



Microsoft Exchange 2010

Упрощаем обслуживание с помощью агентов сценариев

С выходом Microsoft Exchange 2010 изменился подход к вопросу автоматизации труда администратора. Если раньше для этого приходилось писать сложные скрипты, то теперь появились агенты расширения командлетов (Cmdlet Extension Agents), которые призваны упростить написание сценариев, автоматизирующих работу сервера

Ранее концептуальная модель программных продуктов компании Microsoft сводилась к тому, что всю функциональность продукта обслуживала графическая консоль управления, а командная консоль покрывала лишь небольшое подмножество административных задач. С выходом на рынок PowerShell подход к построению модели программных продуктов полностью изменился. Начиная с версии Microsoft Exchange Server 2007 архитектурная модель сервера стала базироваться на Microsoft .NET Framework. В результате главную роль стали играть командлеты PowerShell, именно они теперь покрывают все функциональные возможности сервера, а задача графической консоли сводится лишь к визуализации их использования. В итоге графическая консоль управления сервером Microsoft Exchange 2007/2010 реализует лишь небольшую часть функционала. Исходя из этого можно сделать вывод: использование любых инструментов управления, будь то Exchange Management Console (EMC), Exchange Control Panel (ECP) или тем более Exchange Management Shell (EMS), неминуемо приведет к запуску определенных командлетов.

Для примера рассмотрим достаточно часто используемое действие — создание почтового ящика пользователя. Это можно сделать разными путями, но все они приведут к тому, что будет запущен командлет New-Mailbox. Данный командлет имеет несколько десятков параметров, но тем не менее все они не могут перекрыть то количество настроек, которое содержит в себе объект Active Directory и почтовый ящик, с ним связанный. К примеру, мы не можем при создании почтового ящика включить или отключить такие функции, как доступ по POP3/IMAP, Exchange ActiveSync и т.п., не можем автоматически, в зависимости от определенных условий, таких как принадлежность к определенной Organizational Unit (OU) и т.п., изменить размер ящика или назначить определенную политику для Outlook Web App, не можем отправить письмо новому пользователю с пожеланием плодотворного сотрудничества и списком внутренних телефонных номеров коллег и еще не можем выполнить достаточно много аналогичных действий, которые реализуются на уровне других функциональных механизмов сервера.

Задач по автоматизации и облегчению административного обслуживания достаточно много, и родились они не сегодня, а соответственно ранее уже решались. Что же изменилось в этом плане с приходом сервера Microsoft Exchange 2010? А то, что если ранее приходилось писать, прямо скажем, непростые скрипты по отслеживанию того или иного события, которые в свою очередь часто довольно сильно нагружали систему своими постоянными проверками, то теперь в Exchange 2010 появились агенты расширения командлетов, которые призваны серьезно упростить написание сценариев, нацеленных на автоматизацию работы сервера.

Агенты расширения командлетов (Cmdlet Extension Agents)

Это компоненты Microsoft Exchange Server 2010, которые активируются в момент выполнения определенного командлета. Следовательно теперь не нужно постоянно проверять систему на наличие изменений, достаточно просто использовать агент расширения, который сам запустит нужный скрипт в момент попытки внести искомые изменения.

Примечание: агенты расширения командлетов доступны во всех ролях Microsoft Exchange 2010, кроме роли пограничного транспорта (Edge), и не доступны в любых других продуктах компании Microsoft. При этом Get-командлеты не могут вызывать агенты сценариев.

По умолчанию в Microsoft Exchange 2010 существует несколько встроенных агентов расширения, каждый со своим приоритетом. Приоритет используется для того, чтобы разрешить спорные ситуации. Если два агента пытаются установить одно и то же свойство, то в таком случае будут использованы изменения агента с наименьшим значением приоритета, т.е. внесенные изменения агентом с приоритетом 5 будут игнорированы, если тоже свойство будет изменено агентом с приоритетом 4. Приоритет агентов можно изменять.

Список существующих агентов можно получить командой:

```
Get-CmdletExtensionAgent | ft Name, Priority, Enabled
```

В результате ее выполнения на сервере Microsoft Exchange 2010 SP1 мы увидим следующую таблицу:



Внедрение в MS Exchange 2010 функционала Cmdlet Extension Agents позволяет более удобно использовать PowerShell

Agent name	Priority	Enabled by default
Admin Audit Log Agent	255	True
Scripting Agent	6	False
OAB Resources Management Agent	5	True
Provisioning Policy Agent	4	True
Mailbox Creation Time Agent	3	True
Mailbox Resources Management Agent	2	True
Rus Agent	1	True
Query Base DN Agent	0	True

Настройки агентов хранятся на уровне организации, следовательно внесенные изменения реплицируются на все серверы организации. Исключением здесь являются агенты сценариев (Scripting Agent), которые нужно конфигурировать на каждом сервере отдельно.

Агенты сценариев (Scripting Agents)

Используются для внесения собственной логики в работу командлетов сервера Exchange 2010. При инициализации запуска любого командлета на сервере агентом расширения командлетов будет вызван агент сценария, который проверит наличие для данного командлета пользовательского скрипта. Агенты сценариев имеют свой собственный приоритет, по умолчанию, в Microsoft Exchange 2010 SP1 равный шести, и внесенные ими изменения могут быть перезаписаны другими агентами расширения с более низким значением приоритета (0..5). Принцип работы агентов сценариев сводится к обработке логики, заданной в конфигурационном файле ScriptingAgentConfig.xml.

Конфигурационный файл агентов сценариев представляет собой XML-файл, расположенный в директории установки Exchange, – \bin\CmdletExtensionAgents, по умолчанию это C:\Program Files\Microsoft\Exchange Server\V14\Bin\CmdletExtensionAgents. Файл имеет конструкцию вида:

```
<?xml version="1.0" encoding="utf-8" ?>
<Configuration version="1.0">
  <Feature Name="Name" Cmdlets="Your_Cmdlet">
    <ApiCall Name="API_Name">
      <Your_Script>
        <Common>
          </Common>
        </ApiCall>
      </Feature>
    </Configuration>
```

Основные параметры здесь:

Feature – используется для обозначения имени функции (Name), которая будет запускаться при инициализации указанных командлетов (Cmdlets).

ApiCall – определяет в какой именно момент работы командлета нужно выполнять указанное действие. Может быть использовано четыре вида вызова API:

ProvisionDefaultProperties – используется для установки определенных значений свойств объектов командлета в ситуациях, когда эти свойства либо не заданы пользователем, либо нужно перезаписать указанные пользователем свойства.

UpdateAffectedConfigurable – используется для установки свойств объектов, после того как все обработки уже завершены, но до того, как внесенные командлетом изменения будут записаны.

Validate – используется для проверки значений свойств объекта до того как командлет их применит. В результате если командлет пройдет проверку, то он будет запущен на выполнение, если нет, то его выполнение прервется, и будет выведена соответствующая ошибка.

OnComplete – используется для выполнения каких-либо действий уже после того, как работа командлета будет завершена.

Common – содержит функции, которые могут быть использованы любыми скриптами в этом файле.

В директории \Bin\CmdletExtensionAgents лежит файл ScriptingAgentConfig.xml.sample, который является примером конфигурационного файла, его можно переименовать, отредактировать скрипты как вам нужно и использовать.

По умолчанию агенты сценариев отключены и активируются командой:

```
Enable-CmdletExtensionAgent "Scripting Agent"
```

Важно! Если вы активировали агенты сценариев до того, как создали файл ScriptingAgentConfig.xml, то вы не сможете запустить никакие командлеты, кроме Get-командлетов.

В целях демонстрации функционала решим практическую задачу.

Задача

Необходимо для всех вновь создаваемых на сервере почтовых ящиков выполнять следующие настройки:

- > Отключать возможность доступа по POP3 и IMAP.
- > Сотрудникам, добавляемым в Organizational Unit «Top_Managers» установить увеличенный размер почтового ящика и не отключать POP3/IMAP-доступ.
- > Отправлять всем новым сотрудникам сообщение с определенным текстом и файлом, содержащим внутренние телефонные номера коллег.

Для решения создадим файл ScriptingAgentConfig.xml:

```
<?xml version="1.0" encoding="utf-8" ?>
<Configuration version="1.0">
  <Feature Name="CreatingNewMailbox" Cmdlets="new-mailbox">
    <ApiCall Name="OnComplete">
      if($succeeded) {
        $mailbox = $provisioningHandler. .
          UserSpecifiedParameters["Name"]
        $OU = $provisioningHandler. .
          UserSpecifiedParameters["OrganizationalUnit"]
        if($OU -like "test.local/Top_Managers") {
          set-Mailbox -Identity $mailbox -IssueWarningQuota .
            209715200 -ProhibitSendQuota 262144000 .
            -ProhibitSendReceiveQuota 293601280 .
            -UseDatabaseQuotaDefaults $false
        }
      }
    </ApiCall>
  </Feature>
</Configuration>
```

Здесь указывается, что после удачного (if(\$succeeded)) выполнения (OnComplete) командлета New-Mailbox (Feature Cmdlets) необходимо выполнить ряд действий, для которых в качестве переменных нужно использовать значения Name и OrganizationalUnit, полученные от выполняемого командлета. А также необходимо запустить PowerShell-скрипт c:\MailScript.ps1, передав в него значение переменной \$mailbox.

Скриптом MailScript.ps1 будет отправлено электронное сообщение новому пользователю с помощью командлета Send-MailMessage.

После сохранения файла ScriptingAgentConfig.xml активируем агенты сценариев командой:

```
Enable-CmdletExtensionAgent "Scripting Agent"
```

Аналогичным образом можно решить задачу контроля именования вновь создаваемых групп рассылки в тех случаях, когда у вас существуют корпоративные стандарты такого именования. Делается это с помощью проверки запускаемой команды еще до ее выполнения с помощью вызова API Validate (подробнее см. файл ScriptingAgentConfig.xml.sample).

PowerShell – это мощный инструмент, который позволяет администраторам упростить и автоматизировать выполнение каждодневных операций, а с внедрением в Microsoft Exchange 2010 функционала Cmdlet Extension Agents использование этого инструмента становится еще более удобным и доступным. **EOF**

Реклама

Конференция

Microsoft Quality Assurance Day 22 февраля 2011

Для тех, кому не безразлично качество ПО

Microsoft®

iCAREERLAB
Центр профессионального роста

SQA LAB
ЛАБОРАТОРИЯ ТЕСТИРОВАНИЯ