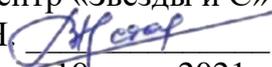


**ЧАСТНОЕ УЧРЕЖДЕНИЕ
ОРГАНИЗАЦИЯ ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО
ОБРАЗОВАНИЯ «УЧЕБНЫЙ ЦЕНТР «ЗВЕЗДЫ И С»**

УТВЕРЖДАЮ

Директор

ЧУ ОДПО «Учебный центр «Звезды и С»

Стародубцев В.Н. 

«19» мая 2021г.



Дополнительная профессиональная образовательная программа
повышения квалификации
М10962 «Расширенные возможности по автоматизации администрирования с
помощью Windows PowerShell»

Москва, 2021г.

1. Целевая установка

2. Планируемые результаты обучения

Реализация Программы направлена на повышение профессионального уровня в рамках имеющейся квалификации, определяемой профессиональным стандартом «06.026 Системный администратор информационно-коммуникационных систем», утвержденным Приказом Минтруда России от 05.10.2015 N 684н "Об утверждении профессионального стандарта "Системный администратор информационно-коммуникационных систем".

Совершенствуемые компетенции

Администрирование системного программного обеспечения инфокоммуникационной системы организации.

№	Компетенция	Код компетенции
1	Установка системного программного обеспечения	F/01.7
2	Оптимизация работы дисковой подсистемы (подсистемы ввода-вывода)	F/02.7
3	Администрирование файловых систем	F/03.7
4	Оценка критичности возникновения инцидентов для системного программного обеспечения	F/04.7
5	Реализация регламентов обеспечения информационной безопасности системного программного обеспечения инфокоммуникационной системы организации	F/05.7

Категория слушателей: администраторы и ИТ-специалисты, которые желают получить практический опыт работы с современными средствами и управлять решениями на технологиях компании Microsoft

Требования к предварительной подготовке: знание и практический опыт администрирования клиентов и серверов семейства Windows; «Английский язык. Уровень 2. Elementary, часть 2», или эквивалентная подготовка

3. Учебный план.

№ п/п	Наименование модулей по программе	В том числе аудиторных			Форма контроля
		Всего	Лекций	Практических занятий	
1	Создание продвинутых функций	6	3	3	Прак. занятие
2	Использование командлетов и .NET Framework и REST API в Windows PowerShell	6	3	3	Прак. занятие
3	Написание управляющих скриптов	6	4	2	Прак. занятие
4	Обработка ошибок скриптов	4	2	2	Прак. занятие
5	Использование файлов с XML, JSON и настраиваемым форматом	4	2	2	Прак. занятие
6	Расширение управления серверами при помощи Desired State Configuration (DSC) и Just Enough Administration (JEA)	4	2	2	Прак. занятие
7	Анализ и отладка скриптов	4	2	2	Прак. занятие
8	Рабочие потоки (Workflow) Windows	4	2	2	Прак. занятие
9	Итоговая аттестация: (Лабораторная работа)	2	-	2	Прак. занятие
	Итого:	40	20	20	

4. Календарный учебный график

Календарный учебный график составляется в форме расписания занятий при наборе группы и прилагается к программе повышения квалификации.

Форма обучения: очная, очная с применением дистанционных технологий.
Трудоемкость программы: 24 ак. ч. Академический час устанавливается продолжительностью 45 минут.
Сроки освоения программы: 3 дн.
Режим занятий: дневной, вечерний.

5. Рабочие программы дисциплин

Модуль 1. Создание продвинутых функций

- Конвертация команд в продвинутые функции.
- Создание модулей скриптов.
- Определение параметров атрибутов и проверки ввода.
- Написание функций, использующих множество объектов.
- Написание функций, принимающих конвейерный ввод (Pipeline).
- Создание комплексного вывода функций.
- Документирование функций при помощи справки на базе комментария.
- Поддержка параметров -Whatif и -Confirm.

Модуль 2. Использование командлетов и .NET Framework и REST API в Windows PowerShell

- Использование Microsoft .NET Framework в Windows PowerShell.
- Использование REST API в Windows PowerShell.

Модуль 3. Написание управляющих скриптов

- Введение в управляющие скрипты.
- Написание управляющих скриптов, которые показывают пользовательский интерфейс.
- Написание управляющих скриптов, которые создают отчеты.

Модуль 4. Обработка ошибок скриптов

- Описание обработки ошибок.
- Обработка ошибок и ожидание в скриптах.

Модуль 5. Использование файлов с XML, JSON и настраиваемым форматом

- Чтение, изменение и запись данных в формате XML.
- Чтение, изменение и запись данных в формате JSON.

- Чтение, изменение и запись данных в настраиваемом формате.

Модуль 6. Расширение управления серверами при помощи Desired State Configuration (DSC) и Just Enough Administration (JEA)

- Описание Desired State Configuration (DSC).
- Создание и развертывание конфигураций DSC.
- Применение Just Enough Administration (JEA).

Модуль 7. Анализ и отладка скриптов

- Отладка в Windows PowerShell.
- Анализ и отладка существующего скрипта.

Модуль 8. Рабочие потоки (Workflow) Windows

Описание рабочих потоков Windows PowerShell.

6. Организационно-педагогические условия реализации программы

6.1. Материально-технические условия реализации программы

Исполнитель обеспечивает для проведения обучения следующие средства вычислительной техники:

- персональный компьютер для преподавателя – 1 шт.;
- персональный компьютер для каждого Слушателя;
- проектор и экран – 1 комплект;
- доска – 1 шт.

Персональные компьютеры объединены в локальную вычислительную сеть. Технические характеристики персональных компьютеров:

- процессор 4 ядра 3,1 ГГц;
- оперативная память - 32 Гб;
- SSD + 2 HDD в RAID0 не менее 500Гб;
- два монитора (24' + 22' FullHD);
- комплект клавиатура и мышь.

6.2. Учебно-методическое обеспечение программы

Каждый Слушатель обеспечивается авторизованным учебным пособием на английском языке.

7. Требования к профессорско-преподавательскому составу

Высшее профессиональное образование и стаж работы в образовательном учреждении не менее 1 года. Статус Microsoft Certified Trainer.

8. Форма аттестации

Текущий контроль успеваемости и качества подготовки, промежуточная и

итоговая аттестации слушателей осуществляются в процессе изучения, освоения данной профессиональной образовательной программы повышения квалификации.

Текущий контроль успеваемости и качества подготовки осуществляется в пределах времени, отведенного на учебные занятия, и выполняет одновременно обучающую функцию. Текущий контроль успеваемости проводится в процессе изучения каждого раздела (темы, подтемы) внутри модуля данной дополнительной профессиональной программы и проводится в форме устного опроса преподавателя. Промежуточная и итоговая аттестации проводятся в форме лабораторных работ на персональном компьютере слушателя, который использовался во время обучения, в классе под наблюдением преподавателя. По окончании каждого модуля рабочей программы проводится промежуточная аттестация в виде промежуточной лабораторной работы по теме каждого модуля данной профессиональной образовательной программы.

Итоговая аттестация проводится в форме итоговой лабораторной работы. В итоговой лабораторной работе задействуются материалы из всех модулей пройденной программы.

Аттестация считается пройденной в случае успешного завершения лабораторной работы, а именно: выполнения поставленной задачи: «Анализ и отладка существующего скрипта».

Время выполнения итоговой аттестации – 2 ак. часа.

9. Оценочные материалы к итоговой аттестации

Итоговая аттестация проводится в форме выполнения задания. Результаты итоговой аттестации слушателей выставляются по двух бальной шкале («зачтено\не зачтено»). Итоговая аттестация считается пройденной («зачтено»), если слушатель выполнил все лабораторные работы и итоговое задание.

Пример решения задач:

Развертывание SharePoint 2010 с помощью скриптов Windows PowerShell.

На протяжении жизненного цикла установки SharePoint определенные задачи, например установку и создание фермы SharePoint приходится выполнять всего один раз. Тем не менее, в крупных организациях с разнообразными средами и большим числом разработчиков зачастую предварительно разрабатывают базовые скрипты, на основе которых выполняется построение интерфейсных веб-серверов SharePoint (WFE) и

серверов приложений SharePoint, где размещаются службы приложений, например службы InfoPath, а также службы поисковых запросов и индексации. В предыдущих версиях SharePoint подобные задачи можно было выполнять с помощью пакетных файлов и средств командной строки SharePoint psconfig и stsadm. Эти средства по-прежнему можно использовать для автоматизации процесса установки SharePoint 2010 и серверов SharePoint Foundation 2010. Дополнительные сведения об использовании этих средств см. в главе 4, посвященной установке SharePoint 2010 (здесь мы не будем подробно останавливаться на этом вопросе). Корпорацией Майкрософт разработан скрипт, рекомендуемый к применению для создания фермы SharePoint с нуля с помощью оболочки Windows PowerShell. Порядок используемых команд и параметров может показаться достаточно сложным, поэтому рекомендуется использовать модуль Windows PowerShell SPModule, который задокументирован в статье библиотеки Microsoft TechNet Установка SharePoint Server 2010 с помощью Windows PowerShell.

Дополнительные сведения о возможностях автоматизации в SharePoint и использовании средств psconfig, stsadm и оболочки Windows PowerShell см. на странице <http://stsadm.blogspot.com/> (Возможно, на английском языке).

При работе с установкой SharePoint, необходимо объединить журналы. В операционных системах Windows ведутся журналы событий, сведения в которых позволяют упростить выполнение задач администрирования. С помощью командлета Windows PowerShell Get-Eventlog можно получить подробные данные из этих журналов. Кроме того, в SharePoint представлена единая служба ведения журнала, в которой содержатся события всех журналов приложений. Также поддерживается интеграция с этой службой стороннего программного обеспечения ведения журнала. В SharePoint 2010 используется несколько командлетов Windows PowerShell для извлечения данных и настройки единой службы ведения журнала. Чтобы определить параметры настройки этой службы в ферме, введите следующую команду. (Также показан пример выходных данных этой команды.)

```
Get-SPDiagnosticConfig
```

```
AllowLegacyTraceProviders : False
```

```
CustomerExperienceImprovementProgramEnabled : True
```

```
ErrorReportingEnabled : True
```

```
ErrorReportingAutomaticUploadEnabled : True
```

```
DownloadErrorReportingUpdatesEnabled : True
```

```
DaysToKeepLogs : 14
```

```
LogMaxDiskSpaceUsageEnabled : True
```

```
LogDiskSpaceUsageGB : 3
```

```
LogLocation : %CommonProgramFiles%\Microsoft Shared\Web Server
```

```
Extensions\14\LOG S\  
LogCutInterval : 30  
EventLogFloodProtectionEnabled : True  
EventLogFloodProtectionThreshold : 5  
EventLogFloodProtectionTriggerPeriod : 2  
EventLogFloodProtectionQuietPeriod : 2  
EventLogFloodProtectionNotifyInterval : 5  
ScriptErrorReportingEnabled : True  
ScriptErrorReportingRequireAuth : True  
ScriptErrorReportingDelay : 60  
AllowLegacyTraceProviders :  
FalseCustomerExperienceImprovementProgramEnabled :  
TrueErrorReportingEnabled
```

В процессе диагностики проблем можно использовать командлеты, связанные с объектом `SPLogEvent`. Для выполнения этих командлетов необходимы права администратора (возможно, потребуется запустить командную консоль SharePoint 2010 от имени администратора). Далее эти команды используются аналогично следующему примеру.

1. Например, настройка двух переменных для хранения дат начала и окончания.
2. `$SSDate = Get-Date -Day 13 -Month 04 -Year 2010`
3. `$EDate = Get-Date -Day 14 -Month 04 -Year 2010` Если требуется найти событие, произошедшее около 15 минут назад, с помощью переменной можно задать интервал между следующими моментами времени: за 20 минут до текущего времени и 10 минут спустя. `$SSDate = (Get-Date).AddMinutes(-20)`
`$EDate = (Get-Date).AddMinutes(-10)`
4. Получение списка событий между двумя заданными датами и моментами времени. `Get-SPLogEvent -StartDate $SSDate -EndTime $Edate` Если параметр `-EndTime` не указан, отображаются журналы событий вплоть до текущих значений даты и времени.

С каждым запросом к SharePoint связан свой номер для отслеживания (идентификатор корреляции). Этот номер хранится в виде значения в свойстве, связанном с объектом `SPLogEvent`. В трассировках приложения SQL Profiler также отображаются идентификаторы корреляции. Этот номер отображается на странице ошибки.

В крупных или активно используемых установках SharePoint при возникновении проблем может создаваться большое число записей журналов. С помощью идентификатора корреляции можно уменьшить объем

возвращаемых сведений в журналах, ограничив их только записями, относящимися к конкретному случаю. Чтобы задать временные рамки проблемы, следует использовать переменные \$sdate и \$edate. Если вам не удастся воспроизвести условия возникновения проблемы, но это возможно на стороне пользователя, обратитесь в службу поддержки и запросите снимок экрана страницы ошибки, чтобы проверить правильность полученного идентификатора корреляции. После этого можно передать результаты команды (показаны на шаге 2) в командлет Where-Object, как показано в следующем примере. PS C:\Users\Peter>Get-SPLogEvent -StartDate \$SDate -EndTime \$EDate | where {\$_.Correlation -eq "68bc5cf4-5a8c-4517-a879-86e35e57c862" }

В выходных данных отображаются только журналы событий в диапазоне между заданными датами, которые относятся к проблеме, связанной с этим идентификатором корреляции.

Управление службами SharePoint

После установки решения SharePoint на серверы с помощью мастера настройки продуктов SharePoint 2010 можно использовать веб-сайт центра администрирования SharePoint 2010 для запуска мастера настройки фермы. С помощью этого мастера можно достаточно быстро настроить ферму и ввести ее в эксплуатацию, однако в некоторых случаях возможности этого средства не удовлетворяют требованиям организации. Например, создаваемые мастером имена будут иметь формат имя_GUID, где GUID — это генерируемый случайным образом глобальный уникальный идентификатор. Также мастер настройки фермы использует учетную запись, указанную как удостоверение пула приложений для установленного по умолчанию веб-приложения контента, и по умолчанию запускает службу веб-аналитики, что может быть нежелательно в рабочей ферме. Мастер обычно используется для быстрого создания прототипа, а также, возможно, при использовании SharePoint в средах разработки. Однако, если любая из этих сред ограничена по вычислительной мощности ЦП и объему доступной памяти, применение мастера не рекомендуется. Например, служба веб-аналитики потребляет большой объем памяти ЦП, в результате чего разработчики в большинстве случаев запрашивают ее остановку. Также может потребоваться регулярное включение или остановка других служб, особенно если компьютеры разработчиков ежедневно используются для подключения к рабочим столам, чтения электронной почты, написанию документации либо для разработки или создания прототипов в среде SharePoint. Для управления конфигурацией можно использовать такие средства, как веб-сайт центра администрирования. Тем не менее, в некоторых случаях можно попросту забыть выполнить все задачи необходимые для полного изменения конфигурации. В таких

ситуациях рекомендуется использовать скрипты Windows PowerShell, которые после разработки проверяются на наличие всех необходимых задач и в дальнейшем позволяют существенно упростить процесс управления конфигурацией и уменьшить число связанных с ним ошибок. Подобные скрипты следует хранить в централизованном месте, что позволит использовать их другим администраторам в своих средах. Также эти скрипты могут быть полезны разработчикам. В качестве примеров задач, автоматизируемых с помощью таких скриптов, можно привести следующие.

- Быстрое удаление заданного веб-приложения: `Get-SPWebApplication http://teams | Remove-SPWebApplication -Confirm`
- Удаление веб-приложения, веб-сайта IIS и всех связанных баз данных: `Remove-SPWebApplication http://teams -Confirm -DeleteIISSite ' - RemoveContentDatabases`

Если мастер настройки фермы не использовался, в некоторых случаях вам могут потребоваться такие службы, о которых вы ранее никогда бы не задумались. Например, если службы приложений не запущены, при попытке использовать один из рабочих процессов SharePoint 2010 не будет отображаться диаграмма Microsoft Visio, на которой показывается ход выполнения процесса. В такой ситуации для устранения проблемы уже потребуется не только создание службы приложения Visio. В этом случае присутствует ряд других зависимостей, в том числе и от приложения-службы состояний. Службы состояний можно создавать только на сайте центра администрирования с помощью мастера фермы, что для вас, как администратора, крайне нежелательно. Для создания службы состояний с помощью Windows PowerShell введите следующую команду. PS:

```
C:\Users\Peter>New-SPStateServiceDatabase -Name Contoso_StateService_DB |  
>>New-SPStateServiceApplication -Name Contoso_StateService | >>New-  
SPStateServiceApplicationProxy -Name Contoso_StateService_Proxy - Default >>
```

Если экземпляр связанной службы не запущен, эта команда может не работать в вашей среде. Полный вариант скрипта см. в статье, посвященной созданию приложения-службы с помощью SP и PS 2010

<https://sharepoint.microsoft.com/blogs/zach/Lists/Posts/Post.aspx?ID=50>
(Возможно, на английском языке).

Каждое приложение-служба по своему уникально. Например, некоторые приложения-службы не используют базу данных. Дополнительные сведения об этом можно найти в главе 7, посвященной масштабированию фермы SharePoint. Тем не менее, каждая служба использует прокси приложения-службы для подключения веб-приложения к приложению-службе.

Использование Windows PowerShell в процессе обновления

Подготовка к обновлению

После установки SharePoint 2010, но до начала обновления, можно воспользоваться командлетом Test-SPContentDatabase для проверки баз данных SharePoint Server 2007 на наличие существующих или потенциальных проблем, в том числе потерянных данных, а также отсутствующих определений сайтов, компонентов или сборок. Этот командлет дополняет средство проверки перед обновлением, описываемое в главе 22. Проверка на наличие таких проблем выполняется без прерывания установки SharePoint. Этот командлет также позволяет проверить состояние баз данных SharePoint 2010 для более эффективного обслуживания и поддержания работоспособности среды. В следующем примере показана команда и образец ее выходных данных.

```
Test-SPContentDatabase -name W_intranet -WebApplication  
http://www.contoso.msft
```

Category : SiteOrphan

Error : True

UpgradeBlocking : False

Message : Database [W_intranet] contains a site (Id = [46ad6d70-9a5c4d e0-8daa-0f73f2466a6a], Url = [/]) whose id is already associated with a different database (Id = [6987d2d8-6291-4ead9eb0- aefe7097a58e], name = [W_Intranet]) in the site map. Consider deleting one of these sites which have conflicting ids.

Remedy : The orphaned sites could cause upgrade failures. Try detach and reattach the database which contains the orphaned sites. Restart upgrade if necessary.

Category : SiteOrphan

Error : True

UpgradeBlocking : False

Message : Database [W_intranet] contains a site (Id = [46ad6d70-9a5c4d e0-8daa-0f73f2466a6a], Url = [/]) whose url is already used by a different site, in database (Id = [6063e77c-991f-4c4bb3a c-68cb62e66502], name = [w_Internet]), in the same web application. Consider deleting one of the sites which have conflicting urls.

Remedy : The orphaned sites could cause upgrade failures. Try detach and reattach the database which contains the orphaned sites. Restart upgrade if necessary.

Выполнение задач после обновления

Процесс обновления подразделяется на два компонента: обновление базы данных контента и визуальное обновление. За счет разделения этих компонентов группа SharePoint может выполнить обновление веб-приложения Microsoft Office SharePoint Server 2007 незаметно для пользователей, сохраняя при этом существующий интерфейс сайтов. С помощью браузера пользователи могут просмотреть внешний вид сайта в новом оформлении SharePoint 2010, после чего принять решение о необходимости постоянного перехода на новый интерфейс посредством визуального обновления. После окончательного перехода возврат к интерфейсу SharePoint Server 2007 из браузера невозможен. В командлете Set-SPWeb отсутствуют параметры для выполнения этой задачи. Для этого следует использовать свойства объекта SPWeb. Чтобы вернуться к старому интерфейсу, введите следующую команду. Start-SPAssignment -Global \$web = Get-SPWeb http://teams ; \$web.UIVersion = 3; \$web.UIVersionConfigurationEnabled = \$true; \$web.Update(); Stop-SPAssignment

Создание переменных для хранения значений и объектов. Перед именами переменных указывается префикс \$. В показанном здесь примере кода используется переменная \$true. Эта переменная, а также другие схожие ключевые слова представляют собой пример специальной переменной. Специальная переменная обрабатывается как зарезервированное слово, которое нельзя использовать для хранения значений или объектов в скриптах. Специальная переменная \$true представляет значение True, \$false — значение False, \$null — значение null, а переменная \$_, которая также используется в этой главе, содержит текущий объект конвейера. Она используется в блоках скриптов, фильтрах и командлете Where. После выполнения этих команд для сайтов восстанавливается интерфейс SharePoint Server 2007. Например, кнопка "Действия сайта" будет располагаться в верхнем правом, а не в верхнем левом углу. В меню "Действия сайта" будет доступен параметр "Визуальное обновление" (тем не менее, визуальное обновление может быть отключено на уровне семейства сайтов). Чтобы включить визуальное обновление на уровне семейства сайтов, используйте следующие команды. \$site = Get-SPSite http://teams \$site.UIVersionConfigurationEnabled = \$true

При обновлении свойств объекта SPSite не используется метод обновления — все изменения вступают в силу незамедлительно. В качестве альтернативы двум предыдущим строкам команд можно ввести одну команду. (Get-SPSite http://teams).UIVersionConfigurationEnabled = \$false

Чтобы восстановить интерфейс SharePoint Server 2007 для всех сайтов группы в семействе, воспользуйтесь следующей командой.

```
PS C:\Users\zzspfarm> Get-SPsite http://teams |  
>> Get-SPWeb -Filter {$_.Template -eq "STS#0"}|
```

```
>> ForEach-Object {  
>> $_.UIVersion =3;  
>> $_.UIVersionConfigurationEnabled = $false;  
>> $_.Update();  
>> }  
>><ENTER>
```