

Создание эластичных и устойчивых облачных приложений Руководство разработчика по пакету интеграции Enterprise Library для Windows Azure

patterns & practices

Справочник\руководство



Microsoft

# Создание эластичных и устойчивых облачных приложений: Руководство разработчика по пакету интеграции Enterprise Library для Windows Azure

patterns & practices

**Обзор:** Эта книга демонстрирует, как можно использовать пакет интеграции Enterprise Library для Windows Azure в существующем приложении Windows Azure, чтобы улучшить надежность, управляемость, масштабируемость, стабильность и расширяемость приложения. В частности рассматриваются вопросы автоматического масштабирования в Windows Azure (Autoscaling Application Block), обработка ошибок исключений и минимизация их влияние на компоненты системы (Transient Fault Handling Application Block) и т.п.

**Категория:** Справочник\руководство

**Аудитория:** Windows Azure

**Источник:** [patterns & practices](#)

**Дата публикации электронной книги:** Сентябрь 2012

Copyright © 2012 by Microsoft Corporation

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.

## Содержание

Предисловие .....	10
Введение.....	14
Для кого предназначена эта книга .....	14
Почему эта книга важна сейчас .....	15
Структура этой книги.....	16
Что нужно для использования кода.....	16
Кто есть кто .....	17
Благодарности .....	19
Группа, создавшая это руководство .....	19
Группа разработки пакета интеграции Enterprise Library для Windows Azure .....	19
Советники.....	19
Консультативный совет.....	19
Сообщество .....	20
1. Введение в Windows Azure .....	21
Сведения о платформе Windows Azure .....	21
Службы и компоненты Windows Azure.....	3
Вычислительная среда.....	3
Управление данными.....	4
Сетевые службы .....	5
Другие службы.....	7
Разработка приложений для Windows Azure .....	7
Мониторинг, отладка приложений Windows .....	8
Azure и управление ими .....	8
Управление базами данных SQL Azure .....	9

Обновление приложений для Windows Azure.....	9
Модель подписок и выставления счетов Windows Azure.....	10
Оценка затрат .....	12
Дополнительные сведения .....	12
2    Описание пакета интеграции Enterprise Library Integration Pack для Windows Azure .....	19
Что такое библиотека Enterprise Library.....	19
Описание пакета интеграции Enterprise Library для Windows Azure .....	22
Дополнительные сведения .....	23
3    Сценарий Tailspin.....	24
Компании Tailspin.....	24
Стратегия компании Tailspin .....	24
Приложение Surveys.....	24
Цели и задачи, стоящие перед компанией Tailspin .....	26
Архитектура приложения Surveys .....	27
4    Автоматическое масштабирование и Windows Azure .....	29
Что такое автоматическое масштабирование .....	29
Что такое программный блок автоматического масштабирования .....	32
Масштабирование экземпляров .....	34
Регулирование приложения .....	35
Правила и действия .....	36
Ведение журнала .....	43
Жизненный цикл автомасштабирования.....	43
Когда следует использовать блок автомасштабирования приложения? .....	45
Требуется, чтобы приложение автоматически отвечало на изменение в потребности .....	45
Необходимо управлять издержками, связанными с эксплуатацией конкретного приложения .....	45

Значения времени, в которые приложению потребуются дополнительные ресурсы, являются прогнозируемыми .....	46
Обстоятельства, в которых не следует использовать блок автомасштабирования приложения.....	46
Использование блока автомасштабирования приложения .....	47
Добавление блока автомасштабирования приложения в проект Visual Studio .....	49
Размещение блока автомасштабирования приложения.....	50
Изменения в приложении Windows Azure.....	51
Служебная информация.....	52
Добавление функции регулирования в приложение .....	54
Правила автомасштабирования .....	55
Мониторинг блока автомасштабирования приложения .....	58
Дополнительные сценарии использования.....	60
Образец параметров конфигурации.....	67
Использование средства «Планирование» .....	72
Как работает блок автомасштабирования приложения .....	73
Метроном.....	74
Сборщики данных .....	74
Хранилище служебной информации.....	75
Хранилище точек данных .....	75
Средство оценки правил.....	75
Хранилище правил .....	75
Средство ведения журнала .....	76
Средство масштабирования .....	76
Объект отслеживания .....	76
Дополнительные сведения.....	77
5    Создание более эластичного приложения Tailspin Surveys.....	82
Предпосылки .....	82

Цели и требования .....	84
Общие сведения о решении автомасштабирования.....	86
Использование блока автомасштабирования прило- жения в Tailspin Surveys .	86
Функции блока автомасштабирования приложения .....	86
Группы масштабирования в Tailspin Surveys .....	89
Правила автомасштабирования в Tailspin Surveys .....	89
Сбор данных для журнала автомасштабирования в Tailspin Surveys....	92
Пользовательский интерфейс конфигурации авто- масштабирования.....	93
Уведомление операторов через SMS о выполнении операций масштабирования.....	93
Процесс реализации .....	93
Включение функции чтения из CSCFG-файла для программного блока автомасштабирования .....	94
Определение служебных сведений Tailspin.....	94
Правила автомасштабирования Tailspin.....	96
Сбор в Tailspin Surveys данных для счетчика производительности .....	102
Реализация поведения регулирования .....	105
Редактирование и сохранение правил .....	107
Редактирование и сохранение сведений о службе.....	111
Визуализация действий автомасштабирования.....	111
Реализация пользовательского действия .....	114
Реализация пользовательских операндов .....	119
Настройка ведения журналов в Tailspin Surveys .....	124
Установка и физическое развертывание .....	126
Сертификаты и развертывание Tailspin Surveys .....	126
<i>Развертывание сертификата службы для включения SSL</i> .....	126
Развертывание Tailspin Surveys в нескольких географических районах	128
Дополнительные сведения .....	132

6	Обработка временного сбоя .....	29
	Что такое временные сбои? .....	29
	Что такое программный блок обработки времен- ных сбоев? .....	134
	Историческая справка .....	135
	Использование программного блока обработки временных сбоев .....	136
	Добавление программного блока обработки времен- ных сбоев к проекту Visual Studio. ....	136
	Создание экземпляров объектов программного блока обработки временных сбоев .....	137
	Определение стратегии повтора .....	137
	Определение правил повторов .....	138
	Выполнение операции с правилами повторов .....	139
	Когда следует использовать программный блок обработки временных сбоев? .....	140
	Вы используете службу Windows Azure .....	140
	Вы используете пользовательску ю службу .....	140
	Дополнительные сведения .....	141
7	Увеличение надежности приложения Tailspin Surveys .....	143
	Предпосылки .....	143
	Цели и требования .....	144
	Общие сведения о решении Transient Fault .....	144
	Handling Application Block .....	144
	Процесс реализации .....	145
	Установка и физическое развертывание .....	148
	Дополнительные сведения .....	148
	Приложение А    Образцы конфигураций Развертывания приложения Tailspin Surveys в нескольких центрах обработки данных .....	149
	Вариант 1 .....	149
	Модель службы .....	149



Вариант 2 .....	154
Модель службы .....	155
Rules .....	156
Приложение Б    Руководство по установке приложения Tailspin Surveys .....	158
Введение.....	158
Предварительные требования.....	160
Установка исходного кода и зависимостей .....	160
УСТАНОВКА ИСХОДНОГО КОДА ПРИЛОЖЕНИЯ TAILSPIN SURVEYS И ЗАВИСИМОСТЕЙ .....	161
Подготовка подписки Windows Azure для развертывания .....	162
Создание сертификата управления Windows Azure и его экспорт как PFX- файла.....	162
СОЗДАНИЕ СЕРТИФИКАТА УПРАВЛЕНИЯ WINDOWS AZURE И ЕГО ИМПОРТ ДЛЯ ИСПОЛЬЗОВАНИЯ В СРЕДЕ VISUAL STUDIO.....	163
ЭКСПОРТ СЕРТИФИКАТА УПРАВЛЕНИЯ WINDOWS AZURE В PFX-ФАЙЛ .....	164
Создание SSL-сертификата .....	165
Создание необходимых размещенных служб.....	167
Передача сертификатов как относящихся к размещенным службам .....	169
ДОБАВЛЕНИЕ СЕРТИФИКАТОВ .....	169
Создание учетной записи хранилища .....	170
Построение решения .....	172
Установка пакетов NuGet.....	172
Изменение сертификатов в облачных проектах Visual Studio .....	173
Подготовка параметров в облачных проектах.....	175
Построение решения .....	177
Развертывание в Windows Azure.....	178
Развертывание решения Tailspin в промежуточном слоте.....	178
Тестирование работы Tailspin Surveys.....	180
Настройка функции автоматического масштабирования .....	181

Tailspin .....	181
Настройка хранилища служебной информации.....	181
Загрузка образцов правил .....	182
Локальный запуск Tailspin Surveys в режиме отладки .....	182
Запуск приложения управления в режиме эмуляции .....	183
Известные проблемы .....	184
Дополнительные сведения .....	185
Приложение В.....	186
Глоссарий .....	186

## Предисловие

Потребление электроэнергии в ИТ-секторе растет быстрее, чем в любой другой отрасли, по мере того, как общество становится все более зависимым от вычислительных ресурсов и возможностей хранения данных, предоставляемых центрами обработки данных. К сожалению, сочетание неэффективного оборудования, устаревших рекомендаций по эксплуатации и отсутствия стимулов приводит к тому, что значительная часть энергии, используемой в традиционных центрах обработки данных, расходуется впустую.

Большинство усилий по обеспечению энергосбережения были сосредоточены на физической инфраструктуре — развертывание более экономичного компьютерного оборудования и систем охлаждения, использование функций автоматического управления питанием операционной системы и сокращение количества серверов в центрах обработки данных с помощью виртуализации оборудования.

Но значительный объем расходуемой энергии связан с тем, как приложения разрабатываются и используются. Большинство приложений получают больше ИТ-ресурсов, чем действительно требуется, для обеспечения приемлемой производительности и защиты от сбоев оборудования. Чаще всего фактические потребности приложения просто никогда не измеряются, не анализируются и не пересматриваются.

Пусть даже приложение было развернуто с большим объемом ресурсов, чем фактически требуется, у разработчиков приложения практически нет стимулов для его переработки с целью упрощения планирования загрузки. А когда пользователи начинают жаловаться на то, что приложение работает медленно, часто легче (и дешевле) просто назначить приложению дополнительные ресурсы. В дальнейшем эти ресурсы удаляются очень редко, даже после того, как спрос на них уменьшается.

Облачные вычисления позволяют разорвать этот порочный круг. Облачные платформы, такие как Windows Azure, выставляют счета за использование ресурсов в небольших приращениях (вычислительных часах) по объему фактического применения ресурсов, поэтому разра-

ботчики теперь могут напрямую влиять на ИТ-расходы и связанное с ними использование ресурсов.

Приложения, которые предназначены для динамического увеличения и уменьшения использования ресурсов в ответ на фактический и предполагаемый спрос, не только являются менее ресурсозатратными, но и значительно более эффективными при использовании ИТ-ресурсов, чем традиционные приложения. Разработчики также могут снизить расходы на размещение, запланировав выполнение фоновых задач в периоды меньшей занятости, когда приложению назначено минимальное количество ресурсов.

Хотя облачные системы предоставляют большие возможности для экономии на размещении, при разработке приложений на основе облачных служб приходится преодолевать определенные трудности. Одной из проблем, которые встают перед разработчиками, являются «временные ошибки». Хотя они возникают довольно редко, приложения должны быть устойчивыми к периодической потере возможности подключения и проблемам с реакцией, чтобы они считались надежными и обеспечивали удобства для пользователей.

До сих пор разработчики для Windows Azure должны были создавать эти функции самостоятельно. После выпуска *пакета интеграции Enterprise Library для Windows Azure* разработчики могут легко создавать надежные приложения, эффективно использующие ресурсы, которые можно интеллектуально масштабировать и регулировать. Кроме того, эти приложения могут справляться с временными ошибками.

Первый основной компонент в пакете интеграции — *программный блок автоматического масштабирования*, известный также как «WASABi». Этот блок помогает разработчикам улучшить оперативность и управление затратами Windows Azure, автоматически масштабируя количество рабочих и веб-ролей в Windows Azure с помощью динамического выделения ресурсов и вывода из эксплуатации экземпляров ролей в различных размещенных службах. WASABi также предоставляет механизмы для управления использованием ресурсов без масштабирования экземпляров ролей с помощью регулирования приложений. Разработчики могут использовать этот блок, чтобы планировать или откладывать фоновую обработку, поддерживая количество экземпляров ролей в определенных границах и используя преимущества периодов простоя.

Одним из главных преимуществ WASABi является расширяемость, обладая которой, решения становятся гораздо более эластичными. Придерживаясь принципов проектирования других программных блоков, WASABi предоставляет механизм для подключения собственных пользовательских показателей и вызова настраиваемых действий. Это позволяет создать набор правил, учитывающих конкретные бизнес-сценарии, а не только стандартные счетчики производительности, доступные в системе диагностики Windows Azure.

Стабилизатор оптимизации гарантирует, что степень масштабирования не будет изменяться слишком быстро. Он также может гарантировать, что операции масштабирования будут соответствовать наиболее оптимальным тарифам на машинное время. Для прило-

жений, которые могут использовать множество экземпляров, этот блок поможет сэкономить на размещении, при этом улучшится «экологичность» хранения данных приложений. Это также поможет приложению выполнить требования соглашений об уровне обслуживания (SLA).

Другим важным компонентом является *программный блок обработки временных ошибок* (известный как «Тораз»). Он помогает разработчикам делать свои приложения более надежными, предоставляя логику для обнаружения и обработки временных ошибок для целого ряда облачных услуг.

Разработчики больше, чем когда-либо, могут играть важную роль в области контроля ИТ-затрат и улучшения энергосбережения без ущерба для надежности. Пакет интеграции Enterprise Library для Windows Azure позволяет быстро создавать надежные приложения на основе Windows Azure, экономичные и эффективно использующие ресурсы.

Руководство разработчика, которое вы держите в руках, написано группой специалистов, которые разработали этот пакет интеграции. В данном документе представлено множество полезных рекомендаций и советов, которые помогут вам быстро освоиться. Важно отметить, что руководство не только охватывает основные разделы, но и содержит описания конкретных действий, с помощью которых реализация образцового приложения Tailspin Surveys становится более эластичой, надежной и устойчивой.

Это руководство, созданное группой Microsoft patterns & practices, используется также в руководстве разработчика и образце реализации. С пакетом поставляются исходный код и все модульные тесты, так что на его основе можно многому научиться.

Я настоятельно рекомендую пакет интеграции Enterprise Library для Windows Azure и это руководство разработчика специалистам по архитектуре, разработчикам программного обеспечения, администраторам и владельцам продукта, которые создают новые или переносят существующие приложения в Windows Azure. Практические советы в этой книге помогут сделать приложения лучше масштабируемыми и более надежными.

Марк Аггар (Mark Aggar), старший директор  
Экологическая устойчивость  
Корпорация Майкрософт

## Введение

Технологическая платформа Windows Azure™ предлагает компаниям и разработчикам потрясающие новые возможности для создания масштабных и сложных облачных приложений. Windows Azure позволяет при создании инфраструктуры приложений воспользоваться преимуществами модели оплаты по объему фактического использования ресурсов и предоставления вычислительных ресурсов по требованию.

Объединяя существующие программные блоки Microsoft® Enterprise Library, которые помогают разрабатывать надежные, настраиваемые и простые в управлении приложения, с новыми блоками, специально разработанными для облачных систем, можно создавать масштабируемые устойчивые приложения, способные использовать все преимущества Windows Azure.

В этой книге описывается сценарий, основанный на вымышленной компании Tailspin, которая решила расширить существующее размещенное приложение для Windows Azure, используя новый программный блок автоматического масштабирования и блок обработки временных ошибок. Это приложение на основе Windows Azure с именем Surveys подробно описано в предыдущей книге из этой серии «Разработка приложений для облака», доступной по адресу <http://msdn.microsoft.com/en-us/library/ff966499.aspx>.

Это руководство сопровождает пример реализации. Мы рекомендуем изучить его и поэкспериментировать с ним, чтобы лучше понять, как действуют новые программные блоки.

В дополнение к описанию приложения Windows Azure и того, как оно использует блоки библиотеки Enterprise Library, в этой книге описываются основные возможности блоков и общие рекомендации по их использованию в собственных приложениях.

После прочтения этой книги вы получите представление о том, как встроить программный блок автоматического масштабирования и блок обработки временных ошибок в приложения Windows Azure.

### Для кого предназначена эта книга

Эта книга демонстрирует, как можно использовать пакет интеграции Enterprise Library для Windows Azure в существующем приложении Windows Azure, чтобы улучшить надежность, управляемость, масштабируемость, стабильность и расширяемость приложения. Данная книга предназначена для всех специалистов по архитектуре, разработчиков и ИТ-специалистов, которые проектируют, создают или используют облачные приложения и службы и хотят научиться реализовывать преимущества применения Enterprise Library в облачных приложениях. Чтобы извлечь наибольшую пользу из этого руковод-

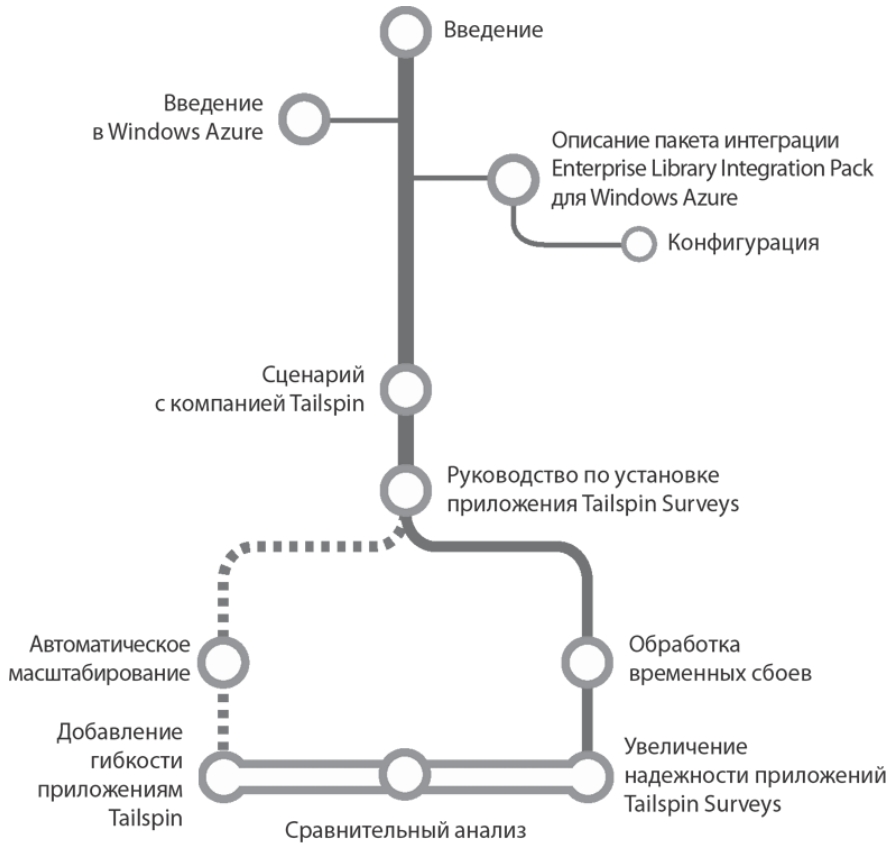
ства, вы должны быть знакомы с платформой Windows Azure, платформой Microsoft .NET Framework, средой разработки Microsoft Visual Studio®, ASP.NET и Microsoft Visual C#®. В следующих двух главах представлены общие сведения о Windows Azure и пакете интеграции Enterprise Library для Windows Azure, которые помогут вам приступить к работе.

## Почему эта книга важна сейчас

В целом облако стало важным средством, позволяющим предоставить доступ к приложениям широкому кругу пользователей. Возможно, вы уже создавали и развертывали приложения в Windows Azure с помощью средств, доступных для Visual Studio, и пакета Windows Azure SDK для .NET. Так же как и Enterprise Library помогает решать общие повседневные задачи, такие как ведение журнала и управление исключениями в локальных приложениях, так и пакет интеграции и соответствующее руководство помогают решать общие задачи для многих облачных приложений. Некоторые из этих задач совпадают с задачами в локальных приложениях, например управление исключениями и кэширование. Некоторые, такие как автоматическое масштабирование для удовлетворения эластичного спроса, будут отличаться. В этой книге показано, как можно решить эти задачи в контексте общего сценария: улучшение существующего приложения для Windows Azure.



## Структура этой книги



## Что нужно для использования кода

Для запуска приложения Tailspin Surveys необходимо следующее.

- Компьютер для ведения разработки, на котором эксплуатируется система разработки Microsoft Visual Studio® 2010 с пакетом обновления 1 (SP1).
- Все необходимые обновления Microsoft Windows®.
- Диспетчер пакетов NuGet (дополнительные сведения см. по адресу <http://nuget.codeplex.com/>).
- Подписка Windows Azure с пространством для двух размещенных служб (если нужно запустить приложение Tailspin Surveys и компонент Autoscaler Host и веб-приложение управления в Windows Azure).
- Учетная запись хранилища Windows Azure.

- Средство проверки зависимостей, которое будет проверять, установлены ли необходимые компоненты, перечисленные ниже. Если некоторых компонентов нет, это средство поможет их установить.
  - Visual Studio 2010.
  - MVC 3 Framework.
  - Пакет SDK Windows Azure для .NET и средств Windows Azure для ноябрьского выпуска Microsoft Visual Studio 2011 года.
  - Среда выполнения Windows Identity Foundation.
  - Необязательно: службы IIS 7 — они необходимы, если вы хотите запускать сайт управления в режиме эмуляции или развернуть роль автомасштабирования локально.

## Кто есть кто

Как упоминалось ранее, в этой книге используются сценарии, которые демонстрируют, как использовать пакет интеграции Enterprise Library для Windows Azure в примере реализации. Процесс разработки комментирует группа экспертов. Группа включает в себя специалиста по облачным технологиям, бизнес-руководителя, специалиста по архитектуре программного обеспечения, разработчика программного обеспечения, знакомого с Enterprise Library, разработчика программного обеспечения, не знакомого с Enterprise Library, и ИТ-специалиста. Сценарии можно рассматривать с каждой из этих точек зрения. В следующей таблице перечислены эксперты для этих сценариев.



Барат — специалист по облачным технологиям. Он проверяет, будет ли облачное решение работать в компании и получит ли компания ощутимые преимущества. Он осторожный человек, и это из благих целей.

Создать одно приложение для облака довольно просто. Реализовать преимущества облачного решения — не всегда так легко.

Яна — специалист по архитектуре программного обеспечения. Она планирует общую структуру приложения. Ее перспективы как практические, так и стратегические. Другими словами, она учитывает не только то, какие технические подходы необходимы сегодня, но и какие направления должна рассматривать компания в будущем. Яна работала на многих проектах, использующих Enterprise Library.



Сбалансировать потребности компании, пользователей, ИТ-организации, разработчиков и технических платформ, которые мы используем, нелегко.



Маркус — разработчик программного обеспечения, не знакомый с Enterprise Library. У него аналитическое мышление, он внимателен к деталям и методичен. Он ориентирован на поставленную задачу — построение качественного облачного приложения. Он знает, что он в конечном счете отвечает за код.

Какую бы платформу вы ни хотели использовать для приложения, я сделаю так, чтобы оно работало.

Эд — опытный разработчик программного обеспечения и эксперт по Enterprise Library. Как настоящий профессионал, он хорошо осведомлен об общих трудностях, с которыми сталкиваются разработчики при построении бизнес-приложений (LOB) для предприятия. В прошлом он создавал собственные библиотеки для решения этих задач, но в последние несколько лет использовал библиотеку Enterprise Library для большинства из этих приложений.



Наши приложения используют библиотеку Enterprise Library для решения общих задач. Это обеспечивает унифицированность во всех наших системах, что облегчает их поддержку и обслуживание. Мы инвестировали в наши локальные приложения много средств, и нам необходима возможность повторного использования этих инвестиций в облаке.



По — ИТ-специалист и эксперт в развертывании и запуске приложений в облаке. По интересуется практическими решениями. В конце концов, именно ему звонят в 3 утра при возникновении проблемы.

При миграции приложений в облако возникают трудности, которые отличаются от управления локальными приложениями. Я хочу сделать так, чтобы наши облачные приложения были такими же надежными и безопасными, как и наши локальные приложения.

Бет — бизнес-менеджер. Она помогает компаниям планировать развитие их бизнеса. Она знакома с рынком, на котором работает компания, знает ресурсы, доступные для компании, а также цели компании. Она обладает стратегическим мышлением и интересуется повседневной деятельностью компании.



Организации сталкиваются со многими конфликтующими требованиями к их ресурсам. Я хочу убедиться, что наша компания сбалансирует эти требования и внедрит бизнес-план, который позволит нам добиться успеха в среднесрочной и долгосрочной перспективе.

Если вас интересует конкретная область, найдите комментарии специалистов, чьи интересы соответствуют вашим.

## Благодарности

### Группа, создавшая это руководство

<b>Авторы</b>	Доминик Беттс, Жереми Бурго, Джулиан Домингес, Эрсенк Керестеси, Григорий Мельник, Фернандо Симонацци и Эрвин ван дер Валк
<b>Технические рецензенты</b>	Нил Маккензи, Валерий Мизонов, Эудженио Пасе, Павел Вилкош и Майкл Вуд
<b>Художник-оформитель</b>	Кэти Нимер
<b>Редакторы</b>	Роанн Корбизье, Нелли Дельгадо и Нэнси Мичелл

### Группа разработки пакета интеграции Enterprise Library для Windows Azure

<b>Продукт/программа Управление</b>	Григорий Мельник (корпорация Майкрософт)
<b>Архитектура/ Разработка</b>	Джулиан Домингес (корпорация Майкрософт), Фернандо Симонацци (Clarius Consulting), Жереми Бурго (nVentive Inc.) и Эрсенк Керестеси (Full Scale 180 Inc)
<b>Тестирование</b>	Мани Кришнасвами и Карлос Фарре (корпорация Майкрософт), Нираж Джаин, Муругеш Мутувилаван, Картик Натарджан, Тамилараси Натарадж и Рати Велусами (Infosys Technologies Ltd.)
<b>Пользовательский интерфейс/ Документация</b>	Доминик Беттс (Content Master Ltd.), Рик Карп (DCB Software Testing, Inc.), Нелли Дельгадо (корпорация Майкрософт) и Эрвин ван дер Валк (Эрвин ван дер Валк)
<b>Управление выпусками</b>	Ричард Берт (ChannelCatalyst.com, Inc.), Григорий Мельник (корпорация Майкрософт) и Жереми Бурго (nVentive Inc.)
<b>Административная поддержка</b>	Керстин Скотт (корпорация Майкрософт)

### Советники

Многие люди помогли созданию этого выпуска. Мы благодарны им всем!

### Консультативный совет

Николай Блэки (Adaptiv), Иван Бодягин (ABBY), Федерико Бор (Southworks), Леандро Боффи (Tellago), Майкл Колье (Independent), Ханс де Грааф (New Day At Work), Джейсон де Оливейра (Cargemini Sogeti), Дэйв Феллоуз (GreenButton), Штефан Филип, (HPC Consulting), Сэм Фолд (Arkadium), Нуну Филипе Годинью (Independent), Нил Маккензи (Independent), Даниэль Пизенс (Zywave), Марк Реммерс (42windmills), Кейт Стоби (Independent), Франсуа Тангей (nVentive),

Михай Татаран (HPC Consulting), Стас Ткаченко (Arkadium), Трент Свенсон (Full Scale 180), Габриэль Шлетчманн (3mellons), Филипп Виалатт (Independent), Гвидо Вилариньо (3mellons/Disney), Олег Волков (New Day at Work), Павел Вилкош (Motorola Solution Systems), Майкл Вуд (Cumulux Inc.) и наши коллеги из Майкрософт: Марк Аггар, Дэвид Айкен, Кашиф Алам, Хайме Альва Браво, Джихан Бийикоглы, Джим Дэвис, Джон Доусон, Скотт Денсмор, Ленни Фенстер, Дэнни Гарбер, Рик Хайнс, Том Холландер, Дмитрий Мартынов, Стив Маркс, Тони Мелег, Сурен Махираджу, Аде Миллер, Валерий Мизонов, Уолтер Майерс, Масаси Нарумото, Бхушан Нене, Эудженио Пасе, Курт Петерсон, Джеймс Подгорски, Шрилата Райасам, Паоло Сальватори, Марк Швайгерт, Марк Симмс, Эрик Слипперн, Виджей Сен и Тим Виман.

Вклад Валерия Мизонова и всей консультационной группы AppFabric (CAT) заслуживает отдельного упоминания. Программный блок обработки временных ошибок основан на стратегиях обнаружения и повтора, первоначально разработанных группой AppFabric CAT. Улучшенный программный блок обработки временных ошибок является результатом сотрудничества группы шаблонов и рекомендаций и группы AppFabric CAT.

Мы также хотели бы подчеркнуть наше сотрудничество с консультационными службами Майкрософт (Ленни Фенстер, Дэнни Гарбер, Рик Хайнс, Уолтер Майерс и Марк Швайгерт), чьи многочисленные отзывы помогли нам остаться в рамках реальных сценариев. Благодарим вас за то, что не позволили нам пропустить многие важные сценарии.

## Сообщество

Большое спасибо Рахулу Рай, Судхару Редди Д. В., Маартену Бальяу, Зойнер Техада и всем тем, кто оценивал проект и участвовал в бета-тестировании первых примеров кода, а также предоставил полезные отзывы. Кроме того, мы благодарим участников симпозиума по шаблонам и рекомендациям, конференций TechReady и TechEd, которые представили неформальные отзывы о нашем продукте.

## 1. Введение в Windows Azure

В этой главе приводится краткое описание технологической платформы Windows Azure™, предоставляемых ею служб и возможностей для облачных вычислений по требованию, где *облако* представляет набор взаимосвязанных вычислительных ресурсов, расположенных в одном или нескольких центрах обработки данных. Эта глава также содержит ссылки, которые помогут вам получить более подробную информацию о возможностях Windows Azure, методах и технологиях, используемых в этой серии руководств, а также образец кода.

Основная цель этого руководства для разработчиков заключается в том, чтобы показать, как использовать пакет интеграции Enterprise Library в приложениях для Windows Azure. Представленный образец реализации (демонстрационный пример) и связанные лабораторные занятия используют многие из возможностей и служб, доступных в Windows Azure, поэтому полезно иметь представление о самой платформе Windows Azure перед тем, как использовать пакет интеграции Enterprise Library. Если вы уже знакомы с Windows Azure, вы можете пропустить эту главу и перейти к следующим главам, описывающим пакет интеграции Enterprise Library для Windows Azure, и образцу реализации — приложению Tailspin Surveys.



платформа, поэтому для получения самых последних сведений о функциях, описанных в этом разделе, используйте приведенные ниже ссылки.

### Сведения о платформе Windows Azure

Организации могут использовать облако для развертывания и запуска приложений, а также хранения данных. Облачные ресурсы могут использовать и локальные приложения. Например, приложение, расположенное на локальном сервере, полнофункциональный клиент, который работает на настольном компьютере или на мобильном устройстве, может использовать хранилище, расположенное в облаке.

Дополнительные сведения о гибридных решениях см. в руководстве по интеграции по адресу <http://wag.codeplex.com/>.

Windows Azure абстрагирует ресурсы оборудования с помощью виртуализации. Каждое приложение, которое развертывается в Windows Azure, работает на одной или нескольких виртуальных машинах (ВМ). Развернутые приложения работают так, будто они распо-



Windows Azure помогает разрабатывать переносимые и масштабируемые приложения, при этом сокращая эксплуатационные расходы и совокупную стоимость владения.

ложены на выделенном компьютере, хотя могут использовать физические ресурсы, такие как место на диске, сетевой канал ввода-вывода или ядра ЦП, совместно с другими виртуальными машинами на том же физическом узле. Два основных преимущества уровня абстракции над физическим оборудованием — мобильность и масштабируемость. Виртуализация служб позволяет перенести их на любое количество физических узлов в центре обработки данных. Сочетание технологий виртуализации, неспециализированного оборудования, многопользовательских сред и агрегирования спроса позволяют Майкрософт и нашим клиентам добиваться значительной экономии. Это дает возможность оптимизировать загрузку центра обработки данных, то есть увеличивать соотношение полезной работы и стоимости оборудования и, соответственно, экономить значительные средства.

Виртуализация также позволяет применять как *вертикальное*, так и *горизонтальное масштабирование*. Вертикальная масштабируемость означает, что по мере роста потребности вы можете увеличивать количество ресурсов, например ядер ЦП и памяти, на определенной виртуальной машине. Горизонтальная масштабируемость означает возможность добавлять новые экземпляры виртуальных машин, которые являются копиями существующих служб. Для всех этих экземпляров применяется балансировка нагрузки на сетевом уровне, чтобы входящие запросы распределялись между ними.

На момент написания этого документа платформа Windows Azure включала в себя Windows Azure и SQL Azure.

**Windows Azure** предоставляет вычислительную среду на основе Microsoft® Windows Server® для размещения приложений, постоянное хранилище структурированных и неструктурированных данных, а также средства асинхронного обмена сообщениями. Кроме того, Windows Azure предусматривает ряд служб, позволяющих подключать пользователей и локальные приложения к облачным приложениям, осуществлять проверку подлинности, обмен сообщениями между службами и управление данными, кэширование и другие функции, связанные с данными.

Windows Azure включает также набор служб управления, которые позволяют контролировать все эти ресурсы с помощью пользовательского веб-интерфейса (веб-портала) либо программными средствами. В большинстве случаев имеется программный интерфейс на основе REST, с помощью которого можно определить то, как будут работать службы. Большинство задач управления, которые могут быть выполнены на веб-портале, также реализуются с помощью API.

По сути дела, **SQL Azure** представляет собой SQL Server®, работающий в виде облачной службы.

И наконец, существует многочисленный набор средств и пакетов средств разработки программного обеспечения для разработки, тестирования и развертывания приложений. Например, разрабатывать и тестировать приложение вы можете в эмулированной локальной среде, созданной с помощью эмулятора вычислений и эмулятора хранилища. Большинство средств также интегрированы в такие среды разработки, как Microsoft Visual Studio®. Кроме того, имеются средства управления сторонних поставщиков.

## Службы и компоненты Windows Azure

Набор служб и компонентов, доступных в Windows Azure и SQL Azure, рассчитан на различные конкретные применения. После подписки на Windows Azure можно выбрать нужные возможности, а оплата будет производиться только за используемые средства. Добавлять компоненты в подписку и удалять их из нее можно в любое время. Механизм выставления счетов для каждой службы зависит от типа предоставляемых возможностей. Дополнительные сведения о модели выставления счетов см. в разделе «*Модель подписок и выставления счетов Windows Azure*» далее в этой главе.

Доступные службы и компоненты изменяются по мере развития Windows Azure. В следующих четырех разделах этой главы кратко описываются основные службы и компоненты, доступные на момент написания этого документа, разделенные на следующие категории: «Вычислительная среда», «Управление данными», «Сетевые службы» и «Другие службы».

Дополнительные сведения обо всех службах и компонентах Windows Azure см. в разделе «*Возможности Windows Azure*» на портале Windows Azure. Конкретные руководства по разработке и использованию каждого из компонентов или служб см. на ресурсах, указанных в следующих разделах.

*Для использования любого из этих компонентов и служб необходима подписка на Windows Azure. Для получения учетной записи Windows Azure требуется действительный идентификатор Windows Live® ID. Дополнительные сведения см. в разделе «Предложения платформ Windows Azure».*



В Windows Azure входит набор служб, которые упрощают разработку, повышают надежность и облегчают управление приложениями, размещенными в облаке.

## Вычислительная среда

Вычислительная среда Windows Azure состоит из платформы для приложений и служб, размещенных в одной или нескольких ролях. В Windows Azure можно реализовать следующие типы ролей.

- **Эмулятор вычислений Windows Azure (веб-роли и рабочие роли).** Приложение Windows Azure состоит из одной или нескольких размещенных ролей, запущенных в центрах обработки данных Azure. Как правило, существует хотя бы одна веб-роль, доступная пользователям приложения. Веб-роль поддерживается службами IIS 7.0 и ASP.NET. Приложение может содержать дополнительные роли, включая рабочие, которые обычно используются для выполнения фоновых задач обработки и поддержки для веб-ролей. Дополнительные сведения см. в разделах «*Общие сведения о создании размещенной службы для Windows Azure*» и «*Построение приложения, работающего в размещенной службе*».
- **Виртуальная машина (роль VM).** Эта роль позволяет разместить собственный экземпляр операционной системы Windows Server 2008 R2 Enterprise или Windows Server 2008 R2 Standard в центре обработки данных Windows Azure. Дополнительные сведения см. в разделе «*Создание приложений с помощью роли VM в Windows Azure*».



Приложение Tailspin Surveys использует веб-роли и рабочие роли. Дополнительные сведения и руководства по использованию веб-ролей и рабочих ролей см. в руководствах *«Перенос приложений в облако»* и *«Разработка приложений для облака»*, которые доступны в разделе *Облачная разработка* на портале MSDN. Каждое из этих руководств также включает в себя набор лабораторных работ.

## Управление данными

Windows Azure, SQL Azure и связанные с ними службы предусматривают возможности для хранения данных и управления ими разными способами. Имеются следующие службы и компоненты управления данными:

- **Хранилище Windows Azure.** Обеспечивает четыре основные службы для длительного хранения данных в облаке. Эти службы поддерживают интерфейс REST, доступ к которому осуществляется как из приложений, размещенных в Windows Azure, так и из локальных (удаленных) приложений. Сведения об интерфейсе REST API см. в разделе *«Справочник по интерфейсу REST API служб хранения Windows Azure»*. Ниже перечислены четыре службы хранения.
- **Служба таблиц Windows Azure** предоставляет механизм хранения данных в виде таблиц и поддерживает запросы для управления данными. Служба таблиц Azure — это предложение NoSQL, обеспечивающее хранение данных без схемы. Она предна- значена прежде всего для тех случаев, когда необходимо хранить большие тома данных, при этом их требуется легко извлекать и обновлять. Дополнительные сведения см. в разделах *«Основные понятия службы таблиц»* и *«Интерфейс REST API службы таблиц»*.
- **Служба больших двоичных объектов (BLOB)** предоставляет ряд контейнеров, предназначенных для хранения текстовых и двоичных данных. Предусмотрены как контейнеры блочных больших двоичных объектов для потоковой передачи данных, так и контейнеры страничных BLOB-объектов для операций произвольного чтения и записи. Дополнительные сведения см. в разделах *«Основные сведения о блочных и страничных больших двоичных объектах»* и *«Интерфейс REST API службы больших двоичных объектов»*.
- **Служба очередей** предоставляет механизм для надежного и постоянного обмена сообщениями между экземплярами ролей, например между веб-ролью и рабочей ролью. Дополнительные сведения см. в разделах *«Основные понятия службы очередей»* и *«Интерфейс REST API службы очередей»*.
- **Диски Windows Azure** предоставляют приложениям механизм для подключения однотомного VHD-диска с файловой системой NTFS как страничного большого двоичного объекта, а также передачи и загрузки виртуальных дисков через BLOB. Дополнительные сведения см. в разделе *«Диск Windows Azure»*.

- **База данных SQL Azure.** Это облачная служба Database Services с высокой степенью доступности и масштабируемости, основанная на технологии SQL Server, которая поддерживает популярную модель реляционных баз данных на основе T-SQL. Ее можно использовать с приложениями, размещенными в Windows Azure, и с другими приложениями, работающими локально или размещенными на других площадках. Дополнительные сведения см. в разделе «*База данных SQL Azure*».
- **Синхронизация данных.** SQL Azure Data Sync — это облачная служба синхронизации данных, основанная на технологиях Microsoft Sync Framework. Эта служба обеспечивает двустороннюю синхронизацию данных и предоставляет возможности по управлению данными, позволяя легко распределять их между несколькими базами данных SQL Azure, а также между локальными базами данных и базами данных SQL Azure. Дополнительные сведения см. в *Центре разработчиков Microsoft Sync Framework*.
- **Служба кэширования.** Эта служба предоставляет для приложений распределенный кэш в памяти с малой задержкой и высокой пропускной способностью, не требующий установки и управления, размер которого динамически увеличивается или уменьшается по мере необходимости. Может использоваться для кэширования данных приложений, получения сведений о состоянии сеансов ASP.NET, а также для кэширования страниц ASP.NET. Дополнительные сведения см. в разделе «*Служба кэширования (Windows Azure)*».

Приложение Tailspin Surveys использует хранилище Windows Azure и SQL Azure. Дополнительные сведения и руководства по использованию хранилища Windows Azure и SQL Azure см. в соответствующих руководствах «*Перенос приложений в облако*» и «*Разработка приложений для облака*», доступных в разделе *Облачная разработка* на портале MSDN. Каждое из этих руководств также включает в себя набор лабораторных работ.

## Сетевые службы

Windows Azure содержит несколько сетевых служб, которые могут быть использованы для оптимизации производительности, реализации проверки подлинности и улучшения управляемости размещенных приложений. Это следующие службы.

- **Сеть кэширующих серверов (CDN).** Сеть CDN позволяет кэшировать общедоступные статические данные для приложений в стратегических точках, расположенных ближе к конечным пользователям (с точки зрения доставки по сети). Сеть CDN использует ряд центров обработки данных во многих местах по всему миру, где данные хранятся в хранилище больших двоичных объектов с анонимным доступом. Это не обязательно должны быть местоположения, где фактически исполняются приложения. Дополнительные сведения см. в разделе «*Доставка контента с высокой пропускной способностью с помощью сети CDN Windows Azure*».

- **Служба подключения к виртуальной сети.** Эта служба обеспечивает такую настройку ролей приложения, работающего в Windows Azure и на компьютерах локальной сети, что позволяет представить, будто эти роли расположены в одной и той же сети. При этом используется запускаемый на локальном компьютере программный агент, который устанавливает защищенное соединение IPsec с ролями Windows Azure в облаке, а также обеспечивает возможность администрирования, отслеживания, отладки и управления ролями напрямую. Дополнительные сведения см. в разделе «*Соединение локальных компьютеров с ролями Windows Azure*».
- **Диспетчер трафика виртуальных сетей.** Эта служба позволяет настроить перенаправление запросов и балансировку нагрузки на основе трех различных методов. Как правило, диспетчер трафика используется для повышения производительности методом, когда запросы перенаправляются к экземпляру приложения в том центре обработки данных, который находится ближе всего к пользователю. Существуют и другие методы балансировки: отработка отказа и циклический перебор. Дополнительные сведения см. в разделе «*Диспетчер трафика Windows Azure*».
- **Управление доступом (ACS).** Эта служба, построенная на основе стандартов, предназначена для управления доступом и удостоверениями и использует множество поставщиков удостоверений (IdP), которые могут выполнять проверку подлинности пользователей. Служба управления доступом (ACS) действует как служба токенов безопасности (STS) и помогает использовать преимущества методов федеративной проверки подлинности, когда подлинность пользователя проверяется не в той области или домене, где находится приложение. В качестве примера можно привести управление доступом пользователей на основе удостоверений, проверяемых поставщиком удостоверений, таким как Windows Live® ID или Google. Дополнительные сведения см. в разделах «*Служба управлением доступом 2.0*» и «*Руководство по управлению доступом и удостоверениями на основе заявок и службы контроля доступом (2-й выпуск)*».
- **Шина обслуживания.** Эта служба обеспечивает безопасный обмен сообщениями и потоками данных в распределенных и гибридных приложениях, например для связи размещенных приложений Windows Azure и локальных приложений, исключая необходимость реализации сложной инфраструктуры брандмауэров и системы безопасности. Она может использовать ряд протоколов и шаблонов взаимодействия и передачи сообщений для гарантированной и надежной доставки сообщений, а также масштабироваться для различных нагрузок и интегрироваться с локальными объектами BizTalk Server. Дополнительные сведения см. в разделе «*Шина обслуживания*».

Дополнительные сведения и руководства по использованию хранилища Windows Azure и SQL Azure см. в соответствующих руководствах «Перенос приложений в облако» и «Разработка приложений для облака», доступных в разделе *Облачная разработка* на портале MSDN. Каждое из этих руководств также включает в себя набор лабораторных работ.

Подробное руководство по использованию ACS можно найти в документе «Руководство по управлению доступом и удостоверениями на основе заявок и службы контроля доступа (2-й выпуск)» и в лабораторных работах для этого руководства.

## Другие службы

Windows Azure предоставляет следующие дополнительные службы.

- **Служба отчетов бизнес-аналитики.** Эта служба предназначена для разработки и развертывания облачных бизнес-отчетов, созданных на основе данных, хранящихся в базе данных SQL Azure. Она построена на основе тех же технологий, что и службы SQL Server Reporting Services, и поэтому позволяет пользоваться привычными средствами для создания отчетов. Доступ к отчетам производится через портал управления Windows Azure, через веб-браузер или непосредственно в приложении Windows Azure или локальном приложении. Дополнительные сведения см. в разделе «Отчеты SQL Azure».
- **Marketplace.** Это портал в сети, на котором разработчики имеют возможность публиковать, искать, приобретать и продавать компоненты приложений, обучающие материалы, шаблоны служб и дополнительные наборы данных, а также готовые службы и приложения, необходимые для создания приложений Windows Azure. Дополнительные сведения см. в статье «Windows Azure Marketplace» на сайте MSDN и «Windows Azure Marketplace» (AppMarket).



## Разработка приложений для Windows Azure

Как правило, в Microsoft® Windows® вы будете пользоваться Visual Studio 2010 со средствами Windows Azure для Microsoft Visual Studio. Средства Windows Azure предлагают все необходимое для создания приложений Windows Azure, включая локальные эмуляторы вычислений и хранилища, работающие на компьютере разработчика. Это означает, что вы можете создавать, тестировать и отлаживать приложения до их развертывания в облаке. К этим средствам также относятся компоненты для развертывания приложений в Windows Azure и управления ими после развертывания.

Средства Windows Azure для Microsoft Visual Studio и средства разработки для других платформ и языков, таких как iOS, Eclipse, Java, Ruby и PHP, можно загрузить в разделе «Средства Windows Azure».

Полезный набор из видеороликов, кратких руководств с примерами и лабораторных работ, который охватывает различные темы,

Приложения Windows Azure можно создавать и тестировать на локальном компьютере разработчика с помощью эмуляторов вычисления и хранения.

необходимые для начала построения приложений Windows Azure, см. в разделах «Обучение работе с Windows Azure и SQL Azure» и «Проектирование. Кодирование. Масштабирование».

В статье «Разработка приложений для Windows Azure» на сайте MSDN представлены конкретные примеры и руководство по созданию размещенных услуг, использованию средств Windows Azure для Microsoft Visual Studio для упаковки и развертывания приложений, а также краткое руководство по использованию.

В обучающем наборе Windows Azure представлены лабораторные работы, которые помогут быстро приступить к созданию приложений.

Сведения о жизненном цикле выполнения и работе ролей Windows Azure см. в разделе «Реальный мир: жизненный цикл запуска роли Windows Azure».

Список полезных ресурсов для разработки и развертывания баз данных в SQL Azure см. в разделе «Разработка (база данных SQL Azure)».

Список средств, которые могут помочь при планировании миграции приложения в Windows Azure, см. в разделе «Оценка и планирование» статьи «Средства Windows Azure».



В Windows Azure имеются возможности, позволяющие отслеживать и отлаживать облачные службы.

## Мониторинг, отладка приложений Windows

### Azure и управление ими

Все подсистемы хранения данных и управления в Windows Azure используют интерфейсы на основе REST. Они не зависят от конкретной технологии .NET Framework или операционной системы Windows. Любая технология, которая может выдавать запросы HTTP или HTTPS, позволяет получить доступ к ресурсам Windows Azure.

Сведения об управляемых и собственных библиотеках API, а также интерфейсе REST API служб хранения Windows Azure см. в разделе «Справочники по интерфейсам API для Windows Azure».

API управления службами на основе REST может использоваться как альтернатива portalу управления Windows Azure. Этот API содержит функции для работы с учеными записями хранения, размещенными службами, сертификатами, территориальными группами, местоположениями и информацией о подписке. Дополнительные сведения см. в статье «Справочник по интерфейсу REST API управления службами Windows Azure». Кроме того, в Windows Azure предусмотрены службы диагностики и API для таких операций, как наблюдение за исправностью приложений. Пакет управления Windows Azure и System Center Operations Manager 2007 R2 может быть использован для обнаружения приложений Windows Azure, получения состояния каждого экземпляра роли, а также сбора и отслеживания информации о производительности, событиях Windows Azure и сообщений трассировки .NET Framework от каждого экземпляра роли. Дополнительные сведения см. в разделе «Отслеживание приложений Windows Azure».

Можно также с помощью командлетов Windows Azure PowerShell производить обзор и управление службами эмулятора вычислений и хранения данных Windows Azure, автоматизировать развертывание

и обновление приложений Windows Azure, а также управлять данными диагностики.

Сведения об использовании встроенных объектов трассировки Windows Azure для настройки диагностики и средств без использования Operations Manager, а также сведения о загрузке результатов см. в разделе *«Сбор данных журналов с помощью службы диагностики Windows Azure»*.

Сведения об отладке приложений Windows Azure см. в разделе *«Устранение неполадок и отладка в Windows Azure»* и *«Отладка приложений в Windows Azure»*.

**Глава 7, «Управление жизненным циклом приложений Windows Azure», руководства «Перенос приложений в облако» содержит сведения об управлении приложениями Windows Azure.**

## Управление базами данных SQL Azure

Приложения получают доступ к базам данных SQL Azure так же, как и к локально установленным экземплярам SQL Server, с помощью управляемых классов доступа к данным ADO.NET, DAAB, OData, собственных классов ODBC, PHP, Ruby или технологий доступа к данным JDBC.

Управление базами данных SQL Azure осуществляется через веб-портал, в среде SQL Server Management Studio, с помощью средств для работы с базами данных Visual Studio 2010 и других инструментов для выполнения таких задач, как перемещение и перенос данных. Доступны и другие средства командной строки для развертывания и администрирования баз данных.

Кроме того, доступен диспетчер баз данных, который облегчает работу с базами данных SQL Azure. Дополнительные сведения см. в разделе *«Портал управления SQL Azure»*. Список других средств см. в разделе *«Средства Windows Azure»*.

SQL Azure поддерживает API управления, а также управление через веб-портал. Сведения об API-интерфейсе управления SQL Azure см. в разделе *«Справочник по интерфейсу управления REST API»*.

## Обновление приложений для Windows Azure

После развертывания приложения для Windows Azure его необходимо обновлять при изменении служб ролей в ответ на новые требования и улучшения в коде или для исправления ошибок. Вы можете просто выполнить повторное развертывание службы. Для этого нужно приостановить и удалить ее, затем выполнить развертывание новой версии. Однако вы можете избежать простоя приложений за счет поэтапного развертывания (загрузки нового пакета и замены существующей рабочей версии этим пакетом) либо выполнения обновления на месте (отправки нового пакета и его применения к выполняемым экземплярам службы).

Сведения об обновлении служб за счет загрузки нового пакета и замены существующей рабочей версии см. в разделе *«Развертывание обновления службы в рабочей среде за счет смены VIP в Windows Azure»*.



Владелец учетной записи и администратор служб, управляющие подпиской, могут (и во многих случаях должны) использовать различные идентификаторы Live ID.

Сведения о выполнении обновлений на месте, а также описание развертывания служб в доменах отказоустойчивости и обновления и сведения о влиянии этого развертывания на параметры обновления см. в разделе «*Выполнение обновлений размещенной службы на месте в Windows Azure*».

*Если необходимо только изменить сведения о конфигурации службы без развертывания нового кода, то можно с помощью веб-портала или API управления изменить файл конфигурации службы или загрузить новый.*

## Модель подписок и выставления счетов Windows Azure

Для работы в Windows Azure необходимо сначала создать учетную запись для выставления счетов, зарегистрировавшись в службах *Microsoft Online Services* или на портале Windows Azure по адресу <https://windows.azure.com/>. Клиентский портал Microsoft Online Services позволяет управлять подписками на все службы Майкрософт. К ним относится и Windows Azure, но есть и другие службы, такие как Business Productivity Online, Windows Office Live Meeting и Windows Intune™.

*Этот раздел подготовлен на основе информации, которая была общедоступной на момент написания этой статьи.*

У каждой учетной записи выставления счетов есть один владелец, который определяется по идентификатору Windows Live ID. Владелец учетной записи может создавать подписки и управлять ими, просматривать сведения о счетах и данные об использовании, а также указывать администратора служб для каждой подписки.

Администраторы управляют отдельными размещенными службами для подписки Windows Azure с помощью портала Windows Azure по адресу <https://windows.azure.com/>. Подписка Windows Azure может включать один или несколько следующих элементов:

- размещенные службы, состоящие из размещенных ролей и экземпляров в каждой роли; роли и экземпляры могут быть остановлены в рабочем режиме или в режиме промежуточного хранения;
- учетные записи хранения, состоящие из экземпляров служб таблиц, больших двоичных объектов и очередей;
- экземпляры CDN;
- базы данных SQL Azure и службу синхронизации данных;
- экземпляры служб Reporting Services SQL Azure;
- экземпляры служб управления доступом, кэширования и шины обслуживания;
- экземпляры подключения виртуальной сети и диспетчера трафика.

На рис. 1 показана конфигурация выставления счетов Windows Azure для стандартной подписки.

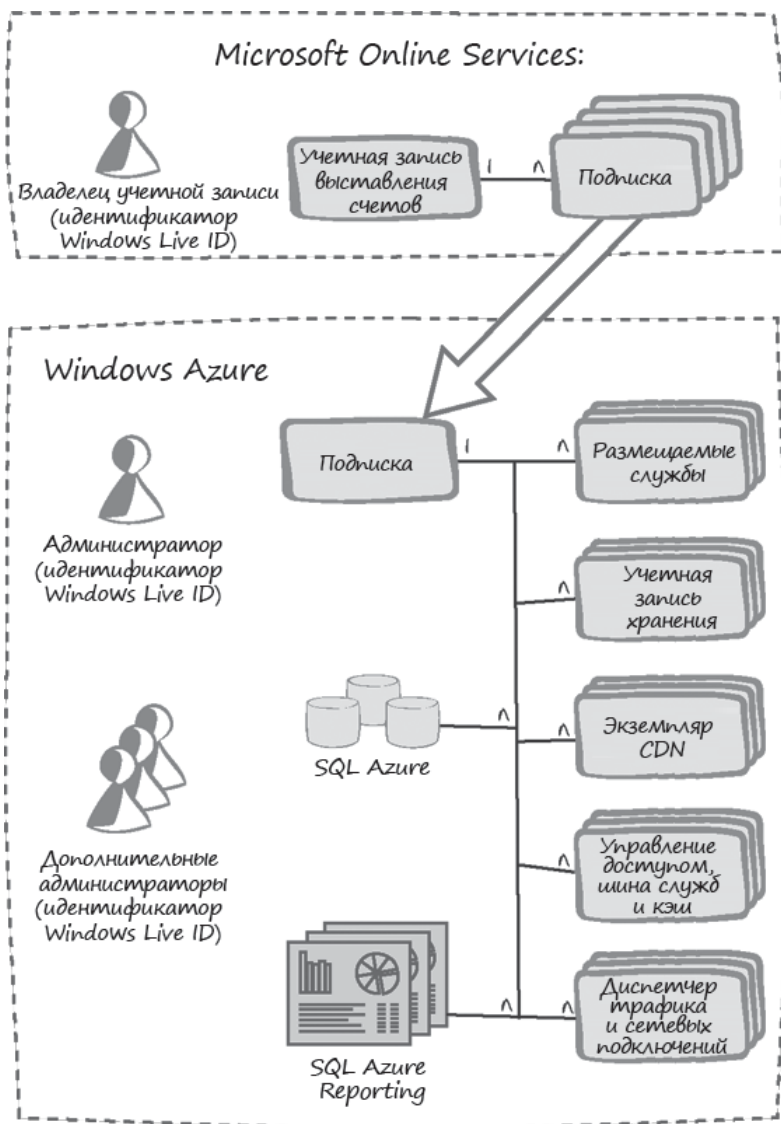


Рис. 1  
Конфигурация выставления счетов Windows Azure для стандартной подписки

Дополнительные сведения о выставлении счетов для Windows Azure см. в разделе «Основные сведения о выставлении счетов для Windows Azure» и «Учетные записи и выставление счетов в SQL Azure».





Вам выставляется счет за ресурсы роли, используемые развернутой службой, даже если роли для этой службы не эксплуатируются. Если вы не хотите, чтобы за службу взималась плата, удалите развертывания, связанные со службой.

## Оценка затрат

Тарифы Windows Azure зависят от показателей использования служб, например от времени вычислений, объема хранилища и пропускной способности. Тарифы на время вычислений рассчитываются согласно почасовой ставке, а также в зависимости от размера экземпляра. Тарифы за хранение данных основаны на объеме данных в гигабайтах и количестве транзакций. Стоимость передачи данных варьируется в зависимости от географического расположения и обычно начисляется на данные, передаваемые между центрами обработки данных Майкрософт и конкретной организацией, но не на операции по передаче данных в пределах одного центра обработки данных.

Для оценки вероятных расходов подписок Windows Azure можно использовать следующие ресурсы.

- Общие сведения о подписке по различным моделям приобретения, например по модели оплаты за фактически потребленные объемы, включая средство замера потребления, см. по адресу <http://www.microsoft.com/windowsazure/pricing/>.
- Калькулятор тарифов по адресу <http://www.microsoft.com/windowsazure/pricing-calculator/>.
- Калькулятор общей стоимости владения по адресу <http://www.microsoft.com/windowsazure/offers/#tcoCompare-LB>.

*В главе 4, «Сколько это будет стоить?», руководства «Перенос приложений в облако» представлены дополнительные сведения об оценке стоимости размещения приложений в Windows Azure.*

## Дополнительные сведения

Много информации о Windows Azure доступно в виде документации, учебных видеофильмов и технических документов. Ниже приведены некоторые веб-сайты, на которых можно получить дополнительные сведения.

- Информационный портал о Windows Azure находится по адресу <http://www.microsoft.com/WindowsAzure/>. На нем представлены ссылки на технические документы, пакет SDK Windows Azure для .NET и многие другие ресурсы. Там же можно получить учетную запись Windows Azure.
- Портал обучения Windows Azure по адресу <http://www.microsoft.com/WindowsAzure/learn>.
- Уэйд Вегнер (Wade Wegner) и Стив Маркс (Steve Marx) создали ряд передач на Channel 9 о Windows Azure, которые доступны по адресу <http://channel9.msdn.com/shows/Cloud+Cover/>.
- Ответы на свои вопросы можно найти на форуме Windows Azure по адресу <http://social.msdn.microsoft.com/Forums/ru-RU/windowsazuredevelopment/threads>.

- Стив Маркс (Steve Marx) — технический руководитель продукта Windows Azure. Его блог находится по адресу <http://blog.smarx.com/>. Это прекрасный источник новостей и информации о Windows Azure.
- Уэйд Вернер (Wade Wegner) — ведущий технический специалист по Windows Azure. В его блоге представлено множество технических деталей и советов. Его можно найти по адресу <http://www.wadewegner.com/>.
- Журнал голосования Windows Azure находится по адресу <http://www.mygreatwindowsazureidea.com> — здесь можно подготовить и отправить свой отзыв, а также проголосовать в пользу той или иной очередности реализации функций платформы.
- На сайте сообщества с шаблонами и практическими руководствами по адресу <http://wag.codeplex.com/> представлены ссылки на ресурсы в сети, образцы кода, лабораторные работы, отзывы и многое другое.
- Сайт сообщества для этого выпуска по адресу <http://entlib.uservoice.com/forums/101257-windows-azure-integration-pack> содержит ссылки на дополнительные ресурсы в сети, внесенные изменения и форум для обсуждений.

Ниже приведены ссылки на ресурсы, представленные в настоящей главе:

- Руководство по Windows Azure на CodePlex:  
<http://wag.codeplex.com/>
- Возможности Windows Azure:  
<http://www.microsoft.com/windowsazure/features/>
- Предложения Windows Azure:  
<http://www.microsoft.com/windowsazure/offers/>
- Общие сведения о создании размещенной службы для Windows Azure:  
<http://go.microsoft.com/fwlink/?LinkID=234572>
- Построение приложения, работающего в размещенной службе:  
<http://go.microsoft.com/fwlink/?LinkID=234587>
- Создание приложений с помощью роли ВМ в Windows Azure:  
<http://go.microsoft.com/fwlink/?LinkID=234590>
- Облачная разработка:  
<http://go.microsoft.com/fwlink/?LinkID=234613>
- Справочник по интерфейсу REST API служб хранения Windows Azure:  
<http://msdn.microsoft.com/ru-ru/library/dd179355.aspx>
- Основные понятия службы таблиц:  
<http://msdn.microsoft.com/ru-ru/library/dd179463.aspx>
- Интерфейс REST API службы таблиц:  
<http://msdn.microsoft.com/ru-ru/library/dd179423.aspx>

- Основные сведения о блочных и страничных больших двоичных объектах:  
<http://msdn.microsoft.com/ru-ru/library/ee691964.aspx>
- Интерфейс REST API службы больших двоичных объектов:  
<http://msdn.microsoft.com/ru-ru/library/dd135733.aspx>
- Основные понятия службы очередей:  
<http://msdn.microsoft.com/ru-ru/library/dd179353.aspx>
- Интерфейс REST API службы очередей:  
<http://msdn.microsoft.com/ru-ru/library/dd179363.aspx>
- Диск Windows Azure:  
<http://go.microsoft.com/?linkid=9710117>
- База данных SQL Azure:  
<http://msdn.microsoft.com/ru-ru/library/ee336279.aspx>
- Центр разработки Microsoft Sync Framework:  
<http://msdn.microsoft.com/ru-ru/sync>
- Служба кэширования (Windows Azure):  
<http://go.microsoft.com/fwlink/?LinkID=234591>
- Доставка контента с высокой пропускной способностью по сети CDN Windows Azure:  
<http://go.microsoft.com/fwlink/?LinkID=234592>
- Соединение локальных компьютеров с ролями Windows Azure:  
<http://go.microsoft.com/fwlink/?LinkID=234593>
- Диспетчер трафика Windows Azure:  
<http://go.microsoft.com/fwlink/?LinkID=234594>
- Служба управления доступом 2.0:  
<http://go.microsoft.com/fwlink/?LinkID=234595>
- Руководство по удостоверениям на основе заявок и службы управления доступом (2-й выпуск):  
<http://msdn.microsoft.com/ru-ru/library/ff423674.aspx>
- Шина обслуживания:  
<http://go.microsoft.com/fwlink/?LinkID=234596>
- SQL Azure Reporting:  
<http://go.microsoft.com/fwlink/?LinkID=234597>
- Windows Azure Marketplace:  
<http://go.microsoft.com/fwlink/?LinkID=234598>
- Windows Azure Marketplace:  
<http://go.microsoft.com/fwlink/?LinkID=234599>
- Средства Windows Azure:  
<http://www.microsoft.com/windowsazure/tools/>
- Обучение работе с Windows Azure и SQL Azure:  
<http://www.microsoft.com/windowsazure/tutorials/>
- Проектирование. Кодирование. Масштабирование:  
<http://www.microsoft.com/windowsazure/getstarted/>

- Разработка приложений для Windows Azure:  
<http://go.microsoft.com/fwlink/?LinkID=234600>
- Обучающий набор Windows Azure:  
<http://go.microsoft.com/fwlink/?LinkID=234601>
- Реальный мир: жизненный цикл запуска роли Windows Azure:  
<http://go.microsoft.com/fwlink/?LinkID=234602>
- Разработка (база данных SQL Azure):  
<http://go.microsoft.com/fwlink/?LinkID=234603>
- Оценка и планирование Windows Azure:  
<http://www.microsoft.com/windowsazure/tools/#assessment>
- Справочники по API для Windows Azure:  
<http://msdn.microsoft.com/ru-ru/library/ff800682.aspx>
- Справочник по API REST управления службами Windows Azure:  
<http://msdn.microsoft.com/ru-ru/library/ee460799.aspx>
- Наблюдение за приложениями Windows Azure:  
<http://msdn.microsoft.com/ru-ru/library/gg676009.aspx>
- Командлеты PowerShell Windows Azure:  
<http://wappowershell.codeplex.com/>
- Сбор данных журналов с помощью службы диагностики Windows Azure:  
<http://go.microsoft.com/fwlink/?LinkID=234604>
- Устранение неполадок и отладка в Windows Azure:  
<http://go.microsoft.com/fwlink/?LinkID=234605>
- Отладка приложений для Windows Azure:  
<http://go.microsoft.com/fwlink/?LinkID=234606>
- «Управление жизненным циклом приложений Windows Azure» в руководстве «Перенос приложений в облако»:  
<http://msdn.microsoft.com/ru-ru/library/ff803362.aspx>
- Портал управления для SQL Azure:  
<http://go.microsoft.com/fwlink/?LinkID=234607>
- Средства Windows Azure:  
<http://www.microsoft.com/windowsazure/tools/#sqlazure>
- Справочник по API REST управления:  
<http://go.microsoft.com/fwlink/?LinkID=234608>
- Развертывание обновления службы в рабочей среде за счет смены VIP в Windows Azure:  
<http://go.microsoft.com/fwlink/?LinkID=234609>
- Как выполнить обновление на месте для размещенной службы в Windows Azure:  
<http://go.microsoft.com/fwlink/?LinkID=234610>
- Microsoft Online Services:  
<https://mosp.microsoftonline.com/>

- «Как выставляются счета в Windows Azure?» в разделе «Платформа Windows Azure»  
<http://go.microsoft.com/fwlink/?LinkID=234611>
- Учетные записи и выставление счетов в SQL Azure:  
<http://go.microsoft.com/fwlink/?LinkID=234612>
- Общие сведения о подписке для различных моделей покупки:  
<http://www.microsoft.com/windowsazure/pricing/>
- Калькулятор тарифов:  
<http://www.microsoft.com/windowsazure/pricing-calculator/>
- Калькулятор совокупной стоимости владения:  
<http://www.microsoft.com/windowsazure/offers/#tcoCompare-LB>
- «Сколько это будет стоить?» в книге «Перенос приложений в облако»:  
<http://msdn.microsoft.com/en-us/library/ff803375.aspx>
- Перенос приложений в облако, 2-е издание:  
<http://msdn.microsoft.com/en-us/library/ff728592.aspx>

Чтобы получить простой доступ к ресурсам в сети, используйте электронную версию библиографии на MSDN:  
[http://msdn.microsoft.com/en-us/library/hh749032\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh749032(v=PandP.50).aspx)

## Описание пакета интеграции Enterprise Library Integration Pack для Windows Azure

Пакет интеграции Microsoft® Enterprise Library Integration Pack для Windows Azure расширяет библиотеку Enterprise Library и обеспечивает поддержку приложений на основе платформы Windows Azure™. Пакет включает дополнительные программные блоки для удовлетворения требований, предъявляемых к облачным приложениям.

### Что такое библиотека Enterprise Library

Библиотека Enterprise Library предоставляет множество настраиваемых функций, которые упрощают управление повторяющимися задачами (часто называемых пересекающимися), которые встречаются в разных частях приложения. К этим задачам относится ведение журнала, проверка ввода, кэширование, перехват исключений и многое другое. Кроме того, контейнер внедрения зависимостей библиотеки Enterprise Library поможет упростить и снизить количество связей в проектах, сделать их понятнее и облегчить процесс тестирования, предоставит средства для более эффективного проектирования и реализации приложений всех типов.

Библиотека Enterprise Library состоит из набора программных блоков и основной инфраструктуры. Все эти компоненты многократного использования помогают разработчикам решать общие задачи при разработке корпоративных приложений. Каждый блок предназначен для решения определенного набора задач. К примеру, программный блок ведения журнала упрощает реализацию в приложении общих функций ведения журнала и позволяет записывать данные журнала в различные места. Блок доступа к данным упрощает решение общих задач по обеспечению доступа к данным, таких как чтение данных для отображения в пользовательском интерфейсе или отправка измененных данных обратно в систему с основной базой данных.

Как показано на рис. 1, в библиотеке Enterprise Library имеются следующие программные блоки.

- *Программный блок кэширования.* Этот блок можно использовать для включения в приложения локального кэша.
- *Программный блок шифрования.* Этот блок может использоваться для включения хэширования и симметричного шифрования в приложения.
- *Программный блок доступа к данным.* Этот блок можно использовать для включения в приложения стандартных функций базы данных.
- *Программный блок обработки исключений.* Разработчики и создатели правил могут использовать этот блок для создания согласованной стратегии обработки исключений, которая будет использоваться на всех уровнях архитектуры корпоративных приложений.
- *Программный блок ведения журнала.* Разработчики могут использовать этот блок для включения стандартных функций ведения журнала в свои приложения, а системные администраторы могут использовать средство настройки для задания гранулярности журнала во время выполнения.
- *Программный блок внедрения правил.* Этот блок содержит устаревший код для обратной совместимости с существующими приложениями. Новые функции представлены в механизме перехвата Unity и обработчиках вызовов из соответствующих сборок программного блока.
- *Программный блок безопасности.* Разработчики могут использовать этот блок для обеспечения авторизации и безопасного кэширования в приложениях.
- *Программный блок проверки.* Разработчики могут использовать этот блок для создания правил проверки бизнес-объектов, применяемых на различных уровнях приложений.
- *Механизм внедрения и перехвата зависимостей Unity.* Разработчики могут использовать эти методы для реализации небольшого, расширяемого контейнера внедрения зависимостей с конструктором, свойством и средством внедрения вызова метода, а также для перехвата вызовов к целевым объектам и добавления к ним дополнительных функций.

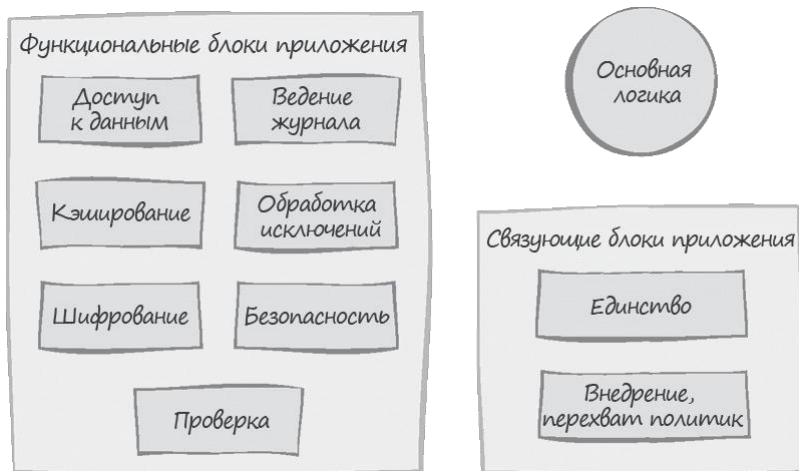


Рис. 1  
Компоненты библиотеки Enterprise Library

Библиотека Enterprise Library основана на конфигурациях и поддерживает как программные, так и внешние конфигурации. Помимо программных блоков, библиотека Enterprise Library содержит средства настройки, а также набор основных функций, которые решают задачи, относящиеся ко всем блокам.

Назначение библиотеки Enterprise Library.

- **Согласованность.** Все программные блоки приложений Enterprise Library имеют согласованную структуру и методы реализации.
- **Расширяемость.** Все программные блоки включают определенные точки расширяемости, которые позволяют разработчикам изменять поведение блоков путем добавления собственного кода.
- **Простота использования.** Библиотека Enterprise Library предлагает множество удобных средств для работы, включая средство настройки с графическим интерфейсом, простую процедуру установки, ясную и полную документацию с образцами использования.
- **Интеграция.** Программные блоки библиотеки Enterprise Library создавались так, чтобы без проблем работать друг с другом, что было подтверждено в ходе тестирования. Однако использовать их вместе не обязательно.

Чтобы узнать больше о библиотеке Enterprise Library и программных блоках, посетите главный сайт библиотеки Enterprise Library на портале MSDN® *Библиотека Enterprise 5.0 — май 2011 года.*



## Описание пакета интеграции Enterprise Library для Windows Azure

Пакет интеграции Enterprise Library для Windows Azure расширяет библиотеку Enterprise Library версии 5.0 и обеспечивает поддержку разработки и управления приложениями на основе платформы Windows Azure. Он предоставляет те же преимущества, что и библиотека Enterprise Library, и позволяет разработчикам достигать тех же целей.

Пакет интеграции Enterprise Library для Windows Azure содержит следующее.

- Программный блок *автоматического масштабирования*, помогающий автоматически масштабировать приложения Windows Azure.
- Программный блок *обработки временных сбоев*, помогающий сделать приложения Windows Azure более устойчивыми к временным ошибкам.
- Большой двоичный объект для хранения конфигурации библиотеки Enterprise Library в хранилище больших двоичных объектов Azure.
- Защищенный поставщик конфигурации.
- Командлеты интерфейса командной строки Windows PowerShell® для управления автоматическим масштабированием приложений.
- Обновленные скрипты создания базы данных (для программного блока ведения журнала и кэширования), использующие технологическую платформу SQL Azure™.
- Справочная документация.
- Руководство разработчика.
- Эталонная реализация, в которой иллюстрируется использование новых программных блоков.

Пакет интеграции Enterprise Library для Windows Azure является одним из нескольких существующих и запланированных пакетов интеграции для библиотеки Enterprise Library.

Дополнительные сведения о библиотеке Enterprise Library см. в статье *Библиотека Microsoft Enterprise Library 5.0 — май 2011 года и Руководство разработчика*.

Дополнительные сведения о пакете интеграции Silverlight, применяемом в приложениях с подключаемым модулем браузера Microsoft Silverlight®, см. в разделе *Пакет интеграции Silverlight библиотеки Enterprise Library 5.0*.

## Дополнительные сведения

Дополнительные сведения о программных блоках Enterprise Library см. в следующих ресурсах на портале MSDN:

- Библиотека Microsoft Enterprise Library 5.0 — май 2011 года:  
<http://msdn.microsoft.com/en-us/library/ff632023.aspx>
- Руководство разработчика:  
[http://msdn.microsoft.com/en-us/library/ff953181\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/ff953181(v=PandP.50).aspx)
- Пакет интеграции Silverlight библиотеки Enterprise Library 5.0:  
<http://entlib.codeplex.com/wikipage?title=EntLib5Silverlight>
- Программный блок кэширования:  
[http://msdn.microsoft.com/en-gb/library/ff664753\(PandP.50\).aspx](http://msdn.microsoft.com/en-gb/library/ff664753(PandP.50).aspx)
- Программный блок шифрования:  
[http://msdn.microsoft.com/en-gb/library/ff664484\(PandP.50\).aspx](http://msdn.microsoft.com/en-gb/library/ff664484(PandP.50).aspx)
- Программный блок доступа к данным:  
[http://msdn.microsoft.com/en-gb/library/ff664408\(PandP.50\).aspx](http://msdn.microsoft.com/en-gb/library/ff664408(PandP.50).aspx)
- Программный блок обработки исключений:  
[http://msdn.microsoft.com/en-gb/library/ff664698\(PandP.50\).aspx](http://msdn.microsoft.com/en-gb/library/ff664698(PandP.50).aspx)
- Программный блок ведения журнала:  
[http://msdn.microsoft.com/en-gb/library/ff664569\(PandP.50\).aspx](http://msdn.microsoft.com/en-gb/library/ff664569(PandP.50).aspx)
- Программный блок внедрения правил:  
[http://msdn.microsoft.com/en-gb/library/ff664572\(PandP.50\).aspx](http://msdn.microsoft.com/en-gb/library/ff664572(PandP.50).aspx)
- Программный блок безопасности:  
[http://msdn.microsoft.com/en-gb/library/ff664771\(PandP.50\).aspx](http://msdn.microsoft.com/en-gb/library/ff664771(PandP.50).aspx)
- Программный блок проверки:  
[http://msdn.microsoft.com/en-gb/library/ff664356\(PandP.50\).aspx](http://msdn.microsoft.com/en-gb/library/ff664356(PandP.50).aspx)
- Механизм внедрения и перехвата зависимостей Unity:  
<http://msdn.microsoft.com/unity>

Дополнительные сведения о программном блоке автоматического масштабирования см. в разделе «Программный блок автоматического масштабирования» на MSDN:

[http://msdn.microsoft.com/en-us/library/hh680892\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680892(v=PandP.50).aspx)

Дополнительные сведения о программном блоке обработки временных ошибок см. в разделе «Программный блок обработки временных ошибок» на MSDN:

[http://msdn.microsoft.com/en-us/library/hh680934\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680934(v=PandP.50).aspx)

Чтобы получить простой доступ к ресурсам в сети, используйте электронную версию библиографии на MSDN:

[http://msdn.microsoft.com/en-us/library/hh749032\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh749032(v=PandP.50).aspx)

### 3 Сценарий Tailspin

В этом разделе представлена вымышленная компания под названием Tailspin. Здесь описывается план компании Tailspin по использованию библиотеки Microsoft Enterprise Library для дальнейшего развития своей главной службы в сети под названием Surveys. Surveys — это облачная служба, размещенная на платформе Windows Azure™, позволяющая различным людям и компаниям проводить свои собственные опросы в сети и анализировать полученные результаты. Как и в любой другой компании, при планировании обновления ключевого приложения возникает много тонкостей, которые следует учесть, и проблем, которые нужно решить, к тому же разработчики компании Tailspin используют библиотеку Enterprise Library впервые. В последующих разделах будут раскрыты преимущества, которая получила компания Tailspin благодаря использованию библиотеки Enterprise Library, и шаг за шагом описаны внесенные в архитектуру приложения Surveys изменения, позволившие использовать библиотеку Enterprise Library.

#### Компании Tailspin

Tailspin — это начинающая независимая компания, состоящая примерно из 20 сотрудников, специализирующаяся на разработке облачных решений с помощью технологий Microsoft®. Разработчики Tailspin осведомлены о различных продуктах и технологиях Майкрософт, включая Windows Azure, .NET Framework, ASP.NET MVC, SQL Server® и среду разработки Microsoft Visual Studio®. Эти разработчики знают о возможностях библиотеки Enterprise Library, но еще не использовали ее в своих облачных приложениях.

Приложение Surveys стало одной из первых инновационных служб в сети, которые Tailspin выпустила на рынок. Будучи начинающей компанией, Tailspin решила специализироваться на облачных решениях, чтобы свести к минимуму затраты на оборудование и максимально расширить аудиторию потенциальных клиентов. Компания Tailspin рассчитывала, что одна из этих облачных служб будет быстро расти, и хотела иметь возможность быстро реагировать на растущий спрос. Кроме того, компания вполне допускала, что неко-

торые облачные службы в сети потерпят провал, и не хотела оставаться в результате с ненужным оборудованием на руках. Приложение *Surveys* стало очень успешным проектом компании *Tailspin*, его используют тысячи клиентов по всему миру, и каждый день регистрируются новые клиенты.

## Стратегия компании *Tailspin*

*Tailspin* является новаторской и эластичой организацией, использующей новые технологии и бизнес-возможности, предоставляемые облачной средой. Стратегия компании *Tailspin* заключалась в том, чтобы освоить облачные технологии и получить конкурентное преимущество за счет быстрого приобретения нового опыта и на его основе развиваться дальше. Эту стратегию можно описать словами «пробуй, быстро обнаруживай неудачу, учишься на ошибках, а затем пробуй снова». Приложение *Surveys* принесло компании *Tailspin* большой коммерческий успех, однако позже были выявлены некоторые проблемы первоначальной реализации с точки зрения гибкости, управляемости и удобства сопровождения приложения.

## Приложение *Surveys*

Приложение *Surveys* позволяет клиентам *Tailspin* разрабатывать и публиковать опросы, а затем проводить анализ полученных результатов. Опрос — это набор вопросов, каждый из которых может иметь различную форму, например вопрос с вариантами ответов, вопрос с ответом в виде числового интервала или простого текста. Работа заказчиков в службе *Surveys* начинается с создания подписки, которая используется для управления опросами и обозначения продукта (брендинга) с помощью стилей и изображений эмблем. Клиенты также могут выбрать географическое местоположение для своей учетной записи и таким образом размещать свои опросы как можно ближе к опрашиваемой аудитории. Приложение *Surveys* позволяет пользователям бесплатно опробовать приложение, а также подписаться на один из нескольких различных пакетов, которые предлагают различные коллекции служб за ежемесячную плату.

На рис. 1 изображено приложение *Surveys*, где отдельно выделены три различные группы пользователей (клиенты, администраторы компании *Tailspin* и участники опросов), взаимодействующие с приложением.



Рис. 1  
Приложение Surveys

Клиенты, оформившие подписку на службу Surveys (или использующие ее в течение бесплатного пробного периода), имеют доступ к веб-сайту подписчика, который позволяет им составлять собственные опросы, добавлять фирменные элементы стиля, собирать и анализировать результаты опроса. В зависимости от выбранного пакета они имеют доступ к различным уровням функциональных возможностей в рамках приложения Surveys. Компания Tailspin предполагает, что ее службой будут пользоваться заказчики и предприятия разных размеров со всего мира, при этом клиенты смогут выбирать географическое местоположение для своей учетной записи и опросов.

Компания Tailspin хочет спроектировать службу таким образом, чтобы большинство задач управления и настройки выполнял с помощью службы сам подписчик при минимальном вмешательстве персонала Tailspin.

Открытый веб-сайт позволяет давать ответы на вопросы в опросе. Создатель опроса предоставляет исследуемой аудитории URL-адрес для прохождения опроса.

Веб-сайт компании Tailspin позволяет ее сотрудникам управлять приложением и учетными записями подписчиков службы. Все три веб-сайта (для подписчиков, открытый веб-сайт и веб-сайт для сотрудников Tailspin) взаимодействуют с основными службами, которые формируют основу приложения Surveys и предоставляют доступ к хранилищу данных приложения.



В мире разработки программного обеспечения как услуг (SaaS) подписчики часто именуются «пользователями». И мы часто называем приложения, подобные Tailspin Surveys, многопользовательскими.

## Цели и задачи, стоящие перед компанией Tailspin

Перед компанией Tailspin стоит несколько задач, связанных с текущей версией приложения Surveys. Первоначальная разработка была проведена быстро, чтобы выпустить продукт на рынок как можно раньше. В ходе процесса разработки были приняты некоторые компромиссы, определенными функциями пришлось пожертвовать, и процесс проектирования был ориентирован на жесткие сроки, а не на оптимизацию возможностей по эксплуатации и расширяемости. Следующий этап существования приложения Surveys рассматривается компанией Tailspin как этап консолидации, на котором будут заложены основы будущего развития приложения и устранены недостатки первоначальной реализации.

Вот как работает исходное приложение. Сначала клиенты создают опросы. Они могут быть связаны с запуском продуктов или маркетинговыми кампаниями; это могут быть и сезонные опросы, например по поводу праздников. Часто заказчики приложения создают опросы в очень короткие сроки. Опросы обычно проводятся в короткий, фиксированный период времени, но могут иметь большое количество респондентов. Это означает, что интенсивность использования приложения Surveys может меняться резкими скачками и весьма непредсказуемо. Сегодня компания Tailspin предлагает свое приложение Surveys клиентам во всем мире, а из-за того, что приложение по своей сути подразумевает внезапные всплески интенсивности использования, компания должна иметь возможность быстро увеличивать или сокращать инфраструктуру приложения в различных географических точках. До сих пор компания Tailspin использовала процесс, при котором оператор вручную добавлял и удалял экземпляры ролей на основе данных о производительности, собранных с помощью приложения, и исходя из сведений об ожидаемых или запланированных событиях. Чтобы улучшить обслуживание все возрастающего количества клиентов и управлять своими расходами, компания Tailspin хотела бы автоматизировать процесс масштабирования.

Веб-сайты, обслуживающие подписчиков и респондентов опроса, также имеют различные требования к масштабируемости. Тысячи пользователей могут пройти опрос, но изменять имеющиеся опросы или создавать новые будет лишь небольшое число пользователей-подписчиков. Tailspin хочет оптимизировать использование ресурсов для каждого из этих сценариев.

При возникновении проблем с приложением Surveys сотрудникам Tailspin иногда приходится прилагать много усилий для быстрого их устранения, чтобы соблюсти соглашения об уровне обслуживания. Tailspin хочет иметь возможность более эффективно реагировать на возникающие проблемы в приложении, имея на руках легко доступные данные диагностики.



Эластичное распределение ресурсов и распределение на основе географического положения являются ключевыми свойствами платформы Windows Azure.

Tailspin хочет сохранить свое конкурентное преимущество, быстро добавляя новые функции в существующие службы, или получить конкурентное преимущество, выходя на рынок первой с новыми продуктами и услугами. Для приложения Surveys компания хочет создать платформу с четкой, согласованной архитектурой, которая легко расширяется и совершенствуется.

Бизнес-модель компании Tailspin подразумевает взимание с подписчиков ежемесячной платы за пользование службами, такими как Surveys, а поскольку компания предоставляет свои услуги клиентам во всем мире, ей необходимо иметь конкурентные цены. Компания Tailspin должна оплачивать фактические расходы по эксплуатации приложения, поэтому для сохранения прибыли Tailspin должна строго управлять расходами на эксплуатацию служб, предлагаемых клиентам.

*В этом сценарии клиенты Tailspin (подписчики) **не являются** клиентами Windows Azure. Подписчики платят компании Tailspin, которая в свою очередь платит Майкрософт за использование функций Windows Azure.*

Tailspin хочет обеспечить безопасное хранение клиентских данных. Например, сведения о клиенте должны быть доступны только самому клиенту, должно существовать несколько физических копий данных опроса, а клиенты должны быть защищены от возможности случайного удаления данных опроса. Кроме того, необходимо сохранять все существующие данные опросов всякий раз, когда Tailspin обновляет приложение.

Так как количество подписчиков растет, Tailspin хочет повысить надежность приложения Surveys для соблюдения своих соглашений об уровне обслуживания. Это особенно важно для некоторых крупных клиентов компании Tailspin.

Наконец, компания Tailspin хотела бы иметь возможность использовать существующие навыки своих разработчиков, свести к минимуму переобучение, необходимое для создания приложения Surveys, и упростить для разработчиков использование полученного ими опыта в рамках работы над приложением Surveys при работе с другими проектами Tailspin.

## Архитектура приложения Surveys

Для достижения целей, поставленных при разработке приложения Surveys, Tailspin реализовала приложение как облачную службу на основе Windows Azure. На рис. 2 показано общее представление этой архитектуры.

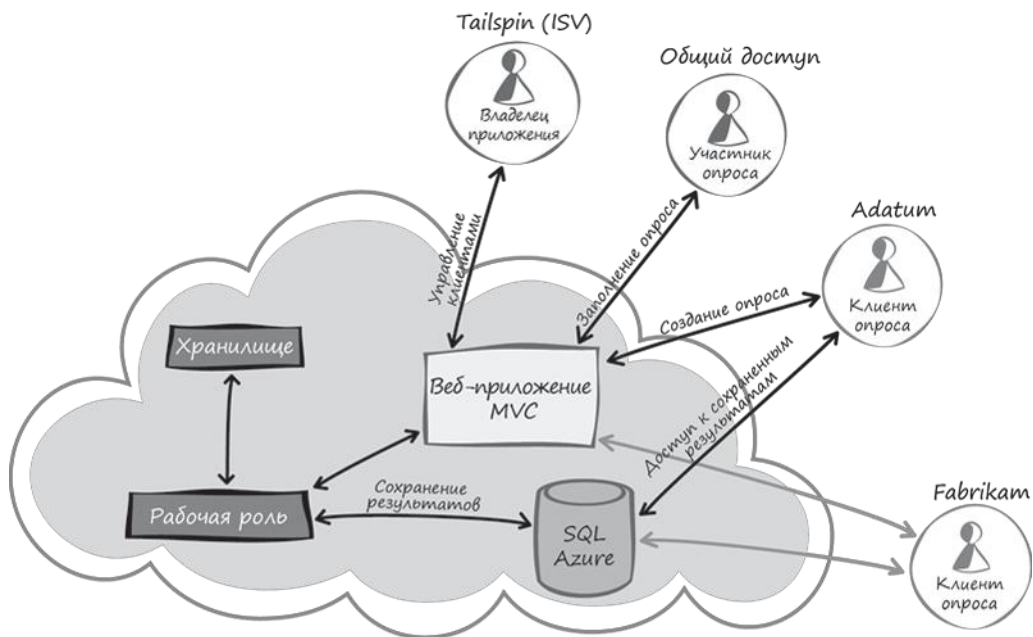


Рис. 2  
Архитектура приложения Surveys

Архитектура приложения Surveys проста и похожа на архитектуру, используемую во многих других приложениях Windows Azure. Ядро приложения использует веб-роли, рабочие роли и хранилище Windows Azure. На рис. 2 показаны три группы пользователей, использующих приложение: владелец приложения, посетители и подписчики Surveys (в этом примере Adatum и Fabrikam). Здесь также показано, как приложение использует технологическую платформу SQL Azure™ для создания механизма, позволяющего подписчикам отправлять результаты опросов в реляционную базу данных для детального анализа результатов.

В данном руководстве подробно описаны аспекты проектирования и разработки и показано, как различные веб- и рабочие роли, входящие в приложение Surveys, используют службы и программные блоки библиотеки Enterprise Library.

В руководстве охватываются некоторые конкретные вопросы, например реализация функции автоматического масштабирования, которая позволяет автоматически увеличивать или уменьшать масштабы приложения за счет добавления и удаления экземпляров ролей, а также обсуждается способ повышения устойчивости приложения к временным отказам.

Компания Tailspin разработала приложение с использованием самых современных технологий: Visual Studio 2010, ASP.NET MVC 3.0 и .NET Framework 4.



## Дополнительные сведения

Сведения о построении клиентского приложения Windows Phone 7 для приложения Tailspin Surveys см. в книге Руководство разработчика для Windows Phone 7 по адресу <http://go.microsoft.com/fwlink/?LinkID=234571>

# 4 Автоматическое масштабирование и Windows Azure

## Что такое автоматическое масштабирование

Одно из главных преимуществ, предлагаемых платформой Windows Azure™, — это способность быстро масштабировать облачное приложение в соответствии с меняющейся нагрузкой.

При развертывании приложения в Windows Azure развертываются роли: веб-роли для внешней части приложений и рабочие роли для выполнения внутренней работы. Во время работы приложения в Windows Azure эксплуатируются экземпляры созданных ролей (можно считать, что экземпляр роли — это виртуальная машина). Можно указать количество экземпляров для каждой роли. Чем больше экземпляров, тем больше вычислительной мощности предоставляется роли, но и затраты будут соответственно выше. Безусловно, имеются некоторые проектные требования, связанные с правильной работой нескольких экземпляров роли одновременно, но Windows Azure берет на себя обеспечение остальных требований к инфраструктуре. Дополнительные сведения о проектных требованиях см. в разделе «Построение масштабируемого, многопользовательского приложения для Windows Azure».

При первом развертывании приложения в Windows Azure можно указать размер и количество экземпляров, необходимых для каждой роли. Во время выполнения приложения на портале Windows Azure можно также динамически добавлять или удалять экземпляры роли вручную или программным путем с помощью API-интерфейса управления Windows Azure.

Добавляя или удаляя экземпляры роли для приложения Windows Azure во время его выполнения, можно согласовывать производительность приложения и эксплуатационные расходы. Предусмотрена возможность добавлять новые экземпляры, когда нагрузка высока, и удалять их, когда они больше не нужны (это ведет к уменьшению эксплуатационных расходов).

Если масштабирование приложения выполняется вручную, то не всегда удается добиться оптимального баланса между затратами и производительностью. Оператор может реагировать с опозданием или неправильно оценивать количество экземпляров роли, необходимых для поддержания пропускной способности.

*Масштабируемость — это ключевая функциональная возможность платформы Windows Azure.*



Масштабирование путем добавления дополнительных экземпляров часто именуется *горизонтальным масштабированием*. Windows Azure также поддерживает *вертикальное масштабирование*, используя большие экземпляры роли вместо дополнительных экземпляров роли.



Необходимо также рассмотреть вопрос о стоимости выполнения этой задачи операторами-людьми, особенно если имеются сотни или даже тысячи экземпляров роли, выполняющихся в центрах обработки данных Windows Azure по всему миру.

*Необходимо регулярно оценивать поведение автомасштабирования. Даже при масштабировании на месте не рекомендуется забывать о контроле приложения после запуска.*



Блок автомасштабирования приложения имеет с другими компонентами Enterprise Library блока приложения много общих конструктивных особенностей, например способ его настройки и использования в коде.

Автоматическое масштабирование уменьшает объем работы, выполняемой вручную при динамическом масштабировании приложения. Осуществляется это двумя способами: либо заранее устанавливаются ограничения на количество экземпляров роли на основе временного графика, либо количество ролей изменяется в зависимости от некоторых счетчиков или показателей, которые можно получать из приложения или из среды Windows Azure.

Для достижения оптимального баланса между расходами и производительностью необходимо регулярно оценивать результаты работы решения по автоматическому масштабированию. Рабочая среда вряд ли будет статичной. В целом можно ожидать изменения количества пользователей и типичных сценариев использования приложения, производительность приложения может измениться при увеличении объема данных, или приложение может быть развернуто в дополнительных центрах обработки данных Windows Azure.

Масштабирование приложения путем изменения количества экземпляров роли может и не быть единственным (или лучшим) способом масштабирования приложения. Например, может потребоваться изменить поведение приложения при всплесках нагрузки либо изменить количество очередей Windows Azure или размер базы данных SQL Azure. Автоматическое масштабирование не ограничивается только корректировкой количества экземпляров ролей.

## Что такое программный блок автоматического масштабирования

Программный блок автоматического масштабирования («Wasabi») является частью пакета интеграции библиотеки Enterprise Library для Windows Azure.

Данный блок позволяет определить, как приложение Windows Azure будет автоматически обрабатывать изменения уровня нагрузки. Он помогает свести к минимуму эксплуатационные расходы, в то же время обеспечивая высокую производительность и доступность приложения для пользователей. Он также помогает уменьшить количество задач, выполняемых операторами вручную.

Программный блок работает на основе набора пользовательских правил, которые управляют поведением приложения при изменении нагрузки. Правила являются *правилами ограничения*, задающими минимальное и максимальное количество экземпляров роли в приложении Windows Azure, или *правилами реагирования*, которые изменяют текущее количество экземпляров роли на основе счетчиков или показателей, полученных из приложения.

Для правил ограничения может задаваться связанный график, в котором указывается время активности правил. Правила ограничения позволяют заранее задать количество экземпляров роли, которые может использовать приложение; минимальное количество экземпляров роли помогает соблюдать соглашения об уровне обслуживания (SLA), максимальное число экземпляров роли помогает управлять эксплуатационными расходами для приложения Windows Azure.

Правила реагирования используют значения, производные от системных показателей, таких как коэффициент использования ЦП, либо от бизнес-показателей, таких как количество необработанных документов в приложении. Программный блок собирает эти показатели и сохраняет их в виде точек данных. Точка данных — это просто измеренное значение и соответствующая отметка времени, указывающая, когда блок получил значение. Правило реагирования использует агрегатное значение (например, среднее, максимум, минимум или последнее), рассчитанное на основе точек данных за определенный период. Правило реагирования сравнивает текущее агрегатное значение и пороговое значение и на основе результата выполняет одно или несколько действий, например добавляет два новых экземпляра веб-роли и уведомляет оператора. Правила реагирования помогают приложению реагировать на непредвиденные всплески (или падения) рабочей нагрузки.

Блок автоматического масштабирования поддерживает следующие методы для обработки различных уровней нагрузки.

- **Масштабирование экземпляров.** Программный блок автоматического масштабирования изменяет количество экземпляров роли в соответствии с изменяющейся нагрузкой в приложении.
- **Регулирование количества запросов.** Программный блок автоматического масштабирования приложения ограничивает или отключает некоторые относительно затратные операции приложения, когда нагрузка превышает определенные пороговые значения.

Эти два метода автоматического масштабирования не являются взаимоисключающими, и их можно использовать совместно для реализации гибридного решения по автоматическому масштабированию приложения.

На рис. 1 показана связь между программным блоком автоматического масштабирования и приложением Windows Azure.



Правила хранятся в XML-документах. Это позволяет их легко изменять. Это также делает возможным построение в приложении пользовательских редакторов для правил. Приложение Tailspin Surveys показывает, как это можно сделать.



В Windows Azure изменение числа экземпляров роли требует определенного времени, поэтому для быстрого реагирования может понадобиться регулировать приложение до тех пор, пока не будут доступны новые экземпляры роли.

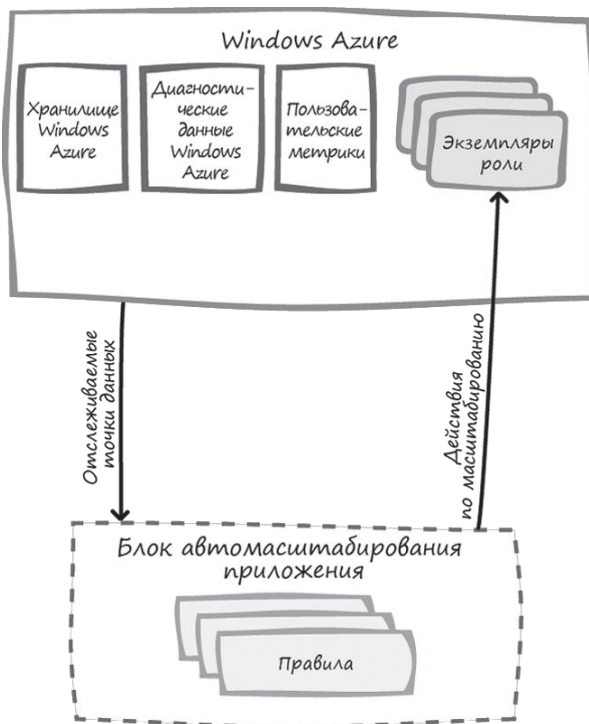


Рис. 1  
Программный блок автоматического масштабирования и Windows Azure

На диаграмме показано, как программный блок автоматического масштабирования собирает данные из среды Windows Azure и использует их в правилах для определения того, нужно ли инициировать действия по масштабированию приложения Windows Azure.

*Блок автоматического масштабирования может размещаться локально или в Windows Azure.*



Важно управлять затратами на выполнение приложения, а сохранение минимального количества экземпляров роли позволяет достичь этой цели.

## Масштабирование экземпляров

Блок автоматического масштабирования приложения позволяет автоматически масштабировать число экземпляров ролей Windows Azure (веб- и рабочих ролей) для полного соответствия требованиям приложения. Это эффективный метод контроля над эксплуатационными расходами приложения, поскольку в Windows Azure счет выставляется только за фактически используемые экземпляры.

Конечно, очень важно задать явные границы автоматического масштабирования для приложения Windows Azure. Так как плата взимается за каждый выделенный экземпляр роли (независимо от того, выполняется ли он или остановлен), следует задать максимальное количество экземпляров для каждого типа роли в приложении. В противном случае ошибка в приложении, приводящая

к увеличению количества экземпляров роли, может обернуться значительной (и непредвиденной) суммой, выставленной в конце месяца. Также можно задать минимальное количество экземпляров ролей, чтобы обеспечить выполнение приложения и его устойчивость к любым сбоям.

*Необходимо применять не менее двух экземпляров роли, чтобы иметь право на гарантии по соглашению об уровне обслуживания Windows Azure.*

Не следует ожидать, что программный блок автоматического масштабирования запустит новые экземпляры роли моментально. В Windows Azure запуск (или прекращение работы) экземпляра роли занимает некоторое время. Обычно этот процесс занимает до 10 минут (по состоянию на момент написания данного руководства), но это время может варьироваться в зависимости от целого ряда факторов, например: от количества добавляемых экземпляров роли, размера добавляемых экземпляров роли и текущего уровня загрузки в центре обработки данных Windows Azure.

*На момент написания этого документа за неполный час выполнения вычислительной операции счет выставляется за полный календарный час развертывания экземпляра. Например, при развертывании вычислительного экземпляра Small в 10:50 и удалении развертывания в 11:10 будет выставлен счет на два часа вычислений Small — один час вычислений за использование в период от 10:50 до 11:00 и еще один час вычислений за использование в период от 11:00 до 11:10. Таким образом, имеет смысл сохранять новые экземпляры действующими в оставшуюся часть часа, в течение которого они были запущены. Дополнительные сведения см. в разделе «Сведения о взимании платы за использование ресурсов Windows Azure».*

*Стабилизатор учитывает эти факторы при активации правил реагирования (описано ниже), но вам следует рассмотреть вопрос о разработке правил ограничения (также объяснено ниже), чтобы они выполняли сокращение лишних экземпляров незадолго до окончания часа.*



Важно внимательно выбирать объекты и цели регулирования. Пользователи будут ожидать, что основные функции приложения будут доступны в любое время.

## Регулирование приложения

Вместо корректировки количества экземпляров ролей в ответ на изменения нагрузки, блок автоматического масштабирования можно использовать для изменения поведения приложения в различных условиях. Этот метод позволяет задать режимы работы, которые соответствуют определенным уровням нагрузки, времени дня или типу пользователя.

Например, можно определить различные режимы работы для нормального функционирования, для режима низкой нагрузки или для случаев экстремальной нагрузки.

- Когда нагрузка на приложение очень низка, можно запускать необходимые фоновые задачи, которые не критичны по времени,



В зависимости от размера и сложности приложения, регулирование не может производиться быстрее, чем добавление нового экземпляра. Поэтому необходимо протестировать его в конкретной среде. Необходимо помнить, что не все экземпляры будут реагировать на регулирование одновременно.



Для обеспечения соблюдения соглашений об уровне обслуживания необходимо задать минимальное значение. Для ограничения затрат и достижения бюджетных целей необходимо задать максимальное значение.

но требуют больших ресурсов для выполнения, например экспорт данных или расчет статистики. Если соглашение об уровне обслуживания требует иметь минимум два экземпляра роли для выполнения приложения, то этот способ можно использовать для выполнения фоновых задач на этих экземплярах.

- При обычной нагрузке может потребоваться исключить выполнение фоновых задач, но обеспечить выполнение всех остальных функций.
- При экстремальных нагрузках, возможно, потребуется отключение определенных функций, чтобы приложение оставалось доступным для пользователей. Например, можно отключить функции автозаполнения, переключиться на упрощенную версию пользовательского интерфейса или отключить некоторые функции для пользователей, использующих пробный период приложения, в то же время обеспечивая полную поддержку клиентам, заплатившим за подписку.

Регулирование приложения можно очень эффективно использовать в сочетании с масштабированием экземпляров. В Windows Azure на добавление нового экземпляра роли может уйти до 10 минут, поэтому, когда происходит внезапный всплеск нагрузки, можно использовать регулирование приложения, чтобы снизить нагрузку на приложение в ожидании запуска нового экземпляра роли. Однако, если имеется большое количество экземпляров, может потребоваться время, чтобы изменения конфигурации распространились на все экземпляры, а за это время могут успеть запуститься новые экземпляры. Кроме того, если приложение уже расширено до максимального количества экземпляров роли, допускаемых правилами ограничения, то регулирование приложения поможет обеспечить максимальную производительность основных функций приложения.

## Правила и действия

Программный блок автоматического масштабирования использует правила и действия для определения реакции приложения на изменения нагрузки. Как описано ранее, существует два типа правил: правила ограничения и правила реагирования, которые имеют собственные действия.

## Правила ограничения

Для многих приложений модель нагрузки является прогнозируемой. Например, в бизнес-приложении высокие нагрузки приходятся на рабочие часы; на потребительском веб-сайте высшая нагрузка наблюдается с 18:00 до 20:00. В этих сценариях можно заранее масштабировать приложение Windows Azure в соответствии с ожидаемым уровнем рабочей нагрузки. В подобных случаях можно использовать правила ограничения.

Правила ограничения состоят из одного или нескольких действий, задающих минимальные и максимальные значения для целевого количества экземпляров, ранг и при необходимости расписание,

определяющее время применения правила. Если расписания нет, правило действует всегда.

Расписание можно использовать для управления количеством экземпляров роли, которые должны быть доступны в определенное время. Например, можно создать правило для увеличения минимального и максимального числа экземпляров веб- и рабочих ролей в приложении с 9:00 до 11:00 в понедельник утром, когда известно, что нагрузка на приложение будет выше обычной.

Можно также указать правила по умолчанию, которые активны всегда и определяют максимальное и минимальное значения по умолчанию для количества экземпляров ролей. Важно отметить, что **правила ограничения всегда имеют приоритет над правилами реагирования**, чтобы правила реагирования не могли добавлять новые экземпляры роли, превышающие максимальное значение, или удалять экземпляры, когда достигнут минимальный уровень.

*По умолчанию (ко времени написания этого документа) подписки Windows Azure могут использовать до 20 ядер ЦП. Это значение может быть увеличено по запросу. Дополнительные сведения см. на странице **Поддержка Windows Azure**.*

Возможно, что несколько правил ограничения будут действовать одновременно, когда их расписания перекрываются. В таком случае программный блок автоматического масштабирования использует *ранг* правил, чтобы определить, какое правило имеет приоритет. Правила высокого ранга перекрывают правила с более низким рангом.

Ниже приведены некоторые примеры правил ограничения.

- Для веб-роли A минимальное количество экземпляров по умолчанию устанавливается равным двум, а максимальное — четырем. Это правило имеет ранг по умолчанию.
- По относительно загруженным пятницам минимальное количество экземпляров веб-роли A равняется четырем и максимум восемь. Это правило имеет более высокий ранг.
- Для рабочей роли B задается ограничение минимум два экземпляра и не более четырех экземпляров по умолчанию.
- По субботам и воскресениям между 14:00 и 18:00 рабочая роль В может иметь минимум три экземпляра, а максимум — шесть.
- В последнюю пятницу каждого месяца для группы масштабирования A задается использование минимум трех экземпляров и максимум шести экземпляров (группы масштабирования описываются далее в этом разделе).

На рис. 2 показано поведение программного блока автоматического масштабирования, когда задано несколько правил ограничения. В сценарии, отображенном на этой диаграмме, используются три отдельных правила ограничения для определения количества экземпляров рабочей роли A в конкретном приложении. В этом простом сценарии нет никаких правил реагирования.



Единица соответствует самому низкому рангу. Этот ранг следует использовать для всех правил по умолчанию. Следует всегда присваивать ранг правилам ограничения таким образом, чтобы было ясно, какое из них должно иметь приоритет.

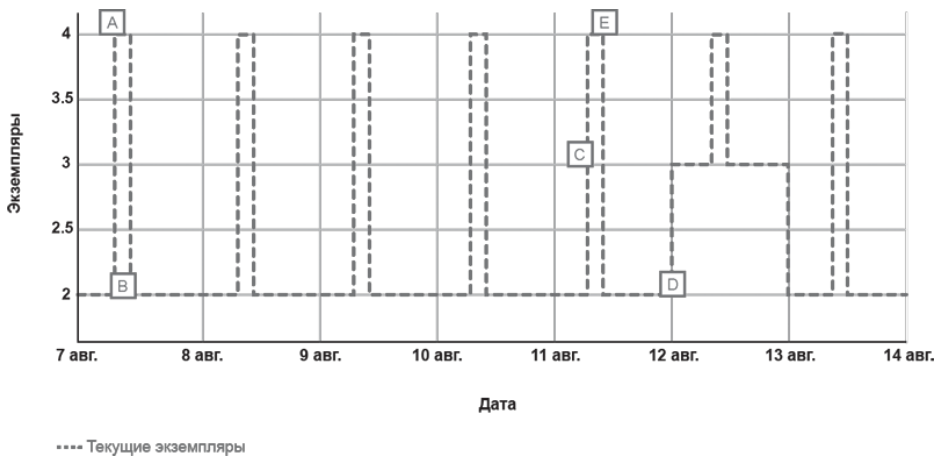


Рис. 2

Использование нескольких правил ограничения при отсутствии правил реагирования

Эти три правила ограничения в данном сценарии фактически работают следующим образом.

1. Первое правило ограничения всегда активно. Оно задает для рабочей роли А максимальное и минимальное количество экземпляров, равное двум. Это правило имеет самый низкий ранг.
2. Второе правило ограничения активно каждые сутки от 8:00 до 10:00. На этой диаграмме метка А показывает, в какое время 7 августа это правило стало активным, а метка В показывает, когда правило стало неактивным. Оно задает максимальное и минимальное количество экземпляров для рабочей роли А, равное четырем. У него самый высокий ранг среди этих трех правил, поэтому при одновременном применении с ними оно получает приоритет.
3. Третье правило ограничения активно каждую пятницу (на диаграмме 12 августа — пятница, а метки С и D показывают, когда это правило становилось активным и неактивным). Оно задает максимальное и минимальное количество экземпляров для рабочей роли А, равное трем. Оно имеет более низкий ранг, чем второе правило, поэтому каждую пятницу с 8:00 до 10:00 второе правило переопределяет это правило; на диаграмме метка Е показывает, когда это происходит.

На рис. 3 показан эффект использования нескольких правил ограничения с различными максимальными и минимальными значениями, но без каких-либо правил реагирования.



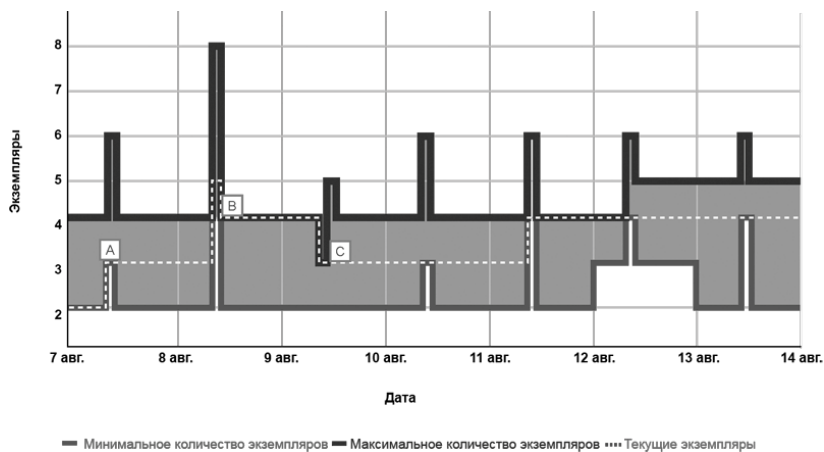


Рис. 3  
Использование нескольких правил ограничения с разными максимальными и минимальными значениями, но без правил реагирования.

В этом сценарии используются несколько правил ограничения с различными максимальными и минимальными значениями. Можно видеть, что количество экземпляров всегда остается между минимальным и максимальным пределами.

Алгоритм согласования, используемый блоком автомасштабирования приложения при оценке правил ограничения, действует следующим образом.

- Если текущее количество экземпляров меньше минимального количества экземпляров, указанного правилами ограничения в текущее время, увеличивает текущее количество экземпляров до минимального значения. На диаграмме это происходит в метке А.
- Если текущее количество экземпляров больше максимального количества экземпляров, указанного правилами ограничения в текущее время, уменьшает текущее количество экземпляров до максимального значения. На диаграмме это происходит в метке В.
- В противном случае текущее количество экземпляров не меняется. На диаграмме это происходит в метке С.

### Правила реагирования

Не всегда можно спрогнозировать, в какой момент времени возрастет потребность в конкретном приложении или когда возникнет ее резкий всплеск. Блок автомасштабирования приложения также позволяет создавать правила реагирования, которые вызывают действие масштабирования, если агрегатное значение, полученное из набора точек данных, превышает заданное пороговое значение.

Блок автомасштабирования приложения позволяет следить за значениями счетчиков производительности, длинами очередей Windows Azure, количеством экземпляров, а также любыми специализированными бизнес-показателями в целях масштабирования

приложения при превышении этими значениями заданных порогов. В блоке приложения эти значения рассматриваются как *операнды*, причем каждый операнд определяет три вещи.

- Счетчик или показатель.
- Агрегатную функцию, такую как среднее или максимум.
- Интервал времени, на котором блок приложения вычисляет агрегатную функцию.

Например, блок автомасштабирования приложения позволяет наблюдать за загрузкой ЦП конкретными экземплярами веб-роли. Если среднее значение счетчика производительности нагрузки ЦП в течение последнего часа превышает пороговое значение 80 %, правило выполняет действие по добавлению новых экземпляров веб-роли, позволяющих справиться с этой нагрузкой, что должно вызвать снижение средних уровней нагрузки ЦП (при условии, что нагрузка не выросла значительно). Добавление экземпляров веб-роли будет продолжаться до тех пор, пока средняя нагрузка ЦП не упадет ниже порогового значения. Действует также обратное правило. Например, если средняя нагрузка ЦП в течение последнего часа упадет ниже порогового значения 40 %, то правило будет выполнять действие по удалению экземпляров веб-роли до тех пор, пока средняя нагрузка процессора не станет выше порогового значения. Правила реагирования позволяют корректировать количество экземпляров роли на абсолютное число или в определенной пропорции.

Обычно правила реагирования объединены в пары — для увеличения (развертывания) масштаба и для уменьшения (свертывания) масштаба.

В правилах реагирования используется выражение для условия, которое должно проверяться при принятии решения, должно ли правило выполнить действие масштабирования. Правила реагирования могут запускать следующие действия.

- Изменение количества экземпляров целевого объекта правила. Действие может увеличить или уменьшить это количество на определенное число или в некоторой пропорции.
- Изменение настройки размещенной службы. Это действие устанавливает новые значения в записях файла приложения `ServiceConfiguration.cscfg`.
- Отправка уведомления оператору.
- Переход в другой режим работы, если для приложения было предусмотрено его регулирование.
- Выполнение настраиваемого действия.

Действие сформирует уведомление, если оно окончится неудачей.

*Если приложение использует несколько веб-ролей и рабочих ролей, необходимо задать действие для каждой роли, которую предполагается масштабировать. Для упрощения этой задачи можно использовать группы масштабирования.*

Ниже приведены примеры правил реагирования.

- Выполнить действие, если счетчик производительности

нагрузки ЦП, усредненный за последний час по всем экземплярам рабочей роли A, превышает 80 %.

- Выполнить действие, если минимальная длина очереди Windows Azure за последние шесть часов превышает 50.

Правила могут содержать простое логическое выражение, в котором единственное значение сравнивается с пороговым значением, или сложное выражение, которое включает несколько связанных логическими операциями сравнений с несколькими операндами. Примером правила со сложным выражением является следующее.

- Выполнить действие, если усредненная за последний час нагрузка ЦП возростала, а длина очереди в течение последних двух часов оставалась выше 120.

На рис. 4 показано поведение блока автомасштабирования приложения, если в дополнение к нескольким правилам ограничения определено правило реагирования. В сценарии, отображенном на этой диаграмме, используются три отдельных правила ограничения для определения минимального и максимального количества экземпляров рабочей роли A в конкретном приложении.

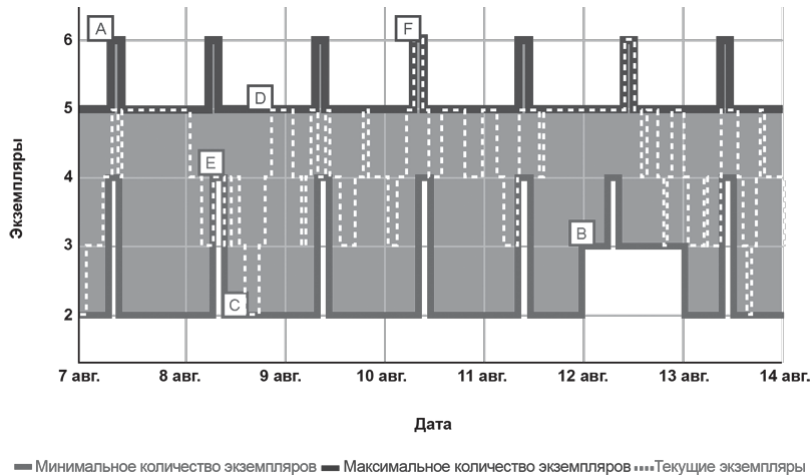


Рис. 4

Правила ограничения, взаимодействующие с правилами реагирования

Для рабочей роли A заданы три следующих правила ограничения.

1. Первое правило ограничения всегда активно. Оно задает для рабочей роли A минимальное количество экземпляров, равное двум, а максимальное — равное пяти. Это правило имеет самый низкий ранг.
2. Второе правило ограничения активно каждые сутки с 8:00 до 10:00 (метка A на диаграмме показывает, когда это правило становится активным впервые). Оно задает для рабочей роли A минимальное количество экземпляров, равное четырем, а максимальное — равное шести. Оно обладает самым

высоким рангом среди этих трех правил, поэтому при одновременном применении с другими правилами получает приоритет над ними.

3. Третье правило ограничения активно каждую пятницу (на диаграмме 12 августа — пятница, а метка В на диаграмме показывает, когда это правило становится активным). Оно задает для рабочей роли А минимальное количество экземпляров, равное трем, а максимальное — равное пяти. Оно имеет более низкий ранг, чем второе правило, поэтому каждую пятницу с 8:00 до 10:00 второе правило будет перепределять это правило.

Кроме того, предусмотрены два правила реагирования, которые могут корректировать количество экземпляров рабочей роли А.

- Если минимальное количество необработанных документов в течение последнего часа было больше 10, увеличить количество экземпляров рабочей роли А на единицу.
- Если максимальное количество необработанных документов в течение последнего часа было меньше 10, уменьшить количество экземпляров рабочей роли А на единицу.

В сценарии, показанном на рис. 4, можно видеть, что правила ограничения всегда ограничивают количество экземпляров, предоставляя абсолютные значения ближайшего меньшего целого числа и ближайшего большего целого числа, которые нельзя пересекать. Правила реагирования позволяют корректировать количество экземпляров роли в пределах этих лимитов. На этой диаграмме метки С и D показывают те моменты времени, в которые для количества экземпляров, предложенного правилом реагирования, принудительно устанавливались пределы согласно первому правилу ограничения. Метки Е и F показывают те моменты времени, в которые для количества экземпляров, предложенного правилом реагирования, принудительно устанавливались пределы согласно второму правилу ограничения; в эти моменты второе правило ограничения перепределяло первое правило ограничения.

*Если работает процесс оценки для правила, но отсутствует активное правило ограничения для роли, а в правиле реагирования предпринимается попытка изменить количество экземпляров роли, то блок автомасштабирования приложения записывает в журнал сообщение о том, что не может применить к роли никаких действий масштабирования. Блок не изменяет текущее количество экземпляров роли.*

При наличии нескольких правил реагирования одновременно могут вызываться несколько различных конфликтующих друг с другом действий. В этом случае блок автомасштабирования приложения может урегулировать конфликтующие действия.

Дополнительные сведения о том, как блок урегулирует конфликтующие правила, см. в разделе «Основные сведения о рангах и согласовании правил» на сайте MSDN.



Правило может инициировать одно или несколько действий.

При присвоении рангов правилам реагирования следует проявлять осторожность. Для определения действия масштабирования, которое должно быть выполнено, лучше использовать процесс согласования.

## Ведение журнала

Независимо от того, используется ли автомасштабирование экземпляра, регулирование приложения или сочетание этих двух подходов, блок автомасштабирования приложения может записывать в журнал информацию об этих действиях. Например, запись журнала может быть зарегистрирована в следующих случаях.

- Если происходит запуск или прекращение работы новых экземпляров, записываются сведения о том, по какой причине блок автомасштабирования приложения добавил этот экземпляр.
- Если приложение переключилось с одного режима работы на другой, записываются сведения о том, что вызвало такое регулирование.

Эту информацию можно использовать в качестве вспомогательной при анализе затрат на Windows Azure, можно также выявлять с ее помощью прогнозируемые шаблоны в уровнях использования конкретного приложения.

## Жизненный цикл автомасштабирования

На рис. 5 показан жизненный цикл процесса автомасштабирования с точки зрения эксплуатационного персонала.

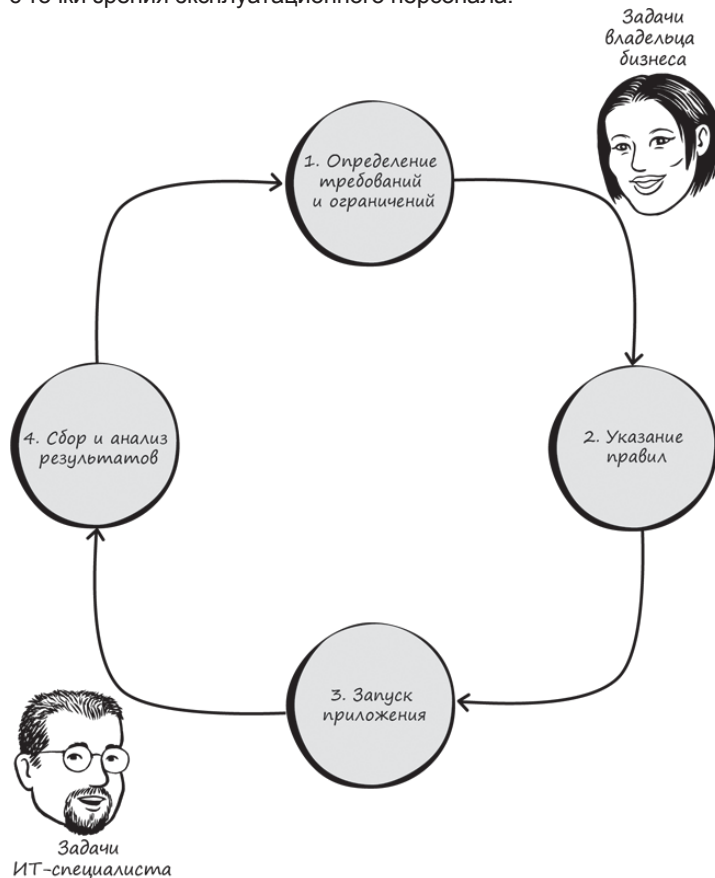


Рис. 5

Жизненный цикл процесса автомасштабирования состоит из четырех этапов, которые могут быть повторены эксплуатационным персоналом много раз по мере усовершенствования ими функции автомасштабирования конкретного приложения.

### ***Определение требований и ограничений***

На первом этапе необходимо определить требования и ограничения в конкретном приложении для функции автомасштабирования. Для определения двух типов требований необходимо выполнить следующее.

- Выявить все прогнозируемые шаблоны потребностей в службах конкретного приложения.
- Указать, как должно приложение отвечать на непредсказуемые резкие повышения и спады потребности в его службах.

Можно столкнуться со следующими ограничениями:

- Бюджетные ограничения на эксплуатационные расходы приложения Windows Azure.
- Любые обязательства по соблюдению соглашения об уровне обслуживания с пользователями конкретного приложения.

### ***Определение правил***

На основании требований и ограничений, выявленных на предыдущем шаге, необходимо с учетом этих ограничений сформулировать набор правил автомасштабирования для приложения. Можно использовать правила ограничения, чтобы определить поведение приложения в ответ на прогнозируемые изменения спроса, и правила реагирования, чтобы определить поведение приложения в ответ на непредсказуемые всплески спроса.

### ***Выполнение приложения***

После настройки правил блок автомасштабирования приложения может оценивать правила и выполнять действия автомасштабирования в конкретном приложении по мере того, как приложение будет сталкиваться с реальными изменениями спроса. Блок автомасштабирования приложения записывает в журнал результаты оценки правила и выполняемые им действия автомасштабирования.

### ***Сбор и анализ результатов***

Необходимо регулярно анализировать информацию о действиях блока автомасштабирования приложения, внесенную им в журнал, чтобы оценить, в какой степени применяемые правила отвечают заданным начальным требованиям и работают в пределах ограничений. Например, может выясниться, что применяемые правила не всегда обеспечивают успешное масштабирование приложения для удовлетворения спроса или что правила не при всех обстоятельствах обеспечивают выполнение требований соглашения об уровне обслуживания. В таких случаях следует пересмотреть применяемые требования и ограничения, чтобы гарантировать их актуальность, и при необходимости скорректировать свои правила. Вы сможете выявлять

новые, прогнозируемые закономерности использования, что позволит заблаговременно масштабировать приложение вместо того, чтобы полагаться на правила реагирования.

Необходимо продолжать циклическое повторение этого процесса, поскольку закономерности использования для конкретного приложения со временем будут изменяться и существующий набор правил может стать неоптимальным для заданных требований и ограничений.

## Когда следует использовать блок автомасштабирования приложения?

В этом разделе описаны три сценария, в которых необходимо рассмотреть возможность использования блока автомасштабирования приложения в решении Windows Azure.

### Требуется, чтобы приложение автоматически отвечало на изменение в потребности

Блок автомасштабирования приложения позволяет справляться с двумя конкурирующими требованиями в приложениях Windows Azure. Первое состоит в поддержании уровня производительности приложения в условиях возможного изменения уровней потребности. Если в веб- или рабочих ролях приложения со временем происходят изменения рабочей нагрузки, существенно варьирующиеся в зависимости от часов, дней или недель, и требуется, чтобы приложение отвечало на эти изменения потребности автоматически, то блок автомасштабирования приложения может увеличивать или уменьшать количество экземпляров роли автоматически с учетом правил, заранее заданных в конфигурации.

Для запуска новых экземпляров роли может потребоваться по меньшей мере 10 минут, поэтому можно также использовать функцию регулирования приложения, предоставляемую блоком автомасштабирования приложения, когда ответ на резкое повышение активности должен быть дан достаточно быстро (в течение секунд или минут).

### Необходимо управлять издержками, связанными с эксплуатацией конкретного приложения

Второе, конкурирующее требование состоит в снижении до минимума расходов на эксплуатацию приложения Windows Azure. Безусловно, дополнительные экземпляры веб- и рабочей роли позволяют обеспечивать необходимые значения времени ответа для пользователей и поддерживать пропускную способность для фоновых задач при резком повышении активности, но за эти дополнительные экземпляры роли приходится платить. Система Windows Azure предусматривает почасовую оплату за экземпляры веб- и рабочей роли, и эти расчетные издержки, как правило, составляют значительную часть расходов на эксплуатацию приложения Windows Azure. Более подробное обсуждение методов оценки своих эксплуатационных расходов для Windows Azure см. в главе «Сколько будет это стоить?» в книге *«Перемещение приложений в облако»*.



Чтобы пользователи продолжали использовать ваше приложение, необходимо всегда быстро давать ответы на их



Рентабельность приложения напрямую зависит от его эксплуатационных расходов.



Для прогнозирования рабочей нагрузки можно собирать и анализировать исторические данные и использовать свои знания внешних факторов, которые инициируют изменения спроса.

Блок автомасштабирования приложения позволяет управлять издержками, удаляя ненужные экземпляры роли и предоставляя возможность задавать максимальные значения количества экземпляров роли. Но могут возникать обстоятельства, при которых приложение обнаруживает дополнительное резкое повышение активности, притом что оно уже эксплуатируется с максимально заданным в конфигурации количеством экземпляров. В этом случае приложение может отвечать с использованием регулирования приложения. Правила регулирования могут определять, когда конкретное приложение должно перейти в режим работы, предусматривающий меньшее потребление ресурсов, или отключить некритические функциональные возможности. Таким образом, приложение может поддерживать высокое быстродействие своего пользовательского интерфейса или необходимую пропускную способность для важных процессов, не запуская дополнительные экземпляры роли.

### **Значения времени, в которые приложению потребуются дополнительные ресурсы, являются прогнозируемыми**

Правила, используемые блоком автомасштабирования приложения, позволяют определить, когда количество экземпляров роли должно увеличиться или уменьшиться. Если заранее известно, что вскоре произойдет резкое повышение потребности, то можно запустить дополнительные экземпляры роли, прежде чем произойдет это резкое повышение, с помощью правил автомасштабирования, определяя тем самым расписание, указывающее количество ролей, которые должны быть доступными в конкретные моменты времени.

### **Обстоятельства, в которых не следует использовать блок автомасштабирования приложения**

В некоторых сценариях использовать блок автомасштабирования приложения в конкретном приложении Windows Azure не следует.



### **Простые приложения**

Автомасштабирование часто не позволяет существенно повысить значение относительно простых приложений или приложений, имеющих ограниченное количество пользователей. Например, для многих небольших веб-приложений может никогда не потребоваться больше двух экземпляров веб-роли, даже во время резкого повышения активности.

Следует стремиться проектировать приложения, для которых сейчас для него не требуется масштабирование. Если приложение невозможно сделать масштабируемым, то его необходимо переписать заново.



Добавление в приложение блока автомасштабирования приложения приводит к повышению сложности этого приложения. Поэтому необходимо оценить, оправдывают ли преимущества добавления функций автомасштабирования ту дополнительную сложность, которая вводится в макет приложения.

### Управление издержками

Если желательно, чтобы некоторые расходы на эксплуатацию приложения Windows Azure стали постоянными, то может потребоваться установить постоянное количество экземпляров роли в этом приложении. Это позволяет точно прогнозировать связанную с этим часть ежемесячного счета за использование Windows Azure. Но все расходы на Windows Azure не могут рассматриваться как постоянные, например, всегда меняются издержки на передачу данных и затраты на хранение в Windows Azure в зависимости от количества передаваемых и хранимых данных.

Приложения, которые не являются масштабируемыми. Автомасштабирование имеет смысл только для приложений, которые предназначены для использования в качестве масштабируемых. Если приложение не масштабируется по горизонтали, поскольку имеет такой проект, что вводом дополнительных экземпляров нельзя повысить его производительность, не следует использовать блок автомасштабирования приложения. Например, может оказаться немасштабируемой простая веб-роль, поскольку в ней используется реализация сеанса, не обеспечивающая успешную работу в веб-ферме. Обсуждение состояния сеанса в приложениях Windows Azure см. в разделе Хранение состояния сеанса книги *«Перевод приложений в облако, 2<sup>е</sup> издание»*. Обсуждение некоторых проблем проектирования, связанных с масштабируемыми рабочими ролями, см. в разделе Масштабирование приложений с использованием рабочих ролей книги *«Разработка приложений для облака, 2<sup>й</sup> выпуск»*.

## Использование блока автомасштабирования приложения

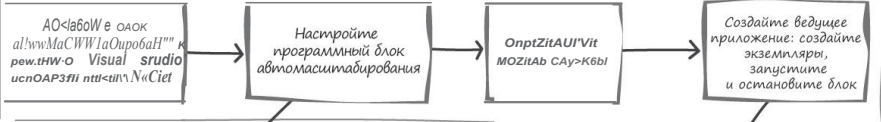
Использование блока автомасштабирования приложения предусматривает выполнение части задач разработчиками, а части — профессионалами ИТ. На рис. 6 показана связь ключевых задач с действиями блока автомасштабирования приложения в Windows Azure.

Блок автомасштабирования приложения автоматизирует процесс масштабирования для приложений, которые уже являются масштабируемыми. Использование блока автомасштабирования приложения не делает автоматически приложение масштабируемым.





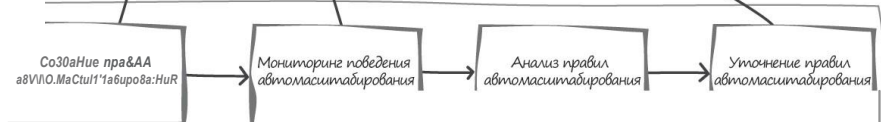
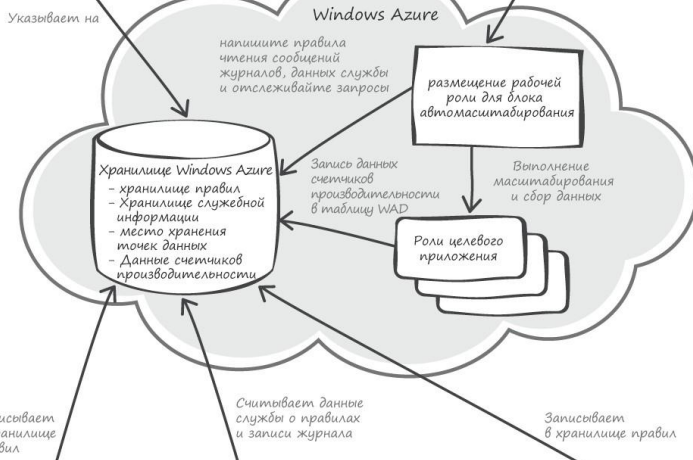
### Задачи разработчика



АаННЬь. КОНфигура. «и соОтр)ф(аW.

- Данные подключения учетной записи хранения
- management certificate
- Конфигурация ведения журналов

Выполните развертывание в Windows Azure



### Задачи ИТ-специалиста

В этом разделе описано с позиций высокого уровня, как использовать блок автомасштабирования приложения. Он состоит из следующих основных подразделов. Порядок этих разделов отражает тот порядок, в котором обычно выполняются соответствующие задачи. Часть из этих задач осуществляют разработчики, а другие — администраторы. Кто должен выполнять ту или иную задачу на практике, следует из описания самой задачи.

- **Добавление блока автомасштабирования приложения в проект Visual Studio.** В этом разделе приведено описание того, как разработчик может подготовить свое решение системы по разработке программ Microsoft Visual Studio® к использованию блока.
- **Размещение блока автомасштабирования приложения.** В этом разделе описано, как разработчик может разместить блок автомасштабирования приложения в конкретном приложении Windows Azure.
- **Изменения в приложении Windows Azure.** Этот раздел описывает, какие изменения должны быть внесены в конкретное приложение Windows Azure, чтобы оно работало с блоком автомасштабирования приложения.
- **Служебная информация.** В этом разделе показано, как разработчик должен определить служебную информацию для конкретного приложения.
- **Добавление функции регулирования в приложение.** Данный раздел описывает, какие изменения может внести разработчик в приложение, чтобы его можно было регулировать по заданным правилам автомасштабирования.
- **Правила автомасштабирования.** Этот раздел описывает, как администратор может определить правила автомасштабирования.
- **Мониторинг блока автомасштабирования приложения.** Этот раздел показывает, как администратор может наблюдать за применением правил автомасштабирования и использовать собранные данные.
- **Дополнительные сценарии использования.** Этот раздел описывает некоторые дополнительные сценарии, такие как использование групп масштабирования и расширение блока автомасштабирования приложения.

## Добавление блока автомасштабирования приложения в проект Visual Studio

Прежде чем получить возможность написания какого-либо кода, в котором используется блок автомасштабирования приложения, разработчик должен настроить проект Visual Studio с учетом всех необходимых сборок, ссылок и других ресурсов, которые могут потребоваться. Сведения о том, как использовать NuGet для подготовки проекта Visual Studio для работы с блоком автомасштабирования



Эта задача обычно выполняется при создании ведущего приложения для блока с привлечением ИТ-специалистов для определения требуемых функций.



NuGet очень облегчает настройку проекта со всеми необходимыми компонентами для использования блока автомасштабирования приложения.

Можно загрузить пакет NuGet, извлечь библиотеки DLL и добавить их в проект вручную или загрузить исходный код для блока приложения и построить его самостоятельно.



Необходимо решить, где должен быть размещен блок приложения: в Windows Azure или в локальном приложении.

*Блок автомасштабирования приложения предназначен для работы с очень большими приложениями Windows Azure с сотнями различных ролей.*

приложения, см. в разделе «Добавление блока автомасштабирования приложения к проекту Visual Studio» на сайте MSDN.

## Размещение блока автомасштабирования приложения

Блок автомасштабирования приложения можно разместить в роли Windows Azure или в таком эксплуатируемом на предприятии приложении, как простое консольное приложение или служба Windows. В этом разделе обсуждаются причины, по которым следует выбрать тот или иной подход, а также представлены ссылки на ресурсы с описанием того, как подготовить код, размещающий блок автомасштабирования приложения.

Блок автомасштабирования приложения позволяет добавлять функции автомасштабирования в приложения Windows Azure. Кроме того, этот блок должен взаимодействовать с Windows Azure для внесения изменений в те экземпляры роли, из которых состоит конкретное приложение. Приложение Windows Azure может представлять собой простое приложение, состоящее из небольшого количества ролей, из которых все выполняются в одной и той же размещенной службе одного и того же центра обработки данных, или иметь сотни разных ролей, эксплуатируемых в многочисленных службах, размещенных в многочисленных центрах обработки данных. Что касается структуры приложения и того, где следует разместить блок автомасштабирования приложения, необходимо иметь возможность взаимодействовать с приложением с использованием API Windows Azure Service Management, а для оценки правил реагирования требуется доступ к диагностическим данным, таким как значения счетчиков производительности.

Если блок автомасштабирования приложения размещается в Windows Azure, то нет необходимости передавать за пределы облака какие-либо используемые им данные. Но может потребоваться передавать между центрами обработки данных диагностические данные, если некоторые части приложения размещаются в других географических местоположениях. Преимущества размещения блока автомасштабирования приложения в Windows Azure совпадают с преимуществами размещения любого приложения в облаке: надежностью и масштабируемостью. Однако за размещение в Windows Azure роли, содержащей блок приложения, приходится платить. Можно разместить блок приложения в рабочей роли, которая выполняет также другие задачи, но с точки зрения управляемости и безопасности следует размещать блок приложения в его собственной рабочей роли или даже в его собственной размещенной службе. Сведения о том, как разместить блок автомасштабирования приложения в Windows Azure, см. в разделе «Размещение блока автомасштабирования приложения в рабочей роли» на сайте MSDN.



Использование блока автомасштабирования приложения в коде аналогично использованию других блоков приложения Enterprise Library. В разделе «Использование Enterprise Library в приложениях» в основной документации по Enterprise Library описывается, как обращаться к сборкам Enterprise Library, как Enterprise Library обрабатывает зависимости и как работать с объектами Enterprise Library.

Если будет решено разместить блок приложения в Windows Azure и запланировано масштабирование экземпляра роли, в котором размещается этот блок, для дополнительной надежности, то необходимо обеспечить настройку блока приложения на использование аренды большого двоичного объекта в дополнительных значениях параметров конфигурации. Этот параметр гарантирует то, что лишь один экземпляр блока приложения будет иметь возможность оценивать правила в любой момент времени. Сведения о том, как выполнить настройку значения этого параметра, см. в разделе «Ввод информации о конфигурации» на сайте MSDN.

*В параметрах конфигурации по умолчанию предполагается, что будет использоваться единственный экземпляр рабочей роли, в котором размещается блок приложения. Это значение необходимо изменить, если планируется масштабирование роли, которая размещает блок автомасштабирования приложения.*

Если блок приложения будет размещен на предприятии, то блок должен получать дистанционный доступ к диагностическим данным из приложения Windows Azure, которые требуются для правил реагирования. Преимущество локального размещения блока приложения состоит в том, что способствует упрощению интеграции с другими средствами и процессами, которые эксплуатируются на предприятии. Локальная эксплуатация блока автомасштабирования приложения может оказаться удобной, если ведется разработка и проверка приложения Windows Azure. Сведения о том, как разместить блок автомасштабирования приложения в локальном приложении, см. в разделе «Размещение блока автомасштабирования приложения в локальном приложении» на сайте MSDN.

## Изменения в приложении Windows Azure

Блок автомасштабирования приложения предназначен для минимизации изменений, которые должны быть внесены в приложение Windows Azure. Блок приложения позволяет добавлять и удалять экземпляры роли из приложения с использованием API Windows Azure Service Management. Для этого не требуется внесения каких-либо изменений в конкретном приложении.

Однако в правилах реагирования могут использоваться данные счетчика производительности для определения того, должен ли блок приложения изменять текущее количество экземпляров роли. Если в правилах реагирования используются счетчики производительности, то необходимо предпринять шаги по обеспечению того, чтобы в приложении данные счетчика производительности сохранялись в системе хранения Windows Azure, где к ним может получить доступ процесс сбора данных блока приложения.

Дополнительные сведения об изменениях в коде, которые необходимо внести в приложение Windows Azure для сохранения данных счетчика производительности, см. в разделе «Сбор данных счетчика производительности» на сайте MSDN.

Для настройки счетчиков производительности можно также использовать файл конфигурации системы диагностики Windows Azure (diagnostics.wadcfg). Дополнительные сведения см. в разделе



Можно также реализовать приложение Windows Azure с пользовательскими счетчиками производительности, чтобы использовать их в правилах реагирования.



Служебная информация определяет аспекты приложения Windows Azure, которые касаются блока автомасштабирования приложения.

«Как использовать файл конфигурации системы диагностики Windows Azure» на сайте MSDN.

## Служебная информация

Прежде чем появится возможность выполнять какие-либо операции автомасштабирования в конкретном приложении Windows Azure с помощью блока автомасштабирования приложения, следует настроить служебную информацию, которая описывает приложение Windows Azure. По умолчанию эта служебная информация хранится в XML-документе в большом двоичном объекте Windows Azure, который является доступным для блока приложения.

Служебная информация включает следующие сведения о функциях Windows Azure, которые составляют конкретное приложение.

- Для каждой подписки Windows Azure, содержащей ресурсы, которые необходимо масштабировать автоматически, служебная информация содержит идентификатор подписки, отпечаток сертификата и подробные сведения о том, где находится сертификат управления для блока приложения, требуемый ему для выдачи запросов масштабирования.
- Для каждой службы, размещенной в Windows Azure, содержащей ресурсы, которые необходимо масштабировать автоматически, служебная информация включает имена слотов развертывания, где функционирует масштабируемое приложение.
- Блок приложения может использовать только те роли Windows Azure, которые перечислены в служебной информации как источники данных счетчика производительности или как цели для автомасштабирования. Для каждой роли, перечисленной в служебной информации, эта служебная информация указывает учетную запись системы хранения, в которой Windows Azure сохраняет диагностические данные роли. Блок приложения считывает из этой учетной записи системы хранения данные счетчика производительности, используемые правилами реагирования.
- Имена всех очередей, длину которых контролирует блок приложения.
- Определения групп масштабирования. Они описаны ниже в этом разделе.

Правила блока автомасштабирования приложения могут действовать только применительно к целям (ролям и группам масштабирования), которые указаны в служебной информации блока приложения. Дополнительные сведения см. в разделе «*Хранение служебной информации*» на сайте MSDN.

Эта служебная информация позволяет также управлять тем, насколько агрессивно должно проводиться автомасштабирование приложения Windows Azure, путем указания периодов снижения активности. Период, наступающий после проведения операции масштабирования, в течение которого блок приложения больше не выполняет какие-либо операции масштабирования, называется *периодом снижения активности*. Период снижения активности включается через функцию

стабилизатора оптимизации блока приложения. Можно определять различные периоды снижения активности для операций увеличения масштаба и уменьшения масштаба, а также указывать предусмотренные по умолчанию периоды снижения активности, которые могут быть переопределены отдельными ролями. Чем короче период снижения активности, тем более агрессивно блок приложения выдает запросы масштабирования. Но задание коротких периодов снижения активности для операций увеличения масштаба и уменьшения масштаба приводит к возникновению риска колебаний, в ходе которых блок приложения многократно увеличивает, а затем уменьшает масштаб для роли. Если не указано иное, то блок приложения использует значение по умолчанию 20 минут для периода снижения активности.

Служебная информация позволяет также указывать в настройке, когда в течение часа следует разрешить выполнение операций масштабирования. В Windows Azure принято выставлять счет по календарным часам, поэтому может потребоваться использовать экземпляры роли максимально долго в течение часа. Для достижения этого можно указать, что операции увеличения масштаба могут происходить только в течение первых X минут часа, а операции уменьшения масштаба — в течение последних Y минут часа.

*Необходимо отвести достаточно времени для операций уменьшения масштаба, чтобы они могли завершаться до конца текущего часа. В противном случае счет будет выставляться и за следующий час.*

За исключением групп масштабирования, которые являются удобными, когда дело касается правил создания, разработчики приложения, как правило, определяют служебную информацию; они знают структуру приложения, а также то, что может и не может безопасно масштабироваться.

Использование блока автомасштабирования приложения не приводит автоматически к тому, что роли Windows Azure становятся масштабируемыми. Безусловно, Windows Azure предоставляет инфраструктуру, которая обеспечивает масштабирование приложений, но за достижение того, будут ли веб- и рабочие роли действовать правильно при наличии больше одного экземпляра роли, отвечает пользователь. Например, может оказаться, что отсутствует возможность параллельного выполнения некоторых алгоритмов.

*См. раздел «Алгоритм сокращения карты» книги *Разработка приложений для облака для ознакомления со сведениями о методах распараллеливания большого объема вычислений по нескольким экземплярам роли.**

Веб-роли могут быть масштабируемыми только при том условии, если они являются «применимыми для работы в веб-ферме». В частности, они являются таковыми, если используют состояние сеанса так, что поставщик состояния сеанса либо обеспечивает совместное использование, либо синхронизирует данные состояния сеанса в экземплярах



Нет смысла устанавливать длительность периодов снижения активности меньше десяти минут. Платформе Windows Azure для выполнения операции масштабирования применительно к роли часто может потребоваться десять минут, и в ходе выполнения этой операции она не принимает дополнительные запросы масштабирования для этой роли.



Необходимо убедиться в том, что служебная информация относится только к ролям, которые являются масштабируемыми.



Для автомасштабирования экземпляра требуются минимальные изменения в приложении, поскольку блок автомасштабирования приложения масштабирует приложение, добавляя или удаляя экземпляры роли. Для регулирования требуются более существенные изменения в приложении.

роли. В Windows Azure можно использовать поставщик состояния сеанса, который хранит состояние сеанса в общем кэше. Дополнительные сведения см. в разделе «Поставщик состояния сеанса» на сайте MSDN.

*Для сведения к минимуму риска раскрытия конфиденциальных данных следует шифровать содержимое хранилища служебной информации. Дополнительные сведения см. в разделе «Шифрование хранилища правил и хранилища служебной информации» на сайте MSDN.*

## Добавление функции регулирования в приложение

Блок автомасштабирования приложения позволяет использовать два различных механизма автомасштабирования в приложениях Windows Azure. Можно использовать правила автомасштабирования либо для изменения количества экземпляров роли, либо для изменения поведения приложения, как правило, путем *регулирования* приложения таким образом, чтобы в нем использовалось меньше ресурсов. К примерам функций регулирования относятся временное отключение некоторых несущественных функций приложения и переключение на менее ресурсоемкую версию пользовательского интерфейса.

Решение об использовании регулирования может быть принято в двух сценариях.

- Можно использовать регулирование вместо автомасштабирования экземпляра для некоторых или всех ролей в конкретном приложении. Этот вариант может быть выбран в случае, если роль не поддерживает эксплуатацию с несколькими экземплярами, или по той причине, что можно достичь лучших результатов масштабирования, изменяя поведение роли, а не добавляя или удаляя новые экземпляры.
- Необходимо добиваться того, чтобы приложение отвечало почти немедленно на резкое повышение потребности. Регулирование позволяет изменить поведение приложения сразу после того, как блок приложения выполнит действие правила реагирования, не будучи вынужденным ожидать, пока Windows Azure запустит новый экземпляр роли. В зависимости от размера и сложности приложения, регулирование может не оказывать воздействие быстрее, чем масштабирование экземпляра.

Чтобы добавить функцию регулирования к приложению Windows Azure, необходимо внести изменения в приложение, чтобы оно отвечало на запросы к этой функции. Дополнительные сведения о том, как приложение Windows Azure может обнаружить запрос для поведения регулирования, см. в разделе «*Реализация поведения регулирования*» на сайте MSDN.

Необходимо также создать ряд правил реагирования, в которых используется действие **changeSetting** для уведомления приложения, что оно должно включить или отключить некоторую функцию регулирования. Сведения о том, как определить правила автомасштабирования регулирования, см. в разделе «*Определение правил автомасштабирования регулирования*» на сайте MSDN.



Полный пример того, как приложение Tailspin Surveys использует поведение регулирования, см. в главе 5, «Создание более эластичного приложения Tailspin Surveys», в этом руководстве.

## Совместное использование автомасштабирования экземпляра и регулирования

В приложении Windows Azure можно использовать только автомасштабирование экземпляра или только регулирование; предусмотрена также возможность использовать эти функции вместе.

Если будет принято решение о совместном использовании функций, то нужно принимать во внимание, как происходит их взаимодействие. Создавая правила автомасштабирования, необходимо учитывать множество различий между указанными функциями.

- Правила автомасштабирования экземпляра могут потребовать до десяти минут для оказания воздействия, поскольку требуется время для запуска новых экземпляров роли в Windows Azure. Правила регулирования автомасштабирования могут повлиять на поведение приложения почти немедленно.
- Правила автомасштабирования экземпляра ограничиваются настраиваемыми периодами снижения активности, которые задают минимальное время перед тем, как блок приложения сможет снова масштабировать одну и ту же роль, но для правил регулирования автомасштабирования периоды снижения активности отсутствуют.
- Правила автомасштабирования экземпляра всегда лимитируются правилами ограничения. Правила регулирования автомасштабирования не лимитируются правилами ограничения.

Одно и то же правило реагирования может иметь действие, которое выполняет автомасштабирование экземпляра, и действие, которое выполняет регулирование.

Можно использовать ранги правил для управления приоритетом правил реагирования, которые выполняют автомасштабирование экземпляра, и правил реагирования, которые выполняют регулирование.

## Правила автомасштабирования

Действия автомасштабирования могут иметь место в ответ на правила, которые определяют, когда блок автомасштабирования приложения должен увеличивать или уменьшать масштаб ролей. По умолчанию эти правила хранятся в XML-документе в большом двоичном объекте Windows Azure, доступном для блока приложения.

Блок автомасштабирования приложения поддерживает два типа правил, которые определяют функцию автомасштабирования: *правила ограничения* и *правила реагирования*. Вообще говоря, за создание, контроль и поддержку этих правил отвечают администраторы приложения. Они могут выполнять указанные задачи, редактируя XML-документ, который содержит правила, или работая с пользовательским интерфейсом (UI), созданным разработчиками приложения.

При создании правил автомасштабирования можно создавать правила только для тех ролей, которые перечислены разработчиками в



В следующей главе описывается приложение Windows Azure с примером пользовательского веб-интерфейса для управления правилами.



Правила по умолчанию служат защитой для соглашений об уровне обслуживания!



Правила по умолчанию снижают затраты!

служебной информации. Необходимо планировать правила в три этапа.

1. Спроектировать применяемые по умолчанию правила ограничения (или «базовый план»).
2. Спроектировать все дополнительные правила ограничения.
3. Спроектировать правила реагирования.

Необходимо создать предусмотренное по умолчанию правило ограничения для каждой роли, которая перечислена в служебной информации. Правило, предусмотренное по умолчанию, не имеет расписания, поэтому всегда активно; оно имеет нулевой ранг, поэтому может быть переопределено любыми другими правилами ограничения; оно должно иметь минимальное и максимальное значения экземпляров роли, определяющие значения по умолчанию, необходимые при отсутствии каких-либо других активных правил ограничения. Эти правила, предусмотренные по умолчанию, должны гарантировать то, что всегда будет в наличии минимальное количество экземпляров роли, которые требуются для выполнения обязательств по соглашению об уровне обслуживания, и что не будет превышен бюджет в связи с запуском слишком большого количества экземпляров роли.

*Блок приложения вносит в журнал ошибку, если в правиле реагирования предпринимается попытка масштабировать цель, не имеющую правила ограничения. В этом сценарии блок приложения не выполняет никаких действий масштабирования применительно к цели.*

После создания предусмотренных по умолчанию правил можно определить любые дополнительные правила ограничения, которые помогут справиться с ожидаемыми периодами рабочей нагрузки приложения, которая находится выше или ниже нормы. Эти дополнительные правила ограничения имеют расписание, которые указывают, когда правило активно, задают ранг больше единицы, который гарантирует переопределение с их помощью предусмотренных по умолчанию правил ограничения, а также задают значения для максимальных и минимальных количеств экземпляров роли. Например, приложение может работать в условиях повышенной рабочей нагрузки с 9:00 до 10:00 каждое утро или в последнюю пятницу каждого месяца либо пониженной рабочей нагрузки с 13:00 до 17:00 каждый день или в течение всего августа.

Если требуется указать постоянное количество экземпляров для роли, можно использовать правило ограничения с максимальным



Можно воспользоваться представлением о времени изменения рабочей нагрузки приложения, основанным на ваших знаниях и опыте работы с приложением в вашей организации. Но благодаря мониторингу и анализу приложения можно получить гораздо более точные данные.

Необходимо контролировать и анализировать поведение приложения, чтобы понять, какие точки данных и агрегаты или какое их сочетание лучше всего действует в качестве меры производительности приложения. Может понадобиться несколько правил, поскольку различные аспекты приложения могут иметь разные характеристики производительности.

и минимальными значениями, заданными равными одному и тому же числу. В этом случае правила реагирования не будут оказывать никакого влияния на количество экземпляров роли.

Правила ограничения позволяют планировать ожидаемые изменения рабочей нагрузки.

Правила реагирования позволяют планировать непредвиденные изменения рабочей нагрузки приложения. Правило реагирования действует путем контролирования агрегатного значения, полученного на основании таких точек данных, как значения счетчика производительности, а затем выполнения операции масштабирования при достижении агрегатным значением пороговой величины. Сложность при использовании правил реагирования состоит в том, что нужно знать, какие агрегатные значения и точки данных или какое сочетание агрегатных значений и точек данных необходимо использовать в правилах реагирования. Например, если имеется правило реагирования, которое контролирует нагрузку ЦП, но приложение ограничено по вводу-выводу, то правило реагирования не будет вызывать действие масштабирования в нужное время. Еще одним примером является наличие правила реагирования, которое контролирует длину очереди Windows Azure. Если для функциональных возможностей приложения не имеет значения, будет ли очередь освобождена раньше или позже, то увеличение масштаба приложения для очистки очереди до того, как это потребуются, приведет к расходованию ресурсов впустую.

Если счетчики производительности или значения длины очереди Windows Azure не действуют должным образом как средство оценки производительности приложения, то разработчики приложения могут его инструментировать для создания настраиваемых бизнес-показателей, применяемых в правилах.

Если правила реагирования используют данные счетчика производительности из приложения Windows Azure, необходимо убедиться в том, что приложение передает данные счетчика производительности, которые правила берут из хранилища системы диагностики Windows Azure. Пример того, как это сделать, см. в разделе «Сбор данных счетчика производительности из приложения Tailspin Surveys» в главе 5, «Создание более эластичного приложения Tailspin Surveys», в этом руководстве.

Сведения об определении правил см. в подразделе «Правила и действия» выше в этом разделе.

*Для сведения к минимуму риска раскрытия конфиденциальных данных следует шифровать содержимое хранилища правил. Дополнительные сведения см. в разделе «Шифрование хранилища правил и хранилища служебной информации» на сайте MSDN.*

## **Реализация автомасштабирования на основе расписания без правил реагирования**

В некоторых сценариях может потребоваться использовать расписание для точного управления количеством экземпляров роли в разное время. Это может потребоваться сделать, чтобы эксплуатаци-

онные затраты были более прогнозируемыми, или если не ожидается непредвиденное резкое возрастание спроса на ваше приложение. Этой цели можно достичь, используя только правила ограничения, без правил реагирования. Кроме того, необходимо, чтобы максимальное количество числа экземпляров каждого правила ограничения было равно минимальному количеству экземпляров.

В следующем фрагменте кода приведен простой набор правил, реализующих автомасштабирование на основе расписания без использования правил реагирования. Правило ограничения по умолчанию задает два экземпляра роли, правило времени пиковой нагрузки задает четыре экземпляра роли.

```
XML
<rules
  xmlns=http://schemas.microsoft.com/practices/2011/entlib/
  autoscaling/rules
  enabled="true">
  <constraintRules>
    <rule name="Default" description="Always active"
      enabled="true" rank="1">
      <actions>
        <range min="2" max="2" target="RoleA"/>
      </actions>
    </rule>

    <rule name="Peak" description="Active at peak times"
      enabled="true" rank="100">
      <actions>
        <range min="4" max="4" target="RoleA"/>
      </actions>
      <timetable startTime="08:00:00" duration="02:00:00">
        <daily/>
      </timetable>
    </rule>
  </constraintRules>

  <reactiveRules/>

  <operands/>
</rules>
```



Блок автомасштабирования приложения может использовать средство ведения журнала Enterprise Library Logging Block, регистрацию данных системы диагностики или пользовательское средство ведения журнала.

## Мониторинг блока автомасштабирования приложения

Со временем модели использования приложения будут меняться. Общее количество пользователей будет увеличиваться или уменьшаться, пользователи начнут использовать приложение в разное время, в приложении могут появиться новые возможности, и пользователи будут использовать некоторые части приложения меньше, а другие больше. Поэтому применяемые правила ограничения

и правила реагирования могут больше не обеспечивать оптимальный баланс производительности и затрат.

Блок автомасштабирования приложения заносит в журнал подробные сведения о поведении правил, чтобы впоследствии можно было проанализировать, какие правила были инициированы и в какое время. Эта информация в сочетании с другими данными мониторинга производительности собираемых данных помогает проанализировать эффективность набора правил и определить, какие изменения следует внести, чтобы снова оптимизировать правила.

Частота анализа поведения правил зависит от степени динамичности среды, в которой работает приложение.

Сведения о регистрации данных, создаваемых в блоке приложения, см. в разделе «Ведение журнала данных блока автомасштабирования приложения» на сайте MSDN.

На рис. 7 показаны источники данных, которые можно использовать при анализе поведения блока автомасштабирования приложения и самого приложения.



Необходимо постоянно поддерживать оптимальность правил автомасштабирования для конкретных требований, эту задачу следует тщательно планировать.

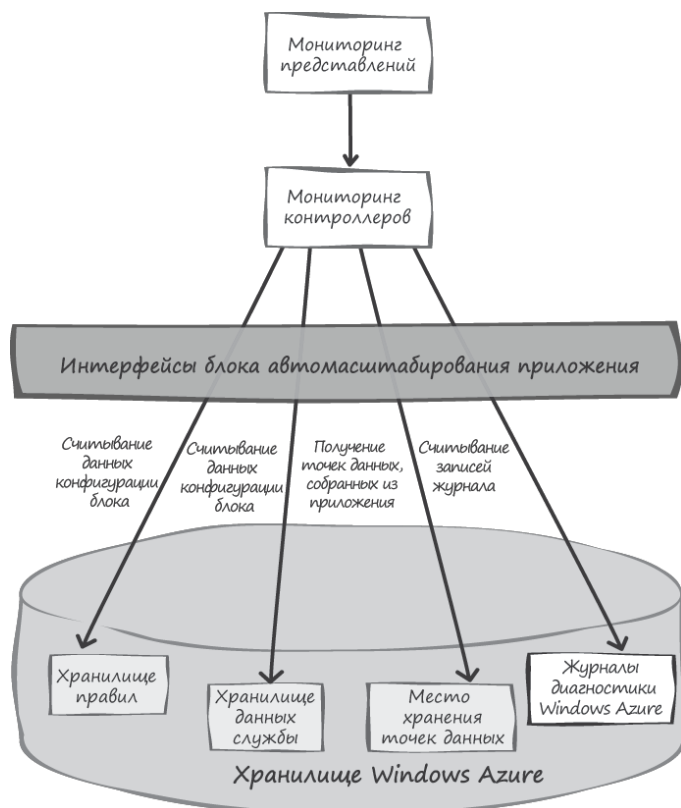


Рис. 7  
Мониторинг поведения автомасштабирования



Очень удобно использовать группы масштабирования. Они помогают свести к минимуму количество правил, которые нужно создать и которыми нужно управлять.

Блок приложения предоставляет интерфейсы, которые позволяют считывать из хранилищ данные конфигурации, правила автомасштабирования и служебную информацию. Можно также получить доступ к точкам данных, собираемых блоком приложения, таким как счетчики производительности и длина очередей, которые блок приложения использует при оценке правил реагирования. Блок приложения предоставляет также некоторые методы, позволяющие считывать и анализировать сообщения журнала, которые созданы и записаны в таблицу журнала системы диагностики Windows Azure с использованием инфраструктуры ведения журнала диагностики системы.

Дополнительные сведения о чтении из хранилища правил см. в разделе **Интерфейс IRulesStore** в документации по API.

Дополнительные сведения о чтении из хранилища служебной информации см. в разделе **Интерфейс IServiceInformationStore** в документации по API.

Дополнительные сведения о чтении из хранилища правил см. в разделе **Интерфейс IDataPointsStore** в документации по API.

Дополнительные сведения о чтении и анализе сообщений журнала блока автомасштабирования приложения см. в разделе «*Чтение сообщений журнала блока автомасштабирования приложения*».

Полный пример использования различных источников данных для визуализации действий блока автомасштабирования приложения см. в разделе «*Визуализация действий автомасштабирования*» в главе 5, «*Создание более эластичного приложения Tailspin Surveys*».

## Дополнительные сценарии использования

Этот раздел содержит рекомендации, когда следует использовать некоторые дополнительные функциональные возможности блока автомасштабирования приложения.

### Группы масштабирования

В приложении с многими веб-ролями и рабочими ролями может быть трудно создать большое количество правил, необходимых для определения поведения автомасштабирования в приложении, а также управлять такими правилами. В этом случае группы масштабирования предоставляют удобный способ определения правил, которые могут действовать одновременно на нескольких ролях. Перед созданием правил следует определить необходимые группы масштабирования.

Чтобы определить группу масштабирования, необходимо определить роли, которые будут составлять группу масштабирования, и назначить коэффициент каждому типу роли в группе масштабирования. Блок приложения использует эти коэффициенты для расчета количества экземпляров каждого члена группы масштабирования при выполнении действия масштабирования. В следующей таблице показан небольшой пример группы масштабирования; на практике группы масштабирования могут состоять из гораздо большего числа целевых объектов.

Целевой объект	Соотношение
Целевой объект А (рабочая роль А на узле службы А)	2
Целевой объект Б (рабочая роль А на узле службы Б)	1
Целевой объект В (веб-роль А на узле службы А)	3

*Группа масштабирования может включать целевые объекты, которые относятся к ролям в различных размещенных службах.*

Блок приложения не использует транзакции для выполнения операций над членами группы масштабирования, а группы масштабирования не гарантируют наличия соотношений между экземплярами роли. Например, правило ограничения может ограничить количество экземпляров, предлагаемых правилом реагирования для некоторых ролей в группе масштабирования, или оператор может вручную изменить количество экземпляров одной или нескольких ролей независимо от правил автомасштабирования.

Правило реагирования может использовать группу масштабирования в качестве своего целевого объекта. В следующей таблице показан эффект масштабирования группы масштабирования путем приращения на два с использованием коэффициентов, приведенных в предыдущей таблице.

Целевой объект	Исходное число экземпляров	Количество экземпляров после масштабирования
Целевой объект А	4	8
Целевой объект Б	2	4
Целевой объект В	6	12

Результат вычисляется следующим образом:

(исходное число экземпляров) + (шаг \* коэффициент)

В следующей таблице показан эффект правила реагирования, масштабирующего группу масштабирования на 50 %.

Целевой объект	Исходное число экземпляров	Количество экземпляров после масштабирования
Целевой объект А	4	8
Целевой объект Б	2	3
Целевой объект В	6	15

Результат вычисляется следующим образом:

(исходное число экземпляров)

+

(шаг \* коэффициент \* исходное число экземпляров)

Группы масштабирования можно также использовать при задании правил ограничения. Правило ограничения использует коэффициенты для определения максимального и минимального значений



Избегайте лишней сложности, к которой приводит включение одной роли в множество групп масштабирования. Это затрудняет понимание причин, по которым блок автомасштабирования приложения установил определенное значение для числа экземпляров.



При оценке или тестировании блока приложения можно использовать уведомления. Уведомления позволяют определить, какие операции будет выполнять блок приложения при текущем наборе правил.

счетчиков экземпляра роли. В примере правила ограничения задается максимальное число, равное пяти, и минимальное число, равное двум. В следующей таблице приведены значения максимального и минимального числа экземпляров отдельных ролей, которые составляют примерную группу масштабирования.

Целевой объект	Минимальное число экземпляров	Максимальное число экземпляров
Целевой объект А	4	10
Целевой объект Б	2	5
Целевой объект В	6	15

### *Использование в разное время разных коэффициентов*

Для применения в разное время разных коэффициентов можно использовать несколько групп масштабирования с одними и теми же членами. Например, можно определить две группы масштабирования, указанные в следующих таблицах.

#### Группа масштабирования А

Целевой объект	Соотношение
Целевой объект А (рабочая роль А на узле службы А)	2
Целевой объект Б (рабочая роль А на узле службы Б)	1
Целевой объект В (веб-роль А на узле службы А)	3

#### Группа масштабирования Б

Целевой объект	Соотношение
Целевой объект А (рабочая роль А на узле службы А)	3
Целевой объект Б (рабочая роль А на узле службы Б)	1
Целевой объект В (веб-роль А на узле службы А)	1

Затем можно определить два правила, как показано в следующей таблице.

Имя роли	Расписание	Ранг	Целевой объект
Правило по умолчанию	Всегда активно	1	Группа масштабирования А
Правило пакетной обработки	Каждое воскресенье с 02:00 до 04:00	20	Группа масштабирования Б

Оба правила направлены на одни и те же роли, но применяют различные коэффициенты. Правило ограничения пакетной обработки будет переопределять правило ограничения по умолчанию по воскресеньям с 02:00 до 04:00 утра и использовать коэффициенты, определенные для группы масштабирования Б.



## Использование уведомлений

Можно решить, что любые операции масштабирования, предлагаемые блоком автомасштабирования приложения, должны просматриваться перед тем, как блок приложения отправит их в Windows Azure. Это может оказаться полезным при тестировании блока приложения, когда нужно дважды проверять операции масштабирования до их выполнения или если нужно использовать набор знаний оператора о приложении, чтобы уточнить правила масштабирования и «настроить» действия масштабирования.

*Блок приложения может отправлять уведомления и одновременно выполнять действия масштабирования, чтобы операторы были уведомлены об операциях масштабирования, которые выполняет блок приложения.*

Блок приложения можно настроить для отправки сообщения электронной почты одному или нескольким операторам или администраторам. Сообщение электронной почты содержит полную информацию обо всех операциях масштабирования, которые блок приложения предлагал на основе текущего набора правил автомасштабирования.

Дополнительные сведения о настройке уведомлений см. в разделе «Использование уведомлений и масштабирования вручную» на сайте MSDN.

## Интеграция с жизненным циклом приложения

При развертывании приложения в Windows Azure развертывание можно выполнить в промежуточный или рабочий слот развертывания. Обычно развертывание сначала производится в промежуточной среде, где можно выполнить окончательные тесты, а затем содержимое промежуточного развертывания переносится в рабочую среду.

Если тестируется поведение автомасштабирования, необходимо иметь отдельные определения служебной информации и правила для каждого слота либо изменить определение служебной информации при переходе из промежуточной среды в рабочую.

В следующем фрагменте кода из файла определения служебной информации показано определение ролей и групп масштабирования для разных слотов развертывания.

### XML

```
<?xml version="1.0" encoding="utf-8"?>
<serviceModel ... >
  <subscriptions>
    <subscription name="Autoscaling Sample" ...>
      <services>
        <service dnsPrefix="stagingautoscalingservice" slot="Staging">
          <roles>
            <role alias="Staging.AutoScaling.WebApp"
              roleName="AutoScaling.WebApp" ... />
          </roles>
        </service>
        <service dnsPrefix="productionautoscalingservice" slot="Production">
          <roles>
            <role alias="Production.AutoScaling.WebApp"
```

```

        roleName="AutoScaling.WebApp" ... />
    </roles>
</service>
<service dnsPrefix="staging" slot="Staging">
    <roles>
        <role alias="Staging.Autoscaling.Scalegroup.Billing"
            roleName="Autoscaling.Scalegroup.Billing" ... />
        <role alias="Staging.Autoscaling.Scalegroup.BillProcessor"
            roleName="Autoscaling.Scalegroup.BillProcessor" ... />
        <role alias="Staging.Autoscaling.Scalegroup.InvoiceReporting"
            roleName="Autoscaling.Scalegroup.InvoiceReporting" ... />
    </roles>
</service>
<service dnsPrefix="production" slot="Production">
    <roles>
        <role alias="Production.Autoscaling.Scalegroup.Billing"
            roleName="Autoscaling.Scalegroup.Billing" ... />
        <role alias="Production.Autoscaling.Scalegroup.BillProcessor"
            roleName="Autoscaling.Scalegroup.BillProcessor" ... />
        <role alias="Production.Autoscaling.Scalegroup.InvoiceReporting"
            roleName="Autoscaling.Scalegroup.InvoiceReporting" ... />
    </roles>
</service>
</services>
<storageAccounts>
    ...
</storageAccounts>
</subscription>
</subscriptions>

<scaleGroups>
    <scaleGroup name="StagingScaleGroupA">
        <roles>
            <role roleAlias="Staging.Autoscaling.Scalegroup.Billing" ... />
            <role roleAlias="Staging.Autoscaling.Scalegroup.BillProcessor" ... />
            <role roleAlias="Staging.Autoscaling.Scalegroup.InvoiceReporting" ... />
        </roles>
    </scaleGroup>
    <scaleGroup name="ProductionScaleGroupA">
        <roles>
            <role roleAlias="Production.Autoscaling.Scalegroup.Billing" ... />
            <role roleAlias="Production.Autoscaling.Scalegroup.BillProcessor" ... />
            <role roleAlias="Production.Autoscaling.Scalegroup.InvoiceReporting" ... />
        </roles>
    </scaleGroup>
</scaleGroups>
</serviceModel>

```



Все псевдонимы ролей и имена групп масштабирования в служебной информации должны быть уникальными.

## Расширение блока автомасштабирования приложения

В библиотеке Enterprise Library почти все является расширяемым. Блок автомасштабирования приложения не является исключением. Если нужно расширить или изменить его функциональные возможности, он предоставляет пять ключевых точек расширения.

Дополнительные сведения см. в разделе «*Расширение и изменение блока автомасштабирования приложения*» на сайте MSDN.

Дополнительные сведения см. в разделе «*Практикум по расширяемости версии Microsoft Enterprise Library 5.0*».



Можно также загрузить исходный код и внести нужные изменения.

## Настраиваемые действия

При необходимости добавления нового действия в существующие действия масштабирования и регулирования можно создать настраиваемое действие. Создать настраиваемое действие можно за три шага.

1. Создайте код, реализующий действие.
2. Создайте код, который может выполнить десериализацию вашего настраиваемого действия из хранилища правил. Если используется встроенное хранилище правил, потребуется десериализация из XML.
3. Настройте блок приложения для использования настраиваемого действия.

Дополнительные сведения о настраиваемых действиях см. в разделе «*Создание настраиваемого действия*» на сайте MSDN.

Пример настраиваемого действия см. в главе 5, «*Создание более эластичного приложения Tailspin Surveys*».

## Настраиваемые операнды

В правиле реагирования операнд определяет агрегатное значение, вычисляемое на основе точек данных, собираемых блоком приложения. Если нужно добавить новый операнд к существующему счетчику производительности и операндам длины очереди, можно создать настраиваемый операнд. Настраиваемый операнд можно создать за три шага.

1. Создайте код, реализующий настраиваемого сборщика данных.
2. Создайте код, который может выполнить десериализацию настраиваемого операнда из хранилища правил. Если используется встроенное хранилище правил, потребуется десериализация из XML.
3. Настройте блок приложения для использования настраиваемого операнда.

Дополнительные сведения о настраиваемых операндах см. в разделе «*Создание настраиваемого операнда*» на сайте MSDN.

Пример настраиваемого операнда см. в главе 5, «*Создание более эластичного приложения Tailspin Surveys*».



С помощью настраиваемых операндов в определениях правил можно использовать бизнес-метрики.



Если приложению требуется много правил, полезно использовать SQL Server в качестве хранилища правил.

## Настраиваемые хранилища

Блок автомасштабирования приложения использует два хранилища, для правил и для служебной информации. Для каждого из этих хранилищ блок приложения включает две реализации: хранение данных в формате XML в большом двоичном объекте Windows Azure или хранение данных в формате XML в файле в локальной файловой системе. Первая реализация используется при размещении Application Block в роли Windows Azure, вторая — в случае размещения компонента его в локальном приложении.

Если нужно иметь возможность обработки правил или служебной информации в другом средстве, можно заменить эти реализации хранилищами, в которых данные хранятся в другом формате и в другом месте, например JSON хранится в локальном файле или в базе данных SQL Server.

Дополнительные сведения о создании настраиваемого хранилища правил см. в разделе «*Создание настраиваемого хранилища правил*» на сайте MSDN.

Дополнительные сведения о создании настраиваемого хранилища служебной информации см. в разделе «*Создание настраиваемого хранилища служебной информации*» на сайте MSDN.

## Настраиваемая регистрация

Блок автомасштабирования приложения может использовать средство ведения журнала в пространстве имен System.Diagnostics или в блоке приложения ведения журнала Enterprise Library для регистрации сведений о действиях автомасштабирования, которые он выполняет.

Если нужно использовать другую инфраструктуру регистрации, можно реализовать пользовательское средство ведения журнала для блока приложения. Это может оказаться полезным, если для хранения всех журналов в одном месте нужно выполнить интеграцию с существующей инфраструктурой регистрации.

Дополнительные сведения о создании пользовательского средства ведения журнала см. в разделе «*Создание пользовательского регистратора*» на сайте MSDN.

## Использование WASABiCmdlets

Командлеты Windows PowerShell WASABi® можно использовать для выполнения операций по отношению к блоку автомасштабирования приложения из скрипта Windows PowerShell. С помощью командлетов WASABi можно включать и отключать правила и оценку правил, изменять ранги правил, изменять поведение стабилизатора и многое другое.

В сочетании с командлетами Windows Azure PowerShell и пакетом System Center Operations Manager (SCOM) или другими средствами управления можно реализовать мощное пользовательское решение автомасштабирования.

Дополнительные сведения о командлетах WASABi см. в разделе «*Использование командлетов Windows PowerShell WASABi*» на сайте MSDN.

Дополнительные сведения о командлетах Windows Azure

PowerShell см. в разделе «Командлеты Windows Azure PowerShell» на сайте CodePlex.

Дополнительные сведения о SCOM см. в разделе «System Center Operations Manager» на сайте TechNet.

## Образец параметров конфигурации

Блок автомасштабирования приложения имеет большое количество параметров конфигурации, которые можно использовать для управления способом выполнения автомасштабирования приложения. В этом разделе описываются некоторые образцы конфигурации для иллюстрации того, как можно настроить блок приложения для удовлетворения конкретных требований. Эти иллюстрации являются только рекомендациями: для определения оптимальных параметров конфигурации приложения следует проанализировать собственные требования и поведение приложения Windows Azure.

Эти образцы конфигурации относятся к параметрам конфигурации блока автомасштабирования приложения и к правилам автомасштабирования.

Дополнительные сведения о настройке блока автомасштабирования приложения см. в разделе «Ввод данных конфигурации» на сайте MSDN.

Дополнительные сведения о написании правил автомасштабирования см. в разделах «Определение правил ограничения» и «Определение правил реагирования» на сайте MSDN.

Определение оптимального набора временных значений для решения обычно является итеративным процессом. Во время этих итераций необходимо учитывать временные значения, приведенные в следующей таблице. В таблице содержатся значения по умолчанию для ключевых параметров конфигурации.

Элемент конфигурации	Местоположение	Значение по умолчанию
Интервал отсчета числа экземпляров	Встроенный	Две минуты
Интервал отсчета счетчика производительности	Встроенный	Две минуты
Интервал отсчета длины очереди	Встроенный	Две минуты
Интервал оценки правила	Файл конфигурации	4 минуты
Интервал отслеживания	Файл конфигурации	5 минут
Интервал мониторинга хранилища правил задает частоту проверки блоком приложения изменений правил	Файл конфигурации	30 секунд
Интервал мониторинга хранилища служебной информации задает частоту проверки блоком приложения изменений служебной информации	Файл конфигурации	30 секунд

Периоды, задаваемые правилами ограничения	Хранилище правил	Нет
Операнд timespan, используемый для расчета агрегатных значений	Хранилище правил	Нет
Периоды снижения активности (увеличения и уменьшения масштаба), используемые стабилизатором	Хранилище служебной информации	20 минут
Периоды в начале и в конце часа, когда операции масштабирования не производятся	Хранилище служебной информации	Нет
Связанные с Windows Azure временные значения, такие как скорость передачи WAD и частоты выборки счетчика производительности.	Конфигурация или код приложения Windows Azure	Нет

На рис. 8 иллюстрируется связь между некоторыми временными параметрами конфигурации из предыдущей таблицы.

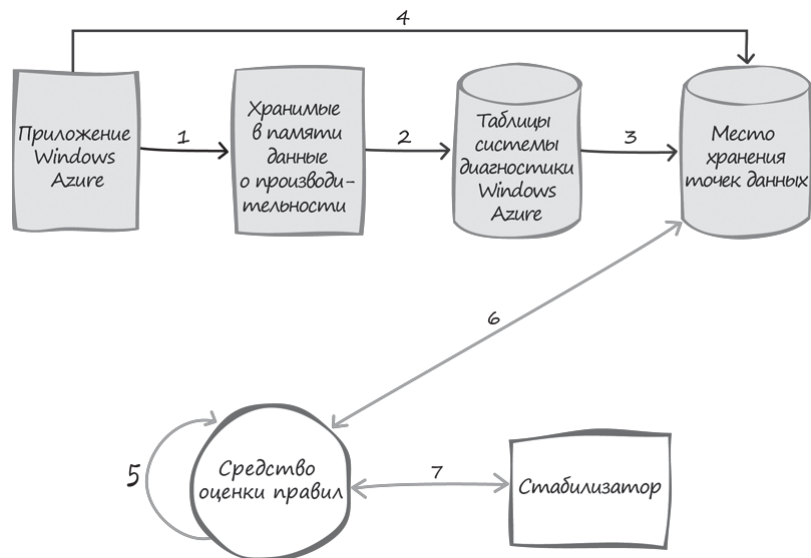


Рис. 8  
Временная связь

Следующий список поясняет, что происходит в каждой точке на диаграмме.

1. Приложение Windows Azure получает данные мониторинга производительности. Эти счетчики обычно настраиваются либо в методе **OnStart** ролей Windows Azure, либо через конфигурацию Windows Azure. Эти данные перехватываются в памяти.

2. Приложение Windows Azure сохраняет данные мониторинга производительности в таблицах системы диагностики Windows Azure (WAD). Периоды передачи обычно настраиваются либо в методе **OnStart** ролей Windows Azure, либо через конфигурацию Windows Azure.
3. Блок автомасштабирования приложения собирает данные счетчика производительности из таблиц системы диагностики Windows Azure и сохраняет их в хранилище точек данных. Это происходит каждые две минуты. Это значение встроено в блок приложения.
4. Блок автомасштабирования приложения собирает такие данные, как количество экземпляров, длина очереди и пользовательские показатели из приложения Windows Azure, и сохраняет их в хранилище точек данных. Это происходит каждые две минуты. Это значение встроено в блок приложения.
5. Запускается средство оценки правил, определяющее правила автомасштабирования, которые применяются в текущий момент времени. Частота, с которой работает средство оценки правил, указывается в файле конфигурации. Значение по умолчанию равно четырем минутам.
6. Средство оценки правил извлекает необходимые точки данных из хранилища точек данных. Объем данных для каждого правила определяется интервалом времени операнда, связанного с правилом. Например, если операнд задает в среднем более десяти минут, средство оценки правил извлекает из хранилища точек данных данные за последние десять минут.
7. Стабилизатор может предотвратить выполнение некоторых операций масштабирования. Например, стабилизатор может задавать периоды снижения активности после выполнения блоком приложения операции масштабирования или ограничивает операции масштабирования определенными периодами в течение часа.

В следующих разделах даны предложения по некоторым параметрам конфигурации для конкретных сценариев.

### **Средний период оценки правила**

Имеется веб-приложение, которое занято в некоторые моменты времени, и эти моменты трудно предсказать. Если вы начинаете замечать увеличение количества отклоненных веб-запросов за последние пять минут, требуется предпринять некоторые действия, чтобы обеспечить, что в течение следующих 20 минут будет достаточно доступных ресурсов для обработки возросшего числа запросов. Вы готовы согласиться с тем, что некоторые запросы могут продолжать отвергаться в течение следующих 20 минут.

Также имеется модель прогнозируемого использования, поэтому будут также использоваться правила ограничения, чтобы зарезервировать достаточное количество экземпляров в то время, когда известно, что будет большее количество запросов.



Для временных параметров, связанных с инфраструктурой Windows Azure, таких как период передачи WAD и частота опроса счетчиков производительности, необходимо определить значения для сценария приложения. Не надо полагать, что для всех сценариев хорошо подходит период передачи, равный одной минуте.



Помните, что Windows Azure требуется время для создания новых экземпляров роли, поэтому в данном случае следует ожидать, что в это время некоторые запросы будут отклонены.

В следующей таблице показаны некоторые образцовые значения конфигурации для данного сценария.

Элемент конфигурации	Значение по умолчанию
Интервал оценки правила	Пять минут
Интервал отслеживания	Пять минут
Интервал отсчета числа экземпляров	Две минуты (встроенный)
Интервал отсчета счетчика производительности	Две минуты (встроенный)
Интервал отсчета длины очереди	Две минуты (встроенный)
Период снижения активности (как при увеличении, так и при уменьшении масштаба)	20 минут
Операнд: приложение ASP.NET перезапускается	30 минут
Операнд: запросы ASP.NET ставятся в очередь	15 минут
Операнд: запросы ASP.NET отклоняются	5 минут

### *Продолжительный период оценки правила*

Обычно приложение имеет очень стабильный и константный уровень спроса, но иногда возникает умеренное увеличение использования. Поэтому принимается решение оценивать правила автомасштабирования каждые 30 минут и блокировать загрузку ЦП, превышающую среднюю. Также имеется правило уменьшения масштаба, которое выполняется, когда загрузка ЦП начинает падать до нормального уровня.

В следующей таблице показаны некоторые образцовые значения конфигурации для данного сценария.

Элемент конфигурации	Значение по умолчанию
Интервал оценки правила	30 минут
Интервал отслеживания	Пять минут
Интервал отсчета числа экземпляров	Две минуты (встроенный)
Интервал отсчета счетчика производительности	Две минуты (встроенный)
Интервал отсчета длины очереди	Две минуты (встроенный)
Период снижения активности (как при увеличении, так и при уменьшении масштаба)	20 минут
Операнд: процент загрузки ЦП	30 минут

### *Настройка стабилизатора*

Стабилизатор выполняет для блока автомасштабирования приложения две функции: предотвращает быстрые повторные изменения (эффект «цикличности») количества экземпляров роли, определяя периоды снижения активности, и помогает оптимизировать затраты путем ограничения операций увеличения масштаба в начале часа и операций уменьшения масштаба в конце часа.

В следующем фрагменте кода определения служебной информации показан пример конфигурации стабилизатора.



#### XML

```
<stabilizer scaleUpCooldown="00:20:00"
            scaleDownCooldown="00:30:00"
            scaleUpOnlyInFirstMinutesOfHour="15"
            scaleDownOnlyInLastMinutesOfHour="10"
            notificationsCooldown="00:25:00">
  <role roleAlias="BillingWorkerRole"
        scaleUpCooldown="00:18:00"
        scaleDownCooldown="00:18:00" />
</stabilizer>
```

*Можно настроить глобальные параметры стабилизатора и переопределять их для конкретных ролей.*

В этом примере параметр **scaleUpCooldown** предотвращает увеличение масштаба роли блоком приложения на 20 минут после каждого изменения числа экземпляров для этой роли. Аналогично параметр **scaleDownCooldown** предотвращает уменьшение масштаба роли блоком приложения на 30 минут после каждого изменения числа экземпляров для этой роли.

Параметр **scaleUpOnlyInFirstMinutesOfHour** гарантирует, что в первые 15 минут часа блок приложения выполняет только операции увеличения масштаба, а параметр **scaleDownOnlyInLastMinutesOfHour** гарантирует, что в последние 10 минут часа блок приложения выполняет только операции уменьшения масштаба. Эти два параметра позволяют оптимизировать использование экземпляров роли с учетом механизма выставления счетов Windows Azure.

*На момент написания этого документа за неполный час выполнения вычислительной операции счет выставляется за полный календарный час развертывания экземпляра. Например, при развертывании вычислительного экземпляра Small в 10:50 и удалении развертывания в 11:10 будет выставлен счет на два часа вычислений Small — один час вычислений за использование в период от 10:50 до 11:00 и еще один час вычислений за использование в период от 11:00 до 11:10. Таким образом, имеет смысл сохранять новые экземпляры действующими в оставшуюся часть часа, в течение которого они были запущены. Дополнительные сведения см. в разделе «Сведения о взимании платы за использование ресурсов Windows Azure».*

Два набора параметров взаимодействуют друг с другом. В этом примере операции увеличения масштаба разрешены только в течение первых 15 минут часа. Если блок приложения увеличивает масштаб роли в пять минут после начала часа, период снижения активности не допустит каких-либо дополнительных операций увеличения масштаба этой роли в течение следующих 20 минут. Это означает, что из-за наличия параметра масштабирования **UpOnlyInFirstMinutesOfHour** стабилизатор не допустит дополнительные операции увеличения масштаба для этой роли в течение этого часа.

## Использование средства «Планирование»

Этот лист поможет понять взаимодействие между разными временными значениями, которые регулируют весь процесс автомасштабирования. Этот лист можно загрузить с сайта сообщества Enterprise Library на сайте CodePlex.

Вводя значения, связанные со средой, можно наблюдать, как различные значения могут взаимодействовать друг с другом.

Например, на рис. 9 операнды 1 и 2 являются счетчиками производительности, а операнд 3 является пользовательским бизнес-показателем. Правила оцениваются через каждые 60 минут.

Соответствующие значения временных параметров		Операнды	
Временные параметры	Минуты	Имя операнда	Интервал агрегирования
Оценка правила	60	Операнд 1	25
Сбор данных	2	Операнд 2	25
Передача журнала	30	Операнд 3	30
Снижение активности	20	Операнд 4	0
		Операнд 5	0

Рис. 9  
Входные значения листа планирования

Лист планирования показывает результаты на рис. 10.

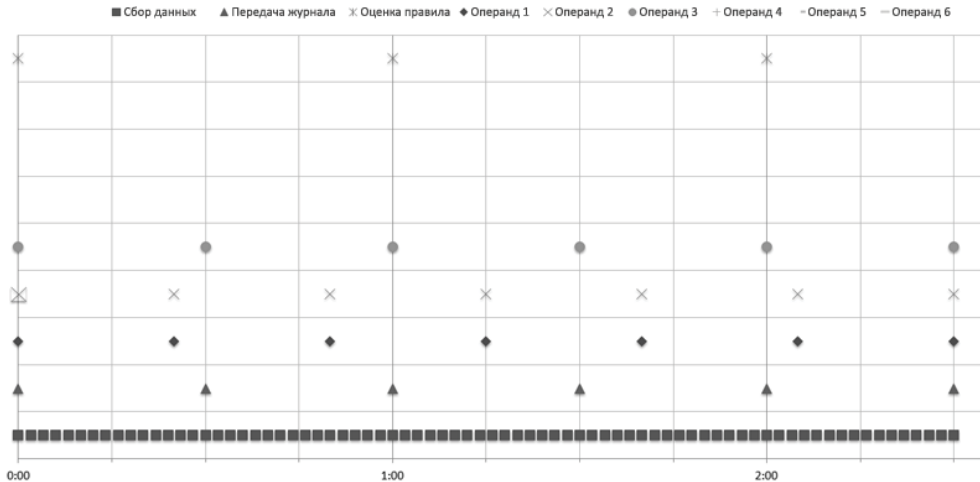


Рис. 10  
Результаты планирования

Этот пример демонстрирует, почему нужно быть осторожным с параметрами конфигурации блока автомасштабирования приложения. В первый час можно увидеть, как значения времени передачи данных показывают, что последние значения операндов 1 и 2 используются. Можно решить изменить интервал агрегирования для

операндов, изменить интервал передачи журнала или решить, что такое поведение применимо для данных, собираемых из приложения.

Реальное значение этого средства становится очевидным, если имеется большое количество операндов.

*Средство работает путем создания набора данных в скрытом листе. Если показать лист, можно будет наблюдать множество значений #N/D. Они заданы преднамеренно и предотвращают появление в диаграмме неровных строк.*

## Как работает блок автомасштабирования приложения

В этом разделе обобщается описание работы блока автомасштабирования приложения. Если вам не интересен внутренний механизм этого блока, то можете пропустить этот раздел. На рис. 11 показано, как ключевые компоненты блока автомасштабирования приложения связаны между собой, с Windows Azure и с внешними компонентами.

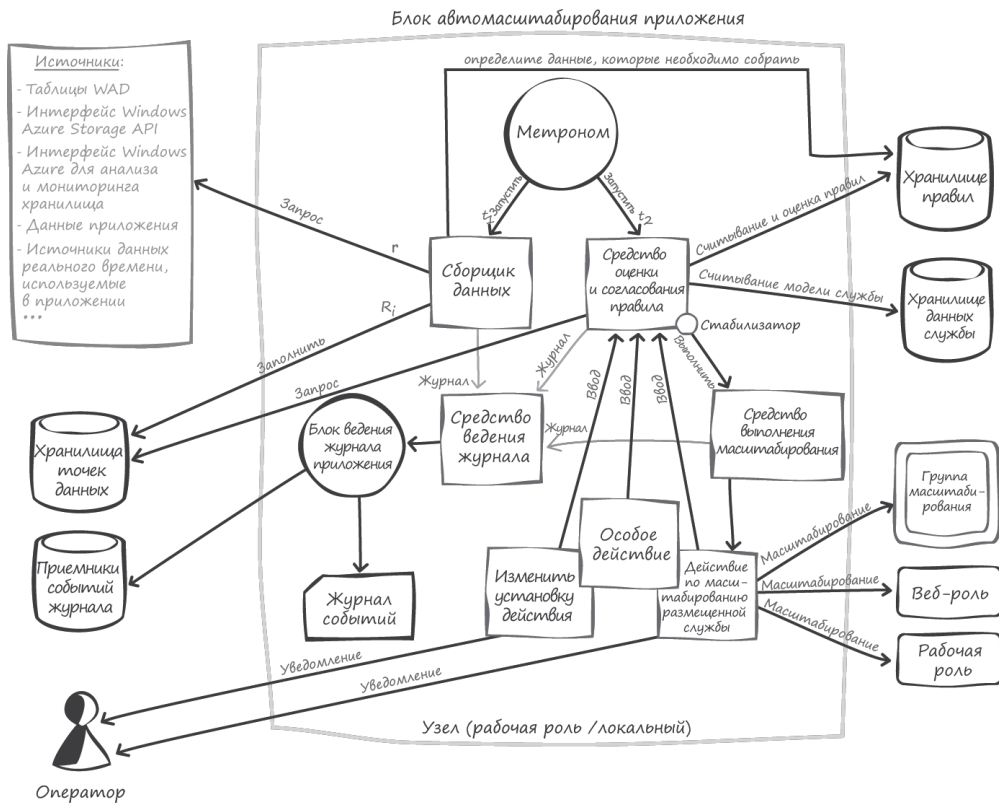


Рис. 11  
Общие сведения о блоке автомасштабирования приложения

## Метроном

Работа, которую выполняет блок автомасштабирования приложения, начинается с метронома. Метроном создает тактовые импульсы, которые позволяют ему запускать другие действия по регулярному расписанию. В блоке автомасштабирования приложения он запускает действие сборщика данных каждые две минуты, действие средства оценки правил каждые  $t_1$  секунд и действие средства отслеживания каждые  $t_2$  секунд. Значение по умолчанию для  $t_1$  — четыре минуты, для  $t_2$  — пять минут, и эти значения можно переопределить в параметрах конфигурации блока приложения.

## Сборщики данных

Перед тем как блок автомасштабирования приложения сможет выполнять правила реагирования, он должен собрать значения точек данных, на основе которых он вычисляет агрегатные значения, используемые в правилах реагирования. Точка данных является значением показателя, таким как загрузка ЦП в определенный момент времени. Действия сборщика данных получают точки данных из среды Windows Azure. В следующей таблице перечислены возможные источники точек данных.

Мониторинг источника данных	Описание
Таблицы системы диагностики Windows Azure	Это таблицы, которые использует система диагностики Windows Azure при сохранении данных диагностики, собираемых во время выполнения из среды Windows Azure. Блок приложения получает данные счетчика производительности из <b>таблицы счетчиков WADPerformance</b> .
Интерфейс Windows Azure API	Сборщик данных может запрашивать хранилище Windows Azure приложения, включая очереди, большие двоичные объекты и таблицы для пользовательских точек данных. Блок приложения получает значения длины очередей Windows Azure с использованием данного API.
Интерфейс API системы аналитики хранилища Windows Azure	Сборщик данных может использовать этот API для получения точек данных, относящихся к конкретной учетной записи хранилища, таких как агрегатные данные и данные производительности.
Данные приложения	Сборщик данных может получать пользовательские точки данных из конкретного приложения. Например, количество заказов, хранящихся в таблице Windows Azure.

Действия **сборщика данных** записывают значения собранных точек данных в **хранилище точек данных**.

Хотя действия не показаны на диаграмме, блок приложения создает действия **сборщика данных** после чтения **хранилища правил** для определения точек данных, которые будут использовать правила реагирования. В результате блок приложения определяет, какие точки данных он должен собирать на основе определений правил в хранилище правил.

Хранилище служебной информации содержит сведения о приложении Windows Azure, а также сведения о том, что действия **сборщика данных** должны иметь возможность доступа к ролям и хранилищу.



Блок приложения не поддерживает размещение хранилища точек данных в эмуляторе локального хранилища Windows Azure. Блок приложения использует вызов интерфейса API Windows Azure, который не поддерживается эмулятором локального хранилища.

## Хранилище служебной информации

Хранилище служебной информации хранит служебную информацию для приложения Windows Azure. Эта служебная информация включает все сведения о приложении Windows Azure, в котором блок приложения должен иметь возможность собирать точки данных и выполнять операции масштабирования.

## Хранилище точек данных

Действия сборщика данных приводят к заполнению хранилища точек данных точками данных. Действие оценщика правил приводит к запросу в хранилище точек данных таких точек данных, которые необходимы для оценки правил реагирования.

По умолчанию для **хранилища точек данных** блок автомасштабирования приложения использует хранилище таблиц Windows Azure.

## Средство оценки правил

Наряду с запуском действия сборщика данных метроном также периодически запускает действие средства оценки правил. Когда запускается задача средства оценки правил, она запрашивает хранилище правил для определения того, какие правила масштабирования следует применять в настоящее время. Хранилище правил кэширует правила в памяти, но через настраиваемый период времени проверяет, изменились ли правила, и, если да, повторно загружает правила из резервного хранилища. Оно затем запрашивает точки данных в хранилище точек данных, чтобы вычислить агрегатные значения, которые ему необходимы для оценки правил реагирования. Оно согласовывает конфликты между правилами перед выполнением действий, запускаемых по правилам.

Дополнительные сведения о том, как блок приложения согласовывает конфликтующие правила и действия, см. в разделе «Основные сведения о рангах правил и согласовании».

## Хранилище правил

Хранилище правил содержит список всех правил автомасштабирования, определенных для приложения Windows Azure. Напомним, что эти правила могут быть правилами ограничения или правилами реагирования.

По умолчанию для **хранилища правил** блок автомасштабирования приложения использует хранилище таблиц Windows Azure.

Правило может инициировать одно или несколько действий. В следующей таблице описываются три вида действий, которые поддерживает блок автомасштабирования приложения.

Тип действия	Описание
Действие масштабирования	Выполняет автомасштабирование экземпляра или отправляет уведомление
Действие регулирования	Выполняет регулирование приложения.
Настраиваемое действие	Выполняет настраиваемое, пользовательское действие.

Блок автомасштабирования приложения также может предлагать оператору действия масштабирования через механизм уведомлений.

### Средство ведения журнала

Компонент Средство ведения журнала может использовать блок приложения регистрации Enterprise Library для сохранения данных системы диагностики из блока автомасштабирования приложения. Компонент Средство ведения журнала можно также настроить для использования других компонентов регистрации, таких как пространство имен System.Diagnostics.

Дополнительные сведения об информации, вносимой в журнал и создаваемой блоком приложения, см. в разделе «Ведение журнала данных блоком автомасштабирования приложения» на сайте MSDN.

Дополнительные сведения о настройке средства ведения журнала см. в разделе «Ввод данных конфигурации» на сайте MSDN.

### Средство масштабирования

Средство масштабирования выполняет обмен данными с Windows Azure для добавления или удаления экземпляров роли с учетом действий правил. Он также включает компонент стабилизатора для предотвращения повторяющегося добавления и удаления экземпляров роли средством масштабирования.

Выполнение операций средства масштабирования может занять некоторое время. **Средство масштабирования** инициирует операции масштабирования и добавляет сообщение в очередь отслеживания для записи того факта, что блок приложения запросил операцию масштабирования.

Вместо непосредственного выполнения действий масштабирования **средство масштабирования** может отправлять уведомления о сообщениях электронной почты оператору с подробным описанием предлагаемых действий масштабирования.

### Объект отслеживания

Действие объекта отслеживания приводит к отслеживанию всех операций масштабирования, инициированных средством масштабирования. По умолчанию метроном запускает действие объекта отслеживания каждую минуту. Затем действие объекта отслеживания

проверяет, какие операции масштабирования в очереди отслеживания завершены успешно или с ошибкой. Оно записывает данные о завершенных операциях масштабирования, включая сведения об ошибке, если операция завершилась с ошибкой, а затем удаляет запись из очереди.

## Дополнительные сведения

Дополнительные сведения о конструктивных требованиях см. в разделе «Построение масштабируемого, многопользовательского приложения для Windows Azure» на сайте MSDN:  
<http://msdn.microsoft.com/en-us/library/ff966483.aspx>

Дополнительные сведения о времени вычисления в Windows Azure см. в разделе «Сведения о плате за использование для счетов Windows Azure»:  
<http://go.microsoft.com/fwlink/?LinkID=234626>

Дополнительные сведения о подписках Windows Azure см. на странице технической поддержки Windows Azure:  
<http://www.microsoft.com/windowsazure/support/>

Дополнительные сведения о том, как блок автомасштабирования приложения согласовывает конфликтующие правила, см. в разделе «Основные сведения о рангах правил и согласовании» на сайте MSDN:  
[http://msdn.microsoft.com/en-us/library/hh680923\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680923(v=PandP.50).aspx)

Более подробное обсуждение методов оценки своих эксплуатационных расходов для Windows Azure см. в главе «Сколько будет это стоить?» в книге «Перемещение приложений в облако»:  
<http://msdn.microsoft.com/en-us/library/ff803375.aspx>

Обсуждение состояния сеанса в приложениях Windows Azure см. в разделе «Сохранение состояния сеанса» в книге «Перемещение приложений в облако»:  
<http://msdn.microsoft.com/en-us/library/ff803373.aspx#sec11>

Обсуждение некоторых вопросов разработки, связанных с масштабируемыми рабочими ролями, см. в разделе «Масштабирование приложений с использованием рабочих ролей» в книге «Разработка приложений для облака»:  
<http://msdn.microsoft.com/en-us/library/hh534484.aspx#sec14>

Сведения о том, как использовать NuGet для подготовки проекта Visual Studio к работе с блоком автомасштабирования приложения, см. в разделе «Добавление блока автомасштабирования приложения на узел» на сайте MSDN:  
[http://msdn.microsoft.com/en-us/library/hh680920\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680920(v=PandP.50).aspx)

Сведения о том, как разместить блок автомасштабирования приложения в Windows Azure, см. в разделе «Размещение блока автомасштабирования приложения в рабочей роли» на сайте MSDN:  
[http://msdn.microsoft.com/en-us/library/hh680914\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680914(v=PandP.50).aspx)

Сведения о том, как можно ссылаться на сборки Enterprise Library, как Enterprise Library обрабатывает зависимости и как работать с объектами Enterprise Library, см. в разделе «Использование Enterprise Library в приложениях» в основной документации по Enterprise Library на сайте MSDN: [http://msdn.microsoft.com/en-us/library/ff664560\(PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/ff664560(PandP.50).aspx)

Сведения о том, как разместить блок автомасштабирования приложения в локальном приложении, см. в разделе «Размещение блока автомасштабирования приложения в локальном приложении» на сайте MSDN:

[http://msdn.microsoft.com/en-us/library/hh680882\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680882(v=PandP.50).aspx)

Дополнительные сведения об изменениях в коде, которые необходимо внести в приложение Windows Azure для сохранения данных счетчика производительности, см. в разделе «Сбор данных счетчика производительности» на сайте MSDN:

[http://msdn.microsoft.com/en-us/library/hh680886\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680886(v=PandP.50).aspx)

Для настройки счетчиков производительности можно также использовать файл конфигурации системы диагностики Windows Azure (diagnostics.wadcfg). Дополнительные сведения см. в разделе «Как использовать файл конфигурации системы диагностики Windows Azure» на сайте MSDN:

<http://go.microsoft.com/fwlink/?LinkID=234617>

Правила блока автомасштабирования приложения могут действовать только на целевых объектах (роли и группы масштабирования), которые определены в служебной информации блока. Дополнительные сведения см. в разделе «Хранение служебных данных» на сайте MSDN:

[http://msdn.microsoft.com/en-us/library/hh680878\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680878(v=PandP.50).aspx)

Сведения о методе распараллеливания больших вычислений на нескольких экземплярах роли см. в разделе «Алгоритм Map Reduce» в книге «Разработка приложений для облака»:

<http://msdn.microsoft.com/en-us/library/ff966483.aspx#sec18>

В Windows Azure можно использовать поставщик состояния сеанса, который хранит состояние сеанса в общем кэше. Дополнительные сведения см. на странице «Поставщик состояния сеанса» на сайте MSDN: <http://msdn.microsoft.com/en-us/library/gg185668.aspx>

Для сведения к минимуму риска раскрытия конфиденциальных данных следует шифровать содержимое хранилища служебной информации.

Дополнительные сведения см. в разделе «Шифрование хранилища правил и хранилища служебной информации» на сайте MSDN:

[http://msdn.microsoft.com/en-us/library/hh680910\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680910(v=PandP.50).aspx)

Дополнительные сведения о том, как приложение Windows Azure может обнаружить запрос для поведения регулирования, см.

в разделе «Реализация поведения регулирования» на сайте MSDN: [http://msdn.microsoft.com/en-us/library/hh680896\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680896(v=PandP.50).aspx)



Сведения о том, как определить правила автомасштабирования регулирования, см. в разделе «Определение правил автомасштабирования регулирования» на сайте MSDN:

[http://msdn.microsoft.com/en-us/library/hh680908\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680908(v=PandP.50).aspx)

Полный пример того, как приложение Tailspin Surveys использует поведение регулирования, см. в главе 5, «Создание более эластичного приложения Tailspin Surveys», в этом руководстве.

Если правила реагирования используют данные счетчика производительности из приложения Windows Azure, необходимо убедиться в том, что приложение передает данные счетчика производительности, которые правила берут из хранилища системы диагностики Windows Azure. Пример того, как это сделать, см. в разделе «Сбор данных счетчика производительности из приложения Tailspin Surveys» в главе 5, «Создание более эластичного приложения Tailspin Surveys», в этом руководстве.

Сведения о регистрации данных, создаваемых блоком автомасштабирования приложения, см. в разделе «Ведение журнала данных блока автомасштабирования приложения» на сайте MSDN:

[http://msdn.microsoft.com/en-us/library/hh680883\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680883(v=PandP.50).aspx)

Дополнительные сведения о чтении и анализе сообщений журнала блока автомасштабирования приложения см. в разделе «Чтение сообщений журнала блока автомасштабирования приложения»:

[http://msdn.microsoft.com/en-us/library/hh680909\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680909(v=PandP.50).aspx)

Дополнительные сведения о настройке блока автомасштабирования приложения и настройке средства ведения журнала см. в разделе «Ввод данных конфигурации» на сайте MSDN:

[http://msdn.microsoft.com/en-us/library/hh680915\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680915(v=PandP.50).aspx)

Дополнительные сведения о чтении из хранилища правил см. в разделе «Интерфейс IRulesStore» в документации по API на сайте MSDN:

<http://go.microsoft.com/fwlink/?LinkID=234680>

Дополнительные сведения о чтении из хранилища служебной информации см. в разделе «Интерфейс IServiceInformationStore» в документации по API на сайте MSDN:

<http://go.microsoft.com/fwlink/?LinkID=234681>

Дополнительные сведения о чтении из хранилища правил см. в разделе «Интерфейс IDataPointsStore» в документации по API на сайте MSDN:

<http://go.microsoft.com/fwlink/?LinkID=234682>

Полный пример использования различных источников данных для визуализации действий блока автомасштабирования приложения см. в разделе «Визуализация действий автомасштабирования» в главе 5, «Создание более эластичного приложения Tailspin Surveys».

Дополнительные сведения о настройке уведомлений см. в разделе «Использование уведомлений и масштабирования вручную» на сайте MSDN:

[http://msdn.microsoft.com/en-us/library/hh680885\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680885(v=PandP.50).aspx)

Дополнительные сведения о расширении и изменении блока автомасштабирования приложения см. в разделе «Расширение и изменение блока автомасштабирования приложения» на сайте MSDN:  
[http://msdn.microsoft.com/en-us/library/hh680889\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680889(v=PandP.50).aspx)

- Дополнительные сведения о настраиваемых действиях см. в разделе «Создание настраиваемого действия» на сайте MSDN:  
[http://msdn.microsoft.com/en-us/library/hh680921\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680921(v=PandP.50).aspx)
- Пример настраиваемого действия и настраиваемого операнда см. в главе 5, «Создание более эластичного приложения Tailspin Surveys».
- Дополнительные сведения о настраиваемых операндах см. в разделе «Создание настраиваемого операнда» на сайте MSDN:  
[http://msdn.microsoft.com/en-us/library/hh680912\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680912(v=PandP.50).aspx)
- Дополнительные сведения о создании настраиваемого хранилища правил см. в разделе «Создание настраиваемого хранилища правил» на сайте MSDN:  
[http://msdn.microsoft.com/en-us/library/hh680933\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680933(v=PandP.50).aspx)
- Дополнительные сведения о создании настраиваемого хранилища служебной информации см. в разделе «Создание настраиваемого хранилища служебной информации» на сайте MSDN:  
[http://msdn.microsoft.com/en-us/library/hh680884\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680884(v=PandP.50).aspx)
- Дополнительные сведения о создании пользовательского средства ведения журнала см. в разделе «Создание пользовательского регистратора» на сайте MSDN: [http://msdn.microsoft.com/en-us/library/hh680926\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680926(v=PandP.50).aspx)

Дополнительные сведения о расширении Enterprise Library см. в разделе «Практикум по расширяемости версии Microsoft Enterprise Library 5.0»:

<http://go.microsoft.com/fwlink/?LinkId=209184>

Дополнительные сведения о командлетах WASABi см. в разделе «Использование командлетов Windows PowerShell WASABi» на сайте MSDN:

[http://msdn.microsoft.com/en-us/library/hh680938\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680938(v=PandP.50).aspx)

Дополнительные сведения о командлетах Windows Azure PowerShell см. в разделе «Командлеты Windows Azure PowerShell» на сайте CodePlex:

<http://wappowershell.codeplex.com/>

Дополнительные сведения о SCOM см. в разделе «System Center Operations Manager» на сайте TechNet:

<http://technet.microsoft.com/en-us/systemcenter/om/default.aspx>

Дополнительные сведения о написании правил автомасштабирования см. в разделах «Определение правил ограничения» и «Определение правил реагирования» на сайте MSDN:

[http://msdn.microsoft.com/en-us/library/hh680917\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680917(v=PandP.50).aspx)

[http://msdn.microsoft.com/en-us/library/hh680897\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680897(v=PandP.50).aspx)

Дополнительные сведения о выставлении счетов в Windows Azure см. в разделе «Сведения о плате за использование для счетов Windows Azure»:

<http://go.microsoft.com/fwlink/?LinkID=234626>

Лист планировщика автомасштабирования поможет понять взаимодействие различных временных значений, которые регулируют весь процесс автомасштабирования. Этот лист можно загрузить с сайта сообщества Enterprise Library на сайте CodePlex:

<http://go.microsoft.com/fwlink/?LinkID=234704>

Чтобы получить простой доступ к ресурсам в сети, используйте электронную версию библиографии на MSDN:

[http://msdn.microsoft.com/en-us/library/hh749032\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh749032(v=PandP.50).aspx)

## 5 Создание более эластичного приложения Tailspin Surveys

В этой главе описываются изменения, которые Tailspin вносит при добавлении блока автомасштабирования приложения в приложение Surveys. Эти изменения позволили автоматизировать процессы добавления и удаления экземпляров роли, а также дали возможность управлять потребностью приложения в ресурсах в ответ на изменение спроса на приложение. В главе также показано, как компания Tailspin настраивает разные наборы правил автомасштабирования для различных элементов приложения Surveys для обеспечения их разных потребностей в автомасштабировании, и описано, как Tailspin планирует контролировать и уточнять свои правила автомасштабирования.

### Предпосылки

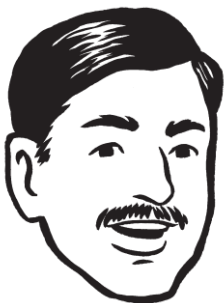
Количество клиентов, использующих приложение Tailspin Surveys, продолжает расти, при этом клиенты создают и публикуют обследованию по всему миру. Также растет число обследований с большим количеством респондентов. В компании Tailspin заметили, что существует все большее количество всплесков спроса, связанных с этими крупными обследованиями, при этом эти всплески спроса происходят вскоре после того, как заказчик публикует обследование. Tailspin не может предсказать, когда будут происходить эти всплески спроса или в каком географическом месте они будут происходить. В Tailspin также заметили, что имеются общие всплески спроса в определенное время в определенных географических местоположениях. Для некоторых из этих всплесков в Tailspin понимают причину, например предстоящий сезон отпусков; для других еще непонятно, что вызывает всплеск спроса.

Операторы Tailspin всегда внимательно отслеживают трафик и уровни использования их систем. При необходимости они вручную регулируют число экземпляров веб- и рабочих ролей, чтобы приспособиться к изменениям в спросе. Однако трудно правильно определить количество экземпляров роли, которые должны быть активны в определенные моменты времени. Чтобы обеспечить высокую производительность и доступность для приложения Surveys, они обычно запускают больше серверов, чем может требоваться. Однако, когда



В настоящее время все операторы Tailspin располагаются в США, поэтому вне обычного рабочего времени на дежурстве находится меньше операторов. В результате они иногда не сразу реагируют на всплески нагрузки в других географических районах.

Мы должны сократить эксплуатационные затраты, чтобы быть более конкурентоспособными. Мы можем этого добиться с помощью автоматического масштабирования приложения Surveys и по-прежнему выполнять требования соглашения об уровне обслуживания.



возникает большой всплеск трафика, операторам компании Tailspin может потребоваться некоторое время для реагирования и повышения мощности сервера, особенно для центров обработки данных, расположенных не в США. Кроме того, операторы Tailspin иногда с запозданием останавливали серверы по окончании всплеска активности.

Результатом этого ручного процесса является то, что часто имеется слишком много экземпляров роли как во время нормальной деятельности, так и во время всплесков активности, что увеличивает эксплуатационные затраты приложения Surveys. Кроме того, когда возникает неожиданный всплеск трафика, добавление новых экземпляров роли может занять много времени, что приводит к низкой производительности и к медленной работе пользователей.

## Цели и требования

Компания Tailspin хочет сделать свое приложение более эластичным, чтобы количество серверов могло расти и уменьшаться автоматически при изменении спроса. Это позволит уменьшить затраты на эксплуатацию приложения Surveys на технологической платформе Windows Azure™ и уменьшить количество нерегламентированных задач, выполняемых вручную операторами Tailspin.

Tailspin хочет установить явные границы, определяющие количество активных экземпляров роли, чтобы сохранять эксплуатационные затраты в прогнозируемом диапазоне и обеспечивать применение соглашений об уровне обслуживания Windows Azure к приложению Surveys.

В прошлом в компании Tailspin возникали некоторые внезапные всплески спроса, например, когда клиенты предлагали награду для первых пользователей, которые заполнили анкету. В компании Tailspin озабочены тем, что новые экземпляры роли не могут быть запущены достаточно быстро для удовлетворения всплесков активности такого типа. В этом сценарии в Tailspin хотели бы иметь возможность немедленно начать отключение некоторых второстепенных функций приложения, чтобы поддерживать требуемое время отклика пользовательского интерфейса до запуска новых экземпляров роли. В Tailspin также хотели бы, чтобы операторы компании уведомлялись SMS-сообщением, когда происходят определенные операции масштабирования.

На основе предварительно собранных данных в Tailspin уже могут предсказывать, когда будут возникать некоторые всплески спроса. В Tailspin хотят иметь возможность заблаговременно составлять расписания добавления и удаления экземпляров роли, чтобы не было запаздывания при возникновении такого всплеска спроса.

В приложении Tailspin Surveys количество экземпляров ролей является ключевым ресурсом, который можно изменять в соответствии с меняющейся рабочей нагрузкой. В других приложениях ресурсы могут включать такие элементы, как количество очередей или размер базы данных.

В целом в Tailspin хотят сократить эксплуатационные затраты, обеспечивая в то же время максимальную производительность для конечных пользователей. Это означает использование минимально необходимого количества экземпляров роли для выполнения задачи на требуемом уровне производительности. Если значения некоторых счетчиков производительности, таких как загрузка ЦП, превышают стандартный порог, система должна добавлять дополнительные экземпляры роли до тех пор, пока значения не упадут до более приемлемого уровня. Когда значения этих счетчиков производительности достаточно снизятся, экземпляры роли следует снова удалять.

Компания Tailspin должна иметь возможность управлять параметрами автомасштабирования путем определения правил. Правило определяет, какие экземпляры будут масштабированы, какие показатели должны контролироваться и какие пороговые значения нужно использовать для увеличения и уменьшения масштаба. Например, если загрузка ЦП в среднем достигает 80 % для всех выполняемых экземпляров, нужно добавить новый экземпляр роли. Должна также иметься возможность использовать правила для установки минимального и максимального количества экземпляров роли в определенные периоды времени. Эти правила следует настраивать через пользовательский интерфейс.

Трудно определить нужное число экземпляров роли для конкретной задачи, даже с помощью решения автомасштабирования. Компания Tailspin хочет иметь доступ к подробным данным регистрации, в которых записаны действия автомасштабирования, имевшие место. Например, в Tailspin хотят знать, когда экземпляр роли был добавлен или удален и что инициировало это действие. В компании планируется анализировать эти данные и использовать результаты для уточнения поведения автомасштабирования приложения Surveys.

Tailspin хочет получить общие сведения об использовании ресурсов с течением времени. Например, нужно знать число активных экземпляров роли в определенный момент времени, загрузку ЦП и количество одновременно работающих активных пользователей в это время. В Tailspin хотят иметь возможность анализировать, когда возникают всплески общего спроса в определенных географических точках, чтобы можно было планировать эти события в будущем. Компания Tailspin планирует также использовать эту информацию для уточнения своей ценовой стратегии, основанной на более детальном анализе прогнозируемых уровней спроса.

Компания развернула приложение Surveys в нескольких центрах обработки данных Windows Azure, чтобы приложение Surveys было доступно в центре обработки данных, расположенном недалеко от большинства пользователей, выполняющих обследования. В Tailspin хотели бы использовать одно приложение управления, средствами которого компания могла бы создавать и изменять различные правила автомасштабирования, используемые в разных географических точках, и таким образом контролировать поведение автомасштабирования во всех центрах данных. В то же время в Tailspin хотят свести к минимуму затраты на это решение распределенного автомасштабирования.



Мы провели тесты на производительность для выявления наиболее ресурсоемких областей нашего приложения. Выяснилось, что некоторые из этих частей можно было отключить или отложить их выполнение, чтобы сократить нагрузку на приложение.

## Общие сведения о решении автомасштабирования

В этом разделе описывается, как приложение Tailspin Surveys стало более эластичным в результате использования блока автоматического масштабирования приложения.

## Использование блока автоматического масштабирования приложения в Tailspin Surveys

В этом разделе описывается, как приложение Surveys использует функции блока автоматического масштабирования приложения.

Дополнительные сведения об автоматическом масштабировании и о том, как работает блок автоматического масштабирования приложения, см. в главе 4, «Автоматическое масштабирование и Windows Azure», в этом руководстве.

### Функции блока автоматического масштабирования приложения

Блок автоматического масштабирования приложения предоставляет два типа правил автоматического масштабирования: правила ограничения и правила реагирования. В приложении Tailspin Surveys используются оба типа правил.

В Tailspin правила ограничения используются для указания минимального и максимального числа экземпляров веб-роли и рабочей роли. В Tailspin можно задавать минимальные значения, чтобы гарантировать соответствие соглашению об уровне обслуживания, и использовать максимальные значения для управления затратами. При этом используются ограничивающие правила, которые в разное время дня, когда прогнозируется изменение рабочей нагрузки приложения Surveys, задают разные максимальные и минимальные значения.

В Tailspin используются также правила реакции, позволяющие приложению Surveys реагировать на внезапные изменения нагрузки. Эти правила основаны на показателях, определенных в Tailspin для приложения Surveys. Часть этих показателей представляет собой стандартные счетчики производительности, другие используют специфические данные приложения Tailspin Surveys.

В дополнение к правилам, которые автоматически изменяют количество экземпляров роли в приложении Surveys, Tailspin использует правила реакции для управления поведением приложения. Эти правила введены, чтобы помочь приложению Surveys быстро реагировать на непредвиденные всплески нагрузки, до того как будут запущены новые экземпляры ролей.

Кроме того, имеющиеся в Tailspin операторы регулярно проверяют все диагностические данные, полученные от приложения Surveys. Это гарантирует оптимальность текущих правил, соблюдение соглашений об уровне обслуживания Tailspin и отсутствие лишних экземпляров ролей. Операторы Tailspin используют эти данные для уточнения набора правил, чтобы обеспечить приложению Surveys

точное соответствие требованиям соглашения об уровне обслуживания и оптимальные затраты.

Дополнительные сведения о функциях и использовании программного блока автомасштабирования см. в главе 4, «*Автомасштабирование и Windows Azure*».



## Размещение программного блока автомасштабирования в приложении Tailspin Surveys

В Tailspin было решено разместить программный блок автомасштабирования в веб-роли Windows Azure. Рассматривался также вариант его размещения в локальном приложении. Поскольку размещение программного блока автомасштабирования в локальном приложении не давало преимуществ, было решено отказаться от планов его интеграции с существующими локальными средствами протоколирования и диагностики. На следующей диаграмме показаны некоторые компоненты высокого уровня приложения Tailspin Surveys, участвующие в процессе автомасштабирования.

*Размещенная служба с автоматическим масштабированием Tailspin включает в себя веб-роль управления автомасштабированием и рабочую роль автомасштабирования. Эти две роли обычно разрабатываются совместно, что упрощает развертывание приложения Surveys для их размещения в одной службе.*

*Если веб-сайт управления, помимо управления процессом автомасштабирования, выполняет другие задачи, следует рассмотреть возможность использования отдельно размещенной службы для каждой роли, чтобы обеспечить их изолированную разработку и развертывание.*

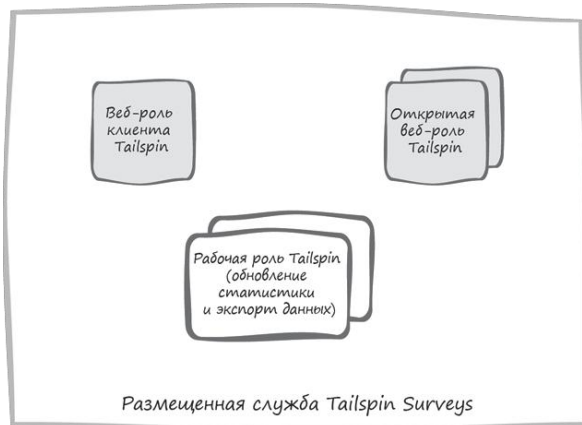


Рис. 1  
Компоненты,  
которые  
участвуют  
в автоматическом  
масштабировании  
в приложении  
Tailspin Surveys





Одна из причин, по которой приложение Surveys было разработано с использованием нескольких веб-ролей, — это различные предполагаемые модели использования приложения. Каждая веб- и рабочая роль может масштабироваться независимо от других.

В приложении Tailspin Surveys в результате масштабирования выигрывают и открытая веб-роль, используемая для отправки ответов на опросы, и рабочая роль. Ранее обе роли подвергались высокой нагрузке. В Tailspin не предполагают, что веб-роли клиентов будут создавать высокую нагрузку, так как клиенты выполняют в этой веб-роли и разработку новых опросов. Тем не менее в Tailspin решили добавить несколько правил автомасштабирования на случай любых непредвиденных всплесков нагрузки.

### Группы масштабирования в Tailspin Surveys

В приложении Tailspin Surveys имеется два типа ролей, которые было намечено масштабировать автоматически: открытая веб-роль и рабочая роль, несущая ответственность за вычисление статистики и экспорт данных. Эти две роли обладают разными шаблонами использования и нуждаются в применении разных правил масштабирования, поэтому не имеет смысла группировать их. Аналогично и экземпляры ролей каждого типа, которые выполняются в разных центрах данных Windows Azure, также должны масштабироваться независимо друг от друга. Поэтому Tailspin не использует группы масштабирования для программного блока автомасштабирования.



Сотрудники Tailspin предусмотрительно обеспечили возможность масштабирования веб-ролей клиента и администратора, чтобы в будущем их можно было автоматически масштабировать.

### Правила автомасштабирования в Tailspin Surveys

В следующей таблице представлен первоначальный набор ограничивающих правил, заданных в Tailspin для использования в приложении Surveys.

Описание	Ранг	Расписание	Роль	Максимальное количество экземпляров	Минимальное количество экземпляров
Ограничения по умолчанию для всех ролей	0	Круглые сутки, каждый день	Рабочая роль Tailspin	5	2
Ограничения по умолчанию для всех ролей	0	Круглые сутки, каждый день	Открытая веб-роль Tailspin	5	2
Ограничения по умолчанию для всех ролей	0	Круглые сутки, каждый день	Веб-роль клиента Tailspin	5	2
Дополнительные экземпляры для открытой веб-роли в часы пик	5	8:00–9:50 по понедельникам, вторникам, средам, четвергам и пятницам	Открытая веб-роль Tailspin	6	3

Эта таблица содержит первоначальный набор правил реакции, определенных в Tailspin для приложения Surveys.



Описание	Целевой объект	Действие
Рассмотрите количество отклоненных запросов и использование ЦП для открытого веб-сайта. Если оба показателя становятся слишком большим, следует увеличить количество экземпляров.	Открытая веб-роль приложения Tailspin Surveys	Добавьте один экземпляр
Рассмотрите количество отклоненных запросов и использование ЦП для открытого веб-сайта. Если оба показателя снизились до приемлемого уровня, следует уменьшить количество экземпляров.	Открытая веб-роль приложения Tailspin Surveys	Удаление одного экземпляра
Когда есть всплеск активности на веб-сайте клиента или в рабочей роли, следует включить регулирование. Это должно уменьшить нагрузку, и сайт останется работоспособным. Если нагрузка остается высокой, то вступят в силу правила масштабирования и увеличат количество экземпляров.	Рабочая роль Tailspin Surveys Веб-роль клиента Tailspin Surveys	Режим регулирования «ReduceSystemLoad» (уменьшение нагрузки на систему)
После окончания всплеска нагрузки следует отключить регулирование.	Рабочая роль Tailspin Surveys Веб-роль клиента Tailspin Surveys	Режим регулирования «Normal» (Обычный)
Когда есть много клиентов или опросов и наблюдается высокая загрузка ЦП, следует добавить дополнительные ресурсы для открытого веб-сайта.	Открытая веб-роль приложения Tailspin Surveys	Добавьте один экземпляр
Когда наблюдается обычное количество клиентов и опросов, следует уменьшить количество экземпляров.	Открытая веб-роль приложения Tailspin Surveys	Удаление одного экземпляра
Рассмотрите количество отклоненных запросов и использование ЦП для веб-сайта клиентов. Если один из показателей становится слишком большим, следует увеличить количество экземпляров.	Веб-роль клиента Tailspin Surveys	Добавьте один экземпляр

Минимальные значения имеют важное значение для компании Tailspin, так они обеспечивают выполнение соглашений об уровне обслуживания; максимальные значения ограничивают эксплуатационные расходы приложения.

Рассмотрите количество отклоненных запросов и использование ЦП для веб-сайта клиентов. Если нагрузка находится в допустимых пределах, следует уменьшить количество экземпляров.	Веб-роль клиента Tailspin Surveys	Удаление одного экземпляра
---	-----------------------------------	----------------------------

В обеих таблицах приведены первоначальные наборы правил, которые Tailspin применяет в рабочей версии приложения Surveys. Tailspin отслеживает и оценивает эти правила, определяя их степень их оптимальности при масштабировании. Tailspin планирует провести следующую настройку набора правил для повышения эффективности.

- Изменить пороговые значения, определяющие необходимость уменьшения или увеличения масштаба и включения или отключения регулирования.
- Добавить дополнительные ограничивающие правила для обработки других прогнозируемых изменений деятельности.
- Изменить метрики и интервалы, используемые правилами реакции для вызова действий масштабирования.
- Использовать разные наборы правил в разных центрах данных Windows Azure, чтобы адаптироваться к различиям их применения в соответствующих географических регионах.

Начиная с первоначального набора правил, Tailspin определил набор показателей, сбор которых необходимо настроить в программном блоке. В следующей таблице представлен первоначальный набор показателей, определенных Tailspin в качестве правил реакции в приложении Surveys.

Описание	Агрегатная функция	Промежуток времени	Источник	Показатель
Использование ЦП в открытой веб-роли Surveys	Среднее	20 минут	Открытая веб-роль Tailspin	\Процессор (_Общее)% процессорного времени
Использование ЦП в рабочей роли Surveys	Среднее	5 минут	Рабочая роль Tailspin	\Процессор (_Общее)% процессорного времени
Использование ЦП в веб-роли клиента Surveys	Среднее	20 минут	Веб-роль клиента Tailspin	\Процессор (_Общее)% процессорного времени
Использование ЦП в веб-роли клиента Surveys	Среднее	5 минут	Веб-роль клиента Tailspin	\Процессор (_Общее)% процессорного времени

Отклоненные запросы ASP.NET для открытой веб-роли Surveys	Среднее	10 минут	Открытая веб-роль Tailspin	\\ASP.NET\ Requests Rejected (отклоненные запросы)
Отклоненные запросы ASP.NET для веб-роли клиента Surveys	Среднее	10 минут	Веб-роль клиента Tailspin	\\ASP.NET\ Requests Rejected (отклоненные запросы)
Количество завершенных опросов	Среднее	10 минут	Приложение Surveys	Количество активных опросов в приложении Surveys
Количество клиентов	Среднее	10 минут	Приложение Surveys	Количество зарегистрированных клиентов
Количество экземпляров открытой веб-роли Surveys	Последнее значение	8 минут	Открытая веб-роль Tailspin	Количество экземпляров ролей



Tailspin разработала специальные операнды для сбора количества активных опросов и активных клиентов в приложении Surveys.

## Сбор данных для журнала автомасштабирования в Tailspin Surveys

Tailspin знает, что способы использования приложения Tailspin Surveys в различных географических регионах со временем меняются. Кроме того, Tailspin известно, что при тщательном анализе работы приложения Surveys можно выделить шаблоны использования.

Для автоматического масштабирования в Tailspin выбран упреждающий подход, когда правила ограничения получают предпочтение перед правилами реакции. Таким образом, можно попытаться подобрать нужное число активных экземпляров, чтобы при изменении рабочей нагрузки обеспечить соответствие соглашению об уровне обслуживания без добавления новых экземпляров. Для этого Tailspin ежемесячно проверяет данные журнала, собранные программным блоком автомасштабирования, чтобы выявлять новые модели использования или изменения в существующих шаблонах. При изменении способа использования будут либо изменены существующие правила ограничения, либо добавлены новые.

Tailspin по-прежнему поддерживает набор правил реакции, чтобы приложение Surveys могло реагировать на непредвиденные изменения рабочей нагрузки. Помимо этого, Tailspin анализирует, когда и почему выполняются эти правила реакции, чтобы убедиться в оптимальности масштабирования.

*Анализ работы приложения поможет оптимизировать правила автоматического масштабирования.*



Администраторы Tailspin обязательно должны иметь возможность изменять другие элементы, такие как список масштабируемых ролей в определении службы и данные подписки. Эти значения задаются один раз при первом развертывании приложения в Windows Azure.

## Пользовательский интерфейс конфигурации автомасштабирования

Хотя администраторы могут непосредственно изменять XML-файл, содержащий правила автомасштабирования Tailspin, данный процесс чреват ошибками. В этом случае редактор XML, учитывающий схему, может выполнить ряд проверок, но некоторые значения в файле определения правил относятся к записям в определении служебных данных, и проверкой XML не сможет обнаружить ошибки в ссылках. Кроме того, администратор должен поместить XML-файл с правилами в правильное место хранения, чтобы программный блок смог загрузить и использовать новые определения правил. Поэтому в Tailspin принято решение создать размещаемый на веб-узле редактор правил, который будет обрабатывать все выявляемые проверкой проблемы и сохранять правила в правильном месте.

Аналогичные проблемы связаны с XML-файлом, содержащим описание служебных сведений Survey. В Tailspin ожидается, что администраторам придется изменять определения групп масштабирования в этом файле, и возможно выполнение этой задачи в том же пользовательском интерфейсе, который служит для изменения правил.

## Уведомление операторов через SMS о выполнении операций масштабирования

Отправка уведомлений через SMS при выполнении действий масштабирования не является встроенной функцией программного блока автомасштабирования. В Tailspin было принято решение создать пользовательское действие для отправки SMS-уведомлений. Tailspin может добавить это пользовательское действие к выбранным правилам реакции, чтобы его операторы всегда получали уведомление о проведении важных операций масштабирования.

Хотя программный блок автомасштабирования содержит функцию, отправляющую уведомления при выполнении действий масштабирования, эта встроенная функция использует электронную почту, а в Tailspin для уведомления операторов предпочтение отдано SMS-сообщениям.

## Процесс реализации

Этот раздел содержит дополнительные сведения том, как в Tailspin размещали программный блок автомасштабирования и модифицировали приложения для работы с этим блоком. Если вам не интересны дополнительные сведения, можно перейти к следующему разделу.

Рекомендуем открыть решение Tailspin в среде разработки Microsoft® Visual Studio®, чтобы иметь возможность обращаться к коду при чтении данного раздела.

Инструкции по установке приложения Tailspin Surveys см. в приложении Б, «Руководство по установке Tailspin Surveys».

## Включение функции чтения из CSCFG-файла для программного блока автомасштабирования

Программный блок автомасштабирования считывает строки подключения из CSCFG-файла, чтобы получить доступ к хранилищу Windows Azure. Это делает метод `FromConfigurationSetting` класса `CloudStorageAccount`. Для этого приложению `Surveys` необходимо настроить издатель параметров конфигурации при запуске. В качестве примера можно привести следующий код из файла `Global.asax.cs` в проекте `Tailspin.Web.Management`.

```
C#
CloudStorageAccount.SetConfigurationSettingPublisher(
    (s, p) => p(RoleEnvironment.GetConfigurationSettingValue(s)));
```

Дополнительные сведения см. в разделе «*CloudStorageAccount.SetConfigurationSettingPublisher*» в библиотеке MSDN.

## Определение служебных сведений Tailspin

Следующий фрагмент кода показывает определение используемых по умолчанию сведений о службе для приложения `Tailspin Surveys`. `Tailspin` определяет содержимое этого файла один раз при начальном развертывании. Предполагается, что менять что-либо после запуска программного блока автомасштабирования не понадобится.

```
XML
<serviceModel ...>
  <subscriptions>
    <!--
      Todo when installing the RI for the first time:
      Update your subscription ID and Certificate Thumbprint
    -->
    <subscription name="TailspinSubscription"
      subscriptionId="[Enter subscription id here]"
      certificateThumbprint="
        [Enter certificate thumbprint here]"
      certificateStoreName="My"
      certificateStoreLocation="LocalMachine">
    <services>
      <service dnsPrefix="Tailspin-Surveys"
        slot="Staging" scalingMode="Scale">
        <roles>
          <role alias="SurveyWorkers"
            roleName="Tailspin.Workers.Surveys"
            wadStorageAccountName="TailspinStorage" />
          <role alias="PublicWebSite"
            roleName="Tailspin.Web.Survey.Public"
            wadStorageAccountName="TailspinStorage" />
```



Вы должны настроить издатель параметров конфигурации для каждой роли, которую отслеживает и масштабирует программный блок автоматического масштабирования.



Tailspin использует программный блок автоматического масштабирования для масштабирования только трех ролей, поэтому группы масштабирования не используются. Если в приложении используются группы масштабирования, их следует определить в файле определения данных службы.

```
<role alias="TenantWebSite"
      roleName="Tailspin.Web"
      wadStorageAccountName="TailspinStorage" />
</roles>
</service>
</services>
<storageAccounts>
  <!--
    Todo when installing the RI for the first time:
    Update the connection string to your storage account
  -->
  <storageAccount alias="TailspinStorage"
                 connectionString=
                 "[Enter connection string here]">
    <queues>
      <queue alias="SurveyAnswerStoredQueue"
            queueName="surveyanswerstored" />
      <queue alias="SurveyTransferQueue"
            queueName="surveytransfer" />
    </queues>
  </storageAccount>
</storageAccounts>
</subscription>
</subscriptions>
<scaleGroups />
<stabilizer scaleUpCooldown="00:10:00"
            scaleDownCooldown="00:10:00"
            notificationsCooldown="00:30:00">
  <role roleAlias="PublicWebSite" scaleUpCooldown="00:08:00"
        scaleDownCooldown="00:15:00" />
</stabilizer>
</serviceModel>
```

*При установке приложения Tailspin Surveys необходимо изменить этот файл, добавив в него сведения, специфичные для вашей учетной записи Windows Azure. Дополнительные сведения см. в разделе «Установка и физическое развертывание» этой главы. Образец файла определения сведений о службе находится в папке «Sample Stores» решения Visual Studio.*

В правилах автомасштабирования Tailspin Survey используются псевдонимы ролей и очередей. Элемент stabilizer показывает периоды снижения активности, заданные Tailspin. Включены глобальные параметры и конкретные параметры для общедоступного веб-сайта. Tailspin увеличил до 15 минут время, которое должно пройти после выполнения масштаби-



рования, прежде чем масштаб общедоступного веб-сайта может быть уменьшен. Увеличение масштаба общедоступного веб-сайта может произойти несколько раньше, чем выполнение других ролей.

## Правила автомасштабирования Tailspin

Следующие фрагменты кода показывают набор правил по умолчанию, которые Tailspin использовал при первом запуске программного блока автомасштабирования. В Tailspin планируется оценить эффективность этих правил при масштабировании приложения Surveys и внести изменения после того, как будет собрано достаточно данных для анализа поведения автомасштабирования в рабочей среде. Tailspin имеет встроенный веб-редактор правил для облегчения работы операторов по их изменению. Дополнительные сведения о веб-редакторе правил Tailspin см. в разделе «Редактирование и сохранение правил» этой главы.

## Правила ограничения в приложении Tailspin Surveys

В следующем фрагменте кода представлен первоначальный набор правил ограничения, настроенных Tailspin для использования в приложении Surveys. Существует правило по умолчанию, которое задает значения по умолчанию для всех ролей в приложении Tailspin Surveys; его ранг равен нулю. Второе правило повышает масштаб рабочей роли Tailspin Surveys в ожидании повышения уровня активности в течение рабочей недели.

### XML

```
<rules ...>
  <constraintRules>
    <rule name="Default constraints for all roles"
      description="This rule sets the default constraints for
        all web and worker roles. The minimum values guard our
        SLA, by ensuring there will never be less than these
        instances. The maximum values guard our wallet, by
        ensuring there will never be more than the configured
        number of instances."
      enabled="true" rank="0">
      <timetable startTime="00:00:00"
        duration="1.00:00:00" utcOffset="+00:00">
        <daily />
      </timetable>
      <actions>
        <range target="SurveyWorkers" min="2" max="5" />
        <range target="PublicWebSite" min="2" max="5" />
        <range target="TenantWebSite" min="2" max="5" />
      </actions>
    </rule>
  </constraintRules>
</rules>
```



Обратите внимание на то, как задаются пары правил реагирования: одно указывает, когда следует выполнять масштабирование с увеличением, а другое — с уменьшением.

```
</rule>
<rule name="Additional instances for public web during peak
hours" description="Our testing has indicated that
there will be additional load during peak hours. To
accommodate for that additional load, there will be
additional instances for the public website. These
peaks occur during working hours and early evenings.
By providing a higher rank, this rule takes precedence
over the default rule."
enabled="true" rank="5">
<timetable startTime="08:00:00" duration="09:50:00"
utcOffset="+00:00">
  <weekly days="Monday Tuesday Wednesday Thursday Friday" />
</timetable>
<actions>
  <range target="PublicWebSite" min="3" max="6" />
</actions>
</rule>
</constraintRules>
...
</rules>
```

### *Правила реактивного масштабирования в Tailspin Surveys*

Следующий фрагмент кода показывает первоначальное определение правил реактивного масштабирования Tailspin для приложения Surveys. Первая пара правил определяет, каким образом должно осуществляться увеличение или уменьшение масштаба открытой веб-роли Surveys исходя из числа отклоненных запросов ASP.NET и уровня загрузки ЦП. Вторая пара правил определяет, каким образом должно осуществляться увеличение или уменьшение масштаба открытой веб-роли Surveys исходя из числа клиентов и активных опросов. Третья пара правил определяет, каким образом должно осуществляться увеличение или уменьшение масштаба веб-роли клиента Surveys исходя из числа отклоненных запросов ASP.NET и уровня загрузки ЦП.

#### XML

```
<rules ...>
...
<reactiveRules>
...
  <rule name="Public Web - Heavy Demand (Increase)"
description="Look at number of rejected requests and
CPU for the public website. If either becomes too
large, then scale up the number of instances."
enabled="true">
```

```

<actions>
  <scale target="PublicWebSite" by="1" />
</actions>
<when>
  <all>
    <greater operand=
      "PublicWeb_AspNetRequestsRejected_Avg_10m" than="5" />
    <greater operand="PublicWeb_CPU_Avg_20m" than="80" />
  </all>
</when>
</rule>
<rule name="Public Web - Normal Demand (Reduce)"
  description="Look at number of rejected requests and
  CPU for the public website. If both drop to acceptable
  levels, then reduce the number of instances."
  enabled="true">
  <actions>
    <scale target="PublicWebSite" by="-1" />
  </actions>
  <when>
    <all>
      <lessOrEqual operand=
        "PublicWeb_AspNetRequestsRejected_Avg_10m" than="1" />
      <lessOrEqual operand="PublicWeb_CPU_Avg_20m" than="40" />
    </all>
  </when>
</rule>

<rule name="PublicWeb - Many Tenants Or Surveys (Increase)"
  description="When there are many tenants or surveys and
  the CPU usage is high, then we'll need more capacity in
  the public website. This rule demonstrates
  the use of the custom operands, called ActiveSurveyCount
  and TenantCount. Using the load simulation page, you can
  easily add and remove tenants and surveys to test the
  load on the system."
  enabled="true">
  <actions>
    <scale target="PublicWebSite" by="1" />
  </actions>
  <when>
    <all>
      <any>
        <greaterOrEqual operand=
          "Tailspin_ActiveSurveyCount_Avg_10m"

```

```

                than="50 * PublicWeb_InstanceCount_Last" />
        <greaterOrEqual operand="Tailspin_TenantCount_Avg_10m"
                than="50 * PublicWeb_InstanceCount_Last" />
    </any>
    <greater operand="PublicWeb_CPU_Avg_20m" than="50"/>
</all>
</when>
</rule>
<rule name="PublicWeb - Normal Tenants And Surveys (Decrease)"
    description="When there are a normal number of tenants and surveys, then
    decrease the number of instances."
    enabled="true">
    <actions>
        <scale target="PublicWebSite" by="-1" />
    </actions>
    <when>
        <all>
            <less operand="Tailspin_TenantCount_Avg_10m"
                than="30 * PublicWeb_InstanceCount_Last" />
            <less operand="Tailspin_ActiveSurveyCount_Avg_10m"
                than="30 * PublicWeb_InstanceCount_Last" />
        </all>
    </when>
</rule>

<rule name="TenantWeb - Heavy demand (Increase)"
    description="Look at number of rejected requests and
    CPU for the tenant website. If either becomes too
    large, then scale up the number of instances."
    enabled="true">
    <actions>
        <scale target="TenantWebSite" by="1" />
    </actions>
    <when>
        <all>
            <greaterOrEqual operand="TenantWeb_AspNetRequestsRejected_avg_10m"
                than="5" />
            <greaterOrEqual operand="TenantWeb_CPU_Avg_20m" than="80" />
        </all>
    </when>
</rule>
<rule name="TenantWeb - Normal Demand (Decrease)"
    description="Look at number of rejected requests and
    CPU for the tenant website. If the load is acceptable,
    then reduce the number of instances."

```

```

        enabled="true">
    <actions>
        <scale target="TenantWebSite" by="-1" />
    </actions>
    <when>
        <all>
<lessOrEqual operand=
    "TenantWeb_AspNetRequestsRejected_avg_10m" than="2" />
<lessOrEqual operand=
    "TenantWeb_CPU_Avg_20m" than="60"/>
</all>
        </when>
    </rule>
    ...
</reactiveRules>
    ...
</rules>

```

*Tails핀 не назначает ранг правилам реакции.*

### Правила регулирования реакции Tails핀 Surveys

В Tails핀 используются правила регулирования для динамического изменения поведения открытой веб-роли Surveys. Выполняется анализ загрузки ЦП, чтобы определить момент включения и отключения регулирования в приложении Surveys.

Дополнительные сведения о реализации регулирующего поведения в приложении Surveys см. в разделе «Реализация поведения регулирования» этой главы.

#### XML

```

<rules ...>
    ...
    <reactiveRules>
        ...
    </reactiveRules>
</rules>
<rule name=
    "TenantWeb & Survey Worker - Burst - Throttle"
    description="When there is a burst in activity on the
    tenant website and the worker role, then enable
    throttling. This should reduce the load so that the
    site remains workable. If the load remains high, then
    the scaling rules will kick in and increase the number
    of instances. Throttling in Tails핀 does the
    following:
    * Disable exporting of values to Microsoft SQL Server

```



Регулирование запускается с помощью действия `changeSetting`.

```

        in the worker role.
        * Only allow paying tenants to the tenant site. Tenants
          on a trial subscription cannot enter."
        enabled="true">
    <actions>
        <changeSetting target="SurveyWorkers"
settingName="ThrottlingMode"
                        value="ReduceSystemLoad" />
        <changeSetting target="TenantWebSite"
settingName="ThrottlingMode"
                        value="ReduceSystemLoad" />
    </actions>
    <when>
        <all>
            <greaterOrEqual operand="TenantWeb_CPU_Avg_5m" than="90" />
            <greaterOrEqual operand="SurveyWorkers_CPU_Avg_5m" than="90" />
        </all>
    </when>
</rule>
<rule name="TenantWeb & Survey Worker - Burst - Stop throttling"
description="When there is no burst in activity, then
disable throttling."
enabled="true">
    <actions>
        <changeSetting target="TenantWebSite" settingName="ThrottlingMode"
                        value="Normal" />
        <changeSetting target="SurveyWorkers" settingName="ThrottlingMode"
                        value="Normal" />
    </actions>
    <when>
        <any>
            <lessOrEqual operand="SurveyWorkers_CPU_Avg_5m" than="50" />
            <lessOrEqual operand="TenantWeb_CPU_Avg_5m" than="50" />
        </any>
    </when>
</rule>
...
</reactiveRules>
...
</rules>

```

## Операнды Tailspin Surveys

Помимо использования встроенных операндов счетчика производительности, в Tailspin были созданы два пользовательских операнда — `activeSurveysOperand` и `tenantCountOperand`, — которые позволяют использовать в правиле количество опросов с числом ответов, превышающим заданное значение, и количество клиентов.

### XML

```
<rules ...>
  ...
  <operands>
    <roleInstanceCount alias="PublicWeb_InstanceCount_Last" timespan="00:08:00"
      aggregate="Last" role="PublicWebSite" />
    <performanceCounter alias="PublicWeb_AspNetRequestsRejected_Avg_10m"
      timespan="00:10:00" aggregate="Average" source="PublicWebSite"
      performanceCounterName="\ASP.NET\Requests Rejected" />
    <performanceCounter alias="PublicWeb_CPU_Avg_20m" timespan="00:20:00"
      aggregate="Average" source="PublicWebSite"
      performanceCounterName="\Processor(_Total)\% Processor Time" />
    <activeSurveysOperand alias="Tailspin_ActiveSurveyCount_Avg_10m"
      timespan="00:10:00" aggregate="Average" minNumberOfAnswers="0"
      xmlns="http://Tailspin/ActiveSurveys" />
    <tenantCountOperand alias="Tailspin_TenantCount_Avg_10m" timespan="00:10:00"
      aggregate="Average" xmlns="http://Tailspin/TenantCount" />
    <performanceCounter alias="TenantWeb_AspNetRequestsRejected_avg_10m"
      timespan="00:10:00" aggregate="Average" source="TenantWebSite"
      performanceCounterName="\ASP.NET\Requests Rejected" />
    <performanceCounter alias="TenantWeb_CPU_Avg_20m" timespan="00:20:00"
      aggregate="Average" source="TenantWebSite"
      performanceCounterName="\Processor(_Total)\% Processor Time" />
    <performanceCounter alias="SurveyWorkers_CPU_Avg_5m" timespan="00:05:00"
      aggregate="Average" source="SurveyWorkers"
      performanceCounterName="\Processor(_Total)\% Processor Time" />
    <performanceCounter alias="TenantWeb_CPU_Avg_5m" timespan="00:05:00"
      aggregate="Average" source="TenantWebSite"
      performanceCounterName="\Processor(_Total)\% Processor Time" />
  </operands>
</rules>
```

Дополнительные сведения о реализации пользовательских операндов Tailspin содержатся в разделе «Реализация пользовательских операндов» этой главы.



приложение для сбора данных счетчиков производительности, используемых правилами реагирования, и для передачи таких данных в хранилище Windows Azure.

## Сбор в Tailspin Surveys данных для счетчика производительности

Правила реакции, применяемые Tailspin для приложения Surveys, используют данные счетчика производительности открытой и рабочей веб-ролей. Программный блок автомасштабирования ищет эти данные в таблице диагностики Windows Azure с именем **WAD PerformanceCountersTable** в хранилище Windows Azure. В Tailspin изменили общую веб-роль и рабочую роль приложения Surveys, чтобы сохранялись данные счетчика производительности, используемые программным блоком для оценки правил реакции.

В следующем образце кода из класса **WebRole** в Tailspin открытая веб-роль настраивает роль для сбора и сохранения данных счетчика производительности.

```
C#
...
using Microsoft.WindowsAzure;
using Microsoft.WindowsAzure.Diagnostics;
using Microsoft.WindowsAzure.ServiceRuntime;

public class WebRole : RoleEntryPoint
{
    public override bool OnStart()
    {
        var config =
            DiagnosticMonitor.GetDefaultInitialConfiguration();

        var cloudStorageAccount =
            CloudStorageAccount.Parse(
                RoleEnvironment.GetConfigurationSettingValue(
                    "Microsoft.WindowsAzure.Plugins.Diagnostics.
                    ConnectionString"));

        // Get the perf counters
        config.PerformanceCounters.ScheduledTransferPeriod =
            TimeSpan.FromMinutes(1);

        // Add the perf counters
        config.PerformanceCounters.DataSources.Add(
            new PerformanceCounterConfiguration
            {
                CounterSpecifier =
                    @"\Processor(_Total)\% ProcessorTime",
                SampleRate = TimeSpan.FromSeconds(30)
            });
        config.PerformanceCounters.DataSources.Add(
```



```

new PerformanceCounterConfiguration
{
    CounterSpecifier = @"\Process(aspnet_wp)\% Processor Time",
    SampleRate = TimeSpan.FromSeconds(30)
});
config.PerformanceCounters.DataSources.Add(
new PerformanceCounterConfiguration
{
    CounterSpecifier = @"\Process(aspnet_wp)\Private Bytes",
    SampleRate = TimeSpan.FromSeconds(30)
});
config.PerformanceCounters.DataSources.Add(
new PerformanceCounterConfiguration
{
    CounterSpecifier =
        @"\Microsoft® .NET CLR Exceptions\# Exceps thrown / sec",
    SampleRate = TimeSpan.FromSeconds(30)
});
config.PerformanceCounters.DataSources.Add(
new PerformanceCounterConfiguration
{
    CounterSpecifier = @"\ASP.NET\Requests Rejected",
    SampleRate = TimeSpan.FromSeconds(30)
});
config.PerformanceCounters.DataSources.Add(
new PerformanceCounterConfiguration
{
    CounterSpecifier = @"\ASP.NET\Worker Process Restarts",
    SampleRate = TimeSpan.FromSeconds(30)
});
config.PerformanceCounters.DataSources.Add(
new PerformanceCounterConfiguration
{
    CounterSpecifier = @"\Memory\Available Mbytes",
    SampleRate = TimeSpan.FromSeconds(30)
});

// Diagnostics Infrastructure logs
config.DiagnosticInfrastructureLogs.ScheduledTransferPeriod =
    System.TimeSpan.FromMinutes(1);
config.DiagnosticInfrastructureLogs.ScheduledTransferLogLevelFilter =
    LogLevel.Verbose;

// Windows Event Logs
config.WindowsEventLog.DataSources.Add("System!*");

```

```

        config.WindowsEventLog.DataSources.Add("Application!*");
config.WindowsEventLog.ScheduledTransferPeriod =
    TimeSpan.FromMinutes(1);
config.WindowsEventLog.ScheduledTransferLogLevelFilter =
    LogLevel.Warning;

        // Azure Trace Logs
config.Logs.ScheduledTransferPeriod =
    TimeSpan.FromMinutes(1);
config.Logs.ScheduledTransferLogLevelFilter =
    LogLevel.Warning;

        // Crash Dumps
CrashDumps.EnableCollection(true);

        // IIS Logs
config.Directories.ScheduledTransferPeriod =
    TimeSpan.FromMinutes(10);

        DiagnosticMonitor diagMonitor =
            DiagnosticMonitor.Start(cloudStorageAccount, config);

        return base.OnStart();
    }
}

```



Параметр файла конфигурации (CSCFG) **Microsoft.WindowsAzure.Plugins.Diagnostics.Connection** типа **String** определяет учетную запись хранилища Windows Azure, используемую для данных счетчика производительности.

Это должна быть та же учетная запись хранения, которая задана Tailspin в параметре **dataPointsStoreAccount** в конфигурации программного блока автоматического масштабирования.

## Реализация поведения регулирования

В Tailspin правила реакции используются для масштабирования и регулирования приложения Tailspin Surveys. Для реализации поведения регулирования в Tailspin изменили поведение приложения Surveys, когда правило реакции меняет режим регулирования. Следующий фрагмент кода показывает образец правил реакции, запрашивающих действия регулирования в приложении Surveys. Эти два образца правила присваивают значения параметру конфигурации **ThrottlingMode** в ролях Windows Azure.

```
XML
<rule name="TenantWeb & Survey Worker - Burst - Throttle"
      description="..."
      enabled="true">
  <actions>
<changeSetting target="SurveyWorkers"
              settingName="ThrottlingMode"
              value="ReduceSystemLoad" />
<changeSetting target="TenantWebSite"
              settingName="ThrottlingMode"
              value="ReduceSystemLoad" />
  </actions>
  <when>
    ...
  </when>
  <rank>0</rank>
</rule>
<rule name="TenantWeb & Survey Worker - Burst - Stop
      throttling"
      description="..."
      enabled="true">
  <actions>
<changeSetting target="TenantWebSite"
              settingName="ThrottlingMode"
              value="Normal" />
<changeSetting target="SurveyWorkers"
              settingName="ThrottlingMode"
              value="Normal" />
  </actions>
  <when>
    ...
  </when>
  <rank>0</rank>
</rule>
```

Эти параметры должны существовать в определениях служб целевой роли. Следующий фрагмент кода демонстрирует это для файла определения службы **Tailspin.Surveys.Cloud** (CSDEF).

```
XML
<WorkerRole name="Tailspin.Workers.Surveys">
  <ConfigurationSettings>
    ...
    <Setting name="ThrottlingMode" />
  </ConfigurationSettings>
  ...
</WorkerRole>
```

Приложение Surveys использует значение режима регулирования для изменения поведения приложения. Например, в рабочей роли Tailspin Surveys классы **QueueHandler** и **BatchProcessingQueueHandler** проверяют значение этого параметра перед обработкой сообщений. В следующем образце кода показано, как класс **TransferSurveysToSqlAzureCommand** проверяет параметры конфигурации.

```
C#
public bool CanRun
{
  get
  {
    return
      !this.configurationSettings.ConfigurationSettingEquals(
        AzureConstants.ConfigurationSettings.ThrottlingMode,
        "ReduceSystemLoad");
  }
}
```

Веб-роль клиента Tailspin также пользуется этим параметром для информирования пользователя о регулировании приложения. Файл `Index.aspx` в папке **Tailspin.Web.Areas.Survey.Views.Surveys** считывает значение конфигурации, а затем отображает сообщение о том, что для режима регулирования установлено значение **ReduceSystemLoad**.

## Редактирование и сохранение правил

Этот раздел описывает, как в Tailspin был создан веб-редактор правил автомасштабирования, который сохраняет правила в нужном месте, обеспечивает соответствие определения правил для приложения Surveys схеме правил программного блока автомасштабирования и выполняет перекрестную проверку с использованием определений служебных сведений приложения Surveys.

## Обнаружение расположения хранилища правил

Для загрузки и сохранения правил автомасштабирования Tailspin Survey в нужном месте приложение должно считать конфигурацию программного блока автомасштабирования из файла конфигурации web.config веб-роли, в которой размещен блок. Этот процесс наглядно показан в следующем образце кода из класса **SharedContainerBootstrapper** проекта **Tailspin.Shared**.

```
C#
private static IConfigurationFileAccess
CreateServiceInformationModelConfigurationFileAccess
(IUnityContainer container)
{
    AutoscalingSettings settings =
        (AutoscalingSettings)ConfigurationManager
        .GetSection("autoscalingConfiguration");

    BlobServiceInformationStoreData serviceInformationStoreData =
        (BlobServiceInformationStoreData)settings
        .ServiceInformationStores
        .Get(settings.ServiceInformationStoreName);

    return new BlobConfigurationFileAccess(
        new AzureStorageAccount(serviceInformationStoreData.
            StorageAccount),
        serviceInformationStoreData.BlobContainerName,
        serviceInformationStoreData.BlobName,
        serviceInformationStoreData.MonitoringRate,
        container.Resolve<ILogger>());
}
```

## Чтение и запись в хранилище правил

Программный блок автомасштабирования включает класс **RuleSetSerializer**, который использует экземпляры класса **RuleSetElement** для выполнения сериализации и десериализации данных для хранилища правил. Методы **LoadRuleSet** и **SaveCurrentRuleSet** класса **RuleSetModelStore** в папке **AutoScaling/Rules** Проекта **Tailspin.Shared** показывают использование класса **RuleSetSerializer** в приложении Surveys.

```

C#
using
    Microsoft.Practices.EnterpriseLibrary.WindowsAzure.Autoscaling.
        Rules.Configuration;
...
private readonly IConfigurationFileAccess fileAccess;
private readonly RuleSetModelToXmlElementConverter
        ruleSetModelToXmlElementConverter;
...
private RuleSetModel currentRuleSet;
private RuleSetSerializer serializer;
public RuleSetModelStore(
    RuleSetModelToXmlElementConverter
    ruleSetModelToXmlElementConverter,
    [Dependency("RuleSetModel")] IConfigurationFileAccess
    fileAccess,
    RetryManager retryManager)
{
    ...
    this.ruleSetModelToXmlElementConverter =
        ruleSetModelToXmlElementConverter;
    this.fileAccess = fileAccess;

    this.CreateSerializer();

    this.LoadRuleSet();
}

private void CreateSerializer()
{
    var allExtensions = new IRuleSerializationExtension[]
    {
        new AssemblyRuleSerializationExtension(
            typeof(ActiveSurveysOperandElement).Assembly.FullName)
    };

    this.serializer = new RuleSetSerializer(
allExtensions.SelectMany(e => e.CustomActionDefinitions),
allExtensions.SelectMany(e =>
        e.CustomParameterDefinitions));
}

private void LoadRuleSet()
{
    string fileContent = this.GetFileContent();
}

```

```

RuleSetElement ruleSetElement;
if (string.IsNullOrEmpty(fileContent))
{
    ruleSetElement = new RuleSetElement();
}
else
{
    ruleSetElement = this.serializer.Deserialize(new
        StringReader(fileContent));
}

this.currentRuleSet = this.ruleSetModelToXmlElementConverter
    .ConvertElementToModel(ruleSetElement);
}

...
public void SaveCurrentRuleSet()
{
    lock (this.syncRoot)
    {
        var writer = new StringWriter();
        RuleSetElement element =
            this.ruleSetModelToXmlElementConverter
                .ConvertModelToElement(this.currentRuleSet);
        this.serializer.Serialize(writer, element);
        this.SetFileContent(writer.ToString());
    }
}

```

### **Создание допустимых правил автоматического масштабирования**

Чтобы редактор правил приложения Surveys создавал такие правила автомасштабирования, которые соответствуют схеме правил автомасштабирования, в Tailspin используют классы из пространства имен **Microsoft.Practices.EnterpriseLibrary.WindowsAzure.Autoscaling.Rules.Configuration** в программном блоке автомасштабирования. Например, класс **Constraint RuleToXmlElementConverter** выполняет преобразования класса **Constraint RuleModel**, используемого редактором правил Tailspin Surveys, в класс **ConstraintRuleElement**, используемым программным блоком автомасштабирования. Дополнительные примеры см. в других классах преобразователей в пространстве имен **Tailspin.Shared.AutoScaling.Rules.XmlElementConverters**.



Проще привязать классы модели Tailspin к пользовательскому интерфейсу, чем привязать классы элементов блока автоматического масштабирования к пользовательскому интерфейсу.

## Проверка целевых имен в определениях правил

Целевые имена в действиях правил являются псевдонимами для ролей и групп масштабирования. Имя источника для некоторых операндов в определениях правил также является псевдонимом для роли. Эти псевдонимы задаются в определении службы приложения. Текстовое поле в редакторе правил Tailspin Surveys, в которое пользователь вводит целевое имя, поддерживает автоматическое завершение на основе списка псевдонимов ролей и имен групп масштабирования из определения службы. Следующий образец кода из файла `_ConstraintRuleActionEditor.cshtml` проекта **Tailspin.Web.Management** демонстрирует построение элемента пользовательского интерфейса.

### CSHTML

```
<tr>
  <td>@Html.HiddenFor(m => m.Id)@Html.EditorFor(m => m.Target,
    "_AutoComplete", new { @class = "targetTextBox",
    url = @Url.Action("GetTargets", "Home",
    new { Area = "ServiceInformation" }),
    placeholder =
    "{Target Name}" })@Html.ValidationMessageFor(m => m.Target)
  </td>
  ...
</tr>
```

Следующий образец кода из класса **HomeController** в пространстве имен **Tailspin.Web.Management.Areas.ServiceInformation.Controllers** показывает метод **GetTargets**, который вызывается из показанного выше представления.

### C#

```
public ActionResult GetTargets()
{
    var roleAliases =
        this.serviceInformationModelStore.GetAllRoles()
            .Select(r => r.Alias);
    var scaleGroups =
        this.serviceInformationModelStore
            .GetScaleGroups().Select(r => r.Alias);

    return this.Json(roleAliases.Union(scaleGroups).Where
        (s => !string.IsNullOrEmpty(s)),
        JsonRequestBehavior.AllowGet);
}
```



## Редактирование и сохранение сведений о службе

Реализация функций подготовки сведений о службе, описанных в этом разделе, очень похожа на реализацию функций редактирования и сохранения правил в Tailspin Surveys.

Класс **ServiceInformationModelStore** отвечает за обнаружение расположения хранилища сведений о службе в файле конфигурации программного блока, а также отвечает за обеспечение чтения и записи сведений о службе приложением Surveys. Программный блок не содержит пользовательского класса сериализатора XML для сведений о службе, поэтому в классе **ServiceInformation ModelStore** используется экземпляр **XMLSerializer** для десериализации и сериализации в XML.

Классы модели сведений о службе в пространстве имен **Tailspin.Shared.AutoScaling.ServiceInformation** обеспечивают их преобразование в классы элементов в программном блоке автомасштабирования **Microsoft.Practices.EnterpriseLibrary.WindowsAzure.Autoscaling.Пространство имен ServiceModel.Configuration**. Tailspin использует классы элементов в блоке, чтобы гарантировать, что приложение Surveys создаст допустимый XML-документ со сведениями о службе для сохранения в хранилище.

## Визуализация действий автомасштабирования

Операторы в Tailspin хотели бы иметь возможность наблюдать за автомасштабированием, которое выполняется в приложении Surveys, чтобы понять, как работает программный блок автомасштабирования. В Tailspin разработан ряд диаграмм визуализации, которые строят график зависимости числа экземпляров от времени и показывают максимальные и минимальные ограничивающие значения.

Для создания этих диаграмм Tailspin необходимы сведения о текущем и предыдущем количестве экземпляров для всех ролей, которыми управляет блок, а также информация о максимальных и минимальных ограничивающих значениях, которые были определены для показанного на диаграмме периода, чтобы также представить их на диаграммах.

Для каждой роли, которой управляет Windows Azure, программный блок автомасштабирования записывает число ее экземпляров в виде точки данных в хранилище точек данных. Это хранилище представляет собой таблицу в хранилище Windows Azure.

Каждый раз, когда блок оценивает свои правила автомасштабирования, он вносит в журнал запись, которая содержит подробные сведения о максимальном и минимальном числе экземпляров, которые были допустимы во время оценки правил, а также подробные сведения о действиях масштабирования, которые были предложены правилами реакции (если таковые имеются).

Следующий образец кода из класса **GraphController** в пространстве имен **Tailspin.Web.Management.Areas.Monitoring.Controllers** показывает, как веб-сайт управления извлекает из хранилища точек данных значение счетчика числа экземпляров, которое будет отображено на диаграмме.

```

C#
private void AddInstanceCountSeries(Chart chart,
                                     DateTimeFilter dateTimeFilter,
                                     string sourceName, string sourceAlias)
{
    IEnumerable<DataPoint> dataPoints = this.dataPointsStore.Get(
        sourceAlias, "RoleInstanceCount",
        "RoleInstanceCount",
        dateTimeFilter.GetEffectiveStartDate(),
        dateTimeFilter.GetEffectiveEndDate());

    Series series = chart.Series.Add(sourceName);
    series.ChartType = SeriesChartType.StepLine;
    series.ToolTip =
        "TimeStamp = #VALX{d} \n Number of instances = #VALY{d}";
    series.ChartArea = ChartArea;
    series.BorderWidth = 5;
    series.Color = this.GetRoleColor(sourceAlias);

    foreach (DataPoint dp in dataPoints)
    {
        series.Points.AddXY(dp.DataTimestamp.DateTime.
            ToLocalTime(), dp.Value);
    }

    if (!dataPoints.Any())
    {
        series.Name += " (No matching datapoints found)";
    }

    AddEmptyStartEndPoints(series, dateTimeFilter);
    this.RememberMaximum(chart,
        dataPoints.MaxOrZero(m => m.Value));
}

```

В этом примере строка **sourceName** представляет имя роли. Для указания диапазона извлекаемых точек данных Tailspin использует объект **DateTimeFilter**. Метод **Get** предоставляется классом **AzureStorage DataPointStore** в программном блоке автомасштабирования.

Следующий образец кода из класса **GraphController** в пространстве имен **Tailspin.Web.Management.Areas.Monitoring.Controllers** показывает процесс извлечения веб-сайтом управления из хранилища точек данных максимального и минимального разрешенных значений для числа экземпляров, которые будут отображены на диаграмме.

C#

```
private void AddMinMaxSeries(Chart chart, DateTimeFilter
    dateTimeFilter, string sourceName, string sourceAlias)
{
    IEnumerable<WADLogsTableEntity> minMaxLogMessages =
        this.logDataStore.Get(
            dateTimeFilter.GetEffectiveStartDate(),
            dateTimeFilter.GetEffectiveEndDate(),
            Constants.Scaling.Events.RequestForConfigurationChange);

    List<MinMaxInstanceCountDataPoint> minMaxLogMessagesForRole =
        minMaxLogMessages.SelectMany(
            l => this.CreateMinMaxModels(l, sourceAlias)).ToList();

    Series minSeries = chart.Series.Add(string.Empty);
    minSeries.ChartArea = ChartArea;
    minSeries.ChartType = SeriesChartType.StackedArea;
    minSeries.IsVisibleInLegend = false;
    minSeries.Color = Color.Transparent;

    foreach (MinMaxInstanceCountDataPoint minMaxLogMessage in
        minMaxLogMessagesForRole)
    {
        minSeries.Points.AddXY(
            minMaxLogMessage.EventDateTime.ToLocalTime(),
            minMaxLogMessage.MinInstanceCount);
    }

    Series maxSeries = chart.Series.Add("Minimum and Maximum
        instance count");
    maxSeries.ChartArea = ChartArea;
    maxSeries.ChartType = SeriesChartType.StackedArea;
    maxSeries.Color = Color.FromArgb(98, 0, 73, 255);
        // Transparent blue

    foreach (MinMaxInstanceCountDataPoint minMaxLogMessage in
        minMaxLogMessagesForRole)
    {
        var index = maxSeries.Points.AddXY(
        var index = maxSeries.Points.AddXY(
            minMaxLogMessage.EventDateTime.ToLocalTime(),
            minMaxLogMessage.MaxInstanceCount -
            minMaxLogMessage.MinInstanceCount);
        maxSeries.Points[index].ToolTip = string.Format(
            "Min Instance Count = {0}\nMax Instance Count = {1}",
```



Если удаление старых данных из журнала диагностики Windows Azure происходит регулярно, это может привести к ограничению диапазона просмотра таких данных на графике.

```
minMaxLogMessage.MinInstanceCount,  
+minMaxLogMessage.MaxInstanceCount);  
}  
  
if (!minMaxLogMessagesForRole.Any())  
{  
    minSeries.Name += "(No matching datapoints found)";  
    maxSeries.Name += "(No matching datapoints found)";  
}  
  
this.RememberMaximum(chart,  
    minMaxLogMessagesForRole.MaxOrZero(m =>  
        m.MaxInstanceCount));  
}
```

В этом примере строка **sourceName** опять представляет имя роли, а Tailspin снова указывает диапазон точек данных для извлечения с помощью объекта **DateTimeFilter**. В этом примере Tailspin реализует метод **Get** класса **AzureStorageWadLogDataStore** в пространстве имен **Tailspin.Web.Management.Areas.Monitoring.Models**.

При построении диаграмм Tailspin использует элементы управления построения диаграмм ASP.NET для включения интерактивного поведения. Пользователь может щелкнуть диаграмму, чтобы получить более подробные сведения о точках данных.

## Реализация пользовательского действия

В этом разделе описано, как в Tailspin реализовали пользовательское действие, которое может быть использовано вместе с существующими действиями масштабирования для уведомления операторов о выполнении важных операций масштабирования через SMS-сообщение. Программный блок автомасштабирования предоставляет точку расширения для создания пользовательских действий. В Tailspin должны гарантировать, что пользовательский интерфейс редактирования правил сможет загружать определения пользовательских действий в хранилище правил и сохранять их там.



Если вы хотите узнать больше о том, как приложение Tailspin Surveys формирует графики, взгляните на классы в пространстве имен **Tailspin.Web.Management.Areas.Monitoring**.

## Интеграция пользовательского действия с программным блоком автомасштабирования

Действия являются частью правил реагирования в автомасштабировании, которые программный блок считывает из хранилища правил. Tailspin по умолчанию использует хранилище правил XML больших двоичных объектов, поэтому программный блок должен иметь возможность выполнять десериализацию пользовательского действия из XML-документа.

Следующий фрагмент кода показывает, как можно добавить пользовательское действие в хранилище правил.

*В настоящее время Tailspin не использует это пользовательское действие.*

```
XML
<reactiveRules>
  <rule ...>
    ...
    <actions>
      <smsAction xmlns="http://Tailspin/SendSMS"
        phoneNumber="+8888" message="Alert, reactive rule..."/>
    </actions>
  </rule>
</reactiveRules>
```

Сначала в Tailspin создали класс, показанный в следующем образце кода, для выполнения десериализации пользовательского действия из хранилища правил XML. Обратите внимание на то, как используются атрибуты для определения элемента, атрибутов и пространства имен XML.

```
C#
[XmlRoot(ElementName = "smsAction",
  Namespace = "http://Tailspin/SendSMS")]
public class SendSmsActionElement : ReactiveRuleActionElement
{
  [XmlAttribute("phoneNumber")]
  public string PhoneNumber { get; set; }

  [XmlAttribute("message")]
  public string Message { get; set; }

  public override ReactiveRuleAction CreateAction()
  {
    return new SendSmsAction
    {
      Message = this.Message,

```



Для Tailspin добавление пользовательских действий требует внесения двух наборов связанных изменений. Во-первых, следует убедиться в том, что программный блок автоматического масштабирования знает о пользовательском действии, а во-вторых, убедиться в том, что пользовательский интерфейс для редактирования правил знает о пользовательском действии.

Если операторы Tailspin редактируют правила в редакторе XML, Tailspin может добавить в редактор функции проверки и IntelliSense®, если для пространства имен `http://Tailspin/SendSMS` была создана XML-схема.

```
        PhoneNumber = this.PhoneNumber
    };
}
}
```

Метод **CreateAction** возвращает экземпляр **SendSmsAction**, выполняющий пользовательское действие. Следующий фрагмент кода показывает класс **SendSmsAction**, который расширяет класс **ReactiveRuleAction**.

```
C#
public class SendSmsAction : ReactiveRuleAction
{
    public SendSmsAction()
    {
    }

    public string PhoneNumber { get; set; }

    public string Message { get; set; }

    public override IEnumerable<RuleEvaluationResult> GetResults(
        ReactiveRule forRule, IRuleEvaluationContext context)
    {
        return new[]
        {
            new SendSmsActionResult(forRule)
            {
                Message = this.Message,
                PhoneNumber = this.PhoneNumber
            }
        };
    }
}
```

Средство оценки правил в блоке вызывает метод **GetResults** для всех действий текущего правила, а затем метод **Execute** для каждого возвращенного объекта **RuleEvaluationResult**. Следующий фрагмент кода показывает класс **SendSmsActionResult** (класс **ExecuteActionResult** расширяет класс **RuleEvaluationResult**).

C#

```
public class SendSmsActionResult : ExecuteActionResult
{
    private readonly ISmsSender smsSender;

    public SendSmsActionResult(Rule sourceRule)
        : base(sourceRule)
    {
        this.smsSender =
EnterpriseLibraryContainer.Current.GetInstance<ISmsSender>();
    }

    public string PhoneNumber { get; set; }

    public string Message { get; set; }

    public override string Description
    {
        get
        {
            return string.Format("Sends an SMS to number: '{0}'
with message: '{1}'", this.PhoneNumber, this.Message);
        }
    }

    public override void Execute(IRuleEvaluationContext context)
    {
        this.smsSender.Send(this.PhoneNumber, this.Message);
    }
}
```

Блок использует свойство **Description** при регистрации отправки SMS-сообщения в журнале.

И наконец, Tailspin с помощью средства настройки Enterprise Library передает программному блоку автомасштабирования сведения о пользовательском действии.

XML

```
<autoscalingConfiguration ... >
...
<rulesStores>
    <add name="Blob Rules Store" type=... >
        <extensionAssemblies>
            <add name="Tailspin.Shared" />
        </extensionAssemblies>
    </add>
```



Если вызывается исключение в методе **Execute**, оно должно иметь тип **ActionExecutionException**.



В примере загружается сборка, содержащая пользовательский класс **ActiveSurveysOperandElement**. Эта сборка также содержит пользовательский класс **SendSmsActionElement**. Расширение загружается прямо в код, поскольку веб-сайт управления не содержит программного блока автоматического масштабирования и поэтому не может использовать параметр конфигурации для его загрузки.

```
</rulesStores>  
...  
</autoscalingConfiguration>
```

Элемент **ExtensionAssemblies** добавляет имя сборки, где содержатся классы, определяющие пользовательское действие.

## *Интеграция пользовательского действия в редактор правил Tailspin Surveys*

Редактор правил Tailspin Surveys дает администраторам возможность изменять правила автоматического масштабирования для приложения Surveys в пользовательском веб-интерфейсе. Этот редактор может считывать определения правил и сохранять их в хранилище правил, которое использует программный блок автоматического масштабирования. Блок считает, что хранилище правил доступно только для чтения, однако он содержит класс **RuleSetSerializer**, обеспечивающий поддержку сохранения определений правил в хранилище.

Tailspin настраивает экземпляр **RuleSetSerializer**, который используется редактором правил, задавая для него сведения о пользовательском действии и об операнде. Следующий фрагмент кода из класса **RuleSetModelStore** демонстрирует процесс загрузки двух расширений (пользовательское действие и операнд).

```
C#  
private RuleSetSerializer serializer;  
  
public RuleSetModelStore(RuleSetModelToXmlElementConverter  
    ruleSetModelToXmlElementConverter,  
    [Dependency("RuleSetModel")] IConfigurationFileAccess  
    fileAccess, RetryManager retryManager)  
{  
    ...  
    this.CreateSerializer();  
    ...  
}  
  
private void CreateSerializer()  
{  
    var allExtensions = new IRuleSerializationExtension[]  
    {  
        new AssemblyRuleSerializationExtension(  
            typeof(ActiveSurveysOperandElement).Assembly.FullName)  
        };  
    this.serializer = new RuleSetSerializer(  
        allExtensions.SelectMany(e => e.CustomActionDefinitions),  
        allExtensions.SelectMany(e => e.CustomParameterDefinitions));  
}
```



После добавления расширений в сериализатор редактор правил может загружать и сохранять правила с пользовательскими действиями и операндами, которые были созданы Tailspin.

Не нужно загружать сборку, программно содержащую ваши расширения в проекте, в котором размещен программный блок автомасштабирования, поскольку он уже содержит код, загружающий сборки согласно записям в разделе **auto scalingConfiguration** файла конфигурации. В решении Tailspin Surveys рабочая роль **Tailspin.Workers.Autoscaling** содержит программный блок автомасштабирования и загружает расширения автоматически. Однако веб-роль **Tailspin.Web.Management** (не содержащая программного блока автомасштабирования) должна загружать расширения программно.



Для пользовательского действия необходимо расширить классы **Reactive RuleAction** и **Execute** Классы **ActionResult**; для пользовательского операнда необходимо предоставить реализацию интерфейса **IDataPoints Collector**.

## Реализация пользовательских операндов

Процесс создания пользовательского операнда очень похож на процесс создания пользовательского действия. Tailspin реализует два пользовательских операнда, которые позволяют правилам использовать количество активных опросов и текущее количество клиентов в качестве показателей правила реагирования.

Программный блок автомасштабирования предоставляет точку расширения для создания пользовательских операндов. Tailspin должен также позаботиться о том, чтобы пользовательский интерфейс редактирования правил мог загружать и сохранять пользовательские операнды в хранилище правил.

## Интеграция пользовательского операнда с программным блоком автомасштабирования

Операнды являются частью правил реагирования в автомасштабировании, которые программный блок считывает из хранилища правил. Tailspin по умолчанию использует хранилище правил XML больших двоичных объектов, поэтому Tailspin необходимо предусмотреть возможность десериализации пользовательского операнда из XML-документа.

Следующий фрагмент содержит образец пользовательского операнда Tailspin **active SurveysOperand** в хранилище правил.

```
XML
<operands>
  ...
  <activeSurveysOperand
    alias="Tailspin_ActiveSurveyCount_Avg_10m"
    timespan="00:10:00"
    aggregate="Average"
    minNumberOfAnswers="0"
    xmlns="http://Tailspin/ActiveSurveys" />
  ...
</operands>
```



Если операторы Tailspin редактируют правила в редакторе XML, Tailspin может добавить в редактор функции проверки и технологии IntelliSense, если в пространствах имен *http://Tailspin/ActiveSurveys* и *http://Tailspin/TenantCount* была создана схема XML.

Сначала в Tailspin создали класс, показанный в следующем образце кода, для выполнения десериализации пользовательского операнда **activeSurveysOperand** из хранилища правил XML. Обратите внимание на то, как используются атрибуты для определения элемента, атрибутов и пространства имен XML.

```
C#
[XmlRoot(ElementName = "activeSurveysOperand",
          Namespace = "http://Tailspin/ActiveSurveys")]
public class ActiveSurveysOperandElement :
    DataPointsParameterElement
{
    [XmlAttribute("minNumberOfAnswers")]
    public int MinNumberOfAnswers{ get; set; }

    protected override string DataPointName
    {
        get
        {
            return this.DataPointType;
        }
    }

    protected override string DataPointType
    {
        get
        {
            return "Number of Active Surveys";
        }
    }

    protected override string SourceName
    {
        get
        {
            return "Tailspin";
        }
    }
}
```

```

protected override Func<IServiceInformationStore,
    IEnumerable<IDataPointsCollector>> GetCollectorsFactory()
{
    var samplingRate = ActiveSurveysDataPointsCollector
        .DefaultPerformanceCounterSamplingRate;
    return (sis) =>
        new[]
        {
            new ActiveSurveysDataPointsCollector(
                EnterpriseLibraryContainer.Current
                    .GetInstance<ISurveyStore>(),
                EnterpriseLibraryContainer.Current
                    .GetInstance<ISurveyAnswersSummaryStore>(),
                samplingRate,
                this.MinNumberOfAnswers,
                this.SourceName,
                this.DataPointType,
                this.DataPointName)
        };
}
}

```

Свойство **MinNumberOfAnswers** определяет необязательный атрибут, который Tailspin использует для фильтрации обрабатываемого списка опросов. Например, если в Tailspin установят для атрибута операнда **minNumberOfAnswers** значение 5000, то **activeSurveysOperand** будет подсчитывать только те опросы, которые к настоящему моменту имеют не менее 5000 собранных ответов.

Метод **GetCollectorsFactory** создает объект **Active SurveysDataPointsCollector**, выполняющий операции сбора пользовательских данных. В следующем фрагменте кода представлен класс **ActiveSurveysDataPointsCollector**, который реализует интерфейс **IDataPointsCollector**. Этот класс отвечает за сбор точек данных. Метод **Collect** с помощью метода **FilterSurveys** извлекает только те опросы, которые имеют по крайней мере минимальное число ответов, обозначенное атрибутом **minNumberOfAnswers** в хранилище правил.

C#

```
public class ActiveSurveysDataPointsCollector : IDataPointsCollector
{
    private readonly ISurveyStore surveyStore;

    private readonly ISurveyAnswersSummaryStore surveyAnswersSummaryStore;

    private readonly TimeSpan samplingRate;

    private readonly int minimumNumberOfAnswers;

    private readonly string sourceName;

    private readonly string dataPointType;

    private readonly string dataPointName;

    public ActiveSurveysDataPointsCollector(ISurveyStore surveyStore,
        ISurveyAnswersSummaryStore surveyAnswersSummaryStore,
        TimeSpan samplingRate, int minNumberOfAnswers, string sourceName, string
        dataPointType, string dataPointName)
    {
        this.surveyStore = surveyStore;
        this.surveyAnswersSummaryStore = surveyAnswersSummaryStore;
        this.samplingRate = samplingRate;
        this.minimumNumberOfAnswers = minNumberOfAnswers;
        this.sourceName = sourceName;
        this.dataPointType = dataPointType;
        this.dataPointName = dataPointName;
    }

    public static TimeSpan DefaultPerformanceCounterSamplingRate
    {
        get { return TimeSpan.FromMinutes(2); }
    }

    public TimeSpan SamplingRate
    {
        get { return this.samplingRate; }
    }

    public string Key
    {
        get { return string.Format(CultureInfo.InvariantCulture,
            "{0}|{1}", this.minimumNumberOfAnswers, this.samplingRate); }
    }
}
```

```

    }

    public IEnumerable<DataPoint> Collect(DateTimeOffset collectionTime)
    {
        IEnumerable<Survey> surveys;
        try
        {
            surveys = this.surveyStore.GetActiveSurveys(FilterSurveys).ToList();
        }
        catch (StorageClientException ex)
        {
            throw new DataPointsCollectionException(
                "Could not retrieve surveys", ex);
        }

        return new[]
        {
            new DataPoint
            {
                CreationTime = collectionTime,
                Source = this.sourceName,
                Type = this.dataPointType,
                Name = this.dataPointName,
                Value = surveys.Count(),
                DataTimestamp = collectionTime
            }
        };
    }

    private bool FilterSurveys(string tenantname, string slugname)
    {
        if (this.minimumNumberOfAnswers == 0)
        {
            return true;
        }
        var answersSummary =
            this.surveyAnswersSummaryStore.GetSurveyAnswersSummary(
                tenantname, slugname);
        if (answersSummary == null)
        {
            return false;
        }
        return answersSummary.TotalAnswers > this.minimumNumberOfAnswers;
    }
}

```



Обратите внимание, что метод **Collect** может вызвать исключение типа **DataPointsCollectionException**. Все исключения, возникшие в этом методе, должны иметь этот тип.

И наконец, Tailspin с помощью средства настройки Enterprise Library передает программному блоку автомасштабирования сведения о пользовательском действии. Поскольку пользовательский операнд и пользовательское действие находятся в одной сборке, элемент **extensionAssemblies** содержит только одну запись.

```
XML
<autoscalingConfiguration ... >
  ...
  <rulesStores>
    <add name="Blob Rules Store" type=... >
      <extensionAssemblies>
        <add name="Tailspin.Shared" />
      </extensionAssemblies>
    </add>
  </rulesStores>
  ...
</autoscalingConfiguration>
```

### *Интеграция пользовательского операнда в редактор правил Tailspin Surveys*

Интеграция выполняется так же, как и интеграция пользовательского действия с помощью редактора правил. Поскольку пользовательский операнд и пользовательское действие находятся в одной сборке, метод **CreateSerializer** класса Rule **SetModelStore** добавляет только одну сборку расширения.

### **Настройка ведения журналов в Tailspin Surveys**

Программный блок автомасштабирования позволяет выбрать реализацию модуля ведения журналов. Так как программный блок автомасштабирования размещается в рабочей роли Windows Azure и Tailspin не требует наличия дополнительных функций, предлагаемых программным блоком ведения журнала Enterprise Library, Tailspin использует инфраструктуру журналов, определенную в пространстве имен **System.Diagnostics**. Следующий фрагмент конфигурации рабочей роли Windows Azure, в котором находится программный блок автомасштабирования, показывает конфигурацию ведения журнала в Tailspin Surveys. Раздел **AutoscalingConfiguration** выбирает инфраструктуру ведения журнала диагностики систем для программного блока автомасштабирования, а раздел **system.diagnostics** настраивает источники ведения журнала для сообщений журнала от блока.

**XML**

```
<autoscalingConfiguration loggerName="Source Logger" ...>
  <loggers>
    <add name="Source Logger" type="Microsoft.Practices.EnterpriseLibrary
      .WindowsAzure.Autoscaling.Logging.SystemDiagnosticsLogger,
      Microsoft.Practices.EnterpriseLibrary.WindowsAzure.Autoscaling />
  </loggers>
  ...
</autoscalingConfiguration>
...
<system.diagnostics>
  <sources>
    <source name="Autoscaling General" switchValue="All">
      <listeners>
        <add name="AzureDiag" />
        <remove name="Default" />
      </listeners>
    </source>
    <source name="Autoscaling Updates" switchValue="All">
      <listeners>
        <add name="AzureDiag" />
        <remove name="Default" />
      </listeners>
    </source>
  </sources>
  <sharedListeners>
    <add type="Microsoft.WindowsAzure.Diagnostics.DiagnosticMonitorTraceListener,
      Microsoft.WindowsAzure.Diagnostics, Version=1.0.0.0, Culture=neutral,
      PublicKeyToken=31bf3856ad364e35"
      name="AzureDiag"/>
  </sharedListeners>
  <trace>
    <listeners>
      <add type="Microsoft.WindowsAzure.Diagnostics.DiagnosticMonitorTraceListener,
        Microsoft.WindowsAzure.Diagnostics, Version=1.0.0.0, Culture=neutral,
        PublicKeyToken=31bf3856ad364e35"
        name="AzureDiagnostics">
        <filter type="" />
      </add>
    </listeners>
  </trace>
</system.diagnostics>
```

*Значения атрибутов **type** показаны с разбивкой на несколько строк.  
Файл конфигурации не должен содержать никаких разрывов строк.*

## Установка и физическое развертывание

В этом подразделе рассматриваются положения, которые следует учитывать при развертывании приложения Tailspin Surveys.

### Сертификаты и развертывание Tailspin Surveys

При развертывании приложения Tailspin Surveys необходимо также развернуть несколько сертификатов. В этом разделе описывается роль сертификатов, места их развертывания и способы получения или создания подходящих сертификатов. Этот раздел посвящен сертификатам, используемым непосредственно приложением Tailspin Surveys и программным блоком автомасштабирования. Дополнительные сведения о сертификатах, используемых имитирующими издателями, которые работают с удостоверениями на основе заявок, можно найти в руководстве «Разработка облачных приложений» в библиотеке MSDN.

При развертывании приложения Tailspin Surveys необходимо развернуть два сертификата. Один из них позволяет Tailspin Surveys использовать конечную точку HTTPS, а другой необходим программному блоку автомасштабирования для вызова API управления службой Windows Azure в размещенной службе Tailspin Surveys. С помощью этих вызовов блок API производит сбор данных приложения Tailspin Surveys и отправляет запросы на масштабирование.

### Развертывание сертификата службы для включения SSL

Средство проверки зависимостей, используемое для установки решения Tailspin Surveys на локальном компьютере разработчика, содержит образец сертификата **localhost**, который может быть использован для включения протокола HTTPS при развертывании приложения Surveys в Windows Azure. Веб-сайт клиента и веб-сайт управления Tailspin Surveys используют конечные точки HTTPS. Следующий фрагмент файла определения службы (CSDEF) для роли **Tailspin.Web** содержит определения сертификата и конечной точки.

```
XML
<WebRole name="Tailspin.Web" ...>
  ...
  <Certificates>
    <Certificate name="localhost_ssl"
      storeLocation="LocalMachine"
      storeName="My" />
  </Certificates>
  <Endpoints>
    <InputEndpoint name="HttpsIn" protocol="https" port="443"
      certificate="localhost_ssl" />
  </Endpoints>
  ...
</WebRole>
```



Файл конфигурации служб определяет используемый сертификат по его отпечатку.

*Сертификат localhost, входящий в состав решения Tailspin Surveys, предназначен только для демонстрационных целей и не должен использоваться в рабочей среде.*

Необходимо передать сертификат службы, который планируется использовать для защиты конечных точек HTTPS, в хранилище сертификатов на портале Windows Azure, а также обеспечить соответствие отпечатка переданного сертификата отпечатку в файле конфигурации службы (CSCFG).

Дополнительные сведения о получении сертификата SSL см. в разделе «Как получить сертификат SSL».

Дополнительные сведения о настройке конечных точек HTTPS в веб-ролях Windows Azure см. в разделе «Как настроить сертификат SSL на конечную точку HTTPS».

## Развертывание сертификата управления для включения операций масштабирования

В приложении Tailspin Surveys программный блок автомасштабирования размещается в отдельной рабочей роле из главного приложения Surveys. Программный блок автомасштабирования использует API управления службой Windows Azure для выполнения действий масштабирования для ролей Tailspin Surveys, и защита этого API выполняется с помощью сертификата управления. В этом разделе описано, как в Tailspin создавали и развертывали этот сертификат управления.

В Tailspin в качестве сертификата управления использовали стандартный сертификат X.509 v3 с длиной ключа 2048 разрядов. Чтобы создать этот самозаверяющий сертификат и установить его в локальном хранилище сертификатов, в Tailspin выполнили следующую команду в окне командной строки Visual Studio.

```
makecert -r -pe -n "CN= Tailspin Management Certificate" -b 05/10/2010 -e 12/22/2012 -ss my -sr localmachine -sky exchange -sp "Microsoft RSA SChannel Cryptographic Provider" -sy 12
```

Затем Tailspin сохраняет открытый ключ в папке **Management Certificates** подписки Windows Azure, где размещено приложение Tailspin Surveys, а закрытый ключ — в папке **Service Certificates** размещенной службы, где содержится программный блок автомасштабирования. Это позволяет программному блоку автомасштабирования обеспечивать безопасность вызовов API управления службой Windows Azure, с которыми он обращается к подписке, в которой размещается приложение Tailspin Surveys.

Дополнительные сведения о сертификатах управления и службах Windows Azure см. в разделе «Управление сертификатами в Windows Azure».



Можно использовать оснастку «Сертификаты» в консоли управления (MMC) для экспорта файла, содержащего открытый ключ (CER), и файла, содержащего закрытый ключ (PFX).

## Развертывание Tailspin Surveys в нескольких географических районах

Версия образца приложения Tailspin Surveys рассчитана на развертывание в одном центре данных, где программный блок автомасштабирования может масштабировать приложение путем добавления и удаления экземпляров роли. Это первый этап плана Tailspin по распространению автомасштабирования на все места, где в настоящее время развернуто приложение Tailspin Surveys, в частности это следующие центры обработки данных: Центральный Север США, Западная Европа и Юго-Восточная Азия. Tailspin стремится получить возможность управлять автоматическим масштабированием во всех центрах обработки данных из единого, централизованного управляющего приложения.

На рис. 2 показана текущая архитектура примера решения, где Tailspin использует программный блок автомасштабирования для управления приложением Surveys в едином центре данных.



Рис. 2  
Приложение Tailspin Surveys, развернутое в едином центре обработки данных.

Tails핀 может использовать аналогичную архитектуру и в других центрах обработки данных, но это будет означать, что каждый центр обработки данных имеет собственный веб-сайт управления. Tails핀 необходимо использовать единый веб-сайт управления, чтобы получить консолидированное представление о полной инфраструктуре автоматического масштабирования приложения Tails핀 Surveys.

Tails핀 рассматривает две альтернативные архитектуры для инфраструктуры автомасштабирования. На рис. 3 наглядно представлена первая альтернатива, где программный блок автомасштабирования и веб-приложение управления размещены в центре обработки данных США.

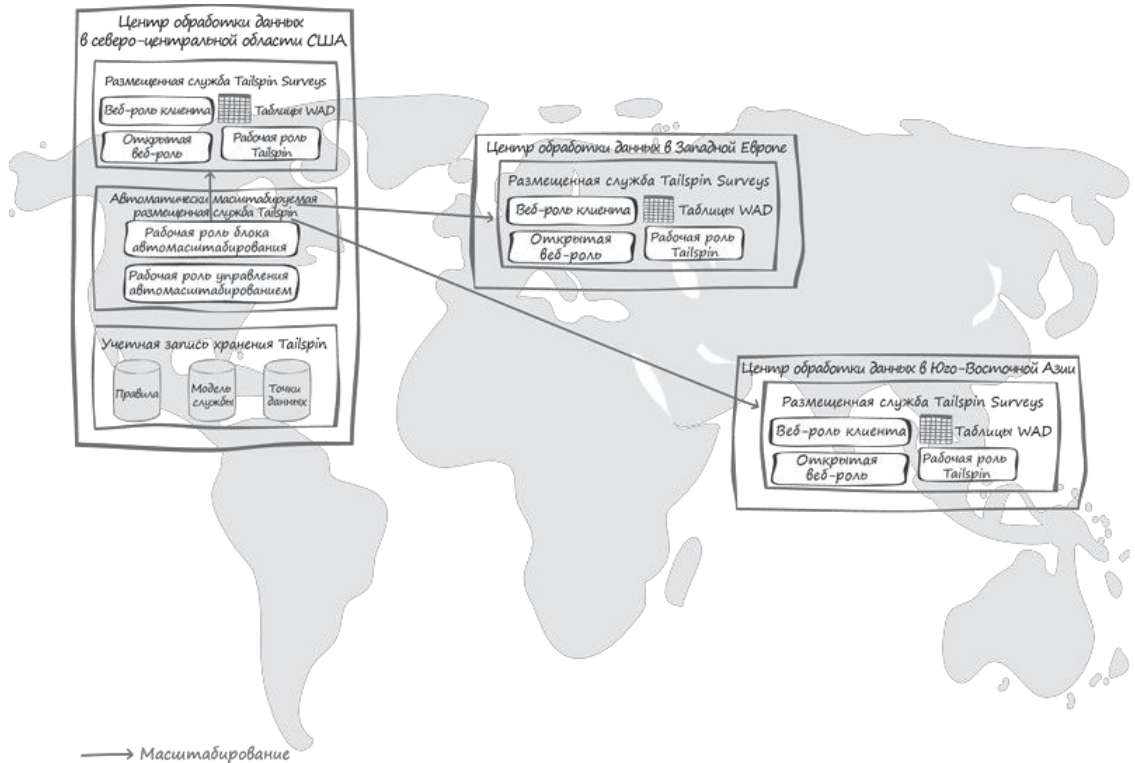


Рис. 3  
Вариант 1. Централизованное развертывание программного блока автомасштабирования

На рис. 4 наглядно показана вторая альтернатива, в которой программный блок автомасштабирования развертывается в каждом из центров обработки данных, а веб-приложение управления по-прежнему размещается в центре обработки данных США.

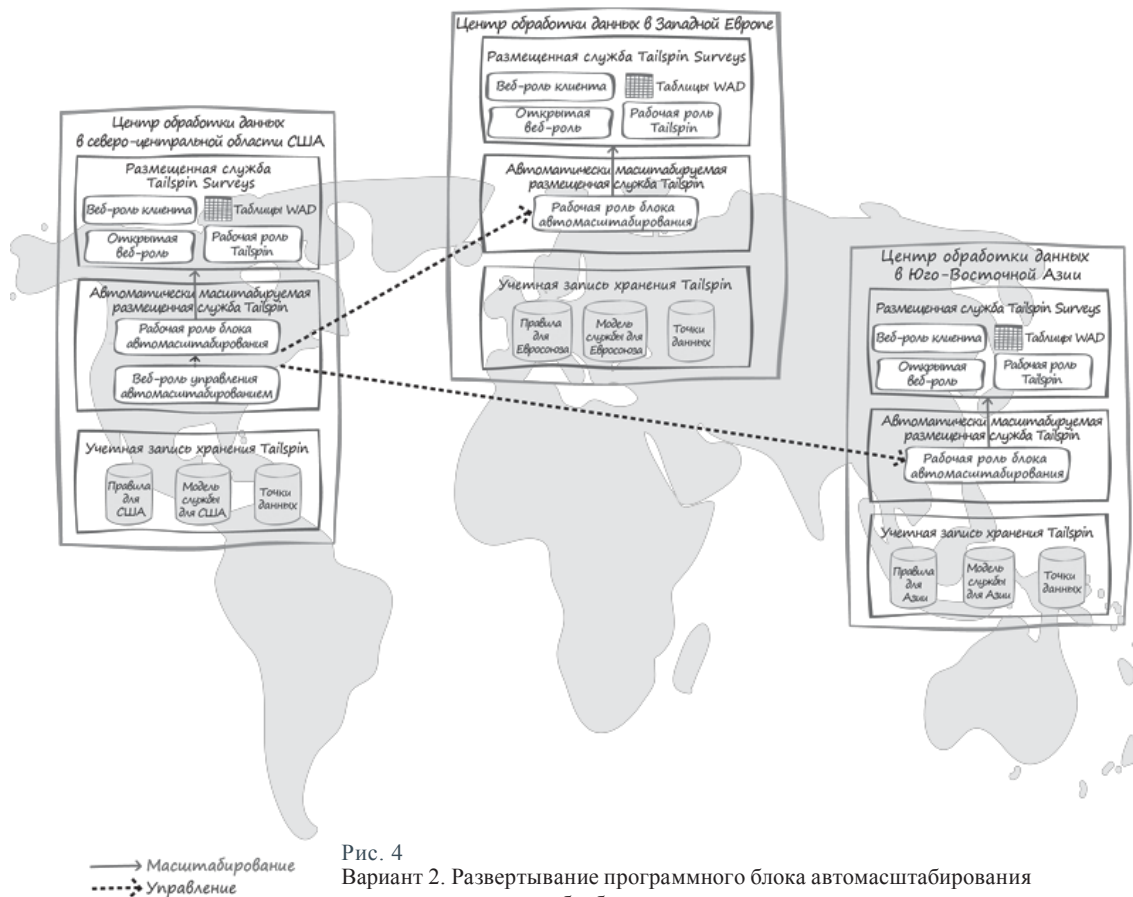


Рис. 4  
 Вариант 2. Развертывание программного блока автоматического масштабирования в каждом из центров обработки данных.

Обе альтернативы позволяют достичь поставленной Tailspin цели управления инфраструктурой автоматического масштабирования из центрального приложения управления, но существует ряд преимуществ и недостатков, которые необходимо рассмотреть. Некоторые из них кратко изложены ниже.

### Затраты на передачу данных

Хотя обе альтернативы требуют передачи данных из удаленных центров данных в центр обработки данных США, в варианте 1 все показатели счетчика производительности, которые программный блок собирает из таблиц диагностики Windows Azure, передаются в центр обработки данных США и сохраняются в хранилище точек данных. В варианте 2 все данные счетчиков производительности хранятся в локальном хранилище точек данных. Однако все метрические данные, которые приложение управления использует для отображения диаграмм и отчетов, по-прежнему должны передаваться по сети.

В Tailspin ожидают, что стоимость передачи данных будет ниже

в варианте 2. Вариант 2 также позволяет сократить время, необходимое для передачи данных в хранилище точек данных, и свести к минимуму риск любых переходных состояний сети, влияющих на процесс авто- масштабирования.

### Экземпляры роли

Обе альтернативы требуют наличия лишь одного экземпляра роли для приложения управления. В Tailspin не прогнозируют большой нагрузки на это приложение, поэтому оно может использовать небольшой экземпляр.

В варианте 1 существует один экземпляр рабочей роли, в котором размещен программный блок автомасштабирования, запущенный из центра обработки данных США. По оценке Tailspin, в данном сценарии можно использовать малый или средний экземпляр роли.

В варианте 2 в каждом центре обработки данных существует один экземпляр рабочей роли с размещенным в нем программным блоком автомасштабирования. По оценкам компании Tailspin, каждый центр обработки данных может использовать малый экземпляр роли для рабочей роли.

В варианте 2 используется большее число экземпляров роли, чем в варианте 1.

### Различия конфигурации

Вариант 1 хранит все служебные сведения и правила автомасштабирования в хранилищах центра обработки данных США. Если Tailspin необходимо применять разные правила в отдельных центрах данных, то важно точно следовать соглашениям об именовании ролей в разных центрах данных, а также применяемым в этих ролях правилам и операндам. В варианте 2 каждый центр обработки данных имеет собственное хранилище правил и свои служебные сведения.

В обоих случаях можно использовать различные правила в каждом центре обработки данных, если требования автомасштабирования отличаются. В варианте 1 Tailspin необходимо дополнительно заботиться об управляемости правил с помощью соответствующих соглашений об именовании.

### Различия приложений

Существующее веб-приложение управления будет работать без изменений с вариантом 1, так как он предназначен для работы с единым хранилищем сведений о службе и единым хранилищем правил. Tailspin будет несложно доработать веб-сайт управления для работы с несколькими хранилищами данных и хранилищами правил согласно варианту 2.

Существующие пользовательские операнды не будут работать с вариантом 1, потому что они не предназначены для работы с несколькими экземплярами приложения Tailspin Surveys. В настоящее время не существует способа их настройки для сбора данных от конкретного экземпляра приложения Surveys. Пользовательские операнды будут работать без изменений с вариантом 2, потому что каждый экземпляр программного блока автомасштабирования управляет одним экземпляром Tailspin Surveys.



Аналитика хранения данных Windows Azure может служить для получения более глубокого представления об использовании данных. Дополнительные сведения см. в разделе «Общие сведения об аналитике хранения».



Tailspin планирует использовать только одиночные экземпляры ролей автоматического масштабирования, так как для этих ролей не требуются гарантии соглашения об уровне обслуживания Windows Azure.

В Tailspin приняли решение о реализации варианта 2. В этой модели каждый центр обработки данных является автономным и использует приложение Tailspin Surveys и программный блок автомасштабирования. Это облегчает Tailspin задачу управления различными требованиями масштабирования каждого программного блока и минимизирует объем данных, передаваемых между центрами обработки данных. В Tailspin планируют усовершенствовать веб-приложение управления автомасштабированием для поддержки данного сценария.

## Дополнительные сведения

Дополнительные сведения об автомасштабировании и о том, как работает блок автомасштабирования приложения, см. в главе 4, «Автоматическое масштабирование и Windows Azure», в этом руководстве. Инструкции по установке приложения Tailspin Surveys см. в приложении Б «Руководство по установке Tailspin Surveys».

Дополнительные сведения о сертификатах, используемых имитирующими издателями, которые работают с управлением идентификаторов на основе заявок, см. в руководстве *Разработка облачных приложений, 2-е издание* в библиотеке MSDN:

<http://msdn.microsoft.com/ru-ru/library/ff966499.aspx>

Дополнительные сведения о методе **CloudStorageAccount.SetConfigurationSettingPublisher** см. в разделе **CloudStorageAccount.SetConfigurationSettingPublisher** библиотеки MSDN: <http://msdn.microsoft.com/ru-ru/library/microsoft.windowsazure.cloudstorageaccount.setconfigurationsettingpublisher.aspx>

Дополнительные сведения о получении сертификата SSL см. в разделе «Как получить SSL-сертификат» в библиотеке MSDN: <http://go.microsoft.com/fwlink/?LinkID=234634>

Дополнительные сведения о настройке конечных точек HTTPS в веб-ролях Windows Azure см. в разделе MSDN «Как настроить SSL-сертификат для конечной точки HTTPS»: <http://go.microsoft.com/fwlink/?LinkID=234623>

Дополнительные сведения о сертификатах управления и службах Windows Azure см. в разделе «Управление сертификатами в Windows Azure» на сайте MSDN:

<http://go.microsoft.com/fwlink/?LinkID=234616>

Аналитика хранения данных Windows Azure может служить для получения более глубокого представления об использовании данных. Дополнительные сведения см. в статье MSDN «Обзор аналитики хранения»: <http://go.microsoft.com/fwlink/?LinkID=234635>

Чтобы получить простой доступ к ресурсам в сети, используйте электронную версию библиографии на MSDN:

[http://msdn.microsoft.com/en-us/library/hh749032\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh749032(v=PandP.50).aspx)

## 6 Обработка временного сбоя

### Что такое временные сбои?

Когда облачные приложения используют другие облачные службы, могут возникать ошибки из-за временно действующих условий, таких как работа службы по расписанию, сбои на уровне инфраструктуры или проблем с сетевой связью. Очень часто, если вы повторите операцию некоторое время спустя (может быть, даже через несколько миллисекунд), операция может выполняться успешно. Подобные типы ошибок называются временными сбоями. Временные сбои обычно происходят весьма редко, и в большинстве случаев для успешного завершения операции требуется выполнение всего нескольких повторных попыток.

К сожалению, не существует простого способа отличить временные сбои от постоянных. Оба типа сбоев, скорее всего, приведут к возникновению исключений в приложении. Повторение операции, которая вызывает постоянную ошибку (такую как ошибка «файл не найден»), скорее всего, приведет к получению того же исключения.

К примеру, в технологической платформе SQL Azure™ одним из важных аспектов является обработка клиентских соединений. SQL Azure может использовать регулирование запросов, когда клиент пытается установить соединение с базой данных или выполнить запросы к ней. SQL Azure регулирует количество подключений к базе данных по целому ряду причин, таких как чрезмерное использование ресурсов, длительные транзакции и действия по балансировке нагрузки и отработке отказа. Это может привести к прекращению существующих клиентских сеансов или временной неспособности установить новые соединения, пока действуют условия, описанные выше. SQL Azure может также закрыть подключения к базе данных по целому ряду причин, связанных с возможностью сетевого подключения между клиентом и удаленным центром обработки данных Майкрософт. Сюда относится качество связи по сети, периодические сетевые сбои в инфраструктуре глобальной сети или локальной сети и другие временные технические причины.

*Нет никакого внутреннего способа отличить временный сбой от постоянного, если разработчик службы явно не изолировал временные сбои в заданном подмножестве типов исключений или кодов ошибок.*



Если клиент превышает цели масштабируемости, регулирование может производиться с помощью хранилища технологической платформы Windows Azure™. Дополнительные сведения см. в разделе «Абстракции хранилищ Windows Azure и цели их масштабируемости».



Для определения того, какие исключения являются результатом временных сбоев службы, требуются детализированный набор знаний и опыт использования службы. Блок инкапсулирует такого рода набор знаний и опыта.



Этот вид логики повтора известен также как логика «условного повтора».

## Что такое программный блок обработки временных сбоев?

Программный блок обработки временных сбоев делает приложение более надежным, предоставляя механизмы для обработки временных сбоев. При этом используются два метода.

Во-первых, блок содержит логику для определения временных сбоев для целого ряда облачных служб в виде стратегий обнаружения сбоев. Эти стратегии обнаружения содержат встроенный набор знаний, который определяет, было ли вызвано конкретное исключение временной ошибкой.

Блок включает стратегии обнаружения для следующих служб:

- SQL Azure;
- шина обслуживания Windows Azure;
- служба хранилища Windows Azure;
- Служба кэширования Windows Azure.

Во-вторых, программный блок позволяет определить стратегии повторных попыток, чтобы в приложении соблюдался согласованный подход при обработке временных сбоев. Стратегия повторов, которая будет использоваться, зависит от ряда факторов, например, насколько интенсивно приложение должно выполнять повторные попытки и как обычно ведет себя служба при выполнении повторов. Некоторые службы могут проводить дополнительное регулирование и даже блокировать клиентские приложения, которые выполняют повторные попытки слишком интенсивно. Стратегия повторов определяет, какое количество повторов следует выполнить, перед тем как отнести сбой к категории постоянных, и какими должны быть интервалы между повторными попытками.

Встроенные стратегии повтора позволяют указать, что повторы должны происходить в фиксированные интервалы, в интервалы, которые каждый раз одинаково увеличиваются, и в интервалы, которые увеличиваются экспоненциально, но с некоторыми случайными вариациями. В следующей таблице приводятся примеры всех трех стратегий.

Стратегия повтора	Пример (интервал в секундах между повторными попытками)
Фиксированный интервал	2, 2, 2, 2, 2, 2
Добавочные интервалы	2, 4, 6, 8, 10, 12
Интервалы случайной экспоненциальной задержки	2, 3,755, 9,176, 14,306, 31,895

*Все стратегии повтора указывают максимальное количество попыток, после чего исходное исключение может быть передано в приложение.*

Во многих случаях следует использовать стратегию случайных экспоненциальных интервалов, чтобы аккуратно снизить нагрузку на службу. Это особенно верно, если служба регулирует количество клиентских запросов.



Можно определить собственные пользовательские стратегии обнаружения сбоев, если встроенные стратегии не соответствуют вашим требованиям. Программный блок также позволяет определить собственные пользовательские стратегии повтора, которые задают дополнительные шаблоны для интервалов повтора.

На рис. 1 показано, как взаимодействуют ключевые элементы программного блока обработки временных сбоев, что позволяет добавить в приложение логику выполнения повторных попыток.

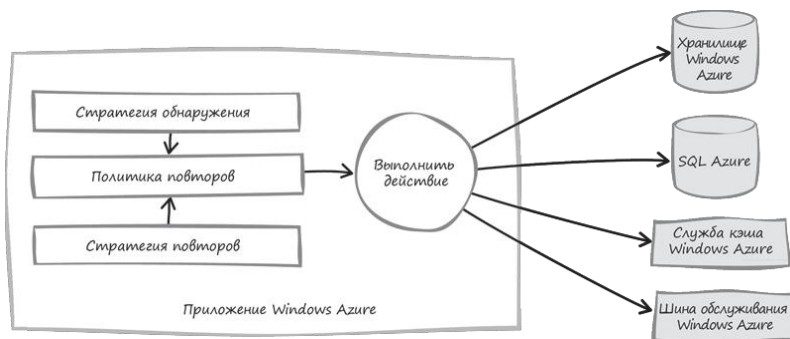


Рис. 1  
Программный блок обработки временных сбоев

Правила повторов сочетают стратегию обнаружения ошибок со стратегией выполнения повторов. Можно использовать одну из перегруженных версий метода **ExecuteAction** в качестве оболочки для вызова, который приложение делает к одной из служб.

## Историческая справка

Программный блок обработки временных сбоев является результатом совместной работы группы Майкрософт по архитектуре и рекомендациям и консультационной группы проекта Windows Azure. Он основан на стратегиях первоначального обнаружения и повтора попыток и поддержки доступа к данным из «Платформы обработки временных сбоев для SQL Azure, хранилища Windows Azure, шины обслуживания и службы кэширования». Новый программный блок теперь включает расширенную поддержку конфигурации, улучшенную поддержку для упаковки асинхронных вызовов, обеспечивает интеграцию стратегий повтора блока с механизмом повторов хранилища Windows Azure и работает с контейнером внедрения зависимостей библиотеки Enterprise Library. Новый программный блок обработки временных сбоев заменяет прежнюю платформу и в настоящее время является рекомендуемым средством для обработки временных сбоев в приложениях Windows Azure.



Приложения с высокой пропускной способностью обычно используют стратегию экспоненциальной задержки. Однако для приложений, представляемых пользователям, например веб-сайтов, может быть целесообразно рассмотреть стратегию линейной задержки для обеспечения быстрой реакции пользователя на интерфейс.



Необходимо выбрать соответствующую стратегию обнаружения для службы, метод которой вызывается из приложения Windows Azure.

## Использование программного блока обработки временных сбоев

В настоящем разделе в общих чертах описывается использование программного блока обработки временных сбоев. Он состоит из следующих основных подразделов. Порядок этих разделов отражает тот порядок, в котором обычно выполняются соответствующие задачи.

- **Добавление программного блока обработки временных сбоев к проекту Visual Studio.** В этом разделе описывается, как подготовить свою среду разработки Microsoft Visual Studio® для использования блока.
- **Определение стратегии повтора.** В этом разделе описываются способы, которыми можно определить стратегию повтора для приложения.
- **Определение правил повторов.** В этом разделе описываются способы, которыми можно определить правила повторов.
- **Выполнение операции с правилами повторов.** В этом разделе описывается способ выполнения действий с правилами повторов с обработкой любых временных сбоев.

*Правила повторов сочетают стратегию обнаружения ошибок со стратегией выполнения повторных попыток. Правила повторов используются при выполнении операций, на которые могут повлиять временные сбои.*

Другие примеры использования программного блока обработки временных сбоев в приложении Windows Azure см. в разделе 7, «Повышение надежности приложения Tailspin Surveys».

Подробные сведения о настройке программного блока обработки временных сбоев и написании кода, который использует данный блок, см. в разделе «Программный блок обработки временных сбоев» на сайте MSDN®.



NuGet очень облегчает настройку проекта со всеми необходимыми компонентами для использования Transient Fault Handling Application Block.

### Добавление программного блока обработки временных сбоев к проекту Visual Studio.

Чтобы получить возможность написания какого-либо кода, в котором используется программный блок обработки временных сбоев, нужно настроить проект Visual Studio с учетом всех необходимых сборок, ссылок и других ресурсов, которые могут потребоваться. Сведения о том, как использовать NuGet при подготовке проекта Visual Studio для работы с программным блоком обработки временных сбоев, см. в разделе «Добавление программного блока обработки временных сбоев к проекту Visual Studio» на сайте MSDN.

## Создание экземпляров объектов программного блока обработки временных сбоев

Для создания экземпляров объектов данного программного блока используются два основных подхода. Первый подход — экземпляры явно создаются в коде, как показано в следующем фрагменте кода:

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

var retryPolicy =
    new RetryPolicy<StorageTransientErrorDetectionStrategy>
        (retryStrategy);
```

*При создании экземпляра объекта **RetryPolicy** с помощью метода **new** нельзя использовать стратегии по умолчанию, определенные в конфигурации.*

Второй подход — используется инфраструктура библиотеки Enterprise Library для создания и управления объектами, как показано в следующем фрагменте кода:

```
C#
var retryManager = EnterpriseLibraryContainer.Current
    .GetInstance<RetryManager>();

var retryPolicy = retryManager.GetRetryPolicy
    <StorageTransientErrorDetectionStrategy>("Incremental Retry
        Strategy");
```

Существует еще один метод, который предоставляется для обеспечения обратной совместимости с «Платформой обработки временных сбоев», использующий класс **RetryPolicyFactory**:

```
C#
var retryPolicy = RetryPolicyFactory.GetRetryPolicy
    <StorageTransientErrorDetectionStrategy>("Incremental Retry
        Strategy");
```

## Определение стратегии повтора

Следует учитывать три аспекта определения стратегий повторов для приложения: какую стратегию повтора использовать, место определения стратегии и необходимость использования стратегии повторов по умолчанию.

В большинстве случаев следует использовать одну из встроенных стратегий повторов: с фиксированным интервалом, с добавочными интервалами или со случайными экспоненциальными интервалами. Каждую из этих стратегий можно настроить с помощью набора пользовательских параметров в соответствии с потребностями приложе-



Если используется хранилище Windows Azure и уже используется механизм правил повтора в пространстве имен **Microsoft.WindowsAzure.StorageClient**, то, чтобы воспользоваться преимуществами расширенных функциональных возможностей повтора, предоставляемых блоком, можно использовать стратегии повтора блока и настроить API клиента хранилища Windows Azure.

ния. Параметры указывают, когда стратегия должна остановить выполнение повторных попыток и какими должны быть интервалы между повторными попытками. Выбор стратегии повторов во многом зависит от конкретных требований приложения. Дополнительные сведения о параметрах для каждой стратегии см. в разделе «*Схема-источник для программного блока обработки временных сбоев*» на сайте MSDN.

Можно определить собственную стратегию повторов. Дополнительные сведения см. в разделе «*Реализация пользовательской стратегии повторов*» на сайте MSDN.

Правила повторов можно определять в коде или в файле конфигурации приложения. Определение правил повтора в коде является наиболее подходящим способом для небольших приложений с ограниченным числом вызовов, которые требуют использования логики выполнения повторных попыток. Определение правил повтора в конфигурации является более полезным способом, когда имеется большое количество операций, которые требуют логику выполнения повторных попыток, потому что при таком подходе легче поддерживать и изменять правила.

Дополнительные сведения о том, как определить стратегии повторных попыток в коде, см. в разделе «*Определение стратегий повторов в коде*» на сайте MSDN.

Дополнительные сведения о том, как определить стратегии повторных попыток в файле конфигурации, см. в разделе «*Определение стратегий повторов в конфигурации*» на сайте MSDN.

Если стратегии повтора определяются в файле конфигурации приложения, то также можно определить стратегии повтора по умолчанию. Блок позволяет указывать стратегии повтора по умолчанию на двух уровнях. Стратегию повтора по умолчанию можно указать для операции каждой из следующих категорий: операции соединения к SQL, команды SQL, операции шины обслуживания Windows Azure, операции кэширования Windows Azure и хранилища Windows Azure. Также можно задать глобальную стратегию повторов по умолчанию.

## Определение правил повторов

Правила повторов сочетают стратегию повторов и стратегию обнаружения, которая используется при выполнении операции, которая может быть затронута временными сбоями. Класс `RetryManager` содержит методы, которые позволяют создавать правила повтора путем явного определения стратегии повторов и обнаружения или путем явного указания стратегии повторов по умолчанию, определенной в файле конфигурации.

Дополнительные сведения об использовании правил повтора см. в разделе «*Ключевые сценарии*» на сайте MSDN.

Дополнительные сведения о делегате **`RetryPolicy`** в пространстве имен **`Microsoft.WindowsAzure.StorageClient`** см. в блоге «*Общие сведения о правилах повтора в библиотеке Windows Azure Storage Client Library*».

## Выполнение операции с правилами повторов

Класс **RetryPolicy** содержит несколько перегруженных версий метода **ExecuteAction**. Метод **ExecuteAction** используется в приложении для упаковки тех вызовов, которые могут быть затронуты временными сбоями. Различные перегруженные версии позволяют создать оболочку для следующих типов вызовов службы.

- Синхронные вызовы, возвращающие **void**.
- Синхронные вызовы, возвращающие значение.
- Асинхронные вызовы, возвращающие **void**.
- Асинхронные вызовы, возвращающие значение.

Метод **ExecuteAction** автоматически применяет настроенную стратегию повторов и обнаружения при вызове указанного действия. Если во время выполнения действия временный сбой не происходит, приложение продолжает работать как обычно, как будто бы между кодом и действием не добавлялся еще один уровень логики. Если временная ошибка проявляется, блок начнет процедуру восстановления, пытаясь вызвать указанное действие несколько раз, как это определено в стратегии повторных попыток. После того как повторная попытка успешно завершится, приложение продолжит работать в нормальном режиме. Если блок не добьется положительного результата за отведенное в стратегии повторов количество попыток, то блок отправит исключение в приложение. Приложение в любом случае должно правильно обработать это исключение.

*Программный блок обработки временных сбоев не заменяет надлежащую обработку исключений. Приложение должно в любом случае обрабатывать все исключения, порождаемые используемыми службами.*

Кроме того, блок включает классы, которые автоматически упаковывают многие общие операции SQL Azure для правил повторов. Использование данных классов сокращает написание кода.

Дополнительные сведения о выполнении операции с правилами повторных попыток см. в разделе «*Ключевые сценарии*» на сайте MSDN.



Для получения в приложении уведомлений об операциях повтора, которые выполняет блок, можно использовать событие **Повтор**.

При работе с SQL Azure блок включает классы, обеспечивающие непосредственную поддержку для SQL Azure, например класс **ReliableSqlConnection**. Эти классы позволяют сократить объем создаваемого разработчиком кода.

## Когда следует использовать программный блок обработки временных сбоев?

В настоящем разделе описаны два сценария, в которых необходимо рассмотреть возможность использования блока обработки временных сбоев в приложении Windows Azure.

### Вы используете службу Windows Azure

Если приложение использует любую службу Windows Azure, поддерживаемую программным блоком обработки временных сбоев (SQL Azure, хранилище Windows Azure, службу кэширования Windows Azure или шину обслуживания Windows Azure), то использование блока сделает приложение более надежным. Любое приложение Windows Azure, использующее эти службы, может иногда испытывать проблемы из-за временных сбоев этих служб. Хотя в приложение можно добавить собственную логику обнаружения подобных сбоев, встроенные стратегии обнаружения сбоев в программном блоке будут обрабатывать более широкий диапазон временных сбоев. Также быстрее и проще использовать программный блок вместо разработки собственного решения.

Клиентский API-интерфейс хранилища Windows Azure уже включает поддержку для пользовательских правил повторов. Стратегии повторов из программного блока можно использовать с клиентским API-интерфейсом хранилища Windows Azure. С помощью стратегий повторов в программном блоке обработки временных сбоев и механизма повторов Windows Azure можно использовать встроенные и пользовательские стратегии повторов и использовать стратегии, определенные в конфигурации приложения.

*Использование правил повторов программного блока обработки временных сбоев вместо встроенных правил повторов Windows Azure позволит использовать преимущества программного блока, связанные с настраиваемой и расширяемой логикой выполнения повторных попыток.*

Дополнительные сведения о повторах в хранилище Windows Azure см. в разделе «Обзор правил повтора в библиотеке Windows Azure Storage Client Library».

### Вы используете пользовательскую службу

Если ваше приложение использует пользовательскую службу, оно по-прежнему может выиграть от использования программного блока обработки временных сбоев. Для службы можно создать пользовательскую стратегию обнаружения ошибок, в которой учитывается (инкапсулируется) имеющийся набор знаний о тех исключениях, которые могут возникнуть при вызове операций службы. Программный блок обработки временных сбоев предлагает средства для определения правил повторов и упаковки вызовов методов для логики повторных попыток.

## Дополнительные сведения

Другие примеры использования программного блока обработки временных сбоев в приложении Windows Azure см. в разделе 7, «Повышение надежности приложения Tailspin Surveys».

Подробные сведения о настройке программного блока обработки временных сбоев и написании кода, который использует данный блок, см. в разделе «Программный блок обработки временных сбоев» на сайте MSDN:

[http://msdn.microsoft.com/en-us/library/hh680934\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680934(v=PandP.50).aspx)

Дополнительные сведения о регулировании в Windows Azure см. в разделе «Абстракции хранилища Windows Azure и цели их масштабируемости» на сайте MSDN:

<http://go.microsoft.com/fwlink/?LinkID=234633>

Сведения о том, как использовать NuGet при подготовке проекта Visual Studio для работы с программным блоком обработки временных сбоев, см. в разделе «Добавление программного блока обработки временных сбоев к проекту Visual Studio» на сайте MSDN:

[http://msdn.microsoft.com/en-us/library/hh680891\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680891(v=PandP.50).aspx)

Существует еще один метод, который предоставляется для обеспечения обратной совместимости с «Платформой обработки временных сбоев», использующий класс **RetryPolicyFactory**:

<http://windowsazurecat.com/2011/02/transient-fault-handling-framework/>

Дополнительные сведения о параметрах для каждой стратегии см. в разделе «Схема-источник для программного блока обработки временных сбоев» на сайте MSDN.

[http://msdn.microsoft.com/en-us/library/hh680941\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680941(v=PandP.50).aspx)

Можно определить собственную стратегию повторов. Дополнительные сведения см. в разделе «Реализация пользовательской стратегии повторов» на сайте MSDN.

[http://msdn.microsoft.com/en-us/library/hh680943\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680943(v=PandP.50).aspx)

Дополнительные сведения о том, как определить стратегии повторных попыток в коде, см. в разделе «Определение стратегий повторов в коде» на сайте MSDN.

[http://msdn.microsoft.com/en-us/library/hh680927\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680927(v=PandP.50).aspx)

Дополнительные сведения о том, как определить стратегии повторных попыток в файле конфигурации, см. в разделе «Определение стратегий повторов в конфигурации» на сайте MSDN:

[http://msdn.microsoft.com/en-us/library/hh680900\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680900(v=PandP.50).aspx)

Дополнительные сведения об использовании правил повтора см. в разделе «Ключевые сценарии» на сайте MSDN:

[http://msdn.microsoft.com/en-us/library/hh680948\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680948(v=PandP.50).aspx)

Дополнительные сведения о делегате **RetryPolicy** в пространстве имен **Microsoft.WindowsAzure.StorageClient** см. в блоге «Общие сведения о правилах повтора в библиотеке Windows Azure Storage Client Library»:

<http://go.microsoft.com/fwlink/?LinkID=234630>

Дополнительные сведения о повторах в хранилище Windows Azure см. в разделе «Обзор правил повтора в библиотеке Windows Azure Storage Client Library»:

<http://go.microsoft.com/fwlink/?LinkID=234630>

Модуль Transient Fault Handling Application Block является продуктом совместной работы центра разработчиков Microsoft patterns & practices (<http://msdn.microsoft.com/practices>) и консультационной группы платформы Windows Azure (<http://windowsazurecat.com/index.php>).

Он основан на стратегиях первоначального обнаружения и повтора и на поддержке доступа к данным платформы, которые описаны в разделе «Transient Fault Handling Framework для SQL Azure, хранилища Windows Azure, шины обслуживания и кэша» в MSDN:

<http://windowsazurecat.com/2011/02/transient-fault-handling-framework/>

Чтобы получить простой доступ к ресурсам в сети, используйте электронную версию библиографии на MSDN:

[http://msdn.microsoft.com/en-us/library/hh749032\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh749032(v=PandP.50).aspx)



## 7 Увеличение надежности приложения Tailspin Surveys

В этой главе описываются изменения, сделанные компанией Tailspin при добавлении модуля Transient Fault Handling Application Block в приложение Surveys для повышения устойчивости приложения к временным сбоям в среде технологической платформы Windows Azure™.

### Предпосылки

Для удовлетворения потребностей своих крупных клиентов компания согласилась повысить уровни обслуживания в их соглашениях об уровне обслуживания, особенно в отношении надежности и доступности приложения Surveys. У таких клиентов также имеются более жесткие требования к производительности, например максимальное время, которое требуется для экспорта данных в технологическую платформу SQL Azure™. Для удовлетворения этих новых требований соглашений об уровне обслуживания в компании Tailspin вновь тщательно протестировали приложение Surveys, чтобы понять, как можно повысить устойчивость приложения.

В Tailspin обнаружили, что неустойчивые условия иногда могут приводить к ошибкам, когда приложение Surveys делает вызовы к SQL Azure или хранилищу Windows Azure. Вызов выполняется успешно, если через короткое время, когда неустойчивые условия исчезнут, выполняется повтор операции.

Приложение Tailspin Surveys использует хранилище Windows Azure и SQL Azure. Определения приложения Surveys хранятся в таблицах Windows Azure, данные конфигурации клиента хранятся в хранилище больших двоичных объектов Windows Azure, ответы опроса также хранятся в хранилище Windows Azure. Приложение Surveys позволяет также клиентам экспортировать данные опросов в SQL Azure, где клиенты могут выполнить свой собственный подробный анализ результатов. Для некоторых клиентов экземпляр SQL Azure расположен в другом центре обработки данных, где размещены опросы клиентов.

Операторы заметили случайные ошибки в файлах журнала приложения Surveys, которые относятся к ошибкам хранилища. Эти ошибки не связаны с конкретными областями функций, но, кажется, происходят случайно. Было небольшое количество сообщений, что

*Приложения, которые работают на платформе Windows Azure, должны правильно и эффективно обрабатывать временные сбои, чтобы уменьшить потенциальное воздействие временных условий на устойчивость приложения.*



Если Tailspin собирается добиться успеха в привлечении крупных клиентов, для нее жизненно важно повышение надежности и устойчивости приложения Surveys.

пользователи, создающие новые опросы, теряли определение опроса при нажатии кнопки «Сохранить» в пользовательском интерфейсе.

Были также случаи сбоя долго выполняющихся заданий, экспортирующих данные в SQL Azure. Поскольку нет метода возобновления частично выполненных задач экспорта, Tailspin должна перезапустить процесс экспорта с самого начала. Приходилось повторно запускать задания, которые не были завершены успешно, но это означает, что компании не удалось выполнить свои соглашения об уровне обслуживания с клиентом. Там, где экспорт выполняется в центр обработки данных, отличный от центра обработки, в котором размещены определения опросов, компания Tailspin несла дополнительные расходы, связанные с пропускной способностью, из-за необходимости повторного запуска задания экспорта.

## Цели и требования

В Tailspin хотят реализовать логику автоматического повтора для всех своих операций с хранилищем Windows Azure для повышения общей надежности приложения. Необходимо свести к минимуму риск потери данных опроса и получения неточных статистических данных. Tailspin стремится обеспечить, чтобы приложение было как можно более устойчивым и могло восстанавливаться после любых временных ошибок без вмешательства оператора. Компания также хочет уменьшить вероятность возникновения ошибок у клиентов, когда они создают новые определения опросов.

В Tailspin хотят повысить надежность задач экспорта, отправляющих данные в SQL Azure, чтобы обеспечить соблюдение соглашений об уровне обслуживания с клиентами.

Tailspin хочет иметь возможность настраивать правила повтора (например, регулируя задержку) в различных сценариях. Некоторые задачи являются больше критичными по времени, например сохранение определения нового опроса, где пользователь ожидает подтверждения сохранения определения; другие задачи меньше критичны по времени, например вычисления статистических данных, которые по существу не могут предоставлять результаты в режиме реального времени.

## Общие сведения о решении Transient Fault

### Handling Application Block

Transient Fault Handling Application Block позволяет добавить логику повтора в облачное приложение. Этот блок можно использовать для применения правил повтора для всех вызовов, в которых могут возникать ошибки из-за временных условий.

Transient Fault Handling Application Block включает в себя стратегии определения, которые могут определять исключения, вызываемые временными сбоями. Программа Tailspin Surveys использует хранилище Windows Azure и SQL Azure; Transient Fault Handling Application Block включает стратегии определения для этих сервисов.

Transient Fault Handling Application Block использует стратегии повтора для определения шаблонов повтора: числа попыток и интервала между ними. Эти стратегии могут быть определены в коде или в конфигурации. Компания Tailspin планирует использовать стратегии повтора, определенные в конфигурации, чтобы было легче настраивать стратегии повтора, используемые приложением Surveys.

## Процесс реализации

Этот раздел содержит некоторые сведения о том, как Tailspin использует Transient Fault Handling Application Block и как эта компания модифицировала приложение Surveys для использования с блоком. Если вам не интересны дополнительные сведения, можно перейти к следующему разделу.

Во время чтения этого раздела может быть полезным иметь открытым решение Tailspin в среде Visual Studio, чтобы иметь возможность непосредственно обращаться к коду.

Инструкции по установке приложения Tailspin Surveys см. в приложении Б, «Руководство по установке Tailspin Surveys».

Tailspin использует Transient Fault Handling Application Block в приложении Surveys везде, где используется API хранилища Windows Azure, или при вызове операции в базе данных SQL Azure. Например, компания использует этот блок в коде, в котором производится обращение к хранилищам правил и служебной информации, в классах-оболочках для типов хранилищ Windows Azure и в классе **SurveySqlStore**. Все эти классы расположены в проекте **Tailspin.Shared**.

Файл конфигурации для каждой рабочей роли и веб-роли в приложении Surveys включает стратегии повтора, показанные в следующем фрагменте кода.

```
XML
<RetryPolicyConfiguration
  defaultRetryStrategy="Fixed Interval Retry Strategy"
  defaultAzureStorageRetryStrategy="Fixed Interval Retry
    Strategy"
  defaultSqlCommandRetryStrategy="Backoff Retry Strategy">
  <incremental name="Incremental Retry Strategy"
    retryIncrement="00:00:01"
    initialInterval="00:00:01"
    maxRetryCount="10" />
  <fixedInterval name="Fixed Interval Retry Strategy"
    retryInterval="00:00:05"
    maxRetryCount="6"
    firstFastRetry="true" />
  <exponentialBackoff name="Backoff Retry Strategy"
    minBackoff="00:00:05"
    maxBackoff="00:00:45"
    deltaBackoff="00:00:04"
```



Программа Tailspin Surveys использует ограниченное количество стратегий повтора из ограниченного количества мест в коде. В этом примере показано несколько стратегий повтора по умолчанию, предназначенных для облегчения поддержки кода.

```
maxRetryCount="10" />  
</RetryPolicyConfiguration>
```

Компания Tailspin использует средство настройки Enterprise Library для изменения этих параметров.

Tailspin использует класс **RetryManager** для загрузки стратегий повтора из файла конфигурации и для создания экземпляра правила повтора. В следующем фрагменте кода класса **RuleSetModelStore** показан пример того, как Tailspin создает новое правило повтора, которая использует стратегию обнаружения хранилища Windows Azure и «Дополнительную стратегию повтора» из конфигурации.

```
C#  
public RuleSetModelStore(  
    RuleSetModelToXmlElementConverter  
        ruleSetModelToXmlElementConverter,  
    [Dependency("RuleSetModel")] IConfigurationFileAccess  
        fileAccess,  
    RetryManager retryManager)  
{  
    this.retryPolicy = retryManager.GetRetryPolicy  
        <StorageTransientErrorDetectionStrategy>  
            (AzureConstants.FaultHandlingPolicies.Incremental);  
  
    ...  
}
```

*Необходимо соблюдать осторожность при попытке загрузить стратегии повтора из файла конфигурации web.config, используя классы **RetryPolicyFactory** или **Retry Manager** в событии веб-роли **OnStart**. Дополнительные сведения см. в разделе «Задание стратегий повтора в конфигурации» в MSDN.*

Если Transient Fault Handling Application Block используется с хранилищем Windows Azure, необходимо быть осторожным и не использовать встроенные правила повтора в интерфейсах API хранилища Windows Azure. В следующем фрагменте кода из класса **AzureQueue** в проекте **Tailspin.Shared** показано, как Tailspin отключает встроенные правила повтора.

```
C#  
var client = this.account.CreateCloudQueueClient();  
client.RetryPolicy = RetryPolicies.NoRetry();
```

В следующем фрагменте кода из метода **GetFileContent** в классе **RuleModelStore** показано, как Tailspin упаковывает вызов, обращающийся к хранилищу Windows Azure, на которое могут влиять условия временного сбоя с правилом повтора.

```

C#
try
{
    return this.retryPolicy.ExecuteAction(
        () => this.fileAccess.GetFileContent());
}
catch (ConfigurationFileAccessException)
{
    return null;
}

```

Tailspin использует тот же подход, когда приложение Surveys сохраняет данные в SQL Azure, как показано в следующем образце кода из класса **SurveySqlStore**. В этом примере также показано, как загрузить правило по умолчанию из конфигурации.

```

C#
public SurveySqlStore()
{
    this.retryPolicy =
        RetryPolicyFactory.GetDefaultSqlCommandRetryPolicy();
}
public void SaveSurvey(string connectionString,
    SurveyData surveyData)
{
    using (var dataContext =
        new SurveySqlDataContext(connectionString))
    {
        dataContext.SurveyDatas.InsertOnSubmit(surveyData);
        try
        {
            this.retryPolicy.ExecuteAction(() => dataContext
                .SubmitChanges());
        }
        catch (SqlException ex)
        {
            Trace.TraceError(ex.TraceInformation());
            throw;
        }
    }
}

```



Если компания Tailspin хотела бы собирать информацию о повторах в приложении, то могла бы использовать событие **Повтор** в правиле повтора, чтобы получать сведения и записывать их в журнал для дальнейшего анализа.



Приложение Tailspin Surveys использует LINQ to SQL как реляционный модуль сопоставления объектов. Все виды взаимодействия базы данных можно получить из модели данных, поэтому для компании Tailspin нет необходимости использовать класс **ReliableSqlConnection** или классы модулей обработки данных SQL Azure, предоставляемые блоком Transient Fault Handling Application Block.

Требования компании Tailspin в отношении доступа к данным относительно просты, поэтому необходимо использовать самую простую версию метода **ExecuteAction**. Не нужно упаковывать все вызовы, которые возвращают значения или выполняют асинхронные вызовы.

## Установка и физическое развертывание

Приложение Tailspin Surveys использует стратегии повтора, определенные в файлах конфигурации для ролей, которые используют хранилище Windows Azure и SQL Azure. В примере все эти роли используют одни и те же стратегии повтора. В реальном развертывании необходимо настроить стратегии повтора для удовлетворения конкретных требований приложения.

## Дополнительные сведения

Инструкции по установке приложения Tailspin Surveys см. в приложении Б, «Руководство по установке Tailspin Surveys» в MSDN:

[http://msdn.microsoft.com/en-us/library/hh680894\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680894(v=PandP.50).aspx)

Дополнительные сведения о стратегиях повтора см. в разделе «Задание стратегий повтора в конфигурации» в MSDN:

[http://msdn.microsoft.com/en-us/library/hh680900\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh680900(v=PandP.50).aspx)

Чтобы получить простой доступ к ресурсам в сети, используйте электронную версию библиографии на MSDN:

[http://msdn.microsoft.com/en-us/library/hh749032\(v=PandP.50\).aspx](http://msdn.microsoft.com/en-us/library/hh749032(v=PandP.50).aspx)

# Приложение А Образцы конфигураций Развертывания приложения Tailspin Surveys в нескольких центрах обработки данных

Это приложение содержит образец служебной информации и определений правил для двух альтернативных архитектур для инфраструктуры автоматического масштабирования компании Tailspin, описанной в главе 5, «Создание более эластичного приложения Tailspin Surveys»

## Вариант 1

В варианте 1 производится развертывание приложения управления и модуля Autoscaling Application Block только в центре обработки данных для центральной и северной части США.

## Модель службы

### XML

```
<?xml version="1.0" encoding="utf-8"?>
<serviceModel ...>
  <subscriptions>
    <subscription name="Tailspin Surveys Production"
      subscriptionId="..."
      certificateThumbprint="..."
      certificateStoreName="My"
      certificateStoreLocation="LocalMachine">

  <services>
    <!--
      Multiple service definitions, one for each data center that
      Tailspin Surveys is deployed to.
      Note: All share the same wadStorageAccountName values because the
      diagnostic is collected centrally in this model.
    -->
    <service dnsPrefix="ustailspinsurveys" slot="Production" scalingMode="Scale">
      <roles>
        <role alias="usTailSpinWorkersSurveys">
```

```

        roleName="TailSpin.Workers.Surveys"
        wadStorageAccountName="usTailspin" />
<role alias="usTailSpinWebSurveyPublic"
    roleName="TailSpin.Web.Survey.Public"
    wadStorageAccountName="usTailspin" />
<role alias="usTailSpinWeb"
    roleName="TailSpin.Web"
    wadStorageAccountName="usTailspin" />
</roles>
</service>
<service dnsPrefix="eutailspinsurveys" slot="Production" scalingMode="Scale">
    <roles>
        <role alias="euTailSpinWorkersSurveys"
            roleName="TailSpin.Workers.Surveys"
            wadStorageAccountName="euTailspin" />
        <role alias="euTailSpinWebSurveyPublic"
            roleName="TailSpin.Web.Survey.Public"
            wadStorageAccountName="euTailspin" />
        <role alias="euTailSpinWeb"
            roleName="TailSpin.Web"
            wadStorageAccountName="euTailspin" />
    </roles>
</service>
<service dnsPrefix="astailspinsurveys" slot="Production" scalingMode="Scale">
    <roles>
        <role alias="asTailSpinWorkersSurveys"
            roleName="TailSpin.Workers.Surveys"
            wadStorageAccountName="asTailspin" />
        <role alias="asTailSpinWebSurveyPublic"
            roleName="TailSpin.Web.Survey.Public"
            wadStorageAccountName="asTailspin" />
        <role alias="asTailSpinWeb"
            roleName="TailSpin.Web"
            wadStorageAccountName="asTailspin" />
    </roles>
</service>
</services>
<!--
    Multiple storage accounts, one for each data center that Tailspin Surveys
    is deployed to. Each storage account has its own connection string.
    These queues are used by the Tailspin Surveys application and are
    included here because we want to use rules that monitor the queue lengths.
-->
<storageAccounts>
    <storageAccount alias="usTailspin" connectionString="...">

```



```

<queues>
  <queue alias="usTailspinSurveyAnswerStoredQueue"
    queueName="surveyanswerstored" />
  <queue alias="usTailspinSurveyTransferQueue"
    queueName="surveytransfer" />
</queues>
</storageAccount>
<storageAccount alias="euTailspin" connectionString="...">
  <queues>
    <queue alias="euTailspinSurveyAnswerStoredQueue"
      queueName="surveyanswerstored" />
    <queue alias="euTailspinSurveyTransferQueue"
      queueName="surveytransfer" />
  </queues>
</storageAccount>
<storageAccount alias="asTailspin" connectionString="...">
  <queues>
    <queue alias="asTailspinSurveyAnswerStoredQueue"
      queueName="surveyanswerstored" />
    <queue alias="asTailspinSurveyTransferQueue"
      queueName="surveytransfer" />
  </queues>
</storageAccount>
</storageAccounts>
</subscription>
</subscriptions>
<scaleGroups />
</serviceModel>

```

## Правила

### XML

```

<?xml version="1.0" encoding="utf-16"?>
<rules ...>
  <constraintRules>
    <!--
      An example constraint rule - there are more.
      We need duplicates of the same rule because the utcOffset values are different.
      In some cases it may be possible to use the same rule with an action for each
      region.
    -->
    <rule name="US WorkerRole reserving instances for midnight survey processing"
      description="..." enabled="true" rank="5">
      <timetable startDate="2011-10-05" endDate="2012-10-06" startTime="00:01:00"
        duration="02:00:00" utcOffset="-08:00">

```

```

    <daily />
  </timetable>
  <actions>
    <range target="usTailSpinWorkersSurveys" min="3" max="6" />
  </actions>
</rule>
<rule name="EU WorkerRole reserving instances for midnight survey processing"
  description="..." enabled="true" rank="5">
  <timetable startDate="2011-10-05" endDate="2012-10-06" startTime="00:01:00"
    duration="02:00:00" utcOffset="+00:00">
    <daily />
  </timetable>
  <actions>
    <range target="euTailSpinWorkersSurveys" min="3" max="6" />
  </actions>
</rule>
<rule name="ASIA WorkerRole reserving instances for midnight survey processing"
  description="..." enabled="true" rank="5">
  <timetable startDate="2011-10-05" endDate="2012-10-06" startTime="00:01:00"
    duration="02:00:00" utcOffset="+09:00">
    <daily />
  </timetable>
  <actions>
    <range target="asTailSpinWorkersSurveys" min="3" max="6" />
  </actions>
</rule>
</constraintRules>

<reactiveRules>
  <!--
    An example reactive rule - there are more.
    We need duplicates of the same rule because we must tie the operand source to
    the correct target.
  -->
  <rule name="US Heavy demand on public site" description="..." enabled="true">
    <actions>
      <scale target="usTailSpinWebSurveyPublic" by="1" />
    </actions>
    <when>
      <greater operand="usASPNET_requests_rejected" than="5" />
    </when>
    <rank>0</rank>
  </rule>

```

```

<rule name="EU Heavy demand on public site" description="..." enabled="true">
  <actions>
    <scale target="euTailSpinWebSurveyPublic" by="1" />
  </actions>
  <when>
    <greater operand="euASPNET_requests_rejected" than="5" />
  </when>
  <rank>0</rank>
</rule>
<rule name="AS Heavy demand on public site" description="..." enabled="true">
  <actions>
    <scale target="asTailSpinWebSurveyPublic" by="1" />
  </actions>
  <when>
    <greater operand="asASPNET_requests_rejected" than="5" />
  </when>
  <rank>0</rank>
</rule>
</reactiveRules>

<operands>
  <!-- Operands for US roles -->
  <performanceCounter alias="usASPNET_requests_rejected" timespan="00:10:00"
    aggregate="Average" source="usTailSpinWebSurveyPublic"
    performanceCounterName="\ASP.NET\Requests Rejected" />
  <performanceCounter alias="usCPU_over_20_minutes_for_TailspinWeb"
    timespan="00:20:00" aggregate="Average" source="usTailSpinWeb"
    performanceCounterName="\Processor(_Total)\% Processor Time" />
  <performanceCounter alias="usCPU_over_30_minutes" timespan="00:30:00"
    aggregate="Average" source="usTailSpinWorkersSurveys"
    performanceCounterName="\Processor(_Total)\% Processor Time" />

  <!-- Operands for EU roles -->
  <performanceCounter alias="euASPNET_requests_rejected" timespan="00:10:00"
    aggregate="Average" source="euTailSpinWebSurveyPublic"
    performanceCounterName="\ASP.NET\Requests Rejected" />
  <performanceCounter alias="euCPU_over_20_minutes_for_TailspinWeb"
    timespan="00:20:00" aggregate="Average" source="euTailSpinWeb"
    performanceCounterName="\Processor(_Total)\% Processor Time" />
  <performanceCounter alias="euCPU_over_30_minutes" timespan="00:30:00"
    aggregate="Average" source="euTailSpinWorkersSurveys"
    performanceCounterName="\Processor(_Total)\% Processor Time" />

  <!-- Operands for Asia roles -->
  <performanceCounter alias="asASPNET_requests_rejected" timespan="00:10:00"

```

```

    aggregate="Average" source="asTailSpinWebSurveyPublic"
    performanceCounterName="\ASP.NET\Requests Rejected" />
<performanceCounter alias="asCPU_over_20_minutes_for_TailspinWeb"
    timespan="00:20:00" aggregate="Average" source="asTailSpinWeb"
    performanceCounterName="\Processor(_Total)\% Processor Time" />
<performanceCounter alias="asCPU_over_30_minutes" timespan="00:30:00"
    aggregate="Average" source="asTailSpinWorkersSurveys"
    performanceCounterName="\Processor(_Total)\% Processor Time" />

<!-- Custom operands - require an attribute to specify the hosted service -->
<activeSurveysOperand alias="usNumberOfSurveysSubmitted" timespan="00:10:00"
    aggregate="Average" minNumberOfAnswers="0" hostedService="USTailspin"
    xmlns="http://Tailspin/ActiveSurveys" />
<tenantCountOperand alias="usNumberOfTenants" timespan="00:10:00"
    aggregate="Average" hostedService="USTailspin"
    xmlns="http://Tailspin/TenantCount" />

<activeSurveysOperand alias="euNumberOfSurveysSubmitted" timespan="00:10:00"
    aggregate="Average" minNumberOfAnswers="0" hostedService="EUTailspin"
    xmlns="http://Tailspin/ActiveSurveys" />
<tenantCountOperand alias="euNumberOfTenants" timespan="00:10:00"
    aggregate="Average" hostedService="EUTailspin"
    xmlns="http://Tailspin/TenantCount" />

<activeSurveysOperand alias="asNumberOfSurveysSubmitted" timespan="00:10:00"
    aggregate="Average" minNumberOfAnswers="0" hostedService="ASTailspin"
    xmlns="http://Tailspin/ActiveSurveys" />
<tenantCountOperand alias="asNumberOfTenants" timespan="00:10:00"
    aggregate="Average" hostedService="ASTailspin"
    xmlns="http://Tailspin/TenantCount" />

</operands>
</rules>

```

## Вариант 2

В варианте 2 производится развертывание приложения управления только в центре обработки данных для центральной и северной части США и модуля Autoscaling Application Block во всех центрах обработки данных. Каждый центр обработки данных имеет собственные определение модели службы и набор правил.

## Модель службы

### XML

```
<?xml version="1.0" encoding="utf-8"?>
<serviceModel ...>
  <subscriptions>
    <subscription name="Tailspin Surveys Production"
      subscriptionId="..."
      certificateThumbprint="..."
      certificateStoreName="My"
      certificateStoreLocation="LocalMachine">
      <services>
        <!--
          Single service definitions.
          Each data center has a copy of this model with the dnsPrefix changed to
          reflect the data center location.
        -->
        <service dnsPrefix="ustailspinsurveys" slot="Production" scalingMode="Scale">
          <roles>
            <role alias="TailSpinWorkersSurveys" roleName="TailSpin.Workers.Surveys"
              wadStorageAccountName="Tailspin" />
            <role alias="TailSpinWebSurveyPublic"
              roleName="TailSpin.Web.Survey.Public"
              wadStorageAccountName="Tailspin" />
            <role alias="TailSpinWeb" roleName="TailSpin.Web"
              wadStorageAccountName="Tailspin" />
          </roles>
        </service>
      </services>
      <!--
        Single storage account, duplicated in each data center.
        Each data center will have a different
        connectionString for its storage account.
      -->
      <storageAccounts>
        <storageAccount alias="Tailspin" connectionString="...">
          <queues>
            <queue alias="TailspinSurveyAnswerStoredQueue"
              queueName="surveyanswerstored" />
            <queue alias="TailspinSurveyTransferQueue" queueName="surveytransfer" />
          </queues>
        </storageAccount>
      </storageAccounts>
    </subscription>
  </subscriptions>
  <scaleGroups />
</serviceModel>
```

## Правила

### XML

```
<?xml version="1.0" encoding="utf-16"?>
<rules ...>
  <constraintRules>
    <!--
      An example constraint rule - there are more.
      Each region has its own rules file - it must be edited to reflect the different
      utcOffset values.
    -->
    <rule name="WorkerRole reserving instances for midnight survey processing"
      description="..." enabled="true" rank="5">
      <timetable startDate="2011-10-05" endDate="2012-10-06" startTime="00:01:00"
        duration="02:00:00" utcOffset="-08:00">
        <daily />
      </timetable>
      <actions>
        <range target="TailSpinWorkersSurveys" min="3" max="6" />
      </actions>
    </rule>
  </constraintRules>

  <reactiveRules>
    <!--
      An example reactive rule - there are more.
      Each region can have the same reactive rules, although you may want to change
      them in each region to reflect different usage patterns.
    -->
    <rule name="Heavy demand on public site" description="..." enabled="true">
      <actions>
        <scale target="TailSpinWebSurveyPublic" by="1" />
      </actions>
      <when>
        <greater operand="ASPNET_requests_rejected" than="5" />
      </when>
      <rank>0</rank>
    </rule>
  </reactiveRules>
```

```
<operands>
  <!-- Operands can be identical for the different regions -->
  <performanceCounter alias="ASPNET_requests_rejected" timespan="00:10:00"
    aggregate="Average" source="TailSpinWebSurveyPublic"
    performanceCounterName="\ASP.NET\Requests Rejected" />
  <performanceCounter alias="CPU_over_20_minutes_for_TailspinWeb"
    timespan="00:20:00" aggregate="Average" source="TailSpinWeb"
    performanceCounterName="\Processor(_Total)\% Processor Time" />
  <performanceCounter alias="CPU_over_30_minutes" timespan="00:30:00"
    aggregate="Average" source="TailSpinWorkersSurveys"
    performanceCounterName="\Processor(_Total)\% Processor Time" />

  <activeSurveysOperand alias="NumberOfSurveysSubmitted" timespan="00:10:00"
    aggregate="Average" minNumberOfAnswers="0"
    xmlns="http://Tailspin/ActiveSurveys" />
  <tenantCountOperand alias="NumberOfTenants" timespan="00:10:00"
    aggregate="Average" xmlns="http://Tailspin/TenantCount" />
</operands>

</rules>
```

# Приложение Б Руководство по установке приложения Tailspin Surveys

## Введение

В этом документе описана установка приложения Tailspin Surveys для пакета интеграции Microsoft® Enterprise Library 5.0 для Windows Azure, включая блок Autoscaling Application Block («WASABi») и блок Transient Fault Handling Application Block («Topaz»). Приложение Surveys является полным примером приложения, которое демонстрирует, как можно комплексно и эластично использовать блок Autoscaling Application Block для масштабирования приложения на технологической платформе Windows Azure™ на основе заданных правил. Кроме того, представлен пример сайта управления, который можно использовать для настройки служебной информации и правил, используемых блоком Autoscaling Application Block для контроля над тем, как изменяются экземпляры роли приложения на основе правил и условий, вызывающих изменения.

Приложение Tailspin Surveys содержит несколько сценариев и функций, но в нем нет многих других полезных функций блока Autoscaling Application Block или Transient Fault Handling Application Block или многих возможных полезных сценариев. Поэтому настоятельно рекомендуется использовать более широкие возможности и не ограничиваться базовыми способами использования блока Application Block в приложении Tailspin Surveys. Полный обзор блоков Application Block см. в разделах «*Autoscaling Application Block*» и «*Transient Fault Handling Application Block*».

Эта версия приложения Tailspin Surveys построена на предыдущих выпусках приложения Surveys центра разработчиков Microsoft patterns & practices. Материалы разделов, не представленных в данном документе, см. в документах по установке предыдущих выпусков.

Установка этого приложения может занять от часа до полутора часов, в зависимости от конфигурации оборудования и программного обеспечения, а также опыта работы с Windows Azure. Этот документ поможет выполнить следующее.

- Установка требуемых зависимостей с использованием включенного в состав поставки средства проверки зависимостей.



- Создание двух размещенных служб для веб-приложения Tailspin Surveys и приложения Tailspin Surveys для управления автоматическим масштабированием во время выполнения.
- Обновление файлов проекта с помощью сведений о подписке Windows Azure.
- Развертывание приложения в Windows Azure.
- Проверка установки на Windows Azure.

## Предварительные требования

Для запуска приложения Tailspin Surveys необходимо следующее.

- Компьютер для ведения разработки, на котором эксплуатируется система разработки Microsoft Visual Studio® 2010 с пакетом обновления 1 (SP1).
- Все необходимые обновления Microsoft Windows®.
- *Диспетчер пакетов NuGet (<http://nuget.codeplex.com/>).*
- Подписка Windows Azure с местом для двух размещенных служб (если нужно запустить приложение Tailspin Surveys и компонент Autoscaler с веб-приложением управления в Windows Azure).
- Учетная запись хранилища Windows Azure.
- Средство проверки зависимостей, обсуждаемое далее, которое будет проверять, установлены ли необходимые компоненты, перечисленные ниже. Если некоторых компонентов нет, это средство поможет их установить.
  - Visual Studio 2010.
  - MVC 3 Framework.
  - Пакет SDK Windows Azure для .NET и средств Windows Azure для ноябрьского выпуска Microsoft Visual Studio 2011 года.
  - Среда выполнения Windows Identity Foundation.
  - Microsoft Internet Information Services 7 (IIS) (необязательно).
    - Это необходимо для работы сайта управления в режиме имитации или для развертывания Autoscaler локально.

## Установка исходного кода и зависимостей

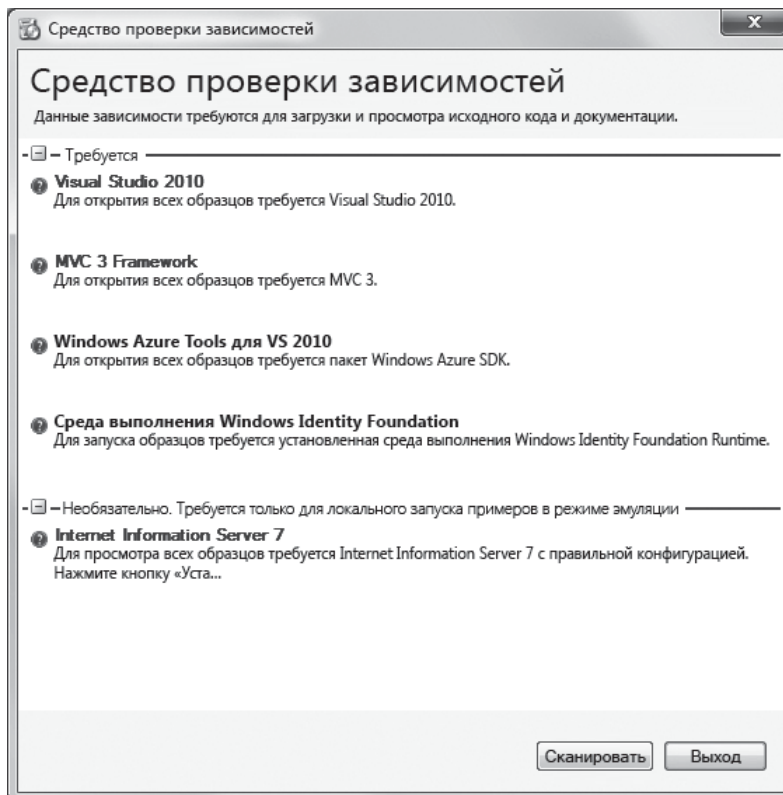
Средство проверки зависимостей проверит, установлены ли необходимые компоненты, и, если нет, поможет установить их.

## УСТАНОВКА ИСХОДНОГО КОДА ПРИЛОЖЕНИЯ TAILSPIN SURVEYS И ЗАВИСИМОСТЕЙ

1. Разархивируйте исходный код в папку на жестком диске.

**Примечание.** Исходный код нельзя помещать в папку, которая слишком глубоко вложена, поскольку средства Windows Azure для Visual Studio не поддерживают пути длиной более 255 символов. Мы протестировали развертывание приложения Tailspin Surveys в каталоге C:\Tailspin.

2. Запустите файл «CheckDependencies.cmd», расположенный в корне папки для загрузки.
3. В средстве проверки зависимостей нажмите кнопку **Просмотреть**.



4. Убедитесь, что установлены все необходимые зависимости. Если нет, установите по одному все необходимые компоненты и нажмите кнопку **Повторить просмотр**.

## Подготовка подписки Windows Azure для развертывания

Чтобы подготовить подписку Windows Azure для развертывания, необходимо сделать следующее:

1. создать сертификат управления Windows Azure и экспортировать его;
2. создать SSL-сертификат;
3. создать необходимые размещенные службы;
4. передать сертификаты как предназначенные для размещенных служб;
5. создать учетную запись хранилища.

В следующих разделах будет рассмотрена каждая из этих задач.

## Создание сертификата управления Windows Azure и его экспорт как PFX-файла

Чтобы иметь возможность выполнять масштабирование приложения, для модуля Enterprise Library Autoscaling Application Block необходимо использовать действительный сертификат управления для подписки Windows Azure. Если сертификата управления еще нет, для его создания можно выполнить следующие действия. Необходимо также передать на портал управления Windows Azure созданный сертификат управления и SSL-сертификат, который будет создан на следующих шагах. В общих словах, описанные ниже действия позволяют выполнять следующее:

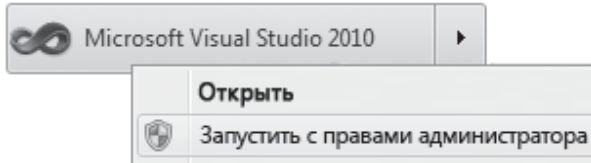
- создание сертификата управления;
- экспорт из папки «Текущий пользователь\Мое хранилище» для передачи на портал управления Windows Azure;
- создание SSL-сертификата для передачи на портал управления Windows Azure.

В отношении сертификатов необходимо учитывать следующее.

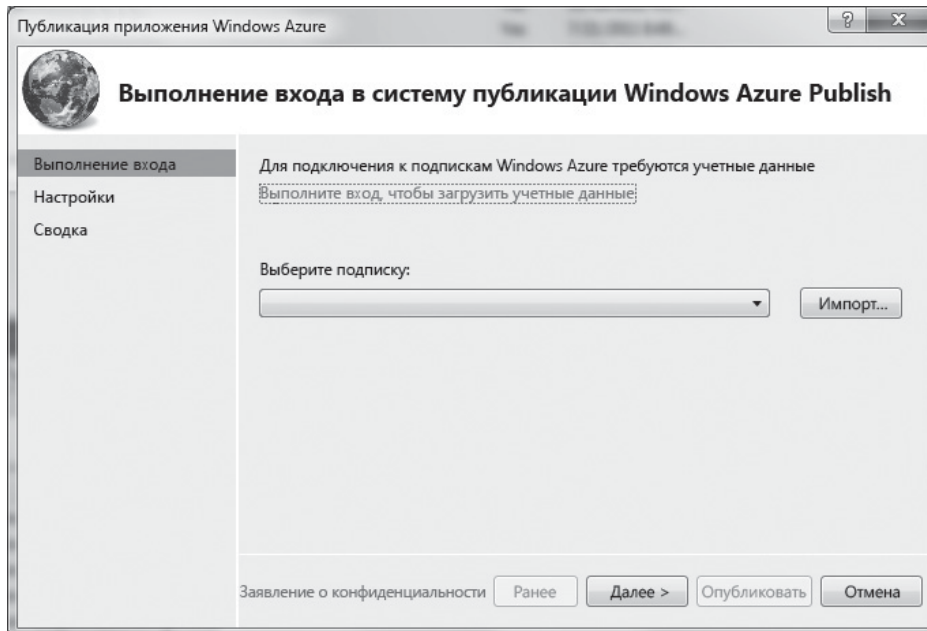
- Для ролей Windows Azure, которые предоставляют конечные точки SSL или влияют на изменения в развертывании Windows Azure, например Autoscaling Block, необходим доступ к соответствующим сертификатам в пределах выполняющихся экземпляров этих ролей. Эти сертификаты должны быть переданы в раздел «Сертификаты служб» соответствующей размещенной службы.
- Сертификаты, созданные в образце приложения для использования SSL, представлены только в целях демонстрации и не предназначены для использования в рабочей среде.
- Файлы управления и PFX-файлы SSL-сертификатов, созданные в процессе установки, должны храниться в безопасном месте со строгими разрешениями через списки управления доступом (ACL) или должны быть удалены после их передачи в Windows Azure.

## СОЗДАНИЕ СЕРТИФИКАТА УПРАВЛЕНИЯ WINDOWS AZURE И ЕГО ИМПОРТ ДЛЯ ИСПОЛЬЗОВАНИЯ В СРЕДЕ VISUAL STUDIO

1. Перейдите на сайт создания профиля Windows Azure по адресу <http://go.microsoft.com/fwlink/?LinkId=229563>.
2. Выполните вход в учетную запись Windows Live®, имеющую подписку Windows Azure.
3. Запустите среду Visual Studio от имени администратора. Для этого щелкните правой кнопкой мыши пункт **Microsoft Visual Studio 2010** и выберите **Запуск от имени администратора**.



4. Откройте решение Tailspin.sln.
5. Щелкните правой кнопкой мыши проект Tailspin.Surveys.Cloud и выберите команду **Опубликовать**. Обратите внимание, что мастер еще не завершил работу; будут выполнены лишь некоторые действия для создания и импорта сертификата управления в рамках подготовки к развертыванию.
6. На странице мастера «Публикация приложения Windows Azure» щелкните ссылку «**Вход в раздел учетных данных загрузки**».

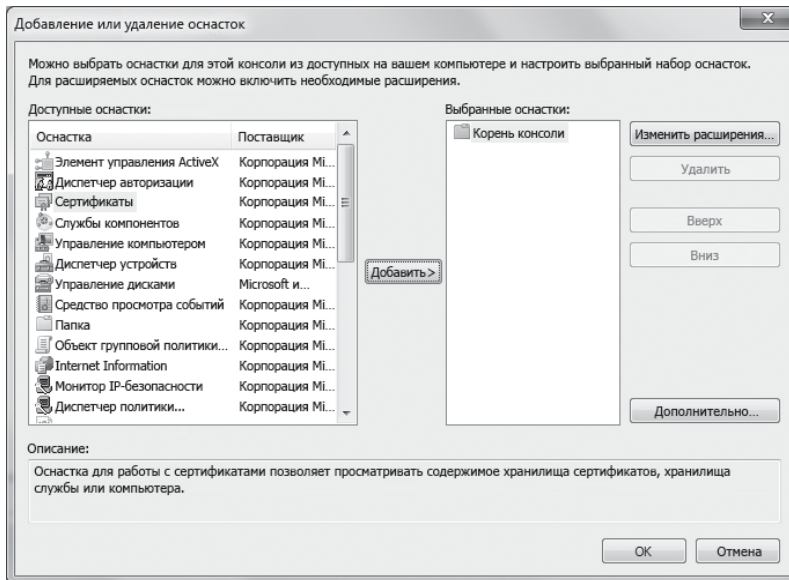


7. По ссылке откроется интернет-браузер, и произойдет переход на сайт Windows Azure. Войдите на сайт с учетной записью Windows Live с подпиской Windows Azure.
8. Выполните на странице следующие действия для загрузки файла с расширением PUBLISHSETTINGS.
9. Вернитесь на страницу мастера публикации приложения Windows Azure в Visual Studio.
10. Нажмите кнопку **Импорт** и перейдите к только что загруженному файлу PUBLISHSETTINGS.
11. Теперь можно нажать кнопку **Отменить**, поскольку перед публикацией в Windows Azure нужно подготовить решение.

**Примечание.** После этого шага среда Visual Studio будет настроена для публикации в подписке Windows Azure, а сертификат управления будет установлен в хранилище «Текущий пользователь\Мой сертификат».

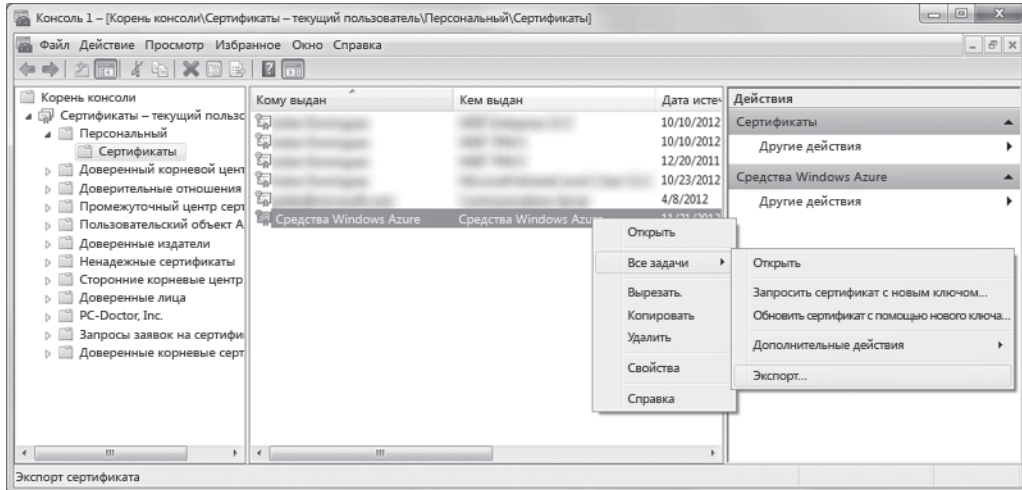
## ЭКСПОРТ СЕРТИФИКАТА УПРАВЛЕНИЯ WINDOWS AZURE В PFX-ФАЙЛ

1. Запустите консоль управления Майкрософт (MMC).
2. Выберите **файл**, затем **Добавить или удалить оснастку**.
3. В диалоговом окне «Добавление или удаление оснасток» в области «Доступные оснастки» выберите **Сертификаты** и нажмите кнопку **Добавить**.



4. В диалоговом окне «Оснастка диспетчера сертификатов» выберите **Моя учетная запись пользователя**. Нажмите кнопку **Готово**.
5. В диалоговом окне «Добавление или удаление оснасток» нажмите кнопку **ОК**.

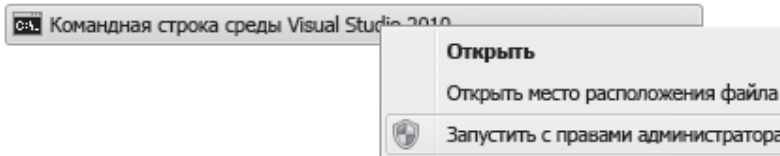
6. На консоли MMC выберите **Сертификаты — текущий пользователь**.
7. Разверните узел **личные данные** и выберите пункт **Сертификаты**.
8. В списке сертификатов консоли управления (MMC) найдите и щелкните правой кнопкой мыши сертификат **Средства Windows Azure**, импортированный на предыдущих шагах, затем выберите **Все задачи** и нажмите кнопку **Экспорт**.



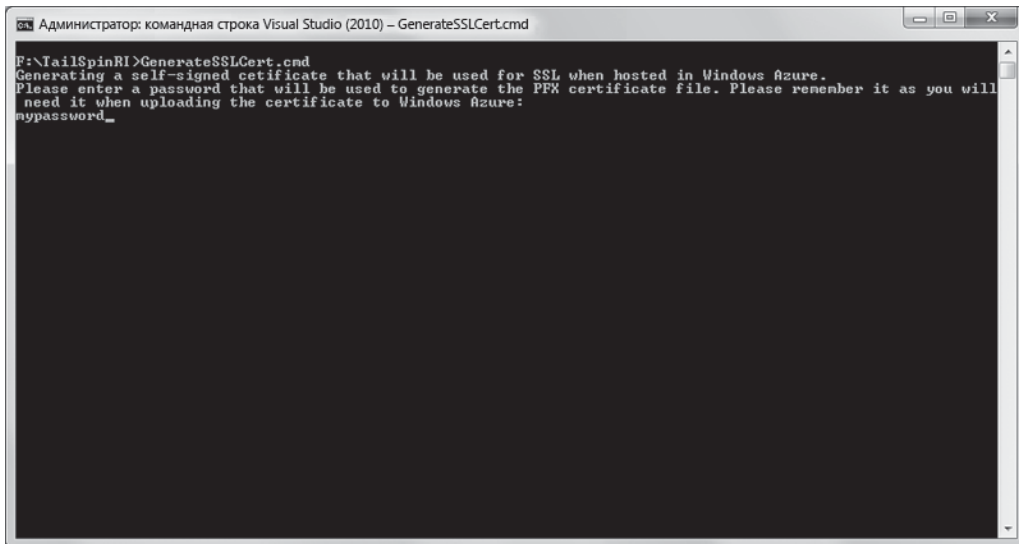
9. В мастере экспорта сертификатов нажмите кнопку **Далее**.
10. Нажмите кнопку **«Да»** и **экспортируйте закрытый ключ**. Нажмите кнопку **Далее**.
11. Определите **файл обмена персональными данными -PKCS #12 (PFX-файл)** как файл по умолчанию. Нажмите кнопку **Далее**.
12. Введите пароль для экспортированного файла. Не следует использовать пустой пароль, поскольку при передаче PFX-файла в Windows Azure пароль является обязательным.
13. Введите каталог, в котором находится приложение, присвойте ему имя, например «C:\Tailspin\AzureManagementCert.pfx», и нажмите кнопку **Далее**.
14. Нажмите кнопку **Готово**.

## Создание SSL-сертификата

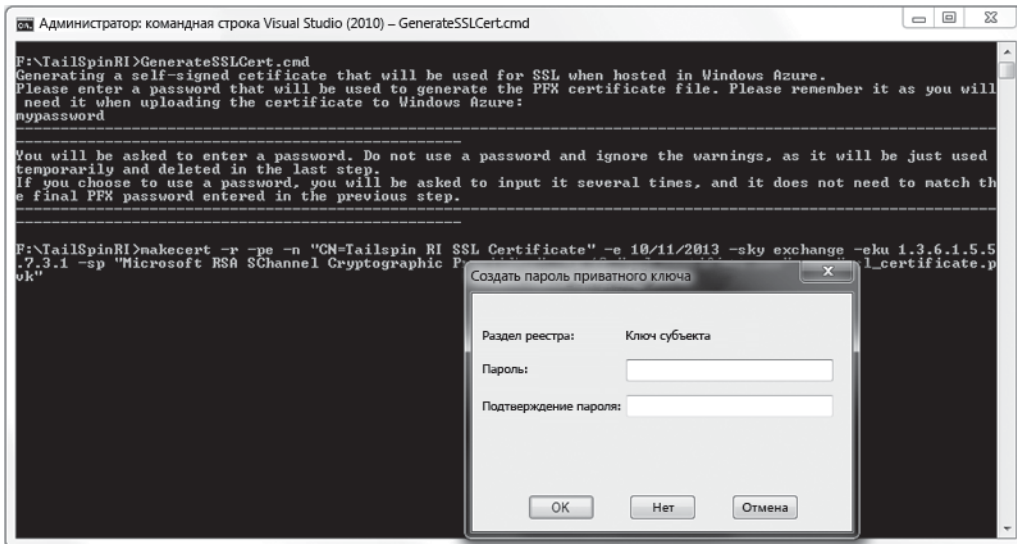
1. Откройте командную строку Visual Studio (2010) от имени администратора (в меню Windows 7 меню «Пуск» введите Visual Studio Command Prompt, выберите «Командная строка Visual Studio (2010)», щелкните правой кнопкой мыши и выберите **Запуск от имени администратора**).



2. Перейдите в каталог, где был разархивирован исходный код приложения Tailspin Surveys.
3. Для создания SSL-сертификата введите «GenerateSSLCert.cmd».
4. Введите пароль закрытого ключа для файла сертификата. Это поле является обязательным.



5. Появится всплывающее окно с запросом предоставить другой пароль. Оставьте это поле пустым и нажмите кнопку **ОК**.

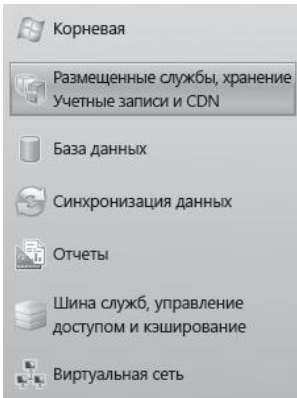


6. Появится всплывающее окно с подтверждением создания сертификата без защиты паролем. Нажмите кнопку **Да**, потому что этот последний пароль будет использоваться только временно, но созданный PFX-файл будет использовать пароль, введенный на предыдущем шаге.
7. После выполнения этих шагов вы будете иметь SSL-сертификат в файле с именем `ssl_certificate.pfx` в текущем каталоге.

## Создание необходимых размещенных служб

Далее необходимо будет создать размещенные службы.

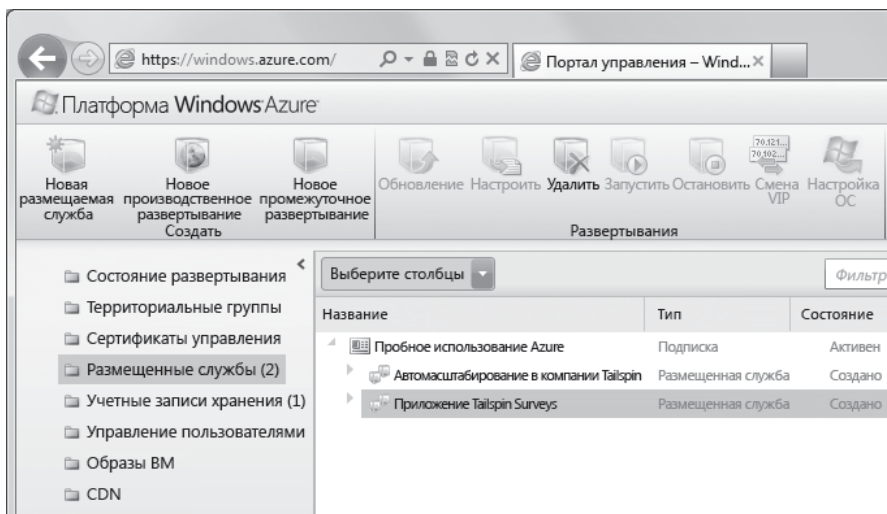
### СОЗДАНИЕ НЕОБХОДИМЫХ РАЗМЕЩЕННЫХ СЛУЖБ

1. На портале управления Windows Azure выберите **Размещенные службы**, **учетные записи хранилища и CDN** на панели навигации слева.
 
2. Выберите **Размещенные службы** в левом меню.
3. Выберите подписку Windows Azure.
4. Выберите **Создать размещенную службу** на панели ленты или в контекстном меню, щелкнув правой кнопкой мыши подписку Windows Azure.
5. В диалоговом окне «Создание новой размещенной службы» выполните следующее.
  - a. В текстовом поле «Введите имя службы» введите Tailspin-Surveys.
  - b. В текстовом поле «Введите префикс URL-адреса службы» введите выбранное имя для обозначения, что это веб-приложение Tailspin. Этот префикс должен быть уникальным.



- c. Из раскрывающегося меню «Создайте или выберите территориальную группу» выберите **Создать новую территориальную группу**.
  - d. В диалоговом окне **Создание новой территориальной группы** введите имя территориальной группы, выберите местоположение и нажмите кнопку **ОК**.
  - e. Выберите **не развертывать**.
  - f. Нажмите кнопку **ОК**.
6. Выберите подписку Windows Azure для создания второй размещенной службы.
7. В диалоговом окне **Создание новой размещенной службы** выполните следующее.
- a. В текстовом поле «Введите имя службы» введите Tailspin-Autoscaling.
  - b. В текстовом поле «Введите префикс URL-адреса службы» введите выбранное имя для обозначения, что это веб-сайт управления автоматическим масштабированием Tailspin и рабочая роль времени выполнения. Этот префикс должен быть уникальным.
  - c. Из раскрывающегося меню «Создайте или выберите территориальную группу» выберите территориальную группу, созданную на предыдущем шаге.
  - d. Выберите «**не развертывать**».
  - e. Нажмите кнопку **ОК**.

Должны быть видны две размещенные службы.



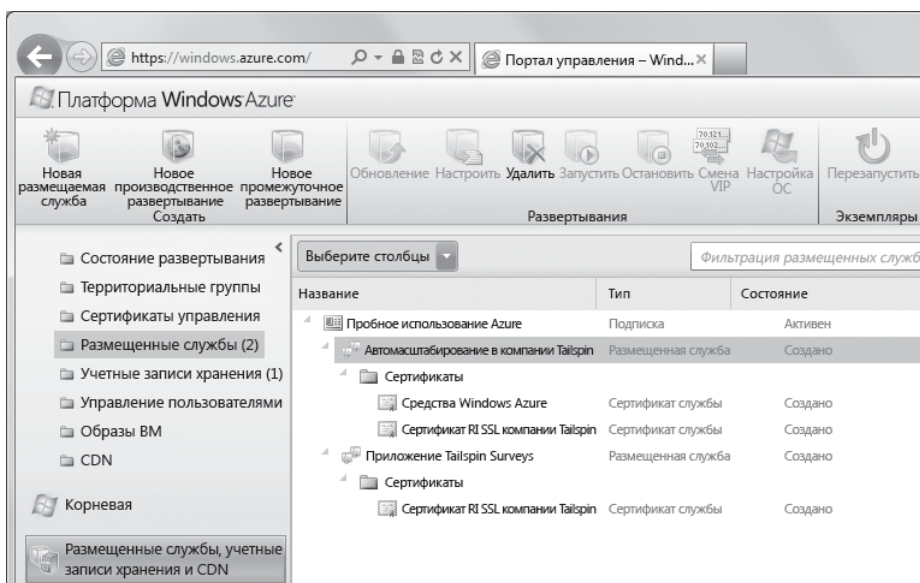
## Передача сертификатов как относящихся к размещенным службам

Облачные проекты, которые будут развернуты в Windows Azure, используют сертификаты служб, которые должны быть переданы в хранилище сертификатов управления службами Windows Azure. Они являются SSL-сертификатами, которые использует веб-сайт, а сертификат управления Windows Azure использует блок Autoscaling Application Block для регулирования количества экземпляров настроенных ролей.

### ДОБАВЛЕНИЕ СЕРТИФИКАТОВ

1. Добавьте SSL-сертификат к размещенной службе Tailspin-Surveys.
  - a. На портале управления Windows Azure выберите папку «Сертификаты» под размещенной службой Tailspin-Surveys и нажмите кнопку **Добавить сертификат**.
  - b. В диалоговом окне «Передача сертификата X.509» перейдите к созданному ранее сертификату `ssl_certificate.pfx`.
  - c. Введите пароль. Нажмите кнопку **ОК**.
  - d. Должен быть виден **сертификат Tailspin RI SSL**, добавленный в список сертификатов служб на портале.
2. Добавьте SSL-сертификат к размещенной службе Tailspin-Autoscaling.
  - a. На портале управления Windows Azure выберите папку «Сертификаты» под размещенной службой Tailspin-Autoscaling и нажмите кнопку **Добавить сертификат**.
  - b. В диалоговом окне **Передача сертификата X.509** перейдите к созданному ранее сертификату `ssl_certificate.pfx`.
  - c. Введите пароль. Нажмите кнопку **ОК**.
  - d. Должен быть виден **сертификат Tailspin RI SSL**, добавленный в список сертификатов служб на портале.
3. Добавьте сертификат управления размещенной службой Tailspin-Autoscaling.
  - a. На портале управления Windows Azure выберите папку «Сертификаты» под размещенной службой Tailspin-Autoscaling и нажмите кнопку **Добавить сертификат**.
  - b. В диалоговом окне **Передача сертификата X.509** перейдите к экспортированному ранее сертификату `Azure ManagementCert.pfx`.
  - c. Введите пароль. Нажмите кнопку **ОК**.
  - d. Должен быть виден сертификат **Средства Windows Azure**, добавленный в список сертификатов служб на портале.

Должны быть видны размещенные службы с сертификатами, как показано на следующем рисунке.



## Создание учетной записи хранилища

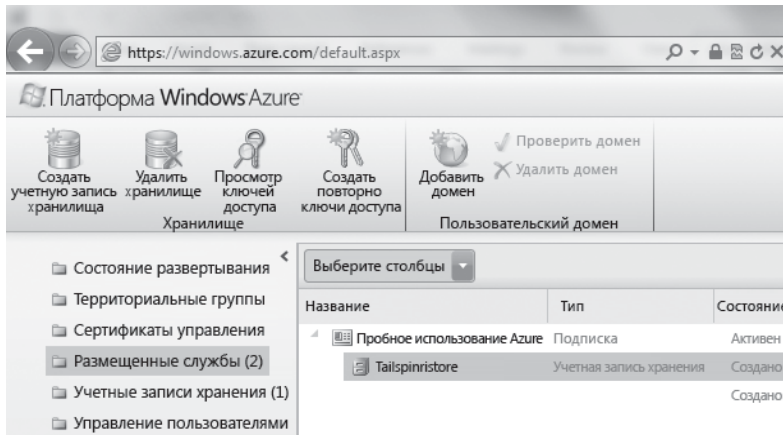
Autoscaling Application Block использует хранилище Windows Azure для своих операций. Дополнительные сведения об автомасштабировании и о том, как работает блок автомасштабирования приложения, см. в главе 4, «Автоматическое масштабирование и Windows Azure», в этом руководстве. На этом шаге будет создана учетная запись хранилища, которую Autoscaling Application Block будет использовать в сценарии Tailspin.

### СОЗДАНИЕ УЧЕТНОЙ ЗАПИСИ ХРАНИЛИЩА

1. На портале управления Windows Azure выберите раздел **Учетные записи хранилища**.
2. Выберите правильную подписку, щелкните правой кнопкой мыши и выберите **Создать учетную запись хранилища**.
3. В диалоговом окне «Создание новой учетной записи хранилища» выполните следующее.
  - a. Выберите подписку.
  - b. Введите URL-адрес.
  - c. Выберите регион или созданную ранее территориальную группу. d. Нажмите кнопку **ОК**.

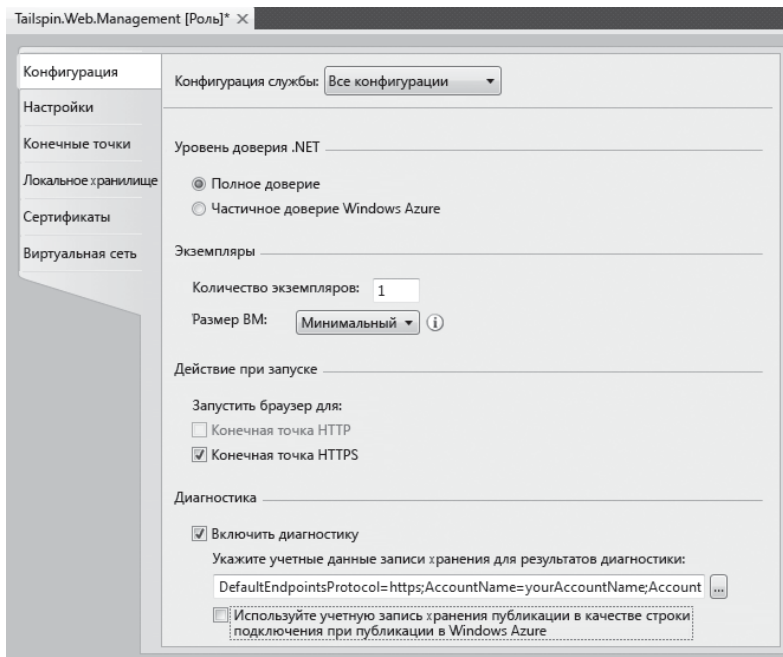
Должна появиться добавленная учетная запись хранилища.

**Примечание.** Необходимо записать имя учетной записи хранилища, которое понадобится для настройки решения Visual Studio.



**ЧТОБЫ УБЕДИТЬСЯ, ЧТО ПРИЛОЖЕНИЕ ИСПОЛЬЗУЕТ СОЗДАННУЮ УЧЕТНУЮ ЗАПИСЬ ХРАНИЛИЩА, ВЫПОЛНИТЕ СЛЕДУЮЩИЕ ДЕЙСТВИЯ**

1. В Visual Studio для каждой из ролей облачных проектов дважды щелкните роль и перейдите на вкладку **Конфигурация**.
2. Снимите флажок «Использовать учетную запись хранилища публикаций как строку подключения при публикации в Windows Azure», если он установлен (он не должен быть установлен).



## Построение решения

Перед построением решения необходимо выполнить следующие действия.

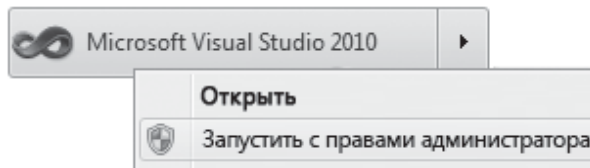
1. Установите пакеты NuGet.
2. Измените сертификаты в облачных проектах Visual Studio.
3. Подготовьте параметры в облачных проектах.

### Установка пакетов NuGet

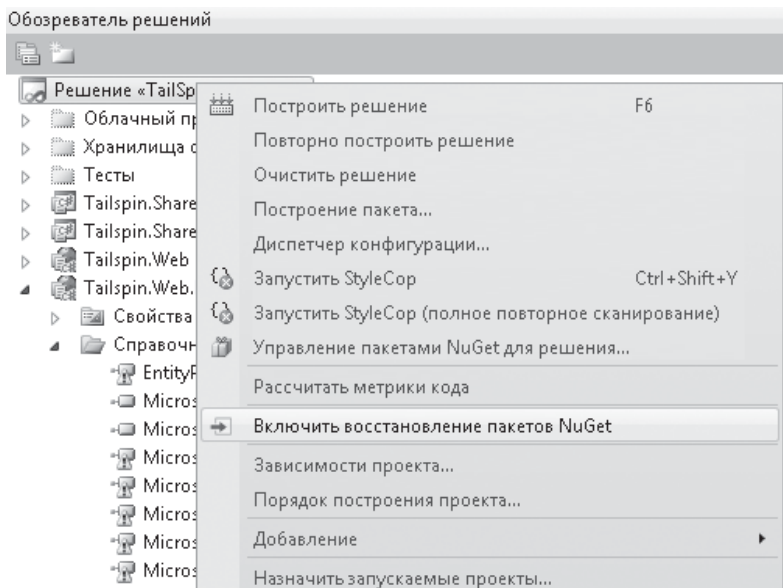
Приложение Tailspin Surveys зависит от определенных двоичных файлов, которые не включены в ZIP-файл. Выполните следующие шаги, чтобы загрузить и установить эти зависимости с использованием диспетчера пакетов NuGet.

#### УСТАНОВКА ПАКЕТОВ NUGETT

1. Запустите среду Visual Studio от имени администратора. Для этого щелкните правой кнопкой мыши пункт **Microsoft Visual Studio 2010** и выберите **Запуск от имени администратора**.



2. Откройте решение Tailspin.sln.
3. В обозревателе решений щелкните правой кнопкой мыши решение Tailspin, выберите **Включить восстановление пакета NuGet** и нажмите кнопку **Да** в диалоговом окне подтверждения.



**Примечание.** Если вы используете NuGet версии 1.5 или более ранней версии, то не увидите элемент меню **Включить восстановление пакета NuGet**. В этом случае введите следующие команды в консоли диспетчера пакетов NuGet. Открыть консоль диспетчера пакетов NuGet можно в меню Visual Studio. Укажите пункт **Сервис**, затем **Диспетчер пакетов библиотек**, выберите пункт **Консоль диспетчера пакетов** и введите следующие команды:

- Install-Package NuGetPowerTools;
- Enable-PackageRestore.

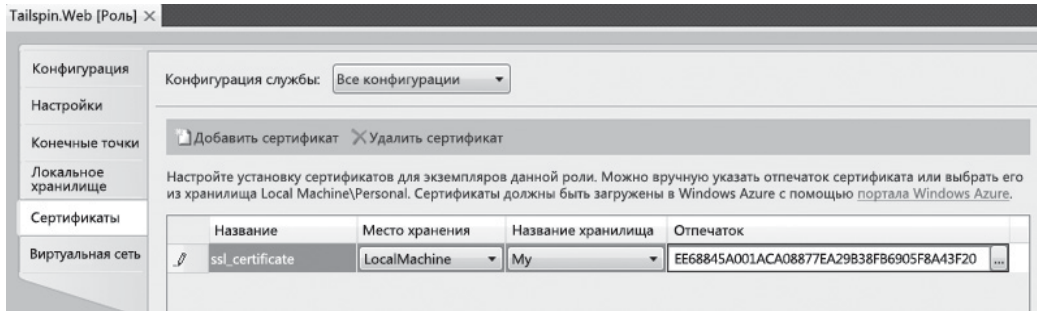
## Изменение сертификатов в облачных проектах Visual Studio

Облачные проекты в сценарии Tailspin используют различные сертификаты, загруженные в Windows Azure в предыдущих разделах. На этом шаге вам предстоит изменить облачные проекты, чтобы правильно указывать ссылки на эти сертификаты для ролей.

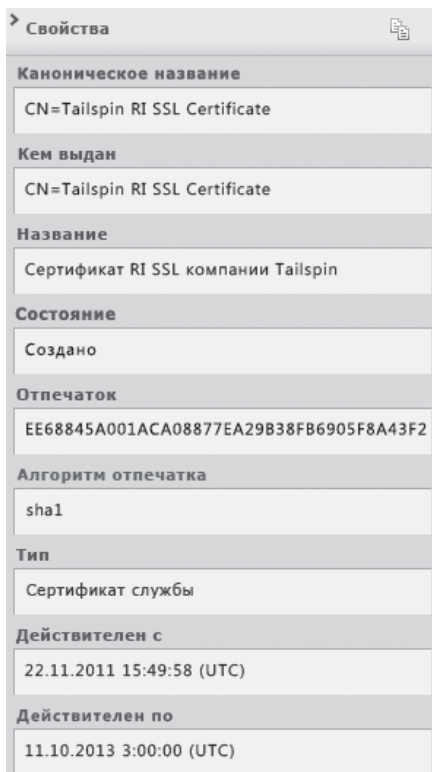
### ИЗМЕНЕНИЕ СЕРТИФИКАТОВ В ОБЛАЧНЫХ ПРОЕКТАХ VISUAL STUDIO

1. В Visual Studio в обозревателе решений выберите роль **Tailspin.Web** в проекте Tailspin.Surveys.Cloud.
  - a. Дважды щелкните роль Tailspin.Web, а затем выберите Сертификаты.
  - b. Необходимо сделать отпечаток **ssl\_certificate**, соответствующий отпечатку **SSL-сертификата Tailspin RI** на портале управления Windows Azure. Обновите отпечаток в Visual Studio, скопировав значение из портала. Кроме того, убедитесь, что в качестве местоположения хранилища указано **LocalMachine**, а в качестве имени хранилища — **My**.

## Отпечаток SSL-сертификата в Visual Studio



## Отпечаток SSL-сертификата в Windows Azure



2. Дважды щелкните роль **Tailspin.Web.Management** в проекте Tailspin.Autoscaling.Cloud, выберите **Сертификаты** и убедитесь, что отпечаток для **ssl\_certificate** соответствует отпечатку для **сертификата Tailspin RI SSL** на портале управления Windows Azure.
3. Дважды щелкните роль **Tailspin.Workers.Autoscaling** в проекте Tailspin.Autoscaling.Cloud, выберите **Сертификаты** и убедитесь, что отпечаток для **TailspinManagementCert** соответствует отпечатку для сертификата **средств Windows Azure** на портале управления Windows Azure.

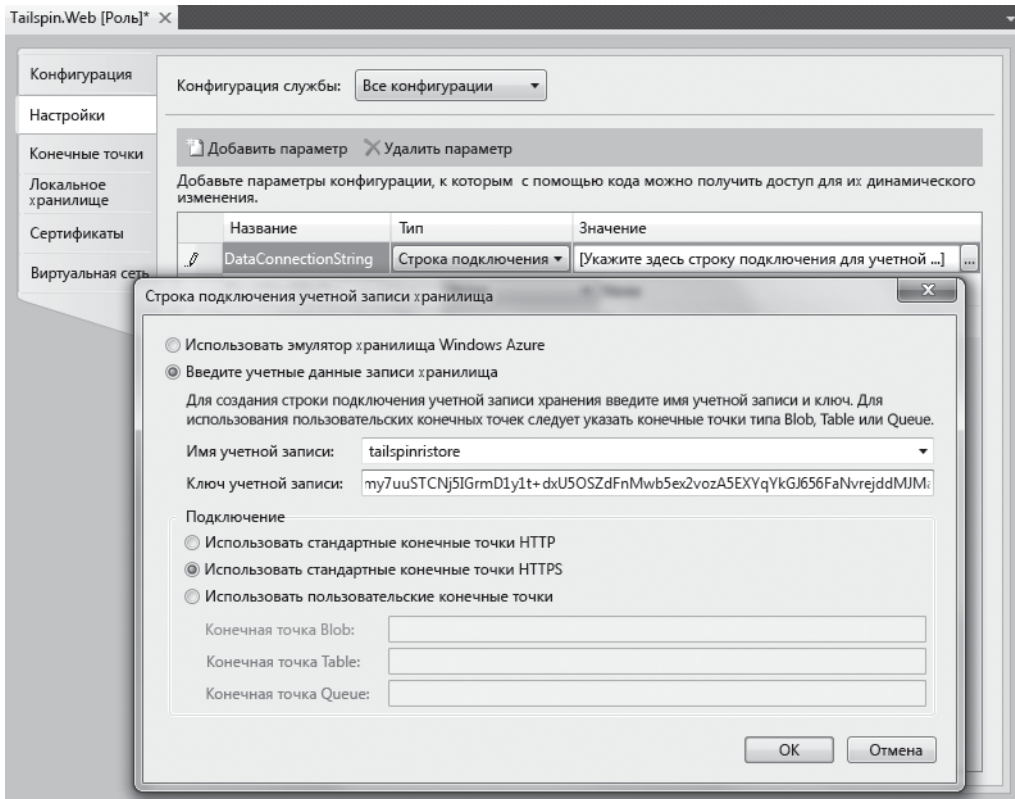
## Подготовка параметров в облачных проектах

На этом шаге необходимо обновить параметры в облачных проектах, чтобы указать учетную запись хранилища, созданную ранее.

### ПОДГОТОВКА ПАРАМЕТРОВ В ОБЛАЧНЫХ ПРОЕКТАХ

1. На портале управления Windows Azure выберите раздел **Учетные записи хранилища**.
2. Выберите учетную запись хранилища, созданную ранее.
  - a. Справа в поле «Первичный ключ доступа» выберите Просмотр.
  - b. В окне «Просмотр ключей доступа к хранилищу» скопируйте **первичный ключ доступа**.
3. В среде Visual Studio дважды щелкните роль **Tailspin.Web** в проекте Tailspin.Surveys.Cloud и выберите **Параметры**.
  - a. В раскрывающемся списке «Конфигурация службы» выберите Все конфигурации.
  - b. Убедитесь, что тип параметра для **DataConnectionString** задан как **Строка подключения**.
  - c. Нажмите кнопку с многоточием (...) справа от поля **Значение DataConnectionString**.
    - i. В диалоговом окне «Строка подключения учетной записи хранилища» выберите параметр **Ввести учетные данные записи хранилища**.
    - ii. Укажите в поле **Имя учетной записи** соответствующее имя учетной записи хранилища, созданной ранее.
    - iii. Укажите в поле **Ключ учетной записи** первичный ключ доступа, скопированный на шаге 2.





4. Укажите в строке подключения одинаковые значения для всех следующих параметров роли.

**Примечание.** Значение строки подключения придется обновить в нескольких параметрах. Чтобы каждый раз не открывать диалоговое окно «Строка подключения учетной записи хранилища», можно скопировать значение строки подключения, созданное на предыдущем шаге, и вставить его в поле **Значение** для всех остальных параметров.

Роль	Имя параметра
Tailspin.Web	DataConnectionString
Tailspin.Web	Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString
Tailspin.Web.Survey.Public	DataConnectionString
Tailspin.Web.Survey.Public	Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString
Tailspin.Workers.Surveys	DataConnectionString
Tailspin.Workers.Surveys	Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString
Tailspin.Web.Management	DataConnectionString
Tailspin.Web.Management	Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString
Tailspin.Web.Management	AutoscalingStorage
Tailspin.Wokers.Autoscaling	DataConnectionString
Tailspin.Wokers.Autoscaling	Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString
Tailspin.Wokers.Autoscaling	AutoscalingStorage

#### СОЗДАНИЕ ПАРОЛЯ ДЛЯ ДОСТУПА К САЙТУ УПРАВЛЕНИЯ АВТОМАТИЧЕСКИМ МАСШТАБИРОВАНИЕМ

1. В Visual Studio дважды щелкните роль **Tailspin.Web.Management** в проекте Tailspin.Autoscaling.Cloud, а затем щелкните **Параметры**.
2. Задайте в параметрах имя пользователя и пароль.

**Примечание.** Установка пароля является обязательной для обеспечения безопасности.

## Построение решения

На данный момент вы готовы к построению решения Tailspin.

#### ПОСТРОЕНИЕ РЕШЕНИЯ

1. В обозревателе решений Visual Studio выберите решение Tailspin.
2. Щелкните решение правой кнопкой мыши и выберите пункт **Перестроить решение**.

## Развертывание в Windows Azure

В Windows Azure необходимо развернуть следующие проекты:

- Tailspin.Surveys.Cloud (в размещенной службе Tailspin-Surveys).
- Tailspin.Autoscaling.Cloud (в размещенной службе Tailspin-Autoscaling).

После развертывания одного проекта можно развернуть следующий, как только вы увидите, что развертывание было добавлено и запущено в окне журнала активности Visual Studio Windows Azure.

**Известная проблема.** Из-за ограничений в Visual Studio 2010 IntelliTrace *вызывается следующее исключение времени выполнения, если программный блок размещается в Windows Azure с включенной функцией IntelliTrace:*

*System.Security.VerificationException: операция может дестабилизировать среду выполнения.*

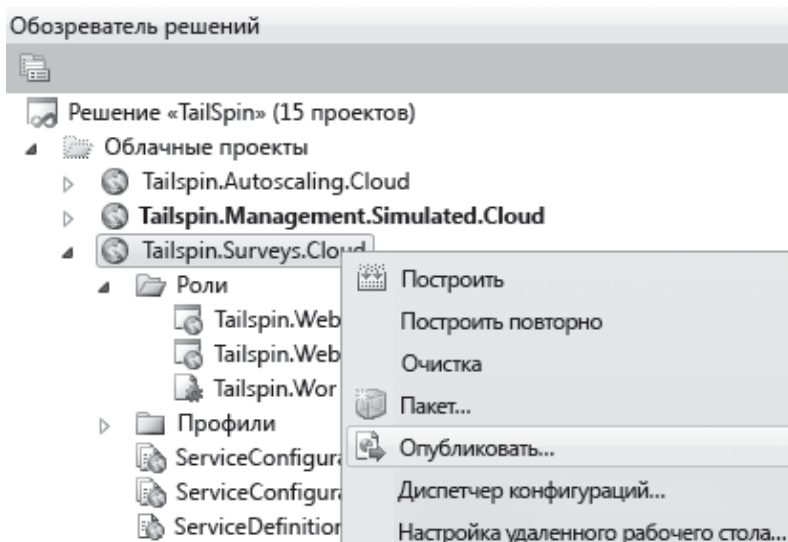
**Способ устранения проблемы.** Для решения этой проблемы необходимо отключить IntelliTrace в сборке программного блока автоматического масштабирования. Для этого добавьте *Microsoft.Practices.EnterpriseLibrary.WindowsAzure.\** в список исключенных модулей. См. инструкции здесь: *VerificationException on Windows Azure IntelliTrace.*

## Развертывание решения Tailspin в промежуточном слоте

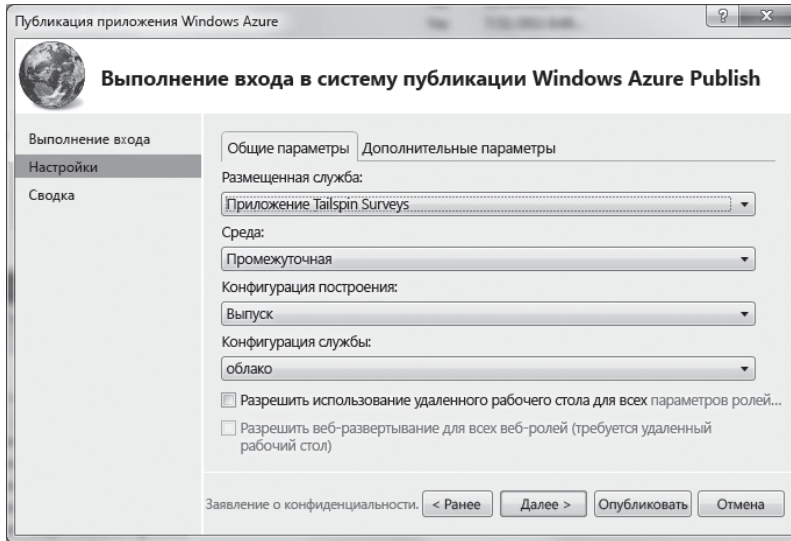
Сначала необходимо развернуть эти проекты в промежуточном слоте развертывания в размещенной службе, так как значения по умолчанию в хранилище служебной информации для Tailspin Surveys ссылаются на развертывание в промежуточном слоте.

### РАЗВЕРТЫВАНИЕ РЕШЕНИЯ TAILSPIN В ПРОМЕЖУТОЧНОМ СЛОТЕ

1. В обозревателе решений щелкните правой кнопкой мыши проект **TailSpin.Surveys.Cloud** и выберите команду **Опубликовать**.



2. В мастере публикации приложений Windows Azure выберите следующее.
  - a. На шаге входа для публикации в Windows Azure выберите подписку, импортированную на предыдущих шагах, и нажмите кнопку «Далее».
  - b. На шаге настройки параметров публикации в Windows Azure выберите **Tailspin-Surveys** в качестве размещенной службы.
  - c. Выберите **Промежуточная** в качестве среды.
  - d. В разделе «Конфигурация службы» выберите параметр **Облако**.
  - e. Нажмите кнопку **Опубликовать**, чтобы опубликовать размещенную службу.



3. Прежде чем перейти на следующий шаг, дождитесь окончания завершения построения проекта и убедитесь, что развертывание было добавлено и запущено в окне журнала активности Visual Studio Windows Azure.
4. В обозревателе решений щелкните правой кнопкой мыши проект **Tailspin.Autoscaling.Cloud** и выберите команду **Опубликовать**.
5. В мастере публикации приложений Windows Azure выберите следующее.
  - a. На шаге входа для публикации в Windows Azure выберите подписку, импортированную на предыдущих шагах, и нажмите кнопку «Далее».
  - b. На шаге настройки параметров публикации в Windows Azure выберите **Tailspin-Autoscaling** в качестве размещенной службы.
  - c. Выберите **Промежуточная** в качестве среды.
  - d. В разделе «Конфигурация службы» выберите параметр **Облако**.
  - e. Нажмите кнопку **Опубликовать**, чтобы опубликовать размещенную службу.

## Тестирование работы Tailspin Surveys

Необходимо убедиться, что приложение Tailspin Surveys работает на общедоступном веб-сайте, веб-сайте клиента и веб-сайте управления.

### Общедоступный веб-сайт

Tailspin включает в себя общедоступный веб-сайт, который позволяет конечным пользователям проходить опросы. Для просмотра этого сайта перейдите к размещенной службе Tailspin-Surveys.

*[https://\[dns for Tailspin-Surveys hosted service\]](https://[dns for Tailspin-Surveys hosted service])*

Этот URL-адрес можно найти на портале управления Windows Azure. Перейдите в раздел «Размещенные службы» и выберите развертывание Tailspin.Surveys.Cloud, которое расположено под размещенной службой Tailspin-Surveys. Вы увидите DNS-имя в сведениях справа. Его можно щелкнуть правой кнопкой мыши и скопировать.

### Веб-сайт клиента

Клиенты могут получить доступ к веб-сайту клиента. Этот веб-сайт позволяет клиентам создавать опросы и анализировать результаты. Доступно два клиента: Adatum и Fabrikam.

Для просмотра этого сайта перейдите к размещенной службе Tailspin-Surveys с использованием HTTPS. Этот URL-адрес можно найти на портале управления Windows Azure.

*[https://\[dns for Tailspin-Surveys hosted service\]](https://[dns for Tailspin-Surveys hosted service])*

URL-адрес совпадает с общедоступным веб-сайтом, но схему нужно изменить с HTTP на HTTPS.

Так как это решение использует тестовый сертификат, вы увидите сообщение: «Продолжить для этого веб-сайта (не рекомендуется)». Щелкните эту ссылку, чтобы продолжить.

При просмотре этого сайта вход выполняется с использованием проверки подлинности на основе заявок (эмулированной).

### Веб-сайт управления

Операторы Tailspin могут использовать этот веб-сайт для управления правилами и отслеживания процесса автоматического масштабирования. Существует также страница для создания моделируемой нагрузки на веб-сайт для демонстрации работы автоматического масштабирования.

Для просмотра этого сайта перейдите к размещенной службе Tailspin-Autoscaling с использованием HTTPS.

*[https://\[dns for Tailspin-Autoscaling hosted service\]](https://[dns for Tailspin-Autoscaling hosted service])*

Этот URL-адрес можно найти на портале управления Windows Azure. Перейдите в раздел «Размещенные службы» и выберите развертывание Tailspin.Autoscaling.Cloud, которое расположено под размещенной службой Tailspin-Autoscaling. Вы увидите DNS-имя в сведениях справа. Его можно щелкнуть правой кнопкой мыши и скопировать, но не забудьте обновить схему с HTTP на HTTPS.

Так как это решение использует тестовый сертификат, вы увидите сообщение: «Продолжить для этого веб-сайта (не рекомендуется)». Щелкните эту ссылку, чтобы продолжить.

# Настройка функции автоматического масштабирования

## Tailspin

Теперь все готово к настройке хранилища служебной информации и загрузке образцов правил.

### Настройка хранилища служебной информации

Хранилище служебной информации определяет роли в приложении, которое можно масштабировать, а также дополнительные параметры для работы программного блока автоматического масштабирования. Чтобы увидеть этот сценарий в работе, необходимо настроить хранилище служебной информации в соответствии с созданной средой.

#### НАСТРОЙКА ХРАНИЛИЩА СЛУЖЕБНОЙ ИНФОРМАЦИИ

1. Перейдите на сайт управления Tailspin.
2. Перейдите по ссылке **Восстановить модель службы по умолчанию** под заголовком **XML-файл служебной информации** в разделе «Правила и хранилища служб». Этот параметр доступен только на домашней странице, но не на вкладке **Служебная информация** на вкладке «Навигация».

**Примечание.** Эта операция передает в хранилище больших двоичных объектов копию файла **default-service-information-set.xml**, расположенного в папке `SourceCode\Tailspin\Sample stores` в папке установки.

3. Файл служебной информации по умолчанию не содержит правильной информации о подписке, поэтому щелкните вкладку **Служебная информация** на панели навигации, чтобы обновить ее в соответствии с текущим развертыванием в Windows Azure.
4. Щелкните в списке **Подписки** строку с именем **TailspinSubscription**. Это значение потребуется вставить на следующем шаге.
5. На портале управления Windows Azure выберите вашу подписку.
  - a. В правой области выберите и скопируйте идентификатор подписки с помощью клавиш «CTRL» + «C» (портал — это приложение подключаемого модуля браузера Microsoft Silverlight®, поэтому контекстное меню недоступно по щелчку правой кнопкой мыши).
6. Вставьте значение в поле **SubscriptionId** на форме **Сведения о подписке** на сайте управления Tailspin.
7. Выполните следующие действия в разделе «Сертификаты управления» на портале управления Windows Azure.
  - a. Выберите загруженный сертификат управления Tailspin с именем «Средства Windows Azure».
  - b. Выберите и скопируйте отпечаток сертификата на правой панели.
  - c. Вставьте значение отпечатка в поле **Отпечаток сертификата управления** на форме **Сведения о подписке** на сайте управления Tailspin.
8. На портале управления Windows Azure выберите размещенную службу **Tailspin-Surveys**.

- a. Выберите значение префикса DNS размещенной службы на правой панели и скопируйте его.
  - b. Вставьте значение в поле **Префикс DNS** в разделе «Развернутые размещенные службы» на форме сайта управления Tailspin.
9. В Visual Studio дважды щелкните роль **Tailspin.Workers.Autoscaling** в списке ролей проекта Tailspin.Management.Worker.Cloud.
- a. Перейдите на вкладку «Параметры».
  - b. Скопируйте значение параметра **AutoscalingStorage**.
  - c. Вставьте значение в поле **Строка подключения** в разделе **Учетная запись хранилища** на форме сайта управления Tailspin.
10. Нажмите кнопку **Сохранить** в нижней части страницы.

## Загрузка образцов правил

1. На сайте управления Tailspin выберите вкладку **Домашняя**.
2. Перейдите по ссылке **Восстановить правила по умолчанию** под заголовком **XML-файл набора правил** в разделе «Хранилища правил и служебной информации». Этот параметр доступен только на домашней странице, но не на вкладке **Служебная информация** на вкладке «Навигация».

**Примечание.** Эта операция позволяет отправить в хранилище больших двоичных объектов копию файла **default-rules-set.xml**, расположенного в папке `SourceCode\Tailspin\Sample stores` в папке установки.

После выполнения этих шагов приложение Tailspin Surveys должно автоматически масштабироваться на основе заданных правил. Перейдите на вкладку **Мониторинг**, чтобы увидеть, как конечное приложение постепенно изменяется, либо перейдите на соответствующие вкладки и измените правила реагирования и правила ограничения во время выполнения, чтобы увидеть, как приложение работает с другими параметрами.

## Локальный запуск Tailspin Surveys в режиме отладки

Программный блок автоматического масштабирования использует хранилище Windows Azure, которое появилось в пакете SDK 1.6, и в настоящее время не поддерживается эмулятором хранилища. Таким образом, при запуске локальной отладки все еще требуется доступ к активной учетной записи хранилища Windows Azure. Поэтому необходимо настроить все строки подключения в разделе «Подготовка параметров в облачных проектах» как Все конфигурации.

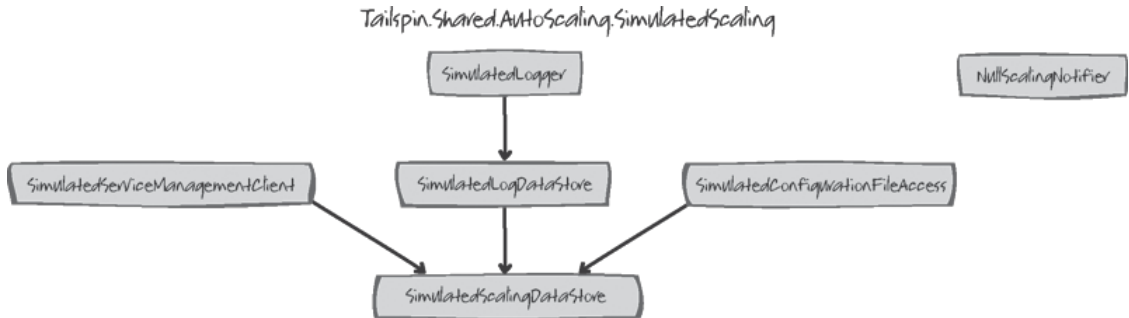
Кроме того, API управления службами Windows Azure не поддерживается эмулятором вычислений. Таким образом, программный блок автоматического масштабирования может быть назначен только для размещенных служб, развернутых в Windows Azure.

Проект **Tailspin.Autoscaling.Cloud** можно запустить локально в режиме отладки, но он должен использовать хранилище Windows Azure для хранения точек данных, а конечное приложение должно размещаться в Windows Azure.

## Запуск приложения управления в режиме эмуляции

При необходимости можно запустить приложение управления в режиме эмуляции. Это позволяет исследовать приложение управления без развертывания полного приложения Tailspin Surveys в Windows Azure. Этот экспериментальный режим позволяет сосредоточиться на взаимодействии с программным блоком автоматического масштабирования, а не заниматься вопросами, связанными с Windows Azure.

Данная реализация заменяет некоторые классы путем реализации интерфейсов, к которым обращается блок, как показано на следующем рисунке:

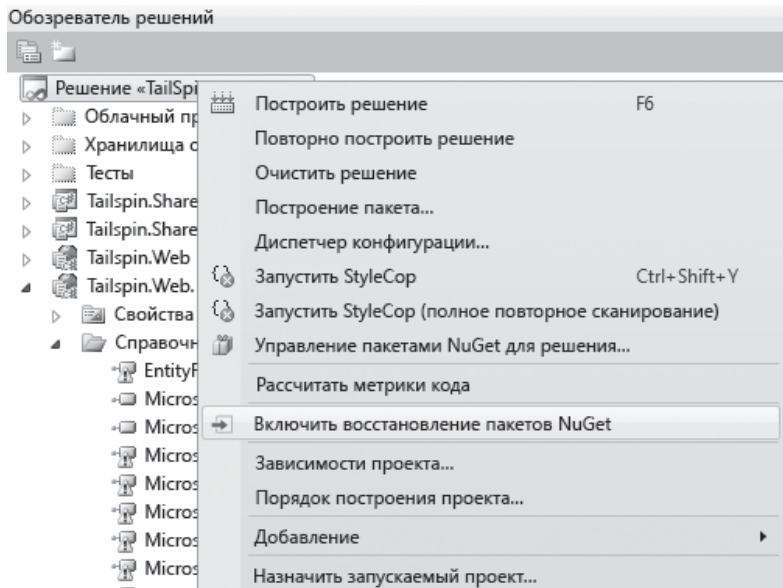


Решение использует хранилище сеанса, которое не сохраняется в нескольких различных сеансах или при запуске в режиме отладки.

### РАБОТА В РЕЖИМЕ ЭМУЛЯЦИИ

1. Запустите среду Visual Studio от имени администратора.
2. Откройте решение Tailspin.Simulated.sln.
3. В обозревателе решений щелкните правой кнопкой мыши решение Tailspin.Simulated, выберите **Включить восстановление пакета NuGet** и нажмите кнопку **Да** в диалоговом окне подтверждения.





**Примечание.** Если используется NuGet версии 1.5 или более ранней версии, то элемент меню **Включить восстановление пакета NuGet** не будет виден. В этом случае введите следующие команды в консоли диспетчера пакетов NuGet. Открыть консоль диспетчера пакетов NuGet можно в меню Visual Studio. Укажите пункт **Сервис**, затем **Диспетчер пакетов библиотек**, выберите пункт **Консоль диспетчера пакетов** и введите следующие команды:

- Install-Package NuGetPowerTools;
  - Enable-PackageRestore.
4. Установите проект **Tailspin.Management.Simulated.Cloud** как начальный проект.
  5. Нажмите клавишу «F5».
  6. Так как в этом решении используется тестовый сертификат, в окне браузера появится следующее сообщение: «Продолжить для этого веб-сайта (не рекомендуется)». Щелкните эту ссылку, чтобы продолжить.
  7. В левом нижнем углу окна браузера должно появиться сообщение «Режим эмуляции масштабирования активен в настоящее время».

## Известные проблемы

Известны следующие проблемы приложения Tailspin Surveys.

- Если XML-файлы хранилища правил и служебной информации не соответствуют включенным схемам, то приложение выдает исключение и перестает работать.
- Из-за *ограничений* в Visual Studio 2010 IntelliTrace вызывается следующее исключение времени выполнения, если программный блок размещается в Windows Azure с включенной функцией IntelliTrace:

`System.Security.VerificationException`:  
операция может дестабилизировать среду выполнения.

**Способ устранения проблемы.** Для решения этой проблемы необходимо отключить `IntelliTrace` в сборке программного блока автоматического масштабирования.

Для этого добавьте `Microsoft.Practices.EnterpriseLibrary.Windows Azure.*` в список исключенных модулей. См. инструкции здесь: `VerificationException` от `Windows Azure IntelliTrace`.

\* В данный момент проверка поля ввода является неполной.

## Дополнительные сведения

Перейдите на домашнюю страницу пакета интеграции корпоративной библиотеки для `Windows Azure`, чтобы прочитать последние новости:

<http://entlib.codeplex.com/wikipage?title=EntLib5Azure>

Материалы разделов, не представленных в данном документе, см. в документах по установке предыдущих выпусков:

<http://wag.codeplex.com/releases/view/71446>

Дополнительные сведения о программном блоке автоматического масштабирования см. в главе 4, «Автоматическое масштабирование и `Windows Azure`», в данном руководстве и в статье «Программный блок автоматического масштабирования» на сайте MSDN: [http://msdn.microsoft.com/ru-ru/library/hh680945\(v=PandP.50\).aspx](http://msdn.microsoft.com/ru-ru/library/hh680945(v=PandP.50).aspx)

Видеоруководство по программному блоку автоматического масштабирования доступно в статье «Автоматическое масштабирование приложений `Windows Azure`» на Channel 9:

<http://channel9.msdn.com/posts/Autoscaling-Windows-Azure-applications>

Дополнительные сведения о программном блоке обработки временных ошибок см. в главе 6, «Обработка временных ошибок», в данном руководстве и в статье «Программный блок обработки временных ошибок» на сайте MSDN:

[http://msdn.microsoft.com/ru-ru/library/hh680934\(v=PandP.50\).aspx](http://msdn.microsoft.com/ru-ru/library/hh680934(v=PandP.50).aspx)

Диспетчер пакетов `NuGet` можно загрузить на сайте сообщества `NuGet` на портале `CodePlex`:

<http://nuget.codeplex.com/>

Дополнительные сведения о создании профиля `Windows Azure` см. на портале `Windows Azure`:

<http://go.microsoft.com/fwlink/?LinkId=229563>.

Дополнительные сведения об ограничении в `Visual Studio 2010 IntelliTrace` и отключении `IntelliTrace` в сборке см. в статье «`RIA`, `Azure` и `IntelliTrace`» в блоге Кайла Мак-Клеллана (Kyle McClellan):

<http://blogs.msdn.com/b/kylemc/archive/2010/06/09/ria-azure-and-intellitrace.aspx>

Чтобы получить простой доступ к ресурсам в сети, используйте электронную версию библиографии на MSDN:

[http://msdn.microsoft.com/ru-ru/library/hh749032\(v=PandP.50\).aspx](http://msdn.microsoft.com/ru-ru/library/hh749032(v=PandP.50).aspx)

# Приложение В

## Глоссарий

**WASABiCmdlets.** Набор командлетов Windows PowerShell®, которые могут быть использованы для управления программным блоком автоматического масштабирования.

**Автоматическое масштабирование.** Использование автоматического механизма масштабирования приложения для платформы Windows Azure™.

**Аренда большого двоичного объекта.** Механизм в Windows Azure, который гарантирует, что только один клиент может получить доступ к большому двоичному объекту.

**Вертикальная масштабируемость** (в контексте программного блока автоматического масштабирования). Возможность масштабирования приложения за счет увеличения размера роли экземпляра при использовании большего числа ядер ЦП и большего объема памяти.

**Временная ошибка.** Ошибка, вызванная некоторым временным состоянием. Например, временная ошибка может возникнуть при использовании такой облачной размещенной службы, как хранилище Windows Azure или платформа SQL Azure™, а также из-за потери соединения по причине временной нехватки ресурсов. Результатом является ошибка, но, если повторить ту же команду через небольшой период времени, она может быть выполнена успешно, так как соединение будет восстановлено.

**Всплеск.** Резкое повышение рабочей нагрузки для приложения Windows Azure.

**Горизонтальная масштабируемость** (в контексте программного блока автоматического масштабирования). Способность приложения к масштабированию за счет добавления дополнительных экземпляров роли в размещенную службу.

**Группа масштабирования.** Способ определения правил автоматического масштабирования, которые могут действовать на несколько ролей одновременно. Группы масштабирования помогают свести

к минимуму количество правил, которые нужно создать и которыми нужно управлять. Они могут включать роли в различных размещенных службах.

**Конечный объект** (в контексте программного блока автоматического масштабирования). Определяет тип веб-роли или рабочей роли, у которого может быть несколько запущенных экземпляров и который можно масштабировать. Операции автоматического масштабирования могут указывать изменения числа экземпляров конечного объекта при выполнении операции правилом автоматического масштабирования. Конечные объекты обычно ссылаются на роли в размещенной службе, отличной от службы, в которой размещается программный блок автоматического масштабирования.

**Операнд**. Определяет способ вычисления значения показателя, который может использоваться в выражении правила реагирования. Например, можно создать операнд счетчика производительности, который отслеживает использование ЦП для рабочей роли и вычисляет среднее значение за период в 10 минут. Операнды используются правилами реагирования.

**Оптимизация расходов** (в контексте программного блока автоматического масштабирования). Способ, который позволяет убедиться, что запущенные экземпляры роли используются оптимально, так как запускаются в начале часа и останавливаются в конце часа.

**Период заморозки** (в контексте программного блока автоматического масштабирования). Период времени сразу после операции развертывания или изменения конфигурации службы приложения Windows Azure, когда Windows Azure запрещает любые дополнительные изменения конфигурации. Продолжительность периода заморозки определяется Windows Azure, и его нельзя изменить. Продолжительность периода заморозки обычно составляет несколько минут, но это величина непостоянная.

**Период снижения активности** (в контексте программного блока автоматического масштабирования). Период времени после применения масштабирования для конкретного приложения Windows Azure, в ходе которого не должны выполняться другие операции автоматического масштабирования. Это позволяет отрегулировать приложение после операции масштабирования и помогает уменьшить количество выполняемых действий масштабирования. Вы можете настроить значение периода снижения активности независимо для операции горизонтального и вертикального масштабирования. Значение периода снижения активности по умолчанию равно 20 минутам.

**Показатель**. Измеряемый параметр. В качестве примера можно привести счетчики производительности, например счетчики использования ЦП и свободной памяти, а также показатели, связанные с бизнес-данными, например число необработанных

заказов и количество зарегистрированных клиентов в приложении. Показатели также могут определяться результатом вычислений, например длина очереди в расчете на экземпляр, когда несколько экземпляров роли используют одну очередь совместно.

**Правила повтора** (в контексте программного блока обработки временных ошибок). Сочетание стратегии обнаружения ошибок и стратегии выполнения повторных попыток.

**Правила ограничения.** Правила, которые задают явные границы процесса масштабирования, определяя минимальное и максимальное количество экземпляров, разрешенных в течение данного периода времени. Для обеспечения выполнения соглашений об уровне обслуживания (SLA) необходимо задать минимальное значение. Необходимо установить максимальное значение, чтобы ограничить расходы. Правила ограничения состоят из максимального и минимального числа экземпляров, рейтинга и дополнительного расписания, которое определяет время применения правила. Если расписания нет, правило действует всегда.

**Правила реагирования.** Правила реагирования на изменяющуюся нагрузку в приложении, по которым вызывается операция масштабирования, если агрегатное значение, полученное из набора точек данных, превышает определенное пороговое значение.

**Ранг.** Свойство правила, которое используется блоком для разрешения конфликтов между правилами. Чем выше рейтинг, тем выше приоритет.

**Расписание** (в контексте программного блока автоматического масштабирования). Определяет, когда правило ограничения является активным. Если у правила ограничения нет расписания, то правило всегда будет включено.

**Регулирование приложения.** Процесс сокращения потребностей приложения в ресурсах, обычно для поддержания производительности основных функций в ответ на изменения рабочей нагрузки. К примерам можно отнести отключение ненужных функций или переход на облегченную версию пользовательского интерфейса.

**Роль.** Определение службы для развертывания кода приложения в Windows Azure. Приложение Windows Azure может состоять из множества веб-ролей и рабочих ролей.

**Сертификат API управления** (в контексте Windows Azure). Сертификат, используемый для защиты вызовов API службы управления Windows Azure.

**Сертификат службы** (в контексте Windows Azure). Сертификат, который может использоваться приложением, работающим в Windows Azure, для шифрования и расшифровки данных.

**Служебная информация.** Определяет аспекты приложения Windows Azure, которые являются релевантными для программного блока автоматического масштабирования. Например, программный блок автоматического масштабирования с помощью файла служебной информации может определить, какие из ролей доступны для масштабирования и какие из очередей доступны для мониторинга.

**Стабилизация** (в контексте программного блока автоматического масштабирования). Запрет операций масштабирования для предотвращения ненужных циклических изменений числа экземпляров роли из-за автоматического масштабирования.

**Стратегия обнаружения** (в контексте программного блока обработки временных ошибок). Определение логики, используемой для идентификации временных ошибок в службе.

**Стратегия повтора** (в контексте программного блока обработки временных ошибок). Определение числа повторных попыток и интервала между попытками.

**Точка данных.** Мгновенное значение со связанной отметкой времени. В следующей таблице показано несколько примеров точек данных.

Показатель	Значение	Отметка времени
Загрузка ЦП	83.7%	10:03:56
Необработанные заказы	2873	10:04:13

**Экземпляр роли.** Экземпляр веб-роли или рабочей роли в Windows Azure. У отдельной веб-роли или рабочей роли может быть несколько запущенных экземпляров, чтобы сделать роль более надежной или способной обрабатывать большие рабочие нагрузки.

**Эластичность.** Способность приложения к автоматическому масштабированию в соответствии с изменяющимися требованиями рабочей нагрузки.