary.WindowsAzure.TransientFaultHandling.SqlAzure.SqlAzureTransientErrorDetectionStrategy`. Этот класс перехватывает коды ошибок, которые связаны с условиями временного сбоя. Затем замените класс `System.Data.SqlClient.SqlConnection` классом `Microsoft.Practices.EnterpriseLibrary.WindowsAzure.TransientFaultHandling.SqlAzure.ReliableSqlConnection`, как показано в следующем примере кода:

```

public static void ReadFromDBWithReliableConnection()
{

    // Define retry Strategy and Policy

    var retryStrategy = new Incremental(5, TimeSpan.FromSeconds(1),
TimeSpan.FromSeconds(2));

    var retryPolicy = new
RetryPolicy<SqlAzureTransientErrorDetectionStrategy>(retryStrategy);

    // Receive notifications about retries.

    retryPolicy.Retrying += new
EventHandler<RetryingEventArgs>(retryPolicy_Retrying);

    using (ReliableSqlConnection conn = new
ReliableSqlConnection(connString,retryPolicy))
    {

        try

        {

            conn.Open();

            SqlCommand selectCommand = new SqlCommand(@"SELECT SOH.SalesOrderID
                                                    FROM SalesLT.SalesOrderHeader SOH
                                                    JOIN SalesLT.SalesOrderDetail SOD ON
SOH.SalesOrderID = SOD.SalesOrderID
                                                    JOIN SalesLT.Product P ON SOD.ProductID =
P.ProductID
                                                    JOIN SalesLT.Customer C ON SOH.CustomerID =
C.CustomerID
                                                    JOIN SalesLT.CustomerAddress CA on C.CustomerID =
CA.CustomerID
                                                    JOIN SalesLT.Address A on CA.AddressID =
A.AddressID
                                                    WHERE A.City=@City");

```

```

        selectCommand.Parameters.Add(new SqlParameter("@City",
SqlDbType.VarChar, 20, ParameterDirection.Input, false, 0, 0, "", DataRowVersion.Current,
"London"));

        selectCommand.CommandType = CommandType.Text;

        IDataReader dataReader =
conn.ExecuteCommand<IDataReader>(selectCommand);

        while (dataReader.Read())
        {
            Console.WriteLine("OrderID: {0}", dataReader["SalesOrderID"]);
        }
    }
    catch (Exception e)
    {
        Console.WriteLine("Exception: {0}", e.Message);
    }
}
}

```

В предыдущем примере кода при добавлении соответствующей логики повторных попыток существующий код `SqlConnection` также заменялся на код `ReliableSqlConnection`. Чтобы свести к минимуму объем переписываемого кода, есть альтернативный подход — использование методов расширения, поставляемых с программным блоком обработки временных сбоев. Такой подход не только уменьшает количество необходимых изменений в коде, но также предлагает универсальный способ для добавления возможностей повторных попыток в приложение ADO.Net. Чтобы использовать методы расширения, замените метод `Open()` и различные методы `Execute` (например, `ExecuteScalar()`, `ExecuteReader()` или `ExecuteNonQuery()`) на их эквивалентные варианты, поддерживающие повторные попытки, например `OpenWithRetry()` или `ExecuteScalarWithRetry()`. Это показано в следующем примере кода:

```

public static void ReadFromDBWithExecute()
{

```

```

        // Define retry Strategy and Policy

        var retryStrategy = new Incremental(5, TimeSpan.FromSeconds(1),
TimeSpan.FromSeconds(2));

        var retryPolicy = new
RetryPolicy<SqlAzureTransientErrorDetectionStrategy>(retryStrategy);


        // Receive notifications about retries.

        retryPolicy.Retrying += new
EventHandler<RetryingEventArgs>(retryPolicy_Retrying);


        try
        {
            retryPolicy.ExecuteAction(
                () =>
                {
                    using (SqlConnection conn = new SqlConnection(connString))
                    {
                        conn.OpenWithRetry();

                        SqlCommand selectCommand = new SqlCommand(@"SELECT
SOH.SalesOrderID
                                FROM SalesLT.SalesOrderHeader SOH
                                JOIN SalesLT.SalesOrderDetail SOD ON
SOH.SalesOrderID = SOD.SalesOrderID
                                JOIN SalesLT.Product P ON SOD.ProductID =
P.ProductID
                                JOIN SalesLT.Customer C ON SOH.CustomerID
= C.CustomerID
                                JOIN SalesLT.CustomerAddress CA on
C.CustomerID = CA.CustomerID
                                JOIN SalesLT.Address A on CA.AddressID =
A.AddressID
                                WHERE A.City=@City", conn);

```



```

        selectCommand.Parameters.Add(new SqlParameter("@City",
SqlDbType.VarChar, 20, ParameterDirection.Input, false, 0, 0, "", DataRowVersion.Current,
"London"));

        selectCommand.CommandType = CommandType.Text;

        // Execute the above query using a retry-aware ExecuteCommand
method which will

        // automatically retry if the query has failed (or connection
was dropped)

        IDataReader dataReader =
selectCommand.ExecuteReaderWithRetry(retryPolicy);

        while (dataReader.Read())
        {
            Console.WriteLine("OrderID: {0}",
dataReader["SalesOrderID"]);
        }
    }

    });
}

catch (Exception e)
{
    Console.WriteLine("Exception: {0}", e.Message );
}
}
}

```

Программный блок обработки временных ошибок поддерживает объявление настраиваемых политик повторных попыток. Дополнительные сведения об объявлении политик повторных попыток см. в разделе [Задание стратегий повтора в конфигурации](#).

Примеры кода, приведенные здесь, предлагают краткий обзор использования программного блока обработки временных ошибок. Более подробную информацию о том, как использовать эту библиотеку приложений, см. в учебнике на вики-сайте TechNet:

[Логика повторных попыток для временных сбоев в базе данных SQL Windows Azure](#)

# Сетевая задержка в База данных SQL Windows Azure

Помимо ошибок соединения, сетевые задержки являются наиболее часто встречающимися проблемами производительности у текущих пользователей База данных SQL Windows Azure.

В то время как влияние интернет-задержки хорошо известно, люди склонны недооценивать влияние задержки между приложением и База данных SQL Windows Azure. Даже если приложение и базы данных содержатся в одном центре обработки данных, задержка обычно больше, чем в традиционной локальной среде. Это более значительная задержка вызывается мультитенантной природой Azure. Кроме того, эта более значительная задержка усиливает влияние поведения приложений, характеризующегося большим количеством запросов, поскольку при каждом вызове к базе данных будет наблюдаться дополнительная задержка, что в совокупности может привести к снижению общей производительности. Следовательно, приложения с большим количеством запросов будут тратить значительно больше времени при доступе к базе данных, размещенной в База данных SQL Windows Azure, чем к базе данных SQL Server, размещенной локально.

Помимо задержки между приложением и База данных SQL Windows Azure, также существует увеличенная задержка при обмене данными между различными распределенными компонентами решения. Этот тип задержки является, пожалуй, одним из самых больших различий между локальными и облачными приложениями. Этот тип задержки возникает при обмене данными между пользователем и приложением, а также между приложением и База данных SQL Windows Azure.

С точки зрения конечного пользователя, все эти причины сетевой задержки соответствуют следующему времени ответа для пользователя:

$$\text{Response Time} = 2 \times (\text{Latency}_1 + \text{Latency}_2) + \text{Application\_Processing\_Time} + \text{Query\_Exec\_Time}$$

Где

- *Latency\_1* — это задержка между конечным пользователем и центром обработки данных, в котором размещается приложение. Этот тип задержки можно также обнаружить и в локальных средах.
- *Latency\_2* — это задержка между приложением и базами данных в База данных SQL Windows Azure.

Для обеспечения оптимизации производительности рекомендуется сначала выполнить следующие основные действия.

- Свести к минимуму задержку *Latency\_1*, выбрав центр обработки данных, расположенный ближе всего к большинству пользователей.
- Свести к минимуму задержку *Latency\_2*, разместив данные совместно с приложением Windows Azure для сведения к минимуму числа циклов приема-передачи по сети.
- Свести к минимуму показатели *Application\_Processing\_Time* и *Query\_Exec\_Time*, следуя общим рекомендациям, используемым в локальных базах данных при доступе к слою данных, управлении производительностью и кодировании для оптимизации.

## Минимизируйте расстояние между данными и приложениями

Важно размещать свои базы данных как можно ближе к месту, где выполняется приложение. Если приложение выполняется в центре обработки данных северного и центрального регионов США, то при размещении базы данных в центре обработки данных этих регионов будет обеспечено наименьшее расстояние обмена данными и, следовательно, наименьшая задержка, связанная с расстоянием.

Хотя возможен доступ к базе данных, размещенной в Windows Azure, из приложения, размещенного локально, необходимо учитывать, что это приведет к более значительным задержкам доступа к данным, а также к необходимости оплаты данных, исходящих из центра обработки данных.

В тех случаях, когда данные должны быть доступны в нескольких географических локалях, можно использовать сети доставки содержимого для распределения статических данных по нескольким сайтам. Кроме того, можно создать несколько копий базы данных в разных центрах обработки данных и использовать Синхронизация данных SQL для синхронизации данных в них. Эти подходы помогают приложениям, выполняющимся в нескольких географических локалях, находить данные, размещенные вблизи различных экземпляров приложения, и уменьшать общую задержку.



### Примечание

Система Синхронизация данных SQL в настоящее время доступна только в качестве предварительной версии и предназначена для составления отзывов для будущих версий. Использовать эту предварительную версию в рабочих средах нельзя.

## Уменьшение сетевых циклов приема-передачи

Уменьшение циклов приема-передачи важно для локальных приложений, но для Windows Azure оно имеет еще большую важность. Запросы, обрабатываемые в База данных SQL Windows Azure, должны проходить через слои сетевой балансировки нагрузки, а также через шлюз протокола потока табличных данных до приема в База данных SQL Windows Azure. Хотя эта абстракция позволяет База данных SQL Windows Azure обеспечивать

масштабирование и доступность для конечных пользователей, она также требует определенного объема обработки, что вызывает небольшую задержку при каждом цикле приема-передачи. В приложениях, отправляющих данные в База данных SQL Windows Azure, через несколько циклов приема-передачи может ощущаться снижение производительности.

Чтобы свести к минимуму эффект от циклов приема-передачи, необходимо следовать рекомендациям, которые используются для локальных приложений.

- **Используйте хранимые процедуры, особенно для инкапсуляции сложной логики доступа к данным и поведения транзакций** — при выполнении последовательных вызовов, когда второй вызов зависит от данных, возвращенных в первом, использование хранимой процедуры для выполнения логики совершения второго вызова устраняет один или несколько циклов приема-передачи и повышает производительность. Этот подход снижает число циклов приема-передачи, интенсивность использования и блокировок ресурсов на стороне сервера.
- **Уменьшайте использование курсоров или доступа по строкам** — используйте операции на основе наборов, где это возможно. Если курсоры использовать необходимо, используйте курсор на стороне клиента.
- **Используйте возвращаемые табличное значение параметры для отправки нескольких строк в База данных SQL Windows Azure в каждом цикле приема-передачи**—База данных SQL Windows Azure использует табличные параметры так же, как и локальные версии компонента Database Engine SQL Server. Возвращающие табличное значение параметры помогают уменьшить количество вызовов к одной и той же хранимой процедуре для обработки набора записей или значений. Можно также передать табличные параметры единому параметризованному запросу, такому как команда SELECT, INSERT, UPDATE или DELETE.  
Дополнительные сведения см. в разделе документации в сети [Использование возвращающих табличное значение параметров](#).
- **Используйте локальное кэширование, где это возможно** — локальное кэширование позволяет повторно использовать результаты без выполнения нескольких циклов приема-передачи База данных SQL Windows Azure. Кроме того, следует учитывать, что можно кэшировать вызовы хранимых процедур, возвращающих значение того же типа, что и тип переданного значения.
- **Используйте кэширование Windows Azure, где это возможно** — используйте кэширование Windows Azure для данных подстановки, доступных только для чтения, чтобы свести к минимуму сетевой трафик к База данных SQL Windows Azure.  
Дополнительные сведения см. в разделе [Сокращение сетевых циклов приема-передачи с использованием кэширования Windows Azure](#).
- **Кэшируйте данные и метаданные, где это возможно.**
- **Избегайте получения метаданных во время выполнения, если это возможно.**

- **Избегайте использования таких классов, как SqlCommandBuilder** — эти классы запрашивают метаданные во время выполнения, что приводит к дополнительным циклам приема-передачи.
- **Объединяйте инструкции SQL в пакеты, если это возможно** — вы можете объединять несколько инструкций Transact-SQL в одну пакетную команду для извлечения нескольких результирующих наборов или для выполнения нескольких операций DML в ходе одного сетевого цикла приема-передачи. Этот подход особенно полезен при работе с большим количеством последовательных операций INSERT.
- **Избегайте управления транзакциями на основе приложений** — инкапсуляция операций управления транзакциями — BEGIN TRAN, COMMIT/ROLLBACK — в хранимые процедуры может снизить количество сетевых циклов приема-передачи и блокировок.

## Соблюдайте общие рекомендации для локальных баз данных

После уменьшения расстояния между данными и пользователями и уменьшения количества сетевых циклов приема-передачи следующий шаг необходим, чтобы убедиться, что приложение следует общим рекомендациям для локальных баз данных. Если применять известные рекомендации к уровню доступа к данным приложения, а также рекомендации по увеличению производительности и оптимизации, должен быть замечен эффект кратного преимущества в среде с высокой задержкой, такой как облако.

Для взаимодействия с локальными базами данных применимы следующие общие рекомендации.

- **Открывайте соединения как можно позднее и закрывайте его как можно раньше** — чтобы свести к минимуму использование ресурсов и возможность регулирования, устанавливайте соединение в приложении только в тот момент, когда это необходимо в коде. Кроме того, соединение следует возвращать в пул сразу же после того, как код завершит с ним работу, удаляя объект соединения. Исключением из этого правила является политика возврата в случае, когда есть более неотложная работа, которую нужно завершить. В данном случае возвращайте соединение только тогда, когда вся непосредственная работа завершена.
- **Используйте включение соединений в пул** — на уровне процессов пулы соединений будут содержать соединения, целью которых является одна база данных и которые используют один контекст безопасности. По возможности используйте одну строку подключения для всех соединений, имеющих одинаковые характеристики.
- **Получайте только те нужны данные** — тщательно определяйте список SELECT и предложение WHERE. Такой подход позволяет свести к минимуму использование пропускной способности сети, а также позволяет эффективно индексировать базу данных для увеличения производительности.
- **Делайте транзакции как можно более короткими и избегайте вовлечения ненужных ресурсов** — неоптимально построенные модели данных и излишнее число сетевых

циклов приема-передачи между кодом приложения и базами данных — это часто встречающиеся проблемы, но в локальных развертываниях они могут не иметь критической важности из-за небольшого объема задержки в среде подключения. Эти приложения также может быть трудно изменить по следующим причинам.

- Существующие архитектурные ограничения.
- Монолитные конструкции, в которых доступ к данным и бизнес-логика тесно связаны и взаимозависимы.
- Некоторые аспекты, такие как построчная обработка данных, которые являются необходимыми для логики приложения и которые трудно изменить без больших изменений кода.

Помимо этих общих рекомендаций существуют дополнительные рекомендации, которые касаются технологий, которые используются в приложении. Например, приложения, которые используют платформу .NET Framework, имеют возможность использовать технологии, такие как ADO.NET, Entity Framework и службы WCF Data Services. Эти технологии предлагают сокращение времени разработки и предоставляют большую гибкость в реализации уровня доступа к данным для любого стиля приложений. Эти технологии также хорошо сочетаются с ориентацией на службы путем разделения слоев, предложения гибкости схемы, использования открытых архитектур данных (например, OData) и предложения по своей природе «отключенной» конструкции. Тем не менее, несмотря на их преимущества, эти технологии могут формировать те же проблемы, которые были описаны для прежних версий приложений, например большое число вызовов, если не принимаются во внимание схема данных и оптимизация операций (включая оптимальное представление данных, использование кэша и объединение пакетных данных и операции над ними для сокращения числа циклов приема-передачи).

В следующих разделах описываются некоторые рекомендации для этих технологий, а также применение асинхронного программирования к приложениям Azure.

## **Рекомендации для ODBC и JDBC**

Для библиотек доступа к данным, таких как ODBC и JDBC, где ограничена поддержка клиентских интеллектуальных операций (например, сортировка, фильтрация и т. д.), можно применить существующие методы оптимизации, такие как сокращение операций с курсором и построчной обработки. Когда необходимо использовать курсоры, используйте исключительно курсоры только для чтения для получения значений и используйте команды SQL для изменения данных, а не позиционированные обновления в курсорах.

## Рекомендации для ADO.NET

ADO.NET обеспечивает несколько методов оптимизации, которые можно применить в целях обеспечения максимальной эффективности вашего кода доступа к данным.

- **Выберите соответствующий режим выполнения с SqlCommand** — например, если нужно получить только скалярное значение, используйте метод `ExecuteScalar()` или `ExecuteNonQuery()`, если не нужно извлекать результирующий набор
- **Используйте свойство `UpdateBatchSize` при использовании класса `SqlDataAdapter` для выполнения нескольких операций** — при этом несколько команд включаются в пакет при их передаче по сети, сводя таким образом к минимуму число сетевых циклов приема-передачи.
- **Используйте `SqlDataAdapter` для реализации прямых взаимодействий с базой данных с помощью хранимых процедур для операций `SELECT`, `INSERT`, `UPDATE` и `DELETE`.**
- **Задайте для свойства `SerializationFormat` значение `Binary` при использовании `Datasets` в качестве кэша на стороне клиента** — это уменьшает объем данных, передаваемых по сети.

## Рекомендации для Entity Framework

Когда речь идет о разработке доступа к данным, очень часто используется платформа Entity Framework по следующим причинам.

- Она обеспечивает мощную среду объектно реляционного сопоставления, которое может значительно сократить время разработки.
- Она предоставляет большую гибкость посредством разделения определения схемы физической базы данных и концептуального представления данных приложения.
- Предоставляет полный набор служб для обработки сохраняемости.
- Уменьшает число циклов приема-передачи, позволяя загружать результирующие наборы из База данных SQL Windows Azure в наборы объектов в памяти приложения и повторно использовать их в отключенном режиме без необходимости взаимодействовать с сервером в ходе каждой операции.

Однако, как и в случае с любым другим инструментом программирования, Entity Framework может создавать некоторые проблемы с производительностью в случае, если не уделяется должное внимание взаимодействию с базами данных. Кроме того, среда Windows Azure усиливает эти проблемы производительности.

Чтобы оптимизировать совместное использование Entity Framework с База данных SQL Windows Azure, следуйте этим рекомендациям.

- **Явно отключите отслеживание состояния объекта на уровне ObjectStateManager для сущностей и связей** — примените эту рекомендацию, если приложению не требуется отслеживать состояние объекта ввиду того, что обычно оно использует доступ только для чтения.
- **Отключите отложенную загрузку, чтобы лучше контролировать взаимодействие между кодом и базой данных** — в целях сокращения числа циклов приема-передачи следует предварительно загрузить все нужные ресурсы в ходе одного взаимодействия. Включение нескольких команд в пакет для загрузки и сохранения данных в хранилище данных можно применить, расширив контекст объекта с помощью определенных методов.
- **Используйте хранимые процедуры для взаимодействия с хранилищем данных, где это возможно** — там, где нельзя использовать хранимые процедуры, используйте параметризованные запросы и индексированные представления для повышения производительности.
- **Выполняйте нескольких операций с базой данных в ходе одного цикла приема-передачи путем интеграции табличных параметров и сопоставления наборов объектов с возвращающими табличные значения параметрами хранимых процедур** — этот подход можно использовать, даже если он не поддерживается напрямую средствами проектирования.
- **Сопоставьте несколько результирующих наборов с объектами в ходе одной операции, где это возможно.**
- **Сопоставьте наборы объектов с методом `System.Data.SqlClient.SqlBulkCopy.WriteToServer()` для выполнения действий массовой вставки данных** — метод `SqlBulkCopy` позволяет компоненту Database Engine SQL Server и База данных SQL Windows Azure использовать оптимизацию протоколирования массовых операций, что позволяет во многих случаях быстрее выполнять объемные вставки.

Дополнительные сведения см. в разделе [Использование Entity Framework для сокращения задержек в сети для базы данных SQL Windows Azure](#).

## Рекомендации по асинхронному программированию

Некоторые операции баз данных, например выполнение команд, могут занимать значительное количество времени. В таких случаях однопоточные приложения должны блокировать другие операции и ожидать завершения команды перед возобновлением других операций. В противоположность этому при назначении длительной операции в фоновый поток основной поток может оставаться активным в течение всей операции.

ADO.NET поддерживает эти же конструкторские шаблоны в своем классе `SqlCommand`. В частности, параллельное выполнение методов `BeginExecuteNonQuery()`, `BeginExecuteReader()` и `BeginExecuteXmlReader()` с методами `EndExecuteNonQuery()`,



EndExecuteReader() и EndExecuteXmlReader() соответственно обеспечивает асинхронную поддержку.

Эти асинхронные методы не отменяют необходимости применения всех оптимизаций и рекомендаций, упомянутых в предыдущих разделах. Однако, позволяя другим операциям выполняться параллельно с запросом в База данных SQL Windows Azure, эти асинхронные методы помогают снизить влияние длительных операций базы данных на вашем уровне доступа к данным.

## Вопросы высокого уровня доступности и аварийного восстановления с помощью базы данных SQL Windows Azure

---

**Авторы:** Кун Чен (Kun Cheng), Сельцин Туркаслан (Selcin Turkarslan)

**Рецензенты:** Стив Ховард (Steve Howard), Адриан Бетун (Adrian Bethune)

При миграции локальной базы данных SQL Server в базу данных SQL Windows Azure (базу данных SQL) часто задают вопрос, как реализовать стратегию резервного копирования и восстановления для защиты данных от ошибок пользователей, сбоя оборудования, ошибок приложения, завершения работы центра обработки данных из-за природных катаклизмов и других аварий базы данных. В отличие от локального развертывания база данных SQL предназначена для маскирования управления файлами базы данных и операциями от администраторов. Обратите внимание на то, что сервер базы данных SQL — это логический сервер, который определяет группу баз данных. Базы данных, связанные с сервером базы данных SQL, могут размещаться на разных физических компьютерах в центре обработки данных Microsoft. Отдельная логическая база данных может совместно использовать место в одной физической базе данных с другой логической базой данных. В мультитенантной среде Windows Azure традиционные средства резервного копирования и восстановления SQL Server не работают.

## Как защитить базы данных от сбоя отдельных серверов, устройств или потери связи с сетью

Каждый экземпляр базы данных SQL имеет три реплики на трех разных физических компьютерах в центре обработки данных, одну первичную и две вторичных. Все операции чтения и записи проходят через первичную реплику, и все изменения асинхронно реплицируются во вторичные реплики.

База данных SQL использует схему фиксации на основе кворума, в которой данные сначала записываются на первичную и одну вторичную реплику, только после этого транзакция считается зафиксированной. При отказе оборудования первичной реплики подключение структура базы данных SQL обнаруживает сбой и переходит на вторичную реплику. Таким образом, в центре обработки данных есть как минимум две физические, согласованные на уровне транзакций копии данных. Три реплики для каждого экземпляра базы данных SQL Windows Azure защищают данные от сбоя отдельных серверов, устройств или сетевого соединения. Помимо избыточных реплик структура базы данных SQL Windows Azure создает внутренние копии данных минимум за последние 14 дней, взятые с интервалами в 5 минут для всех баз данных, имеющихся в центре обработки данных. Эти копии хранятся в центре обработки данных и обеспечивают защиту от одновременных или катастрофических аппаратных или системных сбоев.

Среда базы данных SQL предназначена для поддержания доступности сервера вместе с обеспечением целостности данных в случае аппаратного сбоя. Во время отработки отказа экземпляр базы данных SQL может быть недоступен на короткий момент. На случай отработки отказа приложение должно иметь логику выполнения повторных операций. После сбоя приложение может использовать ту же строку подключения, чтобы повторно установить подключение к вторичной реплике после отработки отказа. Дополнительные сведения о работе с ошибками потери соединения см. в статье [Управление подключениями к базе данных SQL Windows Azure](#) на веб-сайте TechNet Wiki.

## Как защитить свою базу данных от нежелательного удаления или изменений

Ошибка пользователя или приложения является самой частой причиной потери данных или повреждения сценариев во многих приложениях программного обеспечения.

Пользователь может удалить таблицу или приложение по ошибке или отправить транзакцию дважды. За подобными типами ошибок тяжело следить. Следующие инструменты помогают справиться с подобными проблемами:

- Копирование базы данных
- Службы экспорта и импорта базы данных SQL
- Службы SQL Server Integration Services и программа bcp

С помощью **копии базы данных** можно создать копию базы данных на том же или на другом сервере в этом же центре обработки данных. Это асинхронная и согласованная на

уровне транзакций операция в сети. Поскольку операция является асинхронной, вы можете выдать команду `copy` и контролировать ее выполнение, запрашивая системное представление **`sys.dm_database_copies (SQL Database)`**.

Для того чтобы скопировать экземпляр базы данных SQL, ваше имя входа должно являться членом роли `dbmanager` уровня сервера на целевом сервере и быть владельцем базы данных-источника на исходном сервере. Вход должен производиться с одинаковыми именами входа и паролями на обоих серверах базы данных SQL, как на исходном, так и на сервере-назначении. Частота копирования базы данных зависит от потребностей вашего бизнеса. Для восстановления после ошибки пользователя или приложения рекомендуется создавать копии ежедневно и иметь две или три работающие копии, удаляя самую старую копию каждый день после создания свежей копии.

Несмотря на то что мы рекомендуем выполнять копирование ежедневно, копирование базы данных может выполняться еще чаще. Рекомендуется выполнять резервную копию базы данных не чаще чем раз в час. Каждый процесс копирования базы данных, выполняемый независимо от всех других процессов копирования, создаст копию базы данных, согласованную на уровне транзакций в конце процесса копирования. Каждая копия учитывается в максимум 150 базах данных для каждого сервера базы данных SQL, и плата за нее взимается как за отдельную базу данных. Потому что слишком частое копирование может привести к тому, что в вашей учетной записи исчерпается предел доступных баз данных, а средства будут тратиться на оплату практически одинаковых копий базы данных. Дополнительные сведения см. в разделе **Copying Databases in SQL Database** в библиотеке MSDN для библиотеки базы данных SQL.

Помимо средства копирования баз данных можно использовать службу **SQL Database Import/Export Service**. С помощью данной службы можно импортировать или экспортировать данные и схему в пакете с расширением **.bacpac**. Пакет содержит в сжатом формате все совместимые с базой данных SQL объекты, например таблицы, представления, индексы, ограничения, триггеры, хранимые процедуры, имена входа, пользователи и т. д. Служба может напрямую импортировать или экспортировать файлы BACPAC из экземпляра базы данных SQL в хранилище больших двоичных объектов Windows Azure и в обратном направлении. Доступ к службе импорта-экспорта можно получить на портале управления Windows Azure. Если нужно импортировать или экспортировать данные непосредственно с локального сервера SQL в базу данных SQL и в обратном направлении без использования хранилища больших двоичных объектов Windows Azure, воспользуйтесь классами пространства имен [Microsoft.SqlServer.Dac](#). Аналогично можно использовать `DacIESvcCli.exe` в [примерах SQL DAC](#) на сайте CodePlex.

В отличие от средства копирования баз данных служба экспорта и импорта не формирует резервные копии, согласованные на уровне транзакций. Для выполнения резервной копии рекомендуется заблокировать базу данных и остановить транзакции перед экспортированием данных и схемы.

Средства **Bulk copy utility (BCP.exe)**, **SQL Server Integration Services (SSIS)** и **System.Data.SqlClient.SqlBulkCopy** также схожи со службой **Import/Export Service**. В данный момент база данных SQL поддерживает программу BCP, массовое копирование, API-интерфейс и службы SSIS для перемещения данных. Необходимо создать объекты схемы в базе данных SQL перед загрузкой данных. Использование BCP или служб SSIS в качестве механизма массового копирования позволяет контролировать, какие объекты перемещаются в базе данных и какие данные перемещаются из этих объектов. Также можно указывать различные параметры, например размер пакета и количество потоков для получения лучшей пропускной способности, в зависимости от полосы пропускания и задержки сети.

## Защита базы данных от масштабных разрушений в центре обработки данных

Для защиты от потери данных в случае катастрофы в центре обработки данных следует создать хранилище резервных копий базы данных вне пределов того центра обработки данных, на котором размещена база данных приложения. Для этого рекомендуется использовать копирование базы данных, описанное в предыдущем разделе, и службу экспорта и импорта базы данных SQL.

Рекомендуется использовать следующие инструменты для управления общей стратегией резервного копирования и восстановления.

Реализуйте стратегию копирования и восстановления для обработки ошибок приложений и пользователей с помощью следующих средств:

- Копирование базы данных

- Службы экспорта и импорта базы данных SQL

- Службы SQL Server Integration Services и программа Bcp

Реализуйте усовершенствованную стратегию резервного копирования и восстановления для обработки масштабных разрушений в центре обработки данных с помощью следующих средств:

Служба импорта-экспорта для миграции копий базы данных в один или несколько вторичных центров обработки данных и дополнительно в собственный локальный SQL-сервер.

Дополнительные сведения о возможностях резервного копирования, восстановления и аварийного восстановления в Windows Azure см. в статьях **Business Continuity in Windows Azure SQL Database** и [Непрерывность бизнес-процессов для Windows Azure](#) библиотеки MSDN.

## Выбор средств для миграции базы данных в базу данных SQL Windows Azure

---

**Автор:** Шон Тинлайн-Джонс (Shaun Tinline-Jones)

**Соавтор:** Стив Ховард (Steve Howard)

**Рецензент:** Шон Хернан (Shawn Hernan)

В этом разделе рассматривается выбор наиболее подходящего набора средств для миграции базы данных проекта в База данных SQL Windows Azure. В целом миграция базы данных включает в себя передачу схемы и данных. Некоторые средства миграции обрабатывают как схему, так и данные. Другие могут обрабатывать или схему, или данные.

### Выбор средств миграции

Существует несколько процессов и средств, которые можно использовать для миграции базы данных в База данных SQL Windows Azure. Выбор средства зависит от типа, размера и сложности подлежащей переносу базы данных.

### Средства для миграции базы данных, отличной от SQL Server

Помощник по миграции SQL Server используется для миграции базы данных из других продуктов, таких как Access, MySQL, Oracle или Sybase, в База данных SQL Windows Azure. Дополнительные сведения см. в разделе [Как использовать помощник миграции SQL Server с базой данных SQL Windows Azure](#).

Приложение от Майкрософт под кодовым названием Data Transfer может передавать данные файла Excel или CSV-файла в База данных SQL Windows Azure. Дополнительные сведения см. в разделе [Лаборатории по базе данных SQL Windows Azure](#).

## Средства миграции между службами базы данных SQL

Для переноса данных из одной базы данных в База данных SQL Windows Azure в другую воспользуйтесь *копией базы данных SQL* и *службой синхронизации данных SQL*.

База данных SQL Windows Azure поддерживает функцию копирования базы данных. Эта функция создает базу данных в База данных SQL Windows Azure, которая является согласованной на уровне транзакций копией существующей базы данных. Для копирования базы данных необходимо подключение к базе данных master службы База данных SQL Windows Azure, где будет создана новая база данных, и команда CREATE DATABASE:

```
CREATE DATABASE destination_database_name AS COPY OF  
[source_server_name.]source_database_name
```

Новая база данных может быть на той же или другой службе. Пользователь, выполняющий эту инструкцию, должно иметь роль *dbmanager* на целевой службе (для создания новой базы данных) и роль *dbowner* в базе данных-источнике. Дополнительные сведения см. в разделе [Копирование баз данных в базу данных SQL Windows Azure](#).

Синхронизация данных SQL обеспечивает создание и планирование регулярной синхронизации между База данных SQL Windows Azure и базами данных, размещенными либо в SQL Server, либо в База данных SQL Windows Azure. Дополнительные сведения см. в разделе [Синхронизация данных SQL](#).

## Средства для миграции базы данных SQL Server

Проекты миграции SQL Server на База данных SQL Windows Azure можно условно разделить по размеру и сложности.

1. **Размер:** количество данных и количество объектов схемы, которые необходимо передать. Чем больше данных, тем больше времени потребуется для передачи базы данных в База данных SQL Windows Azure и тем более вероятно, что База данных SQL Windows Azure будет регулировать процесс миграции. При крупной миграции важно выбрать один процесс миграции, который может выполнить оптимизацию, например запуск параллельных операций загрузки данных либо разделение операций на более мелкие пакеты, которые, скорее всего, не будут регулироваться. Средство миграции, которое может автоматически выполнять повторную операцию регулирования, является более важным для крупной миграции. Обычно база данных становится большой, если вмещает примерно 200 МБ данных. База данных с очень большим числом объектов, например 1000 или более, также считается большой. Если процесс

миграции выполняет передачу схемы в одной транзакции, передача, скорее всего, будет регулироваться из-за большого объема пространства журнала, которое используется транзакцией.

2. Сложность: объем инженерных изменений в базе данных и связанных приложениях. Большое число объектов со сложными структурами увеличивает вероятность, что база данных содержит элементы синтаксиса, которые не поддерживаются в База данных SQL Windows Azure, в результате могут потребоваться дополнительные усилия разработчиков в рамках проекта миграции. Простой проект миграции включает базу данных, которая не требует изменения схемы для выполнения в База данных SQL Windows Azure, а также приложения, которым требуется только изменение строки подключения. Сложный проект миграции может включать базу данных, которой требуются изменения схемы для обработки элементов, неподдерживаемых в База данных SQL Windows Azure, или приложения, которым необходимы изменения для эффективной работы с удаленной базой данных.

Базы данных могут быть классифицированы на четыре категории, или квадранта.

### Размер проекта и квадранты сложности



Проект миграции, который первоначально оценивается как небольшой и простой, может перейти в другой квадрант по мере поступления новой информации в ходе дальнейшего исследования. Ниже приводится ряд причин.

- Для миграции относительно небольшой базы данных при ограниченных временных параметрах может потребоваться оптимизация передачи данных.

- Базе данных требуются изменения схемы, которые могут повлиять на код в приложениях, использующих базу данных.
- Приложениям требуются изменения для эффективной работы в среде База данных SQL, включая надежную логику повторения операций или изменения кода для снижения задержки в сети.

Наиболее распространенная причина классификации миграции базы данных как сложной — это когда необходимы изменения в схеме базы данных, приложениях или и там и там. Например, для изменений схемы обычно требуются изменения приложений, использующих базу данных. Такие изменения потребуют усилия по разработке в рамках проекта, а также координацию развертывания новой базы данных и новых версий приложений, использующих базу данных. Для этих проектов выберите средство миграции, которое поддерживает разработку проекта. Дополнительные сведения о типах изменений схемы и приложений, которые могут потребоваться, см. в разделе [Управление проектом миграции базы данных SQL Windows Azure](#).

Миграция базы данных в База данных SQL Windows Azure включает перенос как схемы, так и данных. Некоторые средства миграции могут использоваться отдельно (например, мастер миграции и файлы BACPAC (DAC) приложения уровня данных База данных SQL), поскольку они передают и схему и данные. Другие средства могут передавать только схему, например пакеты приложения уровня данных, или только данные, например приложение bcp. Если миграция предполагает использование средства, которое передает только данные или схему, лучше сначала использовать приложение bcp для передачи данных большого проекта, затем использовать это приложение вместе с каким-либо другим для передачи других частей базы данных.

Хотя такие средства, как мастер миграции и средства SQL Server Data Tools (SSDT) База данных SQL, способны обнаружить большинство объектов и распознают синтаксис, неподдерживаемый в База данных SQL Windows Azure, текущие версии этих средств не решают всех проблем. Средства подходят для изначального анализа и выявления большинства проблем в ходе разработки базы данных. Наиболее надежный способ определить, все ли проблемы схемы были разрешены, — это выполнить развертывание тестов базы данных в База данных SQL Windows Azure. Наиболее надежный способ определить, все ли проблемы синтаксиса Transact-SQL были разрешены в коде приложения, — это выполнение функционального теста приложения, запущенного для копии базы данных в тестовой системе База данных SQL Windows Azure.

Для сложных проектов миграции, которым требуются изменения схемы, характерно совмещение нескольких задач и использование нескольких средств, таких как следующие.

- Извлечение схемы в проект базы данных в SSDT.



- Задание База данных SQL Windows Azure в качестве цели проекта для проведения первого анализа объектов, не поддерживаемых База данных SQL Windows Azure. Можно оставить в качестве набора целей База данных SQL Windows Azure для того, чтобы во время последующей разработки базы данных видеть ошибки синтаксиса, неподдерживаемого в База данных SQL Windows Azure.
- Запуск задачи разработки базы данных для внесения всех необходимых изменений схемы. После того как средства SSDT сообщат, что неподдерживаемых объектов не осталось, необходимо выполнить развертывание тестов База данных SQL Windows Azure, чтобы убедиться, что все оставшиеся объекты поддерживаются База данных SQL Windows Azure.
- Запустите задачу параллельной разработки приложения для выполнения всех изменений кода, которые становятся необходимыми в результате изменений схемы. Формирование трассировок инструкций Transact-SQL приложений и использование База данных SQL Мастера миграции для проверки синтаксиса, который не поддерживается в База данных SQL Windows Azure.
- Сборка процесса передачи данных, который выполнит преобразования, необходимые для получения данных из старых структур схемы в новую схему. Эту задачу можно легко реализовать с помощью служб SQL Server Integration Services.
- Выполнение общего тестирования базы данных и приложения. Необходимо выполнить полное функциональное тестирование базы данных после ее запуска на База данных SQL Windows Azure, чтобы удостовериться, что инструкции Transact-SQL, сформированные приложениями, работают в База данных SQL Windows Azure.
- Сборка пакетов развертывания для схемы и приложений базы данных. Сборка скриптов, необходимых для запуска процесса передачи данных в рабочих системах.
- Выполнение общего рабочего развертывания схемы базы данных, приложений и запуск процесса передачи данных.

Дополнительные сведения о запуске проекта База данных SQL Windows Azure см. в разделе [Управление проектом миграции базы данных SQL Windows Azure](#).

Мастеру миграции База данных SQL требуется наименьшая перегрузка для запуска; его можно настроить для выявления ошибок, когда средство не смогло определить проблему, или изменить так, чтобы средство больше не рассматривало что-либо как проблему. Средство можно применять в любом квадранте. Средство, возможно, является лучшим на этапе планирования проекта миграции. Особенностью этого средства, не обеспечиваемой другими средствами, является возможность анализировать файлы трассировки приложения SQL Profiler. Благодаря этому приложение может работать с динамическими инструкциями Transact-SQL. Анализ трассировки также может повысить качество непрерывного функционального теста. Мастер может разбить передачу данных и схемы на несколько операций и повторять операции, завершившиеся ошибкой. Эти возможности

улучшают способность выполнять миграцию больших баз данных. Существует также функция, обеспечивающая простую миграцию федераций.

Средства SQL Server Data Tools (SSDT) обеспечивают полезные практические функции во всех квадрантах. Тесная интеграция с Visual Studio делает средства SSDT особенно полезными при миграции, которая считается сложной. Средства SSDT имеют ограниченную поддержку разработки решений База данных SQL Windows Azure, которые используют федерации. Анализ происходит, когда объекты импортируются в средство, а предупреждения и ошибки возникают во время построения. Средства SSDT обладают мощными функциями и могут выполнять сложные решения. Средства SSDT рекомендуются корпорацией Майкрософт для работы с База данных SQL Windows Azure, а также с локальными базами данных SQL Server.

## Сравнение средств миграции SQL Server

В следующей таблице перечислены характеристики средств и процессов, которые могут быть использованы для миграции базы данных SQL Server в База данных SQL Windows Azure.

Средства	Схема	Проверка совместимости с База данных SQL	Данные	Эффективность передачи данных	Примечание
<a href="#">Мастер миграции базы данных SQL Windows Azure</a>	Да	Да	Да	Хорошая	<ul style="list-style-type: none"><li>• Большие возможности, в том числе оценка файлов трассировки</li><li>• Открытый источник на CodePlex</li><li>• Не поддерживается корпорацией Майкрософт</li></ul>
<a href="#">Средства SQL Server Data</a>	Да	Да	Нет	н/д	<ul style="list-style-type: none"><li>• Подходит для управления работой по разработке</li></ul>

<a href="#">Tools</a>					<p>миграции</p> <ul style="list-style-type: none"> <li>• Обрабатывает сложные изменения схемы</li> <li>• Полная поддержка База данных SQL</li> </ul>
<a href="#">Пакет приложения уровня данных</a>	Да	Да	Нет	н/д	<ul style="list-style-type: none"> <li>• Сущность, содержащая все объекты базы данных, но не данные</li> <li>• Полная поддержка База данных SQL</li> </ul>
<a href="#">Экспорт и импорт DAC ВАСРАС</a>	Да	Да	Да	Хорошая	<ul style="list-style-type: none"> <li>• Экспорт/импорт DAC, а также данные и платформа DAC</li> <li>• Доступна служба поддержки только облака</li> <li>• Примеры SQL DAC доступны на CodePlex</li> </ul>
<a href="#">Мастер формирования скриптов</a>	Да	Частично	Да	Низкая	<ul style="list-style-type: none"> <li>• Имеет явные варианты для создания скриптов База данных SQL</li> <li>• Подходит для небольших баз</li> </ul>

					данных
<a href="#">bcp</a>	Нет	н/д	Да	Хорошая	<ul style="list-style-type: none"> <li>Эффективная передача данных в существующую таблицу</li> <li>Каждая команда bcp передает одну базу данных</li> </ul>
<a href="#">SQL Server Integration Services</a>	Нет	н/д	Да	Хорошая	<ul style="list-style-type: none"> <li>Максимальная гибкость</li> </ul>
<a href="#">Мастер импорта и экспорта SQL Server</a>	Нет	н/д	Да	Хорошая	<ul style="list-style-type: none"> <li>Простой пользовательский интерфейс служб SSIS; также доступен в среде SQL Server Management Studio</li> </ul>

## Как использовать мастер миграции SQL Azure

**Автор:** Шон Тинлайн-Джонс (Shaun Tinline-Jones)

Мастер миграции базы данных SQL Windows Azure — это средство с исходным пользовательским интерфейсом и открытым кодом, помогающее переносить базы данных SQL Server в База данных SQL Windows Azure. Помимо переноса данных это средство также можно использовать для выявления проблем совместимости, их устранения и вывода сведений об этих проблемах.

- Перед началом:** [Рекомендации](#), [Ограничения](#), [Предварительные условия](#)

- **Мастер миграции базы данных SQL Windows Azure можно использовать для [миграции баз данных](#), [анализа баз данных](#), [анализа файлов Transact-SQL](#), [анализа файлов трассировки](#)**

## До начала работы

Мастер миграции баз данных SQL является гибким и простым инструментом для переноса простых баз данных SQL Server в базу данных SQL. Для более сложных баз данных мастер является хорошим инструментом выявления изменений, необходимых для удовлетворения требований База данных SQL Windows Azure.

## Рекомендации

Мастер миграции баз данных SQL поддерживает следующие основные задачи:

- Перенос схемы и данных простой базы данных в База данных SQL Windows Azure. Мастер можно настроить для выполнения нескольких одновременных операций массового копирования при загрузке больших объемов данных.
- Помогает анализировать более крупные и более сложные базы данных на этапах создания или планирования.
  - Анализ базы данных для объектов, не поддерживаемых База данных SQL Windows Azure.
  - Файл Transact-SQL содержит синтаксис, не поддерживаемый База данных SQL Windows Azure. Мастер умеет анализировать как файл скриптов Transact-SQL, так и файл трассировки приложения SQL Server Profiler.

Перед запуском мастера загрузите руководство на вкладке документации в проекте Codeplex и ознакомьтесь с ним.

Мастер можно запускать в графическом режиме или режиме командной строки. Мастер использует два файла конфигурации, которые вы можете изменять, чтобы адаптировать работу мастера. Файлы конфигурации задают поведение по умолчанию при работе в графическом режиме и управляют поведением при работе в режиме командной строки.

- Файл NotSupportedByAzureFile.config содержит записи в виде регулярных выражений, которые определяют объекты, не поддерживаемые База данных SQL Windows Azure. Файл конфигурации можно изменять с целью подбора дополнительных шаблонов, которые необходимо исключить из баз данных, размещаемых на База данных SQL Windows Azure.
- Файл SQLAzureMW.exe.config управляет поведением средства, включая соединение, выбор файла Transact-SQL для анализа или поведения при копировании данных. Файл можно адаптировать под нужды конкретного сайта по умолчанию.

Мастер миграции баз данных SQL сочетает в себе три функции для поддержки более надежных и высокоскоростных операций массового копирования по сравнению с другими вариантами, например с помощью приложения уровня данных (DAC) ВАСРАС.

- Мастер миграции баз данных SQL имеет встроенную логику для обработки потери соединения. Она делит обновления схемы на отдельные пакеты, при этом каждый пакет управляется как отдельная транзакция. Мастер работает до тех пор, пока База данных SQL Windows Azure не разорвет соединение. Если мастер обнаружит ошибку связи до завершения обновления схемы, соединение с База данных SQL Windows Azure будет восстановлено и обработка возобновится с последней успешно завершенной транзакции. Таким же образом при использовании программы bcp для передачи данных в База данных SQL Windows Azure мастер разделяет данные на отдельные пакеты и использует логику повторных попыток, чтобы выявить последнюю успешную запись, загруженную до закрытия соединения. Затем посредством bcp мастер перезапускает загрузку со следующим набором записей.
- Мастер можно настроить для выполнения нескольких одновременных операций массового копирования в целях ускорения загрузки больших объемов данных. Мастер не может выполнять несколько одновременных операций массового копирования для одной таблицы, но может запланировать несколько одновременных операций массового копирования для разных таблиц.
- Вероятность прерывания работы мастера из-за взаимодействия с База данных SQL Windows Azure можно снизить, указав период задержки между операциями массового копирования пакетов и меньший размер пакета. Необходимо сбалансировать размер пакета и количество пакетов. Если размер пакета слишком мал, это может привести к увеличению количества пакетов, которые должны передаваться отдельно в сети и которые будут тем самым создавать проблему задержки. Протестируйте несколько вариантов и найдите размер пакета, который был бы достаточно мал, чтобы избежать заторов, но достаточно велик, чтобы снизить проблему задержки.

[\[В начало\]](#)

## Ограничения



### Примечание

Мастер миграции баз данных SQL — это общедоступное исходное средство, созданное и поддерживаемое сообществом.

Мастер миграции баз данных SQL не включает средство синтаксического анализа Transact-SQL, однако поддерживает поиск шаблонов на основе регулярных выражений в файле `NotSupportedByAzureFile.config`. Некоторые совпадения могут давать ложные положительные результаты. Кроме того, не гарантируется, что файл конфигурации,

поставляемый вместе с мастером, будет иметь шаблоны для всех элементов, не поддерживаемых База данных SQL Windows Azure. Для обеспечения миграции проекта обновите файл конфигурации и добавьте в него необходимые шаблоны. Общие вопросы можно направлять в проект CodePlex, чтобы они были учтены в последующих версиях мастера. Если проекту требуется более тщательный анализ базы данных, стоит обратить внимание на возможность извлечения файла пакета приложения уровня данных и импортирования его в проект SQL Server Data Tools, в котором можно задать базу данных SQL в качестве цели проекта. Средство SQL Server Data Tool анализирует проект с помощью средства синтаксического анализа Transact-SQL, однако оно может не выявить все несоответствия в База данных SQL Windows Azure.

Наиболее надежный способ определить, все ли проблемы схемы были разрешены, — это выполнить развертывание тестов новой схемы базы данных для База данных SQL Windows Azure. Наиболее надежный способ определить, все ли проблемы Transact-SQL были разрешены в коде приложения, состоит в выполнении функционального теста приложения, запущенного для копии базы данных, развернутой для База данных SQL Windows Azure.

Мастер прекрасно выполняет первоначальный анализ любой базы данных. Однако другие инструменты лучше подходят для управления разработкой сложных баз данных, в которые требуется внести много изменений, прежде чем запускать в База данных SQL Windows Azure. Например, в определениях поиска на основе регулярных выражений можно указать замены для шаблонов, обнаруженных мастером, однако эта функциональность ограничена. Для управления более сложными изменениями лучше использовать другой вариант, например извлечь файл пакета приложения уровня данных и импортировать его в проект SQL Server Data Tools.

Создание трассировки профилировщика из рабочей системы может значительно уменьшить производительность. Лучше сформировать трассировку из тестовой системы. Если необходимо профилировать рабочую систему, можно свести к минимуму снижение производительности путем трассировки только событий, связанных с завершенными инструкциями.

[\[В начало\]](#)

## Предварительные требования

Мастер миграции баз данных SQL можно загрузить из проекта мастера миграции базы данных SQL на сайте [Codeplex](#). Распакуйте пакет на локальном компьютере и запустите SQLAzureMW.exe.

[\[В начало\]](#)

## Перенос базы данных с помощью мастера миграции базы данных SQL

Миграция базы данных.

1. Выберите процесс, для которого требуется помощь мастера.
2. Выберите источник, для которого нужно создать скрипт.
3. Выберите объекты базы данных для создания скрипта.
4. Сформируйте скрипт. Скрипт можно изменить впоследствии.
5. Введите данные для соединения с целевым сервером. Целевую базу данных можно создать на База данных SQL Windows Azure.
6. Запустите скрипт с целевого сервера.

[\[В начало\]](#)

## Анализ базы данных с помощью мастера миграции баз данных SQL

Анализ базы данных по проблемам миграции.

1. Выберите процесс, для которого требуется помощь мастера.
2. Выберите источник, который необходимо проанализировать.
3. Выберите объекты базы данных для анализа.
4. Сформируйте скрипт.
5. В области результатов приводится сводка проблем, выявленных мастером.

[\[В начало\]](#)

## Анализ файла Transact-SQL с помощью мастера миграции базы данных SQL

Анализ базы данных по проблемам миграции.

1. Выберите процесс, для которого требуется помощь мастера.
2. Выберите файл Transact-SQL, который необходимо проанализировать как источник.
3. Сформируйте скрипт.
4. В области результатов приводится сводка проблем, выявленных мастером.

[\[В начало\]](#)



## Анализ файла трассировки с помощью мастера миграции базы данных SQL

Анализ базы данных по проблемам миграции.

1. Выберите процесс, для которого требуется помощь мастера.
2. Выберите файл трассировки, который необходимо проанализировать как источник.
3. Сформируйте скрипт.
4. В области результатов приводится сводка проблем, выявленных мастером.

[\[В начало\]](#)

### Ресурсы

- [Использование мастера миграции базы данных SQL Windows Azure](#) (видео)
- [Использование мастера миграции базы данных SQL Windows Azure \(дискуссионный форум\)](#)

## Как использовать SQL Server Data Tools для миграции базы данных в базу данных SQL Windows Azure

---

**Автор:** Шон Тинлайн Джонс (Shaun Tinline-Jones)

**Рецензент:** Билл Гибсон (Bill Gibson)

Средства SQL Server Data Tools (SSDT) используются для разработки баз данных вне сети для SQL Server и База данных SQL Windows Azure. Средства SSDT подходят для управления разработкой баз данных в проектах миграции баз данных.

- **Перед началом:** [Рекомендации](#), [Ограничения](#), [Предварительные условия](#)
- **Используйте средства SQL Server Data Tools с помощью** [пакетов DAC](#), [прямого подключения](#)

### До начала работы

Базам данных, которые переносятся в База данных SQL, могут потребоваться изменения схемы для решения проблемы зависимости функций, не поддерживаемых База данных SQL. Проекты баз данных SSDT подходят для управления разработкой, тестом и развертыванием базы данных. SSDT поддерживают интеграцию проекта базы данных в

решение Visual Studio, которое также включает проекты приложений, если изменения приложений являются частью миграции.

## Рекомендации

В сложные проекты миграции, которым требуются много изменений схемы, обычно встраивается несколько задач, таких как следующие:

- Импорт схемы в проект базы данных в SSDT.
- В результате задания База данных SQL в качестве цели проекта является создание сборки для выполнения первого анализа объектов, не поддерживаемых База данных SQL. Сборка отображает список ошибок для объектов, неподдерживаемых на База данных SQL. Оставьте в качестве набора целей База данных SQL, чтобы средства SSDT проверяли синтаксис на соответствие требованиям База данных SQL по мере внесения изменений в схему базы данных.
- Запустите задачу разработки базы данных для внесения всех необходимых изменений схемы, работающей по списку ошибок сборки и разрешающей все проблемы. После того как не останется неподдерживаемых объектов, необходимо выполнить развертывание тестов База данных SQL, чтобы убедиться, что все оставшиеся в базе данных объекты поддерживаются База данных SQL.
- Запустите задачу параллельной разработки приложения для выполнения всех изменений кода, которые становятся необходимыми в результате изменений схемы. Формирование трассировок инструкций Transact-SQL приложений и использование База данных SQL мастера миграции для проверки синтаксиса, который не поддерживается в База данных SQL. Также необходимо запустить параллельную задачу разработки для сборки процессов передачи данных из базы данных-источника в новую версию.
- Сборка пакета развертывания для схемы базы данных.
- Выполнение общего тестирования базы данных и приложения. Развертывание базы данных в тестовой службе База данных SQL. Импорт репрезентативного набора данных для теста процессов передачи данных. Необходимо выполнить полный функциональный тест приложения для тестовой базы данных, чтобы убедиться, что инструкции Transact-SQL, формируемые приложением, работают на База данных SQL.
- Выполнение общего рабочего развертывания схемы базы данных, приложений и запуск процесса передачи данных.

Дополнительные сведения о запуске проекта База данных SQL см. в разделе [Управление проектом миграции базы данных SQL Windows Azure](#).

Дополнительные сведения о разработке базы данных с использованием SSDT см. в разделе [Средства SQL Server Data Tools \(SSDT\)](#).

Дополнительные сведения об использовании SSDT для миграции базы данных в База данных SQL см. в разделе [Миграция базы данных в базу данных SQL Windows Azure с помощью средств SSDT](#).

## Ограничения

Текущая версия SSDT не обнаруживает все проблемы схемы, если целевое свойство проекта имеет значение База данных SQL. После того как средства SSDT сообщат об отсутствии ошибок схемы База данных SQL, убедитесь в этом путем развертывания базы данных в тестовой службе База данных SQL.

[\[В начало\]](#)

## Предварительные требования

Дополнительные сведения по установке средств SSDT см. в разделе [Установка средств SQL Server Data Tools](#)

[\[В начало\]](#)

## Использование средств SSDT с пакетами приложения уровня данных

При использовании SSDT для управления изменениями базы данных, необходимыми в рамках миграции, пакеты приложения уровня данных можно использовать в качестве механизма для передачи изменений схемы.

1. Используйте среду SQL Server Management Studio или скрипт PowerShell для извлечения пакета DAC из базы данных-источника.
2. Создайте проект базы данных в SSDT и импортируйте пакет приложения уровня данных.
3. Установите целевое свойство проекта базы данных SSDT в значение База данных SQL.
4. Внесите все изменения, необходимые для поддержки всех объектов базы данных База данных SQL.
5. Установите свойство сборки проекта в значение пакета приложения уровня данных.
6. Выполните сборку проекта для создания пакета приложения уровня данных.
7. Используйте среду SQL Server Management Studio или скрипт PowerShell для развертывания пакета приложения уровня данных в службу База данных SQL.

Дополнительные сведения об извлечении и развертывании пакетов приложения уровня данных см. в разделе [Как использовать пакет приложения уровня данных для миграции базы данных в базу данных SQL Windows Azure](#).

[\[В начало\]](#)

## Использование средств SSDT с прямым соединением

Для импорта схемы можно подключиться к базе данных-источнику напрямую. После завершения работы по обеспечению поддержки всех объектов в База данных SQL подключитесь к целевой службе База данных SQL и опубликуйте базу данных, содержащую новую схему.

1. Создайте проект базы данных в SSDT.
2. Подключитесь к базе данных-источнику напрямую и импортируйте схему в проект SSDT.
3. Установите целевое свойство проекта базы данных SSDT в значение База данных SQL.
4. Внесите все изменения, необходимые для поддержки всех объектов базы данных База данных SQL.
5. Чтобы развернуть новую схему, подключитесь к целевой службе База данных SQL и выполните операцию публикации.

Дополнительные сведения об извлечении и развертывании пакетов приложения уровня данных см. в разделе [Как использовать пакет приложения уровня данных для миграции базы данных в базу данных SQL Windows Azure](#).

[\[В начало\]](#)

## См. также

[Выбор средств для миграции базы данных в базу данных SQL Windows Azure](#)

[Как использовать пакет приложения уровня данных для миграции базы данных в базу данных SQL Windows Azure](#)

# Как использовать пакет DAC BACPAC для миграции базы данных в базу данных SQL Windows Azure

---

**Автор:** Шон Тинлайн Джонс (Shaun Tinline-Jones)

**Рецензент:** Адам Махуд (Adam Mahood)

Можно выполнить миграцию схемы и данных из базы данных SQL Server, экспортируя BACPAC из существующей базы данных, поместив файл BACPAC в учетную запись службы больших двоичных объектов Windows Azure, а затем импортировав BACPAC в База данных SQL Windows Azure.

- **Перед началом:** [Рекомендации](#), [Ограничения](#), [Предварительные условия](#)
- **Процесс:** [Миграция в DAC BACPAC](#)

## До начала работы

Приложение уровня данных представляет собой единое средство разработки, развертывания и управления объектами уровня данных. Приложение уровня данных позволяет разработчикам и администраторам баз данных уровня данных упаковывать объекты Microsoft SQL Server, в том числе объекты базы данных и экземпляров, в единый объект — пакет приложения уровня данных (DACPAC-файл). Формат BACPAC расширяет формат пакета приложения уровня данных и позволяет включить метаданные и данные таблиц в формате JSON, связанные с BACPAC, в дополнение к стандартному содержимому пакета приложения уровня данных. Можно упаковать базу данных SQL Server в BACPAC-файл и использовать его для переноса схемы и данных таблицы в База данных SQL Windows Azure.

## Рекомендации

Пакет приложения уровня данных и BACPAC ориентированы на различные сценарии.

1. BACPAC содержит схему и данные, но не поддерживает импорт в проект базы данных для изменения схемы. Основная цель BACPAC — перемещение базы данных из одной службы баз данных в другую (либо экземпляров СУБД или База данных SQL Windows Azure). BACPAC также может использоваться для архивации существующей базы данных в открытом формате. Это делает BACPAC хорошим инструментом для миграции, когда для базы данных не требуется изменение схемы.

2. Пакеты приложения уровня данных содержат только сведения о схеме, однако пакет можно импортировать в проект базы данных SSDT для дальнейшей разработки. В основном пакет приложения уровня данных используется при развертывании схемы базы данных в среде разработки, тестирования, а затем рабочей среде.

Служба импорта и экспорта для базы данных SQL Windows Azure может напрямую импортировать или экспортировать BACPAC-файлы между базой данных в База данных SQL Windows Azure и службой больших двоичных объектов Windows Azure. Служба импорта и экспорта для База данных SQL предоставляет общедоступные конечные точки REST для передачи запросов. [Портал платформы Windows Azure](#) содержит интерфейс для вызова службы импорта и экспорта базы данных SQL.

[\[В начало\]](#)

## Ограничения

DAC BACPAC может использоваться только для миграций, в которых не нужно вносить изменения в базу данных для обработки объектов, которые не поддерживаются в База данных SQL Windows Azure. Если такие изменения необходимы, рассмотрите следующее.

1. Используйте пакет приложения уровня данных и SQL Server Data Tools (SSDT) для изменения схемы базы данных и внесения необходимых изменений перед развертыванием в База данных SQL Windows Azure. Дополнительные сведения см. в разделе [Как использовать пакет приложения уровня данных для миграции базы данных в базу данных SQL Windows Azure](#).
2. Внесите все изменения в схему в базе данных-источнике перед экспортом DAC BACPAC.

Существует проект SQL DAC Examples, создающий неподдерживаемую программу командной строки, которую можно использовать для экспорта и импорта BACPAC. Проект SQL DAC Examples можно загрузить с сайта [CodePlex](#). Для проекта требуется DAC Framework. Дополнительные сведения об использовании программой, созданной проектом, см. в разделе [Справочник по клиентским инструментам DAC Framework](#).

[\[В начало\]](#)

## Предварительные требования

Для работы с DAC BACPAC необходимо установить программный клиент DAC, известный как DAC Framework. DAC Framework входит в состав *SQL Server Data Tools* и программ SQL Server, таких как среда *SQL Server Management Studio*. При работе с База данных SQL рекомендуется использовать версию DAC Framework, включенную в *SQL Server Data Tools* и

SQL Server 2012. Также, можно обновить более ранние версии DAC Framework, установив эти три пакета из [пакета дополнительных компонентов SQL Server 2012](#):

- Системные типы Microsoft CLR для Microsoft SQL Server 2012
- Microsoft SQL Server 2012 Transact-SQL Script DOM
- Платформа приложения уровня данных Microsoft SQL Server 2012

Дополнительные сведения о совместимости версий DAC Framework и SQL Server см. в разделе [Поддержки приложения уровня данных для объектов и версий SQL Server](#).

[\[В начало\]](#)

## Перенос файла DAC BACPAC

Далее представлены действия, необходимые для переноса базы данных из SQL Server в База данных SQL.

### 1. Экспорт файла BACPAC из существующей базы данных SQL Server

Воспользуйтесь мастером *экспорта приложения уровня данных* в среде SQL Server SQL Server Management Studio 2012 для экспорта файла BACPAC непосредственно в учетную запись службы больших двоичных объектов Windows Azure. Для запуска мастера щелкните базу данных правой кнопкой в обозревателе объектов, выберите **Задачи**, а затем выберите **Экспорт приложения уровня данных**. На странице **Параметры экспорта** используйте элемент управления **Сохранить в Windows Azure**, чтобы указать расположение службы больших двоичных объектов Windows Azure. При необходимости можно использовать программу SQL DAC Examples. Дополнительные сведения об экспорте BACPAC из SQL Server см. в разделе [Экспорт приложения уровня данных](#).

Требуется учетная запись хранения Windows Azure для экспорта BACPAC в Windows Azure. Дополнительные сведения см. в разделе [Создание учетной записи хранения](#).

### 2. Перемещение файла BACPAC в службу больших двоичных объектов Windows Azure

Если используется пример SQL DAC для экспорта BACPAC в локальный файл на компьютере и необходимо использовать Портал управления Windows Azure для импорта BACPAC в ssSDS, переместите файл BACPAC в учетную запись больших двоичных объектов Windows Azure. Можно скопировать файл с помощью [средства управления Windows Azure](#) или [приложения от Майкрософт под кодовым названием «Передача данных»](#).

Если планируется использовать пример SQL DAC для импорта в База данных SQL Windows Azure, не требуется копировать файл BACPAC в службу больших двоичных объектов Windows Azure.

### 3. Импорт BACPAC в База данных SQL Windows Azure

После экспорта BACPAC можно импортировать для создания базы данных в База данных SQL Windows Azure. Воспользуйтесь Портал управления Windows Azure для импорта файла BACPAC, который хранится в службе больших двоичных объектов Windows Azure. На ленте выберите команду **Импорт**, чтобы запустить окно **Импорт базы данных из хранилища Windows**. При необходимости можно использовать пример SQL DAC для импорта BACPAC, сохраненного в локальном файле на компьютере. Дополнительные сведения об импорте BACPAC в База данных SQL Windows Azure см. в разделе [Как импортировать приложение уровня данных](#).

[\[В начало\]](#)

## Ресурсы

- [Как использовать службу импорта и экспорта приложений уровня данных с базой данных SQL Windows Azure](#)

# Как использовать пакет приложения уровня данных для миграции базы данных в базу данных SQL Windows Azure

---

**Автор:** Шон Тинлайн Джонс (Shaun Tinline-Jones)

**Рецензент:** Адам Махуд (Adam Mahood)

Приложения уровня данных поддерживают извлечение схемы, кода и конфигурации базы данных в один файл пакета. Пакет приложения уровня данных можно затем использовать для развертывания новой копии базы данных в другой системе или импорта определения базы данных в проект *SQL Server Data Tools* (SSDT) для дальнейшего разработки. Пакеты приложения уровня данных не содержат данных, только определения объектов в базе данных.

- **Перед началом:** [Рекомендации](#), [Ограничения](#), [Предварительные условия](#)
- **Процесс:** [Миграция в пакет приложения уровня данных](#)



## До начала работы

Приложение уровня данных представляет собой единое средство разработки, развертывания и управления объектами уровня данных. Приложение уровня данных позволяет разработчикам и администраторам баз данных уровня данных упаковывать объекты Microsoft SQL Server, в том числе объекты базы данных и экземпляров, в единый объект — пакет приложения уровня данных (DACPAC-файл). Существует два способа создать файл пакета приложения уровня данных. Можно сформировать проект базы данных SSDT для сборки пакета приложения уровня данных или извлечь его из существующей базы данных. Пакет приложения уровня данных представляет собой сжатый файл, содержащий XML-представление определений объектов или метаданных базы данных. Затем можно развернуть этот пакет, чтобы создать копию базы данных в База данных SQL Windows Azure.

## Рекомендации

Пакет приложения уровня данных лучше использовать с *SQL Server Data Tools* для реализации любых изменений базы данных, необходимых для переноса базы данных в База данных SQL Windows Azure. Импортируйте пакет приложения уровня данных для создания проекта базы данных, внесите все необходимые изменения и затем выполните построение проекта для сборки нового пакета приложения уровня данных.

Использование пакета приложения уровня данных и проекта базы данных SSDT для передачи схемы при миграции — это хороший выбор, если после завершения миграции будет осуществляться дополнительная разработка. Для пакетов приложения уровня данных используется управление версиями, и существует процесс обновления приложения уровня данных. Одну версию пакета приложения уровня данных можно использовать для передачи схемы во время миграции. Если после миграции осуществляется дополнительная разработка, можно построить новую версию пакета приложения уровня данных и использовать ее для обновления рабочей базы данных. Дополнительные сведения об обновлениях приложения уровня данных см. в разделе [Обновление приложения уровня данных](#).

[\[В начало\]](#)

## Ограничения

Пакет приложения уровня данных не содержит данных таблиц, поэтому его можно использовать только для переноса определений схемы. Для переноса данных следует применять другой процесс. Дополнительные сведения о выборе процесса передачи

данных см. в разделе [Выбор средств для миграции базы данных в базу данных SQL Windows Azure](#).

Если для миграции не требуется вносить изменения в базу данных, в качестве альтернативы можно извлечь файл DAC BACPAC для переноса определений базы данных и самих данных. BACPAC-файл включает в себя набор данных таблиц в формате JSON и те же определения схемы, что и в пакете приложения уровня данных. Дополнительные сведения см. в разделе [Как использовать пакет DAC BACPAC для миграции базы данных в базу данных SQL Windows Azure](#).

В рамках проекта базы данных SSDT можно задать скрипты, выполняемые перед развертыванием и после развертывания. Это скрипты Transact-SQL, которые могут выполнять любые действия, в том числе вставить данные после развертывания. Однако не рекомендуется вставлять большой объем данных с помощью скриптов развертывания пакета приложения уровня данных.

[\[В начало\]](#)

## Предварительные требования

Для работы с пакетами приложения уровня данных необходимо установить программный клиент приложения уровня данных, известный как DAC Framework. DAC Framework входит в состав *SQL Server Data Tools* и программ SQL Server, таких как среда *SQL Server Management Studio*. При работе с База данных SQL рекомендуется использовать версию DAC Framework, включенную в *SQL Server Data Tools* и SQL Server 2012. Также можно обновить более ранние версии DAC Framework, установив эти три пакета из [пакета дополнительных компонентов SQL Server 2012](#):

- Системные типы Microsoft CLR для Microsoft SQL Server 2012
- Microsoft SQL Server 2012 Transact-SQL Script DOM
- Платформа приложения уровня данных Microsoft SQL Server 2012

Дополнительные сведения о совместимости версий DAC Framework и SQL Server см. в разделе [Поддержки приложения уровня данных для объектов и версий SQL Server](#).

[\[В начало\]](#)

## Миграция пакета приложения уровня данных

Для переноса схемы базы данных SQL Server в База данных SQL Windows Azure сначала извлеките из существующей базы данных, удалите все зависимости объектов, которые не

поддерживаются в База данных SQL Windows Azure, и затем разверните пакет приложения уровня данных в База данных SQL Windows Azure.

### 1. Извлеките пакет приложения уровня данных из базы данных SQL Server.

Пакет приложения уровня данных можно извлечь из существующей базы данных в компонент Datadase Engine SQL Server с помощью скрипта PowerShell или *мастера извлечения приложения уровня данных* в среде *SQL Server Management Studio*.

Сведения о необходимых компонентах и извлечении см. в разделе [Извлечение приложения уровня данных из базы данных](#).

Для извлечения используются следующие основные действия.

- a. Задайте свойства приложения уровня данных, в том числе имя, версию, описание и расположение файла пакета приложения уровня данных.
- b. Проверьте, что все объекты базы данных поддерживаются приложением уровня данных.
- c. Выполните сборку пакета.

### 2. Проверьте пакет приложения уровня данных перед развертыванием в База данных SQL Windows Azure.

Рекомендуется просмотреть содержимое пакета приложения уровня данных перед его развертыванием в рабочей среде, особенно если пакет не был разработан в организации. Дополнительные сведения см. в разделе [Проверка пакета приложения уровня данных](#).

Пакеты приложения уровня данных поддерживают некоторые типы объектов, которые не поддерживаются в База данных SQL Windows Azure. Воспользуйтесь экспериментальной службой *оценки совместимости базы данных SQL Windows Azure*, чтобы определить, содержит ли пакет приложения уровня данных объекты, которые не поддерживаются База данных SQL Windows Azure, прежде чем развернуть пакет в База данных SQL. Дополнительные сведения и учебник по использованию этой службы можно найти в статье [Служба оценки совместимости базы данных SQL Windows Azure](#).

Перед миграцией базы данных в База данных SQL удалите все зависимости объектов, которые отображаются как исключения процессом извлечения приложения уровня данных или службой *оценки совместимости базы данных SQL Windows Azure*. Удаление этих объектов из базы данных, вероятно, потребует внесения изменений в приложения, использующие базу данных.

### 3. Развертывание пакета приложения уровня данных в База данных SQL:

Пакет приложения уровня данных можно развернуть в *ssSDS* с помощью скрипта PowerShell или *мастера развертывания приложений уровня данных* в среде *SQL ServerManagement Studio*. Сведения о необходимых компонентах и развертывании см. в разделе [Развертывание приложения уровня данных](#). *Мастер развертывания приложений уровня данных* также можно запустить на *портале управления базами данных SQL Windows Azure*. Дополнительные сведения см. в разделе [Администрирование баз данных \(портал управления для базы данных SQL Windows Azure\)](#).

Для развертывания используются следующие основные действия.

- a. Выберите пакет приложения уровня данных.
- b. Проверьте содержимое пакета.
- c. Настройте свойства развертывания базы данных, в которых указана база данных в База данных SQL Windows Azure.
- d. Разверните пакет.

[\[В начало\]](#)

## Ресурсы

- [Приложения уровня данных](#)
- [Импорт в проект базы данных](#)

# Как создать скрипты для миграции базы данных в базу данных SQL Windows Azure

---

**Автор:** Шон Тинлайн-Джонс (Shaun Tinline-Jones)

Мастер формирования скриптов может использоваться для создания скриптов Transact-SQL для базы данных SQL Server и/или связанных объектов в выбранной базе данных. Затем скрипты можно использовать для передачи схемы и/или данных в База данных SQL Windows Azure.

- **Перед началом:** [Рекомендации](#), [Необходимые условия](#)
- **Использование мастера формирования скриптов для** [миграции базы данных](#)

# До начала работы

## Рекомендации

Использование мастера формирования скриптов для миграции SQL Server базы данных в База данных SQL должно ограничиваться следующими случаями.

- Наличие команд, которые имеют опыт работы с мастером.
- Миграция простых баз данных, которым требуется небольшое количество изменений схемы для запуска в База данных SQL. Скрипты, сформированные из базы данных-источника, могут быть изменены перед их использованием для создания новой версии базы данных в База данных SQL, но при использовании проекта базы данных в SQL Server Data Tools имеется более обширная поддержка внесения изменений в схему.
- Миграция небольших баз данных, которые не имеют много данных. Мастер создает скрипты, которые используют для передачи данных инструкции INSERT, а не массовое копирование. Инструкции INSERT могут подвергаться регулированию, если таблицы содержат слишком много данных, а также выполняются медленнее операций массового копирования.

Будьте внимательны при выборе параметров в мастере. Можно легко выбрать вариант, который предотвратит передачу важной информации, например параметр, указывающий, что не следует создавать индексы.

[\[В начало\]](#)

## Предварительные требования

Мастер формирования скриптов устанавливается совместно с SQL Server. Используйте мастер из SQL Server 2008 R2 или более поздней версии.

[\[В начало\]](#)

## Миграция базы данных с помощью мастера формирования скриптов

Использование мастера включает следующие основные шаги.

1. Откройте среду SQL Server Management Studio и подключитесь к экземпляру компонента Database Engine.
2. В **обозревателе объектов** щелкните правой кнопкой мыши базу данных, чтобы открыть меню, выберите пункт **Задачи**, затем **Сформировать скрипты**.
3. Выберите объекты для экспорта.

4. Задайте параметры формирования скриптов. У вас есть возможности сохранить скрипт в файл, буфер обмена, окно нового запроса. Также его можно опубликовать на веб-службу.
5. Задайте дополнительные параметры формирования скриптов.

По умолчанию скрипт создается для изолированного экземпляра SQL Server. Чтобы изменить конфигурацию, нажмите кнопку **Дополнительно** в диалоговом окне **Задание параметров скриптов**, а затем установите для свойства **Скрипт для типа компонента Database Engine** значение **База данных SQL**.

Вы также можете установить для свойства **Типы данных для внесения в скрипт** одно из следующих значений на основе ваших требований: **Только схема**, **Только данные**, **Схемы и данные**.

После создания скрипта имеется возможность изменить его перед его запуском в База данных SQL для передачи базы данных.

[\[В начало\]](#)

## Ресурсы

- [Как мигрировать базу данных с помощью мастера формирования скриптов \(база данных SQL Windows Azure\)](#)

# Как использовать bcp для миграции базы данных в базу данных SQL Windows Azure

---

**Автор:** Шон Тинлайн Джонс (Shaun Tinline-Jones)

Можно использовать служебную программу SQL Server **bcp** для высокопроизводительного перемещения базы данных при миграции базы данных в SQL Server в База данных SQL Windows Azure. Сначала используйте программу **bcp** для копирования данных из исходной таблицы в файл данных. Затем запустите программу снова, чтобы скопировать файл данных в целевую таблицу. Программа **bcp** перемещает только данные, поэтому для переноса схемы базы данных следует использовать другой процесс.

- **Перед началом:** [Рекомендации](#), [Ограничения](#), [Предварительные условия](#)
- **Используйте программу bcp для** [переноса данных](#)

## До начала работы

**Вср** — это служебная программа командной строки, которая предназначена для выполнения высокопроизводительной массовой передачи данных на SQL Server или База данных SQL Windows Azure. Это не средство миграции данных. Она не извлекает и не создает схему. Сначала следует перенести схему в базу данных в База данных SQL Windows Azure с помощью одного из инструментов миграции схемы, например мастера создания скриптов, или выполнить извлечение и развертывание пакета приложения уровня данных (DAC). Дополнительные сведения о процессе миграции схемы см. в разделе [Выбор средств для миграции базы данных в базу данных SQL Windows Azure](#).

Программа **Вср** использует функцию массового копирования SQL Server, которая также предоставлена в API-интерфейсе SQL Server. Несколько средств миграции, таких как мастер миграции База данных SQL Windows Azure и пакеты DAC BACPAC, также используют функции массового копирования для передачи данных.

## Рекомендации

Используйте рекомендации по массовому копированию для повышения производительности копирования данных в большую целевую таблицу. Пример.

- Параметр `-N` используется для передачи данных в собственном режиме, поэтому преобразование типов данных не нужно.
- Используйте параметр `-b` для указания размера пакета. Каждый пакет импортируется и регистрируется как отдельная транзакция. По умолчанию импорт всех строк в файле данных выполняется в одном пакете. Если транзакция завершится неудачно, то будет выполнен откат только операций вставки в текущем пакете. Определение наилучшего размера пакета и использование размера пакета — хороший способ уменьшить вероятность потери соединения с База данных SQL Windows Azure во время переноса данных.
- Используйте указания для **вср**.
  - Используйте подсказку `-h «TABLOCK»` при импорте для указания использования блокировки уровня массового обновления таблицы на время выполнения операции массовой загрузки. Это уменьшает издержки блокировки путем использования одной блокировки таблицы, а не блокировки каждой строки.
  - Указание `-h «ORDER(...)»` при экспорте для сортировки файла данных. Производительность операции выполнения массового импорта увеличивается, если импортируемые данные упорядочены согласно кластеризованному индексу таблицы.

- Для больших таблиц можно разбить процесс импортирования копии на несколько потоков, которые могут выполняться параллельно. Если было выполнено массовое копирование данных из исходной таблицы в один файл данных, используйте параметры `-F firstrow` и `-L lastrow`, чтобы указать, какая часть файла данных будет обрабатываться при каждом запуске **bcp**.

Дополнительные сведения о рекомендациях для массового копирования см. в разделе [Оптимизации производительности массового импорта](#).

Если используется **IDENTITY** для формирования в таблице первичных ключей, применяйте параметр `-E` программы **bcp** для сохранения ключей, созданных в базе данных-источнике. Использование параметра `-E` позволяет предотвратить любые нарушения внешних ключей при импорте при условии, что при импорте в таблицах не выполняются никакие другие обновления. Задайте для базы данных режим только для чтения, чтобы обеспечить запрет каких-либо обновлений.



#### Примечание

В определенный момент времени **bcp** работает только с одной таблицей, поэтому целостность транзакций для нескольких таблиц при извлечении данных из базы данных-источника не поддерживается. Эту проблему можно устранить, задав во время экспорта для базы данных-источника режим одного пользователя или режим только для чтения.

[\[В начало\]](#)

## Ограничения

При массовом копировании таблицы в целевой базе данных должны быть пустыми. Нельзя выполнить несколько массовых операций копирования в одну таблицу, если не выполнено усечение или удаление всех строк, вставленных в предыдущей операции массового копирования.

[\[В начало\]](#)

## Предварительные требования

**bcp** поставляется вместе с SQL Server. Установите клиентские служебные программы из SQL Server 2008 R2 или более поздних версий SQL Server для получения версии **bcp**, поддерживаемой для использования с База данных SQL Windows Azure.

[\[В начало\]](#)



## Использование программы **bcp** для переноса данных

Процесс перемещения данных из таблицы в базе данных-источнике в копию таблицы в целевой базе данных с помощью **bcp** состоит из пяти этапов.

### 1. Перенос схемы.

Используйте механизм переноса схемы, например мастер создания скриптов или пакет DAC ВАСРАС, для создания копии базы данных в База данных SQL Windows Azure. После завершения этого этапа в базе данных База данных SQL будут созданы все таблицы. При этом таблицы не будут содержать каких-либо данных.

### 2. Экспортирование данных в файлы данных.

Для каждой таблицы в базе данных-источнике SQL Server запустите операцию **bcp out** для копирования данных из таблицы в файл данных. Это пример экспорта данных из одной таблицы в файл данных.

```
bcp tableName out C:\filePath\exportFileName.dat -S serverName -  
T -n -q
```

Параметр out задает копирование данных из SQL Server. Параметр -n выполняет операцию массового копирования с использованием собственных типов данных базы данных. Параметр -q выполняет инструкцию SET QUOTED\_IDENTIFIER ON в соединении между программой **bcp** и экземпляром компонента Database Engine.

### 3. Оптимизация массового копирования

Внесите необходимые изменения в схему целевой базы данных, чтобы повысить производительность копирования данных в большие таблицы. Для этого можно отключить некластеризованные индексы, триггеры и ограничения.

### 4. Импортирование файла данных База данных SQL

Запустите программу **bcp** для каждой таблицы в целевой базе данных База данных SQL Windows Azure для копирования данных из файла экспортируемых данных в таблицу. В этом примере **bcp** запускается трижды для копирования данных в одну таблицу из файла данных, который имеет приблизительно 300 000 строк. Каждый запуск копирует около 100 000 строк.

```
Bcp tableName in c:\filePath\exportFileName.dat -n -U  
userName@serverName -S tcp:serverName.database.windows.net -P  
password -b 200 -L 99999 -h"TABLOCK"
```

```
Bcp tableName in c:\filePath\exportFileName.dat -n -U
userName@serverName -S tcp:serverName.database.windows.net -P
password -b 200 -F 100000 -L 199999 -h"TABLOCK"

Bcp tableName in c:\filePath\exportFileName.dat -n -U
userName@serverName -S tcp:serverName.database.windows.net -P
password -b 200 -F 200000 -h"TABLOCK"
```

Параметр in обозначает копирование данных в База данных SQL Windows Azure.

Параметр -b указывает количество строк в каждом пакете импортированных данных.

Параметры -L *lastrow* и -F *firstrow* используются для определения, какая часть файла данных обрабатывается при каждом запуске.

## 5. Устранение оптимизации схемы

Восстановите все элементы схемы, которые были убраны для оптимизации операций массовой вставки. Например, включите некластеризованные индексы, триггеры или ограничения, которые были отключены на этапе 3.

[\[В начало\]](#)

## Ресурсы

- [Программа bcp](#)

# Как использовать службы Integration Services для миграции базы данных в базу данных SQL Windows Azure

---

**Автор:** Шон Тинлайн Джонс (Shaun Tinline-Jones)

Службы SQL Server Integration Services (SSIS) могут быть использованы в случаях, когда требуются сложные преобразования данных для миграции базы данных с локального экземпляра SQL Server на База данных SQL Windows Azure.

- **Перед началом:** Рекомендации, [Ограничения](#), [Предварительные условия](#)
- Службы SQL Server Integration Services можно использовать для: [миграции баз данных](#), анализа баз данных, анализа файлов Transact-SQL, анализа файлов трассировки

## До начала работы

Службы SSIS могут использоваться для выполнения широкого круга задач переноса данных. Службы SSIS обеспечивают поддержку сложных преобразований данных и рабочих процессов между источником и пунктом назначения. Это хороший подход к передаче данных для баз данных, которые требуют большого объема изменений для работы в База данных SQL Windows Azure. Пакеты передачи данных служб SSIS можно использовать с другим механизмом для передачи схемы базы данных, таким как пакет приложения уровня данных.

## Рекомендации

Наиболее мощной возможностью служб SSIS является возможность выполнения сложных преобразований для миграций, которые требуют значительных изменений схемы. В этих проектах лучше использовать другой механизм для управления разработкой новой схемы (например, с помощью средств SQL Server Data Tools). Но при этом следует использовать пакеты передачи данных служб SSIS для обработки преобразования исходных данных в формат, указанный для целевой базы данных. Хотя службы SSIS предоставляют типы пакетов для передачи сведений о схеме, они наиболее полезны в случае, если база данных-источник и целевая база данных имеют одинаковый формат. Другое случай, когда следует рассмотреть вопрос об использовании служб SSIS, — когда необходимо оптимизировать время загрузки данных, чтобы обеспечить временные параметры проекта, задаваемые клиентом.

Мастер импорта и экспорта служб SSIS может использоваться для быстрого создания пакетов, которые перемещают данные из одного источника данных в пункт назначения без преобразований. Мастер можно использовать для формирования основных пакетов, сопоставляющих данные из таблицы-источника с местом назначения. Затем вы можете изменить пакет для добавления надежной логики обработки ошибок и повторных попыток.

Адаптер ADO.NET служб SSIS поддерживает База данных SQL. Он предоставляет возможность массовой загрузки данных специально для База данных SQL Windows Azure. Используйте адаптер загрузки данных назначения «ADO.NET» служб SSIS для передачи данных в База данных SQL Windows Azure.

Для каждого назначения База данных SQL Windows Azure «ADO.NET» не забудьте использовать параметр **По возможности использовать массовую вставку**. Это позволит вам использовать возможности массовой загрузки для повышения производительности передачи. Еще одним способом повышения производительности является разбиение

исходных данных среди нескольких файлов в файловой системе. В конструкторе служб SSIS можно ссылаться на файлы с помощью **компонента неструктурированных файлов**.

## Ограничения

Службы SSIS недоступны в качестве службы Windows Azure, схожей с База данных SQL Windows Azure. Пакеты служб SSIS можно запускать на локальном экземпляре SQL Server для передачи данных в База данных SQL Windows Azure.

Выполнение пакета может завершиться со сбоем из-за проблем регулирования или сети. Пакеты следует разрабатывать так, чтобы их можно было возобновить в точке сбоя без повторного выполнения всей работы, выполненной до сбоя.

Соединение с базой данных База данных SQL Windows Azure с помощью OLE DB не поддерживается.

[\[В начало\]](#)

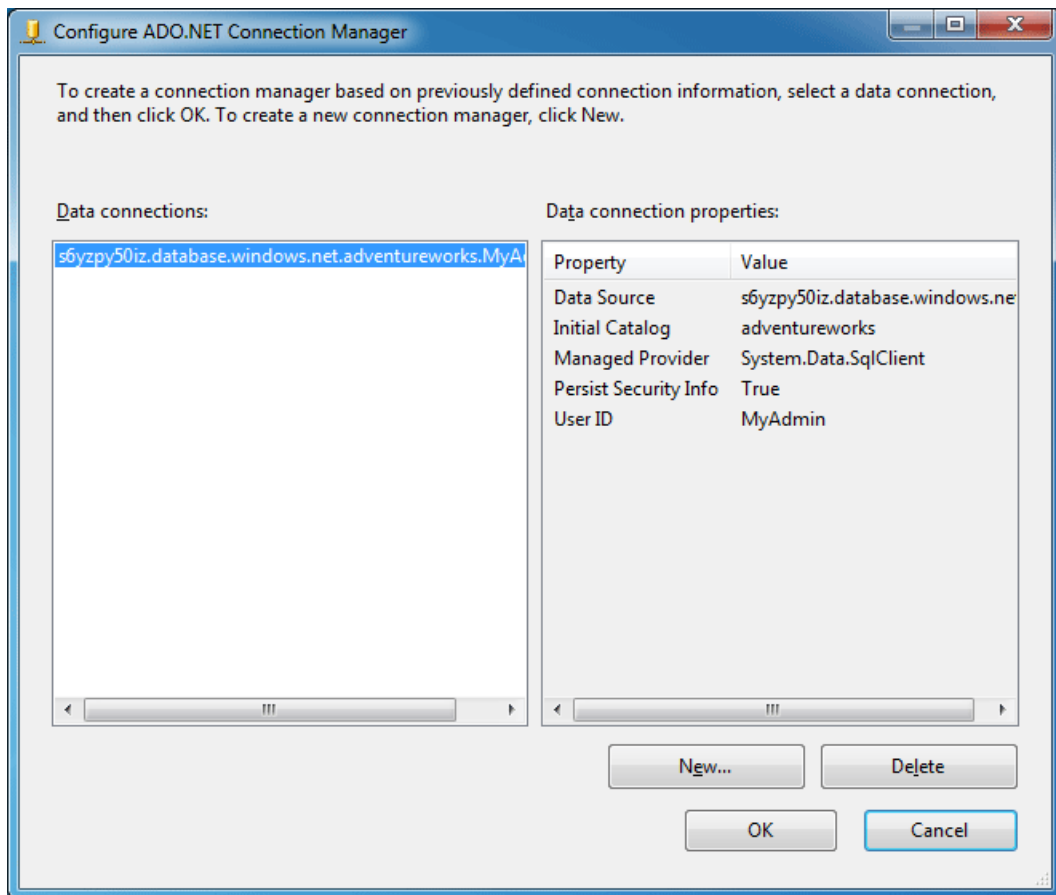
## Предварительные требования

Версия SQL Server служб Integration Services в SQL Server 2008 R2 или более поздние версии поддерживают База данных SQL Windows Azure.

**[В начало]**

## Миграция базы данных с помощью служб SQL Server Integration Services

Ниже приводится снимок экрана настройки соединения ADO.NET с База данных SQL Windows Azure.



[\[В начало\]](#)

## Ресурсы

- [Проектирование и разработка пакетов \(службы Integration Services\)](#)

# Как использовать мастер импорта и экспорта для миграции базы данных в базу данных SQL Windows Azure

**Автор:** Шон Тинлайн Джонс (Shaun Tinline-Jones)

При переносе базы данных SQL Server в База данных SQL Windows Azure мастер импорта и экспорта SQL Server предоставляет простой способ создания пакета служб SQL Server

Integration Services для передачи данных. В этот пакет впоследствии можно добавить улучшенную логику обработки ошибок и повторных попыток.

- **Перед началом:** [Рекомендации](#), [Ограничения](#), [Предварительные условия](#)
- **Используйте мастер экспорта-импорта для** [переноса данных](#)

## Мастер импорта и экспорта SQL Server

### Рекомендации

Мастер экспорта и импорта SQL Server настраивает исходное и конечное соединение с назначением пакета. Затем он добавляет преобразования данных, необходимые для экспорта из одного из нескольких источников данных, таких как база данных SQL Server, или импорта в источник данных, такой как База данных SQL Windows Azure. Можно сразу запустить пакет, сохранить его для последующего запуска или изменить пакет в [конструкторе служб SSIS](#).

Адаптер ADO.NET служб SSIS поддерживает База данных SQL Windows Azure. Он предоставляет возможность массовой загрузки данных специально для База данных SQL Windows Azure. Используйте адаптер загрузки данных назначения «ADO.NET» служб SSIS для передачи данных в База данных SQL Windows Azure.

[\[В начало\]](#)

### Ограничения

Хотя мастер импорта и экспорта SQL Server может передавать сведения о схеме, он перемещает только определения таблиц, а не индексы и другие ограничения. База данных SQL Windows Azure требуется, чтобы у всех таблиц был кластеризованный индекс, поэтому используйте мастер импорта и экспорта только для передачи данных. Используйте другой процесс для передачи схемы, например мастер создания скриптов или пакеты приложения уровня данных. Дополнительные сведения см. в разделе [Выбор средств для миграции базы данных в базу данных SQL Windows Azure](#).

Выполнение пакета может завершиться со сбоем из-за проблем регулирования или сети. Постройте пакет так, чтобы его можно было возобновить в точке сбоя, вместо того, чтобы повторно запускать весь пакет после сбоя.

Соединение с базой данных База данных SQL Windows Azure с помощью OLE DB не поддерживается.



### Примечание

На 64-разрядном компьютере службы Integration Services устанавливают 64-разрядную версию мастера импорта и экспорта SQL Server (DTSWizard.exe). Однако некоторые источники данных, такие как Access и Excel, располагают только 32-разрядным поставщиком. Для работы с этими источниками данных необходимо установить и запустить 32-разрядную версию мастера. Чтобы установить 32-разрядную версию мастера, во время работы программы установки SQL Server нужно выбрать «Клиентские средства» или среду Business Intelligence Development Studio.

[\[В начало\]](#)

## Предварительные требования

Мастер импорта и экспорта SQL Server, установленный с клиентскими вспомогательными программами SQL Server 2008 R2 и более поздних версий, поддерживает База данных SQL Windows Azure.

[\[В начало\]](#)

## Перенос данных с помощью мастера импорта таблиц

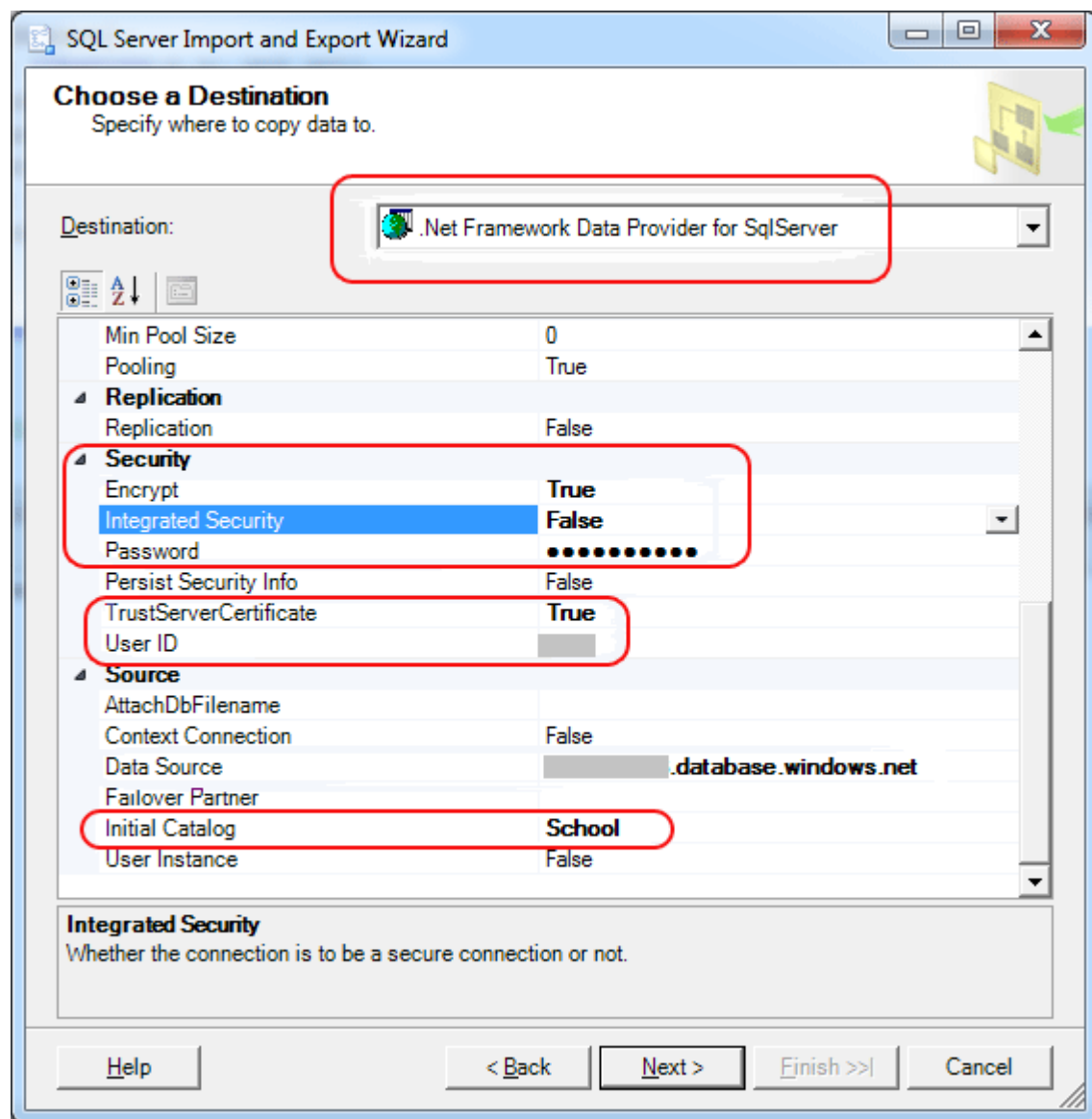
Существует несколько способов запуска мастера из командной строки или различных инструментов SQL Server.

1. В меню **Пуск** выберите пункт **Все программы**. Выберите **Microsoft SQL Server 2012** и затем щелкните **Импорт и экспорт данных (64-разрядная версия)** или **Импорт и экспорт данных (32-разрядная версия)**.
2. В среде Business Intelligence Development Studio щелкните правой кнопкой мыши папку **Пакеты служб SSIS** в обозревателе решений, а затем выберите **Мастер импорта и экспорта служб SSIS**.
3. В среде Business Intelligence Development Studio в меню **Проект** щелкните **Мастер импорта и экспорта служб SSIS**.
4. В среде SQL Server Management Studio подключитесь к нужному типу сервера компонента Database Engine. Разверните узел **Базы данных**, правой кнопкой мыши щелкните базу данных, выберите пункт **Задачи**, затем выберите пункт **Импорт данных** или **Экспорт данных**.
5. В окне командной строки запустите DTSWizard.exe. 64-разрядный мастер расположен в папке C:\Program Files\Microsoft SQL Server\110\DTS\Binn. 32-разрядный мастер расположен в папке C:\Program Files (x86)\Microsoft SQL Server\110\DTS\Binn.

Для миграции используются следующие основные действия.

1. Выберите источник данных для копирования.
2. Выберите место назначения для копирования данных.

Чтобы экспортировать данные в База данных SQL Windows Azure, выберите *Поставщик данных .NET Framework для SQLServer* как место назначения.



3. Выберите копию таблицы или запроса.
4. Выберите исходные объекты.



5. Сохраните и выполните пакет.



#### Примечание

При сохранении пакета его следует добавить в существующий проект служб Integration Services, прежде чем он станет доступным для запуска и изменения в среде BI Development Studio.

[\[В начало\]](#)

## Ресурсы

- [Запуск мастера импорта и экспорта SQL Server](#)
- [Мастер импорта и экспорта SQL Server](#)

# Как использовать помощник миграции SQL Server с базой данных SQL Windows Azure

---

**Автор:** Шон Тинлайн Джонс (Shaun Tinline-Jones)

Существует несколько помощников миграции SQL Server, которые могут использоваться для переноса баз данных в База данных SQL Windows Azure из других продуктов для базы данных, таких как Oracle, MySQL, Sybase и Microsoft Access.

- **Перед началом:** [Обязательные условия](#)
- **Используйте помощник миграции SQL Server для** [переноса базы данных](#)

## До начала работы

Помощник по миграции Microsoft SQL Server (SSMA) — это семейство продуктов, позволяющих сократить стоимость и риск при переносе баз данных из Oracle, Sybase, MySQL или Microsoft Access в База данных SQL Windows Azure и SQL Server. SSMA автоматизирует все аспекты миграции, в том числе оценочный анализ миграции, преобразование схем и инструкций SQL, а также сам перенос данных и их проверку.

## Предварительные требования

SSMA можно загрузить из Интернета. Сведения о загрузке последней версии см. в документации к средствам миграции SQL Server. На момент написания версии, указанные ниже, были последними:

- [Помощник по миграции Microsoft SQL Server для Access версии 5.2](#)
- [Помощник по миграции Microsoft SQL Server для MySQL версии 5.2](#)
- [Помощник по миграции Microsoft SQL Server для Oracle версии 5.2](#)
- [Помощник по миграции Microsoft SQL Server для Sybase версии 5.2](#)

SSMA устанавливается с помощью мастера на основе установщика Windows. SSMA — это бесплатное средство, которое нужно активировать, загрузив регистрационный ключ. После установки и запуска приложение попросит зарегистрироваться и загрузить регистрационный ключ.

[\[В начало\]](#)

## Перенос базы данных с использованием помощника миграции SQL Server

Процесс миграции SSMA для Access включает следующие действия.

1. Создайте новый мастер миграции. Выберите базу данных SQL Windows Azure в поле **Миграция в**.
2. Добавьте базы данных Access.
3. Выберите объекты Access, которые нужно перенести.
4. Подключитесь к База данных SQL Windows Azure.
5. Свяжите таблицы. Если необходимо использовать существующие приложения Access с База данных SQL Windows Azure, свяжите исходные таблицы Access с перенесенными таблицами База данных SQL Windows Azure. Связывание изменяет базу данных Access, чтобы запросы, формы, отчеты и страницы доступа к данным использовали данные в База данных SQL Windows Azure вместо базы данных Access.
6. Преобразуйте выбранные объекты.
7. Загрузите преобразованные объекты в База данных SQL Windows Azure.
8. Перенесите данные для выбранных объектов Access.

[\[В начало\]](#)

## Ресурсы

- [Помощник миграции SQL Server](#)
- [Перенос приложений Microsoft Access в базу данных SQL Windows Azure](#) (видео)

- [SQL Server: управление миграцией](#)

## Вопросы переноса секционированных данных в базу данных SQL Windows Azure

---

**Авторы:** Шон Тинлайн-Джонс (Shaun Tinline-Jones)

**Соавторы:** Шридхар Пеллуру (Sreedhar Pelluru)

**Рецензенты:** Рама Рамани (Rama Ramani), Валерий Мизонов (Valery Mizonov), Чэн Кун (Kun Cheng), Стив Ховард (Steve Howard)

Секционирование данных — это хорошо зарекомендовавший себя и распространенный метод, используемый в локальных решениях для управления системами с очень большими базами данных. При использовании этого метода данные большой базы данных разбиваются и распределяются по нескольким секциям. Прежде всего это делается для улучшения управляемости, производительности и доступности базы данных. Администраторы и разработчики баз данных, которые традиционно работали с SQL Server, используют секционирование по множеству различных причин, например следующих:

- Повышение эффективности управления данными
- Повышение оперативности запросов
- Увеличение параллелизма запросов
- Добавление большей вычислительной мощности путем использования нескольких физических серверов
- Поддержание уровня доступности
- Преодоление ограничений физического хранилища
- Снижение ширины таблицы

При рассмотрении вопроса о миграции на платформу базы данных SQL Windows Azure с локального сервера SQL Server могут существовать дополнительные причины для секционирования данных, например следующие:

- Мультитенантность
- Доступ к большей вычислительной мощности
- Предотвращение регулирования
- Недоступность данных

- Более экономически эффективные расходы на хранение
- Ограничения на размер данных

Функции SQL Server, обычно используемые для секционирования данных в локальных решениях, включают в себя [репликацию](#), [секционирование таблиц](#), [секционированные представления](#), [распределенные секционированные представления](#), [стратегии файловых групп](#) и [межбазовые запросы \(распределенные запросы\)](#). База данных SQL Windows Azure не поддерживает все эти функции, за исключением секционированных представлений. База данных SQL предлагает [федерации](#), которые недоступны в локальной версии.

Если мигрируется простое решение, основанное на одной базе данных, для которой не требуется значительных ресурсов, то есть высокая вероятность того, что оно не использует какие-либо функции, которые прибегают к секционированию данных. Если решение использует упомянутые выше функции или имеет значительные требования к вычислительной мощности, то перенос функций в базу данных SQL может потребовать переработки модели базы данных.

## Перенос существующих локальных функций в базу данных SQL

### Репликация

С точки зрения секционирования репликация может использоваться в таких сценариях, как следующие:

- Перемещение подмножеств данных из нескольких источников в одно место
- Единая база данных служит в качестве базы данных с доступом на чтение и запись, а несколько подписчиков этой базы данных служат в качестве доступных только для чтения баз данных с балансировкой нагрузки.
- Подмножества данных обслуживают различных пользователей на основе таких параметров, как географическое положение, а затем данные реплицируются в решение, которое объединяет все эти подмножества (издатели).

Наиболее типичными причинами для реализации репликации являются следующие:

- Создание доступной только для чтения реплики базы данных OLTP, доступной только для чтения и записи.
- Ведение «горячего» резервирования копии базы данных.
- Заполнение промежуточных областей хранения.

- Обеспечение доступа релевантных данных только подмножеству пользователей.
- Повышение эффективности вычислений и управления на основе подмножеств данных.

База данных SQL не поддерживает репликацию. Функция базы данных SQL, наиболее близкая функции репликации, — это функция [синхронизации данных SQL Windows Azure](#). Эта функция основывается на триггерах, таблицах отслеживания измененных данных (CDC) и запланированных заданиях для обеспечения репликации данных. Это позволяет реплицировать некоторые таблицы, однако не позволяет выбрать подмножество данных из таблицы.

Во-первых, рассмотрим причины локальной конструкции репликации. Если целью является аварийное восстановление, база данных SQL может его обеспечить. Если целью является географическое аварийное восстановление, то база данных SQL пока не может его обеспечить. Если целью является разгрузка вычислительных ресурсов, если подписчики используют идентичную конструкцию, база данных SQL имеет параметр [Федерации](#).

Федерации в базе данных SQL позволяют повысить масштабируемость и увеличить производительность приложения на уровне базы данных приложения за счет горизонтального секционирования. Одна или несколько таблиц в базе данных разбиваются по строкам и разделяются между несколькими базами данных (членами федерации). Этот тип горизонтального секционирования часто называют шардингом.

## Секционирование таблиц

Данные секционированных таблиц подразделяются на блоки, которые могут быть распределены по нескольким файловым группам в базе данных. Данные секционируются горизонтально, поэтому группы строк сопоставляются с отдельными секциями. Как правило, большая таблица секционируется в случае, если она содержит (или предполагается, что она содержит) большое количество данных, которые используются различными способами, а запросы и обновления, выполняемые к таблице, не выполняются, как предполагалось, либо задачи обслуживания, такие как резервное копирование, занимают слишком много времени. Секционирование таблицы обычно реализуется для хранения данных на разных томах для повышения эффективности резервного копирования и восстановления, повышения производительности запросов и увеличения эффективности параллельной обработки запросов.

В данное время база данных SQL не поддерживает секционирование таблиц. База данных SQL позволяет избежать необходимости секционирования таблиц, если целью является решение вопросов емкости диска. Рассмотрите возможность использования [федераций базы данных SQL](#) при миграции на базу данных SQL.

## Секционированные представления

[Секционированные представления](#) позволяют вам разделить большую таблицу в базе данных на несколько небольших таблиц, содержащих определенные подмножества данных, а затем построить представление UNION ALL на основе небольших таблиц, в результате чего представление будет отображаться как одна таблица, содержащая все данные. Секционированные представления обычно используются для тех же целей, что и секционированные таблицы.

База данных SQL поддерживает секционированные представления, поэтому миграция секционированных представлений в базу данных SQL должна быть простой.

## Распределенные секционированные представления

[Распределенные секционированные представления](#) соединяют горизонтально секционированные данные из набора таблиц-элементов, распределенных среди различных экземпляров баз данных на разных серверах баз данных. Распределенные секционированные представления можно применять для использования вычислительной мощности различных серверов баз данных, повышения производительности запросов путем разбиения большой таблицы на несколько таблиц меньшего размера и распределения нагрузки по федерации серверов баз данных.

На данный момент база данных SQL не поддерживает распределенные секционированные представления. Рассмотрите возможность использования [федерации базы данных SQL](#).

## Стратегии файловых групп

Файловые группы представляют собой именованные коллекции файлов и используются для упрощения размещения данных и для выполнения административных задач, таких как функции резервного копирования и восстановления. Они обычно используются для увеличения производительности базы данных путем хранения данных на нескольких дисках, нескольких контроллерах дисков (избыточного количества независимых дисков) или дисковых системах RAID. При использовании нескольких файловых групп для файлов базы данных можно создавать индивидуальные резервные копии и восстанавливать их также индивидуально.

Файловые группы не поддерживаются в базе данных SQL. Если файловые группы используются для административных целей, нет необходимости беспокоиться по этому поводу, поскольку база данных SQL сама выполняет эти задачи. Если они используются для повышения производительности, рекомендуется использовать [федерации базы данных SQL](#).

## Межбазовые запросы

[Межбазовые запросы \(распределенные запросы\)](#) обычно используются для доступа к данным, распределенным по нескольким экземплярам SQL Server.

В данный момент модель доступа к данным базы данных SQL не поддерживает межбазовые запросы и команду USE. Межбазовые соединения или сравнения можно запрограммировать в приложении после того, как данные будут возвращены из соответствующих баз данных. Рассмотрите возможность использования [федераций базы данных SQL](#), поскольку это решение секционирования поддерживается в базе данных SQL.

## Перенос данных в другие службы управления данными в Windows Azure

**Авторы:** Сридхар Пелуру (Sreedhar Pelluru)

**Соавторы:** Рама Рамани (Rama Ramani)

В разделе [Общие сведения о службах управления данными в Windows Azure](#) представлены общие сведения о службах управления данными на платформе Windows Azure. В этом разделе содержатся рекомендации по миграции локальных приложений, которые будут использовать следующие службы управления данными: службу таблиц Windows Azure, службу больших двоичных объектов Windows Azure и диск Azure.

В следующей таблице сравнивается хранилище таблиц, хранилище больших двоичных объектов, диск Azure и локальное хранилище (не считается службой управления данными), что позволит вам выбрать хранилище для своего сценария.

Критерии сравнения	Локальное хранилище	Диск Azure	Хранилище таблиц	Хранилище больших двоичных объектов
Надежность	Ненадежное.  Может сохраняться после переработки одного	Надежное.  Диск Azure поддерживает надежным страничным большим	Надежное.  Хранилище таблиц обеспечивает масштабируемо е и	Надежное.  Хранилище больших двоичных объектов обеспечивает

Критерии сравнения	Локальное хранилище	Диск Azure	Хранилище таблиц	Хранилище больших двоичных объектов
	приложения, но, если экземпляр переключается на другое оборудование, данные не перемещаются с экземпляром.	двоичным объектом. Если на VM, к которой подключен диск, происходит сбой, диск можно подключить к другой VM со всеми данными, которые были сохранены в хранилище больших двоичных объектов.	долговременное хранение структурированных данных.	масштабируемое и долговременное хранение неструктурированных объектов, таких как изображения, аудио- и видеофайлы.
Доступ к данным	API файловой системы.  Доступ к локальному хранилищу можно получить с помощью интерфейсов API файловой системы. Таким образом, приложение можно запустить	API файловой системы.  Доступ к диску Windows Azure можно получить с помощью интерфейсов API файловой системы. Таким образом,	API-интерфейс REST или клиентская библиотека хранилища  Хранилище таблиц доступно из любого места и из любого клиента с помощью API-интерфейса REST. Также	API-интерфейс REST или клиентская библиотека хранилища  Хранилище больших двоичных объектов доступно из любого места и из любого клиента с помощью API-



Критерии сравнения	Локальное хранилище	Диск Azure	Хранилище таблиц	Хранилище больших двоичных объектов
	на платформе Azure с минимальными изменениями кода.	приложение можно запустить на платформе Azure с минимальными изменениями кода.	можно получить доступ к хранилищу таблиц с помощью клиентских библиотек хранилища, которые предоставляют оболочки REST API для различных языков (например, .NET, Java, Node.js и PHP).	интерфейса REST. Также можно получить доступ к хранилищу больших двоичных объектов с помощью клиентских библиотек хранилища, которые предоставляют оболочки REST API для различных языков (например, .NET, Java, Node.js и PHP).
Параллелизм	Нет.  Локальное хранилище доступно только из одного экземпляра приложения. Не используется совместно с другими экземплярами.	Да, но с ограничениями.  Только у одного экземпляра есть доступ к диску Azure на чтение и запись в любой момент времени, но	Да.  Хранилище таблиц совместно используется всеми приложениями, использующими API-интерфейс REST для доступа к хранилищу. Одновременный доступ к	Да.  Хранилище больших двоичных объектов совместно используется всеми приложениями, использующими API-интерфейс REST для доступа к хранилищу.

Критерии сравнения	Локальное хранилище	Диск Azure	Хранилище таблиц	Хранилище больших двоичных объектов
		несколько других экземпляров могут получить доступ только для чтения.	хранилищу таблиц поддерживается с помощью <a href="#">ETags</a> .	Одновременный доступ к хранилищу больших двоичных объектов поддерживается с помощью <a href="#">ETags</a> .
Цены	Необходима учетная запись Windows Azure Compute.  Локальное хранилище включено в стоимость учетной записи Azure Compute счета и ограничено размером вычислительной операции. Дополнительная учетная запись хранения не требуется.	Необходима учетная запись хранения Windows Azure. Обновленные сведения о ценах см. в разделе <a href="#">Сведения о ценах Windows Azure</a> .	Для хранилища таблиц требуется учетная запись хранения Windows Azure.	Для хранилища больших двоичных объектов требуется учетная запись хранения Windows Azure.
Задержка (доступ из экземпляра Windows Azure)	Локальное хранилище находится на ВМ, поэтому	Медленнее локального хранилища, так как	Медленнее локального хранилища, так как данные	Медленнее локального хранилища, так как данные

Критерии сравнения	Локальное хранилище	Диск Azure	Хранилище таблиц	Хранилище больших двоичных объектов
Compute)	доступ к нему быстрее по сравнению с доступом к диску Azure.	данные хранятся не на виртуальных машинах, а в хранилище больших двоичных объектов. Задержка увеличивается, если диск Azure расположен не в том же центре обработки данных, что и экземпляр Compute.	хранятся не на VM. Задержка увеличивается, если хранилище таблиц расположено не в том же центре обработки данных, что и экземпляр роли или VM, обращающиеся к нему.	хранятся не на VM. Задержка увеличивается, если хранилище больших двоичных объектов расположено не в том же центре обработки данных, что и экземпляры роли, виртуальные машины или компьютеры, обращающиеся к нему.
Масштабируемость	Нет  Только один экземпляр приложения может получить доступ к локальному хранилищу. Следовательно, оно не обеспечивает какую-либо	Да, но с ограничениям и.  Только у одного экземпляра приложения может быть доступ к записи к диску Azure, но несколько	Да.  Система хранения Windows Azure автоматически распределяет разделы по всем узлам хранилища на основе шаблонов использования	Да.  Хранилище больших двоичных объектов Azure поддерживает крупную масштабируемую систему распределения больших двоичных

Критерии сравнения	Локальное хранилище	Диск Azure	Хранилище таблиц	Хранилище больших двоичных объектов
	масштабируемо сть.	экземпляров приложения могут получить доступ на чтение.	разделов. Например, если к определенным секциям имеется много обращений, то система автоматически распределит их на отдельные узлы хранения, таким образом нагрузка распределится на несколько серверов.	объектов с помощью <a href="#">сети</a> <a href="#">кэширующих серверов</a> Windows Azure, в которой большие двоичные объекты обслуживаются многими серверами для масштабирования и обработки трафика приложения. Более того, система имеет высокий уровень доступности и надежности.
Высокий уровень доступности и отказоустойчивость	Нет	Да  Диск Azure поддерживает хранилищем больших двоичных объектов, которое обеспечивает высокий уровень	Да.  Большие двоичные объекты, таблицы и очереди, хранящиеся на Windows Azure, реплицируются в трех местах одного центра обработки	Да.  Большие двоичные объекты, таблицы и очереди, хранящиеся на Windows Azure, реплицируются в трех местах одного центра обработки данных для обеспечения

Критерии сравнения	Локальное хранилище	Диск Azure	Хранилище таблиц	Хранилище больших двоичных объектов
		доступности.	данных для обеспечения устойчивости против сбоев оборудования. Кроме того, данные реплицируются в разные домены отказоустойчивости для увеличения доступности, как и в случае со всеми остальными службами хранилища Azure.	устойчивости против сбоев оборудования. Кроме того, данные реплицируются в разные домены отказоустойчивости и для увеличения доступности, как и в случае со всеми остальными службами хранилища Azure.
Аварийное восстановление	Нет	Да  Диск Azure поддерживает хранилищем больших двоичных объектов, которое обеспечивает аварийное восстановление	Да.  Большие двоичные объекты и таблицы Windows Azure также географически реплицируются между двумя географически разделенными	Да.  Большие двоичные объекты и таблицы Windows Azure также географически реплицируются между двумя географически разделенными центрами

Критерии сравнения	Локальное хранилище	Диск Azure	Хранилище таблиц	Хранилище больших двоичных объектов
		ие.	центрами обработки данных, обеспечивая таким образом дополнительную долговечность данных в случае серьезной аварии.	обработки данных, обеспечивая таким образом дополнительную долговечность данных в случае серьезной аварии.
Безопасность	Можно получить доступ только из виртуальной машины, на которой существует.	Только у одного экземпляра приложения может быть доступ на запись к диску Azure, но несколько экземпляров приложения могут получить доступ на чтение.	Каждый запрос, выполняемый к службам хранения Windows Azure, должен пройти проверку подлинности, кроме случаев анонимного запроса к ресурсу общедоступного контейнера. Дополнительные сведения см. в разделе <a href="#">Аутентификация доступа к учетной записи хранилища</a> .	Каждый запрос, выполняемый к службам хранения Windows Azure, должен пройти проверку подлинности, кроме случаев анонимного запроса к ресурсу общедоступного контейнера. Дополнительные сведения см. в разделе <a href="#">Аутентификация доступа к учетной записи хранилища</a> .

Некоторые из сценариев, в которых можно использовать службы управления данными Windows Azure:

- Используйте службу для обеспечения другого расположения аварийного восстановления для локальных данных.
- Совместно используйте часть локальных данных с партнерами без изменения локальной инфраструктуры.
- Переместите данные ближе к вычислительным узлам в облаке.
- Обработывайте пиковые нагрузки доступа к известным заранее данным путем переноса данных в облако, масштабирования и предоставления доступа к ним.

## В этом разделе

- [Перенос данных в хранилище таблиц](#)
- [Перенос данных в хранилище больших двоичных объектов](#)
- [Перенос данных на диски](#)

## Перенос данных в хранилище таблиц

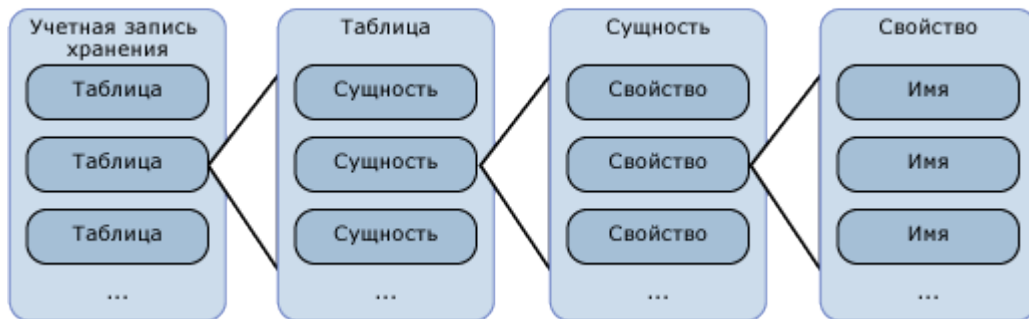
---

**Авторы:** Шридхар Пеллур (Sreedhar Pelluru)

**Соавторы:** Джеймс Подгорскиани (James Podgorski)

**Рецензенты:** Валерий Мизонов (Valery Mizonov), Кун Чен (Kun Cheng), Стив Ховард (Steve Howard)

Хранилище таблиц Azure — это нереляционное структурированное хранилище с высокой степенью масштабируемости в облаке. Таблица Azure представляет собой набор сущностей (строк). Сущность может иметь до 255 свойства (столбцов), где каждое свойство имеет атрибуты имени, типа и значения. Каждая сущность в таблице имеет три зарезервированных свойства: PartitionKey, RowKey и Timestamp. Хранилище таблиц использует ключ секции для секционирования и распределения сущностей по узлам хранения Azure. Секция хранилища таблиц содержит сущности с одинаковым ключом секции. Ключ строки уникально определяет сущность в пределах секции. Отметка времени — системное свойство только для чтения, используемое для отслеживания изменений. В хранилище таблиц не требуется определение схемы сущностей в таблице. Таблица может содержать сущности с разным набором свойств. Подробные общие сведения о хранилище таблиц приведены на [портале Windows Azure](#).



## Вопросы миграции

При миграции приложений, использующих хранилище таблиц Azure, учитывайте следующие факторы.

- Какой тип данных можно хранить в хранилище таблиц?
- Каким образом перенесенное приложение может получить доступ к данным в хранилище таблиц?
- Поддерживает ли хранилище высокий уровень доступности, масштабируемость, аварийное восстановление и требования безопасности перенесенного приложения?
- Как передать существующие данные в хранилище таблиц?

## Вопросы данных

Первый шаг процесса миграции заключается в определении того, подходит ли хранилище таблиц для хранения данных, которые использует локальное приложение. Хранилище таблиц оптимизировано для хранения данных, соответствующих следующим требованиям.

- Данные **структурированы** и обычно хранятся в табличном формате.
- Это **нереляционные** данные. Данные, которые планируется хранить в одной таблице, не связаны с данными в других таблицах. Хранилище таблиц Azure не поддерживает хранение реляционных данных с применением таких механизмов, как обеспечение ссылочной целостности данных в базах данных. Хранилище таблиц оптимизировано для хранения нереляционных структурированных данных.
- Обработка данных **на стороне сервера** не требуется. Убедитесь в том, что для данных не требуется выполнение операций на стороне сервера, таких как соединения, хранимые процедуры, триггеры, поддерживающих реляционную базу данных. В хранилище таблиц не поддерживается обработка данных на стороне сервера. Поддерживаются базовые операции, такие как Insert, Update, Delete и Select, с простой фильтрацией на стороне сервера по свойствам PartitionKey и RowKey.



- Данные в основном **индексируются**; поиск в них выполняется с использованием значения подстановки или ключа. Ключ секции определяет секцию, а ключ строки определяет уникальную строку в пределах секции. Ключ секции и ключ строки вместе уникально определяют сущность в таблице. После выяснения того, что хранилище таблиц хорошо подходит для хранения данных, оцените, какая часть может использоваться в качестве ключа секции для таблицы. См. раздел [Проектирование масштабируемой стратегии секционирования для хранилища таблиц Windows Azure](#).
- Данные могут храниться с использованием **типов данных**, поддерживаемых хранилищем таблиц. [В хранилище таблиц поддерживаются следующие типы данных](#): String, Byte Array, GUID, DateTime, Int32, Int64, Double и Boolean.
- Для данных не требуются **межсекционные транзакции**. Хранилище таблиц не поддерживает распределенные или межсекционные транзакции. Поддерживаются только [транзакции групп сущностей](#). Поэтому важно выбрать правильный ключ секции и хранить взаимосвязанные данные в одной секции. Например, может понадобиться хранить сведения о клиенте и его заказах в одной секции, чтобы можно было одновременно обновлять данные по клиенту и заказам одной транзакцией, что обеспечивает целостность данных.
- **Размер данных** может достигать порядка гигабайтов и терабайтов (необязательно). Максимальный размер таблицы Azure — 100 ТБ, это предел размера для хранилища Windows Azure (включая таблицы, большие двоичные объекты и очереди).
- Данные, хранимые в **одной строке**, могут отличаться по структуре и типу от данных, хранимых в **другой строке той же таблицы** (необязательно). В хранилище таблиц Windows Azure не требуется определять фиксированную схему для строк или сущностей. Таким образом, в одной таблице можно хранить различные типы данных. Например, можно хранить сведения о заказе в одной строке и информацию о клиенте в другой строке той же таблицы.

## Сравнение хранилища таблиц и базы данных SQL Windows Azure

В хранилище таблиц размещаются структурированные данные, так же как и в базе данных SQL Windows Azure. Поэтому при миграции приложений с локальной платформы на платформу Windows Azure часто возникает вопрос, использовать хранилище таблиц или базу данных SQL. Главное различие между базой данных SQL и хранилищем таблиц таково: база данных SQL Database является системой управления реляционными базами данных, предоставляющей функции обработки данных в виде соединений, представлений и хранимых процедур. В свою же очередь хранилище таблиц Windows Azure не является реляционным хранилищем данных и не предоставляет средств обработки данных, которые поддерживает база данных SQL.

Если приложение хранит и извлекает большие наборы данных, но ему не требуются функции обработки данных, то хранилище таблиц Windows Azure является лучшим

выбором. Если приложению требуется выполнять обработку больших наборов данных, реляционных по своей природе, то лучше выбрать базу данных SQL. Также перед совершением выбора между базой данных SQL и хранилищем таблиц Azure следует рассмотреть несколько других факторов. В таблице темы [Общие сведения о службах управления данными в Windows Azure](#) приведено более подробное сравнительное описание указанных методов хранения.

## Вопросы доступа к данным

Клиентские приложения могут осуществлять доступ к хранилищу таблиц Windows Azure посредством HTTP API-интерфейса REST независимо от языка программирования и операционной системы. К хранилищу таблиц можно получить доступ посредством клиентских библиотек, разработанных для определенных ОС и языков программирования. Есть библиотеки для .NET, Node.js, Java и PHP, которые доступны для загрузки в [центре разработчиков Windows Azure](#). Например, клиентская библиотека хранилища .NET обеспечивает строго типизированные оболочки .NET вокруг API-интерфейса REST для облегчения работы разработчиков .NET.

Если существующее локальное приложение использует структурированные, но не реляционные данные и планируется использовать хранилище таблиц для хранения этих данных на платформе Windows Azure, нужно переписать часть кода, который обращается к данным с помощью клиентской библиотеки хранилища.

## Преимущества использования хранилища таблиц

При хранении данных в хранилище таблиц Azure можно автоматически пользоваться несколькими важными преимуществами, такими как следующие:

- **Масштабируемость:** система хранения Windows Azure автоматически распределяет секции по всем узлам хранения на основе шаблонов секций. Например, если к определенным секциям имеется много обращений, то система автоматически распределит их на отдельные узлы хранения, таким образом нагрузка распределится на несколько серверов.
- **Высокий уровень доступности, устойчивость к сбоям:** таблицы в Windows Azure хранятся в трех реплицированных копиях в одном центре обработки данных, что обеспечивает устойчивость к сбоям оборудования. Независимо от используемой службы хранилища данные реплицируются на разные домены ошибок, что увеличивает степень доступности данных.
- **Аварийное восстановление:** таблицы Windows Azure также географически реплицируются между двумя центрами обработки данных на одном континенте, обеспечивая таким образом дополнительную долговечность данных в случае серьезной аварии.

- **Безопасность:** каждый запрос, выполняемый к службам хранения Windows Azure, должен пройти проверку подлинности, кроме случаев анонимного запроса к ресурсу общедоступного контейнера. Дополнительные сведения см. в разделе [Аутентификация доступа к учетной записи хранилища](#).
- **Доступ к данным независимо от клиента и места:** доступ к хранилищу таблиц осуществляется с помощью API-интерфейса REST по HTTP(S)-протоколу. Таким образом, любое клиентское приложение, выполняемое на любой ОС, может обращаться к хранилищу таблиц посредством REST.

## Отправка существующих данных в хранилище таблиц Azure

После перепроектирования приложения для использования преимуществ нереляционного хранилища таблиц с высокой степенью масштабируемости может потребоваться перенести существующие данные из файловой системы или базы данных SQL Server в хранилище таблиц. Для этого можно написать собственный код с использованием HTTP(S) API-интерфейса REST или клиентской библиотеки .NET либо использовать один из следующих инструментов:

- [Программа передачи файлов Azure](#). С помощью данного инструмента можно передать данные из разделенного неструктурированного файла в хранилище таблиц Azure.
- [Программа передачи базы данных Azure](#). С помощью данного инструмента можно передать данные из базы данных SQL Server в хранилище таблиц Azure.
- [Cloud Storage Studio компании Red Gate Software](#). Сторонние инструменты, подобные этому, позволяют управлять облачными хранилищами Azure, включая хранилище таблиц.

## См. также

- [Как наиболее эффективно использовать таблицы Windows Azure](#).

## Перенос данных в хранилище больших двоичных объектов

---

**Авторы:** Шридхар Пеллuru (Sreedhar Pelluru)

**Соавторы:** Джеймс Подгорскиани (James Podgorskiani)

**Рецензенты:** Кристиан Мартинес (Christian Martinez), Валерий Мизонов (Valery Mizonov), Чэн Кун (Kun Cheng), Стив Ховард (Steve Howard)

Служба больших двоичных объектов Windows Azure позволяет приложениям хранить большое количество неструктурированных текстовых или двоичных данных, таких как

файлы видео, аудио и изображений. Хранилище больших двоичных объектов содержит нуль или более контейнеров больших двоичных объектов, а контейнер содержит нуль или более больших двоичных объектов. Большой двоичный объект — это любая сущность, состоящая из двоичных данных, таких как файл или изображение.



Служба хранения предлагает два типа больших двоичных объектов: блочные большие двоичные объекты и страничные большие двоичные объекты.

- **Блочный большой двоичный объект** состоит из блоков, каждый из которых обозначается идентификатором блока. Создать или изменить блочный большой двоичный объект можно, написав набор (или список) блоков и зафиксировав их с использованием идентификаторов блоков. Каждый блок может быть разного размера. Максимальный размер — 4 МБ. Максимальный размер для блочного большого двоичного объекта составляет 200 ГБ, также блочный большой двоичный объект может включать не более 50 000 блоков. Блочные большие двоичные объекты позволяют вставлять, удалять и переупорядочивать блоки и одновременно передавать несколько блоков большого двоичного объекта. Он предназначен для эффективной загрузки и передачи больших двоичных объектов. Блочные большие двоичные объекты имеет смысл использовать, если приложение хранит большие файлы, к которым могут одновременно обращаться несколько читателей.
- **Страничные большие двоичные объекты** представляют собой коллекцию страниц в 512 байт, оптимизированных для операций случайного чтения и записи. Каждая страница в страничном большом двоичном объекте упоминается с использованием смещения от начала большого двоичного объекта. Для добавления или обновления содержимого страничного большого двоичного объекта необходимо написать страницы, задав смещение и диапазон, соответствующий границам 512-байтных страниц. Операция записи в страничный большой двоичный объект может перезаписать одну или несколько страниц — до 4 МБ страничного большого двоичного объекта. Операция записи в страничный большой двоичный объект производится на месте и немедленно фиксируется в страничном большом двоичном объекте. Максимальный размер большого двоичного объекта составляет 1 ТБ, кроме того, размер большого двоичного объекта должен быть кратен 512 байтам.

Подробные общие сведения о хранилище больших двоичных объектов см. на [портале Windows Azure](#).

# Сравнение блочных и страничных больших двоичных объектов

Блочные большие двоичные объекты позволяют эффективно передавать большие двоичные объекты размерами до 200 ГБ. Они оптимизированы функциями, которые помогут вам управлять большими файлами по сети. Одной из таких функций является возможность передавать и скачивать несколько блоков одновременно и определять их последовательность в момент фиксации. С другой стороны, страничные большие двоичные объекты оптимизированы для случайных операций чтения и записи в ситуациях, когда страницы выровнены по границам в 512 байт.

Ниже приведены некоторые из сценариев, где используются страничные большие двоичные объекты.

- Приложение, которое получает доступ к файлам с обновлениями в диапазоне. Приложение обрабатывает страничный большой двоичный объект как файл и использует операции записи в диапазоне для обновления изменившихся частей большого двоичного объекта. Для обновления страничных больших двоичных объектов можно получить монопольный доступ для записи.
- Настраиваемое протоколирование приложений, которое обрабатывает страничный большой двоичный объект как кольцевой буфер. Когда страничный большой двоичный объект заполняется, приложение может начать запись данных с самого начала структуры большого двоичного объекта.

## Вопросы миграции

Рассмотрим различные факторы, например те, которые следует учитывать при миграции приложений для использования хранилища больших двоичных объектов Windows Azure.

- Какие виды данных могут храниться в хранилище больших двоичных объектов?
- Как можно осуществлять доступ к данным, хранящимся в хранилище больших двоичных объектов, из перенесенных приложений?
- Поддерживает ли хранилище высокий уровень доступности, масштабируемость, аварийное восстановление и требования безопасности перенесенного приложения?
- Как можно передать существующие данные в хранилище больших двоичных объектов?

## Вопросы данных

Перед повторным проектированием приложения, чтобы в нем могло использоваться хранилище больших двоичных объектов, следует оценить, хорошо ли подходит хранилище

больших двоичных объектов для хранения соответствующих данных. Хранилище больших двоичных объектов предназначено для хранения **больших** объемов **неструктурированных текстовых или двоичных** данных, таких как документы, изображения, аудио и видео.

Хранилище больших двоичных объектов может также использоваться для хранения стандартных или двоичных файлов, от которых зависит приложение. При хранении зависимых файлов в большом двоичном объекте вы можете обновлять зависимые файлы без обновления или передачи всего файла пакета приложения (CSPKG-файла). Это также позволяет иметь различные версии зависимых файлов в отдельных больших двоичных объектах, а приложениям динамически загружать зависимые файлы, специфичные для определенной версии.

## **Сравнение хранилища больших двоичных объектов и базы данных SQL Windows Azure**

База данных SQL Windows Azure поддерживает тип данных `varbinary(max)` для поддержки хранения больших объектов в базе данных. Если приложение хранит и получает доступ к большим двоичным объектам, таким как изображения, аудио и видео в базе данных SQL Server, необходимо определить, следует ли использовать базу данных SQL или хранилище больших двоичных объектов при миграции приложения на платформу Windows Azure.

Если используется атрибут `FILESTREAM` на столбце `varbinary` для хранения файлов размером больше 2 ГБ в базе данных SQL Server, стоит рассмотреть вопрос об использовании хранилища больших двоичных объектов при миграции на платформу Windows Azure, поскольку база данных SQL на данный момент не поддерживает `FILESTREAM`. Даже если размер файла меньше 2 ГБ и функция `FILESTREAM` системы SQL Server не используется, имеет смысл рассмотреть возможность использования хранилища больших двоичных объектов, потому что в зависимости от характера приложения это может оказаться дешевле и обеспечить большие возможности масштабирования и потому что оно может быть доступно любым клиентам посредством использования API-интерфейса REST.

После того как большие объекты были сохранены в хранилище больших двоичных объектов, можно сохранить ссылку на большой двоичный объект в столбце в таблице экземпляра базы данных SQL. Максимальный размер экземпляра базы данных SQL в настоящее время составляет 150 ГБ. Поэтому, если большие объекты хранятся в экземпляре базы данных SQL, может не хватить пространства. Максимальный размер хранилища больших двоичных объектов составляет 100 ТБ, что также составляет максимальный размер для хранилища Windows Azure. Максимальный размер каждого большого двоичного объекта в хранилище больших двоичных объектов составляет 200 ГБ (блочные большие двоичные объекты) или 1 ТБ (страничные большие двоичные объекты).

Например, при миграции локального веб-приложения, которое имеет графические ресурсы, такие как изображения, можно хранить URL-адрес изображения в базе данных SQL (или хранилище таблиц) и настроить клиентскую программу на получение URL-адреса и отображения изображения по этому URL-адресу.



Производительность приложения может быть снижена при перемещении больших двоичных объектов из базы данных SQL и хранении только ссылок на большие двоичные объекты в хранилище больших двоичных объектов, поскольку клиентское приложение в первую очередь создает запрос к экземпляру базы данных SQL для определения местоположения больших двоичных объектов и только затем запрос к хранилищу больших двоичных объектов для получения больших двоичных данных, например изображений или крупных объектов. Следует учитывать, что невозможно выполнить одновременное резервное копирование и восстановление данных из базы данных SQL и хранилища больших двоичных объектов, поэтому согласованность на уровне транзакций резервных копий базы данных SQL и хранилища больших двоичных объектов не гарантируется.

Также следует учитывать количество транзакций, которые выполняет приложение в хранилище данных. За транзакции с базой данных SQL не взимается дополнительная плата, тогда как за транзакции в хранилище Windows Azure взимается. Данные, доступ к которым осуществляется менее часто, могут быть хорошим кандидатом для хранения в хранилище Windows Azure, тогда как часто используемые данные более выгодно хранить в базе данных SQL.

## Вопросы доступа к данным

Клиентские приложения, написанные на любом языке программирования и работающие в любой операционной системе, могут получить доступ к хранилищу больших двоичных объектов Window Azure с помощью HTTP(S) API-интерфейса REST. Доступ к хранилищу больших двоичных объектов также можно получить с помощью клиентских библиотек, предназначенных для конкретных операционных систем и языков программирования.

Библиотеки для .NET, Node.js, Java и PHP доступны в [центре разработчиков Windows Azure](#). Например, клиентская библиотека хранилища .NET обеспечивает строго типизированные оболочки .NET вокруг API-интерфейса REST для облегчения работы разработчиков .NET.

Если будет решено хранить неструктурированные данные, используемые приложением, в хранилище больших двоичных объектов на платформе Windows Azure, необходимо переписать часть кода, которая обращается к данным с помощью библиотеки Storage Client Library.

## Преимущества использования хранилища больших двоичных объектов

При хранении данных в хранилище больших двоичных объектов Windows Azure можно автоматически получить несколько важных преимуществ, например следующие:

- **Масштабируемость.** Хранилище больших двоичных объектов Azure поддерживает высокомасштабируемую систему распределения больших двоичных объектов посредством [CDN](#) Windows Azure. Сеть CDN обслуживает горячие, наиболее часто используемые большие двоичные объекты с многих серверов для обеспечения масштабирования и удовлетворения потребностей трафика приложения. Более того, система имеет высокий уровень доступности и надежности.
- **Высокий уровень доступности/отказоустойчивость.** Большие двоичные объекты, хранящиеся на платформе Windows Azure, реплицируются в три местоположения в одном центре обработки данных для обеспечения устойчивости против сбоев оборудования. Кроме того, данные реплицируются в разные домены отказоустойчивости для увеличения **доступности**, как и в случае со всеми остальными службами хранилища Azure.
- **Аварийное восстановление.** Большие двоичные объекты в Windows Azure также реплицируются в два географически разделенных центра обработки данных на одном континенте, обеспечивая таким образом дополнительную долговечность данных в случае серьезной аварии.
- **Безопасность.** Каждый запрос, выполняемый к службам хранения Windows Azure, должен пройти проверку подлинности, кроме случаев анонимного запроса к ресурсу общедоступного контейнера. Дополнительные сведения см. в разделе [Аутентификация доступа к учетной записи хранилища](#).
- **Доступ к данным от любого клиента, в любом месте.** Доступ к хранилищу больших двоичных объектов Windows Azure можно получить с помощью API-интерфейса REST через HTTP. Любое клиентское приложение в любой операционной системе может получить доступ к хранилищу больших двоичных объектов с помощью интерфейса REST.



# Перенос существующих данных в хранилище больших двоичных объектов Azure

После переработки приложения для использования высокомасштабируемого хранилища больших двоичных объектов может потребоваться перенести данные из базы данных SQL Server или файловой системы. Чтобы сделать это, можно написать код, используя HTTP(S) API-интерфейс REST или клиентскую библиотеку .NET для хранилища больших двоичных объектов, или использовать инструменты, такие как [Cloud Storage Studio компании Red Gate Software](#).

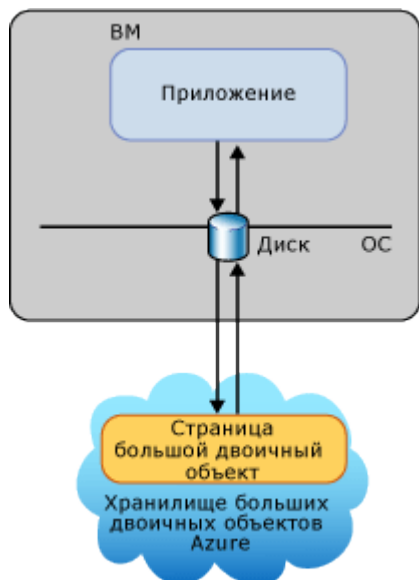
## Перенос данных на диски

---

**Авторы:** Сридхар Пеллур (Sreedhar Pelluru)

**Рецензенты:** Валерий Мизонов (Valery Mizonov), Кун Чен (Kun Cheng), Стив Ховард (Steve Howard)

Диск Windows Azure является страничным большим двоичным объектом, который содержит виртуальный жесткий диск (VHD) с форматом NTFS. Можно создать VHD в компьютере, передать его в хранилище больших двоичных объектов или использовать API-интерфейс REST, чтобы на платформе Windows Azure был создан большой двоичный объект, который будет подключен как диск Windows Azure к виртуальной машине (связанной с вычислительным узлом), выполняющей код приложения. Виртуальная машина имеет драйвер устройства для системы, которая принимает запросы ввода-вывода файловой системы и преобразует их в операции в VHD. При сбое машины тот же код может выполнять другая машина, которая подключит VHD, получая доступ к данным, которые были предварительно сохранены.



Подробный обзор дисков Windows Azure см. в разделе [Технический документ по дискам Windows Azure](#).

## Вопросы миграции

Приложения Windows Azure в облаке могут использовать существующие API-интерфейсы NTFS для доступа к диску Windows Azure. Это облегчает миграцию локальных приложений, которые используют API-интерфейс NTFS (или стандартный API-интерфейс .NET Framework, такой как FileStream) для сохранения данных и доступа к ним в файловой системе платформы Azure с минимальными изменениями в коде.

Важно обратить внимание на то, что в определенный момент времени только одна машина может подключить диск Windows Azure для записи. Поэтому эта возможность не предназначена для приложений, которые горизонтально масштабируются в несколько экземпляров, которые требуют доступа для записи. Однако несколько виртуальных машин могут одновременно подключить версию большого двоичного объекта только для чтения (моментальный снимок).

Если требуется надежная длительная сохранность данных, нужно совместно использовать данные в экземплярах или нужно иметь доступ вне Windows Azure, вместо диска Windows Azure используйте службу таблиц Windows Azure, службу больших двоичных объектов или базу данных SQL Windows Azure. Если нужно совместно использовать данные в экземплярах, при этом только один экземпляр должен иметь доступ к данным для записи, рассмотрите использование дисков Windows Azure. Подробное сравнение между этими

хранилищами см. в таблице сравнения [Перенос данных в другие службы управления данными в Windows Azure](#).

## Передача данных в диск Windows Azure

Можно создать VHD со всеми данными, передать этот диск в страничный большой двоичный объект и подключить этот объект как диск к виртуальной машине (связанной с вычислительным узлом), в которой размещен экземпляр приложения. Облегчить этот процесс может следующее средство.

- Служебная программа передачи VHD из [набора обучения платформы Windows Azure](#) служит для передачи файла VHD в страничный большой двоичный объект. Исходный код для этого средства находится в папке <WindowsAzurePlatformTrainingKit\Folder>\Labs\ExploringWindowsAzureStorageVS2010\Source\Assets\VHDUpload.
- [Azure Drive Explorer](#) позволяет несложно управлять дисками Windows Azure.
- [Cloud Storage Studio компании Red Gate Software](#). Функциональные возможности этого ПО позволяют создавать пустые страничные большие двоичные объекты или передавать существующие жесткие диски (VHD) из компьютера как страничные большие двоичные объекты, которые впоследствии можно подключить как диски Windows Azure.

## См. также

[Технический документ по дискам Windows Azure](#)

## Вопросы миграции в службу кэша Windows Azure

---

**Авторы:** Джейми Альва Браво (Jaime Alva Bravo), Сридхар Пелуру (Sreedhar Pelluru)

**Соавторы:** Джейсон Рот (Jason Roth)

Служба кэширования предоставляет распределенный, хранимый в памяти кэш для приложений Windows Azure. Кэширование повышает производительность благодаря временному хранению данных из других источников в памяти. Приложения обращаются к данным в кэше намного быстрее, чем в хранилищах внутреннего сервера, таких как база данных SQL Windows Azure. Это также снижает расходы, связанные с транзакциями баз данных в облаке. Дополнительные сведения см. в разделе [Документация по службе кэширования Windows Azure](#).

## Вопросы миграции

Если локальное приложение использует кэш Windows Server, то переадресовать его на использование кэша Windows Azure будет просто. Если локальное приложение не использует кэш Windows Server, реструктурируйте его на использование кэша Windows Azure. В следующей таблице описаны типы данных, наиболее подходящих для хранения в кэше Windows Azure.

Тип данных	Описание
Эталон	<p>Эталонные данные — это версия исходных данных, которые не меняются или изменяются редко. Это либо прямая копия, либо исходные данные, либо агрегированные и преобразованные данные из нескольких источников. Эталонные данные периодически обновляются, обычно в заданные интервалы или при изменении данных.</p> <p>Так как эталонные данные нечасто меняются, они хорошо подходят для хранения в кэше. Эталонные данные могут сохраняться в кэше и использоваться в последующих запросах, что способствует экономии вычислительных ресурсов, поскольку не приходится повторно выполнять агрегирование и преобразовывать эталонные данные каждый раз, когда они потребуются. Кэширование эталонных данных среди нескольких приложений или пользователей таким образом увеличивает производительность приложения и дает возможности масштабирования.</p> <p>Примерами эталонных данных служат расписания полетов и каталоги продукции. Например, есть приложение каталога, которое собирает сведения о продуктах</p>

Тип данных	Описание
	<p>среди нескольких приложений и источников данных. Самая частая выполняемая операция над данными каталога — это общее чтение, или просмотр каталога. Подобная операция проходит по множеству данных о продуктах, фильтрует их, персонализирует и представляет выбранные данные нескольким пользователям.</p>
Действия	<p>Данные о действиях формируются как часть бизнес-транзакции при выполнении действия в приложении. Данные являются частью бизнес-транзакции. По завершении транзакции данные сохраняются в источнике данных как часть сведений журнала или истории.</p> <p>Сюда можно отнести данные о заказах на покупку, состоянии сеансов приложения или покупки в сети. Рассмотрим данные о корзине покупок в приложении магазина в сети. Каждая корзина покупок является монопольной для каждого сеанса покупок в сети и содержит индивидуальный набор данных. Во время сеанса покупок корзина кэшируется и обновляется выбранными продуктами. Корзина видима и доступна только для транзакции покупки. При выписке чека корзина убирается из кэша в источник данных для дополнительной обработки. По завершении транзакции исходного приложения данные о корзине покупок сохраняются в журнале для аудита и отслеживания истории покупок.</p> <p>Пока активен сеанс покупок, сведения о корзине доступны для чтения и записи, но эти данные не находятся в общем доступе.</p>

Тип данных	Описание
	Монопольный доступ и отношение к данным о действиях позволяет сделать эти данные кандидатом для распределенного кэширования.
Ресурс	<p>И эталонные данные (общее чтение), и данные действий (монопольная запись) идеально подходят для кэширования. Однако не все данные приложения относятся к этим двум категориям. Данные определенного типа, характеризующиеся тем, что они являются общими, обеспечивают параллельное чтение и запись в операциях, а доступ к ним предоставляется во многих транзакциях. Это данные ресурсов.</p> <p>К примерам данных ресурсов относятся учетные записи пользователей и элементы аукционов. Рассмотрим лот на аукционе. Для лота есть описание и сведения о текущей ставке (кто сделал ставку и сколько). Подобные сведения часто изменяются, уникальны для каждой ставки, к ним одновременно обращаются множество пользователей. Как показано в этом образце, данные ресурсов являются кандидатами для кэширования, потому что бизнес-логика, работающая с данными ресурсов, также может быть кэширована.</p>

Если локальное приложение использует кэш Windows Server, имейте в виду некоторые различия между этим кэшем и службой кэширования Windows Azure и внесите соответствующие изменения в приложение. См. статью [Различия между локальным и облачным кэшированием](#)

Самое большое различие — служба кэширования Azure поддерживает распределенное кэширование в памяти без необходимости настройки соответствующей инфраструктуры. При работе на локальной платформе нужно получать машины, ставить на каждую из них Windows Server, создавать кластер кэша и управлять им на этих машинах. В облачном решении платформа Windows Azure сама выполняет подобные задачи по администрированию кэша. Управление кэшем выполняется в сети, что предоставляет необходимые сведения о безопасности и соединения, необходимые для использования кэша.

Доступны различные [по цене](#) предложения использования кэша Azure, которые зависят от общего требуемого размера кэша, объема передаваемых данных и одновременных соединений. Эти ограничения введены из-за особенностей общего использования службы. Приобретение услуг по любому предложению равносильно приобретению возможности получения доступа к части общей памяти, выделенной для кэширующего кластера, поэтому принудительно применяются эти квоты (лимитирующие объем памяти, передачу данных и количество параллельных соединений), чтобы обеспечить равноправное использование услуг каждым клиентом.

В локальной среде кэширующий кластер Windows Server AppFabric обычно выделяется отдельно для подобных нужд. В локальном сценарии требуется планировать емкость кластера, нужно понять, какие требуются для него характеристики, чтобы избежать чрезмерного использования ресурсов кластера (пропускной способности, соединений и памяти). Даже если кластер перегружен, он будет обслуживать запросы (хоть и не так эффективно). Но при достижении подобных пределов в Azure служба перестанет быть доступной до завершения часа выбора квоты. Например, если квота была превышена в 13:55, то служба будет недоступной до 14:00 (по времени центра обработки данных, в котором приобретены функции службы). См. следующие две статьи о планировании емкости в MSDN: [Руководство по планированию емкости кэша Windows Server AppFabric Caching](#) и [Руководство по планированию емкости Windows Azure](#).

## Windows Azure Caching (Preview)

Предварительная технология кэширования Windows Azure предоставляет новую возможность размещать службу кэширования в ролях Azure. Такую функциональность также называют кэшированием на основе ролей. Есть две основные топологии развертывания подобного вида кэширования: выделенная и совместно размещаемая. В выделенной топологии задается рабочая роль, отведенная под кэширование. Вся доступная память рабочей роли используется для кэширования и сопутствующих операций обработки данных. В совместно размещаемой топологии часть доступной памяти ролей приложения используется для кэширования. Например, можно выделить 20 % физической

памяти каждого экземпляра веб-роли для кэширования. В обоих случаях пользователь платит только за вычислительные ресурсы, необходимые для выполнения экземпляров ролей. Дополнительные сведения см. в разделе [Обзор предварительной технологии кэширования Windows Azure](#).

В этой предварительной версии кэш распределяется по всем экземплярам ролей, размещающих кэш. Основное отличие от общего кэша в том, что кэширование на основе ролей доступно только для размещенной службы. Общий кэш используется как мультитенантная служба. В результате для кэша на основе ролей нет квот на используемые ресурсы. Размер кэша определяется несколькими переменными. Размер зависит от количества выполняемых экземпляров ролей, для которых включен кэш. Также он зависит от размера виртуальных машин и процента памяти роли, отведенного под кэш.

Предварительная технология кэша Windows Azure имеют следующие преимущества над общим кэшем Windows Azure.

- Пользователь платит только за вычислительные экземпляры, размещающие кэш. Нет отдельных расходов на кэш.
- При масштабировании ролей размер кэша меняется автоматически. При добавлении или удалении ролей размер памяти, доступной для кэша, также соответственно меняется.
- Благодаря резервным копиям обеспечивается высокий уровень доступности. Если одна роль перестает работать, резервная копия элементов в кэше по-прежнему остается доступной.
- Производительность выше, так как кэширующий кластер выделен для облачной службы и расположен ближе к экземплярам ролей.
- Поддерживаются уведомления, ключевые слова, работа с регионами.
- Можно создавать дополнительные именованные кэши.
- Поддерживается протокол Memcache.

Возможно выполнить миграцию с Windows Azure Shared Caching на Azure Caching (Preview) с минимальными изменениями в коде. См. статью [Миграция с Windows Azure Shared Caching на Azure Caching \(Preview\)](#).



### **Важно!**

В данный момент технология Windows Azure Caching (Preview) доступна для ознакомления. Рабочая версия не поддерживается, но разработчикам рекомендуется оценить преимущества данной новой модели кэша.



## См. также

[Часто задаваемый вопрос о технологии Windows Azure Caching \(Preview\)](#)

[Как использовать технологию Windows Azure Caching \(Preview\)](#)

[Вопросы, учитываемые при планировании емкости кэша для Windows Azure Caching \(Preview\)](#)

## Миграция приложений, которые используют технологии обмена сообщениями

---

**Авторы:** Кун Чен (Kun Cheng)

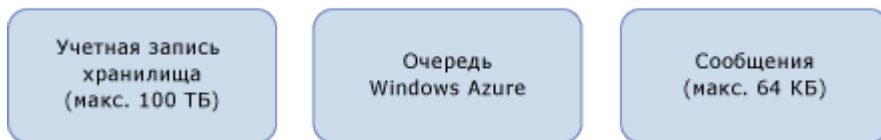
**Соавторы:** Сридхар Пелуру (Sreedhar Pelluru), Валерий Мизонов (Valery Mizonov), Кристиан Мартинес (Christian Martinez), Рама Рамани (Rama Ramani)

**Рецензенты:** Стив Ховард (Steve Howard)

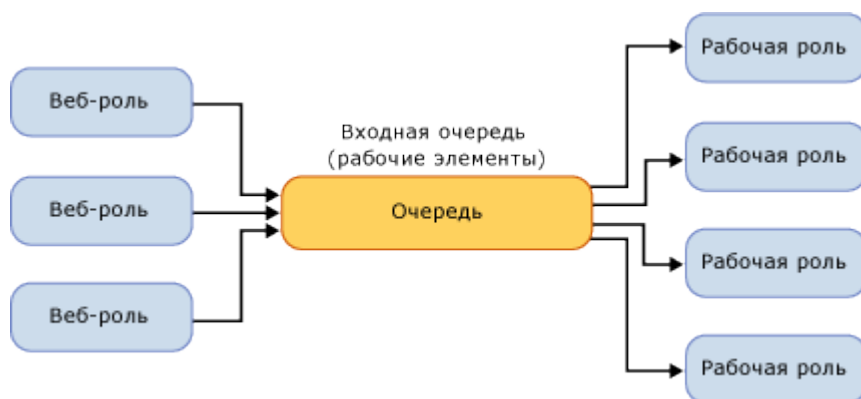
Очереди сообщений обычно используются для взаимодействия приложений или для обмена данными между компонентами одного приложения. При проектировании приложений архитекторы и разработчики могут использовать очереди сообщений для организации асинхронного коммуникационного канала. Очередь сообщений позволяет отправителям передать сообщения и продолжить выполнение других задач без ожидания ответа от получателей. Получатели получают и обрабатывают сообщения независимо, без блокировки отправителей. Этот механизм помогает разграничить компоненты приложения, которые в противном случае были бы тесно связаны друг с другом, позволяя построить более гибкое и масштабируемое решение.

При переносе приложения из локальной платформы в Azure архитекторам и разработчикам рекомендуется изучить текущую архитектуру и определить возможности использования очередей Windows Azure и шины обслуживания, чтобы воспользоваться преимуществами слабосвязанной архитектуры и масштабирования на платформе Windows Azure.

Очереди Windows Azure основаны на хранилище Windows Azure и предоставляют базовый механизм для поддержки связи «точка-точка». Механизм очередей поддерживает доступ на основе протоколов REST HTTP или HTTPS. Каждая очередь сообщений поддерживает емкость до 100 ТБ (ограничение текущей учетной записи хранения). Каждое сообщение может быть размером до 64 КБ. Дополнительные сведения см. [в данной статье](#).



Наиболее распространенное применение очередей Windows Azure — веб-роль, отправляющая рабочие элементы, и рабочая роль, асинхронно принимающая и обрабатывающая сообщения.



Платформа Windows Azure также позволяет использовать обмен сообщениями на основе очередей с помощью шины обслуживания Windows Azure. В дополнение к использованию очередей шина обслуживания предоставляет функции безопасного обмена сообщениями Windows Azure, а также возможности ретрансляции для поддержки распределенных приложений в Azure или гибридного развертывания приложений на локальной платформе и в Azure. Для обеспечения механизма обмена сообщениями шина обслуживания поддерживает оба вида соединения «точка-точка» через [очереди шины обслуживания](#) и модели подписок ([темы и подписки шины обслуживания](#)). Темы и подписки шины обслуживания позволяют нескольким подписчикам одновременно прослушивать одного издателя. [Возможности по ретрансляции](#) позволяют использовать сценарий гибридного решения, где ресурсы организации находятся на локальной или частной облачной платформе и могут быть расширены для взаимодействия с облачными ресурсами. Шина обслуживания поддерживает доступ по протоколам TCP или HTTP/HTTPS на основе REST. Каждая очередь шины обслуживания имеет емкость до 5 ГБ. Каждое сообщение может быть размером до 256 КБ.

Есть много различий между очередью Windows Azure и очередью шины обслуживания. Сюда относятся функции аутентификации, поддержки транзакций и интеграции с WCF.

Подробное сравнение подобных очередей приведено в статье [Сравнение очередей обслуживания Windows Azure](#).

## Миграция MSMQ

Обычно приложения Windows в качестве механизма очередей используют службу очередей сообщений (Майкрософт) (MSMQ). Это позволяет приложениям, которые выполняются на отдельных серверах в отдельных процессах, взаимодействовать друг с другом в надежной, слабосвязанной манере. Он также позволяет приложениям, которые находятся в разнородных сетевых средах, обмениваться данными, даже когда они не в сети. Механизмы обеспечивают гарантированную доставку сообщений, поддержку распределенных транзакций, эффективную маршрутизацию, безопасность и обмен сообщениями на основе приоритетов.

При миграции приложений, которые используют технологию MSMQ, на платформу Windows Azure имейте в виду, что на данный момент Windows Azure не поддерживает эту технологию. При миграции необходимо изменить код для использования очередей Windows Azure. В остальной части этого подраздела описаны различные варианты переноса приложений, использующих MSMQ, на платформу Windows Azure.

## Шина обслуживания Windows Azure

Во многих отношениях шина обслуживания — самая близкая функция Windows Azure по отношению к MSMQ. Они имеют много схожих функций, в том числе основную очередь операций, поддержку транзакций и недоставленные сообщения. Однако для использование шины обслуживания нужны другие API-интерфейсы с отличной семантикой. В следующем списке приведены несколько ключевых отличий с точки зрения размера и производительности.

- Сообщения шины обслуживания ограничены размером до 256 КБ (заголовок и текст), сообщения MSMQ могут быть размером до 4 МБ.
- Очереди шины обслуживания ограничены размером до 5 ГБ. Размер очереди MSMQ ограничивается настраиваемыми квотами или ресурсами компьютерного оборудования.
- Пропускная способность очереди шины службы может достигать до 2000 сообщений в секунду, тогда как для MSMQ этот параметр может достигать свыше 6000 сообщений в секунду (на основе тестов с 1000 сообщений). Дополнительные сведения см. в техническом документе [Оптимизация производительности в среде сервера очереди сообщений Microsoft](#).

Для облегчения миграции возможно установить связь между MSMQ на локальной платформе и шиной обслуживания на платформе Azure через мост. Образец кода представлен [здесь](#).

## Очередь Windows Azure

Очереди Windows Azure обеспечивает базовый коммуникационный канал «точка-точка». Разнородные среды, как в среде MSMQ, не поддерживаются. Кроме того, не поддерживаются функции, доступные в типичной среде MSMQ, такие как автоматическое удаление устаревших сообщений, транзакции и приоритет очередности. Однако разработчики приложений могут реализовать необходимые функции на основе очереди Windows Azure для реализации функций, подобных MSMQ. Для этого потребуется изменить приложение.

## Рабочая роль Windows Azure

В Windows Server встроена поддержка MSMQ, поэтому запуск MSMQ на том же узле в качестве рабочей роли обеспечивает полную функциональность локальной версии MSMQ. Однако рабочая роль должна зависеть от механизма отработки отказа и обслуживания. При отказе теряются и становятся невозможными все сведения о состоянии, такие как сообщения, хранящиеся в локальном хранилище MSMQ. Если служба MSMQ не использовалась без сохранения состояния, а приложение не было разработано с учетом возможной отработки отказа для роли, то не рекомендуется эксплуатировать MSMQ в экземпляре рабочей роли.

# Миграция приложений для использования локального хранилища

---

**Авторы:** Сридхар Пеллур (Sreedhar Pelluru)

**Рецензенты:** Валерий Мизонов (Valery Mizonov), Кун Чен (Kun Cheng), Стив Ховард (Steve Howard)

Локальное хранилище предоставляется как часть Windows Azure Compute и обеспечивает временное хранилище для запуска экземпляра приложения. При запуске приложения в Windows Azure оно размещается на виртуальной машине (ВМ), к которой подключен виртуальный жесткий диск. Локальное хранилище представляет собой каталог в файловой системе на этом жестком диске.

Можно создать несколько локальных хранилищ для каждого экземпляра. Размер локального хранилища по умолчанию составляет 1 МБ. Размер хранилища можно увеличить до максимального, допускаемого вычислительной операцией. Максимальное место на диске для вычислительной операции зависит от размера ВМ, выбранной для экземпляра.

## Вопросы миграции

Приложения Windows Azure в облаке могут использовать существующие API-интерфейсы NTFS для доступа к локальному хранилищу. Это облегчает миграцию локальных приложений, которые используют API-интерфейс NTFS (или стандартный API-интерфейс .NET Framework, такой как FileStream) для сохранения временных данных и доступа к ним в файловой системе платформы Azure с минимальными изменениями в коде.

Важно отметить, что локальное хранилище на ВМ доступно только для локальных экземпляров приложения на виртуальной машине. Его можно настроить для сохранения, если веб-роль или рабочая роль экземпляра перерабатывается. Но это относится только к простой переработке роли. Если экземпляр перезапускается на другом оборудовании, например как в случае сбоя или обслуживания оборудования, данные в локальном хранилище не перемещаются вместе с экземпляром, даже если он был настроен для сохранения после переработки.

Если требуется надежная длительная сохранность данных, нужно совместно использовать данные в экземплярах или нужно иметь доступ вне Windows Azure, вместо локального хранилища используйте службу таблиц Windows Azure, службу больших двоичных объектов или базу данных SQL Windows Azure. Если нужно совместно использовать данные в экземплярах, при этом только один экземпляр должен иметь доступ к данным для записи, рассмотрите использование дисков Windows Azure. Подробное сравнение между этими хранилищами см. в таблице сравнения [Перенос данных в другие службы управления данными в Windows Azure](#).

## См. также

[Настройка размера виртуальной машины](#)

[Настройка ресурсов локального хранилища](#)