

ФИШИНГ, СМИШИНГ, ВИШИНГ

# Windows IT Pro/RE

№10 ОКТЯБРЬ 2017 | WWW.WINDOWSITPRO.RU | ИНФО ДЛЯ ИТ-ПРО

Мобильная  
версия



ИДЕТ  
РЕКОНСТРУКЦИЯ

# Обновление до SQL Server 2016

ISSN 1563-101X



9 771563 101008

# МИР ЦОД

[2017]

Сервисы. Облака.

11 октября

ЦОД для облака, облачные сервисы, услуги КЦОД.  
В центре внимания – новые тенденции использования облачных  
сред с учетом перспектив перехода к цифровой экономике.

## Основные темы форума

- ◆ Развитие услуг и сервисов КЦОД
- ◆ Инфраструктура
- ◆ Территориально распределенные облака
- ◆ Безопасность
- ◆ Облака для разработчиков
- ◆ Децентрализация ЦОД

### Золотые партнеры



### Партнеры



Цена участия  
до 10 сентября  
**5940 руб.**



Реклама  
12+

Цена участия с 11 сентября

**9900 руб.**

По вопросам участия: Ольга Пуркина



+7 (499) 703-1854, +7 (495) 725-4780



kon@osp.ru

Издание для специалистов, интересующихся технологиями компаний Microsoft.

Главный редактор: Д. Ю. Торопов (totoropov@osp.ru)

Ответственный редактор: Е. Петровичева

Корректор: Л. Теременко

Верстка и дизайн: О. Шуранова

Номер также готовили: Е. Овсянников

Т. Евдокимова, А. Китаев, А. Федотов,

Н. Басалова, Ю. Власов, Д. Щепкин, А. Адзиев

Адрес для писем: 123056, Москва, а/я 82

Телефоны: (495) 725-4780/83, (499) 703-1854

Факс: (495) 725-4783

E-mail: windowsitpro@osp.ru

© 1999-2017 Издательство «Открытые системы»

© 1999-2017 Penton Media, Inc.

Журнал зарегистрирован в Роскомнадзоре.

Свидетельство о регистрации средства массовой информации ПИ № ФС 77-63737 от 16 ноября 2015 г.

Дата выхода в свет — 13.10.2017 г.

Цена свободная. Выходит 12 раз в год.



**ОТКРЫТИЕ  
СИСТЕМЫ**

Open Systems Publications

#### Учредитель и издатель:

ООО «Издательство «Открытые системы»  
127254, Москва, пр-д Добролюбова, д. 3,  
стр. 3, каб. 13.

Президент М. Е. Борисов

Генеральный директор Г. А. Герасина

Директор ИТ-направления П. В. Христов

Коммерческий директор Т. Н. Филина

#### Подписные индексы:

Объединенный каталог «Пресса России» — 38185,  
«Каталог российской прессы» — 99483,  
ФГУП «Почта России» — П2337

Отпечатано в ООО «Богородский полиграфический комбинат»,  
142400, Московская обл., г. Ногинск,  
ул. Индустриальная, д. 406

Тираж: 6900 экз. — печатная версия,  
3280 экз. — PDF-версия

Редакция не несет ответственности за содержание рекламных материалов. Все права защищены. Полное или частичное воспроизведение или размножение каким бы то ни было способом материалов, опубликованных в настоящем издании, допускается только с письменного разрешения ООО «Издательство «Открытые системы».

Windows®, Windows Vista® и Windows Server® — зарегистрированные торговые марки корпорации Microsoft. Название Windows IT Pro используется Penton Media, Inc. в соответствии с соглашением с владельцем торговой марки. Название Windows IT Pro/RE используется ООО «Издательство «Открытые системы» по лицензионному соглашению с Penton Media, Inc. Windows IT Pro/RE — независимое от корпорации Microsoft издание. Корпорация Microsoft не несет ответственности за редакционную политику и содержание журнала. Редакция оставляет за собой право не вступать в переписку.

Отобранные для публикации письма редактируются в соответствии с терминологическими нормами, принятыми в издательстве.

Названия продуктов и компаний, упомянутых в журнале, могут быть товарными знаками их владельцев.



Penton Media, Inc.

#### ИТ И БИЗНЕС

## 2 Знакомимся с Windows 10 S

РИЧАРД ХЭЙ

## 4 Уборка в цифровом пространстве

ЛИЗА ШМАЙСЕР

#### ТЕМА НОМЕРА

## 6 Переходим на SQL Server 2016. Часть 1

ТОМАС ЛА РОК

## 10 Переходим на SQL Server 2016. Часть 2

ТОМАС ЛА РОК

## 12 Переходим на SQL Server 2016. Часть 3

ТОМАС ЛА РОК

#### SQL SERVER

## 14 Запрос функции T-SQL

ИЦИК БЕН-ГАН

## 20 Клонирование объектов безопасности между серверами

ТИМ ФОРД

#### ПИШЕМ СЦЕНАРИИ

## 26 Своя команда

СЕРГЕЙ ВАСИН

#### УПРАВЛЕНИЕ

## 32 Один ответ на два вопроса

СЕРГЕЙ ВАСИН

#### ОБНОВЛЕНИЯ

## 38 Коммуникационные сайты

ЛИАМ КЛИРИ

## 40 Сборки для участников программы тестирования

Windows Insider

РИЧАРД ХЭЙ

## 42 Средства очистки жесткого диска в Windows 10 Redstone 3

РИЧАРД ХЭЙ

## 44 Функция OneDrive Files On-Demand

РИЧАРД ХЭЙ

#### OFFICE SYSTEM

## 47 Microsoft Forms: практические занятия

ЛИАМ КЛИРИ

#### ВВОДНЫЙ КУРС

## 50 Учимся бороться с шифровальщиками

ВЛАДИМИР БЕЗМАЛЫЙ

## 55 Что знает о вас «облако»

ВЛАДИМИР БЕЗМАЛЫЙ

## 59 Фишинг, смишинг, вишинг

ВЛАДИМИР БЕЗМАЛЫЙ

## 63 Эффективная работа с OneNote

РИЧАРД ХЭЙ

# Знакомимся с Windows 10 S

**Н**а недавнем мероприятии Microsoft, ориентированном на образовательный сектор, представление нового члена семейства Windows 10 стало связующим звеном между анонсами оборудования и служб. Появление операционной системы Windows 10 S обещает стать важным событием. Новая операционная система установлена на мобильном компьютере Surface Laptop, совместима с приложениями Intune for Education и Set UP School PCs и будет работать на некоторых бюджетных OEM-устройствах таких производителей, как HP, Acer, Asus, Dell, Toshiba, Samsung и Fujitsu.

Приобретая устройство с Windows 10 S, студенты и преподаватели получат следующие приятные дополнения:

- бесплатную годовую подписку на Minecraft: Education Edition;
- бесплатный Office 365 for Education с Microsoft Teams;
- бесплатное обновление до Windows 10 S для школ на уже имеющихся устройствах Windows Pro;
- бесплатную пробную версию Microsoft Intune for Education.

Преимущества распространяются и на покупателей ноутбуков Surface Laptop с Windows 10 S. Они будут иметь право на бесплатную годовую подписку на Office 365 Personal и бесплатное же обновление до Windows 10 Pro до 31 декабря 2017 года.

Ключевая особенность Windows 10 S состоит в том, что эта операционная система допускает использование исключительно приложений из магазина Windows. Для тех, кто захочет установить Chrome, iTunes или другое приложение рабочего стола, будет два возможных пути: надеяться, что это приложение есть в Магазине Windows (что вполне вероятно, если разработчик возьмет на себя труд преобразовать его в приложение, пригодное для размещения в магазине, с помощью Desktop App Converter), либо выполнить обновление до Windows 10 Pro. Последнее можно сделать за 50% обычной цены обновления, составляющей 99 долл.

Неудивительно, что первые комментарии пользователей отражают взгляд на Windows 10 S как на надувательство, ведь за возможность пользоваться привычными приложениями для рабочего стола придется заплатить 50 долл., чтобы обновиться до Windows 10 Pro. В других

комментариях отмечается, что Windows 10 S предусматривает плату в размере 50 долл. за установку Chrome, iTunes и прочих нужных приложений рабочего стола. Чтобы лучше понять позицию разработчиков Windows 10 S, посмотрим на результаты сравнения набора компонентов этой операционной системы с возможностями двух других потребительских версий Windows 10 — Windows 10 Home и Pro. Сравнение проводилось на основе ответов на часто задаваемые вопросы по Windows 10 S и данных, приведенных на странице описания продукта в Магазине Windows (см. таблицу).

Как можно заметить, у Windows 10 S больше общих компонентов с Windows 10 Pro, чем с Home. Полагаю, это указывает на то, что Windows 10 S в конечном счете станет потребительской версией Windows 10, заменив Home.

При такой замене опытные пользователи получат гибкие возможности доступа к дополнительным средствам, обычно доступным только для пользователей Pro, и окажутся в шаге от возможности пользоваться приложениями рабочего стола после обновления до Windows 10 Pro за 50 долл. Таким образом, обновление обойдется на 50% дешевле, чем сегодня для пользователей Home (99 долл. в магазине Windows).

После более близкого знакомства с Windows 10 S мне предстоит исследовать еще одно важное преимущество для опытных пользователей: доступ к реализованной в Windows 10 Pro Creators Update возможности отложить получение обновлений компонентов или приостановить обновления Windows.

Для обычных пользователей также преимущества очевидны. Защищенная Windows 10 S — это шаг в направлении реализации особого типа безопасности, когда внешние угрозы полностью исключены, ведь пользоваться можно только приложениями из магазина Windows. Браузером по умолчанию является защищенный Microsoft Edge. В качестве поисковой системы по умолчанию используется Bing. Уже сегодня система Windows 10 Home с браузером Edge защищена гораздо лучше, чем системы с другими браузерами.

Замечу, что многие из тех, кто читает эту статью, не относятся к «повседневным» пользователям, но обеспечивают поддержку пользователей такого типа, например членов семей или своих друзей. Основываясь на личном опыте, позволившем мне убедиться в том, насколько Windows 10 и Edge сократили частоту обращений в службу поддержки по одному конкретному

Таблица. Сравнение Windows 10 S с Windows 10 Home и Pro

Компонент Windows 10	Home	S	Pro
Microsoft Edge	Да	Да	Да
Приложения Магазина Windows	Да	Да	Да
Несколько рабочих столов и функция Snap	Да	Да	Да
OneDrive	Да	Да	Да
Управление мобильными устройствами (MDM)	Ограничено	Да	Да
Установка и запуск приложений рабочего стола	Да	Нет	Да
Обновление до Windows 10 Pro через Магазин	Да (99 долл.)	Да (49 долл.)	Нет сведений
Присоединение к локальному домену	Нет	Нет	Да
Удаленный рабочий стол	Нет	Да	Да
Hyper-V	Нет	Да	Да
Магазин Microsoft для бизнеса и образования	Нет	Да	Да
Центр обновления Windows для бизнеса	Нет	Да	Да
Присоединение к Azure Active Directory/домену	Нет	Да	Да
Enterprise State Roaming в Azure AD	Нет	Да	Да
Общая конфигурация ПК	Нет	Да	Да
Браузер по умолчанию	Настраивается	Edge (фикс.)	Настраивается
Поисковая система по умолчанию	Настраивается	Bing (фикс.)	Настраивается

поводу, я могу сделать вывод о лучшей защищенности всей совокупности устройств Windows.

На некоторых сайтах Windows 10 S позиционируется как облегченная версия операционной системы, которая должна хорошо работать на OEM-устройствах младших моделей. Это недалеко от реальности, ведь достаточно взглянуть на приведенную таблицу, чтобы увидеть, что у Windows 10 S возможности шире, чем у Windows 10 Home. Я ознакомился с некоторыми OEM-устройствами, которые демонстрировались на мероприятии #MicrosoftEDU. Все эти устройства, которые создавались для Windows 10 и не были оптимизированы для Windows 10 S, очень хорошо работали в рамках тех сценариев использования, для которых предназначались, то есть при посещении веб-сайтов, обработке текстов и использовании облегченных версий игр.

Что касается перехода с Windows 10 S на Windows 10 Pro, следует также иметь в виду, что это обновление — «поездка в один конец», как сказано в заявлении Microsoft. Пока не вполне ясно, означает ли это, что вы выполняете чистую установку Pro на устройстве и имеете цифровую привязку лицензии, позволяющую перейти на Pro-версию Windows 10, или же вам предстоит вернуться к Pro путем переустановки Windows 10 S. Учитывая подход к обновлениям и цифровым привязкам, реализованный в Windows 10 и используемый с момента выпуска этой операционной системы, я полагаю, что все, что потребуется, — это чистая установка обновленной версии операционной системы. Разумеется, возврат в исходное состояние через модуль «Параметры Windows» позволит восстановить на устройстве текущую установленную подлинную операционную систему.

Далее я планирую продолжить сравнительное исследование функциональных возможностей

Windows 10 S, таких как упомянутые ранее блокировка обновлений Windows и будущие обновления компонентов.



Ричард Хэй (winobs@outlook.com) имеет звание Microsoft MVP в категории Windows Operating System с 2010 года

# DeviceLock® DLP

**20 лет**

**70 000 клиентов**

**7 000 000 инсталляций**



[www.smartline.ru](http://www.smartline.ru)

Реклама

# Уборка в цифровом пространстве

**М**ногие из нас хорошо помнят затевавшиеся в конце учебного года «ритуальные» уборки с расчисткой бумажных завалов в недрах письменных столов. В июне, когда заканчивалась учеба и начинались каникулы, что-то неизменно побуждало нас к наведению порядка. Лишь немногие, став взрослыми, по-прежнему каждый год в июне разбирают свой рабочий стол, но беспорядок есть у всех: файлы, иногда загруженные для однократного использования и никогда больше не пригодившиеся; сообщения, прочитанные и не удаленные; документы PDF со сведениями для налоговой отчетности и т.д. Однако хранилища стоят недорого и обеспечить их базовую безопасность очень просто. Зачем же затрудняться наведением порядка в своем цифровом пространстве? Вот лишь несколько причин:

1. Чем меньше у вас файлов с персональной информацией, тем меньше риск, что кто-то сможет получить к ней доступ для кражи личных данных, взлома или других неблаговидных целей.
2. Ликвидация старых учетных записей, которые больше не нужны, также снижает вероятность того, что кто-то начнет пользоваться вашим забытым персональным хранилищем Dropbox, старым почтовым ящиком на Yahoo.com и т. д.
3. Чем скромнее ваше «цифровое достояние», тем проще вам будет за ним следить и обеспечивать его безопасность. Здесь работает тот же принцип, что

и в обычной жизни: чем больше у вас вещей, тем больше энергии они у вас забирают.

Как же проще всего навести порядок в цифровом пространстве? Опишу методы, которые представляются мне полезными.

Разбейте уборку на отдельные задачи. Это избавит вас от ощущения невыполнимости дела и поможет взять верный темп. Например, сегодня вы потратите час на просмотр загруженных файлов и папок с документами на рабочем столе под аккомпанемент любимой музыки, которая так скрашивает вам обычную уборку, а на другой день проанализируете свою электронную почту и решите, от каких рассылок можно отписаться, а что оставить и т.д. Нормальная цифровая уборка занимает примерно неделю, при условии что задачи продуманно разбиты на этапы.

Найдите и удалите ненужные учетные записи. Помимо старых почтовых ящиков, учетных записей «Яндекса» (например, личного хранилища «Яндекс.Диск», в которое вы не заглядывали уже несколько лет) и редко используемых интернет-хранилищ, удалите свои личные кабинеты на веб-сайтах, где однажды что-то покупали.

Как найти эти учетные записи? В своей почте я выполняю поиск по таким фразам, как «обновите вашу учетную запись», «управляйте своей учетной записью», «данные учетной записи», «обновите пароль», «управляйте предпочтениями», «ваш заказ» и «отписаться», а затем нахожу домены и сайты, которые давно не посе-



щались. Особенно тщательно я анализирую диспетчер паролей на наличие непосещаемых веб-сайтов, для которых все еще сохраняются учетные данные.

Составив список учетных записей, я нахожу время, чтобы сразу удалить те из них, для которых предусмотрены ссылки для удаления, либо ввожу поисковый запрос «как удалить учетную запись» у конкретного поставщика. Существует очень полезный ресурс, который может помочь в решении проблем удаления, — <http://backgroundchecks.org/justdeleteme/>. Дважды проверьте настройки браузера и диспетчера паролей. Это позволит исключить накопление вышедших из употребления или устаревших данных.

В календаре Outlook зарезервируйте несколько часов, когда вас никто не побеспокоит, и проанализируйте свою почту. Мне всегда нравился подход Inbox Zero (ни одного письма в папке «Входящие»), который заключается в том, чтобы прочитать сообщение, ответить на него, после чего переместить его в папку «Архив», если оно содержит полезные сведения, либо удалить, если оно бесполезно.

Единственная проблема такого подхода заключается в том, что со временем архивная папка разрастается до огромных размеров. Время от времени полезно просматривать ее для выявления переписки, относящейся к давним проектам, которую уже можно отправить в корзину. Впрочем, вы, возможно, предпочтете оставлять свою почту в папке для входящих сообщений. У всех есть привычные схемы действий, которые для каждого из нас работают лучше прочих. Почту можно отсортировать по отправителю, теме или дате, чтобы увидеть, от чего пришло время избавиться.

Просматривая папку входящих сообщений, отпишитесь от всех рассылок и автоматически удалите их. Если для вас очевидно, что вы не станете открывать данное письмо, откажитесь от подписки.

Есть ли у вас флеш-накопители, старые жесткие диски или компьютеры, ожидающие очистки? Очистите и утилизируйте все лишнее. Если вы никогда не очищали диск, предназначенный для утилизации, вот последовательность действий: отмените разрешения для всех приложений, имеющих доступ к службам потоковой передачи или к вашим цифровым ресурсам; отмените разрешения для всех сторонних приложений, установленных на вашем устройстве; если речь идет о жестком диске, воспользуйтесь какой-нибудь утилитой для очистки жестких дисков. У многих поставщиков есть бесплатные демоверсии утилиты для очистки твердотельных накопителей (SSD). Чтобы очистить USB-накопитель, достаточно подключить его к компьютеру (при этом желательно получить хотя бы приблизительное представление о том, что на нем находится, либо на всякий случай запастись резервной копией текущего состояния системы), после чего удалить файлы и переформатировать накопитель.

Что касается избавления от лишнего оборудования, то не составит проблемы найти утилизатора техники,

благотворительную программу или торгового посредника, который возьмет ваши вещи.

Уберите ненужные приложения с мобильных устройств и компьютеров. Если вы не пользуетесь приложением, не поленитесь его открыть, чтобы удалить устаревшие данные и закрыть свою учетную запись у разработчика, а затем удалите приложение. Помните, что закрытие учетной записи не гарантирует того, что компания удалит все ваши данные, но это помешает компании собирать о вас новые сведения. Кроме того, это уменьшит число накопившихся у вас действующих учетных записей. В результате риск тоже снизится.

Пересмотрите загруженные файлы, документы и содержимое рабочего стола. Всем нам знакома кошмарная картинка с тысячами файлов на рабочем столе. Многие не осознают, что каждый ярлык файла, отображаемый на рабочем столе, использует оперативную память и замедляет работу компьютера. Поэтому наведите порядок на рабочем столе! Просмотрите папку для загружаемых файлов и удалите все ненужное. Перейдите в общую папку с документами и подумайте, не перенести ли что-нибудь в более подходящее место, например OneNote или Evernote. Пересмотрите «облачные» службы автоматизации, такие как Microsoft Flow или If This Then That. Автоматизация удобна, но никогда не помешает взвесить, так ли уж необходимо собирать все данные о своем статусе в Facebook в электронную таблицу или сохранять каждый твит в OneNote.

Просмотрите «облачные» диски и цифровые репозитории. Чем больше существует мест, в которых вы держите свою информацию, тем большему риску вы подвергаетесь. Такими службами, как Dropbox или «Яндекс Диск», удобно пользоваться для размещения файлов, предназначенных для коллективного доступа. Просмотрите то, что вы храните в таких местах, и решите, действительно ли стоит это хранить.

На этот последний шаг часто уходит больше всего времени, особенно если вам придется просматривать содержимое OneNote или Evernote для удаления сведений, связанных с ненужными учетными записями, или гостиничных квитанций со временем давно прошедших отпусков.

Подумайте о том, как упростить следующую уборку. Это одно из дел, которые труднее всего выполнять впервые. Задача состоит в том, чтобы в следующий раз воспринимать его лишь как небольшое неудобство. Если это вам поможет, то запрограммируйте в своем календаре ежемесячные напоминания о проведении проверки различных «облачных» сервисов. Возможно, вам захочется наметить день для ежегодной отписки от рассылок по электронной почте. Просто заведите привычку совершать небольшие шаги регулярно, чтобы поддерживать свое «цифровое достояние» в порядке и безопасности. ◇

Лиза Шмайсер ([Lisa.Schmeiser@penton.com](mailto:Lisa.Schmeiser@penton.com)) — главный редактор сайта SuperSite for Windows

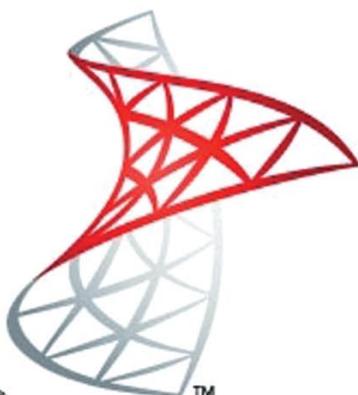
# Переходим на SQL Server

2016

**Томас Ла Рок**

Как известно, обновление сервера базы данных сопряжено с большой неопределенностью. Данные и базы данных — наиболее ценные ресурсы компании, поэтому легко понять, отчего мы всегда проявляем повышенную осторожность при модернизации.

Эта публикация открывает цикл статей, которые вместе составят руководство по переходу на Microsoft SQL Server 2016. Однако помните, что не существует двух одинаковых процессов модернизации. То, что кто-то потерпел неудачу (или, напротив, добился успеха) не означает, что вы получите такой же результат.



Microsoft®  
**SQL Server®**

## Зачем обновлять

«Зачем обновлять?» — первый вопрос, который приходится слышать на начальном этапе каждого проекта обновления. Кто-то всегда хочет понять, зачем нужно вносить изменения в исправную и безупречно функционирующую систему.

Ниже приводится краткий список причин, побуждающих пользователей к переходу на SQL Server 2016.

### 1. Новые функции SQL Server 2016

В любой новой версии SQL Server всегда присутствует какое-нибудь свойство, с которым интересно экспериментировать. В SQL Server 2016 реализованы следующие новые функции:

- постоянное шифрование Always Encrypted (<https://msdn.microsoft.com/en-us/library/mt163865.aspx>);
- динамическое маскирование данных Dynamic Data Masking (<https://msdn.microsoft.com/en-us/library/mt130841.aspx>);
- защита на уровне строки Row Level Security (<https://msdn.microsoft.com/en-us/library/dn765131.aspx>);
- расширяемая база данных Stretch Database (<https://msdn.microsoft.com/en-us/library/dn935011.aspx>);
- временные таблицы (<https://msdn.microsoft.com/en-us/library/dn935015.aspx>);
- автоматическая поддержка программной реализации технологии NUMA (<https://blogs.msdn.microsoft.com/psssql/2016/03/30/sql-2016-it-just-runs-faster-automatic-soft-numa/>);
- хранение запросов (<https://msdn.microsoft.com/en-us/library/dn817826.aspx>).

Кроме того, усовершенствованы функции, появившиеся в предыдущих версиях:

- выполнение запросов OLTP в памяти In-Memory OLTP (<https://msdn.microsoft.com/en-us/library/dn133186.aspx>);
- группы постоянной доступности Always On (<https://msdn.microsoft.com/en-us/library/hh510230.aspx>);
- обновляемый некластеризованный столбчатый индекс columnstore (<https://msdn.microsoft.com/en-us/library/gg492088.aspx>);
- обновленная версия DBCC CHECKDB (<https://blogs.msdn.microsoft.com/psssql/2016/02/25/sql-2016-it-just-runs-faster-dbcc-scales-7x-better/>).

Полный список усовершенствованных компонентов SQL Server 2016 можно найти в статье SQL Server 2016: It Just Runs Faster (<https://thomaslarock.com/2016/06/sql-server-2016-just-runs-faster/>).

## **2. Условия поддержки**

Срок завершения поддержки прежних версий SQL Server приближается (<https://support.microsoft.com/en-us/lifecycle/search?alpha=SQL%2520Server>). Конечно, можно

приобрести расширенную поддержку, но ее стоимость велика. Действительно, компания Microsoft продлила период поддержки Windows Server 2008 и SQL Server 2008 (<http://www.computerworld.com/article/3150245/enterprise-applications/why-microsoft-added-6-years-to-windows-server-support.html>), но это не означает, что дальнейшее их использование целесообразно.

## **3. Требования поставщиков программного обеспечения**

В компании могут использоваться программные продукты стороннего поставщика со строгими требованиями к версии SQL Server. Конечно, возможны различные варианты — иногда требуются более новые версии, иногда, наоборот, старые.

## **4. Стандарты компании и отрасли**

Некоторые компании не допускают отставания при обновлении какого-либо программного продукта более чем на одну полную основную версию. Такое же требование действует и в ряде отраслей. Не забывайте также об аудиторах, у них есть собственное мнение на этот счет.

## **5. Масштабируемость**

За прошедшее десятилетие в ядро SQL Server было внесено много улучшений, направленных на решение проблемы масштабируемости. Выше перечислено несколько из них (columnstore, группы доступности, OLTP в памяти), но собственно ядро модернизировано и дополнено, в частности, новыми методами прогнозирования числа строк в запросе (<https://msdn.microsoft.com/en-us/library/dn600374.aspx>), чтобы можно было строить оптимальные планы на основе распределения ваших данных. Переход на SQL Server 2016 обеспечивает более широкие возможности масштабирования по сравнению с предыдущими версиями.

Итак, какие выводы мы можем сделать из всего изложенного? Если какой-либо из приведенных аргументов окажется достаточно убе-

дительным, чтобы вы задумались о необходимости обновления, то пора начинать готовить план проекта. Самый простой план состоит из трех шагов:

1. Задачи, связанные с подготовкой к обновлению.
2. Задачи обновления.
3. Задачи, выполняемые после обновления.

На первый взгляд все просто. Иногда это действительно так, особенно если вы найдете время, чтобы прочитать эту серию статей. В частности, в данной статье рассматриваются задачи, связанные с подготовкой к обновлению.

Прежде чем приступить к обновлению, придется немало потрудиться. Поверьте, дополнительные усилия на данном этапе помогут избежать хлопот впоследствии. Ниже приводится список действий, которые стоит выполнить перед переносом данных.

### **1) Составьте план.**

Возможен прямой переход к SQL Server 2016 со следующих версий:

- SQL Server 2014;
- SQL Server 2012 с пакетом обновления 1 (SP1);
- SQL Server 2008 R2 с пакетом обновления 2 (SP2);
- SQL Server 2008 с пакетом обновления 3 (SP3).

Пользователям SQL Server 2005 и более ранних версий следует перейти на промежуточную версию перед обновлением до SQL Server 2016.

### **2) Ознакомьтесь с изменениями в условиях лицензирования.**

Изучите изменения в условиях лицензирования, которые, возможно, произошли в SQL Server 2016 по сравнению с вашей текущей версией. В SQL Server 2012 лицензирование производится по ядрам, а не разъемам процессора. Однако в редакции SQL Server 2016 Standard Edition допускается также лицензирование типа Server + CAL. Вследствие перехода от разъема процессора к ядру варианты обновления «на месте» могут сопровождаться значительным повышением стоимости. Необходимо тщательно изучить особенности лицензиро-

# Обновление до SQL Server 2016

вания SQL Server 2016 (<https://www.microsoft.com/en-us/sql-server/sql-server-2016-pricing>).

Помните, что не существует двух одинаковых процессов модернизации. То, что кто-то потерпел неудачу (или, напротив, добился успеха) не означает, что вы получите такой же результат

Также достойно упоминания, что SQL Server 2016 с пакетом обновления 1 (SP1) располагает многими функциями (<https://blogs.msdn.microsoft.com/sqlreleaseservices/sql-server-2016-service-pack-1-sp1-released/>), ранее представлявшимися лишь в корпоративной редакции, в том числе группами доступности, сжатием данных, секционированием, индексом columnstore и т. д. Прежде чем начинать проект модернизации, оцените расходы на лицензирование и изучите список функций, доступных теперь в стандартной редакции продукта.

## 3) Рассмотрите все варианты.

Каким бы сложным ни было обновление, оно относится к одному из двух типов: на месте и параллельное.

При обновлении на месте вы обновляете текущий экземпляр SQL Server, запуская мастер установки. Его проще выполнить, но сложнее осуществить возврат к предыдущему состоянию.

Параллельные обновления заключаются в установке новой версии SQL Server в качестве нового экземпляра на существующем сервере (или новом сервере, причем это предпочтительный вариант, особенно для производственных сценариев) и переносе баз данных по мере необходимости. Свежая установка SQL Server позволяет лучше протестировать систему перед ее запуском в производство. Существует также метод последовательного обновления rolling upgrade. В таком случае используется функция высокой доступности, такая как зеркалирование, кластеризация или группы доступности. Вы обновляете вторичный узел, пере-

ключаетесь на него и продолжаете обновлять все узлы до тех пор, пока не будет обновлен основной узел,

ных, которые выиграют от новых функций SQL Server 2016. Кроме того, DMA обеспечит перенос данных.

а затем при необходимости выполняете обратное переключение.

В целом я сторонник последовательного обновления, но только когда оно необходимо для уменьшения времени простоев в соответствии с требованиями бизнеса. Если такие требования отсутствуют, я предполагаю параллельное обновление и перенос данных путем их восстановления из резервной копии на новом сервере. Но это мои личные предпочтения, которые не обязательно совпадают с вашими. Вам следует выбирать оптимальный вариант для своей компании.

## 4) Соберите данные инвентаризации.

Также полезно собрать информацию об экземплярах серверов и баз данных, затронутых обновлением. Существует много инструментов для сбора таких сведений, в частности Microsoft Assessment and Planning (MAP) Toolkit (<https://www.microsoft.com/en-gb/download/details.aspx?id=7826>) и SQL Power Doc (<https://sqlpowerdoc.codeplex.com/>), а также инструменты независимых компаний.

## 5) Познакомьтесь с мастером Data Migration Assistant.

В прошлом известный как помощник по обновлению Microsoft SQL Server, Data Migration Assistant (DMA), описанный в документе по адресу: [www.microsoft.com/en-us/download/details.aspx?id=53595](http://www.microsoft.com/en-us/download/details.aspx?id=53595), поможет обнаружить любые критические изменения, а также устаревшие функции. Вы можете воспользоваться DMA для поиска проблем, которые необходимо устранить перед переходом на другую версию SQL Server. DMA также может помочь идентифицировать части базы дан-

## 6) Узнайте об устаревших функциях.

С выходом каждой новой версии SQL Server некоторые функции отмечаются как устаревшие. Это не означает, что данные функции удалены. Устаревшие функции, возможно, будут удалены в следующей версии, и их не следует применять в новых разработках. Список устаревших функциональных возможностей компонента Database Engine для SQL Server 2016 опубликован в документе по адресу: [msdn.microsoft.com/en-us/library/ms144262.aspx](https://msdn.microsoft.com/en-us/library/ms144262.aspx).

## 7) Идентифицируйте функции, поддержка которых прекращена.

Начиная с версии SQL Server 2016 компания Microsoft начала публиковать список функций, поддержка которых прекращена. Это означает, что данные функции полностью удалены. Список опубликован в документе по адресу: [msdn.microsoft.com/en-us/library/ms144262.aspx](https://msdn.microsoft.com/en-us/library/ms144262.aspx). Следует ознакомиться с этим списком и убедиться, что ваши приложения не зависят от функций, которые после обновления исчезнут.

## 8) Узнайте о критических изменениях.

Известно ли вам, что Microsoft публикует список критических изменений (<https://msdn.microsoft.com/en-us/library/ms143179.aspx>) для каждой версии SQL Server? Что ж, теперь вы это знаете. Необходимо внимательно ознакомиться с ним. Хочется верить, что вы получите предупреждения о многих критических изменениях от DMA, но на самом деле DMA не успевает за электронной документацией. В электронной документации могут быть сведения, которых нет в проверочном списке

DMA, поэтому нужно ознакомиться с соответствующим разделом.

## **9) Получите информацию об изменениях поведения.**

В предыдущих версиях SQL Server был опубликован список изменений поведения для компонента Database Engine. Аналогично критическим изменениям, изменения поведения могут оказаться неблагоприятное влияние. Их полезно изучить, и они часто не отражаются в DMA, так как не обязательно ведут к проблемам; это просто изменения, которые потенциально могут стать причиной проблем.

Однако я не смог найти раздел электронной документации по SQL Server 2016 для компонента Database Engine. Есть страницы для Analysis Services, Integration Services и Reporting Services. Подозреваю, что такой материал для компонента Database Engine появится по мере распознавания проблем. Вы всегда можете просмотреть предыдущие версии на странице SQL Server 2012 (<https://msdn.microsoft.com/en-us/library/cc707785%2528v=sql.110%2529.aspx>) и воспользоваться ею как отправной точкой.

## **10) Прочтайте заметки об изменениях по отношению к предыдущей версии.**

Выделите несколько минут на то, чтобы прочитать заметки об изменениях по отношению к предыдущей версии (<https://docs.microsoft.com/en-us/sql/sql-server/sql-server-2016-release-notes>). Нет, они не такие интересные, как заметки о выпуске приложений для телефона, но все же могут быть полезны. Желательно иметь как можно более полную картину новой версии, если что-то будет работать не так, как предполагалось. Кроме того, в заметках об изменениях по отношению к предыдущей версии содержатся подробности, которых вы не найдете в других местах.

## **11) Выясните новые требования среды.**

На странице <https://msdn.microsoft.com/en-us/library/ms143506.aspx> приведены минимальные требования для установки SQL Server 2016. Если ваши серверы не соответствую-

ют этим требованиям, то, скорее всего, вы не собираетесь выполнять обновление в обозримом будущем. Но если вы готовитесь к обновлению, то, вероятно, пора обновить и аппаратные средства. Можно даже подумать о переходе в виртуальную среду (если это уже не сделано), что также сопряжено с определенными требованиями к аппаратным средствам.

Однако настоящая причина для обновления аппаратных средств — новые функциональные возможности. Предположим, вы обдумываете переход на SQL Server 2016, чтобы воспользоваться преимуществами технологии Hekaton (<https://msdn.microsoft.com/en-us/library/dn170449.aspx>). Учитывая, как много новшеств в SQL Server 2016, вам придется проделать дополнительную работу, чтобы определить необходимые аппаратные средства. В противном случае вам не удастся воспользоваться многими новыми функциями.

## **12) Определите базовые показатели производительности.**

Перед началом процесса обновления рекомендуется собрать базовые показатели производительности. Если этого не сделать, вы не сможете выяснить, как изменилась производительность после обновления. Каждая версия SQL Server уникальна, поэтому для разных пользователей из сферы бизнеса важны различные метрики производительности.

## **13) Сравните нагрузки от приложений.**

Вы можете использовать функцию распределенного воспроизведения (<https://msdn.microsoft.com/en-us/library/ff878183.aspx>), чтобы извлечь нагрузку от приложений с исходного сервера и воспроизвести ее на целевом сервере. Это поможет оценить результаты обновления SQL Server, сравнивая производительность приложений на двух системах. Распределенное воспроизведение наиболее полезно в случаях с высоким параллелизмом, когда нельзя достоверно моделировать нагрузку от приложений на одном клиенте. Database Experimentation Assistant (<https://www.microsoft.com/en-us/download/details.aspx? id=54090>) —

новый инструмент, на сегодня представленный ознакомительной технической версией. В нем используется распределенное воспроизведение наряду со службами R, чтобы представить пользователю возможность автоматического получения показателей тестирования при работе приложений. Статистический анализ нагрузок от приложений позволяет более уверенно выполнять переход к новым версиям SQL Server.

## **14) Протестируйте характеристики сервера.**

С помощью таких инструментов, как iPerf (<https://iperf.fr/>) и DskSpd (<https://github.com/Microsoft/diskspd>), проверьте соответствие производительности сети и дисков сервера ожидаемым показателям перед установкой SQL Server. Эти инструменты помогут обнаружить потенциальные проблемы в настройках, связанные с сетью и дисками.

## **15) Создайте резервные копии.**

Прежде чем приступить к обновлению, не забудьте сделать резервные копии баз данных, файлов приложений и операционной системы сервера. Иногда можно задействовать моментальный снимок (или контрольную точку) виртуальной машины, чтобы упростить процесс. Рекомендую применять тройное правило к резервному копированию компьютера (<https://www.hanselman.com/blog/TheComputerBackupRuleOfThree.aspx>).

Еще следует отметить: полезны лишь такие резервные копии, которые можно восстановить. Поэтому протестируйте процесс восстановления перед обновлением.

Обновления — неизбежная часть любого цикла разработки. Вероятность успешного обновления возрастает пропорционально усилиям, затраченным на планирование и подготовку безупречного процесса обновления. В следующей статье мы рассмотрим типы обновления и задачи, решаемые в процессе обновления.



---

Томас Ла Рок (ThomasLaRock@confio.com) — главный технический эксперт в компании SolarWinds, имеет сертификат Microsoft Certified Master

# Переходим на SQL Server

## Часть 2

2016

Томас Ла Рок

**В** первой статье об обновлении Microsoft SQL Server 2016 мы рассмотрели причины обновления, а также некоторые предварительные задачи, о которых необходимо помнить. Теперь мы остановимся на задачах, выполняемых в ходе обновления.

### В процессе обновления

Как отмечалось в предыдущей статье, существует два основных типа обновления: на месте и параллельное. При обновлении на месте нет необходимости беспокоиться о передаче данных, тогда как для параллельного обновления требуется перемещать данные с одного сервера на другой. Существует четыре основных варианта миграции данных.

- **Резервное копирование и восстановление:** удачный вариант для небольших систем и поэтапных миграций; также можно использовать метод отсоединения и присоединения.
- **Предварительное размещение:** подготовка данных с использованием полного, разностного резервного копирования или резервного копирования журналов транзакций для уменьшения объема перемещаемых данных; еще заслуживает внимания метод доставки журналов.

- **Зеркальное отображение базы данных:** обеспечивает простой перенос данных со старой системы на новую.

- **Группы доступности:** более сложный метод, чем зеркальное отображение базы данных, так как может охватывать несколько баз данных.

В первой статье была еще представлена концепция последовательного обновления. В этом случае используются функции высокой доступности, такие как зеркальное отображение, кластеризация или группы доступности. Идея состоит в том, что вы можете обновить вторичный узел, перейти на другой ресурс и продолжать обновление всех узлов до тех пор, пока не будет обновлен первичный узел, а затем при необходимости выполнить переход на другой ресурс. Рассмотрим шаги, необходимые для реализации каждого подхода.

### Обновление на месте

Выполнить обновление на месте проще всего, но сложно вернуться назад в случае возникновения неполадок. Обновление на месте предполагает следующие действия:

1. Убедитесь, что существуют резервные копии всех баз данных (пользовательских и системных). Если имеется база данных, для которой не используется модель восстановления SIMPLE,

проверьте, существует ли резервная копия журнала транзакций. Убедитесь, что эти резервные копии могут быть восстановлены.

2. Просмотрите список необходимых компонентов для SQL Server 2016 (<https://msdn.microsoft.com/en-us/library/ms143712.aspx>) и установите нужные.
3. Запустите программу установки из дистрибутива SQL Server 2016.
4. Выполните задачи, которые нужно завершить после обновления.
5. Проведите тщательное тестирование и убедитесь в корректной работе всех функций.

Необходимых действий при параллельном обновлении больше, и они считаются более сложными, но при этом обеспечивается более гибкий возврат к предыдущему состоянию, поскольку вы не вступаете в контакт с исходной системой, пока она еще эксплуатируется.

Шаги параллельного обновления похожи как для текущего, так и для нового сервера базы данных. Единственное различие в том, что для нового сервера необходимо установить SQL Server.

1. Убедитесь, что существуют резервные копии для всех баз данных (пользовательских и системных). Если имеется база данных, для которой не используется модель восстановления SIMPLE, проверьте, существует ли резервная копия журнала транзакций. Убедитесь, что эти резервные копии могут быть восстановлены.
2. С помощью сценариев определите все необходимые системные объекты.
3. С помощью сценариев определите необходимые пакеты SSIS (из MSDB или как неструктурированные файлы).
4. Для нового экземпляра на новом сервере:
  - просмотрите список необходимых компонентов для SQL Server 2016 (<https://msdn.microsoft.com/en-us/library/ms143712.aspx>) и установите нужные;
  - установите нужную версию и редакцию SQL Server 2016.
5. Используйте сценарии со старого сервера, чтобы создать необходи-

мые системные объекты на новом сервере.

6. Перенесите пакеты SSIS в MSDB (или неструктурированные файлы, если применимо).
7. Выберите базы данных для переноса, переведите в автономный режим.
8. Перенесите базу данных на новый экземпляр. Повторите это действие для каждой базы данных.
9. Выполните задачи, которые нужно завершить после обновления.
10. Проведите тщательное тестирование и убедитесь в корректной работе всех функций.

### **Последовательное обновление**

Последовательный метод поможет уменьшить простой в процессе обновления. Последовательное обновление с использованием зеркального отображения базы данных для меня является предпочтительным методом обновления SQL Server, но вы можете еще выбрать доставку журналов или группы доступности. Только убедитесь, что у вас есть надежный план отката для любого метода.

Последовательное обновление состоит из следующих шагов:

1. Выберите метод высокой доступности (доставка журналов, зеркальное отображение базы данных, группы доступности).
2. Выберите один из вариантов:
  - обновите один из вторичных узлов, следя приведенным выше инструкциям по обновлению на месте;
  - установите SQL Server 2016 на новом сервере (и добавьте его в качестве узла, если это возможно).
3. Выполните переключение на данный вторичный узел.
4. Выполните задачи, которые нужно завершить после обновления.
5. Проведите тщательное тестирование и убедитесь в корректной работе всех функций.
6. Повторите обновление для всех оставшихся вторичных узлов.
7. Выполните задачи, которые нужно завершить после обновления.
8. Проведите тщательное тестирование и убедитесь в корректной работе всех функций на каждом узле.

9. Повторите обновление для первичного узла.

10. Выполните задачи, которые нужно завершить после обновления.
11. Проведите тщательное тестирование и убедитесь в корректной работе всех функций на первичном узле.

При последовательном обновлении не требуется переключаться на исходный сервер (первичный узел). Вполне достаточно настроить зеркальное отображение базы данных с единственной целью выполнить последовательное обновление. После переключения на вторичный узел можно устраниТЬ зеркальное отображение и удалить сервер. Результат будет тот же, что при параллельной миграции, но с меньшим временем простоя, чем при использовании традиционных методов резервного копирования и восстановления или отключения и подключения. Для очень больших баз данных такая концепция важна, поскольку восстановление может оказаться трудной задачей.

Имейте в виду, что перемещение данных при последовательном обновлении происходит лишь в одном направлении. Вы можете перейти со старой версии SQL Server на новую, но не наоборот. Поэтому если вы выполняете последовательное обновление и перемещаете данные на обновленный узел, то не сможете вернуться назад без восстановления из резервных копий на исходном сервере. Вы получите соответствующие сообщения об ошибках, будьте к этому готовы. Просто помните, что SQL Server предупреждает вас о невозможности вернуться к предыдущей версии.

Процесс обновления SQL Server несложен сам по себе, если составить правильный план, но остаются дополнительные задачи, которые необходимо завершить после обновления и прежде чем система будет передана конечным пользователям для тестирования. О задачах, которые необходимо выполнить после обновления, речь пойдет в следующей статье серии.



Томас Ла Рок ([ThomasLaRock@confio.com](mailto:ThomasLaRock@confio.com)) — главный технический эксперт в компании SolarWinds, имеет сертификат Microsoft Certified Master

# Переходим на SQL Server

## Часть 3

2016

Томас Ла Рок

**В** предыдущих двух статьях мы рассмотрели аргументы в пользу перехода на Microsoft SQL Server 2016 и некоторые подготовительные задачи для обновления, а также задачи, которые необходимо выполнить в процессе обновления. Теперь перейдем к задачам, выполняемым после обновления.

### По завершении обновления

После того как переход на SQL Server 2016 завершен, необходимо выполнить последовательность шагов, дабы убедиться, что база данных готова к передаче конечным пользователям для дальнейшего тестирования. Рассмотрим этот этап подробнее. Первое, что нам предстоит, — это создание резервных копий. Немедленно. Прежде, чем предпринимать любые другие действия. Ведь вы администратор базы данных, и резервное копирование должно быть вашей привычкой. Вы должны получить резервную копию перед началом любого обновления или миграции, и настоятельно рекомендуется сделать это сейчас и повторно перед передачей базы данных конечным пользователям. Кроме того, следует сохранить все выходные данные из перечисленных здесь

элементов. Они пригодятся, если что-то пойдет не так. И не забудьте проверить корректность резервных копий (см. материал [thomaslarock.com/2010/05/statistical-sampling-for-verifying-database-backups/](http://thomaslarock.com/2010/05/statistical-sampling-for-verifying-database-backups/)).

### DBCC CHECKDB

После миграции или обновления необходимо выполнить следующую инструкцию (дополнительная информация приведена в статье по адресу: [msdn.microsoft.com/en-us/library/ms176064\(v=sql.120\).aspx](http://msdn.microsoft.com/en-us/library/ms176064(v=sql.120).aspx)):

`DBCC CHECKDB WITH DATA_PURITY;`

Эта инструкция позволит проверить ваши данные на предмет значений, которые более недействительны для типа данных столбца. Предполагается, что для баз данных, созданных в SQL Server 2005 и более новых версиях, проверка DATA\_PURITY выполняется автоматически с помощью обычной команды CHECKDB. Иная ситуация с версиями, предшествующими SQL Server 2005 (они еще эксплуатируются), поэтому данный шаг еще более важен.

Но что делать с базой данных, созданной в SQL Server 2000, перенесенной в экземпляр SQL Server 2008 и оставленной в режиме обратной совместимости SQL Server 2000 (номер режима совместимости 80)? Исходить

из того, что проверка DATA\_PURITY выполнена? Совет: в любом случае запустите ее самостоятельно. Так вы в любом случае будете уверены в том, что проверка выполнена.

Кроме того, следует отметить, что проверки целостности столбца не выполняются, когда используется параметр PHYSICAL\_ONLY.

### DBCC UPDATEUSAGE

Эта команда не так важна, как DATA\_PURITY, но все же принимает участие в любом процессе миграции или обновления:

```
DBCC UPDATEUSAGE (db_name);
```

С помощью этой команды (дополнительная информация приведена в статье по адресу: [msdn.microsoft.com/en-us/library/ms188414\(v=sql.120\).aspx](http://msdn.microsoft.com/en-us/library/ms188414(v=sql.120).aspx)) можно исправить любые неточности в подсчете страниц, возникающие из-за неверных результатов, возвращенных хранимой процедурой sp\_spaceused. Лучше всего выполнить эту команду сразу, но помните, что для ее выполнения требуется некоторое время, в зависимости от размера таблицы или базы данных. Предпочтительно запускать ее регулярно вследствие одной из перечисленных ниже причин.

1. Есть подозрения, что вы видите неверные значения, возвращенные для sp\_spaceused.
2. Ваша база данных содержит много инструкций DDL (CREATE, ALTER или DROP).

### Обновление статистики

Этот шаг нельзя пропускать. Он просто обязателен для любой миграции или обновления:

```
USE db_name; GO EXEC sp_updatestats;
```

Приведенная команда обновит статистику для всех таблиц в базе данных. Она выдает команду UPDATE STATISTICS, которая заслуживает упоминания, так как ее полезно использовать с параметром FULLSCAN. Я предполагаю проявить излишнюю осторожность, чтобы впоследствии не жалеть об опрометчивости, поэтому выполняю примерно такую команду:

```
USE db_name; GO EXEC sp_MSforeachtable  
    @command1='UPDATE STATISTICS  
    WITH FULLSCAN';
```

Важно не забыть об обновлении статистики после модернизации. Без этого скорость выполнения запросов может уменьшиться, когда вы приступите к тестированию, и в итоге вы впустую потратите время, пытаясь устраниТЬ возможные узкие места. В SQL Server 2016 также появился новый модуль оценки мощности, или количества элементов в наборе, Cardinality Estimator (CE). Оценка планов оптимизатором запросов зависит от точности статистических данных, поэтому, прежде чем начать тестирование, необходимо собрать как можно более точные статистические данные. Позаботьтесь о статистических данных заблаговременно, чтобы не беспокоиться о них впоследствии.

### Обновление определений представления

Верьте или нет, но некоторые пользователи строят представление, распространяющееся на другую базу данных на том же экземпляре. Порой, как ни удивительно, эти представления охватывают и связанный сервер. И невероятный шаг: иногда эти представления создаются с помощью синтаксиса SELECT \*. Какова вероятность, что подобный программный код существует в вашей компании? Такое случается. И если имеется некорректный программный код сверх представлений, охватывающих другие базы данных (или представлений чего-то еще, спроектированного какими-нибудь фанатом представлений), вам следует использовать хранимую процедуру sp\_refreshview для обновления этих представлений.

### Проверка уровней совместимости

Если вам случалось выполнять обновления SQL Server, то вы, вероятно, заметили, что уровень совместимости не распространяется на новейшую версию после завершения миграции. Необходимо самостоятельно вручную назначить уровень совместимости. В SQL Server 2016 это важнее, чем в предыдущих версиях, из-за нового модуля CE.

Рекомендуется обновить каждую базу данных на экземпляре SQL Server 2016 до режима совместимости 130, а затем выполнить тщательное тестирование. Предполагается, что известны эталонные уровни производительности для важнейших запросов до миграции, и вы можете убедиться, что новый модуль CE не работает против вас.

### Проверка количества объектов

Помните результаты прошлых подсчетов количества объектов, таких как таблицы и хранимые процедуры? Выполните подсчеты заново. Убедитесь, что количество объектов такое же, как до обновления и миграции. Не забывайте о принципе обновлений SQL Server: не потерять ни единой таблицы!

### Проверка настроек

В процессе подготовки к обновлению следует собрать сведения о приложениях собственной разработки и сторонних поставщиков, использующих сервер базы данных. Также следует собрать информацию о конкретных настройках, применяемых к операционной системе сервера, экземпляру базы данных и самой базе данных. После этого необходимо изучить эти данные и убедиться, что настройки применены к новому серверу.

Обновления — неотъемлемая часть любого цикла разработки. Вероятность успешного обновления увеличивается пропорционально количеству усилий по планированию и подготовке, затраченных на организацию процесса обновления. Возможно, вы уже продумали обновление до SQL Server 2016 или только намереваетесь сделать это (кстати, время пришло). Тогда вы можете воспользоваться данной серией статей, чтобы составить контрольный список задач перед модернизацией, во время модернизации и после нее. Убежден, что это позволит вам избежать многих проблем.



Томас Ла Рок ([ThomasLaRock@confio.com](mailto:ThomasLaRock@confio.com)) — главный технический эксперт в компании SolarWinds, имеет сертификат Microsoft Certified Master

# Запрос функции T-SQL

Добавляем  
предложение  
RESET WHEN  
для пересоздания  
оконного раздела

**Ицик Бен-Ган**

**C**помощью оконных функций можно изящно и эффективно решать многие задачи T-SQL, связанные с обработкой запросов. И все же в версии SQL Server 2017 остаются задачи, для которых трудно найти решение на основе набора, но зато их можно обработать, если дополнить T-SQL поддержкой для оконного оператора с именем RESET WHEN. Этот оператор осуществляет пересоздание оконного раздела, когда выполняется определенное условие — возможно, на основе оконной функции. Это нестандартная функция, но сейчас она поддерживается компанией Teradata (документацию по функции можно найти по адресу: [http://www.info.teradata.com/HTMLPubs/DB\\_TTU\\_14\\_00/index.html#page/SQL\\_Reference/B035\\_1145\\_111A/Ordered\\_Analytical\\_Functions.083.010.html#ww1285495](http://www.info.teradata.com/HTMLPubs/DB_TTU_14_00/index.html#page/SQL_Reference/B035_1145_111A/Ordered_Analytical_Functions.083.010.html#ww1285495)). Хочу выразить благодарность Александру Месе (обладателю статуса Microsoft Data Platform MVP), познакомившему меня с этой функцией.

В данной статье я представлю две классические задачи обработки запросов T-SQL, опишу решения, используемые в настоящее время, и объясню, как применить усовершенствованные решения с использованием оператора RESET WHEN. Надеюсь, прочитав статью до конца, вы поймете, насколько важна эта функция, и проголосуете-

txid	qty	totalqty	depletionqty
1	2	2	0
2	5	0	7
3	4	4	0
4	1	5	0
5	10	0	15
6	3	3	0
7	1	4	0
8	2	0	6
9	1	1	0
10	2	3	0
11	1	4	0
12	9	0	13

Экран 1. Желаемый результат для тестовых данных и емкости входного контейнера, равной 5

те за нее на сайте Microsoft Connect (<https://connect.microsoft.com/SQLServer/feedback/details/2748755>).

### Убавление величины

Начнем с задачи, связанной с вычислением нарастающих итогов, которые нужно убавить, когда выполняется определенное условие. Джери Решеф, обладатель статуса Microsoft Data Platform MVP, представил исходную задачу, а затем Шарон Ример из компании Naya Technologies предложил вариант этой задачи.

Среди тестовых данных для рассматриваемой задачи — таблица с именем Transactions, которую вы создаете и заполняете с помощью программного кода листинга 1.

Транзакции добавляют величины некоторого элемента в контейнер на основе упорядочения по столбцу txid. Если кумулятивная величина превышает емкость контейнера (представленного как вход), то контейнер должен быть убавлен. Ваше решение должно показать состояние контейнера (общую величину) после каждой транзакции: 0 это общая величина после убавления, а также величина убавления, когда это применимо. На экране 1 приведен желаемый результат для тестовых данных и емкости входного контейнера, равной 5.

### Решение на основе курсора

Такие задачи часто пытаются решать с использованием рекурсивных запросов. Эти решения могут быть изящными, но они не очень эффективны. Также применяется метод, известный как «подстановочное обновление» (quirky update), который очень эффективен, но не гарантирует результата, поскольку зависит от физического порядка обработки. Пока я не нашел эффективного, гарантировавшего результат решения на основе наборов для этой задачи и вынужден использовать итеративные решения (на основе T-SQL или CLR).

В листинге 2 приведен пример простого итеративного решения T-SQL с использованием курсора.

### Листинг 1. Создание и заполнение таблицы Transactions

```
SET NOCOUNT ON;
USE tempdb;

DROP TABLE IF EXISTS dbo.Transactions;
GO
CREATE TABLE dbo.Transactions
(
    txid INT NOT NULL CONSTRAINT PK_Transactions PRIMARY KEY,
    qty INT NOT NULL
);
GO

TRUNCATE TABLE dbo.Transactions;

INSERT INTO dbo.Transactions(txid, qty)
VALUES(1,2),(2,5),(3,4),(4,1),(5,10),(6,3),
      (7,1),(8,2),(9,1),(10,2),(11,1),(12,9);
```

### Листинг 2. Пример простого итеративного решения с использованием курсора

```
SET NOCOUNT ON;

DECLARE @maxallowedqty AS INT = 5;

DECLARE @C AS CURSOR, @txid AS INT, @qty AS INT, @totalqty AS INT = 0, @depletionqty AS INT = 0;

DECLARE @Result AS TABLE
(
    txid     INT NOT NULL PRIMARY KEY,
    qty      INT NOT NULL,
    totalqty INT NOT NULL,
    depletionqty INT NOT NULL
);

SET @C = CURSOR FORWARD_ONLY STATIC READ_ONLY FOR
    SELECT txid, qty
    FROM dbo.Transactions
    ORDER BY txid;

OPEN @C;

FETCH NEXT FROM @C INTO @txid, @qty;

WHILE @@FETCH_STATUS = 0
BEGIN
    SELECT @totalqty += @qty, @depletionqty = 0;

    IF @totalqty > @maxallowedqty
    BEGIN
        SET @depletionqty = @totalqty;
        SET @totalqty = 0;
    END;

    INSERT INTO @Result(txid, qty, totalqty, depletionqty)
    VALUES(@txid, @qty, @totalqty, @depletionqty);

    FETCH NEXT FROM @C INTO @txid, @qty;
END;

SELECT txid, qty, totalqty, depletionqty
FROM @Result
ORDER BY txid;
```

Программный код определяет переменную курсора и использует ее, чтобы извлекать транзакции по одной в хронологическом

**Листинг 3. Решение с использованием RESET WHEN**

```

DECLARE @maxallowedqty AS INT = 5;

SELECT txid, qty,
SUM(qty) OVER(ORDER BY txid
RESET WHEN
    -- сбросить секцию окна, когда
    -- сумма с накоплением до предшествующей строки > @maxallowedqty
    SUM(qty) OVER(ORDER BY txid
        ROWS BETWEEN UNBOUNDED PRECEDING
        AND 1 PRECEDING)
    > @maxallowedqty
ROWS UNBOUNDED PRECEDING) AS runsum
FROM dbo.Transactions;

```

**Листинг 4. Полный программный код решения**

```

DECLARE @maxallowedqty AS INT = 5;

WITH C AS
(
    SELECT *,
    SUM(qty) OVER(ORDER BY txid
    RESET WHEN
        SUM(qty) OVER(ORDER BY txid
            ROWS BETWEEN UNBOUNDED PRECEDING
            AND 1 PRECEDING)
        > @maxallowedqty
    ROWS UNBOUNDED PRECEDING) AS runsum
    FROM dbo.Transactions
)
SELECT txid, qty, totalqty, runsum - totalqty AS depletionqty
FROM C
CROSS APPLY
( VALUES(CASE WHEN runsum > @maxallowedqty THEN 0 ELSE runsum END ) )
AS A(totalqty);

```

**Листинг 5. Решение с выражением и внешним запросом**

```

WITH C AS
(
    SELECT *,
    SUM(qty) OVER(ORDER BY txid
    RESET WHEN
        SUM(qty) OVER(ORDER BY txid
            ROWS BETWEEN UNBOUNDED PRECEDING
            AND 1 PRECEDING)
        > @maxallowedqty
    ROWS UNBOUNDED PRECEDING) AS runsum
    FROM dbo.Transactions
)
SELECT COUNT(CASE WHEN runsum > @maxallowedqty THEN 1 END ) AS timesexceeded
FROM C;

```

порядке. Величины накапливаются в переменной с именем @totalqty. После извлечения каждой строки программный код проверяет, не превышает ли накопленная величина емкость контейнера. Если происходит превышение, переменной с именем @depletionqty присваивается значение @totalqty,

а затем текущее значение @totalqty обнуляется. После этого программный код записывает сведения о текущей транзакции (txid и qty) наряду с текущими значениями @totalqty и @depletionqty в табличную переменную. После прохода по всем транзакциям код запрашивает табличную перемен-

txid	qty	runsum
1	2	2
2	5	7
3	4	4
4	1	5
5	10	15
6	3	3
7	1	4
8	2	6
9	1	1
10	2	3
11	1	4
12	9	13

**Экран 2. Результаты решения с использованием RESET WHEN**

ную, чтобы получить желаемый результат.

**Решение с использованием RESET WHEN (не поддерживается в SQL Server 2017)**

Недостатки применения итеративных решений хорошо известны. Вопрос в том, существует ли удачная альтернатива на основе набора. До настоящего времени я не нашел таковой для поставленной задачи с использованием существующих инструментов T-SQL, но хотел бы, чтобы она когда-нибудь появилась. Как отмечалось выше, компания Teradata поддерживает оконный оператор с именем RESET WHEN, который пересоздает оконный раздел при выполнении определенного условия. Ценность этого предложения в том, что в условии можно использовать оконную функцию и вы можете узнать, что аккумулировано до предыдущей строки. В нашей задаче оконный раздел пересоздается, когда сумма величин от начала секции до предыдущей строки превышает лимит входного контейнера (листинг 3). Помните, что на сегодня этот программный код не поддерживается в SQL Server. Если бы он поддерживался, то были бы получены такие выходные данные, как показано на экране 2.

**Листинг 6. Создание и заполнение таблицы Stocks**

```

SET NOCOUNT ON;
USE tempdb;

DROP TABLE IF EXISTS dbo.StockRates;
GO
CREATE TABLE dbo.StockRates
(
    stockid INT NOT NULL,
    dt DATE NOT NULL,
    val INT NOT NULL,
    CONSTRAINT PK_StockRates PRIMARY KEY(stockid, dt)
);
GO
INSERT INTO dbo.StockRates VALUES
(1, '2017-08-01', 13),
(1, '2017-08-02', 14),
(1, '2017-08-03', 17),
(1, '2017-08-04', 40),
(1, '2017-08-05', 45),
(1, '2017-08-06', 52),
(1, '2017-08-07', 56),
(1, '2017-08-08', 60),
(1, '2017-08-09', 70),
(1, '2017-08-10', 30),
(1, '2017-08-11', 29),
(1, '2017-08-12', 35),
(1, '2017-08-13', 40),
(1, '2017-08-14', 45),
(1, '2017-08-15', 60),
(1, '2017-08-16', 60),
(1, '2017-08-17', 55),
(1, '2017-08-18', 60),
(1, '2017-08-19', 20),
(1, '2017-08-20', 15),
(1, '2017-08-21', 20),
(1, '2017-08-22', 30),
(1, '2017-08-23', 40),
(1, '2017-08-24', 20),
(1, '2017-08-25', 60),
(1, '2017-08-26', 80),
(1, '2017-08-27', 70),
(1, '2017-08-28', 70),
(1, '2017-08-29', 40),
(1, '2017-08-30', 30),
(1, '2017-08-31', 10),
(2, '2017-08-01', 3),
(2, '2017-08-02', 4),
(2, '2017-08-03', 7),
(2, '2017-08-04', 30),
(2, '2017-08-05', 35),
(2, '2017-08-06', 42),
(2, '2017-08-07', 46),
(2, '2017-08-08', 50),
(2, '2017-08-09', 60),
(2, '2017-08-10', 20),
(2, '2017-08-11', 19),
(2, '2017-08-12', 25),
(2, '2017-08-13', 30),
(2, '2017-08-14', 35),
(2, '2017-08-15', 50),
(2, '2017-08-16', 50),
(2, '2017-08-17', 45),
(2, '2017-08-18', 50),
(2, '2017-08-19', 10),
(2, '2017-08-20', 5),
(2, '2017-08-21', 10),
(2, '2017-08-22', 20),
(2, '2017-08-23', 30),
(2, '2017-08-24', 10),
(2, '2017-08-25', 50),
(2, '2017-08-26', 70),
(2, '2017-08-27', 60),
(2, '2017-08-28', 60),
(2, '2017-08-29', 30),
(2, '2017-08-30', 20),
(2, '2017-08-31', 1);

```

Как мы видим, оконный раздел пересоздан после транзакций с идентификаторами 2, 5 и 8.

Чтобы получить окончательный желаемый результат, присваиваем общей величине контейнера (назовем ее totalqty) значение 0, когда сумма с накоплением превышает лимит контейнера, и значение суммы с накоплением в противном случае. Затем можно вычислить величину убывания (назовем ее depletionqty) как сумму с накоплением за вычетом общей величины. В листинге 4 приведен полный программный код решения.

Как видите, решение простое, компактное и изящное.

Недавно Шарон Ример из компании Naya Technologies представил вариант этой задачи на основе заказа от одного из клиентов компании.

stockid	startdate	enddate	maxvalue
1	2017-08-06	2017-08-18	70
1	2017-08-25	2017-08-28	80
2	2017-08-08	2017-08-18	60
2	2017-08-25	2017-08-28	70

**Экран 3. Желаемый результат для решения задачи с островами сложности**

Требовалось вычислить, сколько раз контейнер превышает входной лимит. Для этой цели следует иметь такое же определение для CTE C и использовать внешний запрос для подсчета.

```

SELECT COUNT (CASE WHEN runsum >
    @maxallowedqty THEN 1 END)
    AS timesexceeded
    FROM C;

```

Полное решение выглядит таким образом, как показано в листинге 5. Опять-таки компактно и изящно.

**Островки сложности**

Есть много других задач, для которых на сегодня существует приемлемое решение T-SQL на основе набора, но их проще решить с помощью оператора RESET WHEN. Хороший пример — задачи об островах сложности, в которых нужно определить новый остров всякий раз, когда выполняется условие, сравнивающее какой-то элемент из текущей строки с элементом из предыдущей. Чтобы

**Листинг 7. Программный код для решения задачи с островами сложности**

```
SELECT stockid, dt, val,
CASE
    WHEN DATEDIFF(day, LAG(dt) OVER(PARTITION BY stockid ORDER BY dt), dt)
        < 7
    THEN 0
    ELSE 1
END AS isstart
FROM dbo.StockRates
WHERE val >= 50;
```

**Листинг 8. Формирование идентификатора острова**

```
WITH C1 AS
(
    SELECT *,
    CASE
        WHEN DATEDIFF(day, LAG(dt) OVER(PARTITION BY stockid ORDER BY dt), dt)
            < 7
        THEN 0
        ELSE 1
    END AS isstart
    FROM dbo.StockRates
    WHERE val >= 50
),
C2 AS
(
    SELECT *,
    SUM(isstart) OVER(PARTITION BY stockid ORDER BY dt
        ROWS UNBOUNDED PRECEDING) AS grp
    FROM C1
)
SELECT stockid,
    MIN(dt) AS startdate,
    MAX(dt) AS enddate,
    MAX(val) as maxvalue
FROM C2
GROUP BY stockid, grp
ORDER BY stockid, startdate;
```

**Листинг 9. Пример вычисления идентификатора острова с RESET WHEN**

```
SELECT stockid, dt, val,
    MIN(dt) OVER(PARTITION BY stockid
        ORDER BY dt
        RESET WHEN DATEDIFF(day,
            MIN(dt) OVER(
                PARTITION BY stockid ORDER BY dt
                ROWS BETWEEN 1 PRECEDING AND 1 PRECEDING),
            dt) >= 7
        ROWS UNBOUNDED PRECEDING) AS grp
FROM dbo.Stocks
WHERE val >= 50;
```

продемонстрировать такую задачу, я использую таблицу с именем Stocks, созданную и заполненную с помощью программного кода листинга 6.

В этой таблице отслеживаются курсы акций на момент закрытия дневных торгов. Предположим, вам требуется идентифицироватьperi-

оды (острова), в которых значение курса акций больше или равно 50, и для каждого периода необходимо показать время начала и конца, а также максимальное значение курса в течение периода. При этом необходимо игнорировать периоды продолжительностью до 6 дней. На экране 3 показан

желаемый результат для тестовых данных.

Например, обратите внимание, что для первого острова для акции с идентификатором 1 игнорируется период между Aug. 9, 2017 и Aug. 15, 2017, поскольку его длительность не превышает 6 дней, но не игнорируется период между Aug. 18, 2017 и Aug. 25, 2017, так как его продолжительность 7 дней.

**Поддерживаемое решение T-SQL**

Как отмечалось выше, на сегодня существуют поддерживаемые решения на основе набора для задач с островами, подобных рассматриваемой, но они длиннее и сложнее, чем предусматривающие использование оператора RESET WHEN. При применении одного из решений, поддерживаемых в настоящее время, на первом шаге вычисляется флаг (назовем его isstart), которому присваивается значение 0, если это не начало острова, путем сравнения какого-нибудь элемента текущей строки с неким элементом предшествующей (полученным с помощью функции LAG). В противном случае флагу присваивается значение 1. В нашем случае после фильтрации только строк, в которых значение акции превышает или равно 50, флаг получает значение 0, когда разница между предыдущей датой и текущей менее 7 дней; в противном случае флаг имеет значение 1. В листинге 7 показан программный код, реализующий этот шаг. Выходные данные, которые он формирует, приведены на экране 4. На втором шаге мы получаем идентификатор острова путем вычисления суммы с накоплением флага isstart. Наконец, данные группируются по stockid и isstart, и возвращаются даты начала и конца острова, а также максимальный курс акций в течение периода. В листинге 8 приводится полный текст решения.

**Решение****с использованием RESET WHEN (не поддерживается в версии SQL Server 2017)**

С помощью оператора RESET WHEN решить задачу было бы

stockid	dt	val	isstart
1	2017-08-06	52	1
1	2017-08-07	56	0
1	2017-08-08	60	0
1	2017-08-09	70	0
1	2017-08-15	60	0
1	2017-08-16	60	0
1	2017-08-17	55	0
1	2017-08-18	60	0
1	2017-08-25	60	1
1	2017-08-26	80	0
1	2017-08-27	70	0
1	2017-08-28	70	0
2	2017-08-08	50	1
2	2017-08-09	60	0
2	2017-08-15	50	0
2	2017-08-16	50	0
2	2017-08-18	50	0
2	2017-08-25	50	1
2	2017-08-26	70	0
2	2017-08-27	60	0
2	2017-08-28	60	0

Экран 4. Результаты работы кода листинга 7

stockid	dt	val	grp
1	2017-08-06	52	2017-08-06
1	2017-08-07	56	2017-08-06
1	2017-08-08	60	2017-08-06
1	2017-08-09	70	2017-08-06
1	2017-08-15	60	2017-08-06
1	2017-08-16	60	2017-08-06
1	2017-08-17	55	2017-08-06
1	2017-08-18	60	2017-08-06
1	2017-08-25	60	2017-08-25
1	2017-08-26	80	2017-08-25
1	2017-08-27	70	2017-08-25
1	2017-08-28	70	2017-08-25
2	2017-08-08	50	2017-08-08
2	2017-08-09	60	2017-08-08
2	2017-08-15	50	2017-08-08
2	2017-08-16	50	2017-08-08
2	2017-08-18	50	2017-08-08
2	2017-08-25	50	2017-08-25
2	2017-08-26	70	2017-08-25
2	2017-08-27	60	2017-08-25
2	2017-08-28	60	2017-08-25

Экран 5. Выходные данные с использованием функции LAG

проще, так как можно всего лишь пересоздать оконный раздел, когда выполняется условие для начала нового острова. Затем можно воспользоваться минимальной датой в секции как идентификатором острова.

В листинге 9 показан пример вычисления идентификатора острова (назовем его grp).

В приведенном программном коде используется функция окна MIN, чтобы получить дату из предыдущей строки с помощью экстента фрейма окна ROWS BETWEEN 1 PRECEDING AND 1 PRECEDING. Иначе можно получить дату из предшествующей строки с помощью функции LAG, например так (показан только альтернативный оператор RESET WHEN):

```
RESET WHEN DATEDIFF (day,
LAG (dt) OVER (PARTITION BY stockid
    ORDER BY dt),
dt) >= 7
```

Выходные данные этого шага на экране 5 показывают полученный идентификатор группы для каждого острова.

Затем потребуется всего один шаг для группирования этих данных

#### Листинг 10. Шаг для группирования данных

```
WITH C AS
(
    SELECT *,
        MIN(dt) OVER(PARTITION BY stockid
            ORDER BY dt
            RESET WHEN DATEDIFF(day,
                LAG(dt) OVER(PARTITION BY stockid ORDER BY dt),
                dt) >= 7
                ROWS UNBOUNDED PRECEDING) AS grp
    FROM dbo.Stocks
    WHERE val >= 50
)
SELECT stockid,
    grp AS startdate,
    MAX(dt) AS enddate,
    MAX(val) as maxvalue
FROM C
GROUP BY stockid, grp
ORDER BY stockid, grp;
```

по stockid и grp. Возвращаются даты начала и конца каждого периода, а также максимальный курс акции в каждом из них (листинг 10).

Как видите, это решение короче и проще, чем поддерживаемое в настоящее время.

Оконные функции, бесспорно, обладают выдающимися возможностями, но они могут быть еще более эффективными. Оператор

RESET WHEN в случае реализации в T-SQL позволит заменить существующие итеративные решения изящными и более мощными решениями на основе набора. ♦

Ицик Бен-Ган (Itzik@SolidQ.com) — преподаватель и консультант, автор книг по T-SQL, имеет звание SQL Server MVP

# Клонирование объектов безопасности между серверами

Тим Форд

П ривычная задача администратора базы данных — реплицировать имена учетных записей для регистрации между несколькими экземплярами SQL Server. Возможно, вы строите отдельную среду, имитирующую производственную (Dev, UAT и т. д.) или готовите сменный сервер или несколько узлов, которые составят группу доступности. В таком случае вам потребуются идентично настроенные объекты безопасности (имена учетных записей для регистрации, принадлежность к роли, пользователи). Часто возникают различные проблемы, из-за кото-

рых этот вопрос становится гораздо более сложным, чем кажется на первый взгляд. Различия в идентификаторе безопасности (SID) между экземплярами для имен учетных записей SQL Server, несовпадающие пароли и потерянные учетные записи пользователей — лишь некоторые из них. Я придерживаюсь правила: если нужно многократно выполнить функцию, то следует подготовить сценарий, повторное использование которого упрощает жизнь администратора и освобождает время для работы с другими задачами или позволяет добиться желаемых результатов ценой меньших усилий. Много лет назад я напи-



principal_id	name	default_database_name	script_command	user_command
269	austenford	master	EXEC [sp_help_revlogin] 'austenford';	USE [master]; CREATE USER [austenford] FROM LOGI...
271	chriscornell	master	EXEC [sp_help_revlogin] 'chriscornell';	USE [master]; CREATE USER [chriscornell] FROM LOG...
270	davidbowie	master	EXEC [sp_help_revlogin] 'davidbowie';	USE [master]; CREATE USER [davidbowie] FROM LOG...
272	prince	master	EXEC [sp_help_revlogin] 'prince';	USE [master]; CREATE USER [prince] FROM LOGIN [p...
267	timford	master	EXEC [sp_help_revlogin] 'timford';	USE [master]; CREATE USER [timford] FROM LOGIN [ti...
268	trevorford	master	EXEC [sp_help_revlogin] 'trevorford';	USE [master]; CREATE USER [trevorford] FROM LOGI...

Экран 1. Результаты работы листинга 3

сал сценарий, которым хочу поделиться в данной статье. С его помощью можно создать следующие объекты безопасности на одном исходном экземпляре SQL для распространения на другие экземпляры по мере необходимости:

- имена учетных записей SQL Server;
- имена учетных записей доверенной проверки подлинности (Active Directory или AD);
- группы доверенной проверки подлинности (AD);
- пользователи базы данных по умолчанию;
- принадлежность к роли сервера.

Наша цель — убедиться, что все идентификаторы безопасности совпадают, все членства в ролях уровня сервера назначены и нет риска, что назначение базы данных по умолчанию не будет иметь соответствующего объекта пользователя, что потенциально может привести к ошибкам подключения.

### Необходимые условия

Уже в течение десяти с лишним лет в сообществе SQL Server широко распространены два сценария, необходимые для корректного выполнения моего сценария: `sp_help_revlogin` и `sp_hexadecimal`. `sp_help_revlogin` генерирует программный код, который воссоздает имя учетной записи для входа и принудительно формирует SID, чтобы уменьшить вероятность рас согласования SID между именами входа и пользователями, копируемыми между экземплярами. `sp_hexadecimal` необходим для `sp_help_revlogin`, чтобы преобразовать хеш-код пароля в текстовую форму, которая используется в сценарии. Программный код обеих хранимых процедур приводится в листин-

### Листинг 1. Создание sp\_hexadecimal

```
=====
-- СОЗДАНИЕ sp_hexadecimal
=====
USE [master]
GO

CREATE PROCEDURE sp_hexadecimal
    @binvalue varbinary(256),
    @hexvalue varchar (514) OUTPUT
AS
DECLARE @charvalue varchar (514)
DECLARE @i int
DECLARE @length int
DECLARE @hexstring char(16)
SELECT @charvalue = '0x'
SELECT @i = 1
SELECT @length = DATALENGTH (@binvalue)
SELECT @hexstring = '0123456789ABCDEF'
WHILE (@i <= @length)
BEGIN
    DECLARE @tempint int
    DECLARE @firstint int
    DECLARE @secondint int
    SELECT @tempint = CONVERT(int, SUBSTRING(@binvalue,@i,1))
    SELECT @firstint = FLOOR(@tempint/16)
    SELECT @secondint = @tempint - (@firstint*16)
    SELECT @charvalue = @charvalue +
        SUBSTRING(@hexstring, @firstint+1, 1) +
        SUBSTRING(@hexstring, @secondint+1, 1)
    SELECT @i = @i + 1
END
SELECT @hexvalue = @charvalue;
GO
```

гах 1 и 2. Выполните его, прежде чем продолжать.

### Сценарий клонирования имени входа

После того как выполнены предварительные условия, пришло время рассмотреть сценарий клонирования. Он разделен на отдельные секции, соответствующие объектам безопасности. В целях удобства чтения каждая секция разделена на описания программного кода, результатов и действий, производимых с результатами, чтобы применить их к целевому «клону».

### Секция 1: имена входа SQL

В этой секции (листинг 3) две хранимые процедуры используются для формирования пяти столбцов, три из которых предназначены только для идентификации, а на два последних (`script_command` и `user_command`) следует обратить особое внимание. `script_command` формируется с помощью вызовов к `sp_help_revlogin`, чтобы создать программный код, при выполнении которого предоставляются команды `CREATE LOGIN`, выполняемые на целевом клоне. `User_command` — динамически формируемый программный код,

**Листинг 2. СОЗДАНИЕ sp\_help\_revlogin**

```

=====
--СОЗДАНИЕ sp_help_revlogin
=====

USE [master]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[sp_help_revlogin] @login_name sysname =
NULL AS
DECLARE @name sysname
DECLARE @type varchar (1)
DECLARE @hasaccess int
DECLARE @denylogin int
DECLARE @is_disabled int
DECLARE @PWD_varbinary varbinary (256)
DECLARE @PWD_string varchar (514)
DECLARE @SID_varbinary varbinary (85)
DECLARE @SID_string varchar (514)
DECLARE @tmpstr varchar (1024)
DECLARE @is_policy_checked varchar (3)
DECLARE @is_expiration_checked varchar (3)

DECLARE @defaultdb sysname

IF (@login_name IS NULL)
    DECLARE login_curs CURSOR FOR

        SELECT p.sid, p.name, p.type, p.is_disabled, p.default_database_
name, l.hasaccess, l.denylogin FROM
sys.server_principals p LEFT JOIN sys.syslogins l
        ON ( l.name = p.name ) WHERE p.type IN ( 'S', 'G', 'U' ) AND p.name
<> 'sa'
ELSE
    DECLARE login_curs CURSOR FOR

        SELECT p.sid, p.name, p.type, p.is_disabled, p.default_database_
name, l.hasaccess, l.denylogin FROM
sys.server_principals p LEFT JOIN sys.syslogins l
        ON ( l.name = p.name ) WHERE p.type IN ( 'S', 'G', 'U' ) AND p.name =
@login_name
OPEN login_curs

FETCH NEXT FROM login_curs INTO @SID_varbinary, @name, @type, @
is_disabled, @defaultdb, @hasaccess, @denylogin
IF (@@fetch_status = -1)
BEGIN
    PRINT 'No login(s) found.'
    CLOSE login_curs
    DEALLOCATE login_curs
    RETURN -1
END
SET @tmpstr = /* sp_help_revlogin script */
PRINT @tmpstr
SET @tmpstr = '** Generated ' + CONVERT (varchar, GETDATE()) + ' on '
+ @@SERVERNAME + '*'
PRINT @tmpstr
PRINT ''
WHILE (@@fetch_status <> -1)
BEGIN
    IF (@@fetch_status <> -2)
    BEGIN
        PRINT ''
        PRINT '====='
        PRINT '-- Login: ' + @name
        PRINT ''
        IF (@type IN ( 'G', 'U'))
        BEGIN
            PRINT '-----'
            PRINT '      -- NT authenticated account/group'
            PRINT ''
        END
        SET @tmpstr = 'CREATE LOGIN ' + QUOTENAME( @name ) + ' FROM
WINDOWS WITH DEFAULT_DATABASE = [' + @defaultdb + ']'
        PRINT @tmpstr
        IF (@type = 'G')
        BEGIN
            PRINT ''
            PRINT '      -- obtain password and sid'
            PRINT '      SET @PWD_varbinary = CAST( LOGINPROPERTY( @name,
>PasswordHash' ) AS varbinary (256) )'
            EXEC sp_hexadecimal @PWD_varbinary, @PWD_string OUT
            EXEC sp_hexadecimal @SID_varbinary, @SID_string OUT
            PRINT ''
            PRINT '      -- obtain password policy state'
            PRINT '      SELECT @is_policy_checked = CASE is_policy_checked WHEN 1
THEN ''ON'' WHEN 0 THEN ''OFF'' ELSE NULL END FROM sys.sql_logins
WHERE name = @name'
            PRINT '      SELECT @is_expiration_checked = CASE is_expiration_checked
WHEN 1 THEN ''ON'' WHEN 0 THEN ''OFF'' ELSE NULL END FROM sys.
sql_logins WHERE name = @name'
            PRINT ''
            SET @tmpstr = 'CREATE LOGIN ' + QUOTENAME( @name ) + '
WITH PASSWORD = ' + @PWD_string + ' HASHED, SID = ' + @SID_string
+ ', DEFAULT_DATABASE = [' + @defaultdb + ']'
            PRINT @tmpstr
            IF ( @is_policy_checked IS NOT NULL )
            BEGIN
                SET @tmpstr = @tmpstr + ', CHECK_POLICY = ' + @is_policy_
checked
                PRINT ''
            END
            IF ( @is_expiration_checked IS NOT NULL )
            BEGIN
                SET @tmpstr = @tmpstr + ', CHECK_EXPIRATION = ' + @
is_expiration_checked
                PRINT ''
            END
            IF (@denylogin = 1)
            BEGIN -- login is denied access
                SET @tmpstr = @tmpstr + '; DENY CONNECT SQL TO ' +
QUOTENAME( @name )
                PRINT ''
            END
            ELSE IF (@hasaccess = 0)
            BEGIN -- login exists but does not have access
                SET @tmpstr = @tmpstr + '; REVOKE CONNECT SQL TO ' +
QUOTENAME( @name )
                PRINT ''
            END
            IF (@is_disabled = 1)
            BEGIN -- login is disabled
                SET @tmpstr = @tmpstr + '; ALTER LOGIN ' + QUOTENAME( @name )
+ ' DISABLE'
                PRINT ''
            END
            PRINT @tmpstr
        END
        FETCH NEXT FROM login_curs INTO @SID_varbinary, @name, @type, @
is_disabled, @defaultdb, @hasaccess, @denylogin
        PRINT ''
        CLOSE login_curs
        DEALLOCATE login_curs
        RETURN 0
    END
END

```

который создаст объект пользователя в базе данных master (по умолчанию,

но при необходимости программный код легко изменить).

Как показано на экране 1, формируются два столбца кода. Результаты

## Листинг 3. Формирование пяти столбцов

```
-- КОМАНДЫ SP_HELP_REVLOGIN И КОМАНДЫ СОЗДАНИЯ ПОЛЬЗОВАТЕЛЯ
-- БАЗЫ ДАННЫХ ПО УМОЛЧАНИЮ
=====

SELECT SP.[principal_id]
    , SP.[name]
    , SP.[default_database_name]
    , 'EXEC [sp_help_revlogin] ' + "" + SP.name + "" + ';' AS script_command
    , 'USE [master]; CREATE USER [' + SP.name + '] FROM LOGIN [' + SP.name + ']' AS user_
command
FROM master.sys.[server_principals] SP
WHERE SP.[type] = 'S'
    AND SP.name != 'sa'
    AND SP.name NOT LIKE ('#%')
ORDER BY SP.[name];
```

script\_command можно скопировать как столбец и одновременно выполнить на исходном сервере,

на котором выполнялся код данной секции. Я всегда использую в таком случае второе окно запроса, так как

## Листинг 4. Запрос к экземпляру SOURCE

```
EXEC [sp_help_revlogin] 'austenford';
EXEC [sp_help_revlogin] 'chriscornell';
EXEC [sp_help_revlogin] 'davidbowie';
EXEC [sp_help_revlogin] 'prince';
EXEC [sp_help_revlogin] 'timford';
EXEC [sp_help_revlogin] 'trevorford';
```

полезно иметь под рукой результаты начального запроса.

На основе приведенного выше примера вновь выполняется запрос к экземпляру SOURCE, приведенный в листинге 4.

В результате будут получены выходные данные, показанные на экране 2. Каждое имя входа SQL имеет блок кода, предоставляющий необходимые параметры для репликации имени входа экземпляра SOURCE

```
/* Сценарий sp_help_revlogin
** Generated Jun 29 2017 10:46PM on TIMF-X1-SEA */
-- Имя входа: austenford
CREATE LOGIN [austenford] WITH PASSWORD = 0x0200FEE6F12D296A9E96D0CAB6B3AF6B0A83F8A6B3B57A403356CFE1F888B8E7D2E20FD7E
4E5FA99742326453A1F291988BBB278443391EEA085EEEED5C7678BE2C5756C7DBF HASHED, SID = 0x4AD4F9838102434D846EFE5CC75B9D01,
DEFAULT_DATABASE = [master], CHECK_POLICY = OFF, CHECK_EXPIRATION = OFF
/* Сценарий sp_help_revlogin
** Generated Jun 29 2017 10:46PM on TIMF-X1-SEA */
-- Имя входа: chriscornell
CREATE LOGIN [chriscornell] WITH PASSWORD = 0x0200559FC679B189BE312D416AD5A2EC60D1F628D9A696291B0BAD3C9633A7CD295D7F29
981481FE2490E4AECEB414E74EC4BEEB0CDD1E0666C7596167C85C75ACCD11888D0 HASHED, SID = 0x4FF2C5D0B93ED145B9F96FC7BA265884,
DEFAULT_DATABASE = [master], CHECK_POLICY = OFF,
CHECK_EXPIRATION = OFF
/* Сценарий sp_help_revlogin
** Generated Jun 29 2017 10:46PM on TIMF-X1-SEA */
-- Имя входа: davidbowie
CREATE LOGIN [davidbowie] WITH PASSWORD = 0x0200BB0728E043983CA0A9E5F1D459EBE43CF3BB1D226FCBD8C71A12E63C3EF3579A97FC23
8FA23AA8B8923D64898C0F521A2D050C4984F7DF1CA516844127DF7808E2BB34CD HASHED, SID = 0x3231CA55367D734DB3BD259C27AE1A15,
DEFAULT_DATABASE = [master], CHECK_POLICY = OFF,
CHECK_EXPIRATION = OFF
/* Сценарий sp_help_revlogin
** Generated Jun 29 2017 10:46PM on TIMF-X1-SEA */
-- Имя входа: prince
CREATE LOGIN [prince] WITH PASSWORD = 0x02001850FE91CDEFC1180A2B512556405AF866B2E1E21E4224E3A885F07694B8AC61A0F99F32DC42
59CFD8B3EFFCB4CF1456C94919D2B8ADFE16C9B370368AA236D6E28C643A HASHED, SID = 0x76AB2554249C29428504129EEE13C6F7, DEFAULT_
DATABASE = [master], CHECK_POLICY = OFF,
CHECK_EXPIRATION = OFF
/* Сценарий sp_help_revlogin
** Generated Jun 29 2017 10:46PM on TIMF-X1-SEA */
-- Имя входа: timford
CREATE LOGIN [timford] WITH PASSWORD = 0x020090D83AF25B9841B1AD32C4EE2729C000AA8BE5B6399FBDCADA9D3E20FCFB0F4CE276C8200
D0E7035B5237CBD7054FB37ACBD90C7399D9EE0BF9AEA71ED54A85AB113C931 HASHED, SID = 0x75D053277EF82D419D20E92D21C2801C,
DEFAULT_DATABASE = [master], CHECK_POLICY = OFF,
CHECK_EXPIRATION = OFF
/* Сценарий sp_help_revlogin
** Generated Jun 29 2017 10:46PM on TIMF-X1-SEA */
-- Имя входа: trevorford
CREATE LOGIN [trevorford] WITH PASSWORD = 0x02003F9AF29FF033CA2C57044B1922A9909BFF5317E976A109B6DFE3662912BE72B303C718FAC
51C7049379490E95A70A113556D991E78BFE920F267196484D8A239492EFE2B HASHED, SID = 0x95C4115AAC64C14EAEE73A7CC6012E4B, DEFAULT_
DATABASE = [master], CHECK_POLICY = OFF,
CHECK_EXPIRATION = OFF
```

## Экран 2. Результаты работы листинга 4

**Листинг 5. Программный код из столбца user\_command**

```
USE [master]; CREATE USER [austenford] FROM LOGIN [austenford];
USE [master]; CREATE USER [chriscornell] FROM LOGIN [chriscornell];
USE [master]; CREATE USER [davidbowie] FROM LOGIN [davidbowie];
USE [master]; CREATE USER [prince] FROM LOGIN [prince];
USE [master]; CREATE USER [timford] FROM LOGIN [timford];
USE [master]; CREATE USER [trevorford] FROM LOGIN [trevorford];
```

**Листинг 6. Создание имен учетных записей Active Directory**

```
-- ИМЕНА ВХОДА АД КОМАНДЫ СОЗДАНИЯ ИМЕН ВХОДА И ПОЛЬЗОВАТЕЛЯ БАЗЫ ДАННЫХ
-- ПО УМОЛЧАНИЮ
=====
SELECT SP.[principal_id]
      , SP.[name]
      , SP.[default_database_name]
      , 'CREATE LOGIN [' + SP.name + ']' FROM WINDOWS WITH DEFAULT_DATABASE = [master];'
AS login_command
      , 'USE [master]; CREATE USER [' + SP.name + ']' FROM LOGIN [' + SP.name + ']' WITH
DEFAULT_SCHEMA=[dbo];' AS user_command
FROM master.sys.[server_principals] SP
WHERE SP.[type] = 'U'
      AND SP.name NOT LIKE 'NT %'
ORDER BY SP.[name];
```

**Листинг 7. Связанные столбцы script\_command и user\_command**

```
CREATE LOGIN [SQLCRUISE\timf] FROM WINDOWS WITH DEFAULT_DATABASE = [master];
CREATE LOGIN [SQLCRUISE\app_service] FROM WINDOWS WITH DEFAULT_DATABASE =
[master];
CREATE LOGIN [SQLCRUISE\comms] FROM WINDOWS WITH DEFAULT_DATABASE = [master];

USE [master]; CREATE USER [SQLCRUISE\timf] FROM LOGIN [SQLCRUISE\timf] WITH DEFAULT_
SCHEMA=[dbo];
USE [master]; CREATE USER [SQLCRUISE\app_service] FROM LOGIN [SQLCRUISE\app_service]
WITH DEFAULT_SCHEMA=[dbo];
USE [master]; CREATE USER [SQLCRUISE\comms] FROM LOGIN [SQLCRUISE\comms] WITH
DEFAULT_SCHEMA=[dbo];
```

**Листинг 8. Создание групп AD**

```
-- ГРУППЫ АД КОМАНДЫ СОЗДАНИЯ ИМЕН ВХОДА И ПОЛЬЗОВАТЕЛЯ БАЗЫ
-- ДАННЫХ ПО УМОЛЧАНИЮ
=====
SELECT SP.[principal_id]
      , SP.[name]
      , SP.[default_database_name]
      , 'CREATE LOGIN [' + SP.name + ']' FROM WINDOWS WITH DEFAULT_DATABASE = [master];'
AS login_command
      , 'USE [master]; CREATE USER [' + SP.name + ']' FROM LOGIN [' + SP.name + ']' WITH
DEFAULT_SCHEMA=[dbo];' AS user_command
FROM master.sys.[server_principals] SP
WHERE SP.[type] = 'G'
      AND SP.name NOT LIKE 'NT %'
ORDER BY SP.[name];
```

на экземпляр TARGET: хешированный пароль, принудительно созданный SID и последующее назначение базе данных master по умол-

чанию; политика пароля и проверка окончания срока действия отключены. Найти и заменить эти значения в сценарии при необходи-

мости можно с помощью простого нажатия клавиш Control+H.

На данном этапе я создаю окно запроса, связанное с экземпляром TARGET, и вставляю приведенный в листинге 4 программный код. За ним следует программный код из столбца user\_command (листинг 5).

При выполнении на экземпляре TARGET вы передаете клонированные имена входа для проверки подлинности SQL из SOURCE в TARGET с совпадающими идентификаторами SID.

Это единственная секция, в которой требуется открыть второе окно для экземпляра SOURCE. Закройте окно запроса, открытое для выполнения сформированных команд sp\_help\_revlogin, и можно перейти к секции 2.

**Секция 2: имена учетных записей AD**

Создать имена учетных записей Active Directory гораздо проще, так как не существует препятствия в виде несоответствия идентификаторов SID. Программный код этой секции, в сущности, представляет собой динамический SQL для доступа к именам учетных записей AD в master.sys.server\_principals для построения сценария CREATE LOGIN ...FROM WINDOWS для любого имени учетной записи AD, фильтрованного, чтобы устраниТЬ зависимые от компьютера локальные имена учетных записей AD (листинг 6).

Связанные столбцы script\_command и user\_command выглядят таким образом после вставки в запрос к моему экземпляру TARGET, как в листинге 7.

**Секция 3: группы AD**

Группы AD обрабатываются почти как имена учетных записей AD. Единственное отличие между двумя категориями — значение типа столбца в master.sys.server\_principals («U» для имен учетных записей AD, «G» для групп AD). Я разделил группы AD и имена учетных записей AD, так как в SQL Server 2005 не разрешалось назначать схему по умолчанию для группы AD. Но SQL Server 2005 более не поддерживается, поэтому необ-

ходимости в этом больше нет. Вы можете без труда изменить сценарий для своей среды (введите = «G» или введите = «U»), как показано в листинге 8.

#### Секция 4: принадлежность к роли сервера

В последней секции рассматривается принадлежность к роли сервера. При переносе баз данных между экземплярами, например для проверки заполнения, обучения или сред UAT, ни одно из прав роли уровня сервера не перемещается вместе с базами данных. Эта секция общего сценария обеспечивает сохранение разрешений уровня сервера на экземпляре TARGET (клонированном) (листинг 9).

В отличие от других секций этот программный код формирует только разовый специальный текст SQL для вызова команды sp\_addrolemember. Результат выглядит примерно следующим образом:

```
EXEC master..sp_addsrvrolemember
N'SQLCRUISEtimf', N'sysadmin';
```

#### Листинг 9. Принадлежность к роли сервера

```
-- ЧЛЕНЫ РОЛИ СЕРВЕРЫ
=====
SELECT R.name AS server_role
    , P.name AS role_member
    , 'EXEC master..sp_addsrvrolemember N' + '''' + P.name + ''' + ', N' + '''' + R.name + ''' + ';
AS command
FROM sys.server_role_members RM
    INNER JOIN sys.server_principals P ON RM.member_principal_id = P.principal_id
    INNER JOIN (SELECT principal_id, name FROM sys.server_principals WHERE type_desc = 'SERVER_ROLE') R
        ON RM.role_principal_id = R.principal_id
WHERE P.name NOT LIKE '#%'
    AND P.name NOT LIKE 'NT %'
    AND P.type_desc <> 'SERVER_ROLE'
    AND P.name NOT IN ('sa')
ORDER BY R.[name], P.[name];
```

Просто выполните этот программный код на экземпляре TARGET, и все готово.

Итак, с помощью данного сценария можно без особого труда перемещать объекты безопасности между экземплярами. В следующей статье мы попытаемся выяснить, насколько изменился мир с появлением PowerShell и каким образом с помо-

щью Powershell можно еще больше упростить решение задачи.

Для удобства чтения полный текст сценария приведен в листинге 10. ◉

**Тим Форд (timothyrford@gmail.com)** — независимый консультант, автор и специалист по данным. С 2000 года работает администратором баз данных в области здравоохранения

#### Листинг 10. Полный текст сценария

```
-- SP_HELP_REVLOGIN COMMANDS AND DEFAULT DB CREATE USER
COMMANDS
=====

-- AD GROUPS CREATE LOGINS AND DEFAULT DB USER COMMANDS
=====
SELECT SP.[principal_id]
    , SP.[name]
    , SP.[default_database_name]
    , 'CREATE LOGIN [' + SP.name + '] FROM WINDOWS WITH DEFAULT_
DATABASE = [master];' AS login_command
    , 'USE [master]; CREATE USER [' + SP.name + '] FROM LOGIN [' +
SP.name + '];' AS user_command
FROM master.sys.[server_principals] SP
WHERE SP.[type] = 'S'
    AND SP.name != 'sa'
    AND SP.name NOT LIKE ('#%')
ORDER BY SP.[name];

-- AD LOGINS CREATE LOGINS AND DEFAULT DB USER COMMANDS
=====
SELECT SP.[principal_id]
    , SP.[name]
    , SP.[default_database_name]
    , 'CREATE LOGIN [' + SP.name + '] FROM WINDOWS WITH DEFAULT_
DATABASE = [master];' AS login_command
    , 'USE [master]; CREATE USER [' + SP.name + '] FROM LOGIN [' +
SP.name + ']' WITH DEFAULT_SCHEMA=[dbo];' AS user_command
FROM master.sys.[server_principals] SP
WHERE SP.[type] = 'U'
    AND SP.name NOT LIKE 'NT %'
ORDER BY SP.[name];

-- SERVER ROLE MEMBERS
=====
SELECT R.name AS server_role
    , P.name AS role_member
    , 'EXEC master..sp_addsrvrolemember N' + '''' + P.name + ''' + ', N' + '''' + R.name + ''' + ';
AS command
FROM sys.server_role_members RM
    INNER JOIN sys.server_principals P ON RM.member_principal_id = P.principal_id
    INNER JOIN (SELECT principal_id, name FROM sys.server_principals WHERE type_desc = 'SERVER_ROLE') R
        ON RM.role_principal_id = R.principal_id
WHERE P.name NOT LIKE '#%'
    AND P.name NOT LIKE 'NT %'
    AND P.type_desc <> 'SERVER_ROLE'
    AND P.name NOT IN ('sa')
ORDER BY R.[name], P.[name];
```

# Своя команда



**Сергей Васин**

Создаем пользовательский объект на основе запроса к Active Directory о действующих компьютерах домена

Зачастую в своей работе мы используем команды, предназначенные для выполнения каких-либо задач на нескольких компьютерах одновременно. При этом в качестве аргументов требуется указать имена соответствующих компьютеров. Для того чтобы получить их имена, мы можем воспользоваться несколькими способами, например хранить их в текстовом файле или csv-файле. Но в этом случае при каждом вызове команд с использованием данного файла мы не можем быть уверены в том, что информация в нем актуальна. Соответственно перед каждым использованием их придется обновлять. Второй способ заключается в том, что мы можем запрашивать информацию непосредственно из Active Directory, однако в этом случае потребуется каждый раз действовать команды из модуля ActiveDirectory и задавать необходимые условия фильтрации. Зачастую, когда мы хотим ограничить возвращаемый результат при помощи каких-то особых условий, нам приходится выполнять запросы несколько раз, исправляя и оптимизируя элементы фильтрации и тем самым создавая дополнительную нагрузку на контроллер домена.

Давайте напишем функцию, которая получала бы информацию обо всех действующих компьютерах домена, то есть с атрибутом Enabled равным True, в виде удобной для дальнейшего использования структуры, а именно пользовательского объекта (назовем его PSCustomObject).

Кроме того, обычно, чтобы иметь возможность запроса к Active Directory, на компьютере устанавливается набор инструментов для удаленного администрирования Remote Server

Administration Tools (RSAT), который, помимо всего прочего, содержит модуль ActiveDirectory. Еще одним вариантом будет использование неявных удаленных подключений, Implicit Remoting, при помощи команд Import-PSSession, Export-PSSession или команды Import-Module с параметром PSSession.

Мы же будем исходить из того, что наша функция не зависит от наличия на компьютере модуля ActiveDirectory в том или ином виде, поэтому для запроса к Active Directory будем использовать интерфейс управления Active Directory Service Interfaces (ADSI). Теперь что касается возвращаемого объекта. Так как мы собираемся использовать результат выполнения функции для обращения к удаленным компьютерам, решим, что возвращаемый объект будет содержать четыре свойства: XP, Seven, Eight и Ten. Несмотря на то что Windows XP уже довольно давно не поддерживается, во многих сетях все еще присутствуют компьютеры с этой операционной системой, так что такое свойство вряд ли будет лишним.

В каждом из названных свойств будет находиться массив пользовательских объектов — компьютеров, сгруппированных по установленной на них операционной системе. Это решение вызвано тем, что зачастую при создании какой-либо команды, предназначенной для выполнения на удаленных компьютерах, нам приходится обращать внимание на используемую ими операционную систему.

Например, в Windows XP максимальной поддерживаемой версией PowerShell является версия 2.0. Это означает, что на таких компьютерах мы не сможем использовать некоторые команды и кон-

структур, например такие, как \$Array.PropertyOfElementsInTheArray, то есть мы не сможем обратиться к свойствам элементов массива напрямую через объект массива.

В Windows 7, независимо от установленной версии PowerShell, отсутствует пространство имен WMI ROOT/StandardCimv2, и, таким образом, нам будут недоступны модули для управления сетью, такие как NetTCPIP и NetAdapter.

Поэтому в качестве средства первичной фильтрации мы будем использовать операционную систему. Однако мы сделаем так, что все атрибуты объекта компьютера ActiveDirectory будут присутствовать в создаваемых нами пользовательских объектах, поэтому дальнейшую фильтрацию мы сможем осуществить сами. При этом мы будем работать уже с локальным объектом, не задействуя контроллеры домена сверх необходимого.

Кроме того, для тех случаев, когда нам нужно обратиться ко всем компьютерам или сделать выборку, не обращая внимания на операционную систему, мы создадим еще одно свойство нашего объекта и назовем его All. Оно будет содержать все объекты компьютеров из четырех приведенных выше свойств. Причем, чтобы упростить функцию, создаваться это свойство будет не в ней, а с помощью файла типов (о чем я расскажу чуть позже).

Еще одной особенностью нашей функции станет то, что возвращать она будет только объекты клиентских компьютеров, так как обращаться к серверам мы, скорее всего, будем на основе совсем других критериев, нежели операционная система. Также, забегая вперед, скажу, что кроме самой функции нам понадобятся файлы типов — \*.types.ps1 xml и формата — \*.format.ps1 xml, так что реализация нашей идеи в конечном итоге будет представлена в виде модуля. Теперь определимся с именами.

## Get-sthLDAPComputersBy OperatingSystem

Как мы знаем, имя функции должно отражать ее предназначение.

### Листинг 1. Код функции sthTools.psm1

```
function Get-sthLDAPComputersByOperatingSystem
{
    Param()

    $os = [ordered]@{
        XP = '&(objectClass=computer)(OperatingSystem=Windows XP*)(!userAccountControl:1.2.840.113556.1.4.803:=2)'
        Seven = '&(objectClass=computer)(OperatingSystem=Windows 7*)(!userAccountControl:1.2.840.113556.1.4.803:=2)'
        Eight = '&(objectClass=computer)(OperatingSystem=Windows 8*)(!userAccountControl:1.2.840.113556.1.4.803:=2)'
        Ten = '&(objectClass=computer)(OperatingSystem=Windows 10*)(!userAccountControl:1.2.840.113556.1.4.803:=2)'
    }

    $RootDSE = [ADSI]::LDAP://RootDSE
    $NC = $RootDSE.defaultNamingContext
    $SearchRoot = 'LDAP://' + $NC

    $Searcher = New-Object -TypeName System.DirectoryServices.DirectorySearcher
    $Searcher.SearchRoot = $SearchRoot

    $ComputersByOperatingSystem = @{}

    foreach ($osname in $os.Keys)
    {
        $Searcher.Filter = $os["$osname"]
        $SearchResult = $Searcher.FindAll()

        $CompsFamily = @()
        $CompsFamily = foreach ($s in $($SearchResult | Sort-Object -Property Path))
        {
            $Comp = @{}
            $s.Properties.GetEnumerator() | ForEach-Object -Process { $Comp.Add($PSItem.Key, $($PSItem.Value)) }
            [PSCustomObject]$Comp | Add-Member -TypeName sth.Computer -PassThru
        }

        $ComputersByOperatingSystem.Add($osname, $CompsFamily)
    }

    [PSCustomObject]$ComputersByOperatingSystem | Add-Member -TypeName sth.ComputersByOperatingSystem -PassThru
}
```

Кроме того, чтобы снизить вероятность совпадения имени вашей функции с какой-либо другой, рекомендуется после дефиса указывать некий префикс. Я в своих функциях использую sth, а вы можете указать любой другой.

Таким образом, наша функция будет называться Get-sthLDAPComputersByOperatingSystem, и, так как мы решили, что конечным результатом работы будет модуль, нам нужно определиться и с его именем. Назовем его sthTools. Следовательно, код нашей функции будет находиться в файле sthTools.psm1 (листинг 1), который в свою очередь будет расположен в каталоге sthTools.

Перейдем к определению функции. Параметров у нее не будет, поэтому блок Param оставляем пустым.

## Ldap Filter

В целях поддержания чистоты кода я предлагаю все строки фильтров сохранить в переменной и при выполнении запросов обращаться уже к ней. Так как мы используем ADSI, условия фильтрации должны быть представлены в формате LDAP. Что нам понадобится? Во-первых, нам нужны только объекты компьютеров, поэтому одним из условий станет то, что свойство objectClass должно содержать 'computer'. Во-вторых, для опреде-

ления версии операционной системы можно было бы воспользоваться атрибутом `operatingSystemVersion`, но, так как мы хотим, чтобы функция возвращала только клиентские компьютеры, в этом случае нам потребовалось бы также обратиться к атрибуту `operatingSystem`, где указывается название операционной системы. Поэтому в целях минимизации количества условий фильтра мы будем задействовать только атрибут `operatingSystem`, этого вполне достаточно для реализации наших намерений.

Теперь что касается действующих компьютеров. Мы решили, что возвращать функция будет только компьютеры, у которых атрибут `Enabled` установлен в `True`. Вот только присутствует этот атрибут исключительно при использовании команд модуля `ActiveDirectory`, поскольку является искусственной конструкцией. А в действительности активен объект или нет, определяется вторым разрядом `ADS_UF_ACCOUNTDISABLE`, атрибута `userAccountControl`. В том, что название этого атрибута говорит о его принадлежности к пользователю, нет ничего удивительного, так как родительским классом `computer` является именно `user`.

Здесь мы используем следующую конструкцию:

```
! userAccountControl:  
 1.2.840.113556.1.4.803:=2.
```

Давайте разберем, что означает каждая ее часть.

`userAccountControl` — это атрибут, со значением которого мы и будем работать.

Конструкция `:1.2.840.113556.1.4.803:` между именем атрибута и знаком равенства означает, что при анализе значения атрибута мы будем использовать поразрядное ‘И’.

Набор цифр между двумя двоеточиями называется `ruleOID`, и приведенное выше значение — это `LDAP_MATCHING_RULE_BIT_AND`.

К слову, поразрядное ‘ИЛИ’ обозначается как `1.2.840.113556.1.4.804 — LDAP_MATCHING_RULE_BIT_OR`. Цифра 2 — это значение, с которым мы будем сравнивать содержимое атрибута `userAccountControl`.

Так как нас интересует исключительно второй бит, мы указываем здесь его десятичное значение.

Восклицательный знак означает инверсию. Таким образом, все это выражение получает объекты, у которых второй разряд атрибута `userAccountControl` установлен в ноль.

Каждое из этих трех условий заключено в скобки, равно как и все выражение целиком. Символ `&` в начале, еще одно обозначение оператора AND, говорит о том, что при использовании данного фильтра будут возвращены только те объекты, которые удовлетворяют каждому из трех указанных условий. Если бы нам было достаточно выполнения любого из условий, мы могли бы воспользоваться символом `|`, то есть оператором OR.

Все четыре фильтра мы сохраним в переменной `$os` в виде хеш-таблицы (хотя в данном случае, если быть более точным, `OrderedDictionary`) под именами, соответствующими операционным системам. Указание `[ordered]` позволяет нам придать некоторую определенность последующим запросам, так как по умолчанию хеш-таблицы не гарантируют, что порядок их элементов будет соответствовать тому, который был задан при их создании.

### DirectorySearcher

Далее, чтобы нам не приходилось каждый раз указывать домен, в котором мы находимся, сделаем так, чтобы функция находила эту информацию сама. Для этого получим объект корня дерева службы каталогов, он же `RootDSE`. Сохраним его в одноименной переменной.

Запросив значение свойства `defaultNamingContext` этого объекта, мы получим имя домена, в котором и будем производить поиск нужных нам объектов компьютеров. Его мы сохраним в переменной `$NC`. В свою очередь, добавив содержимое переменной `$NC` к строке `‘LDAP://’`, мы получим значение, которое чуть позже присвоим свойству `SearchRoot` объекта `DirectorySearcher`.

Теперь создадим объект типа `System.DirectoryServices.DirectorySearcher`, который и будем использовать для поиска нужных объектов компьютеров. Для этого мы воспользуемся командой `New-Object`. Полученный в результате ее выполнения объект сохраним в переменной `$Searcher`. В качестве области поиска, которая определяется значением свойства `SearchRoot` объекта `DirectorySearcher`, мы укажем переменную `$SearchRoot`, содержащую сконструированную ранее строку. Затем мы создадим переменную `$ComputersByOperatingSystem`, значением которой будет пустая хеш-таблица. Ее мы будем использовать для хранения массивов объектов компьютеров, сгруппированных по используемой операционной системе.

### ForEach

Напомню, что мы создали четыре фильтра LDAP в переменной `$os`, и теперь нам нужно поочередно назначить каждый из них свойству `Filter` объекта `DirectorySearcher`, находящегося в переменной `$Searcher`, и выполнить поиск объектов, соответствующих этому фильтру. Для этого воспользуемся конструкцией `foreach`. В скобках, следующих за `foreach`, укажем, что имя каждой записи хеш-таблицы (`$os.keys`), а это `XP`, `Seven`, `Eight` и `Ten`, будет назначаться переменной `$osname`. Внутри фигурных скобок мы назначаем соответствующую строку фильтра свойству `Filter` объекта `DirectorySearcher`, расположенного в переменной `$Searcher`, и затем вызываем метод `FindAll()`. Результат его выполнения мы сохраним в переменной `$SearchResult`.

Теперь в переменной `$SearchResult` у нас находится коллекция объектов `System.DirectoryServices SearchResult`. Так как наша цель — создать удобный в обращении пользовательский объект, подобный вид представления компьютеров нам не подойдет. Следовательно, на основе полученной информации потребуется создать отдельный пользовательский объект для каждого компьютера. Кроме того, будет

удобнее, если объекты компьютеров будут представлены в отсортированном виде.

### **sth.Computer**

Для начала определим переменную \$CompsFamily со значением в виде пустого массива. Затем создадим выражение, в котором значением этой переменной будет результат выполнения еще одной конструкции foreach. В ней в круглых скобках мы укажем, что переменной \$s будет назначаться каждый объект из отсортированного по значению свойства Path массива \$SearchResults по очереди. Для того чтобы операция сортировки происходила до назначения объектов переменной \$s, команду \$SearchResult | Sort-Object -Property Path мы указываем в виде подвыражения (subexpression). Для этого мы заключаем ее в скобки с символом \$ в начале.

Напомню, что мы решили создать пользовательский объект для каждого компьютера. Начнем с того, что создадим переменную \$Comp в виде пустой хеш-таблицы. Далее, для того чтобы получить имена и значения свойств объекта System.DirectoryServices.ResultPropertyCollection, который находится в свойстве Properties переменной \$s, потребуется воспользоваться методом GetEnumerator(). Его результаты мы передаем команде ForEach-Object, где имя и значение каждого свойства добавляются в хеш-таблицу переменной \$Comp. Причем, когда мы получаем значение очередного свойства, мы снова используем подвыражение (subexpression), на этот раз для того, чтобы представить каждое значение в виде соответствующего ему типа данных, а не в виде объекта System.DirectoryServices.ResultPropertyValueCollection.

После этого на основе находящейся в переменной \$Comp хеш-таблицы мы создаем пользовательский объект. Хотя создание пользовательского объекта может производиться несколькими способами, мы задействуем один из самых удобных с точки зрения объема кода — это указание [PSCustomObject] перед именем переменной, содержащей хеш-таблицу.

Как уже говорилось, мы собираемся использовать файлы типов и формата, поэтому нам нужно назначить своим объектам какой-то тип, отличающий их от других объектов. Для этого мы воспользуемся командой Add-Member. В качестве значения параметра TypeName укажем sth.Computer. Параметр PassThrough нужен для того, чтобы измененный объект передавался дальше по конвейеру, так как по умолчанию команда Add-Member этого не делает.

Так как это последняя команда в блоке foreach, это приведет к тому, что созданный объект будет добавлен в массив, находящийся в переменной \$CompsFamily. После того как обработает внутренний foreach, мы добавляем группу компьютеров в хеш-таблицу, находящуюся в переменной \$ComputersByOperatingSystem.

### **sth.ComputersByOperatingSystem**

Таким образом, после четырех проходов и обработки всех фильтров в переменной \$os переменная \$ComputersByOperatingSystem будет включать хеш-таблицу, содержащую четыре элемента. С переменной \$ComputersByOperatingSystem мы поступим таким же образом, как и с переменной \$Comp, и создадим на ее основе пользовательский объект, с той лишь разницей, что в качестве типа укажем sth.ComputersByOperatingSystem.

Так как команда Add-Member является последней в функции, результат ее выполнения и будет результатом функции. Таким образом, мы получим объект типа sth.ComputersByOperatingSystem, содержащий свойства XP, Seven, Eight и Ten, каждое из которых включает массив объектов типа sth.Computer, представляющих компьютеры, использующие соответствующую имени свойства операционную систему.

### **sthTools.types.ps1 xml**

Теперь поговорим о том, зачем нам понадобились файлы типов и формата. Начнем с типов (листинг 2). Если мы внимательно присмотримся к свойствам объектов sth.Computer, то заметим, что такие атрибуты, как objectGUID и objectSID, представ-

лены в виде массива байтов, то есть именно в том виде, в каком они хранятся в Active Directory. Так как это достаточно важные элементы, стоит их представить в более привычном виде. Что касается objectGUID, то здесь не требуется каких-то особых действий, и привести его к типу GUID мы можем следующим образом: [GUID]\$ByteArray.

С атрибутом objectSID все несколько сложнее. Для преобразования его в привычную форму в виде строки нам придется задействовать сценарий. Мы не будем убирать из объектов компьютеров значения objectGUID и objectSID в их текущем виде, а добавим их видоизмененные значения в виде дополнительных свойств — GUID и SID соответственно. Для этого создадим файл типов sthTools.types.ps1 xml.

Для типа данных sth.Computer мы добавим два элемента ScriptProperty. Первый из них — GUID, и в качестве сценария, вычисляющего значение создаваемого элемента, мы укажем [guid]\$This.objectGUID. Переменная \$This олицетворяет текущий объект, а его свойство objectGUID — как раз тот самый массив байтов, который мы собираемся представить в виде объекта System.Guid. Второй элемент — это свойство SID, представленное в виде сценария, задача которого — сконвертировать массив байтов в строку SID. Не буду производить его детальный анализ. Скажу только, что структуру массива байтов и значение каждого из них вы можете найти на сайте MSDN в статье по адресу: <https://msdn.microsoft.com/en-us/library/gg465313.aspx>.

Итак, с определением дополнительных свойств объектов sth.Computer мы закончили. Переходим теперь ко второму типу объектов — sth.ComputersByOperatingSystem.

Как уже говорилось выше, нам бы хотелось, чтобы все объекты компьютеров, вне зависимости от используемой операционной системы, были доступны через свойство All. Это мы обеспечим опять же при помощи элемента ScriptProperty файла sthTools.types.ps1 xml. Сценарием, используемым для вычисления значения этого

**Листинг 2. Файл описания типов**

```

<?xml version="1.0" encoding="utf-8" ?>
<Types>
<Type>
<Name>sth.Computer</Name>
<Members>
<ScriptProperty>
<Name>GUID</Name>
<GetScriptBlock>
$Result = @"
"@
$Lines = $This.Xp.count, $This.Seven.count, $This.Eight.count,
$This.Ten.count | Sort-Object -Descending | Select-Object -First 1

$LeftTemplate = ""
if ($This.XP) {$LeftTemplate += "{0,-30}"}
if ($This.Seven) {$LeftTemplate += "[1,-30]"}
if ($This.Eight) {$LeftTemplate += "[2,-30]"}
if ($This.Ten) {$LeftTemplate += "[3,-30]"}

$Result += "$LeftTemplate" -f "Windows XP Computers", "Windows
7 Computers", "Windows 8 Computers", "Windows 10 Computers"
$Result += "`n$LeftTemplate" -f "-----", "-----",
-----, "-----", "-----"

for ($i = 0; $i -lt $Lines; $i++)
{
    $RightTemplate = @()
    if ($This.XP) {$RightTemplate += $This.XP[$i].name} else
    {$RightTemplate += $Null}
    if ($This.Seven) {$RightTemplate += $This.Seven[$i].name} else
    {$RightTemplate += $Null}
    if ($This.Eight) {$RightTemplate += $This.Eight[$i].name} else
    {$RightTemplate += $Null}
    if ($This.Ten) {$RightTemplate += $This.Ten[$i].name} else
    {$RightTemplate += $Null}

    $Result += "`n$LeftTemplate" -f $RightTemplate
}
return $Result
</GetScriptBlock>
</ScriptProperty>
</Members>
</Type>
<Type>
<Name>sth.ComputersByOperatingSystem</Name>
<Members>
<ScriptProperty>
<Name>All</Name>
<GetScriptBlock>
$This.XP + $This.Seven + $This.Eight + $This.Ten
</GetScriptBlock>
</ScriptProperty>
</Members>
</Type>
</Types>

```

свойства, будет: \$This.XP + \$This.Seven + \$This.Eight + \$This.Ten. Еще необходимо учесть, что, если мы сейчас обратимся к полученному в результате выполнения функции объекту, его стандартное представление будет достаточно трудным для понимания. Поскольку это пользовательский объект, для него еще не существует никаких заданных представлений, поэтому он попытается вывести все свое содержимое целиком.

И хотя представления задаются в файлах формата (\*.format.ps1 xml), использование в нем сценариев является нежелательным, поэтому предлагаю создать еще одно свойство, которое будет содержать информацию об именах компьютеров в виде четырех столбцов, соответствующих определенной операционной системе, а уже потом в файле

формата мы укажем это свойство как представление по умолчанию. Назовем его Summary.

Чем занимается сценарий, используемый для вычисления значения свойства Summary, так это проверкой наличия компьютеров в каждой из четырех групп и созданием столбцов с указанием имени используемой операционной системы и соответствующих ей имен компьютеров. Результат представляется в виде here-string, особого рода строки, которая, являясь, по сути, одним элементом, может состоять из нескольких отдельных строк.

**sthTools.format.ps1 xml**

Теперь перейдем к файлу форматов (листинг 3). Начнем опять же с объектов sth.Computer. Так как объекты компьютеров содержат

относительно большое количество атрибутов, было бы удобнее, если бы по умолчанию выводились только те, что используются чаще всего.

Так как мнений относительно того, какие именно атрибуты должны быть включены в представление по умолчанию, может быть несколько, воспользуемся уже существующим набором — тем, что используется командами Get-ADComputer: DistinguishedName, DNSHostName, Enabled, Name, ObjectClass, ObjectGUID, SID.

С поправкой на то, что GUID и SID будут представлены созданными нами ранее свойствами, а Enabled будет отсутствовать, так как сама идея нашей функции — это получение информации о действующих объектах компьютеров.

Что касается объекта sth.ComputersByOperatingSystem, то мы уже решили, что по умолчанию

будет выводиться значение свойства Summary. Поскольку присутствие имени свойства в выводе нам в данном случае не нужно, мы воспользуемся элементом CustomControl.

### sthTools.psd1

Для того чтобы объединить три файла (sthTools.psm1, sthTools.types.ps1 xml и sthTools.format.ps1 xml) в модуль, нам нужно создать файл sthTools.psd1. Сделать это можно и вручную, но удобнее задействовать команду New-ModuleManifest. Сделаем мы это таким образом:

```
New-ModuleManifest
    -Path sthTools.psd1-RootModule
    sthTools.psm1-TypeToProcess
    sthTools.types.ps1 xml
    -FormatsToProcess sthTools.format.ps1 xml
```

Теперь, поместив каталог sthTools и содержащиеся в нем четыре файла в одно из мест, указанных в переменной среды PSModulePath, к примеру: C:\Program Files\Windows PowerShell\Modules, мы сможем обращаться к функции Get-sthLDAPComputersByOperatingSystem без необходимости предварительного импорта модуля sthTools.

### Execute

Использовать функцию Get-sthLDAPComputersByOperatingSystem вы можете по-разному. Для начала стоит сохранить результат ее выполнения в переменной, например так:

```
$Computers = Get-sthLDAPComputers
    ByOperatingSystem
```

Затем, чтобы получить объекты всех действующих компьютеров, использующих операционную систему Windows 10, введите:

```
$Computers.Ten
```

Имена этих компьютеров вы можете получить следующим образом:

```
$Computers.Ten.Name
```

Для того чтобы выполнить какое-либо действие на этих компьютерах, введите команду:

```
Invoke-Command -ComputerName
    $Computers.Ten.name -ScriptBlock
        {какое-либо действие}
```

При этом нам никто не запрещает использовать средства фильтрации, для того чтобы сузить круг компьютеров до тех, что отвечают только каким-то особым требованиям.

### Листинг 3. Файл форматов

```
<configuration>
    <ViewDefinitions>
        <View>
            <Name>sth.Computer</Name>
            <ViewSelectedBy>
                <TypeName>sth.Computer</TypeName>
            </ViewSelectedBy>
            <ListControl>
                <ListEntries>
                    <ListEntry>
                        <ListItems>
                            <ListItem>
                                <PropertyName>DistinguishedName</PropertyName>
                            </ListItem>
                            <ListItem>
                                <PropertyName>DnsHostName</PropertyName>
                            </ListItem>
                            <ListItem>
                                <PropertyName>Name</PropertyName>
                            </ListItem>
                            <ListItem>
                                <PropertyName>ObjectClass</PropertyName>
                            </ListItem>
                            <ListItem>
                                <PropertyName>GUID</PropertyName>
                            </ListItem>
                            <ListItem>
                                <PropertyName>SamAccountName</PropertyName>
                            </ListItem>
                            <ListItem>
                                <PropertyName>SID</PropertyName>
                            </ListItem>
                        </ListItems>
                    </ListEntry>
                </ListEntries>
            </ListControl>
        </View>
        <View>
            <Name>sth.ComputersByOperatingSystem</Name>
            <ViewSelectedBy>
                <TypeName>sth.ComputersByOperatingSystem</TypeName>
            </ViewSelectedBy>
            <CustomControl>
                <CustomEntries>
                    <CustomEntry>
                        <CustomItem>
                            <ExpressionBinding>
                                <EnumerateCollection/>
                                <PropertyName>Summary</PropertyName>
                            </ExpressionBinding>
                        </CustomItem>
                    </CustomEntries>
                </CustomControl>
            </View>
```

Например, для того чтобы выполнить некую команду на компьютерах с операционной системой Windows 10, входящих в группу some\_group, можно поступить следующим образом:

```
Invoke-Command -ComputerName
    $($Computers.Ten |? memberof
        -match some_group |% name)
```

# Один ответ на два вопроса



**Сергей Васин**

Внедряем Just Enough Administration

**К**огда администратору по каким-либо причинам требуется передать часть своих задач другому сотруднику, перед ним обычно встает два вопроса: как сделать так, чтобы при выполнении новых задач этот сотрудник с точки зрения прав и возможностей был ничуть не хуже администратора, а с другой стороны, как добиться, чтобы он был ничуть не хуже администратора исключительно при решении делегированных задач. Технология Just Enough Administration, или JEA, дает ответы на оба вопроса.

## Что такое JEA

JEA — это технология, появившаяся в языке написания сценариев PowerShell версии 5.0, которая позволяет настроить параметры сеанса, Session Configuration, часто упоминаемые как конечная точка, Endpoint, таким образом, чтобы пользователь, не являющийся администратором на компьютере, к которому подключается, установив с ним соединение с использованием указанных параметров, получил на данном компьютере административные права. Делается это при помощи виртуальных учетных записей, которые создаются при подключении пользователя и прекращают свое существование с окончанием сессии. Таким образом, эти виртуальные учетные записи не могут быть использованы для чего-либо другого, например для подключения по протоколу удаленного доступа RDP или для управления через консоль MMC.

Здесь стоит упомянуть, что и до JEA вы могли указать учетную запись, от имени которой будет действовать под-

ключившийся пользователь. Для этого в команде Register-PSSessionConfiguration используется параметр RunAsCredential. Однако, во-первых, в этом случае указываемая в параметре учетная запись уже должна обладать административными правами. А во-вторых, каждый подключающийся пользователь будет действовать от имени одной и той же учетной записи, что приведет к некоторым сложностям при попытке проверки, кто из пользователей чем занимался.

Но вернемся к виртуальной учетной записи. Для того чтобы ее задействовать, мы устанавливаем значение параметра RunAsVirtualAccount в команде New-PSSessionConfigurationFile как \$True.

По умолчанию при подключении к рабочей станции или автономному серверу (имеется в виду к серверу, который не является контроллером домена) виртуальная учетная запись, от имени которой действует пользователь, будет членом локальной группы Administrators. В случае подключения к контроллеру домена эта виртуальная учетная запись будет входить в группу Domain Admins. В какие группы должна входить виртуальная учетная запись, вы можете указать самостоятельно при помощи параметра RunAsVirtualAccountGroups в той же команде New-PSSessionConfigurationFile. Стоит упомянуть, что указание какой-либо группы в качестве значения параметра RunAsVirtualAccountGroups приведет к тому, что группы по умолчанию (Administrators или Domain Admins) применяться уже не будут. Поэтому, если вам нужно, чтобы виртуальная учетная запись входила в группы, например Administrators и Some\_Special\_Group, в качестве значения параметра

RunAsVirtualAccountGroups, необходимо указать обе.

Несмотря на то что группа Domain Admins предоставляет широкие возможности на уровне домена, виртуальная учетная запись, от имени которой будет действовать пользователь, в своих административных правах будет ограничена компьютером, на котором она создана. Таким образом, если в процессе работы пользователю нужно будет обратиться по сети к какому-либо другому компьютеру, это обращение будет происходить не от имени члена группы Domain Admins (в случае контроллера домена), а от имени учетной записи данного компьютера. Это справедливо и для рабочих станций и автономных серверов.

Если же ваш метод применения JEA предполагает взаимодействие с другими компьютерами, то здесь можно воспользоваться появившейся в среде управления Windows Management Framework 5.1 возможностью использования вместо виртуальной учетной записи так называемой групповой учетной записи управляемой службы, Group Managed Service Account (gMSA). В этом случае значение параметра RunAsVirtualAccount мы указываем как \$False (или не указываем вообще), а в параметре GroupManagedServiceAccount задаем соответствующую учетную запись. Понятно, что компьютеры, к которым мы будем обращаться, должны иметь право ее использовать.

### RestrictedRemoteServer

Вторая важная задача — это ограничение возможностей пользователя, получившего при подключении административные права. Как известно, существует два основных способа контроля доступности команд, привилегий, возможностей и т. д. Первый — это разрешить все, а потом запрещать нежелательное. Второй — запретить все и затем разрешить то, что необходимо. Так как сама идея JEA состоит в том, чтобы пользователи обладали минимальным набором возможностей, очевидно, что в данном случае актуален второй подход.

По умолчанию при задании настроек сессии (конечной точки) пользователю доступен весь набор команд, и в этом случае его возможности ограничиваются только его правами, которые в свою очередь в большинстве случаев определяются членством в группах. Однако при создании настроек сессии мы можем ограничить набор доступных команд до минимально необходимого. Таким образом, доступны будут только те команды, которые нужны для функционирования самой удаленной сессии. Это следующие команды: Clear-Host, Exit-PSSession, Get-Command, Get-FormatData, Get-Help, Measure-Object, Out-Default, Select-Object и псевдонимы (алиасы) некоторых из них: clear, cls, exsn, gcm, measure, select.

Чтобы реализовать сказанное выше, в качестве значения параметра SessionType в команде New-PSSessionConfigurationFile нужно указать RestrictedRemoteServer. В этом случае все, что потребуется сверх приведенного набора, нужно определять явным образом. Кроме того, установка параметра SessionType в значение RestrictedRemoteServer неявным образом влияет на значение еще одного параметра, а именно LanguageMode, задавая его значение как NoLanguage. Это приводит к тому, что все языковые конструкции в пределах удаленной сессии будут недоступны.

Если задачи пользователя все-таки потребуют какого-либо взаимодействия с возможностями языка PowerShell, вы можете организовать это двумя способами. Во-первых, можно определить функции, которые будут доступны пользователю в рамках сессии в файле возможностей ролей, Role Capabilities, о чем мы поговорим чуть позже. Во-вторых, вы можете создать набор функций на локальном компьютере пользователя, предусмотрев в нем все необходимые возможности для взаимодействия с удаленной сессией. Таким образом, пользователь будет работать с локальными функциями, а те в свою очередь будут задействовать удаленные подключения для выполнения требуемых задач. Теперь, запретив все, кроме команд, необ-

ходимых для работы самой сессии, перейдем к разрешениям.

### Role Capabilities

Что именно будет доступно конкретному пользователю, определяется при помощи файлов описания возможностей ролей, Role Capabilities. Они представляют собой текстовые файлы с расширением .psrc. Каждый файл определяет какую-либо роль, которая в свою очередь определяет, что именно будет доступно пользователю (или группе), которому эта роль будет назначена, а именно: модули, псевдонимы, команды, функции, внешние команды и провайдеры. Создание файлов .psrc происходит при помощи команды New-PSRoleCapabilityFile, а сопоставление групп пользователей тем или иным ролям — при помощи параметра RoleDefinitions в команде New-PSSessionConfigurationFile.

Поговорим о файлах .psrc. С файлами настройки сессии .pssc все более или менее понятно. Они используются в процессе регистрации настроек сессии Session Configuration, а именно указываются в качестве значения параметра Path в команде Register-PSSessionConfiguration и служат своего рода шаблоном для создаваемой настройки. Таким образом, они нужны только в момент регистрации этой настройки и могут находиться в любом месте, на которое будет ссылаться параметр Path. Что же касается файлов описания возможностей ролей Role Capabilities, то здесь все несколько иначе. Они тоже представляют собой текстовые файлы, но имеют расширение .psrc и создаются командой New-PSRoleCapabilityFile. Еще более важное отличие состоит в следующем: для того, чтобы содержащиеся в таких файлах настройки ролей применялись к создаваемым сессиям, файлы .psrc должны находиться в определенных местах, а точнее, в папке RoleCapabilities какого-либо модуля.

Здесь может возникнуть некоторая путаница: можно предположить, что файл описания возможностей ролей, размещенный в каталоге RoleCapabilities опре-

деленного модуля, будет влиять только на команды данного модуля. Однако это не так. Файл .psrc должен находиться в указанном месте для того, чтобы PowerShell мог его найти, но это никоим образом не ограничивает пределы его использования.

Более того, имя роли определяется именем файла описания возможностей ролей: к примеру, роль Maintenance определяется файлом Maintenance.pscc. Это приводит к тому, что файлы с одним и тем же именем, расположенные в разных модулях, будут считаться описанием одной и той же роли. Но PowerShell в этом случае не будет пытаться создать какую-то общую структуру из всех имеющихся файлов, а просто выберет первый встретившийся файл (из файлов с одинаковыми именами) и будет его использовать в качестве источника информации об определенной роли. Понятно, что взгляд PowerShell на то, какой файл будет считаться первым, может не совпадать с вашим.

Поскольку Microsoft не рекомендует без необходимости трогать файлы в каталоге \$PSHOME (обычно C:\Windows\System32\WindowsPowerShell\v1.0), стоит рассмотреть вариант хранения файлов настроек в каком-либо собственном модуле, если таковой имеется. Если же в своей среде вы не используете собственноручно созданные модули, можно создать модуль исключительно для хранения файлов .pscc, что мы и сделаем чуть позже. Что еще стоит рассмотреть, так это ограничение прав доступа к файлам описания возможностей ролей. Так как эти файлы содержат всю информацию о возможностях и полномочиях пользователей, их случайное изменение может предоставить подключающемуся пользователю привилегии, которых у него быть не должно. Минимальный набор прав должен включать в себя возможность доступа к ним учетной записи Local System, так как это необходимо для функционирования самой технологии JEA. Все остальные права — на ваше усмотрение, в соответствии с принятыми в организации стандартами.

### TranscriptDirectory

Еще неплохим вариантом будет включение протоколирования всех выполненных пользователями команд, а также полученных ими результатов. Сделать это можно при помощи параметра TranscriptDirectory в команде New-PSSessionConfigurationFile. В качестве значения мы указываем каталог, где и будут храниться журналы. Опять же, для функционирования процесса протоколирования учетная запись Local System должна иметь доступ к указанной папке.

### Пример настройки

Теперь давайте попробуем реализовать технологию Just Enough Administration на практике. Предположим, у нас есть две группы пользователей: DNS\_Managers и AD\_Managers. В них входят пользователи, которые будут выполнять некоторые административные функции по управлению сервером DNS и службой каталогов Active Directory соответственно. Причем никто из них не входит в административные группы, такие как Administrators, Domain Admins или DNSAdmins, и поэтому административными правами не обладает. Далее будем исходить из того, что для членов группы DNS\_Managers мы хотим предоставить возможность задействовать все команды модуля DNSServer и утилиту командной строки dnscmd.exe.

Группе AD\_Managers мы хотим предоставить возможность применять без ограничений команды Get-ADForest и Get-ADDomain. Кроме того, им должна быть доступна команда Get-ADComputer, причем использовать они смогут только параметр Identity, а в качестве его значения указывать исключительно имена компьютеров, начинающиеся с 'cl'. Чтобы продемонстрировать еще одну возможность управления доступностью команд и их параметров, сделаем так, что им будет доступна команда Get-ADUser, где они без ограничений смогут использовать параметр Filter, а для параметра Properties будут доступны значения только Description и Department. Кроме

того, предоставим им доступ к утилитам dsget.exe и dsquery.exe.

Для начала нам понадобятся каталоги для временного хранения файлов настройки сессии, а также для хранения журналов работы подключающихся к серверу пользователей. Пусть это будут C:\WorkBench и C:\WhatAreYouDoing, соответственно.

```
New-Item-ItemTypeDirectory-Path: \\WorkBench
```

```
New-Item-ItemTypeDirectory-Path: \\WhatAreYouDoing
```

### New-PSSessionConfigurationFile

Далее нам нужно создать файл настроек сессии. Сделаем мы это при помощи команды New-PSSessionConfigurationFile. Однако, поскольку параметров достаточно много и их значения порой представляют собой несколько строк, предлагаю воспользоваться технологией подстановок splatting и сначала создать в переменной \$PSSessionConfigurationFile хеш-таблицу с именами параметров и их значениями, а уже затем вызывать команду для создания файла.

Итак, мы будем использовать следующие параметры. Для начала — путь и имя файла. Пусть это будет 'C:\WorkBench\Management.pssc'. Далее — SessionType. Как уже говорилось выше, его значением должно быть 'RestrictedRemoteServer'. Еще один обязательный в нашем случае параметр — RunAsVirtualAccount, со значением \$True. Параметр TranscriptDirectory определяет, где мы хотим хранить журналы. А также нам понадобится параметр RoleDefinitions, который назначает конкретные роли пользователям, в зависимости от их членства в группах. Формат его определения следующий: @{'группа' = @{RoleCapabilities = 'соответствующая\_ей\_роль'} }.

Причем нам никто не запрещает задать пользователю или группе несколько ролей, указав их через запятую, например: @{'группа' = @{RoleCapabilities = 'роль\_1', 'роль\_2'}}. Это приведет к тому, что пользователи будут располагать возможностями обеих ролей. К примеру, если 'роль\_1' позволяет задействовать только команду

Get-Item, а ‘роль\_2’ — только Get-DNSServer, то пользователи будут иметь возможность запустить и Get-Item, и Get-DNSServer.

Однако вернемся к нашему файлу Management.pssc. Выше мы говорили о том, что при указании значения параметра SessionType как ‘RestrictedRemoteServer’ доступными для использования будут только несколько команд, необходимых для функционирования самой сессии. И хотя полномочия пользователей относительно доступных команд мы будем определять в файлах описания возможностей ролей, предлагаю кое-что сделать на уровне настройки сессии.

А именно, если предполагается, что пользователи будут работать с удаленными сессиями интерактивно, то есть без применения заранее написанных функций, соответствующих определенным задачам, им пригодится возможность использования сочетаний клавиш Tab и Ctrl+пробел для автоматического завершения имен команд и параметров, а также получения возможных вариантов. По умолчанию в сессиях RestrictedRemoteServer эти функции отсутствуют.

Давайте предположим, что администраторам сервера DNS и Active Directory пригодилась бы данная функция, и сделаем ее доступной для обеих групп. Здесь мы могли бы воспользоваться файлами описания возможностей ролей для каждой из групп, но, так как эти сочетания клавиш будут применяться всеми пользователями настроек сессии, определим их доступность на данном уровне.

Что касается технической реализации, то мы можем воспользоваться параметром VisibleCmdlets как при вызове команды New-PSSessionConfigurationFile для создания настройки сессий, так и при вызове New-PSRoleCapabilityFile для создания файла описания возможностей ролей. Отличие в том, что команды, заданные при вызове New-PSSessionConfigurationFile, будут доступны пользователям всех групп, которым сопоставлены какие-либо роли в пределах конкретной настройки сессии.

### Листинг 1. Настройки сессии

```
$PSSessionConfigurationFile= @{
    Path = 'C:\WorkBench\Management.pssc'
    SessionType = 'RestrictedRemoteServer'
    RunAsVirtualAccount = $True
    TranscriptDirectory = 'c:\WhatAreYouDoing'
    RoleDefinitions = @{'domain_name\DNS_Managers' = @{RoleCapabilities = 'DNS_Administration'};
        'domain_name\AD_Managers' = @{RoleCapabilities = 'AD_Administration'}}
    VisibleCmdlets = 'TabExpansion2'
}
```

З а сочетания клавиш Tab и Ctrl + пробел отвечает функция TabExpansion2, поэтому именно ее мы укажем в параметре VisibleCmdlets при создании настройки сессии (листинг 1).

Теперь для того, чтобы указать, что мы собираемся использовать технологию splatting, а не просто задаем переменную \$PSSessionConfigurationFile в качестве значения для какого-либо параметра, при вызове команды New-PSSessionConfigurationFile символ \$ перед ее именем мы заменим на @.

```
New-PSSessionConfigurationFile@  
PSSessionConfigurationFile
```

### Register-PSSessionConfiguration

Теперь давайте зарегистрируем настройки сессии. Сделаем мы это при помощи команды Register-PSSessionConfiguration. В качестве параметров задействуем Name (его значение будет являться именем набора настроек сессии и будет применяться пользователями при подключении (назовем его Management), а также Path, где мы укажем путь к только что созданному файлу Management.pssc.

Тот факт, что у нас еще не созданы файлы описания возможностей ролей, не помешает нам создать конечную точку, хотя для указанных групп сделает ее использование невозможным.

Как мы помним, по умолчанию возможность подключения к конечным точкам имеют пользователи групп ‘Administrators’ и ‘RemoteManagementUsers’, и для того, чтобы это изменить, требуется задействовать параметры SecurityDescriptorSddl или ShowSecurityDescriptorUI. Однако если файл \*.pssc содержит сопоставление ролей определенным

группам, эти группы автоматически получают право подключения к регистрируемым настройкам.

```
$PSSessionConfiguration= @{
    Name = 'Management'
    Path = 'C:\WorkBench\Management.pssc'
}
Register-PSSessionConfiguration@  
PSSessionConfiguration
```

### New-PSRoleCapabilityFile

Итак, мы создали файл настроек сессии, в котором сопоставили группу DNS\_Managers роли DNS\_Administration, а группу AD\_Managers — роли AD\_Administration. Теперь следовало бы определить полномочия этих ролей, но сначала стоит решить, где эти файлы ролей будут храниться.

Как уже говорилось выше, если у вас нет какого-либо собственного модуля, где вы бы могли хранить эти файлы, можно создать пустой модуль (не содержащий каких бы то ни было функций и других элементов) с целью использования исключительно для этих целей. Создадим мы его в предлагаемом для хранения собственных модулей каталоге C:\ProgramFiles\WindowsPowerShell\Modules и назовем, к примеру, Maintenance (листинг 2).

Для начала мы сохраним путь к корневому каталогу нашего нового модуля в переменной \$ModulePath, а затем создаем этот каталог. В нем мы создаем пустой файл Maintenance.psm1, куда в случае необходимости сможем добавить наши функции.

Далее мы создаем файл описания, или манифест, для модуля, где как единственный параметр используем RootModule. В качестве его значения мы указываем только что созданный файл Maintenance.psm1.

### Листинг 2. Создание модуля для хранения файлов настроек сессии

```
$ModulePath= 'C:\Program Files\WindowsPowerShell\Modules\Maintenance'  
New-Item-ItemTypeDirectory-Path$ModulePath  
New-Item-ItemTypeFile-Path $($ModulePath+ '\Maintenance.psm1')  
  
New-ModuleManifest-Path $($ModulePath+ '\Maintenance.psd1')-RootModuleMaintenance.psm1  
  
$RoleCapabilitiesPath= $ModulePath+ '\RoleCapabilities'  
New-Item-ItemTypeDirectory-Path$RoleCapabilitiesPath
```

### Листинг 3. Создание файла описания возможностей ролей

```
$PSRoleCapability_DNS_Administration= @{  
Path = $($RoleCapabilitiesPath+ '\DNS_Administration.psrc')  
VisibleAliases = 'Export-DnsServerTrustAnchor', 'Get-DnsServerRRL', 'Set-DnsServerRRL'  
VisibleCmdlets = 'DNServer'*'  
VisibleExternalCommands = 'C:\Windows\system32\dnscmd.exe'  
}  
  
New-PSRoleCapabilityFile@PSRoleCapability_DNS_Administration
```

### Листинг 4. Описание роли для администрирования Active Directory

```
$PSRoleCapability_AD_Administration= @{  
Path = $($RoleCapabilitiesPath+ '\AD_Administration.psrc')  
VisibleCmdlets = 'Get-ADDomain', 'Get-ADForest',  
@{Name = 'Get-ADUser'; Parameters = @{Name = 'Filter'}, @{Name = 'Properties'; ValidateSet= 'Description', 'Department'}},  
@{Name = 'Get-ADComputer'; Parameters = @{Name = 'Identity'; ValidatePattern= 'cl.*'}}  
VisibleExternalCommands = 'C:\Windows\system32\dsget.exe', 'C:\Windows\system32\dsquery.exe'  
}  
  
New-PSRoleCapabilityFile@PSRoleCapability_AD_Administration
```

Затем в каталоге нашего нового модуля мы создаем папку RoleCapabilities, где будут храниться файлы описания возможностей ролей, и можем перейти непосредственно к их созданию. Причем действовать мы будем таким же образом, как и при создании файлов настройки сессии, то есть с применением технологии splatting.

#### DNS\_Administration.psrc

Как уже говорилось выше, пользователи групп, которым будет назначена роль DNS\_Administration, должны иметь возможность задействовать все команды модуля DNSServer, а также утилиту командной строки dnsclient.exe. И здесь, казалось бы, мы могли бы использовать параметр ModulesToImport, для того чтобы предоставить пользователям доступ ко всем командам модуля DNSServer. Однако, если мы это сделаем, то увидим, что ни одной ожидаемой команды для

управления сервером DNS в сессии нет. Почему? Все дело в том, что чуть раньше, при создании файла настроек сессии, мы использовали параметр VisibleCmdlets, для того чтобы присутствие в сессии функции TabExpansion2 сделало возможным использование сочетаний клавиш Tab и Ctrl+пробел.

Как только мы задействуем какой-либо из параметров Visible\* (VisibleAliases, VisibleCmdlets, VisibleFunctions, VisibleExternalCommands или VisibleProviders), будь то в команде New-PSSessionConfigurationFile или New-PSRoleCapabilityFile, PowerShell предполагает, что наши намерения состоят в использовании более гранулярного подхода к доступности пользователю сессии определенных команд, и ориентируется на соответствующие параметры.

Об этом важно помнить не только при создании новых настроек сессии, но и при их редактировании. Потому что при их создании мы точно знаем,

чего хотим добиться, и заметим, если итоговый результат не соответствует нашим ожиданиям.

Когда мы редактируем уже имеющиеся настройки, мы можем не знать, чего именно хотели добиться их авторы, и привести их к такому виду, что сессия будет способна выполнять новые задачи, но перестанет подходить для тех задач, для которых она изначально создавалась.

Например, если ни в настройках сессии, ни в существующих файлах описания возможностей ролей не используются параметры Visible\*, то все модули, указанные в ModulesToImport, пользователям будут доступны. Но, если мы создадим новый файл ролей, где эти параметры будут использоваться, и добавим конкретную роль какой-либо группе, которой ранее уже была назначена некая роль, то пользователи этой группы потеряют возможность задействовать любые команды из модулей, определенных в параметре ModulesToImport. Вернемся к созданию файла описания возможностей ролей. Вместо параметра ModulesToImport мы можем использовать параметр VisibleCmdlets со значением 'DNServer\*'. Это приведет к тому, что все команды модуля DNSServer будут доступны. Однако в случае необходимости мы вполне можем указать в качестве значения 'DnsServer\Get-\*' или 'DnsServer\*-DnsServer'.

Кроме команд, модуль DNSServer содержит еще несколько псевдонимов, или алиасов. Раз уж мы решили, что пользователям должно быть доступно все содержимое модуля DNSServer, то предоставим им возможность применять и алиасы тоже. Поступить с ними так же, как с командами, а именно указать их следующим образом — VisibleAliases = 'DNSServer\*' — у нас не получится. Потребуется указать имя каждого из них по отдельности. Кроме того, нам нужно предоставить пользователям возможность задействовать утилиту командной строки dnsclient.exe. Ее полный путь мы укажем в параметре VisibleExternalCommands (листинг 3).

О чем еще стоит сказать, так это о том, что если вы уже занимались

настройкой удаленных подключений, то можете подумать: зачем нам создавать роли, если эти же ограничения, касающиеся доступности команд, мы можем задать и при создании настройки сессий?

Действительно, такой подход вполне возможен; только в этом случае доступность команд определяется на уровне настройки сессии. То есть, если вам требуется несколько наборов команд для разных групп, придется создать несколько наборов настроек, по одному для каждой группы пользователей. С помощью JEA это можно сделать в пределах единственного набора настроек.

### **AD\_Administration.psrc**

Теперь перейдем к роли для администрирования Active Directory. Здесь мы решили ограничить возможность использования не только команд, но и их параметров. В частности, команды Get-ADForest и Get-ADDomain можно будет применять без ограничений, а в командах Get-ADUser и Get-ADCComputer из всех возможных параметров мы оставим только несколько.

Среди параметров команды Get-ADUser будут доступны: Filter без каких-либо ограничений и Properties, в качестве значений которого можно будет указать только ‘Description’ и ‘Department’. Для параметра Properties мы воспользуемся выражением ValidateSet, которое позволяет явным образом задать возможные значения.

Что касается команды Get-ADCComputer, то мы решили, что доступен будет единственный параметр Identity, и его значения должны начинаться с символов ‘cl’. Здесь нам пригодится выражение ValidatePattern, которое позволяет указать, как должно выглядеть значение параметра с использованием регулярных выражений. И так же, как и в предыдущем случае, мы укажем полный путь к утилитам dsget.exe и dsquery.exe в параметре VisibleExternalCommands (листинг 4). Здесь стоит сказать, что на общие параметры, такие как ErrorAction или WarningVariable, подобные ограничения не влияют, поэтому

### **Листинг 5. Изменение файла описания роли**

```
$PSRoleCapability_DNS_Administration=@{
Path = $($RoleCapabilitiesPath+ '\DNS_Administration.psrc')
VisibleAliases = 'Export-DnsServerTrustAnchor', 'Get-DnsServerRRL', 'Set-DnsServerRRL'
VisibleCmdlets = 'DNSServer\*'
VisibleExternalCommands = 'C:\Windows\system32\dnscmd.exe'
FunctionDefinition = @{[Name = 'Test-DNSService'; ScriptBlock = {if($dns= Get-Service-NameDNS){Write-Output-InputObject»Service status: OK», $dns}}}
}
New-PSRoleCapabilityFile@PSRoleCapability_DNS_Administration
```

мутакие параметры всегда будут доступны.

### **Get-PSSessionCapability**

После того как мы создали все нужные файлы описания возможностей ролей, было бы неплохо проверить, ко всем ли необходимым командам они предоставляют доступ. Естественно, сделать это можно, установив подключение к созданным нами настройкам сессии с применением учетных данных пользователя, введя команду:

Get-Command- CommandTypeAll

Установка значения параметра CommandType в ‘All’ нужна для того, чтобы в дополнение к командам и функциям была выведена информация и об алиасах и внешних компонентах, таких как утилиты командной строки и файлы сценариев. Однако удобнее это сделать, введя следующую команду непосредственно на сервере:

Get-PSSessionCapability-

ConfigurationNameconfiguration\_name-Usernameuser\_name

В этом случае мы получим информацию обо всех доступных пользователю командах без необходимости установки подключения и использования его учетных данных.

### **New-PSSession**

Теперь пользователи могут подключиться с использованием созданной нами настройки при помощи команд New-PSSession и Enter-PSSession.

```
$session= New-PSSession-
ComputerNamecomputer_name-ConfigurationNameManagement
Enter-PSSession-Session$session
```

### **Изменение файлов описания возможностей ролей**

Если по ходу работы мы придем к выводу, что роли в их нынеш-

нем виде не соответствуют нашим представлениям о прекрасном, мы вполне можем это исправить. Сделать это можно как при помощи редактирования файла описания возможностей ролей вручную, так и посредством запуска команды New-PSRoleCapabilityFile с указанием нужных параметров. При этом изменения вступят в силу уже при следующем подключении, перезагрузка службы WinRM не потребуется.

Предположим, что в созданной ранее роли DNS\_Administration нам очень не хватает функции для проверки состояния службы сервера DNS. Для того чтобы это изменить, возьмем уже использовавшуюся нами команду и добавим туда параметр FunctionDefinition. Кроме иллюстрации того, что в файле возможностей ролей мы можем задавать доступные пользователям функции, это еще и возможность убедиться, что в них мы можем задействовать компоненты языка, которые в сессиях типа ‘RestrictedRemoteServer’ отсутствуют. Например, конструкции if и переменные, как показано в листинге 5.

Итак, в этой статье мы познакомились с технологией Just Enough Administration, рассмотрели вопросы, касающиеся ее внедрения и использования, а также изучили возможности, которые она предоставляет в области организаций и отслеживания действий при администрировании серверов в сети предприятия. 

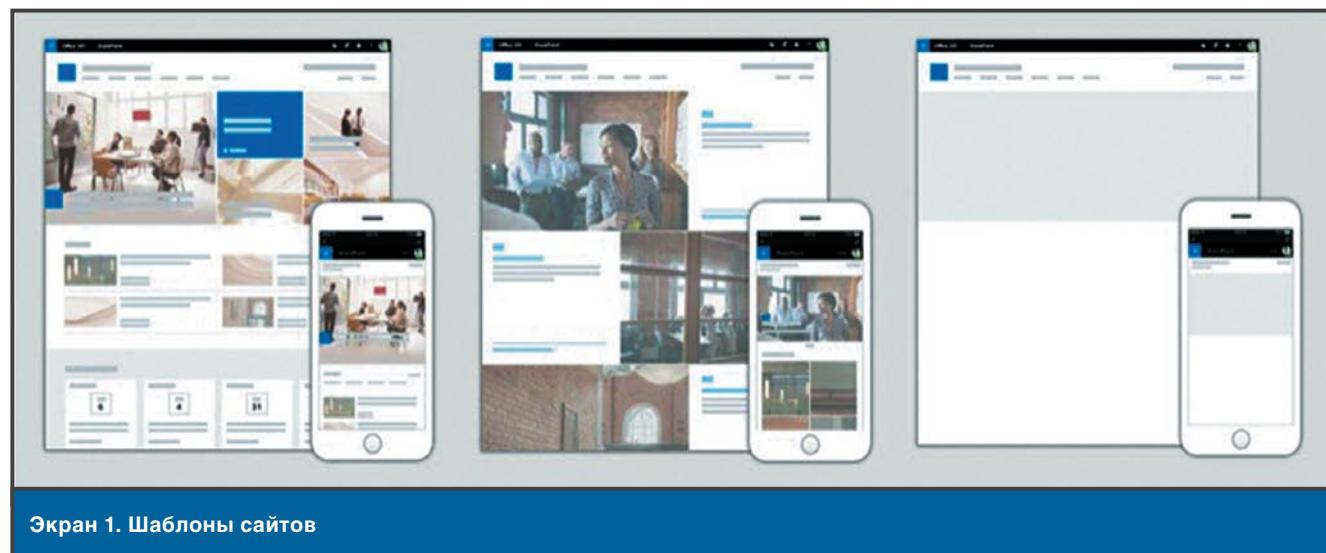
# Коммуникационные сайты

**Лиам Клири**

**Н**а недавнем мероприятии Virtual Summit компанией Microsoft были представлены Communication Sites. О коммуникационных сайтах говорили уже давно, но до этого события их никто не видел. На мероприятии состоялась официальная демонстрация Communication Sites, которые можно создавать на основе базовых шаблонов, успешно работающих на PC, MAC и SharePoint Mobile App, как показано на экране 1. Пользовательский интерфейс очень прост: вы перетаскиваете компоненты на холст веб-страницы. Структуру страницы можно изменить, по собственному желанию выбирая компоновку столбцов для страниц сайта (экран 2). На страницы можно добавлять веб-части с широкими возможностями, которые позволяют не только показывать введенную статическую информацию, но и выводить на страницу другие

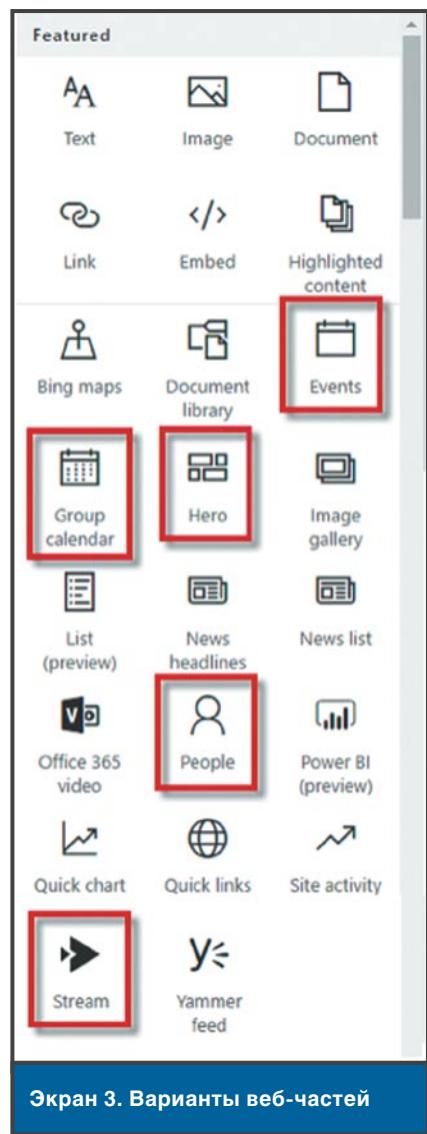
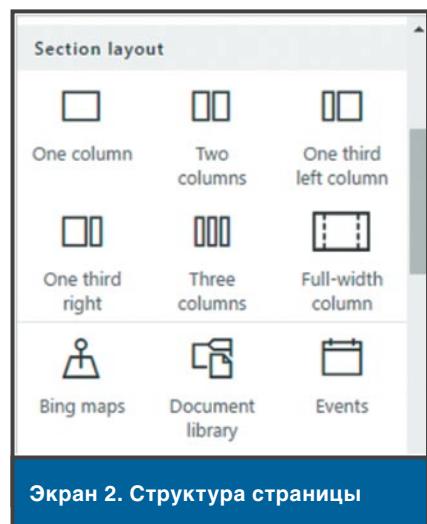
службы, в том числе Events («События»), Group Calendar («Календарь группы»), веб-часть Него («Герой»), People («Люди») и даже контент из службы Microsoft Stream (экран 3).

Коммуникационные сайты позволяют создавать повторяющиеся обновления и обмениваться ими не только по электронной почте. В страницы можно встраивать документы, видеоматериалы и динамически извлекать данные в режиме реального времени из Office 365, как показано на экране 4. Могу с уверенностью сказать, что вы будете постоянно использовать такие сайты. Структура сайтов и страниц такова, что они воспроизводятся в SharePoint Mobile App и функционируют почти так же, как на обычном рабочем столе. Кроме того, представители Microsoft утверждают, что предусмотрены и другие функции, благодаря которым использовать эти сайты будет еще удобнее.



## Вид домашней страницы и подстраниц

**Макеты полной ширины.** Веб-части Hero и Image могут быть помещены в макет раздела, который охватывает страницу слева направо, позволяя



выделить наиболее важную информацию.

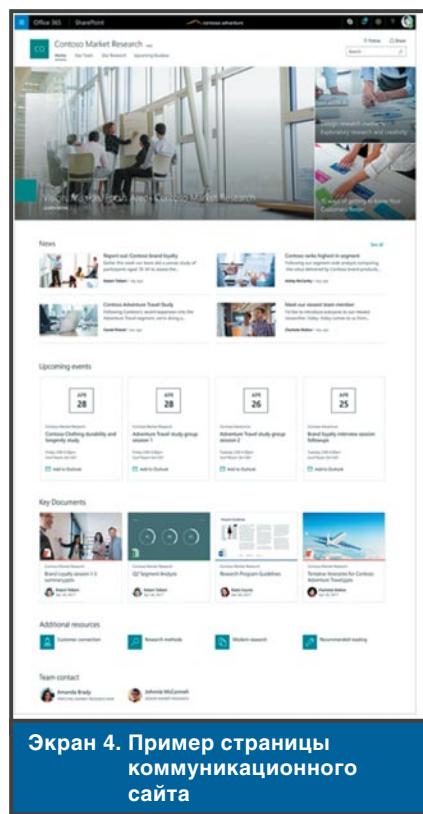
ногого сайта — не просто ссылка; это визуальное информативное пред-

Коммуникационные сайты позволяют создавать повторяющиеся обновления и обмениваться ими не только по электронной почте

**Область заголовка с настраиваемым изображением.** Визуально представляет домашнюю страницу, новости и подстраницу с графическим изображением и заголовком. Вы определяете наиболее важную часть изображения и можете оформить ее по своему вкусу как в Интернете, так и в мобильной среде.

**Комментарии на страницах.** Каждая новостная статья и страница может иметь собственный набор комментариев. Можно воспользоваться веб-частью Yammer для организации дискуссий и целенаправленных откликов, чтобы вовлечь посетителей в обсуждение одного сообщения и контента на странице. Все эти действия выполняются в контексте страницы.

**Обмен новостями по электронной почте.** Обмен новостями по электронной почте из коммуникацион-



ставление, которое добавляет контекст как к сообщению электронной почты, так и самой новостной статье. Получатели почтового сообщения увидят миниатюрное изображение, заголовок, описание и дополнительное сообщение от отправителя.

## Динамическое извлечение и отображение данных, документов и информации через веб-части

**Power BI и Microsoft Stream.** Составляйте интерактивные отчеты с использованием веб-части Power BI и встраивайте одиночные видеоролики или полные каналы из Microsoft Stream — единого центра в Office 365 для управления видеоматериалами между компаниями. Сегодня как Power BI, так и Microsoft Stream доступны широкой аудитории пользователей.

**Поддержка GIF.** Добавляя веб-часть Image в новостную статью или страницу, вы можете вставлять в макет анимированные рисунки GIF.

**Новые страницы See all («Видеть все»).** Если содержимого у вас больше, чем можно показать с помощью веб-частей Highlighted content и Site activity, щелкните See all, чтобы перейти к полностраничному представлению для просмотра всего контента и активности.

**Веб-часть Updated News.** Представьте свои новости с помощью нескольких макетов, чтобы было удобно выделять важные элементы. Вы можете использовать макет по умолчанию Top story, просматривать новости списком или размещать их рядом друг с другом.



Лiam Клири — архитектор решений, имеет сертификат Microsoft MVP

# Сборки для участников программы тестирования

Windows Insider

## Ричард Хэй

**В** прошлом году на конференции Microsoft Ignite в Атланте Гейб Аул отметил, что участников программы тестирования Windows Insider, подписанных на режим быстрых обновлений, намного меньше, чем тех, кто выбрал режим медленного получения тестовых сборок.

Признаюсь, это заявление меня удивило. Казалось бы, ситуация должна быть прямо противоположной, ведь «быстрое обновление» означает более ранний доступ к новым возможностям и исправлениям ошибок.

Учитывая график выпуска сборок для режима медленного обновления, участники Windows Insider должны вот-вот получить сборку обновления Windows 10 Creators Update. Последний раз они получали сборку 14986 еще в прошлом году — 14 декабря 2016 года, то есть более ста дней назад. Временной зазор между 14986 и 15048 велик, в то время как участники, подписанные на режим быстрых обновлений, за этот же период получили девять новых сборок. Последние сборки содержали мало нововведений, но все же там

было кое-что, с чем остальным еще только предстоит познакомиться.

В ожидании новой сборки участники «режима медленного обновления» программы Windows Insider могут пока ознакомиться с приведенным ниже списком новшеств, которые они обнаружат в сборке 15048.

### Сборка 15002 (выпущена 9 января 2017 г.)

Важное обновление, содержащее усовершенствования следующих аспектов операционной системы:

- Microsoft Edge.
- Меню «Пуск» и Shell.
- Windows Ink.
- Cortana.
- Специальные возможности.
- Защитник Windows.
- Параметры Windows.
- Сенсорная панель.
- «Синий экран» стал «зеленым экраном» для сборок, тестируемых в рамках программы Insider.
- Виртуальные машины.
- Центр обновлений Windows.

## Сборка 15007 (выпущена 12 января 2017 г.)

### Microsoft Edge

- Возможность делиться отложенными вкладками.
- Импорт данных из других браузеров.
- Возможность запуска загруженного файла программы, минуя процесс сохранения.
- Интерфейс веб-заметок теперь реализован в стиле Windows Ink.

### Темы Windows 10

- В приложении «Параметры Windows» скоро появится ссылка для прямой загрузки тем из Магазина Windows.

### Cortana

- Возможность быстро вернуться к работе, к тому месту, где она была прервана, на других устройствах.

### Центр уведомлений

- В уведомления встроен индикатор хода выполнения.

### Пользовательский интерфейс и средства навигации

- Усовершенствована полоса прокрутки в приложениях на платформе UWP, которая теперь предусматривает более плавную прокрутку и занимает меньше места на экране.
- Сочетание клавиш ALT + N позволяет сделать снимок области экрана.

### Windows Hello

- Визуальное руководство, фиксирующее ваше лицо в реальном времени.
- Усовершенствованный индикатор хода выполнения.

## Сборка 15014 (выпущена 19 января 2017 г.)

Microsoft официально объявила, что выпуск панели My People отложен до следующего крупного обновления, которое ожидается предстоящей осенью. Сборка включает следующие новшества:

- Возможность покупки и чтения электронных книг из Microsoft Edge. В Магазине Windows теперь есть раздел, посвященный электронным книгам, а в Microsoft Edge появилась специальная вкладка для электронной библиотеки в разделе Hub.
- Возможность выбора цветовой темы в настройках персонализации.

- Функция автоматического освобождения пространства за счет удаления временных и неиспользуемых файлов, включаемая в «Параметрах Windows».
- Объединение настроек параметров Wi-Fi.

## Сборка 15019 (выпущена 27 января 2017 г.)

- Встроенная функция потоковой передачи Beam.

- Новая категория «Игры» в «Параметрах Windows».
- Игровой режим.
- Полнотонные цветные смайлы в Microsoft Edge.
- Обновление первоначальной настройки операционной системы (OOBE) (параметры конфиденциальности, настройка Windows Hello, обновленный голос, субтитры).
- Функция Blue Light («Синий цвет») переименована в Night Light («Ночной свет»).
- Изменение размера окна виртуальной машины в Hyper-V.
- При загрузке приложения или игры в Центре уведомлений теперь отображается полоса индикатора процесса загрузки.
- В «Параметрах Windows» добавлен раздел «Устранение неполадок».

## Сборка 15025 (выпущена 1 февраля 2017 г.)

- Поддержка шрифта Брайля в «Экранном дикторе».
- Коллекции в Центре отзывов, в которых отзывы сгруппированы по проблемам и темам.
- Улучшения функции Night Light («Ночной свет»).

## Сборка 15031 (выпущена 8 февраля 2017 г.)

- Режим компактного наложения («картинка в картинке» для приложений).
- Функция динамической блокировки с телефона по соединению Bluetooth.
- Новый значок «Поделиться».

## Сборка 15042 (выпущена 24 февраля 2017 г.)

- Новая анимация Cortana в режиме OOBE.

- Запрос на разрешение подключения флеш-накопителя. В Microsoft Edge рядом с адресной строкой добавлено окно, уведомляющее пользователя о том, что флеш-накопитель заблокирован, и его запуск требуется подтвердить щелчком мыши.
- Усовершенствованный интерфейс чтения электронных книг в Microsoft Edge.

## Сборка 15046 (выпущена 28 февраля 2017 г.)

- Значок «Зашитника Windows» на панели задач позволяет увидеть текущий статус защиты, а раздел «Приложения и браузер» теперь работает полноценно.
- Изменена функция, позволяющая продолжить работу с того места, где вы остановились. Теперь приложения и файлы, а также ранее открытые веб-сайты можно увидеть из Microsoft Edge, тогда как раньше это работало только в Центре уведомлений.
- Усовершенствованы возможности перевода.
- Значок игр приведен в соответствие с другими значками в «Параметрах Windows».
- В области управления установкой приложений теперь можно ограничивать установку приложений, разрешая установку только приложений из Магазина Windows.

## Сборка 15048 (выпущена 3 марта 2017 г.)

- Новых возможностей нет.
- Следует отметить уменьшение числа новых функций и усовершенствований существующих возможностей в последних сборках, что естественно, учитывая выпуск обновления Creators Update для всех пользователей Windows 10. Теперь все группы разработки трудятся в основном над исправлением ошибок и обеспечением высокой производительности.

Ричард Хэй ([winobs@outlook.com](mailto:winobs@outlook.com)) имеет звание Microsoft MVP в категории Windows Operating System с 2010 года

# Средства очистки жесткого диска в Windows 10 Redstone 3

**Ричард Хэй**

**Я** все держу в цифровом виде — почту, документы, старые веб-сайты, где я работал, и прочие файлы. Если мой жесткий диск достаточно велик, чтобы хранить все это без проблем, то почему бы нет? Недавно Лиза Шмайсер написала статью об очистке цифрового пространства от всего ненужного под названием «Уборка в цифровом пространстве» (опубликована в этом же номере журнала). И как раз предстоящее четвертое основное обновление компонентов Windows 10, сегодня известное как Fall Creators Update, а при тестировании называвшееся Redstone 3, содержит дополнение, расширяющее возможности компонента Storage Sense. Оно может оказаться полезным, по крайней мере, при очистке одной из областей, о которых говорится в упомянутой статье, а именно папки «Загрузки».

Впрочем, необходимо сделать оговорку: Windows 10 Redstone 3 — предварительный выпуск, и нельзя гарантировать, что все добавленное в него на этапе тестирования останется в окончательном варианте продукта. Не вижу, однако, причин, почему бы такому простому инструменту обслуживания не остаться в окончательном выпуске. Эта новая возможность Storage Sense в текущей сборке Redstone 3 под номером 16199 работает следующим образом.

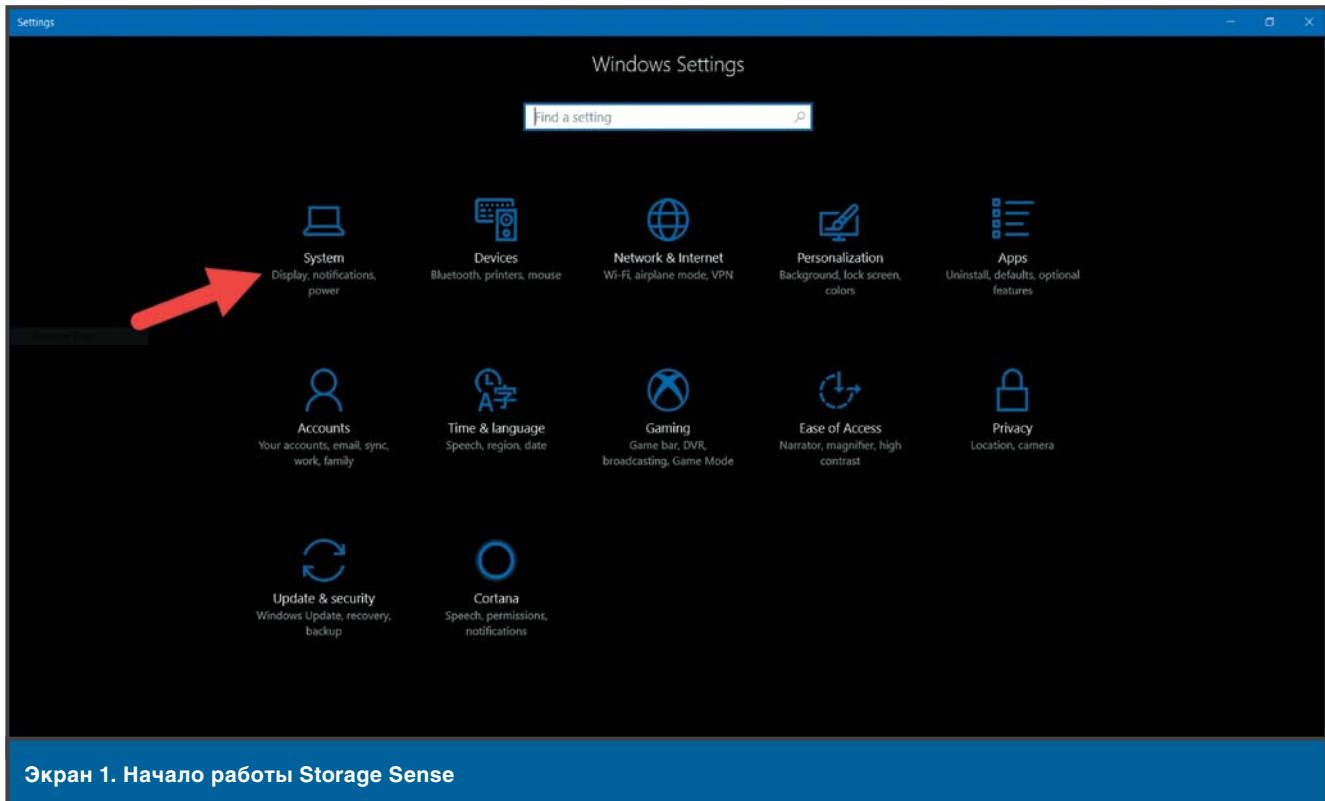
## Шаг 1

Откройте «Параметры Windows» (клавиша Windows + I) и выберите «Система», как показано на экране 1.

## Шаг 2

На левой панели выберите «Хранилище», затем нажмите «Изменить способ освобождения пространства» (экран 2).



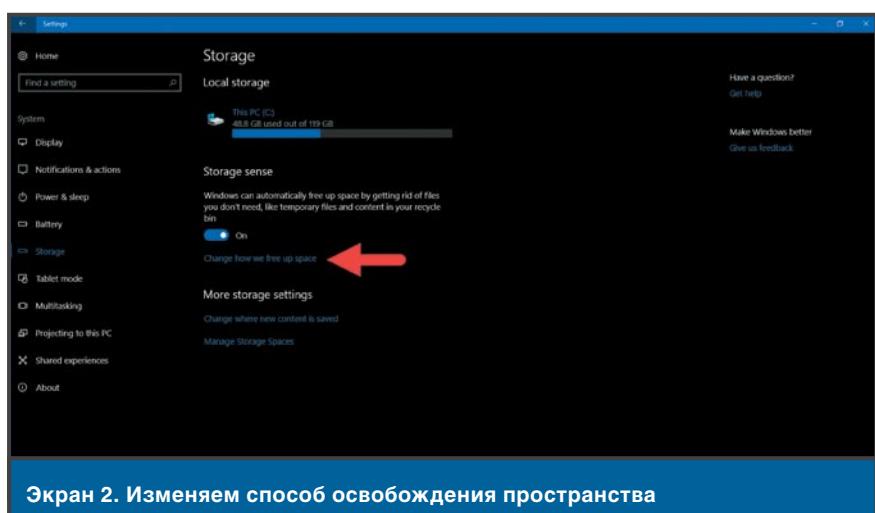


Экран 1. Начало работы Storage Sense

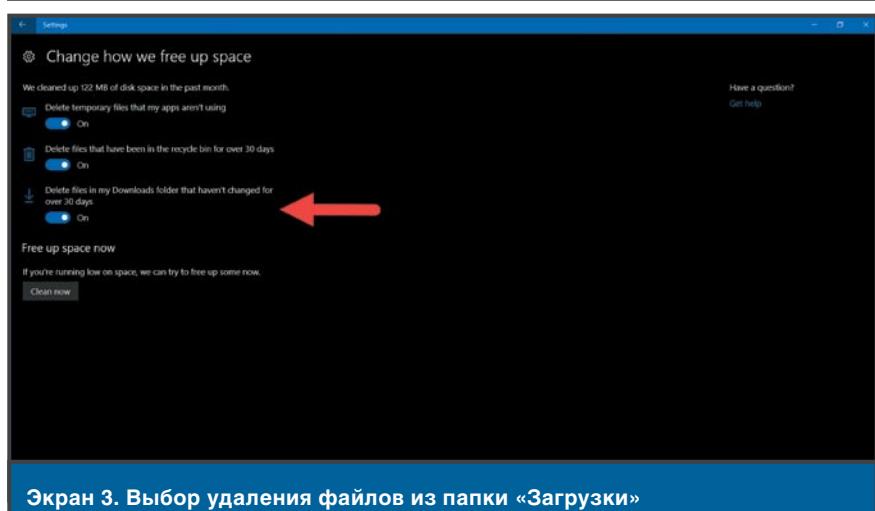
**Шаг 3**

Переведите переключатель «Удалять из папки ‘Загрузки’ файлы, которые не изменились свыше 30 дней» в положение On («Вкл.»), как показано на экране 3.

Теперь Storage Sense будет автоматически очищать папку «Загрузки» в рамках обычного обслуживания системы. Можно также задействовать еще два варианта освобождения других областей операционной системы от ненужных файлов. Учтите только, что эти два варианта доступны в обновлении Windows 10 Creators Update, выпущенном в апреле этого года. Включив эту установку, потребуется выполнить некоторые действия. Например, если в вашей папке «Загрузки» есть файлы, которые необходимо сохранить, перенесите их в постоянное место хранения. Это позволит вернуться к использованию папки «Загрузки» в качестве временного места хранения, как и было задумано изначально. 



Экран 2. Изменяя способ освобождения пространства



Экран 3. Выбор удаления файлов из папки «Загрузки»

Ричард Хэй (winobs@outlook.com) имеет звание Microsoft MVP в категории Windows Operating System с 2010 года

# ФУНКЦИЯ

## OneDrive Files On-Demand

**Ричард Хэй**

**Н**а конференции Build, которая состоялась в этом году в Сиэтле, было объявлено, что в состав пакета обновлений Windows 10 Fall Creators Update войдет новый компонент, пришедший на смену функции OneDrive Placeholders. В результате пользователи получат средство, восстановления которого они активно требовали с тех пор, как функция Placeholders была исключена из продукта еще до первоначального выпуска ОС Windows 10 в начале 2015 года.

Обращение OneDrive UserVoice с требованием восстановить возможность использования меток-заместителей для хранилища OneDrive появилось в Интернете в июле 2015 года, и с тех пор под ним под-

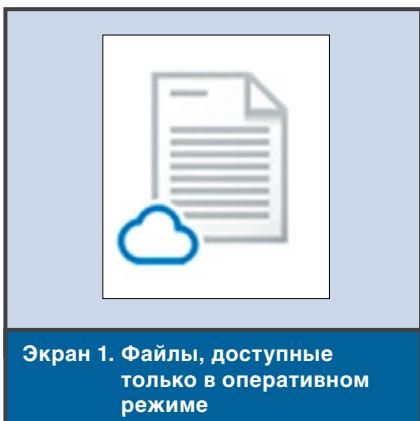
писалось почти 26 000 человек, а более 1000 пользователей оставили свои комментарии.

Когда вице-президент Microsoft и руководитель группы разработки Windows Джо Бельфиоре сделал заявление о новом компоненте в своем докладе на конференции Build 2017, оно было встречено с энтузиазмом, которого удостаивались лишь немногие ораторы на протяжении двух дней, когда вниманию участников предлагались основные доклады.

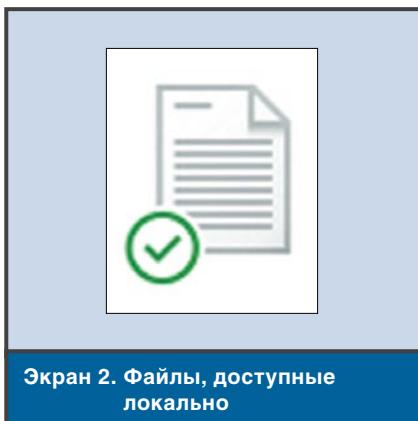
Для тех, кто, возможно, никогда не слышал о метках-заместителях или не понимает, в чем состоит их ценность, я приведу краткое описание этой функции, опубликованное мною ранее:

«На вашем локальном жестком диске заместители служили маркерами для

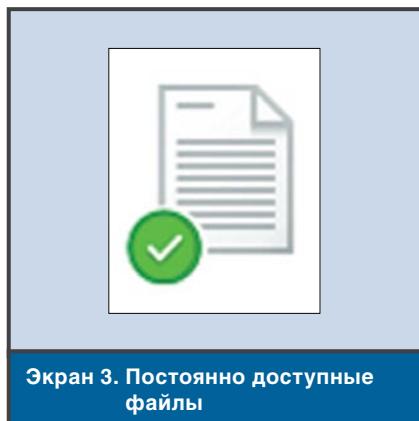




Экран 1. Файлы, доступные только в оперативном режиме



Экран 2. Файлы, доступные локально



Экран 3. Постоянно доступные файлы

файлов, которые вы помещали в хранилище OneDrive. Они позволяли видеть все файлы, перенесенные в «облако», причем синхронизация с локальным диском для этого не требовалась. Эти маркеры отличались весьма компактными размерами и давали возможность с легкостью синхронизировать и открывать файл на локальной системе, а затем, после окончания работы с ним, синхронизировать обновленный вариант с файлом, записанным в хранилище OneDrive в «облаке». Эта функция позволяла с большой точностью указывать, какая часть имеющихся в облаке данных физически хранится локально, но в то же время давала возможность в любое время видеть структуру всего каталога OneDrive. Изменить синхронизационный статус любого файла или каталога можно было с помощью контекстного меню, вызываемого щелчком правой кнопки мыши на имени соответствующего файла или каталога».

Функция OneDrive Files On-Demand предоставит пользователям те же возможности, что и метки-заместители, но теперь эти возможности будут открыты и для посетителей сайтов групп OneDrive for Business и SharePoint.

Вскоре после конференции Build 2017 я вновь написал об этой функции. Это была статья, написанная на основе снятого в Microsoft деморолика, авторы которого обращались в основном к пользователям из сферы бизнеса. Если судить по сегодняшнему выпуску, получается, что моя мысль была верной. В Microsoft решили создать единый

клиент синхронизации для всех пользователей OneDrive и открывать доступ к определенным функциям для пользователей на базе учетных записей, под которыми они регистрируются на своих системах. Чтобы начать работу с функцией OneDrive Files On-Demand, необходимо установить на своей системе версию Windows Insider со сборкой выше 16215 и загрузить новый клиент синхронизации OneDrive, который Microsoft предоставляет по адресу: <https://go.microsoft.com/fwlink/?linkid=851311>. Учтите, что Microsoft будет обновлять клиенты синхронизации OneDrive в автоматическом режиме. Впрочем, что-то подсказывает мне, что те, кто работает с программой Windows Insider и получает обновления в режиме Fast Ring, не станут дожидаться получения новой функции в ходе обновления в автоматическом режиме.

Когда метки-заполнители были удалены из OneDrive, как утверждали представители Microsoft, проблема была связана с тем, что некоторые пользователи не могли понять, размещены ли те или иные файлы локально на их устройствах. Можно себе представить, какие чувства испытывает пользователь, когда видит, что копии интересующего его файла на локальном диске нет, а связь с Интернетом никак не устанавливается.

Проблема сохраняет актуальность и с появлением функции OneDrive Files On-Demand. Поэтому для нас чрезвычайно важно организовать управление «облачным» хранилищем так, чтобы нужные файлы были представлены на нашем устройстве в тот момент, когда они нам понадобятся.

OneDrive Files On-Demand даст пользователю представление о структуре всего «облачного» хранилища службы. Вы сможете увидеть каждый каталог или файл. И каждый из них будет отмечен как находящийся в одном из трех перечисленных ниже состояний, которые определяют его статус в хранилище.

- Файлы в «облаке».** Файлы, доступные только в оперативном режиме, не занимают место на вашем компьютере. В окне File Explorer пользователь видит значок с облачком рядом с именем такого файла (экран 1), но этот файл загружается на локальное устройство лишь после того как пользователь решит его открыть. При отсутствии соединения с Интернетом открыть файл, доступный только в оперативном режиме, невозможно.

- Файлы, доступные локально.** Когда вы открываете файл, доступ к которому возможен только через Интернет, он загружается на ваше устройство и становится файлом с локальным доступом (экран 2). С такими файлами можно работать в любое время, даже если Интернет недоступен. Если вы считаете, что дискового пространства на вашем устройстве недостаточно, можете вернуть файлу прежний статус. Для этого нужно щелкнуть на файле правой кнопкой мыши и в открывшемся меню выбрать пункт Free up space («Высвободить пространство»).

- Постоянно доступные файлы.** Только те файлы, которые вы помечаете как Always keep on this device («Постоянно сохранять на этом устройстве»), отобража-

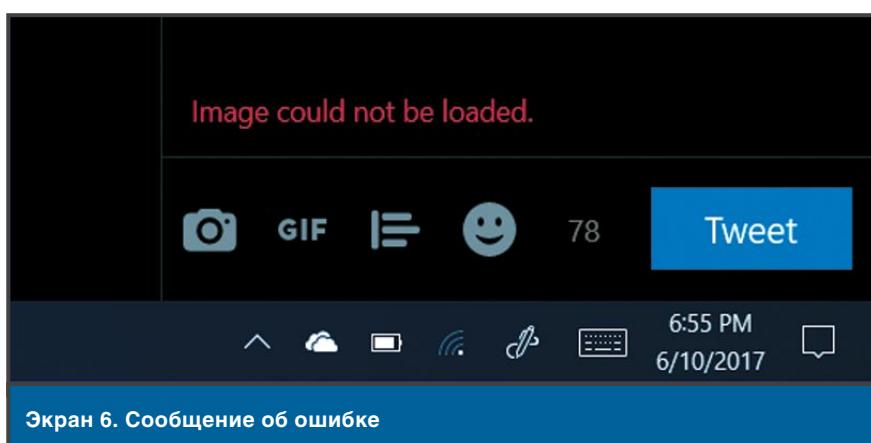
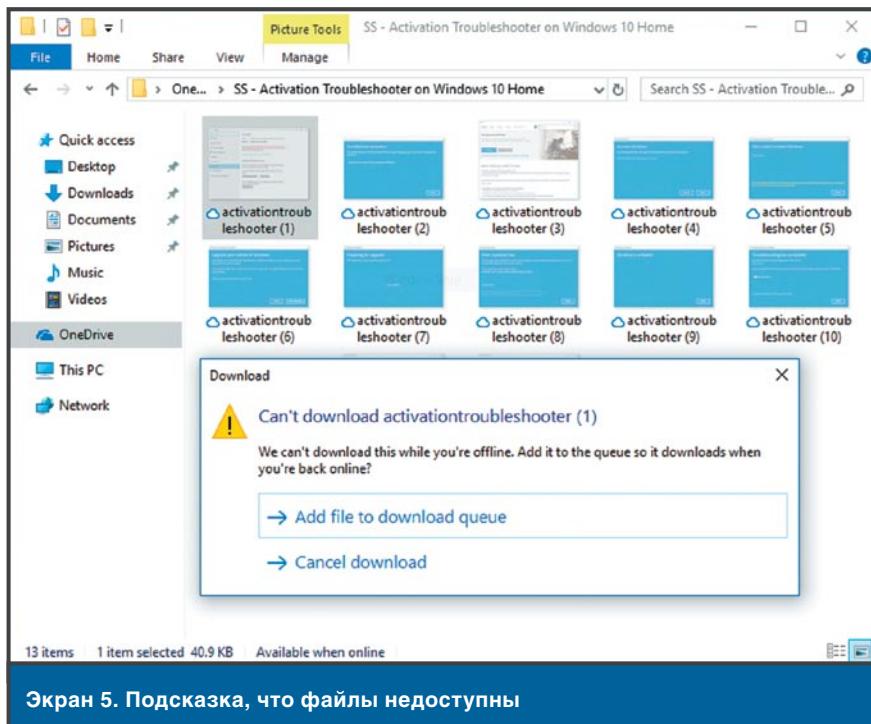


ются значками с зеленым кружком и белой «галочкой» внутри (экран 3). Такие файлы загружаются на ваше устройство и занимают дисковое пространство, но они всегда у вас под рукой — даже когда вы работаете в автономном режиме.

Существует еще один декоративный значок, который иногда применяется при использовании «облачных» файлов по запросу OneDrive Files On-Demand; это временный элемент, поскольку он обозначает файл, находящийся в процессе синхронизации на локальном накопителе из «облака» (экран 4).

Если пользователи, просматривая «облачное» хранилище OneDrive, потеряют связь с Интернетом, они получат еще одну визуальную подсказку, которая известит их о том, что файлы недоступны. Как показано на экране 5, эти миниатюрные картинки отображаются в приглушенных тонах, поскольку представленные ими файлы нельзя открыть в автономном режиме.

Последнее замечание: когда речь заходит о поиске в файлах, расположенных в «облачном» хранилище OneDrive, и когда вы при этом используете функцию Files On-Demand, внутри размещенных в «облаке» файлов нельзя выполнять операцию поиска таким же образом, каким она выполняется внутри файлов, хранящихся на диске вашего устройства. Однако при выполнении любой операции поиска система будет реагировать



на наличие искомых элементов в именах файлов, ибо они известны локальному устройству, так как обладают статусом заместителей. Другая особенность функции OneDrive Files On-Demand состоит в том, что файлы, размещенные в «облаке» и не синхронизированные с вашим локальным устройством, не занимают пространство на жестком диске. Если вы просто щелкните на файле, доступном только в оперативном режиме, этот файл будет загружен на ваше устройство и помечен как файл, доступный на локальной системе, то есть вы сможете обрабатывать его с помощью своих приложений и программ.

Я протестировал функцию OneDrive Files On-Demand, чтобы выяснить, как работают с «облачными» фай-

лами по запросу некоторые приложения и программы, установленные на моей системе. Единственная проблема, с которой мне пришлось пока столкнуться, связана с официальным приложением Twitter. При любой попытке захватить «картинку», хранившуюся только в «облаке», я получал сообщение об ошибке (экран 6).

Я убежден, что проблема может быть решена с помощью небольшого модуля коррекции, в котором будет реализована логика, необходимая для загрузки файла, и тогда приложение будет работать с ним как с локальным файлом. ♦

Ричард Хэй ([winobs@outlook.com](mailto:winobs@outlook.com)) имеет звание Microsoft MVP в категории Windows Operating System с 2010 года

# Microsoft Forms: практические занятия

Как вы, возможно, заметили, Microsoft Forms стали доступны широкой аудитории пользователей (экран 1). Forms можно связать с пользователем, так как данный компонент охватывает лицензия Office 365 плана E3, как показано на экране 2.

После того как вы определили лицензию, можно назначить некоторые основные параметры, разрешив или запретив внешний общий

доступ к формам. Это можно сделать, выбрав Settings («Параметры»), затем Services & add-ins («Службы и надстройки») и Microsoft Forms из Центра администрирования (экран 3).

Затем вы просто указываете, следует ли разрешить совместную работу и ответы от внешних пользователей. Имейте в виду, что этот режим включен по умолчанию, поэтому, если вы в своей организации не используете внешний общий

New feature: Microsoft Forms Preview  
MC106358  
Published On : June 16, 2017  
Expires On : October 15, 2017

Microsoft Forms is a simple and lightweight app for creating polls, surveys, and quizzes that lets your users easily collect customer feedback, measure satisfaction, learn what employees think, organize team events, and more. It works on any device, and you can even embed it directly into your SharePoint site or Yammer feed.

**Экран 1. Microsoft Forms стали доступны**

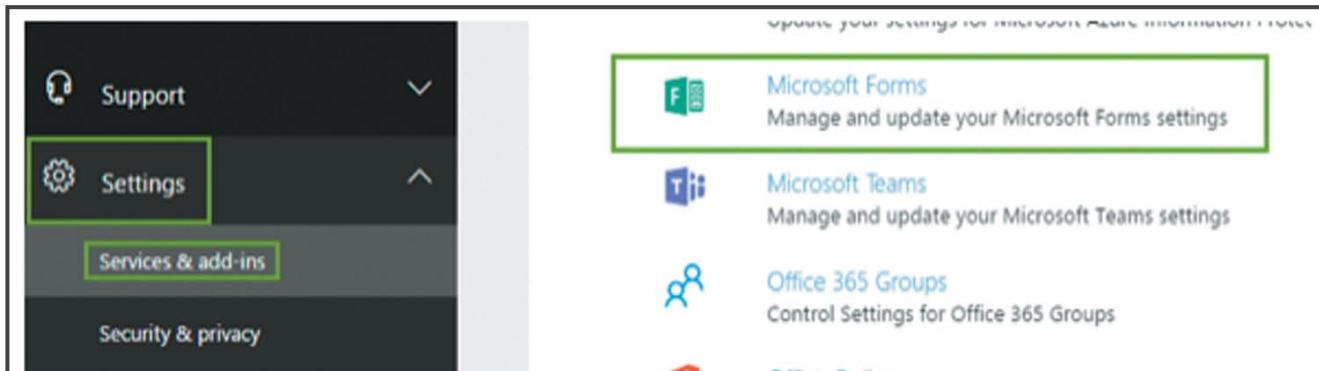
to buy an additional license for you.

- Office 365 Enterprise E3  On
- Licenses are available
- Microsoft Forms (Plan E3)  On
- Microsoft Stream for O365 E3 SKU  On
- Microsoft StaffHub  On
- Flow for Office 365  On

**Экран 2. Microsoft Forms с лицензией Office 365 плана E3**

Изучаем  
возможности  
форм

**Лиам Клири**



Экран 3. Общий доступ к формам

доступ, нужно отключить его, как показано на экране 4.

После того как настройки завершены, можно щелкнуть новую плитку Forms на панели навигации, доступной из меню (экран 5).

Можно просто перейти к <https://forms.office.com> или <https://forms.microsoft.com>, чтобы получить доступ к веб-консоли (экран 6).

Для создания формы достаточно нажать кнопку New Form («Новая форма») и добавить нужный вопрос. Добавляя вопрос, необходимо выбрать один из четырех типов управления (экран 7).

Выбирая пункт Choice («Выбор»), вы можете ввести вопрос и различные ответы (экран 8).

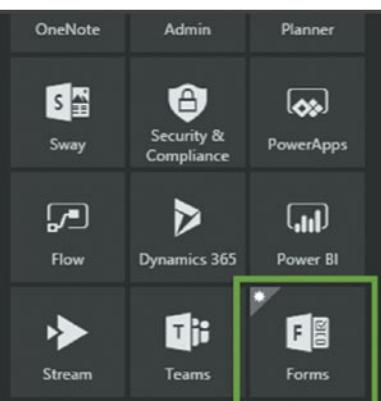
Для каждого вопроса предусмотрено несколько вариантов ответа, что позволяет задать режим Allow multiple answers («Разрешить несколько ответов») и указать, что он необходим (required), как показано на экране 9. Каждому варианту можно сопоставить картинку, просто нажав значок для мультимедиа рядом

## External collaboration

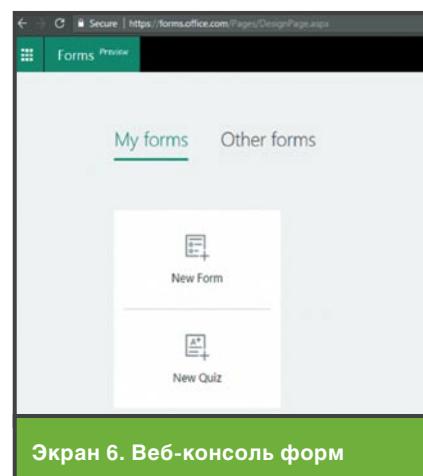
Let people in your organization collaborate on forms and responses with external people



Экран 4. Включение режима общего доступа к форме



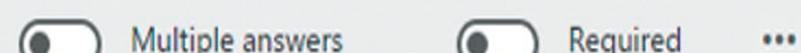
Экран 5. Выбор форм в меню



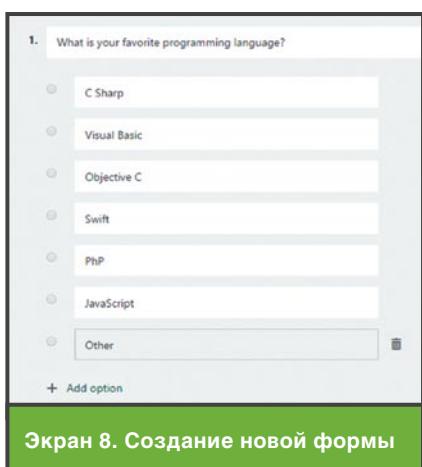
Экран 6. Веб-консоль форм



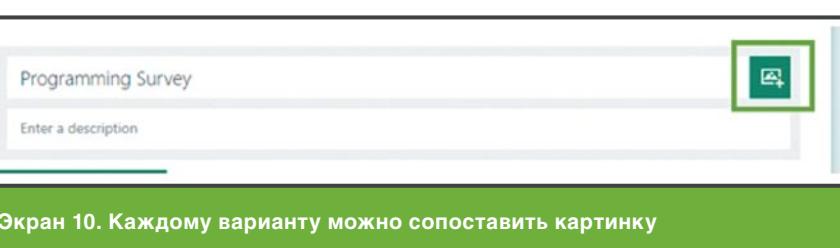
Экран 7. Типы новой формы



Экран 9. Режимы ответов



Экран 8. Создание новой формы



Экран 10. Каждому варианту можно сопоставить картинку

Programming Survey

Enter a description

+ Add question

**Экран 11. Добавление картинки в форму**

с местоположением на форме (экран 10).

Затем можно выбрать в локальной системе и отправить мультимедиа или взять какое-нибудь изображение из Bing (экран 11)

Создав нужную форму, нажмите кнопку Preview («Просмотр»), чтобы увидеть, как она будет отображаться на экране (экран 12).

Обратите внимание, что просмотр возможен как на компьютере, так и на мобильном устройстве. Если вы

Send and collect responses

Only people in my organization can respond ▾

Anyone with the link can respond

✓ Only people in my organization can respond

**Экран 13. Указание ответственного за форму**

удовлетворены результатом, то можно применить тему, выбрав пункт Theme («Тема») и заранее подготовленное или новое изображение, которое будет отображаться как фоновое.

Теперь у вас есть нужная форма. Нажмите кнопку Share («Общий доступ») и укажите, как следует организовать общий доступ к форме. Прежде всего необходимо определить, кто может отвечать на форму (экран 13).

Затем на экране 14 можно выбрать один из четырех вариантов общего доступа: Link («Ссылка»), QR Code («QR-код»), Embed («Внедрить») и Email («Электронная почта»).

Конечно, все это бесполезно, если вам не удастся увидеть ответы. Щелкните на сайте форм, выберите форму, а затем вкладку Responses («Ответы»), на которой будут показаны ответы (экран 15). Как видите, создавать формы очень просто. У них богатые перспективы и,

Hi Liam, when you submit this form, the owner will be able to see your name and email address.

\* Required

1. What is your favorite programming language? \*

C#  
 Visual Basic  
 Objective C  
 Swift  
 PHP  
 JavaScript  
 Other

Submit

**Экран 12. Предварительный просмотр формы**

надеюсь, в скором времени их функциональность будет расширена.

Лiam Клири — архитектор решений, имеет сертификат Microsoft MVP

Responses 5 Average time to complete 00:04 Active Status

View results

1. What is your favorite programming language?

Language	Count
C#	2
Objective C	1
JavaScript	1
Swift	1
PHP	1

**Экран 15. Готовая форма с ответами**

**Link**

Send and collect responses

Only people in my organization can respond ▾

<https://forms.office.com/Pages/ResponsePage.aspx>

**QR Code**

Send and collect responses

Only people in my organization can respond ▾

Recipients can scan the code on a phone or tablet to access the form.

**HTML Embed**

Send and collect responses

Only people in my organization can respond ▾

Copy this code and paste it in a webpage or Sway.

```
<iframe width="640px" height="480px" src="https://forms.office.com/Pages/ResponsePage.aspx?id=bzJKEF1k601m1NU7_69IP2aeP04PnEnLLOV13acI9UQjU2RvpBQklyUw0M08wRfV1WVVC0xK54u">
```

**Email Link**

Here's the link to the form "Programming Survey":

[https://forms.office.com/Pages/ResponsePage.aspx?id=bzJKEF1k601m1NU7\\_69IP2aeP04PnEnLLOV13acI9UQjU2RvpBQklyUw0M08wRfV1WVVC0xK54u](https://forms.office.com/Pages/ResponsePage.aspx?id=bzJKEF1k601m1NU7_69IP2aeP04PnEnLLOV13acI9UQjU2RvpBQklyUw0M08wRfV1WVVC0xK54u)

**Экран 14. Четыре варианта общего доступа**

# Учимся бороться с шифровальщиками



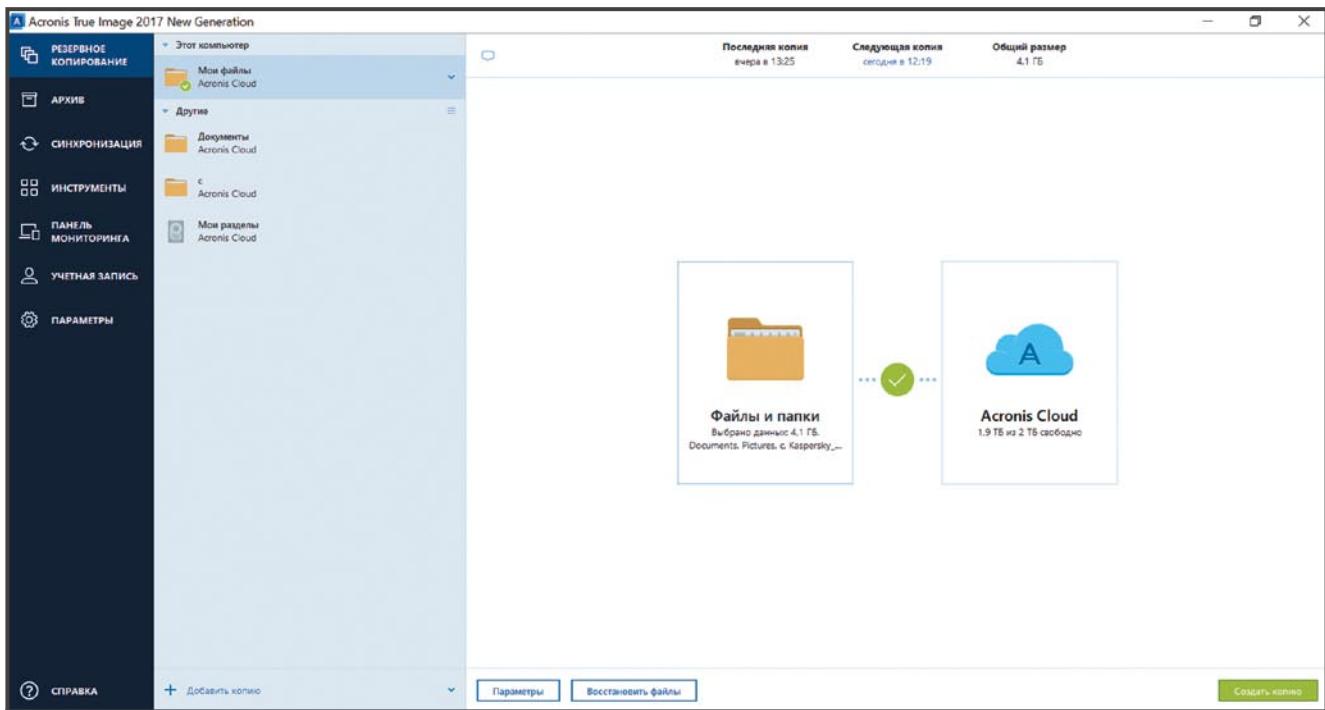
**Владимир  
Безмалый**

**В** последнее время произошел резкий рост количества атак вредоносов-шифровальщиков. Причем надо отметить, что этот рост наблюдается практически для всех операционных систем. В силу массовости устройств на базе операционных систем Windows и Android мы будем говорить исключительно о них. Для начала немного статистики (см. рисунок). В 2016 году шифровальщиками было атаковано 1 445 434 пользователей. Каждые 100 дней программы-вымогатели делают своих создателей богаче на 30 млн долл. (согласно отчету Dell SecureWorks). Количество новых модификаций таких программ выросло в 11 раз. Число атак на компании увеличилось в три раза: если раньше атаки проводились в сред-

нем каждые две минуты, то теперь уже каждые 40 секунд. Интенсивность атак на индивидуальных пользователей удвоилась: атаки проводились в среднем раз в 20 секунд в начале периода и раз в 10 секунд — в конце. Однако если раньше атакам в основном подвергались компьютеры с операционными системами Microsoft, то сегодня ситуация изменилась. В сети опубликован исходный код одного из самых популярных семейств программ для вымогательства для устройств с операционной системой Android — Slocker, число новых вариантов которого за последние полгода возросло в шесть раз (код опубликован на портале GitHub). Код троянца разместил некто под псевдонимом fs0 c1 ety, призвавший пользователей ресурса внести



Рисунок. Количество пользователей, атакованных шифровальщиками за последние годы



Экран 1. Главное окно Acronis True Image 2017

свой вклад в его разработку и предоставить отчеты о найденных в нем уязвимых местах.

SLocker (или Simple Locker) представляет собой программу для вымогательства, шифрующую файлы на мобильном устройстве и блокирующую его экран. Для связи с C&C-сервером используется сеть Тор. Что можно предпринять для борьбы с ним и профилактики? На самом деле рекомендации будут стандартными для обеих рассматриваемых операционных систем.

1. Используйте последнюю версию операционной системы.
2. Регулярно устанавливайте обновления.
3. Используйте и регулярно обновляйте антивирусное программное обеспечение, причем обращайте внимание на то, какие именно программы вы используете. На мой взгляд, они должны лидировать в независимых тестах и показывать хорошие результаты при обнаружении вирусов.
4. Не работайте с правами учетной записи локального администратора.
5. Не работайте на устройстве с измененной прошивкой, то есть рутованном.

## 6. И последняя, самая главная рекомендация: ДЕЛАЙТЕ РЕЗЕРВНЫЕ КОПИИ!

### Советы для Windows

Многообразие операционных систем Windows, представленных сегодня на рынке, включает версии XP, 7, 8.1 и 10. Тем не менее количество домашних компьютеров под управлением Windows XP остается достаточно высоким. Что можно посоветовать пользователям этой операционной системы? Только посочувствовать. Им просто следует понять, что пора двигаться вперед. Нельзя требовать безопасности от операционной системы, не сопровождаемой производителем. Их безопасность — это только их проблема.

Что же касается версий 7, 8.1 и 10, то здесь необходимо вспомнить, что в данных операционных системах вы сможете задействовать как встроенные средства резервного копирования, так и программы резервного копирования от независимых производителей. Единственное, о чем необходимо помнить, — внешний носитель для резервного копирования должен подключаться только на время создания резервной копии. Это

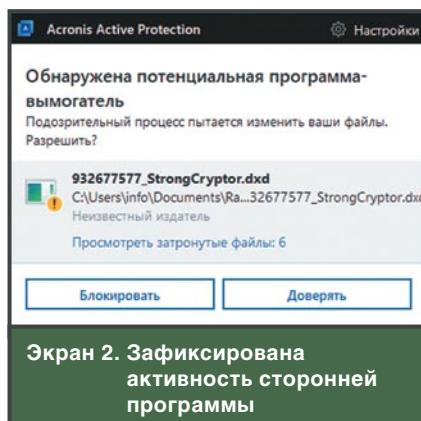
позволит вам избежать ситуации, когда одновременно будет зашифрован как основной, так и резервный носитель.

Если же вы захотите использовать в качестве носителя для резервной копии «облачное» хранилище, то можете выбрать как хранилище от Microsoft, так и хранилище от Google или воспользоваться программным обеспечением компании Acronis. Именно на нем я и остановлюсь, потому что сам использую продукт Acronis True Image 2017 (экран 1). Согласитесь, очень удобно, когда проактивный, активный и реактивный виды защиты сосредоточены в одном продукте. Таким образом, данное решение не только позволит вам создать резервные копии устройств в «облаке», но и предоставит уникальные технологии безопасности, включая активную защиту от шифровальщиков.

### Acronis Active Protection

До появления вирусов-шифровальщиков, будем откровенны, не многие пользователи задумывались о резервном копировании. Технология Acronis Active Protection использует эвристические методы обнаружения

## Вводный курс



для мониторинга подозрительной активности с файловой системой в целом, а не только с файлами, для которых настроены задания резервного копирования. Защита работает постоянно, о чем свидетельствует значок в системном лотке на экране, который предоставляет доступ к настройкам.

Таким образом, резидентная утилита постоянно отслеживает сторонние приложения, которые пытаются зашифровать данные. Пользователь может сам как разрешить активность отдельных программ, так и отключить защиту определенных папок и файлов (экран 2). Более того, вы можете настроить автоматическое восстановление файлов, которые могли

быть затронуты заблокированной операцией шифрования.

### Функция Notary

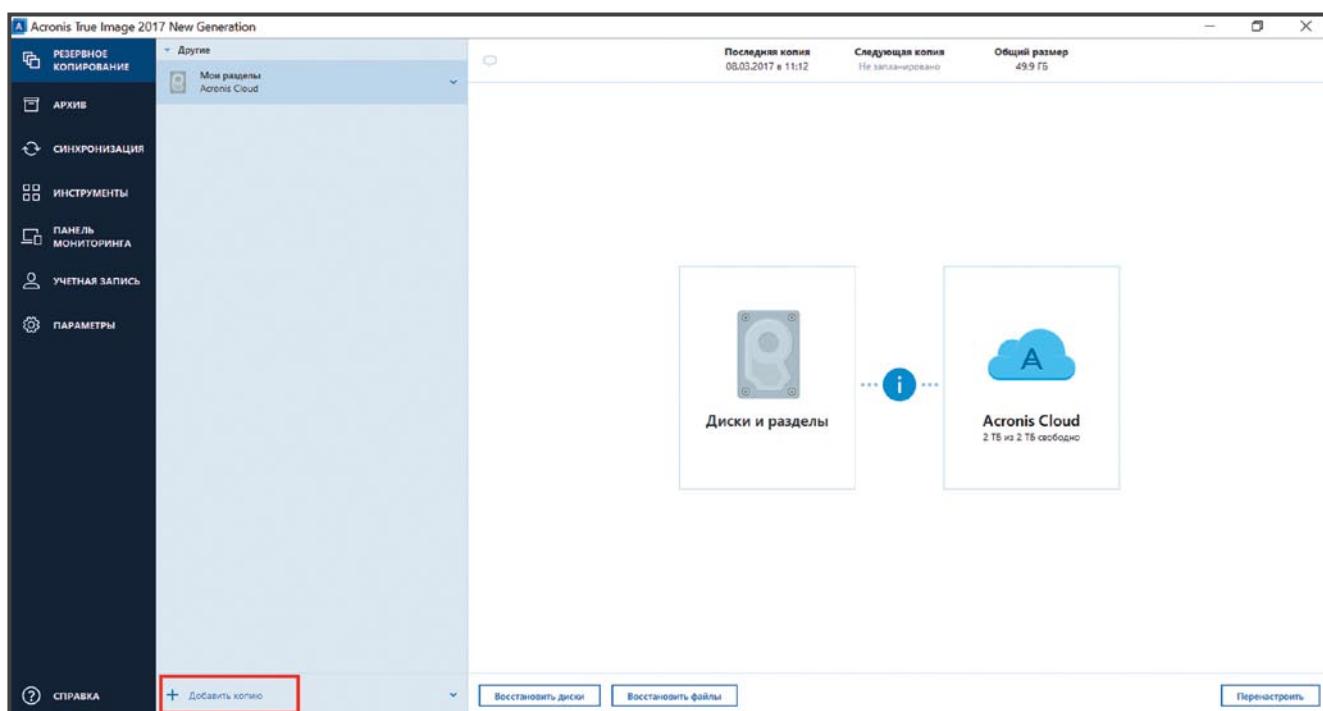
В качестве дополнительного подтверждения того, что файлы остались в таком же состоянии, как и при резервном копировании, Acronis предоставляет функцию Notary. Данная функция использует известную технологию цепочки блоков под названием «блокчейн», которая применяется в криптовалютах для обеспечения гарантии. Чтобы создать «заверенную» копию, нужно подготовить новое задание на резервное копирование и при выборе источника резервного копирования выбрать вариант «Файлы для заверения». Затем следует выбрать файлы и папки для обработки и в качестве места назначения указать локальное место или Acronis Cloud. Затем появится анимированное уведомление «Заверение».

После завершения процедуры создания резервной копии вы можете проверить файл и даже посмотреть его официальный сертификат, в котором в качестве правообладателя указан Acronis Notary. По утверждению специалистов Acronis, это неопровергимое доказательство, что файл не был изменен.

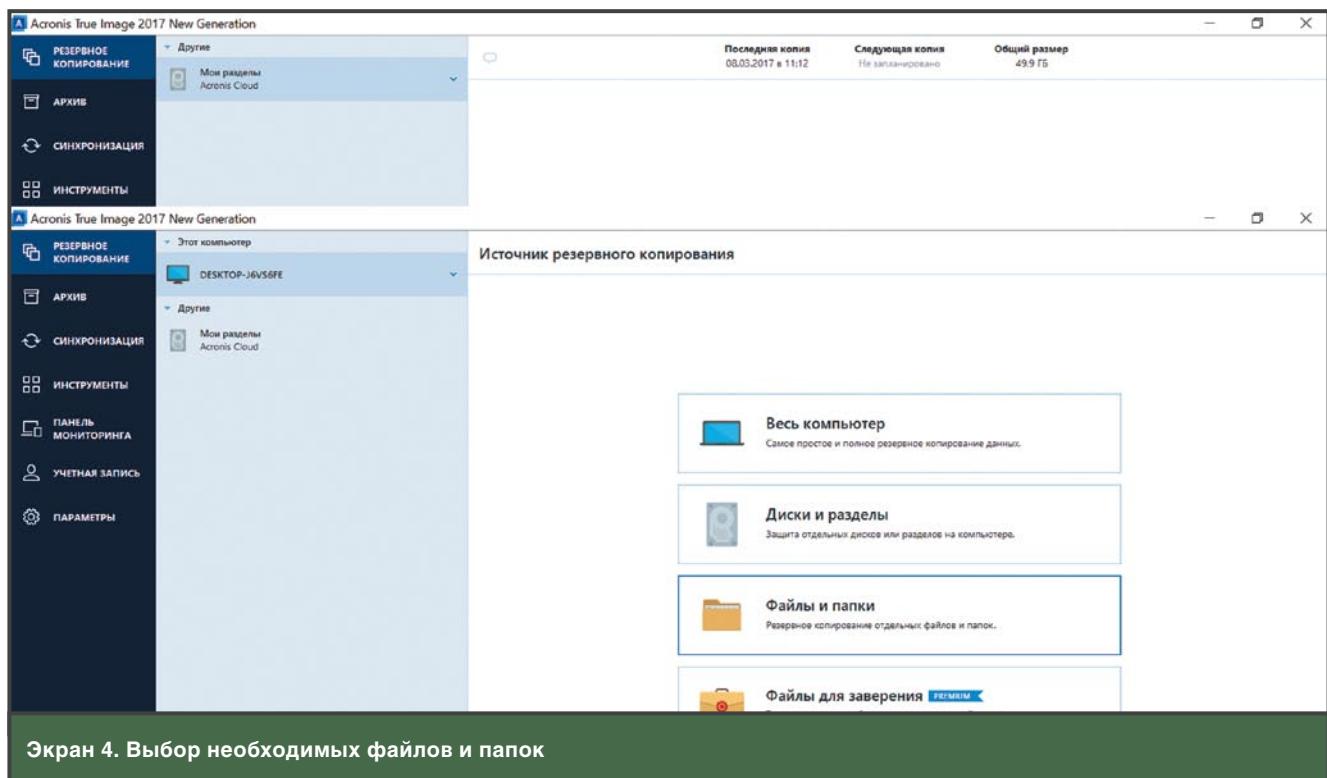
Итак, для создания резервной копии данных на компьютере вам необходимо установить и запустить программное обеспечение. В дальнейшем — определить, что именно вы хотите копировать. Я рекомендую делать на всякий случай три копии ваших наиболее важных файлов и компьютера в целом. Три копии на двух разных носителях, причем одна копия должна храниться вне вашего дома или офиса.

При использовании первой копии в случае неприятностей вам нужно будет заново установить все необходимое программное обеспечение, включая операционную систему, а затем восстановить свои данные. Очевидно, что в этом случае время создания резервной копии, как и время восстановления данных из нее, будет минимальным. Для такой копии я бы предложил «облачное» хранилище. Ведь неизвестно, насколько надежно ваше соединение с Интернетом. Выбираем в окне программы вариант «Добавить копию», как показано на экране 3.

Выбираем нужные файлы и папки (экран 4) и запускаем процесс



Экран 3. Создание копии



Экран 4. Выбор необходимых файлов и папок

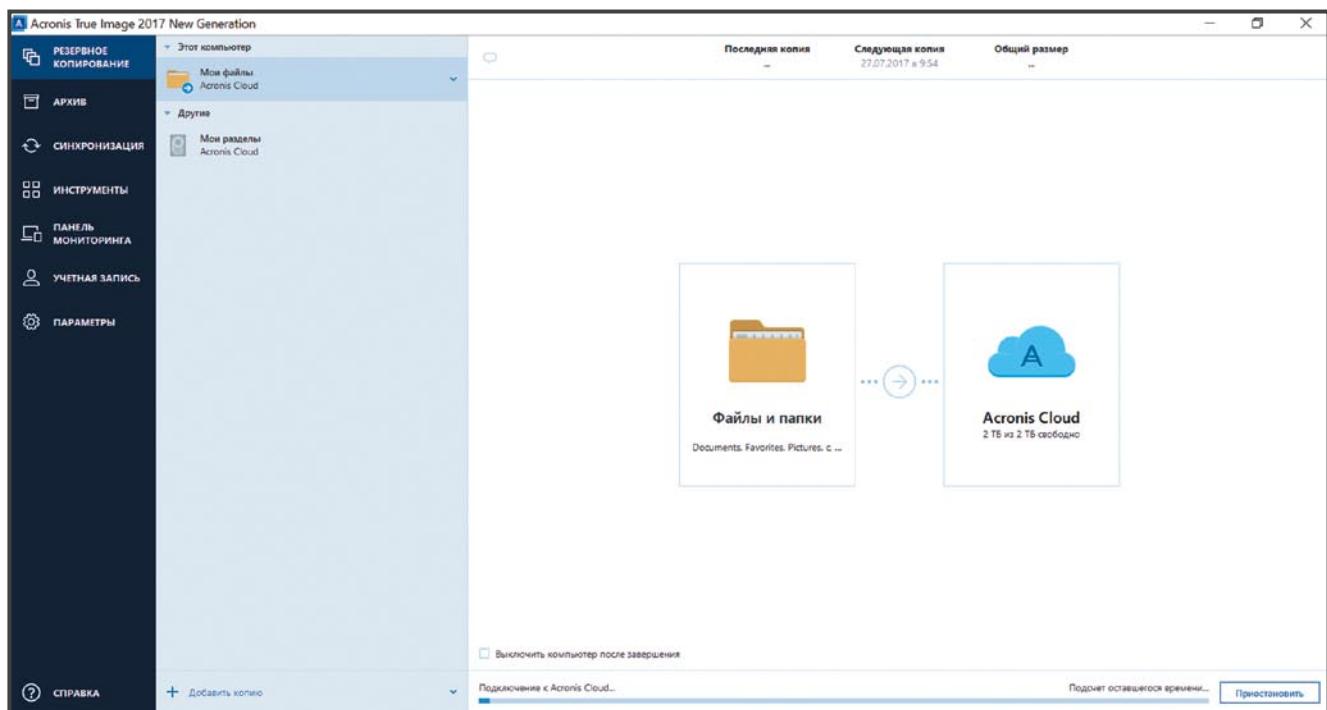
копирования в «облачное» хранилище (экран 5).

Создание резервной копии всего компьютера тоже процесс несложный. Для этого вам нужно также выбрать вариант «Добавить копию», но затем указать «Изменить место хранения» и выбрать ваше подключенное

внешнее хранилище. Почему я не рекомендую выбирать «облачное» хранилище? Все просто. Дело в объеме информации, которую требуется перенести в это «облачное» хранилище. Безусловно, если у вас высокоскоростной Интернет, то можно воспользоваться и «облачком».

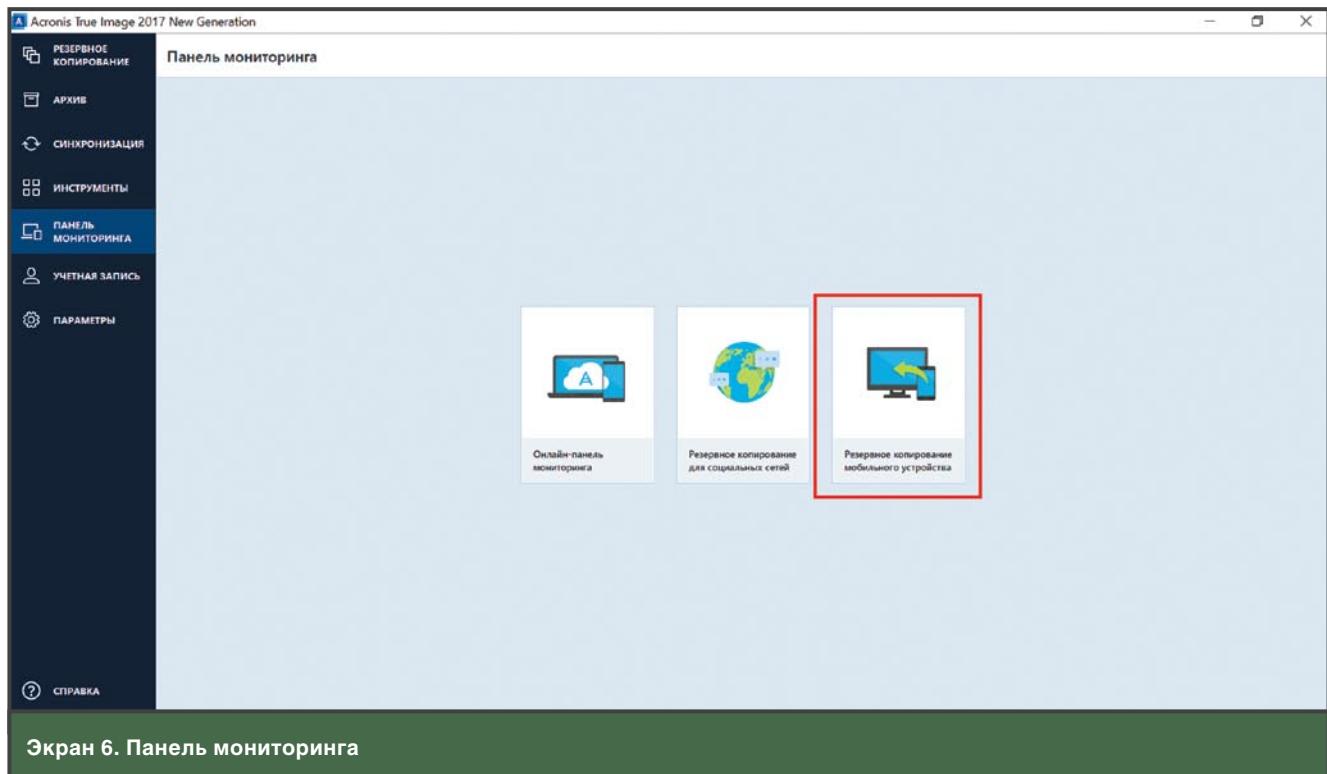
## Создание резервной копии для устройств Android

Однако гораздо чаще сегодня встречается ситуация, когда пользователю нужно создать резервную копию своего смартфона (планшета). Особенно на базе операционной системы Android. Увы, несмотря на то что на данном рынке средства



Экран 5. Создание «облачной» резервной копии

## Вводный курс



Экран 6. Панель мониторинга

резервного копирования довольно многочисленны, чаще всего это либо специализированные решения от конкретного производителя, либо программы, требующие прав администратора устройства, что само по себе уже снижает безопасность вашего смартфона или планшета. Что можно предложить в таких условиях? Тот же продукт Acronis.

Наверное, многие из вас сталкивались с тем, что по какой-то причине смартфон выходил из строя. Рано или поздно это случается. Вы шли в сервисный центр и слышали там сакраментальную фразу: «Резервная копия есть? Мы не гарантируем сохранность данных!». Поэтому еще раз повторю: резервные копии жизненно необходимы, нравится нам это или нет. Для создания резервной копии устройства под управлением Android необходимо:

1. Выбрать раздел «Панель мониторинга».
  2. Выбрать вариант «Резервное копирование мобильного устройства» (экран 6).
  3. Загрузить приложение Acronis Mobile и следовать полученным инструкциям.
- В случае заражения вашего смартфона необходимо сбросить его

в заводские настройки (этому посвящено достаточно много статей в Интернете). Учтите, что вам потребуется способ сброса при помощи кнопок. У каждой модели устройства есть стандартное сочетание кнопок, которое переключает его на меню Recovery. Для этого выключите свой телефон (планшет). Дождитесь полного отключения. Учтите, что комбинация для вашей модели может отличаться от общепринятых. Уточните на сайте производителя.

Как правило, это:

- кнопка «уменьшить громкость» + «включение» (она же Power) — самая распространенная комбинация;
- на некоторых телефонах компании LG нужно нажать названные выше клавиши, дождаться появления на экране логотипа, отпустить кнопку включения и затем снова ее нажать;
- комбинация кнопок «громкость вверх» + «громкость вниз» + «включение»;
- комбинация кнопок Power + Home.

Используйте одну из комбинаций, пока не войдете в режим Recovery, чтобы затем сбросить устройство в заводские настройки. Перемещение по пунктам меню

происходит кнопками увеличения и уменьшения громкости. Если версия Recovery сенсорная, то можно выполнить перезагрузку и стандартным образом (прикосновениями к экрану). Для подтверждения выбора нужно нажимать кнопку Power или «Контекстное меню».

Далее:

1. Выберите пункт Clear eMMC или wipe data/factory reset, иногда он еще называется Clear Flash.
2. Подтвердите действие yes — delete all user data, чтобы сбросить данные.

3. После завершения процесса выберите Reboot System.

Затем вы можете восстановить свои данные и установить те приложения, которые были у вас до сброса. Как видите, все достаточно просто. Таким совсем не сложным образом вы сможете спасти свои данные в случае непредвиденной аварии (совсем не обязательно это будет атака вредоносного-шифровальщика). Надеюсь, рекомендации, приведенные в этой статье, помогут вам.



---

Владимир Безмалый (vladb@windowslive.com) — специалист по обеспечению безопасности, Kaspersky Lab Certified Consultant, Kaspersky Lab Certified Trainer, имеет звания MVP Consumer Security, Microsoft Security Trusted Advisor

# Что знает о вас «облако»

**Л**юбое мобильное устройство содержит конфиденциальные данные пользователя, и очень важно, чтобы сведения обо всем, что вы делаете со своим смартфоном Android, iPhone или устройством с системой Windows Phone, хранились как можно надежнее. Причем не следует забывать, что извлечь информацию можно и из «облачной» копии мобильного устройства, так что ее защита не менее важна. Ведь на самом деле, если телефон попадет в чужие руки, максимум, что удастся из него извлечь, это файлы с фото, видео, музыкой, список контактов и журнал звонков. Да и то если ваш телефон не заблокирован устойчивым PIN-кодом. Однако в случае с «облачной» копией ситуация иная, и человеку, обладающему соответствующим инструментарием, может попасть в руки гораздо больше информации. В этой статье мы рассмотрим, какие данные можно извлечь из «облака», в частности создаваемые при синхронизации мобильного устройства с учетной записью Microsoft ID.

Для работы нам потребуется программное обеспечение Elcomsoft Phone Viewer и Elcomsoft Phone Breaker. Сначала нужно извлечь данные с помощью Elcomsoft Phone Breaker (экран 1).

Если учетная запись использует двухэтапную аутентификацию, от вас потребуется либо ввести доверенный адрес электронной почты (тот, который вы используете для восстановления пароля Windows ID), либо последние четыре цифры номера телефона, на который придет короткое сообщение (экран 2). Учтите, что, даже если для учетной записи не включена двухфакторная аутентификация, до журнала вызовов все равно можно добраться, получив только код по SMS или электронной почте. Эдакая принудительная двухфакторная аутентификация для защиты данных, которые компания Microsoft считает наиболее критичными. После этого вы должны выбрать, какие данные хотите загрузить, как показано на экране 3.

После загрузки вы получите архив с именем вида Backup\_суXXXXXXX@outlook.com\_20170621T162047.zip, где суXXXXXXX@outlook.com — имя соответствующей учетной записи.

Для просмотра полученного архива нам понадобится программное обеспечение Elcomsoft Phone Viewer.

Из окна этого приложения, которое вы видите на экране 4, можно выбрать необходимые данные:

- звонки;
- контакты;

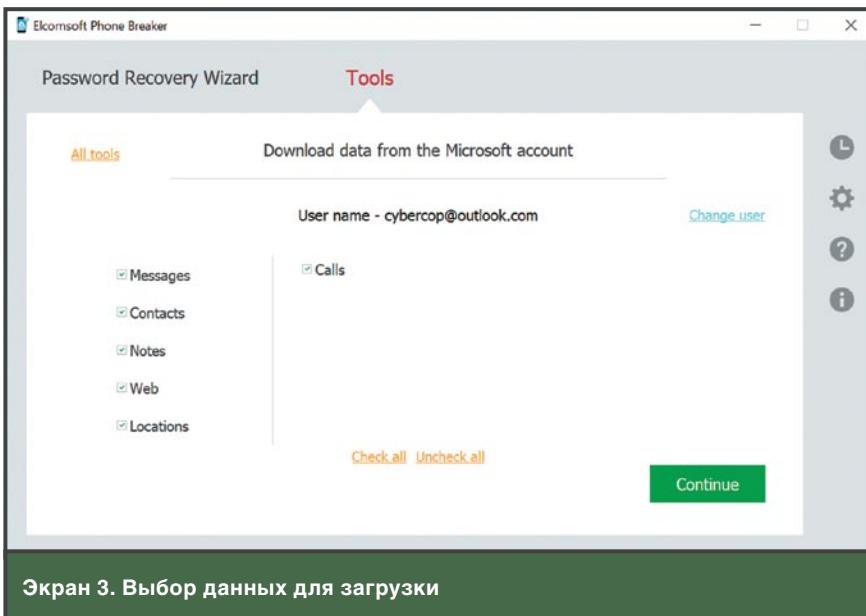


**Владимир  
Безмалый**

Экран 1. Ввод учетных данных

Экран 2. Введите код доступа

# Вводный курс



- расположение;
- сообщения;
- заметки;
- Web.

Давайте рассмотрим все элементы подробнее.

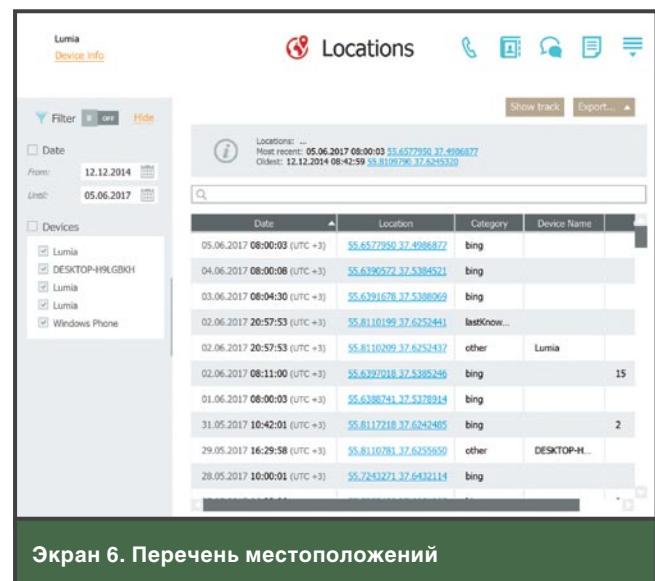
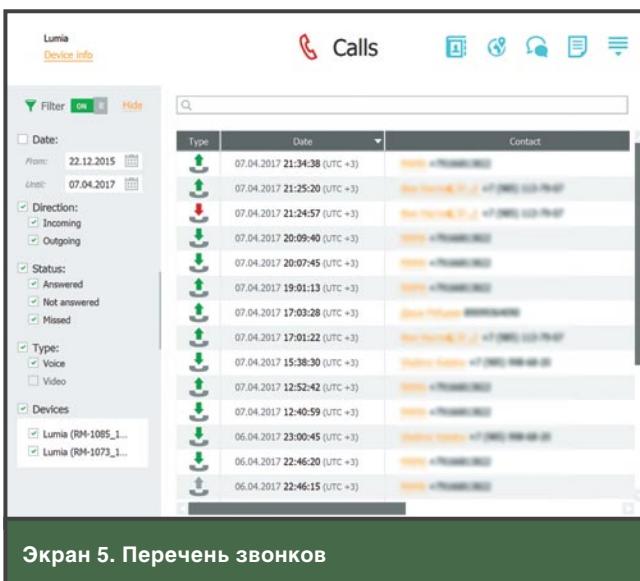
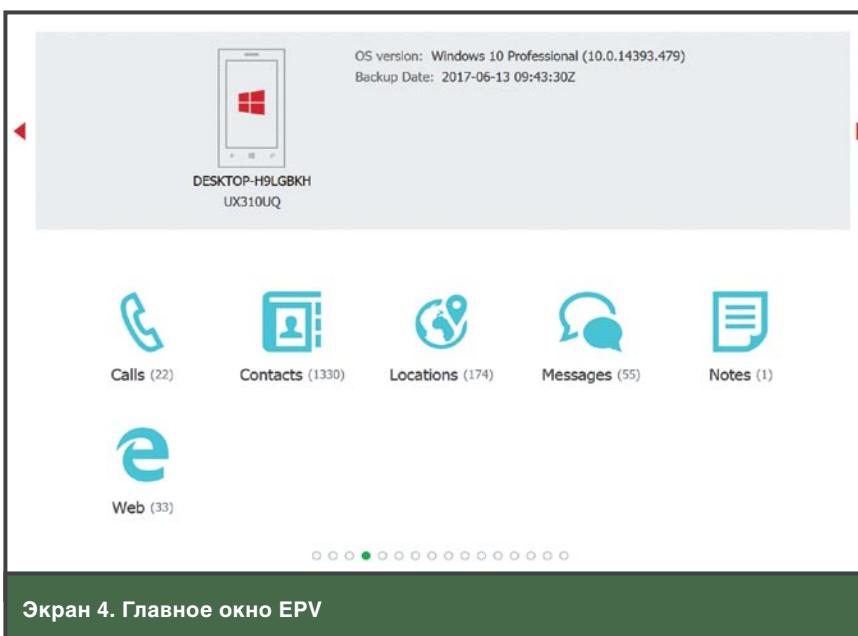
**Звонки.** Количество записей о звонках поражает воображение. Так, в ходе эксперимента мне удалось извлечь из моего телефона 2300 звонков примерно за год. Перечень звонков можно фильтровать (фильтр показан слева на экране 5).

**Locations.** Стоит отметить, что местоположения (Locations) можно обнаружить только при использовании службы поиска Bing (экран 6).

**История поиска.** Опять-таки, историю поиска можно пока найти только при помощи поисковой системы Bing (экран 7).

**SMS.** Необходимо отметить, что можно восстановить лишь синхронизированные сообщения (экран 8). Такая возможность появилась еще в Windows Phone 8, а затем уже в Windows 10 Mobile. Для этого необходимо войти в «Сообщения», «Параметры», «История и синхронизация», нажать «Синхронизировать сообщения между устройствами» (Messaging | Settings | History & sync) и выбрать период синхронизации («Последний месяц», «Последний год», «Всегда»). Если данный параметр включен, то сообщения синхронизируются с «облаком», а значит, их можно вытащить оттуда.

Кроме того, следует отметить, что в «облаке» хранятся, а следователь-



Lumia  
Device info

Filter OFF Hide

Date:  
From: 07.09.2016 Until: 25.05.2017

Records: 15  
Most recent record: 25.05.2017 11:01:15 (UTC +3)  
Oldest record: 23.05.2017 13:44:51 (UTC +3)

Date	Name	URL	Vis
25.05.2017 11:01:15 (UTC +3)	www.microsoft.com	<a href="https://www.microsoft.com...">https://www.microsoft.com...</a>	3
25.05.2017 11:01:00 (UTC +3)	social.technet.microsoft.com	<a href="https://social.technet.micro...">https://social.technet.micro...</a>	1
25.05.2017 11:00:16 (UTC +3)	www.microsoft.com	<a href="https://www.microsoft.com...">https://www.microsoft.com...</a>	1
24.05.2017 09:29:46 (UTC +3)	account.microsoft.com	<a href="https://account.microsoft.c...">https://account.microsoft.c...</a>	1
24.05.2017 09:29:42 (UTC +3)	login.live.com	<a href="https://login.live.com/ppsec...">https://login.live.com/ppsec...</a>	1
24.05.2017 09:29:32 (UTC +3)	login.live.com	<a href="https://login.live.com/login....">https://login.live.com/login....</a>	1
24.05.2017 09:29:29 (UTC +3)	account.microsoft.com	<a href="https://account.microsoft.c...">https://account.microsoft.c...</a>	1
23.05.2017 16:34:07 (UTC +3)	www.elcomsoft.com	<a href="https://www.elcomsoft.com/">https://www.elcomsoft.com/</a>	1
23.05.2017 16:34:03 (UTC +3)	www.passware.com	<a href="https://www.passware.com/">https://www.passware.com/</a>	1

Экран 7. История поиска

но, извлекаются EPV, сведения обо всех устройствах, привязанных к учетной записи. У меня это соответственно:

- Lumia 950 XL Dual SIM;
- Desktop (N150/N210/N220);
- Desktop (Latitude E5450).

Кроме того, для смартфонов показывается и телефонный номер, для десктопов — версия Windows 10 с номером сборки. Даже сохраняются данные по старым устройствам, которые когда-то были (у меня это Lumia 800 и Lumia 900).

Device info

Messages

verify (13) 21 June 2017  
Sms 18:55:16 (UTC +3)  
Use 5969027 as Microsoft account security code

infosms (1) 17 May 2017  
Sms 07:48:49 (UTC +3)  
Код подтверждения для LinkedIn – 038374.

volia (3) 14 May 2017  
0920003700 (90)  
chervonyi (10)

Экран 8. Перечень сообщений

OneDrive

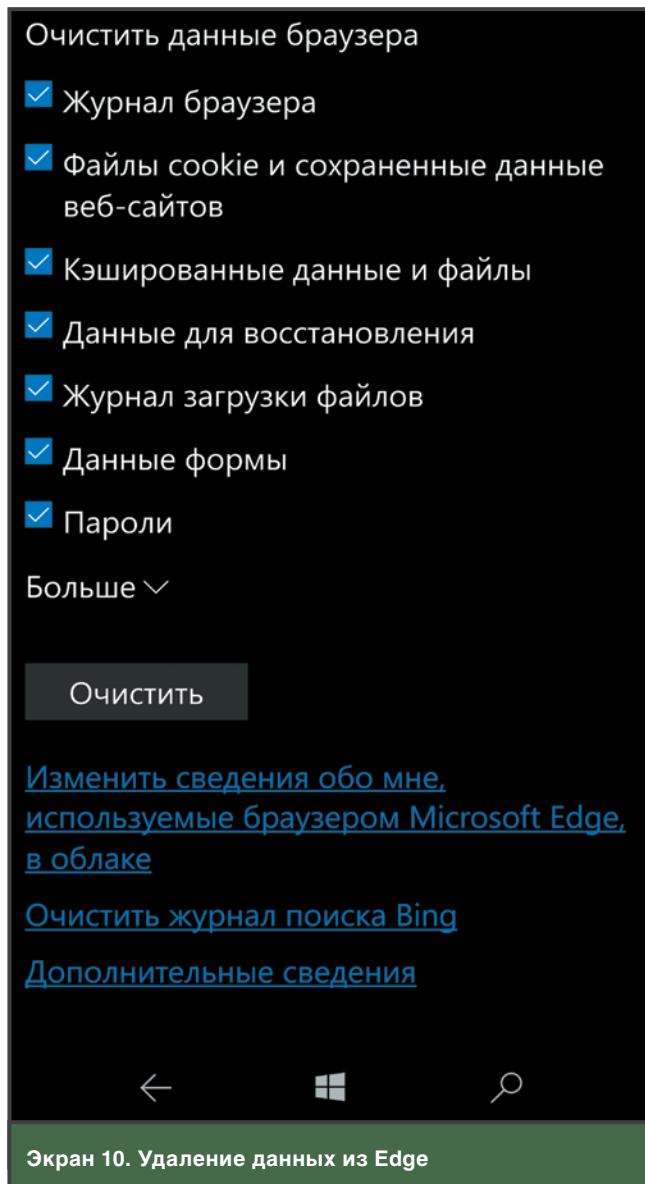
Безмалый Владимир

Заархивированные параметры устройств

Microsoft (Windows phone)	Последнее резервное копирование 19.06.2017	Удалить
Microsoft (Windows phone)	Последнее резервное копирование 11.11.2016	Удалить
Microsoft (Windows phone)	Последнее резервное копирование 12.06.2016	Удалить
Microsoft (Windows phone)	Последнее резервное копирование 16.08.2016	Удалить
Microsoft (Windows phone)	Последнее резервное копирование 18.04.2017	Удалить
Microsoft (Windows phone)	Последнее резервное копирование 01.02.2017	Удалить
Microsoft (Windows phone)	Последнее резервное копирование 09.09.2016	Удалить
Microsoft (Windows phone)	Последнее резервное копирование 26.07.2016	Удалить
Microsoft (Windows phone)	Последнее резервное копирование 16.06.2016	Удалить
Microsoft (Windows phone)	Последнее резервное копирование 24.05.2016	Удалить

ondrive.live.com/Options

Экран 9. Удалить ненужные резервные копии



Экран 10. Удаление данных из Edge

Можно ли удалить данные о вас? Безусловно. Для этого необходимо войти в «Параметры», «Служба архивации», «Другие параметры», «Удалить резервные копии», перейти к OneDrive.com и удалить ненужные резервные копии (экран 9).

Естественно, если у вас нет резервных копий, то и извлекать нечего. Данных просто нет.

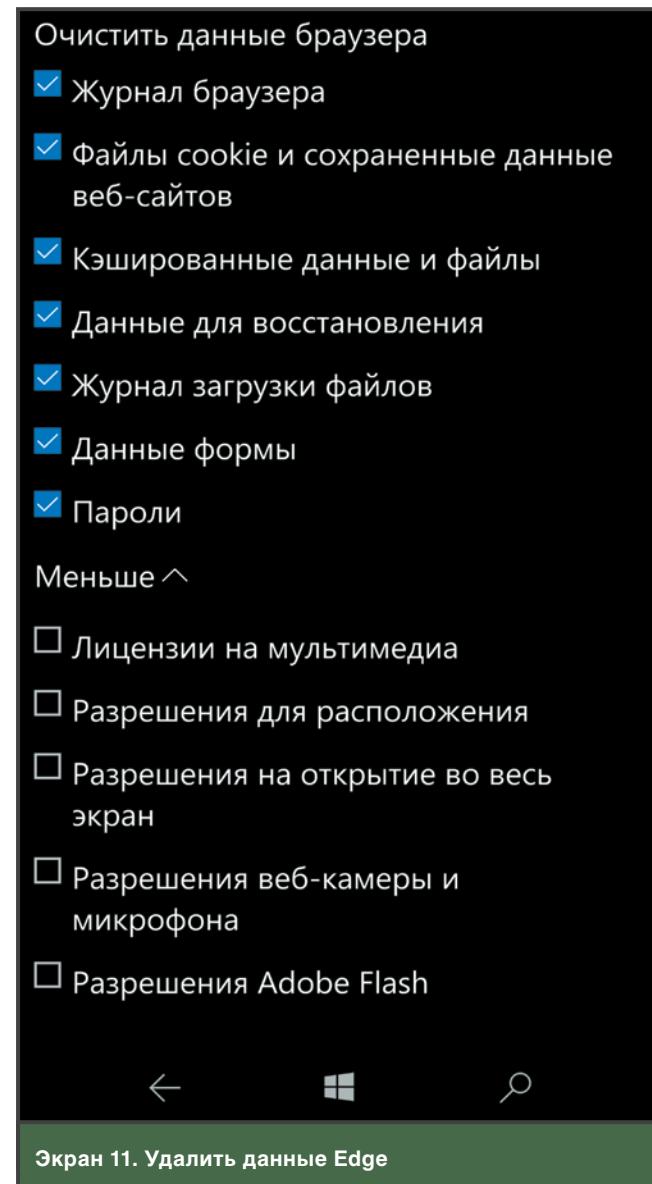
Если же вы хотите очистить историю браузера Edge, то сделать это можно таким образом, как показано на экране 10.

Чтобы удалить дополнительные данные, выберите на экране 10 раздел «Больше». Раскроются дополнительные варианты данных для удаления, показанные на экране 11.

В случае если вы захотите удалить данные браузера Edge, хранящиеся

в «облаке», выберите «Изменить данные обо мне, используемые браузером Microsoft Edge в облаке». На странице <https://account.microsoft.com/privacy/browse?ref=privacy-edge-browse> можно просмотреть и очистить журнал браузера Microsoft Edge, связанный с учетной записью Microsoft, с помощью которой была выполнена регистрация в системе Windows. При этом самые последние данные могут быть еще недоступны. Для просмотра и очистки истории поиска воспользуйтесь страницей <https://account.microsoft.com/privacy/#/search>.

Если же вы хотите запретить Microsoft Edge сохранять журнал браузера в «облаке», вам необходимо на своем устройстве выбрать поле поиска на панели задач и открыть



Экран 11. Удалить данные Edge

«кабинет Cortana». Затем выберите пункт «Заметки», потом «Разрешения». Убедитесь, что параметр «Журнал браузера» отключен. Как видите, для запрета хранения данных о вас в «облаке» нужно выполнить совсем немного действий. Но все же делать это придется вам самим.

Таким образом, мы убедились, что сегодня «облака» добавляют нам удобства и позволяют быстро настроить новые устройства, но необходимо осознавать связанные с этим риски. ♦

# ФИШИНГ, СМИШИНГ, ВИШИНГ

**Н**ачну без предисловий, с определения фишинга из статьи в Википедии (<https://ru.wikipedia.org/wiki/%D0%A4%D0%B8%D1%88%D0%B8%D0%BD%D0%B3>): «Фишинг (англ. phishing, от fishing — рыбная ловля, выуживание — вид интернет-мошенничества, целью которого является получение доступа к конфиденциальным данным пользователей — учетным записям и паролям. Это достигается путем проведения массовых рассылок электронных писем от имени популярных торговых марок, а также личных сообщений внутри различных служб, например от имени банков или внутри социальных сетей. В письме часто содержится прямая ссылка на сайт, внешне неотличимый от настоящего, либо на сайт с редиректом (включенной функцией перенаправления). После того как пользователь попадает на поддельную страницу, мошенники пытаются различными психологическими приемами побудить пользователя ввести на поддельной странице свои имя учетной записи и пароль, которые используются для доступа к определенному сайту, что позволяет мошенникам получить доступ к конфиденциальным учетным записям и банковским счетам».

Фишинг — разновидность социальной инженерии, основанная на незнании пользователями основ сетевой безопасности: в частности, многие не знают о том, что службы не рассылают писем с просьбами сообщить свои учетные данные, пароль и пр. Для защиты от фишинга производители основных интернет-браузеров договорились о применении одинаковых способов информирования пользователей о том, что они открыли подозрительный сайт, который может принадлежать мошенникам.

Вместе с тем стоит отметить, что, несмотря на популярность данного метода у злоумышленников, большое распространение получили и другие разновидности фишинга — смишинг и вишинг.

Смишинг (англ. SMiShing — от «SMS» и «фишинг») — вид фишинга, при котором используются короткие сообщения SMS. Мошенники отправляют жертве SMS-сообщение, содержащее ссылку на фишинговый сайт и мотивирующее получателя зарегистрироваться на этом сайте. Как вариант жертве предлагается отправить в ответном сообщении SMS конфиденциальную информацию, касающуюся реквизитов или персональных параметров доступа на информационно-платежные ресурсы в Интернете. Давайте рассмотрим эту тему подробнее (<https://ru.wikipedia.org/wiki/%D0%A1%D0%BC%D0%B8%D1%88%D0%B8%D0%BD%D0%B3>) и разберемся, как выглядят подобные SMS.

## Мошенничество с использованием SMS и вредоносных программ

Предположим, вы получили SMS-сообщение от компании «Avto\*\*\*» о том, что вы выиграли автомобиль. Текст SMS-сообщения выглядит приблизительно так:

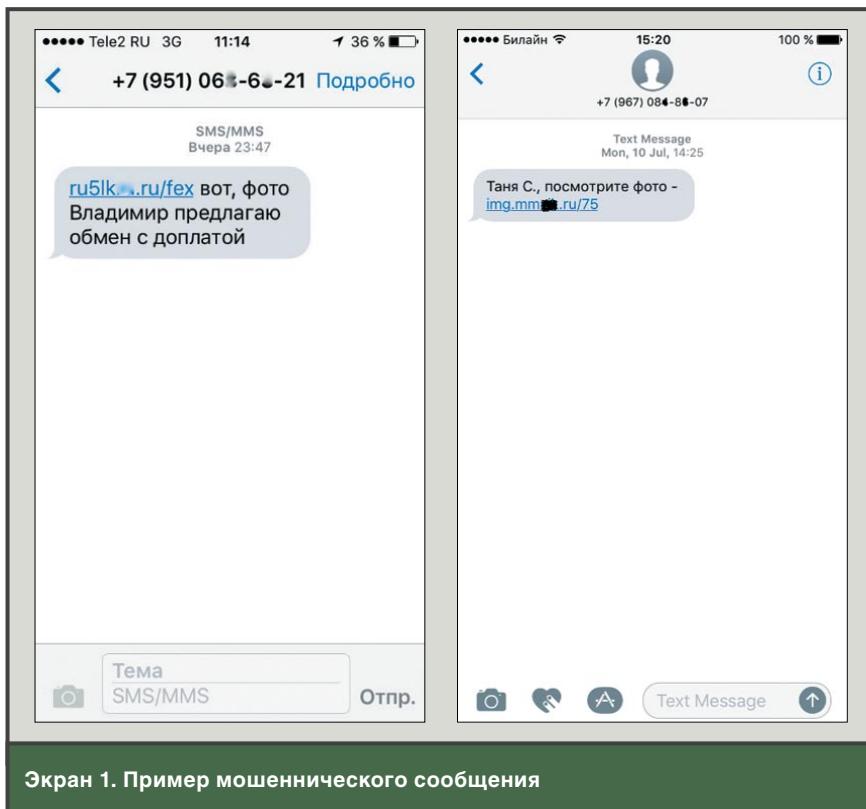
*Pozdravlyayem, po itogam akcii Vi stali obладателем автомobiliya «Chevrolet-AVEO» Info na [www.Avto-\\*\\*\\*\\*\\*.ru](http://www.Avto-*****.ru) ili po tel: +7(919)804-\*\*\_\*\**

Перейдя по ссылке, вы попадете на весьма качественно сделанный сайт, в котором есть раздел об истории компании, гостевая книга, контакты и даже интернет-магазин. У вас попросят копии документов и небольшую сумму на оформление выигрыша.

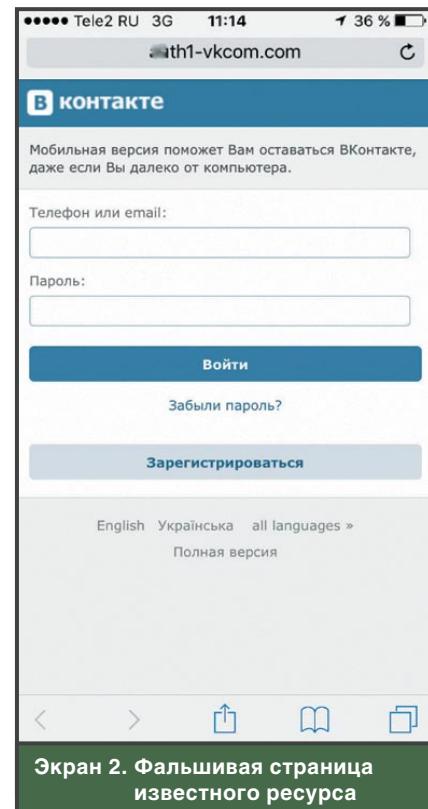
Разбираемся  
с инструментарием  
интернет-  
мошенников

**Владимир Безмалый**

## Вводный курс



Экран 1. Пример мошеннического сообщения



Экран 2. Фальшивая страница известного ресурса

Естественно, никакого автомобиля вы не получите, а деньги просто уйдут злоумышленникам.

Или ситуация может выглядеть так: пользователь получает SMS-сообщение от неизвестных отправителей с содержанием, показанным на экране 1. Получив сообщение, пользователь может заинтересоваться и захочет узнать, что же предлагают взамен и какое фото просят посмотреть. Расчет отправителей таких сообщений прост: усыпить бдительность получателя и так или иначе заставить его перейти на указанный ресурс. Для этого мошенники часто используют персонализированные тексты, с указанием имени и фамилии пользователя. В итоге при переходе по ссылке на мобильное устройство пользователя устанавливается вредоносная программа, которая пытается украсть личную информацию, получить доступ к функциям телефона либо заблокировать или зашифровать смартфон для дальнейшего получения выкупа за разблокировку.

Аналогичная схема применяется для кражи учетных данных социальных сетей. Переходя по ссылкам в сообщении, пользователь может попасть на страницы, очень похожие на глав-

ные страницы популярных ресурсов, например как на экране 2.

Введя свои данные в соответствующее поле, пользователь отдает в руки злоумышленников доступ к учетной записи и, как следствие, к контактам друзей и личным фотографиям. Все это можно использовать в других схемах мошенничества.

### Платные номера

Еще один распространенный вид мошенничества — указание платных номеров в просьбах о помощи. К примеру, сообщения в социальных сетях о срочном поиске донорской крови для умирающего ребенка с указанием лишь номера мобильного телефона. При звонке на данный номер со счета человека, желающего помочь и сдать кровь, снимается определенная сумма денег. Или другой очень распространенный пример: угроза жизни породистых щенков, которых почему-то собираются усыпить, а не продать. Главная опасность такой мошеннической схемы — очень быстрое распространение в социальных сетях. Пользователи делятся данной записью со своими подписчиками и призывают помочь людям или животным. Указанную в записи информацию почти никто не проверяет.

### «Голодные номера»

Другая беда, связанная с благотворительностью и телефонными номерами, — фальшивые сборы средств на так называемые «голодные номера»: номера телефонов, на которые просят перевести деньги. Рассчитывая на человеческое сострадание, мошенники наживаются на самых благих побуждениях, лишая финансовой поддержки тех, кто в этом действительно нуждается. Проверьте реквизиты, указанные в объявлении или сообщении для сбора средств, с помощью поисковых систем. Злоумышленники могут копировать реальные просьбы о помощи, создавать новые похожие сайты и сообщения в социальных сетях, подставляя свои реквизиты, которые могут не меняться продолжительное время.

### Угроза для бизнеса

Стать жертвой мошенников, которых можно вычислить по номеру телефона, могут не только отдельные пользователи, но и компании. Злоумышленники создают точные копии сайтов крупных компаний и размещают их на доменах, немногого отличающихся от оригинальных. На поддельном сайте указывают недостоверные контактные дан-

ные (например, в качестве контакта отдела продаж). Такие сайты используются для незаконного завладения денежными средствами потребителей продукции подлинной компании, чей сайт был скопирован путем получения предоплаты за обещанные контракты. Результатом такой атаки для компании могут стать многомилионные убытки, а также ущерб репутации.

### ВЫМОГАТЕЛЬСТВО ПО SMS

Запугать человека — любимая уловка злоумышленников. Например, отправкой сообщения якобы от близкого человека жертвы с просьбой о срочной финансовой помощи (экран 3).

С подобными рассылками сталкивались многие, и все говорят о том, что автором такого сообщения является мошенник. Однако страх за близких заставляет нас порой совершать нерациональные поступки.

### ВИШИНГ

Впервые такая разновидность мошенничества, как вишинг (vishing — voice fishing), была зафиксирована в 2006 году. Вишинг представляет собой разновидность фишинга и реализуется с использованием программ для автоматического набора номеров, war diallers, а также расширенных функций интернет-телефонии (VoIP). Схема обмана мало чем отличается от фишинга: пользователи платежной системы получают якобы от администрации сообщения по почте, в которых им предлагается прислать свои пароли и счета. Но если в случае с фишингом прилагается ссылка на поддельный сайт, то при вишинге пользователю предлагают позвонить по городскому телефонному номеру. При звонке автомат зачитывает сообщение, в котором абонента просят предоставить свои конфиденциальные данные. Сложность в раскрытии этого вида мошенничества заключается в том, что развитие интернет-телефонии позволяет перенаправлять звонки на городской номер в любую точку мира, причем абонент даже не будет об этом подозревать.

Компания Secure Computing сообщила о самом изощренном способе обмана по схеме вишинга: элек-

тронная почта вообще не использовалась, так как злоумышленники запрограммировали компьютер так, чтобы тот набирал телефонные номера из базы данных и проигрывал заранее записанное сообщение, в котором абонента предупреждали, что сведения о его кредитной карте оказались в руках мошенников, поэтому ему необходимо с телефонной клавиатуры ввести номер кредитной карты и другую информацию.

Теоретически клиент звонит в банк, а на самом деле телефонная линия уже находится под контролем хакеров. В этом случае мошенник просит звонящего сообщить некую учетную информацию, чтобы связаться со службой поддержки банка. По информации Secure Computing, мошенники настраивают программу на набор номеров в конкретном регионе. В момент ответа происходит следующее.

- Автоответчик информирует пользователя, что с его кредитной картой производятся мошеннические действия, и рекомендует быстро перезвонить по некоему номеру.
- Когда жертва перезванивает по этому номеру, ей отвечает «компьютерный голос», говорящий, что пользователь должен пройти идентификацию и ввести номер карты и другие данные с клавиатуры телефона.
- Как только номер карты введен, мошенник получает всю информацию (адрес, номер телефона, полное имя).
- Используя этот звонок, мошенник может получить и другую дополнительную информацию, такую как срок действия карты, PIN-код, номер банковского счета и дата рождения.
- Как защититься от подобного вида мошенничества? Существует несколько простых способов, которые обезопасят вас.
- Все кредитные организации по электронной почте или телефону обращаются к клиенту по имени и фамилии. Если в обращении это не указано, то, скорее всего, имеет место факт мошенничества.
- Ни в коем случае не звоните по вопросам безопасности банковского счета или кредитной



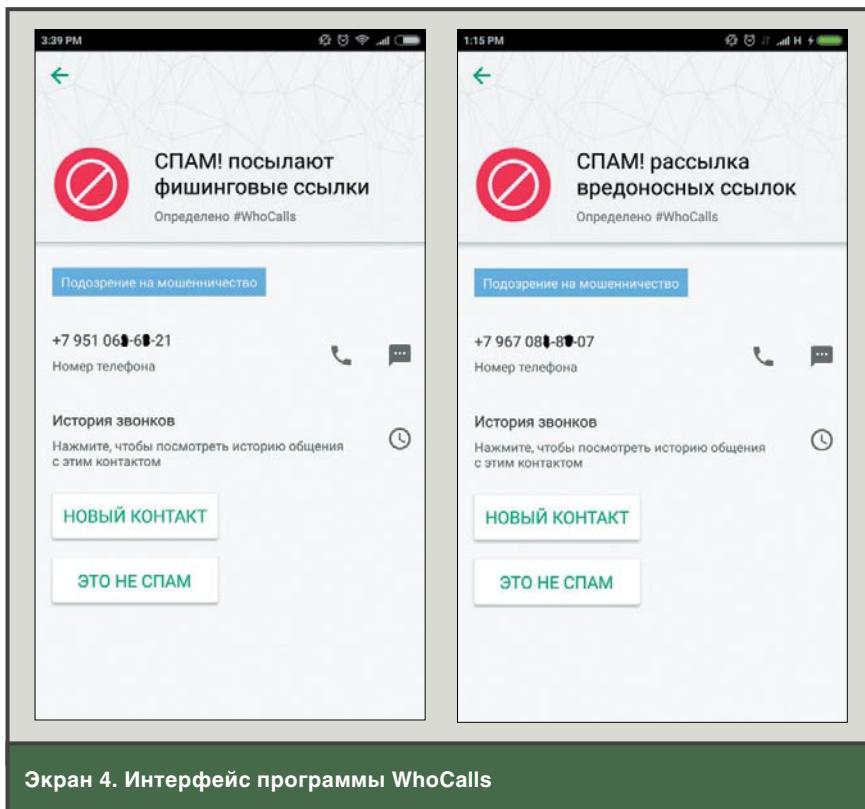
**Экран 3. Пример мошеннического сообщения с требованием перевода денег**

карты по предложенному номеру телефона. На всех платежных картах указывается специальный телефонный номер, по которому вы должны звонить.

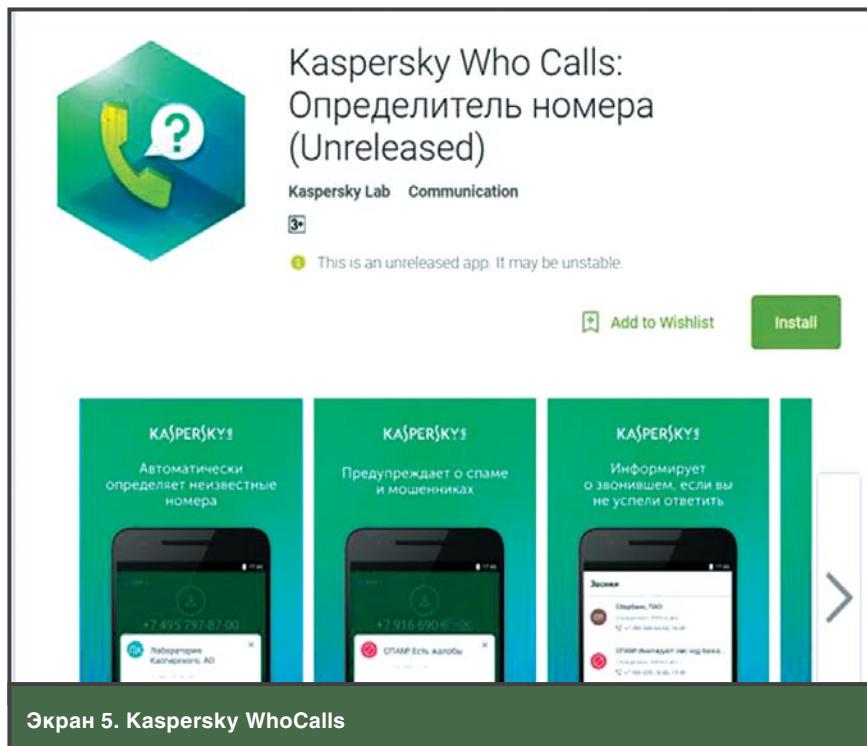
• Если звонящий представляется вашим провайдером и задает вопросы относительно конфиденциальных данных, то он, скорее всего, мошенник.

Что общего в двух описанных способах мошенничества? В обоих случаях используется телефон. И если бы вы знали заранее, кому принадлежит тот или иной номер и могли бы проверить его по базе спамеров, жизнь злоумышленников была бы намного сложнее. Не секрет, что такие программы с базами злоумышленников существуют. Однако стоит упомянуть, что в базах, как правило, представлены иностранные номера. А телефонов из России очень мало. Сегодня ситуация изменилась с выходом нового продукта — Kaspersky Lab WhoCalls.

Программа WhoCalls помогает не только бороться со спамом и избегать нежелательных звонков, ее можно использовать для получения данных о неизвестных отправителях сообщений. Достаточно



Экран 4. Интерфейс программы WhoCalls



Экран 5. Kaspersky WhoCalls

ввести в интерфейсе программы WhoCalls номер телефона, с которого пришло сообщение, и, если в базе есть такой номер, вы увидите всю информацию о конкретном отправителе (экран 4). Что в итоге убережет вас от мошеннических схем и позволит сохранить нервы и деньги. Загрузить данное при-

ложение можно в магазине Google Play.

### Kaspersky Who Calls

Kaspersky WhoCalls — это бесплатный автоматический определитель номера (экран 5), который проверяет все входящие вызовы с неизвестных номеров, чтобы вы точно

знали, стоит ли брать трубку. Вы даже можете самостоятельно добавлять подозрительных личностей в базу данных спамеров, ведь общая безопасность начинается с каждого из нас.

**Сведения о звонящем.** Автоматический определитель номера мгновенно отображает данные о входящем или пропущенном вызове с неизвестного номера, в том числе название организации, на которую зарегистрирован телефон, ее категорию и репутацию номера. Определитель поможет не упустить важные звонки в потоке информационного шума.

**Предупреждение о спаме и мошенничестве.** Составленный пользователями список спамеров оградит вас от навязчивых продавцов или мошенников. Kaspersky WhoCalls может сразу предупредить вас, показав подробную информацию о звонящем, а может автоматически заблокировать известного спамера, чтобы избавить вас от неприятного разговора.

**Блокирование подозрительных вызовов по вашему обращению.** Ваша помощь пригодится для защиты общества от спама и мошенников: достаточно добавить нежелательный номер в список антиспама.

Данная программа определяет неизвестные номера во входящих и пропущенных вызовах, помогает проверять неизвестные телефонные номера, блокирует нежелательные звонки и показывает сведения о них, постоянно повышает точность определения спама за счет обновления соответствующих баз.

Для определения неизвестных и анонимных вызовов Kaspersky WhoCalls не требует от вас номера телефона и списка контактов: программа не извлекает ваши данные и не публикует их. Вместе с тем необходимо учесть, что метки о спаме предоставляются только на основании данных других пользователей, их нельзя считать оценкой компании. 

---

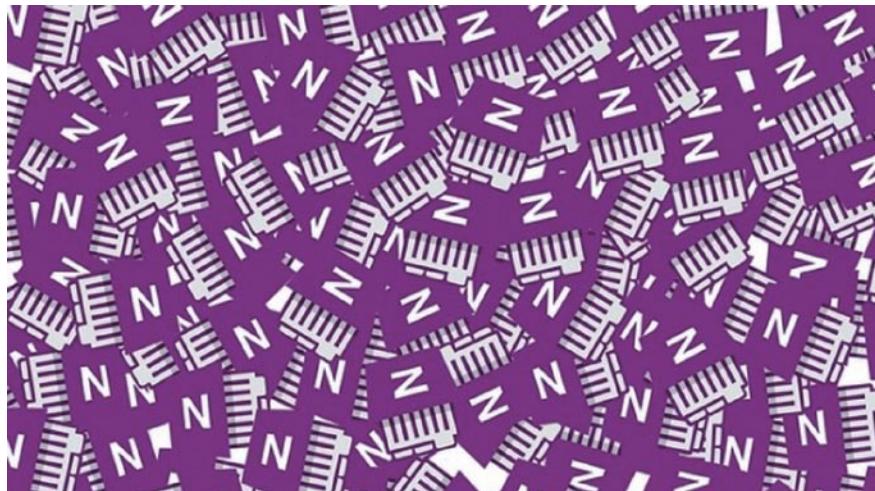
Владимир Безмалый (vladb@windowslive.com) — специалист по обеспечению безопасности, Kaspersky Lab Certified Consultant, Kaspersky Lab Certified Trainer, имеет звания MVP Consumer Security, Microsoft Security Trusted Advisor

# Эффективная работа с OneNote

Использование документа на нескольких устройствах

**M**не уже приходилось в своих статьях рассказывать о том, как я использую OneNote, и вы уже знаете, что эта программа стала для меня незаменимым помощником при работе с документами. Я имею в виду частную переписку, и выполнение профессиональных задач.

Несколько лет назад я принял участие в организованном корпорацией Microsoft семинаре для специалистов, имеющих статус Most Valuable Participant. Это мероприятие было посвящено углубленному изучению программы OneNote. Так вот, после этого семинара я полностью перенес процесс подготовки своих обзоров в среду OneNote. И совершенно забыл о необходимости еженедельно распечатывать кипы документов. Теперь все мои документы представлены в электронном виде и доступны для различных устройств. В этой статье я хочу рассказать вам, как с помощью OneNote повысить эффективность работы с документами, которые я собираю в течение дня для подготовки еженедельных обзоров.



## Как это делается

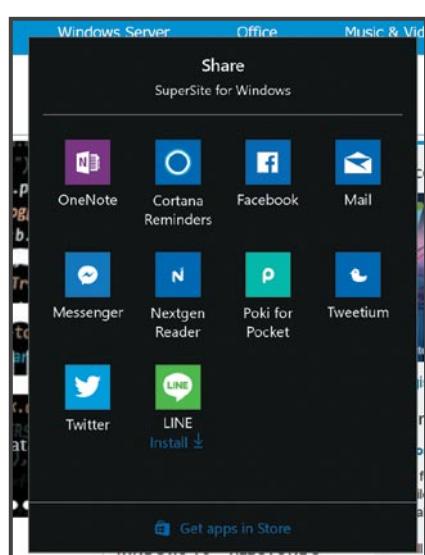
Всякий раз, когда мне попадается интересная статья, которую я, возможно, включу в свой еженедельный обзор, я загружаю ее в браузер Microsoft Edge, а затем с помощью диалогового окна Share Dialog пересылаю в программу OneNote (экран 1).

Я выбираю OneNote в качестве целевого объекта, и на экране появляется диалоговое окно, показанное на экране 2.

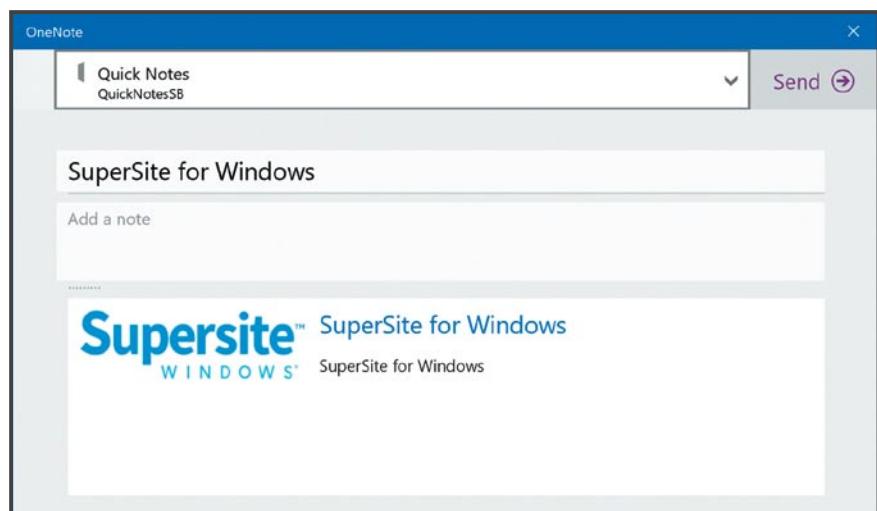
Все это превосходно, однако в качестве объекта разделения в данном окне по умолчанию указывается

элемент Notebook и разделы, к которым я обращался в последнем сеансе работы с приложением OneNote на платформе UWP. Для обзора у меня определен особый элемент Notebook, и я всегда стараюсь перед выходом из приложения не забыть о выделении соответствующего раздела, но, увы, чаще всего я об этом все-таки забываю.

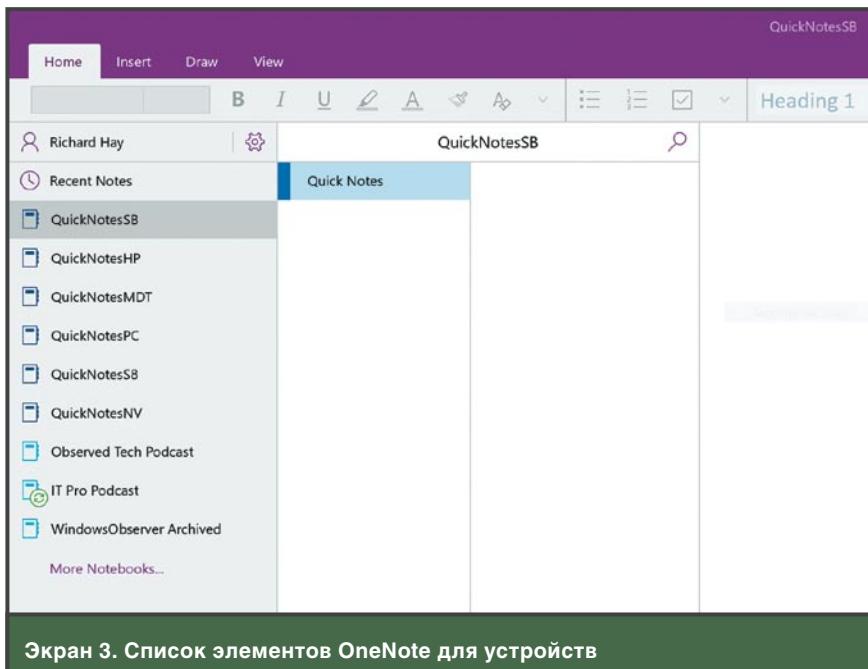
В итоге при попытке совместного использования файла разными устройствами в OneNote у меня возникала проблема. Когда на экране появлялось соответствующее диалоговое окно, я постоянно отмечал не то, что



Экран 1. Отправка документа в OneNote



Экран 2. Вновь созданный элемент Notebook



Экран 3. Список элементов OneNote для устройств

нужно. И мне приходилось закрывать это диалоговое окно, открывать приложение OneNote на платформе UWP и выбирать пункт Podcast Items Section, чтобы определить общий файл в нужное место соответствующего элемента Notebook. Со временем это, конечно, начинает немного раздражать.

Надо сказать, что в программе OneNote реализована функция, предусматривающая определение для каждого установленного в системе экземпляра OneNote выделенного раздела Quick Notes Section. Эта функция всегда доступна при раскрытии списка окна использования файлов, по умолчанию отображаемого в верхней части экрана. Моя идея состояла в том, чтобы создать новый элемент OneNote Notebook, разместить в нем применяемое по умолчанию приложение Quick Notes и затем использовать один и тот же раздел Quick Notes Section для всех своих устройств (а это две настольные системы, два ноутбука и небольшой планшет).

В результате я получил бы единую папку для совместно используемых файлов, которые были бы также доступны в диалоговых окнах OneNote Sharing Dialog на всех моих устройствах. Во всяком случае, такого результата я ожидал.

Но как только я начал открывать этот элемент OneNote Notebook на своих устройствах и указывать его в качестве

используемого по умолчанию места для хранения моих заметок Quick Notes, список папок начал множиться по всем моим устройствам в данном элементе Notebook и приобрел в результате следующий вид:

- Quick Notes;
- Quick Notes 1;
- Quick Notes 2;
- Quick Notes 3;
- Quick Notes 4.

Иными словами, одна папка Quick Notes не могла совместно использоваться всеми моими экземплярами OneNote как один целевой объект Quick Notes.

## Решение

Я рассудил, что обойти эти ограничения можно, в частности, с помощью создания нового элемента Notebook для каждого из устройств. Этому элементу можно дать имя QuickNotesXXX, где XXX — это кодовое обозначение для идентификации устройства. Далее нужно указать данный элемент Notebook в качестве местонахождения хранилища заметок Quick Notes на соответствующей системе.

В этом случае, надо отметить, я всегда буду иметь доступ к данной области для совместного использования в раскрывающем списке представленного на экране 2 диалогового окна OneNote Sharing Dialog.

Далее я решил открыть каждый из элементов Notebook в программе

OneNote на всех своих устройствах, а также открыть элементы Notebook, выделенные для обзоров, что позволило бы мне с легкостью перемещать общие объекты из различных экземпляров Quick Notes в папку Podcast Items и соответственно использовать их в ходе подготовки еженедельных обзоров.

На экране 3 показано, как выглядит этот список элементов в моих устройствах при использовании указанных настроек.

В процессе усовершенствования приложения OneNote на платформе UWP набор его функциональных возможностей значительно расширился. Так, сегодня пользователи могут перемещать объекты из одного элемента Notebook в другой, не выходя из программы. Я, например, пользуюсь этой возможностью регулярно, когда готовлю заметки для своих еженедельных обзоров. Скажу больше, сейчас я вообще не работаю с настольной версией пакета OneNote, поскольку вполне обхожусь без нее.

Не могу не упомянуть и еще об одном обстоятельстве. Если раньше мне постоянно приходилось укорять себя за то, что, выходя накануне из приложения OneNote на платформе UWP, я не выделял нужный элемент Notebook, теперь такой проблемы не существует. Когда мне требуется сделать зафиксированный браузером Microsoft Edge объект достоянием всех моих устройств, нужная целевая область постоянно находится у меня перед глазами, и все рабочие процессы выполняются гораздо проще.

Конечно, я понимаю, что эта ситуация отражает специфику моей работы. Но из нее можно извлечь и более общий урок, и сводится он к следующему. Чтобы та или иная технология эффективно выполняла стоящие перед нами задачи, мы всегда можем, оставаясь в рамках этой технологии, изыскивать некие альтернативные методы. Между прочим, именно эта возможность гибкого подхода к решению привлекает меня в технологиях вообще.



Ричард Хэй ([winobs@outlook.com](mailto:winobs@outlook.com)) имеет звание Microsoft MVP в категории Windows Operating System с 2010 года



[www.dgl.ru](http://www.dgl.ru)



СТАТЬИ

НОВОСТИ

ОБЗОРЫ

25 ОКТЯБРЯ

# SMART INDUSTRY & CITY 2017

МЕСТО ПРОВЕДЕНИЯ

Radisson Blu Belorusskaya  
г. Москва, 3-я улица Ямского Поля, 26А



ЦИФРОВАЯ ЭКОНОМИКА – наше будущее  
ЦИФРОВАЯ ТРАНСФОРМАЦИЯ – наше настоящее



Андрей Белозеров  
советник министра  
Правительства Москвы  
Артема Ермолаева по стратегическим  
проектам и инновациям



Гульнара Хасьяднова  
генеральный директор,  
«Микрон»



Максим Сонных  
руководитель отдела  
промышленной  
автоматизации, «Бош Рекрот»



Вадим Пестун  
директор по ИТ,  
СУЭК



Константин Горбач  
директор по продажам,  
Yandex Data Factory



Александр Герасимов  
директор направления  
анализа рынков облачных  
и ИТ-услуг, Json & Partner Consulting



Владислав Беляев  
директор по ИТ,  
Группа «Черкизово»

12+  
Реклама

Цена участия  
до 31 августа

**5940 руб.**

Цена участия с 1 сентября **9900 руб.**

По вопросам участия: Ольга Пуркина



+7 (499) 703-1854, +7 (495) 725-4780



kon@osp.ru