

# Возможности расширения жизненного цикла

28 декабря 2020 г.

vRealize Automation 7.6

Актуальная техническая документация доступна на веб-сайте VMware:

<https://docs.vmware.com/ru/>

**VMware, Inc.**  
3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

**VMware Россия**  
Россия, 125284, г. Москва  
ул. Беговая, д.3, стр.1  
Бизнес-центр "NORDSTAR TOWER" 30й этаж  
Телефон: +7 495 212 29 00  
[www.vmware.com/ru](http://www.vmware.com/ru)

# Содержание

<b>1</b>	<b>Возможности расширения жизненного цикла</b>	<b>5</b>
	Улучшенная расширяемость	5
	Обзор расширяемости компьютера	5
	Возможности расширения жизненного цикла компьютера	6
	Выбор сценария расширения жизненного цикла	7
	Расширение жизненных циклов компьютеров за счет использования vRealize Orchestrator	8
	Расширение жизненных циклов компьютеров за счет использования контрольного списка vRealize Orchestrator	8
	Настройка подключаемого модуля vRealize Automation для расширяемости компьютера	9
	Настройка рабочих процессов инфраструктуры как услуги с помощью vRealize Orchestrator	14
	Настройка подписок на рабочие процессы для расширения возможностей vRealize Automation	15
	Темы событий, предоставляемые с помощью vRealize Automation	15
	Терминология брокера событий и подписок на рабочие процессы	17
	Темы событий с возможностью блокировки и ответа	18
	Практические рекомендации по созданию рабочих процессов vRealize Orchestrator для подписок на них	19
	Настройки подписки на рабочий процесс	20
	Работа с подписками на рабочие процессы подготовки и жизненного цикла	26
	Работа с подписками на рабочие процессы подтверждения	46
	Устранение неполадок подписок на рабочие процессы	53
	Расширение жизненных циклов компьютеров за счет использования vRealize Automation Designer	55
	Расширение жизненных циклов компьютеров за счет использования контрольного списка vRealize Automation Designer	56
	Установка и настройка vRealize Automation Designer;	57
	Настройка рабочих процессов инфраструктуры как услуги с помощью vRealize Automation Designer	61
	Рабочие процессы и распределенное управление	78
	Связывание рабочих процессов и процессов DEM с помощью навыков	78
	Удаление связей между навыками и рабочими процессами DEM	79
	Удаление связей между навыками и рабочими процессами	79
	Удаление навыка	80
	Ссылка на команду CloudUtil	80
	Команды DEM	80
	Команды файла	82
	Команды операций	85
	Команды для навыков	87
	Команды рабочих процессов	88
	Команды импорта	90

Справка по действиям рабочих процессов vRealize Automation	92
DynamicOps.Repository.Activities	93
DynamicOps.Cdk.Activities	96

# Возможности расширения жизненного цикла

1

Использование vRealize Orchestrator с vRealize Automation позволяет расширить возможности управления жизненным циклом компьютеров Инфраструктура как услуга.

Для расширения возможностей vRealize Automation требуется использование предоставленных рабочих процессов vRealize Orchestrator и создание настраиваемых рабочих процессов.

В эту главу входят следующие разделы:

- [Улучшенная расширяемость](#)
- [Обзор расширяемости компьютера](#)
- [Расширение жизненных циклов компьютеров за счет использования vRealize Orchestrator](#)
- [Настройка подписок на рабочие процессы для расширения возможностей vRealize Automation](#)
- [Расширение жизненных циклов компьютеров за счет использования vRealize Automation Designer](#)
- [Рабочие процессы и распределенное управление](#)
- [Ссылка на команду CloudUtil](#)
- [Справка по действиям рабочих процессов vRealize Automation](#)

## Улучшенная расширяемость

Настоящее руководство *Увеличение жизненного цикла* обновляется с каждым выпуском продукта или по мере необходимости.

В следующей таблице приведена история обновлений руководства *Увеличение жизненного цикла*.

Редакция	Описание
4 октября 2018 г.	Незначительные изменения.
20 сентября 2018 г.	Первоначальная редакция.

## Обзор расширяемости компьютера

Подготовка нового компьютера или вывод его из эксплуатации, особенно если дело касается критически важных систем, обычно связано со взаимодействием с разными системами управления, включая DNS-

серверы, подсистемы балансировки нагрузки, базы данных CMDB, системы управления IP-адресами и другие системами.

## Возможности расширения жизненного цикла компьютера

Можно применять настраиваемую логику на разных предварительно определенных этапах жизненного цикла инфраструктуры как услуги (IaaS) за счет использования рабочих процессов изменения состояния IaaS, известных как заглушки рабочих процессов. Можно использовать заглушки рабочих процессов, чтобы вызвать vRealize Orchestrator для двусторонней интеграции с внешними системами управления.

Создание рабочего процесса изменения состояния позволяет запустить выполнение рабочего процесса до того, как основной рабочий процесс инфраструктуры как услуги перейдет в определенное состояние.

Например, можно создать настраиваемые рабочие процессы для интеграции с внешней базой данных и записи информации на различных этапах жизненного цикла компьютера.

- Создание настраиваемого рабочего процесса, который выполняется до того, как основной рабочий процесс перейдет в состояние **MachineProvisioned**, для записи такой информации, как владелец компьютера, утверждающие и т. д.
- Создание настраиваемого рабочего процесса, который запускается до выбора на компьютере состояния **MachineDisposing** для записи времени его удаления и таких данных, как использование его ресурсов при последнем сборе данных, входе в систему и т. д.

В примерах основного рабочего процесса показаны его основные состояния, где желтым выделены состояния с возможностью настройки заглушек рабочих процессов инфраструктуры как услуги. В таблице **Настраиваемые рабочие процессы изменения состояния** перечислены доступные заглушки рабочих процессов, их соответствующее место в состоянии основного рабочего процесса, а также примеры настраиваемой логики, которую можно использовать в каждом состоянии, чтобы продлить срок службы компьютера.

Рис. 1-1. Состояния основного рабочего процесса для подготовки компьютеров

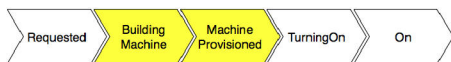


Рис. 1-2. Состояния основного рабочего процесса для импорта компьютеров

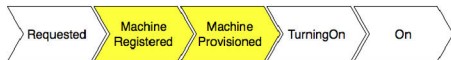


Рис. 1-3. Состояния основного рабочего процесса для завершения срока действия аренды компьютера



Рис. 1-4. Состояния основного рабочего процесса для списания компьютера

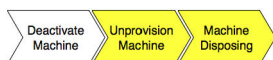


Таблица 1-1. Настраиваемые рабочие процессы изменения состояния

Состояние основного рабочего процесса	Имя настраиваемого рабочего процесса	Примеры расширяемости
BuildingMachine	WFStubBuildingMachine	Подготовка компьютера к созданию записи hypervisor.Create в базе данных управления конфигурациями (CMDB) и вызову внешней системы для назначения IP-адреса компьютеру, а затем использование параметра WFStubMachineDisposing для возврата IP-адреса в пул во время удаления компьютера.
RegisterMachine	WFStubMachineRegistered	Добавление импортированного компьютера в инструмент подготовки приложения для получения обновлений и выполнения проверки соответствия.
MachineProvisioned	WFStubMachineProvisioned	Компьютер существует в гипервизоре, и все дополнительные настройки будут выполняться на этом этапе, например, настройки гостевого агента. Используйте эту заглушку рабочего процесса для обновления IP-адреса DHCP и информации о хранении в записи базы данных управления конфигурациями. Настройки, сделанные с помощью параметра WFStubMachineProvisioned, как правило, восстанавливаются с помощью WFStubUnprovisionMachine.
Expired	WFStubMachineExpired	Перенос компьютеров с истекшим сроком действия в более дешевое хранилище для снижения затрат на архивирование, а также обновления записи CMDB и системы выставления счетов, чтобы отразить изменения, касающиеся затрат и хранения.
UnprovisionMachine	WFStubUnprovisionMachine	Удаление компьютеров из учетных записей Active Directory. Настройки, сделанные с помощью параметра WFStubMachineProvisioned, как правило, восстанавливаются с помощью WFStubUnprovisionMachine.
Disposing	WFStubMachineDisposing	Возврат IP-адресов обратно в пул.

## Выбор сценария расширения жизненного цикла

Можно использовать vRealize Orchestrator или vRealize Automation Designer, чтобы расширить жизненные циклы компьютеров.

Можно расширить жизненный цикл компьютера, используя vRealize Automation Designer для вызова в vRealize Orchestrator или непосредственно vRealize Orchestrator. Оба подхода позволяют добавлять настраиваемую логику на предварительно определенных этапах жизненного цикла компьютера инфраструктуры как услуги путем создания настраиваемых рабочих процессов vRealize Orchestrator, а затем вставлять эти процессы в заглушки рабочих процессов изменения состояния. Однако можно

ограничить настраиваемую логику изменения состояний в пределах конкретных схем элементов только при непосредственном использовании vRealize Orchestrator. Можно лишь ограничить выполнение процессов в пределах конкретных диспетчеров Distributed Execution Manager с помощью vRealize Automation Designer.

**Примечание** Заглушки рабочих процессов заменяются подписками на рабочие процессы брокера событий. Они остаются доступны, сохраняется поддержка и возможность использования, но в следующей версии vRealize Automation планируется их удаление. Чтобы обеспечить совместимость последующих продуктов, используйте подписки на рабочий процесс для запуска настраиваемых рабочих процессов с примененными изменениями. См. раздел [Настройка подписок на рабочие процессы для расширения возможностей vRealize Automation](#).

Таблица 1-2. Выбор сценария расширения жизненного цикла

Сценарий	Процедура
Добавьте настраиваемую логику в предварительно определенные этапы жизненного цикла компьютера инфраструктуры как услуги и примените эту логику к конкретным схемам элементов.	<a href="#">Расширение жизненных циклов компьютеров за счет использования контрольного списка vRealize Orchestrator</a>
Добавьте настраиваемую логику в предварительно определенные этапы жизненного цикла компьютера инфраструктуры как услуги и примените эту логику ко всем схемам элементов.	<a href="#">Расширение жизненных циклов компьютеров за счет использования контрольного списка vRealize Automation Designer</a>
Ограничьте выполнение рабочих процессов рамками определенных диспетчеров Distributed Execution Manager, используя навыки в vRealize Automation Designer. Навыки похожи на теги, которые можно применить как к рабочим процессам, так и к экземплярам процессов DEM Worker. Например, может потребоваться ограничить рабочие процессы облачной подготовки до определенного процесса DEM, запущенного на узле с требуемым сетевым доступом к URL-адресам Amazon.	<a href="#">Связывание рабочих процессов и процессов DEM с помощью навыков</a>

## Расширение жизненных циклов компьютеров за счет использования vRealize Orchestrator

Можно вставить настраиваемую логику на заранее определенных этапах жизненного цикла компьютера инфраструктуры как услуги путем создания настраиваемых рабочих процессов vRealize Orchestrator, а затем, использования vRealize Orchestrator для их вставки в жизненный цикл компьютеров, построенных из определенных схем элементов.

### Расширение жизненных циклов компьютеров за счет использования контрольного списка vRealize Orchestrator

Расширение жизненных циклов компьютера за счет использования контрольного списка vRealize Orchestrator обеспечивают общий обзор этапов, необходимых для установки и конфигурации vRealize Orchestrator, чтобы настроить жизненные циклы компьютера инфраструктуры как услуги.



Таблица 1-3. Расширение жизненных циклов компьютеров за счет использования контрольного списка vRealize Orchestrator

Задача	Сведения
<input type="checkbox"/> Настройте узел vRealize Automation для vRealize Orchestrator.	<a href="#">Добавление узла vRealize Automation</a>
<input type="checkbox"/> Настройте узел инфраструктуры как услуги для vRealize Orchestrator.	<a href="#">Добавление узла инфраструктуры как услуги</a>
<input type="checkbox"/> Установите настройки vRealize Orchestrator для расширенных жизненных циклов компьютеров инфраструктуры как услуги.	<a href="#">Установка настройки vRealize Orchestrator</a>
<input type="checkbox"/> Создать конечную точку vRealize Automation для экземпляра vRealize Orchestrator.	<a href="#">Создание конечной точки vRealize Orchestrator</a>
<input type="checkbox"/> Используйте шаблон рабочего процесса, представленный в подкаталоге «Расширяемость» библиотеки подключаемых модулей vRealize Automation, для создания настраиваемого рабочего процесса vRealize Orchestrator, который выполняется в течение всего жизненного цикла компьютера. Можно запускать несколько рабочих процессов в одном состоянии для одной и той же схемы элементов, пока они вложены в отдельный рабочий процесс оболочки.	Информацию по разработке рабочих процессов с помощью vRealize Orchestrator см. в документации vRealize Orchestrator. Чтобы обучаться в процессе разработки vRealize Orchestrator для интеграции vRealize Automation, см. учебные курсы, доступные на портале VMware Education, и учебные материалы, представленные на платформе VMware Learning.
<input type="checkbox"/> Запустите предоставленный рабочий процесс, который вставляет настраиваемый рабочий процесс в заглушку рабочего процесса инфраструктуры как услуги и конфигурирует схему элементов для вызова заглушки рабочего процесса инфраструктуры как услуги.	<a href="#">Назначение рабочего процесса изменения состояния для схемы элементов и ее виртуальных машин</a>
<b>Примечание</b> Заглушки рабочих процессов заменяются подписками на рабочие процессы брокера событий. Они остаются доступны, сохраняется поддержка и возможность использования, но в следующей версии vRealize Automation планируется их удаление. Чтобы обеспечить совместимость последующих продуктов, используйте подписки на рабочий процесс для запуска настраиваемых рабочих процессов с примененными изменениями. См. раздел <a href="#">Настройка подписок на рабочие процессы для расширения возможностей vRealize Automation</a> .	

## Настройка подключаемого модуля vRealize Automation для расширяемости компьютера

Настройте узлы инфраструктуры как услуги и vRealize Automation, установите настройки для расширяемости компьютера и создайте конечную точку vRealize Automation для экземпляра vRealize Orchestrator.

## Добавление узла vRealize Automation

Можно запустить рабочий процесс добавления узла vRealize Automation и настройки параметров его подключения.

### Процедура

1. В раскрывающемся меню клиента Orchestrator выберите пункт **Запуск** или **Проектирование**.
2. Выберите представление **Рабочие процессы**.
3. Разверните элементы **Библиотека > vRealize Automation > Конфигурация**.
4. Щелкните правой кнопкой мыши рабочий процесс **Добавить узел vRA** и укажите **Запустить рабочий процесс**.
5. В текстовом поле **Имя узла** введите уникальное имя для узла.
6. В текстовом поле **URL-адрес узла** укажите URL-адрес узла.

Например: *https://hostname*.

7. (Обязательно) В текстовом поле **Арендатор** укажите имя арендатора.  
Чтобы использовать все функции подключаемого модуля для арендатора, создайте выделенный узел vRealize Automation для каждого арендатора.
8. Укажите, нужно ли устанавливать сертификаты SSL автоматически без подтверждения пользователя.
9. (дополнительно) Чтобы настроить время, в течение которого в vRealize Orchestrator ожидается подключение или ответ от vRealize Automation, введите интервалы времени ожидания в текстовых полях **Время ожидания подключения (в секундах)** и **Время ожидания операции (в секундах)**.
10. Выберите тип подключения к узлу в раскрывающемся меню **Режим сеанса**.

Параметр	Действия
Общий сеанс	Введите учетные данные для пользователя vRealize Automation в текстовых полях <b>Имя пользователя для проверки подлинности</b> и <b>Пароль проверки подлинности</b> .
На сеанс пользователя	<p>Выполните подключение с учетными данными пользователя, который в данный момент вошел в систему. Необходимо войти в клиент Orchestrator под учетными данными системного администратора vRealize Automation.</p> <p>Чтобы использовать этот параметр с внешним сервером vRealize Orchestrator, необходимо зарегистрировать сервер Orchestrator в реестре компонентов vRealize Automation.</p> <p><b>Примечание</b> Чтобы зарегистрировать внешний сервер vRealize Orchestrator в реестре компонентов, необходимо настроить для Orchestrator использование vRealize Automation в качестве поставщика средств проверки подлинности. Дополнительные сведения см. в разделе <i>Установка и настройка VMware vRealize Orchestrator</i>.</p>

11. Нажмите кнопку **Отправить**.

## Следующие шаги

Добавьте узел администрирования инфраструктуры vRealize Automation.

## Добавление узла инфраструктуры как услуги

Можно запустить рабочий процесс, чтобы добавить узел инфраструктуры как услуги узла vRealize Automation и настроить параметры соединения.

### Процедура

1. В раскрывающемся меню клиента Orchestrator выберите пункт **Запуск** или **Проектирование**.
2. Выберите представление **Рабочие процессы**.
3. Разверните элементы **Библиотека > vRealize Automation > Администрирование инфраструктуры > Конфигурация**.
4. Щелкните правой кнопкой мыши **Добавить узел инфраструктуры как услуги** и укажите **Запустить рабочий процесс**.
5. Выберите узел vRealize Automation, для которого необходимо настроить узел инфраструктуры как услуги из раскрывающегося меню **Узел vCAC**.
6. В текстовом поле **Имя узла** введите уникальное имя для узла.
7. Введите URL-адрес компьютера, на котором установлен диспетчер моделей.  
Например, [https://model\\_manager\\_machine.com](https://model_manager_machine.com).
8. Чтобы установить сертификаты SSL, нажмите **Да**.
9. Чтобы использовать прокси для доступа к компьютеру диспетчера моделей, нажмите **Да**.  
При выборе этого параметра необходимо предоставить прокси-сервер и его порт на следующей странице.
10. Нажмите кнопку **Далее**.
11. При настройке явного прокси предоставьте URL-адрес прокси-сервера и порт.
12. Нажмите кнопку **Далее**.
13. Чтобы настроить собственные значения времени ожидания, нажмите **Нет**.
14. (дополнительно) Чтобы настроить время, в течение которого в vRealize Orchestrator ожидается подключение или ответ от vRealize Automation, введите интервалы времени ожидания в текстовых полях **Время ожидания подключения (в секундах)** и **Время ожидания операции (в секундах)**.
15. Нажмите кнопку **Далее**.

**16. Выберите тип проверки подлинности узла.**

Параметр	Описание
Единый вход	Выберите этот параметр, чтобы использовать службу vCenter Single Sign-On.
NTLM	<p>Выберите этот параметр, чтобы включить функцию проверки подлинности только на основе протокола NT LAN Manager (NTLM), если инфраструктура Active Directory использует проверку подлинности NTLM.</p> <p>При выборе этого параметра необходимы дополнительные учетные данные NTLM и параметры проверки подлинности.</p>

**17.** При выборе NTLM нажмите кнопку **Далее** и введите имя компьютера рабочей станции, а также имя домена NetBIOS.

**18.** Нажмите кнопку **Отправить**.

## Установка настройки vRealize Orchestrator

Можно запустить рабочий процесс, чтобы установить заглушки настроенного рабочего процесса изменения состояния и рабочие процессы операций меню Orchestrator.

**Примечание** Заглушки рабочих процессов заменяются подписками на рабочие процессы брокера событий. Они остаются доступны, сохраняется поддержка и возможность использования, но в следующей версии vRealize Automation планируется их удаление. Чтобы обеспечить совместимость последующих продуктов, используйте подписки на рабочий процесс для запуска настраиваемых рабочих процессов с примененными изменениями. См. раздел [Настройка подписок на рабочие процессы для расширения возможностей vRealize Automation](#).

### Процедура

1. В раскрывающемся меню клиента Orchestrator выберите пункт **Запуск** или **Проектирование**.
2. Выберите представление **Рабочие процессы**.
3. Выберите **Библиотека > vCloud Automation Center > Администрирование инфраструктуры > Расширяемость > Установка**.
4. Щелкните правой кнопкой мыши рабочий процесс **Установить настройку vCO** и укажите **Запустить рабочий процесс**.
5. Выберите узел инфраструктуры как услуги.
6. Нажмите кнопку **Далее**.
7. Выберите этапы жизненного цикла, к которым необходимо добавить настраиваемую логику, выбрав одну или несколько заглушек рабочего процесса изменения состояния для установки.
8. Нажмите кнопку **Отправить**.

## Создание конечной точки vRealize Orchestrator

Можно создать конечную точку vRealize Orchestrator для подключения к серверу vRealize Orchestrator.

Вы можете настроить несколько конечных точек, чтобы подключиться к разным серверам vRealize Orchestrator, однако для каждой конечной точки нужно задать приоритет.

При выполнении рабочих процессов vRealize Orchestrator vRealize Automation сначала использует конечную точку vRealize Orchestrator с наивысшим приоритетом. Если к ней невозможно подключиться, то устройство переходит к следующей конечной точке с наивысшим приоритетом, пока сервер vRealize Orchestrator не сможет выполнить рабочий процесс.

#### Необходимые условия

- Войдите в службу vRealize Automation в качестве **администратора инфраструктуры как услуги**.

#### Процедура

1. Выберите **Инфраструктура > Конечные точки > Конечные точки**.
2. Выберите **Создать > Оркестрация > vRealize Orchestrator**.
3. Введите имя и, при необходимости, описание.
4. Введите URL-адрес с полным квалифицированным именем или IP-адрес сервера vRealize Orchestrator и номер порта vRealize Orchestrator.

Следует использовать транспортный протокол HTTPS. Если порт не указан, используется порт по умолчанию 443.

Чтобы использовать экземпляр vRealize Orchestrator по умолчанию, который содержится в устройстве vRealize Automation, введите

**`https://vrealize-automation-appliance-hostname:443/vco`**.

5. Укажите свои учетные данные vRealize Orchestrator в текстовых полях **Имя пользователя** и **Пароль**, чтобы подключиться к конечной точке vRealize Orchestrator.

Используемые учетные данные должны иметь разрешения на выполнение любых рабочих процессов vRealize Orchestrator, которые можно вызвать из Инфраструктура как услуга

Чтобы использовать экземпляр vRealize Orchestrator по умолчанию, содержащийся в устройстве vRealize Automation, используйте имя пользователя **`administrator@vsphere.local`** и пароль администратора, указанный при настройке службы единого входа.

6. В текстовом поле **Свойство** введите целое число не меньше 1.  
Чем меньше значение, тем выше приоритет.
7. (дополнительно) Выберите элемент **Свойства** и добавьте стандартные настраиваемые свойства, группы свойств или собственные определения свойств для конечной точки.
8. Нажмите кнопку **ОК**.

## Настройка рабочих процессов инфраструктуры как услуги с помощью vRealize Orchestrator

Используйте один рабочий процесс в vRealize Orchestrator, чтобы вставить настраиваемую логику в заглушки рабочих процессов инфраструктуры как услуги и назначить настроенные жизненные циклы для схем элементов компьютера.

---

**Примечание** Заглушки рабочих процессов заменяются подписками на рабочие процессы брокера событий. Они остаются доступны, сохраняется поддержка и возможность использования, но в следующей версии vRealize Automation планируется их удаление. Чтобы обеспечить совместимость последующих продуктов, используйте подписки на рабочий процесс для запуска настраиваемых рабочих процессов с примененными изменениями. См. раздел [Настройка подписок на рабочие процессы для расширения возможностей vRealize Automation](#).

---

Необходимо разработать настраиваемые рабочие процессы vRealize Orchestrator, чтобы принять строковые входные значения. Если в настраиваемом рабочем процессе ожидается сложный тип данных, создайте рабочий процесс оболочки, который найдет это сложное значение и переведет его в строку. В качестве примера рабочего процесса оболочки см. шаблон рабочего процесса, предоставленный в **Библиотека > vRealize Automation > Инфраструктура > Расширяемость**.

### Назначение рабочего процесса изменения состояния для схемы элементов и ее виртуальных машин

Выполните настройку рабочих процессов vRealize Orchestrator для запуска на определенных этапах в рабочем процессе основного компьютера, связав настраиваемый рабочий процесс с заглушкой рабочего процесса изменения состояния и назначив рабочие процессы для схемы элементов.

---

**Примечание** Заглушки рабочих процессов заменяются подписками на рабочие процессы брокера событий. Они остаются доступны, сохраняется поддержка и возможность использования, но в следующей версии vRealize Automation планируется их удаление. Чтобы обеспечить совместимость последующих продуктов, используйте подписки на рабочий процесс для запуска настраиваемых рабочих процессов с примененными изменениями. См. раздел [Настройка подписок на рабочие процессы для расширения возможностей vRealize Automation](#).

---

#### Необходимые условия

Используйте шаблон рабочего процесса, представленный в подкаталоге «Расширяемость» библиотеки подключаемых модулей vRealize Automation для создания настраиваемого рабочего процесса для запуска в течение всего жизненного цикла компьютера.

#### Процедура

1. В раскрывающемся меню клиента Orchestrator выберите пункт **Запуск** или **Проектирование**.
2. Выберите представление **Рабочие процессы**.
3. Выберите **Библиотека > vRealize Automation > Инфраструктура > Расширяемость**.

4. Щелкните правой кнопкой мыши рабочий процесс **Назначить рабочий процесс изменения состояния для схемы элементов и ее виртуальных машинах** и выберите **Запустить рабочий процесс**.
5. Выберите этап жизненного цикла, на котором необходимо запустить рабочий процесс, выбрав заглушку в раскрывающемся меню **Заглушки рабочего процесса vCAC для включения**.
6. Выберите узел инфраструктуры как услуги.
7. Нажмите кнопку **Далее**.
8. Выберите схему элементов, для которой необходимо назначить рабочий процесс.
9. Выберите, следует ли применять эти рабочие процессы для существующих компьютеров, подготовленных с помощью этой схемы элементов.
10. Выберите рабочий процесс, который необходимо запускать в течение всего жизненного цикла компьютера.
11. Настройте входные значения рабочего процесса, которые добавляются в качестве настраиваемых свойств компьютера.
  - а) Добавьте выходные значения рабочего процесса vCO в качестве свойств схемы элементов .
  - б) Добавьте последние входные значения запуска рабочего процесса vCO в качестве свойств схемы элементов.
12. Нажмите кнопку **Отправить**.

## Настройка подписок на рабочие процессы для расширения возможностей vRealize Automation

Создайте подписки на рабочие процессы, которые используют службу брокера событий, чтобы выполнять мониторинг зарегистрированных служб для сообщений о событиях в vRealize Automation, а затем запустите указанный рабочий процесс vRealize Orchestrator, когда выполняются условия в подписке. Чтобы настроить подписку, укажите тему события, инициирующие условия и рабочий процесс, который выполняется при запуске.

Администраторы арендатора могут создавать специальные подписки на рабочие процессы для арендатора и управлять ими.

Системный администратор может создавать подписки на рабочие процессы и управлять ими. Созданные подписки на рабочие процессы системы являются активными для любых событий в арендаторе, а также системных событий.

## Темы событий, предоставляемые с помощью vRealize Automation

Темы событий описывают тип сообщения о событии, которое отправляется в службу брокера событий другими службами. Выберите тему событий и настройте подписку на рабочий процесс в соответствии с темой.

Таблица 1-4. Темы событий

Имя темы события	Описание	Служба
Компонент схемы элементов выполнен	Подготовка компонента схемы элементов, который является частью составной схемы элементов, завершена. Компонент — это любая схема элементов, которая является частью составной схемы элементов.	composition-service
Компонент схемы элементов запрошен	Запрошен компонент схемы элементов, который является частью составной схемы элементов. Компонент — это любая схема элементов, которая является частью составной схемы элементов.	composition-service
Конфигурация схемы элементов	Создание, обновление или удаление схемы элементов	composition-service
Запрос схемы элементов выполнен	Подготовка составной схемы элементов завершена. Эта тема события включает все компоненты схемы элементов. В нее не входят самостоятельные схемы элементов Все как услуга.	composition-service
Запрошенная схема элементов	Запрошена составная схема элементов. Эта тема события не включает схемы элементов Все как услуга.	composition-service
Конфигурация бизнес-группы	Создание, обновление или удаление бизнес-группы	identity
Действие компонента выполнено	Действие запущено для компонента развернутой схемы элементов при запросе действия развертывания.	composition-service
Действие компонента запрошено	Действие, которое следует запустить для компонента развернутой схемы элементов, запрошено при запросе действия развертывания.	composition-service
Действие развертывания выполнено	Завершено выполнение действия для развернутой схемы элементов, в том числе выполнение всех действий компонентов.	composition-service
Действие развертывания запрошено	Запрошено действие для развернутой схемы элементов.	composition-service
Стандартное событие EventLog	В журнал событий добавляется стандартная запись. Запись журнала не отправляется подписчикам.	eventlog-service
Завершение события жизненного цикла IP-адресов для управления IP-адресами	Выполнен запрос на выделение или отмену выделения IP-адреса.	ipam-service
Жизненный цикл компьютера	Предоставленное действие Инфраструктура как услуга запускается на подготовленном компьютере.	iaas-service



Таблица 1-4. Темы событий (продолжение)

Имя темы события	Описание	Служба
подготовка компьютера.	Компьютер Инфраструктура как услуга находится в процессе подготовки.	iaas-service
Конфигурация сервера оркестрации	vRealize OrchestratorМожно создать, обновить, удалить или изменить конфигурацию сервера, чтобы использовать другой экземпляр по умолчанию.	o11n-gateway-service
Конфигурация сервера оркестрации (все как услуга) — устаревшая	vRealize OrchestratorМожно создать, обновить, удалить или изменить конфигурацию сервера, чтобы использовать другой экземпляр по умолчанию.	advanced-designer-service
Последующее подтверждение	Уровень политики последующего подтверждения настроен для использования параметра подписки на событие.	approval-service
Предварительное подтверждение	Уровень политики предварительного подтверждения настроен для использования параметра подписки на событие.	approval-service
Событие завершения реорганизации ресурса	Срок действия договора аренды ресурсов истек, и ресурсы реорганизованы.	management-service

## Терминология брокера событий и подписок на рабочие процессы

При работе с подписками на рабочие процессы и службой брокера событий вы можете столкнуться со специфичной терминологией.

Таблица 1-5. Терминология брокера событий и подписки на рабочие процессы

Временная лицензия	Описание
Тема события	Описывает набор событий с одинаковой логической целью и структурой. Каждое событие является экземпляром темы события.
Событие	Обозначает изменение состояния в авторе или сущностях под его управлением. Событие является сущностью, записывающей информацию о наступлении события.
Сообщение	Передаёт информацию о событии между различными службами и компонентами. Например, от автора в службу брокера события или из службы брокера события к подписчикам.
Служба брокера события	Служба, подготавливающая к отправке сообщения, которые автор опубликовал для подписчиков.
Полезная нагрузка	Данные о событии.

Таблица 1-5. Терминология брокера событий и подписки на рабочие процессы (продолжение)

Временная лицензия	Описание
Подписка	Показывает, что подписчик заинтересован в получении уведомлений о событии, подписавшись на тему события и указав критерии отправки уведомления.
Подписчик	Использует события, опубликованные в службе брокера событий, в рамках описания подписки. Подписчик также может называться потребителем.
Поставщик	Регистрирует темы события в службе брокера событий
Автор	Публикует события в службе брокера событий.
Системный администратор	Пользователь с привилегиями создания, чтения, обновления и удаления подписки на рабочие процессы арендатора и системы, использующие API или подключаемый модуль vRealize Automation. vRealize Automation не включает в себя пользовательский интерфейс системного администратора.
Администратор арендатора	Пользователь с привилегиями создания, чтения, обновления и удаления подписок арендатора на его рабочие процессы.
Подписка на рабочий процесс	Определяет тему и условия события для запуска рабочего процесса vRealize Orchestrator.
Подписка на рабочий процесс системы	Подписка на специализированный рабочий процесс, который реагирует на системные события и события, происходящие во всех арендаторах.
Подписка на рабочий процесс арендатора	Подписка на специализированный рабочий процесс, который определяет условия запуска рабочего процесса vRealize Orchestrator для событий в одном арендаторе.

## Темы событий с возможностью блокировки и ответа

Темы событий могут поддерживать события с возможностью блокировки и ответа. Поведение подписки на рабочий процесс зависит от поддержки этих типов событий в теме и способа настройки самой подписки.

### Темы событий без возможности блокировки

Темы событий без возможности блокировки позволяют создавать только неблокирующие подписки. Неблокирующие подписки запускаются асинхронно, поэтому не следует полагаться на порядок их запуска. Однако запускающее событие гарантированно происходит, и запускается связанный с подпиской рабочий процесс vRealize Orchestrator. Неблокирующие подписки возвращают ответ только для темы с соответствующей возможностью.

### Темы событий с возможностью блокировки

Для тем некоторых событий поддерживается блокировка. Если подписка на рабочий процесс отмечена как блокирующая, то другие подписки с соответствующими условиями не получают сообщения, отвечающие настроенным условиям, пока первый рабочий процесс не будет завершен. Если есть несколько блокирующих подписок на рабочий процесс с одной темой события, необходимо установить приоритет подписок.

Блокирующие подписки запускаются в порядке приоритета. Высший приоритет имеет значение 0. При наличии нескольких блокирующих подписок с одной темой события и одинаковым уровнем приоритета подписки запускаются в алфавитном порядке по именам. После обработки всех блокирующих подписок отправляется сообщение всем неблокирующим подпискам одновременно. Поскольку блокирующие подписки на рабочий процесс запускаются синхронно, в измененную полезную нагрузку события входит обновленное событие при поступлении уведомления о последующих подписках на рабочий процесс.

Применяйте блокировку для одной или нескольких подписок на рабочий процесс в зависимости от выбранного рабочего процесса и целей.

Например, есть подписки на два рабочих процесса подготовки, где второй процесс зависит от результатов первого. В первом из них изменяется свойство во время подготовки, а второй записывает новое свойство, например имя виртуальной машины, в файловую систему. Приоритет подписки на процесс `ChangeProperty` равен 0, а на процесс `RecordProperty` — 1, поскольку он использует результаты подписки `ChangeProperty`. После подготовки виртуальной машины запускается подписка на процесс `ChangeProperty`. Поскольку условия подписки на процесс `RecordProperty` основаны на условии, сформированном после подготовки, сообщение инициирует подписку на процесс `RecordProperty`. Однако поскольку рабочий процесс `ChangeProperty` является блокирующим, до его окончания сообщение не будет доставлено. После изменения имени и завершения первого рабочего процесса запускается второй рабочий процесс, чье имя записывается в файловую систему.

Даже если тема событий поддерживает блокировку, можно создать неблокирующую подписку на рабочий процесс в случае, когда подписка на рабочий процесс не содержит зависимых последующих рабочих процессов. Подписка на рабочий процесс инициирует и запускает рабочий процесс `vRealize Orchestrator` без дальнейшего обращения из `vRealize Automation` или внешней системы.

## Темы событий с возможностью ответа

Некоторые темы событий поддерживают ответы службы с подпиской. Служба, в которой зарегистрирована тема события с возможностью ответа, может принять событие ответа, предоставляющее выходное значение рабочего процесса, как правило, в результате обращения к системе или пользователю. Выходные параметры ответа должны соответствовать критериям, определенным в схеме ответа, чтобы служба `vRealize Automation`, опубликовавшая исходное событие с возможностью ответа, могла обработать его. Например, в подписках на предварительное или последующее подтверждение есть возможность ответа. Если создать рабочий процесс, который отправляет запрос на подтверждение во внешнюю систему, то `vRealize Automation` обработает подтвержденный или отклоненный ответ, будет подготовлен элемент каталога или пользователь получит уведомление об отклонении запроса.

Ответ может быть выходным значением рабочего процесса `vRealize Orchestrator` или значением ошибки, если истекло время рабочего процесса или в нем произошел сбой. Если получен ответ от выходных параметров рабочего процесса, он должен быть в правильном формате схемы ответа.

## Практические рекомендации по созданию рабочих процессов vRealize Orchestrator для подписок на них

Подписка на рабочий процесс основана на конкретной схеме темы. Чтобы убедиться, что подписки могут запускать рабочие процессы `vRealize Orchestrator`, необходимо настроить их с правильными входными параметрами для работы с данными о событии.

## Входные параметры рабочего процесса

Создаваемый настраиваемый рабочий процесс может содержать все параметры или только один, который использует все данные полезной нагрузки.

- Чтобы включить отдельные параметры, настройте один или несколько параметров. Убедитесь, что имя и тип совпадают с именем и типом, указанными в схеме. Сложные типы из схемы должны быть определены в рабочем процессе как «Свойства».
- Чтобы использовать отдельный параметр, настройте один параметр с типом **Properties**. Можно предоставить любое подходящее имя. Например, можно использовать `payload` в качестве имени параметра.

## Выходные параметры рабочего процесса

Создаваемый настраиваемый рабочий процесс может содержать выходные параметры, соответствующие последующим событиям, необходимым для типа темы события ответа.

Если в теме события ожидается ответ, выходные параметры рабочего процесса должны соответствовать схеме ответа.

## Настройки подписки на рабочий процесс

Параметры подписки определяют, когда запускается рабочий процесс на основе сообщений о событиях в vRealize Automation. Используйте параметры, чтобы управлять подписками.

Подписка представляет собой намерение пользователя подписаться на события данной темы и запустить рабочий процесс при получении события по этой теме, в котором указаны определенные условия.

Только администратор арендатора может создавать подписку на рабочий процесс. Все подписки на рабочие процессы являются специфическими для арендатора.

Чтобы управлять подписками на рабочие процессы, выберите **Администрирование > События > Подписки**.

Таблица 1-6. Параметры подписки на рабочий процесс

Параметр	Описание
Создать	Создание новой подписки.
Изменить	Изменение выбранной подписки. Если подписка опубликована, сохраненные изменения сразу же становятся активными. Нельзя редактировать тему событий или изменять параметр блокировки в опубликованной или неопубликованной подписке.
Опубликовать	Активация подписки. Обрабатываются события службы брокера событий, и оцениваются условия подписки. Если настроенное условие истинно, то запускается рабочий процесс.

Таблица 1-6. Параметры подписки на рабочий процесс (продолжение)

Параметр	Описание
Отменить публикацию	Возврат подписки в состояние черновика. Подписка перестает быть активной в среде и больше не получает события. При повторной публикации подписки она начинает получать новые события. Прошедшие события не доставляются.
Удалить	Удаление выбранной подписки.

## Назначение темы события для подписки

Темы событий являются классами событий, представленных в vRealize Automation. Выберите тему события для определения подписки.

Темы событий — это категории, которые группируют сходные события. Когда тема события назначается подписке, это определяет, какое событие будет запускать эту подписку.

### Процедура

1. Нажмите **Администрирование > События > Подписки**
2. Нажмите **Создать** и выберите тему события.

Таблица 1-7. Сведения о теме события

Сведения о теме события	Описание
Идентификатор темы	Идентификатор темы события.
Имя	Имя темы события.
Описание	Описание темы события.
Издатель	Имя службы, для которой регистрируется тема события.
Возможность блокировки	Указывает на возможность создания блокирующей подписки для этой темы события. Блокирующие подписки используются для изменения полезных данных события или запуска настраиваемых логических схем, когда результаты второго рабочего процесса для того же события зависят от результатов первого.
Возможность ответа	Указывает, можно ли в теме события подписки опубликовать ответное событие в службе, в которой изначально было создано событие. Если значение «Да», по завершению рабочего процесса ответ отправляется в службу, опубликовавшую исходное событие. В ответе содержатся выходное значение рабочего процесса vRealize Orchestrator и сведения о любых ошибках.
Схема	Описывает структуру полезной нагрузки события. Можно использовать схему для создания рабочих процессов, которые могут использовать информацию о полезной нагрузке.

## Назначение условий рабочего процесса для подписки

Условия, настраиваемые для подписки, определяют, инициирован ли запуск рабочего процесса с учетом данных о событии.

Для рабочего процесса можно определить условия, с помощью которых можно управлять его запуском.

Если выбрана функция **Запускать при определенных условиях**, могут быть доступны следующие варианты.

- **Данные**

В сообщении о событии включается информация, относящаяся к выбранной теме события. Например, если создано условие для темы события «жизненный цикл виртуальной машины», поля данных будут связаны со схемами элементов и виртуальными машинами. Если выбрана тема события «предварительное утверждение», поля данных будут связаны с политиками утверждения.

Можно также добавить условия для полей, которые не включены в схему, указав путь в текстовом поле над деревом. Используйте формат **`${PATH}`**. PATH — путь в схеме. Разделите узлы знаком **~**. Например, **`data~machine~properties~SomeCustomProperty`**.

- **Значения сообщения об основном событии**

Сюда входят общие сведения о данном сообщении о событии. Например, тип события, метка времени или имя пользователя.

### Необходимые условия

#### Процедура

1. Нажмите **Администрирование > События > Подписки**
2. Нажмите **Создать** и выберите тему события.

### 3. Нажмите кнопку **Далее** и задайте нужные условия рабочего процесса.

Таблица 1-8. Типы условий

Условие	Описание
Запускать для всех событий	Выбранный рабочий процесс запускается при получении сообщения на эту тему события.
Запускать при определенных условиях	<p>Выбранный рабочий процесс запускается при обнаружении сообщения о событии и соответствии события настроенным условиям.</p> <p>При выборе этого параметра необходимо указать условия на основе данных о событии, которые будут запускать выбранный рабочий процесс для данной подписки.</p> <ul style="list-style-type: none"> <li>■ <b>Одно условие.</b> Рабочий процесс запускается, если настроенное предложение является истинным.</li> <li>■ <b>Все перечисленное.</b> Рабочий процесс запускается, если все предложения являются истинными и указаны по крайней мере два условия.</li> <li>■ <b>Любое из перечисленного.</b> Рабочий процесс запускается, если запрос соответствует хотя бы одному предложению и указаны по крайней мере два условия.</li> <li>■ <b>Ничего из перечисленного.</b> Рабочий процесс запускается, если запрос не соответствует всем условиям.</li> </ul> <p>Если условие создано на основе постоянного значения, это значение будет обрабатываться без учета регистра. Например, если задано условие «Имя схемы элементов содержит UNIX», но в именах схем используется слово Unix, это условие будет обрабатываться правильно.</p> <p>Чтобы привести имя условия в соответствие с именем схемы элементов, необходимо сначала изменить значение на другое, не содержащее такую же строку. Например, чтобы изменить условие UNIX, можно изменить значение на xxxx, сохранить его, а затем изменить xxxx на Unix и снова сохранить.</p>

## Назначение рабочего процесса для подписки

Выбранный рабочий процесс vRealize Orchestrator запускается, когда условия подписки оцениваются как истинные.

Рабочие процессы объединяют действия ABX, решения и результаты. При выполнении этих действий в определенном порядке в виртуальной среде выполняется определенная задача или определенный процесс. Рабочие процессы выполняют такие задачи, как подготовка виртуальных машин, резервное копирование, регулярное обслуживание, отправка электронной почты, выполнение операций по протоколу SSH, управление физической инфраструктурой, а также другие стандартные служебные операции. Рабочие процессы принимают входные данные в соответствии с их функциями. Рабочие процессы могут также вызывать другие рабочие процессы. Например, можно использовать в нескольких разных рабочих процессах рабочий процесс, который запускает виртуальную машину.

Можно связать рабочие процессы в подписке и автоматизировать процедуру, которая будет запускаться в ответ на определенное событие. Это позволяет рабочему процессу выполняться и создавать результаты без вмешательства пользователя. В частности, появляется возможность запуска на виртуальной машине рабочих процессов в ответ на события жизненного цикла подготовки. Также можно использовать выходные данные подписки для обмена данными между рабочими процессами, находящимися в одном состоянии. Рабочие процессы, зарегистрированные в одном состоянии жизненного цикла, могут объединять полезные выходные данные.

#### Необходимые условия

Рабочий процесс должен существовать в vRealize Orchestrator и присутствовать в списке **Администрирование > Конфигурация vRO > Конфигурация сервера**.

#### Процедура

1. Нажмите **Администрирование > События > Подписки**
2. Нажмите **Создать** и выберите **тему события**.
3. Нажмите кнопку **Далее** и задайте **условия рабочего процесса**.
4. Нажмите кнопку **Далее** и выберите **рабочий процесс**, который нужно назначить этой подписке.

Таблица 1-9. Вкладка «Рабочий процесс»

Вкладка «Рабочий процесс»	Описание
Выбор рабочего процесса	Переход к рабочему процессу.
Выбранный рабочий процесс	Отображает информацию о рабочем процессе, в том числе входных и выходных параметрах, чтобы можно было проверить, что именно этот процесс необходимо запустить.

#### Укажите сведения о подписке для рабочего процесса

В сведениях о подписке указывается способ ее обработки.

Для подписки можно также указать дополнительную информацию.

#### Процедура

1. Выберите **Администрирование > События > Подписки**
2. Нажмите **Создать** и выберите **тему события**.
3. Нажмите кнопку **Далее** и задайте **условия рабочего процесса**.
4. Нажмите кнопку **Далее** и выберите **рабочий процесс**, который нужно назначить этой подписке.



**5. Нажмите кнопку Далее и введите сведения о подписке для рабочего процесса.****Таблица 1-10. Сведения о рабочем процессе**

Сведения	Описание
Имя	<p>По умолчанию отображаемое имя является именем выбранного рабочего процесса.</p> <p>Имя отображается в списке подписок. Имя должно быть уникальным в арендаторе.</p>
Приоритет	<p>Порядок запуска блокирующих подписок.</p> <p>Ноль означает высший приоритет. Если в теме события заданы несколько блокирующих подписок на рабочий процесс с одинаковым приоритетом, подписки обрабатываются в алфавитном порядке по их именам.</p> <p>Этот параметр доступен только для блокирующих подписок на рабочий процесс.</p>
Время ожидания (мин.)	<p>Укажите предельное время выполнения рабочего процесса (в минутах), после которого фиксируется его сбой.</p> <p>Если рабочий процесс не удастся завершить в указанный промежуток времени, он отменяется и отправляется сообщение для следующей подписки из списка приоритетов.</p> <p>Если значение не указано, время ожидания не ограничено.</p> <p>В службах, ожидающих ответа, для блокирующих событий и событий с возможностью ответа могут быть указаны отдельные значения времени ожидания по умолчанию.</p> <p>Например, для тем событий жизненного цикла и подготовки Инфраструктура как услуга установлено время ожидания 30 минут. Это значение настроено на сервере Инфраструктура как услуга. Для тем подтверждения по умолчанию установлено значение «24 часа». Это значение настроено в системе.</p>
Описание	<p>По умолчанию отображаемое описание является описанием рабочего процесса.</p>

Таблица 1-10. Сведения о рабочем процессе (продолжение)

Сведения	Описание
Блокирование	<p>Определяет, может ли рабочий процесс блокировать получение сообщений о событиях в последующих рабочих процессах с одинаковой темой события во время ожидания ответа.</p> <p>В блокирующих подписках сообщения доставляются в порядке приоритета, но раньше, чем в подписках с той же темой события, которые не настроены как блокирующие. После завершения рабочего процесса отправляется сообщение следующей по приоритету блокирующей подписке. После обработки блокирующих подписок одновременно отправляется сообщение всем неблокирующим подпискам.</p> <p>Параметр блокирования доступен только для темы события с возможностью блокировки. Эта информация представлена на вкладке «Тема события».</p> <p>Применимость блокировки указана на вкладке «Тема события».</p> <ul style="list-style-type: none"> <li>■ Если флажок не установлен, брокер событий не блокирует последующие рабочие процессы.</li> <li>■ Если флажок установлен, брокер событий вычисляет на основе настроенных условий подходящие для этого события подписки на рабочий процесс, а затем запускает рабочие процессы в порядке приоритета. Брокер событий ожидает ответа от каждого рабочего процесса перед запуском следующего. Все параметры, измененные в ходе запуска текущего рабочего процесса, передаются следующему в очереди.</li> </ul> <p>Во время ожидания ответа никакие другие рабочие процессы не будут уведомлены о событии до получения ответа системы.</p> <p>Этот параметр нельзя изменить после создания подписки на рабочий процесс.</p>
При возникновении сбоя рабочего процесса остановите обработку.	<p>В случае сбоя блокирующей подписки на рабочий процесс последующие рабочие процессы не выполняются, пока не будет устранена данная ошибка. Сообщение об ошибке добавляется в журнал событий, и запрашивающему пользователю отправляется сообщение электронной почты.</p>

## 6. Щелкните элемент **Готово**.

## Работа с подписками на рабочие процессы подготовки и жизненного цикла

Создайте подписки на рабочие процессы подготовки и жизненного цикла, чтобы использовать vRealize Orchestrator для расширения возможностей управления компьютерами Инфраструктура как услуга. Подписки на подготовку предоставляют расширенные возможности в этом процессе. Подписки на жизненный цикл предоставляют пользователям расширенные возможности для управления подготовленными элементами.

## Интеграция службы Инфраструктура как услуга

Создайте подписку на рабочий процесс в соответствии с темой события жизненного цикла, которое запускает настраиваемый рабочий процесс vRealize Orchestrator на основе сообщения, сформированного службой Инфраструктура как услуга. vRealize Automation содержит две темы событий, которые можно использовать для интеграции Инфраструктура как услуга.

- Подготовка компьютера. Создайте подписки на рабочие процессы, которые запускают эти процессы во время подготовки и удаления компьютеров Инфраструктура как услуга.
- Жизненный цикл компьютера. Создайте подписки на рабочие процессы, которые запускают рабочие процессы, связанные с действиями управления, выполняемыми ответственным пользователем на подготовленном компьютере.

## Настройка рабочих процессов vRealize Orchestrator для подготовки и жизненного цикла

Необходимо настроить рабочие процессы vRealize Orchestrator для поддержки сообщения службы Инфраструктура как услуга.

### Схема темы событий подготовки и жизненного цикла

В темах событий подготовки и жизненного цикла компьютера используется одинаковая схема жизненного цикла. Различия заключаются в состояниях инициирования. В события подготовки компьютера доставляются сообщения, основанные на состояниях и событиях подготовки, а в события жизненного цикла — основанные на активных состояниях и событиях. В некоторые состояния подготовки входят значения BuildingMachine и Disposing. В некоторые состояния жизненного цикла включены значения InstallTools и Off.

Сообщение о событии — это полезная нагрузка данных о событии. Ниже приведена структура полезных данных о событии.

```
{
  machine : {
    id           : STRING,      /* IaaS machine ID */
    name         : STRING,      /* machine name */
    externalReference : STRING,  /* machine ID on the hypervisor */
    owner        : STRING,      /* machine owner */
    type         : INTEGER,     /* machine type: 0 – virtual machine; 1 – physical machine; 2
– cloud machine */
    properties   : Properties   /* machine properties, see notes below how to expose virtual
machine properties */
  },
  blueprintName : STRING,      /* blueprint name */
  componentId   : STRING,      /* component id */
  componentTypeId : STRING,    /* component type id */
  endpointId    : STRING,      /* endpoint id */
  requestId     : STRING,      /* request id */
  lifecycleState : {                                /* see Life Cycle State
Definitions*/
    state : STRING,
    phase : STRING,
    event : STRING
  }
}
```

```

},
virtualMachineEvent      : STRING,      /* fire an event on that machine – only processed
by Manager Service as consumer */
workflowNextState       : STRING,      /* force the workflow to a specific state – only
processed by Manager Service as consumer */
virtualMachineAddOrUpdateProperties : Properties, /* properties on the machine to add/update – only
processed by Manager Service as consumer */
virtualMachineDeleteProperties  : Properties /* properties to remove from the machine – only
processed by Manager Service as consumer */
}

```

Параметры vRealize Orchestrator сопоставляются с полезной нагрузкой события по имени и типу.

При использовании `virtualMachineEvent` и `workflowNextState` в качестве выходных параметров предоставляемые значения должны представлять состояние или событие из рабочего процесса, которое инициировало событие и запустило текущий рабочий процесс vRealize Orchestrator. Возможные состояния и события жизненного цикла см. в разделах [Состояния жизненного цикла основного рабочего процесса VMPS](#) и [Подготовка состояний жизненного цикла по типу компьютера](#).

### Работа с настраиваемыми свойствами расширяемости

Настраиваемые свойства виртуальной машины не включены в полезную нагрузку события, если они не указаны в качестве настраиваемого свойства расширяемости для состояния жизненного цикла. Можно добавить эти свойства к конечным точкам, резервированиям, схемам элементов, запросам и другим объектам Инфраструктура как услуга, которые поддерживают настраиваемые свойства.

Формат настраиваемого свойства, которое добавляется к объекту, имеет следующий вид `Extensibility.Lifecycle.Properties.{workflowName}.{stateName}`.

Например, если необходимо включить скрытые свойства и все свойства, начинающиеся со слова «Virtual», когда виртуальная машина находится в состоянии `BuildingMachine`, добавьте настраиваемые свойства к машине в схеме элементов. Именем настраиваемого свойства в этом примере является `Extensibility.Lifecycle.Properties.VMPSMasterWorkflow32.BuildingMachine`, а значениями — `__*` и `Virtual*`, которые разделяются запятой.

Двойное подчеркивание (`__*`) означает скрытые свойства. Значение `Virtual*` содержит все свойства, начинающиеся со слова «Virtual». Звездочка (\*) является подстановочным символом и может быть использована в качестве единственного значения, но ее использование в таком случае приведет к передаче больших объемов данных.

Если у вас несколько последовательно активируемых подписок на рабочий процесс, включающих настраиваемые свойства, вам необходимо добавить соответствующие записи в рабочие процессы, чтобы при проверке полезной нагрузки настраиваемые свойства сохранились.

Таблица 1-11. Записи задачи для сохранения настраиваемых свойств

Состояние	Записи задачи
Добавленные или обновленные настраиваемые свойства	<code>virtualMachineAddOrUpdateProperties = payload.virtualMachineAddOrUpdateProperties    new Properties();</code>
Удаленные настраиваемые свойства	<code>virtualMachineDeleteProperties = payload.virtualMachineDeleteProperties    new Properties();</code>

### Создание рабочего процесса vRealize Orchestrator на основе схемы жизненного цикла или подготовки

Создаваемый настраиваемый рабочий процесс должен содержать входной параметр `payload` с типом `Properties`. При запуске рабочего процесса в vRealize Orchestrator полезная нагрузка данных о событии подготовки или жизненного цикла будет вставлена в этот параметр. Помимо этого, можно включить отдельные входные параметры, которые соответствуют имени и типу полей в полезной нагрузке события.

### Описания состояний жизненного цикла подписки на рабочий процесс

При настройке условий подписки на рабочий процесс в соответствии с состояниями жизненного цикла следующие определения полезны для идентификации значений.

Каждое сообщение содержит элемент `lifecycleState`, который основан на изменениях состояний компьютера Инфраструктура как услуга.

Элемент имеет следующую структуру в сообщении.

```
lifecycleState : {
  state : STRING,
  phase : STRING,
  event : STRING
}
```

Таблица 1-12. Элементы LifecycleState

Свойство	Описание	Формат и значения	Примеры
state	Содержит имена рабочего процесса и состояния.	{workflowName}.{stateName}	<ul style="list-style-type: none"> <li>■ VMPSMasterWorkflow32.Requested</li> <li>■ VMPSMasterWorkflow32.MachineActivated</li> <li>■ BasicVmWorkflow.BuildComplete</li> </ul>
phase	Содержит этап запуска сообщения.	PRE, POST, EVENT	<ul style="list-style-type: none"> <li>■ PRE. Событие публикуется, когда переходит в это состояние.</li> <li>■ POST. Событие публикуется, когда находится в этом состоянии.</li> <li>■ EVENT. Событие публикуется, когда событие инфраструктуры как услуги переходит в это состояние.</li> </ul>
event	Содержит событие. Это свойство является дополнительным и существует только на этапе EVENT.	{workflowName}. {stateName}.EVENT.{eventName}	<ul style="list-style-type: none"> <li>■ VMPSMasterWorkflow32.Requested.EVENT.OnProvisionMachine</li> <li>■ VMPSMasterWorkflow32.VMPSMasterWorkflow32.EVENT.OnBuildSuccess</li> <li>■ BasicVmWorkflow.CreatingMachine.EVENT.OnCreatingMachineComplete</li> </ul>

### Состояния жизненного цикла основного рабочего процесса VMPS

Состояния жизненного цикла основного рабочего процесса VMPS отображают этапы жизненного цикла виртуальной машины Инфраструктура как услуга, от запроса до удаления. События и состояния основного рабочего процесса VMPS можно использовать при создании инициирующих условий на основе имен и событий состояний жизненного цикла.

Каждая виртуальная машина проходит через четыре этапа.

- Запрос, в том числе подтверждения.
- Подготовка, в том числе различные типы подготовки, например создание, клонирование, запуск или WIM.
- Управление, в том числе действия, например включение, выключение или создание моментального снимка.
- Удаление, в том числе деактивация, отмена подготовки и удаление компьютера.

Эти базовые этапы включены в основной рабочий процесс. Состояния *VMPSMasterWorkflow32* можно использовать при создании условий для следующих тем событий:

- жизненный цикл компьютера;
- подготовка компьютера.

Состояния глобальных событий являются сообщениями, отправляемыми брокеру событий основным процессом VMPS. Глобальные события можно инициировать в любое время.

Можно подписаться на клиент, который будет прослушивать события, но события не будут генерироваться, если в таблице для данной записи не задано инициирующее строковое значение. Например, События [инициирующая строка] (тема).

Таблица 1-13. Глобальные события

Состояние (тема)	События [строка инициации] (тема)
Global	<ul style="list-style-type: none"> <li>■ onBuildFailure (Provision)</li> <li>■ OnBuildSuccess (Provision)</li> <li>■ OnFinalizeMachine [Destroy] (Provision)</li> <li>■ OnForceUnregisterEvent [ForceUnregister] (Provision)</li> <li>■ ReconfigureVM.Pending [ReconfigureVM.Pending] (Active)</li> <li>■ ReconfigureVM.ExecutionUpdated (Active)</li> <li>■ ReconfigureVM.RetryRequestMade (Active)</li> <li>■ ReconfigureVM.Failed (Active)</li> <li>■ ReconfigureVM.Successful (Active)</li> <li>■ ReconfigureVM.Complete (Active)</li> <li>■ ReconfigureVM.Canceled (Active)</li> </ul>

Активные глобальные состояния — это действия, которые можно запускать на подготовленных компьютерах.

Таблица 1-14. Активные события

Состояние	События [строка инициации] (тема)
Active	<ul style="list-style-type: none"> <li>■ OnExpireLease [Expire] (Active)</li> <li>■ OnForceExpire [ForceExpire] (Active)</li> <li>■ onReprovision [Reprovision] (Active)</li> <li>■ onResetBuildSuccess [ResetBuildSuccess] (Active)</li> </ul>

В основном рабочем процессе события подготовки происходят во время жизненного цикла подготовки компьютера. Активные состояния — это действия, которые можно запускать на подготовленных компьютерах. Пример основного рабочего процесса см. в разделе [Пример основного рабочего процесса VMPS](#).

Для каждого типа компьютеров существует собственный рабочий процесс подготовки. Подробные сведения об отдельных типах компьютеров см. в разделе [Подготовка состояний жизненного цикла по типу компьютера](#).

Таблица 1-15. События и состояния VMPSMasterWorkflow32

Состояние (тема)	События [строка инициации] (тема)
BuildingMachine	
■ Pre(Provision)	
■ Post(Provision)	
DeactivateMachine	
■ Pre(Provision)	
■ Post(Provision)	
Disposing	<ul style="list-style-type: none"> <li>■ OnDisposeComplete(Provision)</li> <li>■ OnDisposeTimeout(Provision)</li> <li>■ OnUnregisterMachine [Unregister] (Provision)</li> </ul>
■ Pre(Provision)	
■ Post(Provision)	

Таблица 1-15. События и состояния VMPSMasterWorkflow32 (продолжение)

Состояние (тема)	События [строка инициации] (тема)
Expired	■ OnActiveExpiredMachine [ActivateExpiredMachine] (Active)
■ Pre(Active)	■ TurnOffFromExpired [TurnOffExpiredMachine] (Active)
■ Post(Active)	
InstallTools	■ InstallToolsComplete(Active)
■ Pre(Active)	■ TimeoutInstallTools(Active)
■ Post(Active)	
Leased	■ OnChangeLease (Active)
	■ OnUpdateDescription (Active)
	■ OnUpdateOwner (Active)
MachineActivated	■ OnCatalogRegistrationComplete (Provision)
■ Pre(Provision)	
■ Post(Provision)	
MachineProvisioned	
■ Pre(Provision)	
■ Post(Provision)	
Off	■ OnForceOn [ForceOn] (Active)
■ Pre(Active)	■ OnResetOff [Turn Off] (Active)
■ Post(Active)	■ OnTurnOn [Turn On] (Active)
On	■ OnForceOff [ForceOff] (Active)
■ Pre(Active)	■ onInstallTools [InstallTools] (Active)
■ Post(Active)	■ OnReboot [Reboot] (Active)
	■ OnReset [Reset] (Active)
	■ OnResetOn [Turn On] (Active)
	■ OnShutdown [Shutdown] (Active)
	■ OnSuspend [Suspend] (Active)
	■ OnTurnOff [Turn Off] (Active)
Rebooting	■ OnRebootComplete(Active)
■ Pre(Active)	■ TimeoutFromReboot(Active)
■ Post(Active)	
RegisterMachine	■ onRegisterComplete(Provision)
■ Pre(Provision)	■ RegisterTimeout(Provision)
■ Post(Provision)	
Requested	■ OnProvisionMachine [Provision] (Provision)
■ Pre(Provision)	
■ Post(Provision)	
Resetting	■ OnResetComplete(Active)
■ Pre(Active)	■ TimeoutFromReset(Active)
■ Post(Active)	
ShuttingDown	■ OnShutdownComplete(Active)
■ Pre(Active)	■ TimeoutFromShutdown(Active)
■ Post(Active)	

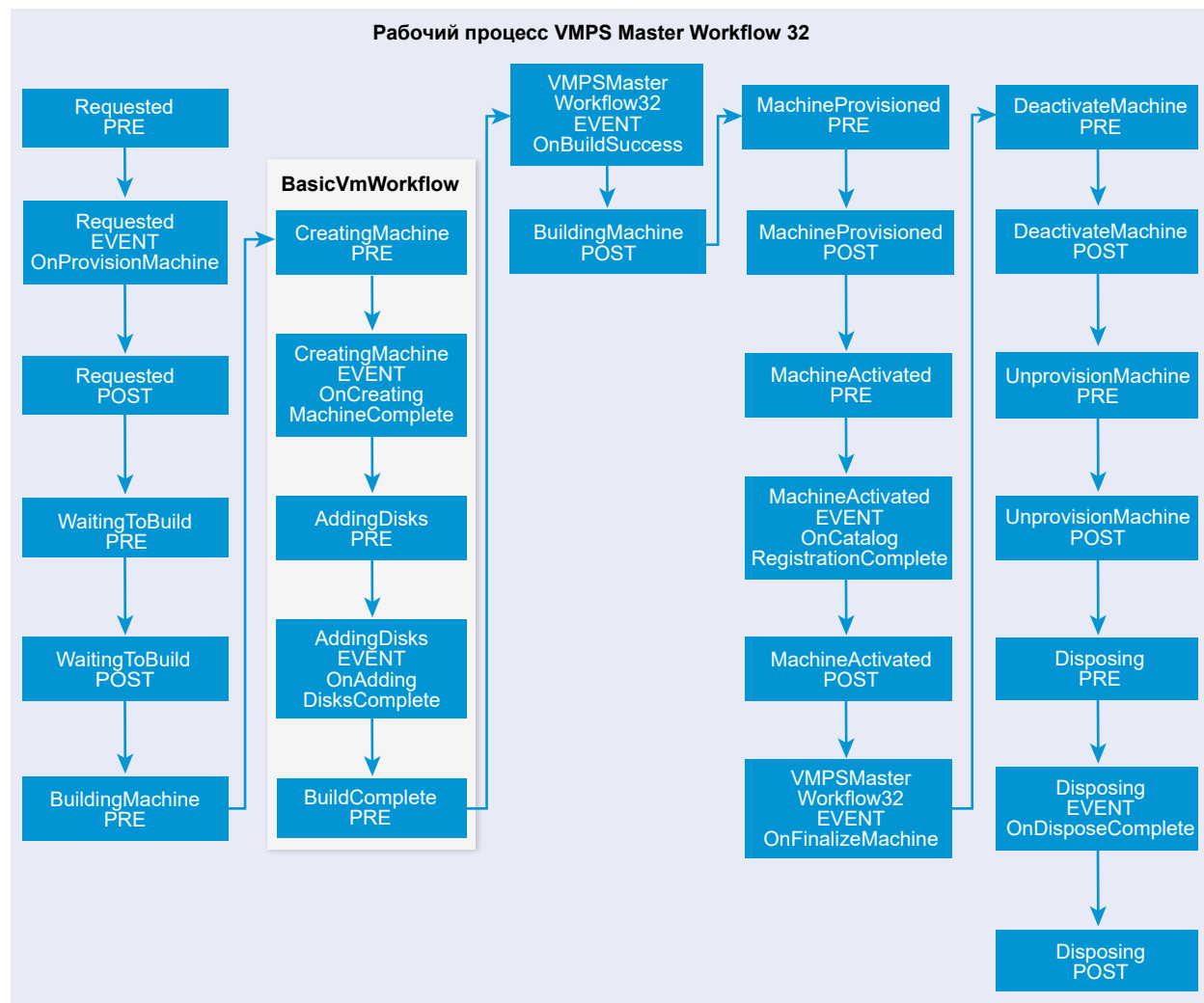


Таблица 1-15. События и состояния VMPSMasterWorkflow32 (продолжение)

Состояние (тема)	События [строка инициации] (тема)
Suspending ■ Pre(Active) ■ Post(Active)	■ OnSuspendComplete(Active) ■ TimeoutFromSuspend(Active)
TurningOff ■ Pre(Active) ■ Post(Active)	■ OnTurningOffComplete(Active) ■ TimeoutFromPowerOff(Active)
TurningOn ■ Pre(Active) ■ Post(Active)	■ OnTurningOnComplete(Active) ■ TimeoutPowerOn(Active)
UnprovisionMachine ■ Pre(Provision) ■ Post(Provision)	
WaitingToBuild ■ Pre(Provision) ■ Post(Provision)	

### Пример основного рабочего процесса VMPS

VMPS является основным рабочим процессом, в который встраиваются другие рабочие процессы подготовки. Этот пример включает в себя основной рабочий процесс виртуальной машины для иллюстрации жизненного цикла виртуальной машины. Он не представляет конкретный рабочий процесс в среде.



## Подготовка состояний жизненного цикла по типу компьютера

В зависимости от типа виртуальной машины используются определенные состояния жизненного цикла по типу компьютера. В дополнение к основному рабочему процессу можно использовать события и состояния рабочего процесса подготовки при создании иницирующих условий для подписок на рабочие процессы.

Можно подписаться на клиент, который будет прослушивать события, но события не будут генерироваться, если в таблице для данной записи не задано иницирующее строковое значение.

Например, События [иницирующая строка] (тема).

## Исходное состояние системы BladeLogic

Состояние (тема)	События (тема)
BuildFinished	
■ Pre(Provision)	
CreatingMachine	
■ Pre(Provision)	

## Исходное состояние системы Opsware

Состояние (тема)	События (тема)
BuildFinished	
■ Pre(Provision)	
OpswareRegister	■ OnOpswareRegister(Provision)
■ Pre(Provision)	

## Рабочий процесс подготовки облака

Состояние (тема)	События (тема)
BuildComplete	
■ Pre(Provision)	
CloudProvisioning	■ OnCloudProvisioningTimeout(Provision)
■ Pre(Provision)	
FailedProvisioning	
■ Pre(Provision)	

## Рабочий процесс подготовки службы приложения

Состояние (тема)	События (тема)
AppServiceProvisioning	■ OnAppServiceProvisioningTimeout(Provision)
■ Pre(Provision)	
BuildComplete	
■ Pre(Provision)	
FailedProvisioning	
■ Pre(Provision)	

## Базовый рабочий процесс виртуальной машины

Состояние (тема)	События (тема)
AddingDisks	■ OnAddingDisksComplete(Provision)
■ Pre(Provision)	■ OnAddingDisksTimeout(Provision)
BuildComplete	
■ Pre(Provision)	
CreatingMachine	■ OnCreatingMachineComplete(Provision)
■ Pre(Provision)	■ OnCreatingMachineTimeout(Provision)
FailedProvisioning	
■ Pre(Provision)	

## Opware Virtual

Состояние (тема)	События (тема)
AddingDisks	■ OnAddingDisksComplete(Provision)
■ Pre(Provision)	■ OnAddingDisksTimeout(Provision)
BuildFinished	
■ Pre(Provision)	
CreatingVM	■ OnCreateVMComplete(Provision)
■ Pre(Provision)	■ OnCreateVMTimeout(Provision)
InitialPowerOn	■ OnInitialPowerOnComplete(Provision)
■ Pre(Provision)	■ OnInitialPowerOnTimeout(Provision)
OpwareRegister	■ OnOpwareRegister(Provision)
■ Pre(Provision)	

## Облачный рабочий процесс запуска Linux

Состояние (тема)	События (тема)
BuildComplete	
■ Pre(Provision)	
CreatingMachine	■ OnCreatingMachineComplete(Provision)
■ Pre(Provision)	■ OnCreatingMachineTimeout(Provision)
CustomizeOS	■ OnCustomizeOSComplete(Provision)
■ Pre(Provision)	■ OnCustomizeOSTimeout(Provision)
FailedProvisioning	
■ Pre(Provision)	
InitialPowerOn	■ OnInitialPowerOnComplete(Provision)
■ Pre(Provision)	■ OnInitialPowerOnTimeout(Provision)
InstallingOS	■ OnInstallingOSComplete(Provision)
■ Pre(Provision)	■ OnInstallingOSTimeout(Provision)

## Рабочий процесс клонирования

Состояние (тема)	События (тема)
BuildComplete	
■ Pre(Provision)	
CloneMachine	■ OnCloneMachineComplete(Provision)
■ Pre(Provision)	■ OnCloneMachineTimeout(Provision)
CustomizeMachine	■ OnCustomizeMachineComplete(Provision)
■ Pre(Provision)	■ OnCustomizeMachineTimeout(Provision)
CustomizeOS	■ OnCustomizeOS(Provision)
	■ OnCustomizeOSComplete(Provision)
	■ OnCustomizeOSTimeout(Provision)
EjectCD	■ OnEjectCDComplete(Provision)
■ Pre(Provision)	■ OnEjectCDTimeout(Provision)

Состояние (тема)	События (тема)
FailedProvisioning	
■ Pre(Provision)	
FinalizeProvisioning	■ OnFinalizeComplete(Provision)
■ Pre(Provision)	■ OnFinalizeTimeout(Provision)
InitialPowerOn	■ OnInitialPowerOnComplete(Provision)
■ Pre(Provision)	■ OnInitialPowerOnTimeout(Provision)
InstallSoftware	■ OnInstallSoftwareComplete(Provision)
■ Pre(Provision)	■ OnInstallSoftwareTimeout(Provision)
MountCD	■ OnMountCDComplete(Provision)
■ Pre(Provision)	■ OnMountCDTimeout(Provision)
PostInstallSoftwareChecks	
■ Pre(Provision)	
PrepareInstallSoftware	
■ Pre(Provision)	

### Облачный рабочий процесс образа WIM

Состояние (тема)	События (тема)
BuildComplete	
■ Pre(Provision)	
CreatingMachine	■ OnCreatingMachineComplete(Provision)
■ Pre(Provision)	■ OnCreatingMachineTimeout(Provision)
FailedProvisioning	
■ Pre(Provision)	
InitialPowerOn	■ OnInitialPowerOnComplete(Provision)
■ Pre(Provision)	■ OnInitialPowerOnTimeout(Provision)
InstallOS	■ onInstallOSComplete(Provision)
■ Pre(Provision)	■ OnInstallOSTimeout(Provision)
Перезагрузка	■ OnRebootComplete(Provision)
■ Pre(Provision)	■ OnRebootTimeout(Provision)
SetupOS	■ OnSetupOSComplete(Provision)
■ Pre(Provision)	■ OnSetupOSTimeout(Provision)

### Внешний рабочий процесс подготовки

Состояние (тема)	События (тема)
AddingDisks	■ OnAddingDisksComplete(Provision)
■ Pre(Provision)	■ OnAddingDisksTimeout(Provision)
BuildComplete	
■ Pre(Provision)	
CreatingMachine	■ OnCreatingMachineComplete(Provision)
■ Pre(Provision)	■ OnCreatingMachineTimeout(Provision)

Состояние (тема)	События (тема)
EpiRegister	■ OnEpiRegisterComplete(Provision)
■ Pre(Provision)	
FailedProvisioning	
■ Pre(Provision)	
InitialPowerOn	■ OnInitialPowerOnComplete(Provision)
■ Pre(Provision)	■ OnInitialPowerOnTimeout(Provision)

## Рабочий процесс запуска Linux

Состояние (тема)	События (тема)
AddingDisks	■ OnAddingDisksComplete(Provision)
■ Pre(Provision)	■ OnAddingDisksTimeout(Provision)
BuildComplete	
■ Pre(Provision)	
CreatingMachine	■ OnCreatingMachineComplete(Provision)
■ Pre(Provision)	■ OnCreatingMachineTimeout(Provision)
CustomizeOS	■ OnCustomizeOSComplete(Provision)
■ Pre(Provision)	■ OnCustomizeOSTimeout(Provision)
EjectingCD	■ OnEjectingCDComplete(Provision)
■ Pre(Provision)	■ OnEjectingCDTimeout(Provision)
FailedProvisioning	
■ Pre(Provision)	
InitialPowerOn	■ OnInitialPowerOnComplete(Provision)
■ Pre(Provision)	■ OnInitialPowerOnTimeout(Provision)
InstallingOS	■ OnInstallingOSComplete(Provision)
■ Pre(Provision)	■ OnInstallingOSTimeout(Provision)

## Физический рабочий процесс подготовки

Состояние (тема)	События (тема)
FailedProvisioning	
■ Pre(Provision)	
FinalizeProvisioning	■ OnFinalizeProvisioningTimeout(Provision)
■ Pre(Provision)	
InitializeProvisioning	■ OnInitializeProvisioningTimeout(Provision)
■ Pre(Provision)	
InitialPowerOn	■ OnInitialPowerOnTimeout(Provision)
■ Pre(Provision)	
InstallOS	■ OnInstallOSComplete(Provision)
■ Pre(Provision)	■ OnInstallOSTimeout(Provision)

Состояние (тема)	События (тема)
Перезагрузка	■ OnRebootComplete(Provision)
■ Pre(Provision)	■ OnRebootTimeout(Provision)
SetupOS	■ OnSetupOSComplete(Provision)
■ Pre(Provision)	■ OnSetupOSTimeout(Provision)

### Физический рабочий процесс подготовки PXE

Состояние (тема)	События (тема)
CheckHardwareType	
■ Pre(Provision)	
CleanPxe	■ OnCleanPxeTimeout(Provision)
■ Pre(Provision)	
FailedProvisioning	
■ Pre(Provision)	
FinalizeProvisioning	■ OnFinalizeProvisioningTimeout(Provision)
■ Pre(Provision)	
InitializeProvisioning	■ OnInitializeProvisioningTimeout(Provision)
■ Pre(Provision)	
InitialPowerOn	■ OnInitialPowerOnTimeout(Provision)
■ Pre(Provision)	
InstallOS	■ OnInstallOSComplete(Provision)
■ Pre(Provision)	■ OnInstallOSTimeout(Provision)
Перезагрузка	■ OnRebootComplete(Provision)
■ Pre(Provision)	■ OnRebootTimeout(Provision)
SetupOS	■ OnSetupOSComplete(Provision)
■ Pre(Provision)	■ OnSetupOSTimeout(Provision)
SetupPxe	■ OnSetupPxeTimeout(Provision)
■ Pre(Provision)	

### Физический рабочий процесс подготовки SCCM

Состояние (тема)	События (тема)
CheckHardwareType	
■ Pre(Provision)	
Complete	■ OnCompleteProvisioningComplete(Provision)
■ Pre(Provision)	■ OnCompleteProvisioningTimeout(Provision)
FailedProvisioning	■ OnFailedProvisioningTimeout(Provision)
■ Pre(Provision)	
FinalizeProvisioning	■ OnFinalizeProvisioningTimeout(Provision)
■ Pre(Provision)	
InitializeProvisioning	■ OnInitializeProvisioningTimeout(Provision)
■ Pre(Provision)	

Состояние (тема)	События (тема)
InitialPowerOn ■ Pre(Provision)	■ OnInitialPowerOnTimeout(Provision)
SccmRegistration ■ Pre(Provision)	■ OnSccmRegistrationTimeout(Provision)

### Физический рабочий процесс подготовки SCCM PXE

Состояние (тема)	События (тема)
CheckHardwareType ■ Pre(Provision)	
CleanPxe ■ Pre(Provision)	■ OnCleanPxeTimeout(Provision)
Complete ■ Pre(Provision)	■ OnCompleteProvisioningComplete(Provision) ■ OnCompleteProvisioningTimeout(Provision)
Disposing ■ Pre(Provision)	
FailedProvisioning ■ Pre(Provision)	■ OnFailedProvisioningTimeout(Provision)
FinalizeProvisioning ■ Pre(Provision)	■ OnFinalizeProvisioningTimeout(Provision)
InitializeProvisioning ■ Pre(Provision)	■ OnInitializeProvisioningTimeout(Provision)
InitialPowerOn ■ Pre(Provision)	■ OnInitialPowerOnTimeout(Provision)
SccmRegistration ■ Pre(Provision)	■ OnSccmRegistrationTimeout(Provision)
SetupPxe ■ Pre(Provision)	■ OnSetupPxeTimeout(Provision)

### Рабочий процесс клонирования vApp

Состояние (тема)	События [строка инициации] (тема)
Global	■ OnFailProvisioning(Provision) ■ OnMasterProvisioned(Provision)
BuildComplete ■ Pre(Provision)	
CloneMachine ■ Pre(Provision)	■ OnCloneMachineComplete(Provision) ■ OnCloneMachineTimeout(Provision)
CustomizeMachine ■ Pre(Provision)	■ OnCustomizeMachineComplete(Provision) ■ OnCustomizeMachineTimeout(Provision)



Состояние (тема)	События [строка инициации] (тема)
CustomizeOS	<ul style="list-style-type: none"> <li>■ OnCustomizeOS(Provision)</li> <li>■ OnCustomizeOSComplete(Provision)</li> <li>■ OnCustomizeOSTimeout(Provision)</li> </ul>
FailedProvisioning	
■ Pre(Provision)	
FinalizeProvisioning	<ul style="list-style-type: none"> <li>■ OnFinalizeComplete(Provision)</li> <li>■ OnFinalizeTimeout(Provision)</li> </ul>
■ Pre(Provision)	
InitialPowerOn	<ul style="list-style-type: none"> <li>■ OnInitialPowerOnComplete(Provision)</li> <li>■ OnInitialPowerOnTimeout(Provision)</li> </ul>
■ Pre(Provision)	
WaitingForMaster	<ul style="list-style-type: none"> <li>■ OnWaitingForMasterTimeout(Provision)</li> </ul>
■ Pre(Provision)	

## Виртуальный рабочий процесс подготовки SCCM

Состояние (тема)	События (тема)
AddingDisks	<ul style="list-style-type: none"> <li>■ OnAddingDisksComplete(Provision)</li> <li>■ OnAddingDisksTimeout(Provision)</li> </ul>
■ Pre(Provision)	
BuildComplete	
■ Pre(Provision)	
CreatingMachine	<ul style="list-style-type: none"> <li>■ CreatingMachineComplete(Provision)</li> <li>■ OnCreatingMachineTimeout(Provision)</li> </ul>
■ Pre(Provision)	
Disposing	
■ Pre(Provision)	
EjectingCD	<ul style="list-style-type: none"> <li>■ OnEjectingCDComplete(Provision)</li> <li>■ OnEjectingCDTimeout(Provision)</li> </ul>
■ Pre(Provision)	
FailedProvisioning	
■ Pre(Provision)	
InitialPowerOn	<ul style="list-style-type: none"> <li>■ OnInitialPowerOnComplete(Provision)</li> <li>■ OnPowerOnTimeout(Provision)</li> </ul>
■ Pre(Provision)	
InstallingOS	<ul style="list-style-type: none"> <li>■ OnInstallingOSComplete(Provision)</li> <li>■ OnInstallingOSTimeout(Provision)</li> </ul>
■ Pre(Provision)	
SccmRegistration	<ul style="list-style-type: none"> <li>■ OnSccmRegistrationTimeout(Provision)</li> </ul>
■ Pre(Provision)	

## Рабочий процесс образа WIM

Состояние (тема)	События (тема)
AddingDisks	<ul style="list-style-type: none"> <li>■ OnAddingDisksComplete(Provision)</li> <li>■ OnAddingDisksTimeout(Provision)</li> </ul>
■ Pre(Provision)	
BuildComplete	
■ Pre(Provision)	

Состояние (тема)	События (тема)
CreatingMachine ■ Pre(Provision)	■ OnCreatingMachineComplete(Provision) ■ OnCreatingMachineTimeout(Provision)
EjectingCD ■ Pre(Provision)	■ OnEjectingCDComplete(Provision) ■ OnEjectingCDTimeout(Provision)
FailedProvisioning ■ Pre(Provision)	
InitialPowerOn ■ Pre(Provision)	■ OnInitialPowerOnComplete(Provision) ■ OnInitialPowerOnTimeout(Provision)
InstallOS ■ Pre(Provision)	■ onInstallOSComplete(Provision) ■ OnInstallOSTimeout(Provision)
Перезагрузка ■ Pre(Provision)	■ OnRebootComplete(Provision) ■ OnRebootTimeout(Provision)
SetupOS ■ Pre(Provision)	■ OnSetupOSComplete(Provision) ■ OnSetupOSTimeout(Provision)

## Настройка значений времени ожидания для состояний и событий

По умолчанию значение времени ожидания для всех состояний и событий составляет 30 минут и устанавливается в глобальных параметрах vRealize Automation. Для успешного запуска некоторых рабочих процессов может потребоваться больше времени. Чтобы обеспечить различные рабочие процессы в среде, можно добавить значения переопределения времени ожидания для отдельных рабочих процессов или состояний.

Чтобы изменить значения времени ожидания по умолчанию, выберите **Инфраструктура > Администрирование > Глобальные параметры** и измените значение для параметра **Время ожидания сообщения о жизненном цикле расширяемости**. После внесения изменений в глобальные параметры необходимо перезапустить службу диспетчера.

Чтобы настроить отдельные значения времени ожидания, добавьте рабочий процесс или свойство события в раздел appSetting файла `ManagerService.exe.config` file, расположенного на сервере Инфраструктура как услуга. Файл, как правило, расположен в каталоге `%System-Drive%\Program Files x86\VMware\VCAC\Server`. Прежде чем изменять файл, следует сделать его копию. После внесения изменений в отдельные параметры необходимо перезапустить службу диспетчера.

Базовый формат для ключей похож на следующие примеры.

- Для рабочего процесса: `Extensibility.{workflow}.Timeout.`
- Для событий: `Extensibility.{workflow}.{state}.EVENT.{event}.Timeout.`
- Для состояний: `Extensibility.{workflow}.{state}.(PRE/POST).Timeout.`

Используйте следующие примеры при добавлении ключей в раздел `appSetting`. Формат значения времени ожидания: D.HH:mm:ss.ms. D — день, ms — миллисекунды. День и миллисекунды необязательные значения. А часы, минуты и секунды являются обязательными значениями.

- Чтобы установить время ожидания 30 минут для всего рабочего процесса `BasicVmWorkflow`, добавьте `<add key="Extensibility.BasicVmWorkflow.Timeout" value="00:30:00"/>`.
- Чтобы установить время ожидания для глобального события `OnFinalizeMachine` в рабочем процессе `VMPSMasterWorkflow32` равное двум часам, добавьте `<add key="Extensibility.VMPSMasterWorkflow32.VMPSMasterWorkflow32.EVENT.OnFinalizeMachine.Timeout" value="02:00:00"/>`.
- Чтобы установить время ожидания для состояния предварительного запроса `VMPSMasterWorkflow32` равное двум дням, добавьте `<add key="Extensibility.VMPSMasterWorkflow32.Requested.PRE.Timeout" value="2.00:00:00"/>`.

## Настройка поведения при возникновении ошибки для состояний и событий

При обработке ошибки и времени ожидания для подписки на рабочий процесс существует поведение по умолчанию. Можно настроить поведение для компьютеров среды.

Инфраструктура как услуга обрабатывает время ожидания события и ошибку при выполнении процесса из службы брокера событий.

При переходе из каждого состояния `SendEBSMessage` отправляет событие в службу брокера событий и ожидает ответа. По умолчанию, если служба брокера события сообщает о возможном окончании времени ожидания или ошибке, это событие регистрируется в журнале и рабочий процесс возобновляется.

Если истекает время ожидания или возникает ошибка в следующих состояниях основного рабочего процесса, происходит принудительный переход рабочего процесса в состояние ошибки, а не его возобновление.

Таблица 1-16. Исключения, при которых рабочий процесс не возобновляется

Состояния, при которых возникает ошибка	Состояние ошибки
PRE MachineProvisioned	UnprovisionMachine
PRE BuildingMachine	Disposing
PRE RegisterMachine	Finalized

Чтобы настроить поведение при истечении времени ожидания или возникновении ошибки, можно добавлять настраиваемые свойства компьютеру для любых событий или состояний, в которых необходимо запустить событие или принудительно изменить состояние. Используйте следующие примеры для конфигурации настраиваемых свойств.

- `Extensibility.Lifecycle.Error.Event.{Workflow}.{State}`. Значением свойства является имя события, которое необходимо запустить в рабочем процессе в случае истечения времени ожидания или возникновения ошибки.

- `Extensibility.Lifecycle.Error.State.{Workflow}.{State}`. Значением свойства является имя состояния, в которое рабочий процесс будет принудительно переведен в случае истечения времени ожидания или возникновения ошибки.

## Сценарий. Создание моментального снимка последующей подготовки виртуальной машины

Администратору арендатора необходимо, чтобы у пользователей каталога служб был моментальный снимок последующей подготовки их виртуальных машин. В этом случае они смогут откатываться до свежей версии компьютера, а не запрашивать новый компьютер.

### Процедура

1. [Сценарий. Создание рабочего процесса vRealize Orchestrator для действия моментального снимка последующей подготовки](#)

Можно создать рабочий процесс vRealize Orchestrator, который принимает необходимый входной параметр. Разработайте рабочий процесс для достижения цели последующей подготовки.

2. [Сценарий. Создание подписки на рабочий процесс формирования моментального снимка для последующей подготовки](#)

После создания виртуальной машины администратору арендатора необходимо создать ее моментальный снимок. Настройте подписку на рабочий процесс на основе темы события подготовки компьютера, и опубликуйте ее, чтобы сделать активной.

## Сценарий. Создание рабочего процесса vRealize Orchestrator для действия моментального снимка последующей подготовки

Можно создать рабочий процесс vRealize Orchestrator, который принимает необходимый входной параметр. Разработайте рабочий процесс для достижения цели последующей подготовки.

Дополнительные сведения по созданию папок и рабочих процессов vRealize Orchestrator см. в разделе *Разработка с помощью VMware vRealize Orchestrator*.

### Необходимые условия

Войдите в продукт vRealize Orchestrator, который является экземпляром, настроенным для vRealize Automation с привилегиями на создание рабочего процесса.

### Процедура

1. Создайте папку для рабочих процессов подписки на рабочий процесс в соответствующей библиотеке.
2. Создайте новый рабочий процесс.

Для этого сценария назовите рабочий процесс

**Моментальный снимок последующей подготовки автоматизации.**

3. Добавьте следующий входной параметр.

Имя	Тип
payload	Properties

4. Создайте задачу с возможностью добавления сценариев, которая принимает входной параметр и создает моментальный снимок виртуальной машины.
5. Сохраните рабочий процесс.

#### Следующие шаги

Создайте подписку рабочего процесса, которая запускает рабочий процесс моментального снимка последующей подготовки автоматизации. [Сценарий. Создание подписки на рабочий процесс формирования моментального снимка для последующей подготовки](#)

#### Сценарий. Создание подписки на рабочий процесс формирования моментального снимка для последующей подготовки

После создания виртуальной машины администратору арендатора необходимо создать ее моментальный снимок. Настройте подписку на рабочий процесс на основе темы события подготовки компьютера, и опубликуйте ее, чтобы сделать активной.

Настройте подписку на рабочий процесс, чтобы запустить рабочий процесс создания моментального снимка после подготовки виртуальной машины и обнаружения сообщения об активном состоянии события.

#### Необходимые условия

- Войдите в vRealize Automation в качестве **администратора арендатора**.
- Настройте подключаемый модуль vCenter Server в качестве конечной точки vRealize Orchestrator. См. *Настройка vRealize Automation*.
- Убедитесь в наличии схемы элементов виртуальной машины vSphere.
- Убедитесь в наличии рабочего процесса vRealize Orchestrator, создающего моментальный снимок виртуальной машины. Нельзя использовать рабочий процесс по созданию моментального снимка, который предлагается в подключаемом модуле vRealize Automation. Предоставленный рабочий процесс по созданию моментального снимка Все как услуга См. [Настройка рабочих процессов vRealize Orchestrator для подготовки и жизненного цикла](#).

#### Процедура

1. Выберите **Администрирование > События > Подписки**
2. Выберите значок **Создать (+)**.
3. Выберите **Подготовка компьютера**.
4. Нажмите кнопку **Далее**.
5. На вкладке «Условия» настройте условия запуска.
  - а) Выберите **Запускать при определенных условиях**.
  - б) В раскрывающемся меню **Предложение** выберите **Все из указанного**.

в) Настройте следующие условия.

Свойство	Оператор	Значение
Данные > Компьютер > Тип компьютера	Равен	Константа > Виртуальная машина
Данные > Состояние жизненного цикла > Имя состояния жизненного цикла	Равен	Константа > VMPSMasterWorkflow32.MachineActivated
Данные > Состояние жизненного цикла > Фаза состояния	Равен	Константа > POST

г) Нажмите кнопку **Далее**.

**6.** На вкладке «Рабочий процесс» выберите дерево Orchestrator и укажите рабочий процесс

**Автоматизация создания моментального снимка последующей обработки.**

**7.** Нажмите кнопку **Далее**.

**8.** На вкладке «Сведения» введите **Имя** и **Описание**.

В этом сценарии введите

**Моментальный снимок виртуальной машины последующей обработки** в качестве имени и **Создать моментальный снимок после подготовки и активации новой виртуальной машины** в качестве описания.

**9.** Щелкните элемент **Готово**.

**10.** Выберите строку «Моментальный снимок виртуальной машины последующей обработки» и нажмите кнопку **Опубликовать**.

## Результаты

Подписка на рабочий процесс является активной и запускает рабочий процесс по созданию моментального снимка, когда сообщение о событии указывает на завершение подготовки и активации запрашиваемой виртуальной машины.

## Следующие шаги

Для проверки подписки на рабочий процесс запросите виртуальную машину в каталоге служб. После получения ответа на запрос, где указано успешное завершение подготовки, убедитесь, что моментальный снимок был создан.

## Работа с подписками на рабочие процессы подтверждения

Создайте подписки на рабочие процессы предварительного и последующего подтверждения, чтобы можно было отправить запрос подтверждения на обработку во внешнюю систему. Затем подтвержденный или отклоненный ответ будет обработан vRealize Automation.

## Интеграция службы подтверждения

Создайте подписку на рабочий процесс предварительного или последующего подтверждения, которая запускает настраиваемый рабочий процесс vRealize Orchestrator обработки запроса на подтверждение в системе за пределами vRealize Automation.

На уровне подтверждения в политике подтверждения можно выбрать **Использовать подписку на событие** в качестве утверждающего. Это один из нескольких уровней в политике подтверждения. Если пользователь каталога служб запрашивает элемент, к которому применяется политика подтверждения, содержащая параметр **Использовать утверждающего подписки на событие**, то служба подтверждения отправляет сообщение в службу брокера событий со следующими результатами.

- При наличии опубликованной подписки на рабочий процесс, соответствующей критериям, vRealize Orchestrator запускает рабочий процесс подтверждения или отклонения.
- Если у вас есть опубликованная подписка на рабочий процесс, но несоответствующая критериям, вы отменили публикацию подписки на рабочий процесс, или у вас нет опубликованной подписки, то уровень подтверждения принимает значение «утверждено» и процесс подтверждения переходит на следующий уровень.

Подписка на рабочий процесс подтверждения получает сообщения от службы подтверждения и сравнивает сообщения с настроенными критериями для подписки подтверждения. Когда найдено сообщение, соответствующее критериям, запускается выбранный рабочий процесс vRealize Orchestrator. Опубликованные данные о событии передаются в рабочий процесс в качестве входных данных и обрабатываются с помощью метода, указанного в рабочем процессе. Результаты процесса возвращаются в vRealize Automation, и обрабатывается запрос. При значении «утверждено» выполняется оценка следующего уровня подтверждения. При значении «отклонено» запрос отклоняется. Если служба подтверждения не получает ответ в течение 24 часов (время ожидания по умолчанию для этой службы), запрос обрабатывается как отклоненный.

## Настройка рабочих процессов vRealize Orchestrator для тем событий подтверждения

Необходимо настроить рабочий процесс vRealize Orchestrator для поддержки сообщения подтверждения и ответа с правильно отформатированными сведениями, которые может обработать vRealize Automation.

### Схема тем событий подтверждения

В схему сообщений о событии предварительного и последующего подтверждения входят имена и значения полей, а также информация, содержащаяся в запросе, и сведения о его источнике.

Ниже приведена структура полезных данных о событии.

```
{
  fieldNames : Properties,           // Property names

  fieldValues : Properties,          // Property values

  // Information about the request
  requestInfo : {
    requestRef : STRING,              // Identifier for the source request
    itemName : STRING,               // Name of the requested item
    itemDescription : STRING,         // Description of the requested item
    reason : STRING,                  // Justification provided by the user specifying why the
request is required
    description : STRING,             // Description entered by the user specifying the purpose of
the request
  }
}
```

```

        approvalLevel:ExternalReference,// Approval level ID. This is a searchable field
        approvalLevelName : STRING,      // Approval level name
        createDate : DATE_TIME,          // Time the approval request is created
        requestedFor : STRING,            // Principal id of the user for whom the source request is
initiated
        subtenantId : STRING,              // Business group id
        requestedBy : STRING                // Principal id of the user who actually submits the request
    },

    // Information about the source of the request
    sourceInfo : {
        externalInstanceId : STRING,      // Identifier of the source object, as defined by the
initiator service
        serviceId : STRING,              // Identifier of the service which initiated the approval
        externalClassId : STRING          // Identifier of the class to which the source object belongs
    }
}

```

Имена свойств и их значения могут быть настраиваемыми свойствами или свойствами системы, которые настраиваются на уровне политики подтверждения. Эти свойства настраиваются в политике подтверждения, позволяя утверждающему изменять значения в процессе подтверждения. Например, при включенном процессоре утверждающий может уменьшить количество процессоров в форме запроса подтверждения.

Полезными данными о событии ответа является информация, которую он вернул в vRealize Automation с помощью рабочего процесса. Содержимое полезной нагрузки ответа определяет, утвержден ли запрос или отклонен.

```

{
    approved : BOOLEAN,

    // Property values
    fieldValues : Properties
}

```

Подтвержденный параметр в полезной нагрузке события ответа принимает значение «истина» для подтвержденных запросов или «ложь» для отклоненных. Значения свойства являются настраиваемыми свойствами или свойствами системы, которые были изменены рабочим процессом vRealize Orchestrator, возвращены vRealize Automation и включены в процесс подтверждения.

Рекомендуется настраивать рабочий процесс vRealize Orchestrator с помощью выходного параметра для **businessJustification**. Можно использовать этот параметр, чтобы передать комментарии, предоставленные утверждающим во внешней системе в процессе подтверждения vRealize Automation. Эти комментарии могут касаться как подтверждений, так и отклонений.

### Создание рабочего процесса vRealize Orchestrator на основе схемы подтверждения

Создаваемый настраиваемый рабочий процесс подтверждения должен содержать входной параметр с любым подходящим именем и настроенным типом **Properties**. Полезные данные о событии подтверждения вставляются в этот параметр при инициировании запуска подписки на рабочий процесс.



Выходными параметрами рабочего процесса, отправляемыми обратно в vRealize Automation, являются `approved : Boolean` и `fieldValues : Properties`. Возвращаемый параметр `approved : Boolean` определяет уровень подтверждения — подтверждено или отклонено. Параметр `fieldValues : Properties` содержит значения, которые были изменены во внешней системе.

## Сценарий. Отправка запроса на программное обеспечение внешней системе для подтверждения

Администратору арендатора необходимо, чтобы пользователи за пределами vRealize Automation подтвердили компонент программного обеспечения, когда пользователь каталога служб запрашивает компьютер, содержащий программное обеспечение. Настройте политику подтверждения, для которой требуется подтверждение для всех подготовленных программ, и подписки на рабочий процесс, настроенной для запуска при получении сообщений о предварительном подтверждении, которые соответствуют определенным условиям.

### Процедура

#### 1. Сценарий. Создание рабочего процесса vRealize Orchestrator для подписки на рабочий процесс подтверждения

Создайте рабочий процесс vRealize Orchestrator, который принимает необходимые входные параметры подтверждения от vRealize Automation и возвращает необходимые выходные параметры для завершения процесса подтверждения.

#### 2. Сценарий. Создание политики подтверждения для внешнего подтверждения

Администратор арендатора создает политику подтверждения, которая генерирует сообщение о событии, опубликованном службой подтверждения. При настройке подписки на рабочий процесс с критериями, соответствующими сообщению о событии, vRealize Orchestrator запускает выбранный рабочий процесс.

#### 3. Сценарий. Создание подписки на рабочий процесс предварительного подтверждения

Администратор арендатора создает подписку на рабочий процесс предварительного подтверждения, который запускает рабочий процесс vRealize Orchestrator при формировании в запросе каталога служб запроса на подтверждение, который соответствует настроенным условиям.

## Сценарий. Создание рабочего процесса vRealize Orchestrator для подписки на рабочий процесс подтверждения

Создайте рабочий процесс vRealize Orchestrator, который принимает необходимые входные параметры подтверждения от vRealize Automation и возвращает необходимые выходные параметры для завершения процесса подтверждения.

Необходимо разработать рабочий процесс для достижения цели подтверждения. Дополнительные сведения по созданию папок и рабочих процессов vRealize Orchestrator см. в разделе *Разработка с помощью VMware vRealize Orchestrator*.

### Необходимые условия

Войдите в продукт vRealize Orchestrator, который является экземпляром, настроенным для vRealize Automation с привилегиями на создание рабочего процесса.

## Процедура

1. Создайте папку для рабочих процессов подписки на рабочий процесс в соответствующей библиотеке.
2. Создайте новый рабочий процесс.

Для этого сценария назовите рабочий процесс **Запрос подтверждения автоматизации**.

- а) Добавьте следующий входной параметр.

Имя	Тип
input	Properties

- б) Добавьте следующие выходные параметры.

Имя	Тип
approved	boolean
fieldValues	Properties

3. Создайте задачу с возможностью добавления сценариев, которая обрабатывает входные и выходные параметры.
4. Сохраните рабочий процесс.

## Следующие шаги

Создайте политику подтверждения, которая использует подписку на рабочий процесс в качестве утверждающего. [Сценарий. Создание политики подтверждения для внешнего подтверждения](#)

## Сценарий. Создание политики подтверждения для внешнего подтверждения

Администратор арендатора создает политику подтверждения, которая генерирует сообщение о событии, опубликованном службой подтверждения. При настройке подписки на рабочий процесс с критериями, соответствующими сообщению о событии, vRealize Orchestrator запускает выбранный рабочий процесс.

## Необходимые условия

- Войдите в службу vRealize Automation как **администратор арендатора** или **администратор подтверждения**.

## Процедура

1. Выберите **Администрирование > Политики подтверждения**.
2. Создайте политику подтверждения для компонентов программного обеспечения.
  - а) Выберите значок **Создать** (+).
  - б) Выберите **Выбрать тип политики подтверждения**.
  - в) В списке последовательно выберите **Каталог служб — Запрос на элемент каталога — Компонент ПО**.

- г) Нажмите кнопку **ОК**.
- д) Настройте следующие параметры.

Параметр	Конфигурация
Имя	Введите <b>Внешнее подтверждение ПО</b> .
Описание	Введите <b>Запрос на подтверждение отправлен внешней системе подтверждения.</b>
Состояние	Выберите значение <b>Active</b> .

- 3. На вкладке **Предварительное подтверждение** щелкните значок **Добавить (+)**.
- 4. Настройте критерии запуска и действия подтверждения на вкладке **Сведения об уровне**.
  - а) В текстовом поле **Имя** введите **Внешний уровень программного обеспечения**.
  - б) В текстовом поле **Описание** введите  
**Запрос подтверждения программного обеспечения отправлен внешней системе подтверждения.**
  - в) Выберите **Требуется всегда**.
  - г) Выберите **Использовать подписку на событие**.
- 5. Нажмите кнопку **ОК**.

#### Следующие шаги

- Создайте подписку на рабочий процесс предварительного подтверждения, который принимает сообщения о событиях на основе настроенного уровня подтверждения. См. раздел [Сценарий. Создание подписки на рабочий процесс предварительного подтверждения](#).
- Примените политику подтверждения к компоненту программного обеспечения через право. См. *Настройка vRealize Automation*.

#### Сценарий. Создание подписки на рабочий процесс предварительного подтверждения

Администратор арендатора создает подписку на рабочий процесс предварительного подтверждения, который запускает рабочий процесс vRealize Orchestrator при формировании в запросе каталога служб запроса на подтверждение, который соответствует настроенным условиям.

#### Необходимые условия

- Войдите в vRealize Automation в качестве **администратора арендатора**.
- Настройте для программного обеспечения уровень политики подтверждения с именем «Внешний уровень». См. [Сценарий. Создание политики подтверждения для внешнего подтверждения](#).
- Создайте настраиваемый рабочий процесс vRealize Orchestrator, который отправляет запрос внешней системе. В этом сценарии используйте рабочий процесс «Запрос подтверждения автоматизации».

## Процедура

1. Выберите **Администрирование > События > Подписки**

2. Выберите значок **Создать (+)**.

3. Щелкните **Предварительное подтверждение**.

4. Нажмите кнопку **Далее**.

5. На вкладке **Условия** настройте условия запуска.

а) Выберите **Запускать при определенных условиях**.

б) В раскрывающемся меню **Предложение** настройте следующее условие:

Свойство	Оператор	Значение
Данные > Информация о запросе > Имя уровня подтверждения	Равен	Внешний уровень ПО

в) Нажмите кнопку **Далее**.

6. На вкладке «Рабочий процесс» выберите дерево Orchestrator и укажите рабочий процесс **Запрос подтверждения автоматизации**.

7. Нажмите кнопку **Далее**.

8. На вкладке «Сведения» введите имя и описание.

В этом сценарии введите **Software External** в качестве имени и **Отправляет запрос подтверждения внешней системе** в качестве описания.

9. В текстовом поле **Время ожидания (мин.)** введите значение 120.

Указанное количество времени до окончания ожидания рабочего процесса подписки зависит от целевой системы. Если vRealize Automation не обрабатывает ответ от целевой системы за указанное количество минут, запрос автоматически отклоняется.

Если значение не указано, время ожидания составляет 24 часа.

10. Щелкните элемент **Готово**.

11. Выберите строку Software External и нажмите **Опубликовать**.

## Результаты

Подписка на событие предварительного подтверждения Software External теперь получает сообщения о событии предварительного подтверждения.

## Следующие шаги

- Если применяется политика подтверждения к компоненту программного обеспечения в активном праве, запросите элемент в каталоге служб и убедитесь в надлежащей работе политики подтверждения и подписки на рабочий процесс.

## Устранение неполадок подписок на рабочие процессы

Устранение неполадок подписок на рабочие процессы включает в себя некоторые распространенные проблемы. Кроме того, возможно, потребуется изучить различные журналы.

- **Устранение неполадок рабочих процессов vRealize Orchestrator, которые не запустились**

Настройте подписку на рабочий процесс для запуска настраиваемого рабочего процесса, когда сообщение о событии получено, но рабочий процесс не запускается.

- **Устранение неполадок в запросах на подготовку, занимающую слишком много времени**

Подготовка компьютера Инфраструктура как услуга занимает более десяти часов.

- **Устранение неполадок в рабочих процессах vRealize Orchestrator, не запускающихся для запроса подтверждения**

Настроена подписка на рабочий процесс предварительного или последующего подтверждения для запуска рабочего процесса vRealize Orchestrator. Рабочий процесс не запускается, если компьютер, соответствующий определенным критериям, запрашивается в каталоге служб.

- **Устранение неполадок отклоненных запросов подтверждения, которые должны быть подтверждены**

Настройте подписку на рабочий процесс предварительного или последующего подтверждения, которая запускает указанный рабочий процесс vRealize Orchestrator, но запрос отклонен, хотя известно, что он был одобрен.

- **Устранение неполадок отклоненного запроса подтверждения**

Вы настраиваете подписку на рабочий процесс предварительного или последующего подтверждения, который запускает указанный рабочий процесс vRealize Orchestrator, но запрос непредвиденно отклонен.

## Устранение неполадок рабочих процессов vRealize Orchestrator, которые не запустились

Настройте подписку на рабочий процесс для запуска настраиваемого рабочего процесса, когда сообщение о событии получено, но рабочий процесс не запускается.

### Решение

1. Убедитесь, что подписка на рабочий процесс опубликована.
2. Убедитесь, что условия подписки на рабочий процесс настроены правильно.
3. Убедитесь в наличии указанного рабочего процесса на сервере vRealize Orchestrator.

## Устранение неполадок в запросах на подготовку, занимающую слишком много времени

Подготовка компьютера Инфраструктура как услуга занимает более десяти часов.

### Причина

Если для подписки на рабочий процесс в качестве фактора, запускающего выполнение, выбрано состояние подготовки, в среде могут быть запущены два экземпляра службы диспетчера инфраструктуры как услуги.

## Решение

- ◆ Убедитесь, что активен только один экземпляр службы диспетчера инфраструктуры как услуги. При наличии более одного активного экземпляра в журнале появятся ошибки такого типа:

```
[EventBrokerService] Failed resuming workflow b6e9276a-f20f-40f1-99ad-6d9524560cc2 on queue
3679fa71-ac2a-42d5-8626-f98ea096f0d3. System.Workflow.Runtime.QueueException. Операция создания
очереди события завершается ошибкой MessageQueueErrorCode QueueNotFound для очереди 3679fa71-
ac2a-42d5-8626-f98ea096f0d3. в
System.Workflow.Runtime.WorkflowQueueingService.EnqueueEvent(IComparable queueName, Object item) в
System.Workflow.Runtime.WorkflowExecutor.EnqueueItem(IComparable queueName, Object item,
IPendingWork pendingWork, Object workItem) в
System.Workflow.Runtime.WorkflowInstance.EnqueueItem(IComparable queueName, Object item,
IPendingWork pendingWork, Object workItem) в
DynamicOps.VMPS.Service.Workflow.Services.EventBrokerService.OnMessage(EventObject obj)
[UTC:2015-11-14 07:14:25 Local:2015-11-13 23:14:25] [Error]: Thread-Id="15" - context="HKBsp6Tt"
token="JeuTG7ru" [EventBrokerClient] Не удалось выполнить обратный вызов подписки. Операция очереди
событий завершается ошибкой MessageQueueErrorCode QueueNotFound для очереди 3679fa71-ac2a-42d5-8626-
f98ea096f0d3.
```

## Устранение неполадок в рабочих процессах vRealize Orchestrator, не запускающихся для запроса подтверждения

Настроена подписка на рабочий процесс предварительного или последующего подтверждения для запуска рабочего процесса vRealize Orchestrator. Рабочий процесс не запускается, если компьютер, соответствующий определенным критериям, запрашивается в каталоге служб.

### Причина

Для успешного выполнения подписки на рабочий процесс для подтверждения необходимо убедиться, что все компоненты правильно настроены.

### Решение

1. Убедитесь, что политика подтверждения активна и для уровня подтверждения в ней установлен флажок **Использовать подписку на событие**.
2. Убедитесь, что политика подтверждения правильно применена в праве.
3. Убедитесь, что подписка на рабочий процесс правильно настроена и опубликована.
4. Проверьте в журналах событий сообщения, относящиеся к подтверждениям.

## Устранение неполадок отклоненных запросов подтверждения, которые должны быть подтверждены

Настройте подписку на рабочий процесс предварительного или последующего подтверждения, которая запускает указанный рабочий процесс vRealize Orchestrator, но запрос отклонен, хотя известно, что он был одобрен.

## Решение

1. Просмотрите рабочий процесс в vRealize Orchestrator.
  - а) Войдите в vRealize Orchestrator с привилегиями администратора.
  - б) Убедитесь, что рабочий процесс прошел без ошибок.
  - в) Убедитесь, что для параметров `approval` и `fieldValues` возвращены ожидаемые значения.
2. Просмотрите запрос в vRealize Automation.
  - а) Войдите в vRealize Automation как пользователь, который запросил отклоненный элемент.
  - б) Откройте вкладку **Развертывание**.
  - в) Щелкните имя отклоненного запроса на развертывание и нажмите кнопку **Журнал**.
  - г) Нажмите данное состояние и просмотрите табличку с дополнительной информацией.

При возникновении ошибки информация об ошибке отобразится в столбце «Обоснование».

## Устранение неполадок отклоненного запроса подтверждения

Вы настраиваете подписку на рабочий процесс предварительного или последующего подтверждения, который запускает указанный рабочий процесс vRealize Orchestrator, но запрос непредвиденно отклонен.

### Проблема

Были получены подтверждения на всех уровнях вплоть до этого внешнего уровня, на котором вместо подтверждения запроса произошло его отклонение.

### Причина

Одной из возможных причин является внутренняя ошибка при попытке vRealize Orchestrator запустить рабочий процесс. Например, отсутствует рабочий процесс или не запущен сервер vRealize Orchestrator.

## Решение

1. Выберите **Администрирование > События > Журналы событий**.
2. Проверьте в журналах сообщения, относящиеся к подтверждениям.

## Расширение жизненных циклов компьютеров за счет использования vRealize Automation Designer

Можно вставить настраиваемую логику на заранее определенных этапах жизненного цикла компьютера инфраструктуры как услуги с помощью vRealize Automation Designer для непосредственного

редактирования заглушек рабочих процессов изменения состояния и при необходимости вызова настраиваемых рабочих процессов vRealize Orchestrator.

**Примечание** Заглушки рабочих процессов заменяются подписками на рабочие процессы брокера событий. Они остаются доступны, сохраняется поддержка и возможность использования, но в следующей версии vRealize Automation планируется их удаление. Чтобы обеспечить совместимость последующих продуктов, используйте подписки на рабочий процесс для запуска настраиваемых рабочих процессов с примененными изменениями. См. раздел [Настройка подписок на рабочие процессы для расширения возможностей vRealize Automation](#).

## Расширение жизненных циклов компьютеров за счет использования контрольного списка vRealize Automation Designer

Расширение жизненных циклов компьютера за счет использования контрольного списка vRealize Automation Designer обеспечивает общий обзор этапов, необходимых для установки и конфигурации vRealize Automation Designer для настройки жизненных циклов компьютера инфраструктуры как услуги.

Таблица 1-17. Расширение жизненных циклов компьютеров за счет использования контрольного списка vRealize Automation Designer

Задача	Сведения
<input type="checkbox"/> Загрузить и установить vRealize Automation Designer.	<a href="#">Установка vRealize Automation Designer</a>
<input type="checkbox"/> Создать конечную точку vRealize Automation для экземпляра vRealize Orchestrator.	<a href="#">Создание конечной точки vRealize Orchestrator</a>
<input type="checkbox"/> Связать конечную точку vRealize Orchestrator со схемой элементов компьютера.	<a href="#">Связывание конечной точки vRealize Orchestrator со схемой элементов</a>



Таблица 1-17. Расширение жизненных циклов компьютеров за счет использования контрольного списка vRealize Automation Designer (продолжение)

Задача	Сведения
<p><input type="checkbox"/> Настроить заглушку рабочего процесса инфраструктуры как услуги с помощью действий vRealize Automation Designer.</p> <hr/> <p><b>Примечание</b> Заглушки рабочих процессов заменяются подписками на рабочие процессы брокера событий. Они остаются доступны, сохраняется поддержка и возможность использования, но в следующей версии vRealize Automation планируется их удаление. Чтобы обеспечить совместимость последующих продуктов, используйте подписки на рабочий процесс для запуска настраиваемых рабочих процессов с примененными изменениями. См. раздел <a href="#">Настройка подписок на рабочие процессы для расширения возможностей vRealize Automation</a>.</p> <hr/> <p>Можно использовать действия рабочего процесса vRealize Orchestrator для вызова настраиваемых рабочих процессов vRealize Orchestrator (дополнительно).</p>	<p><a href="#">Настройка рабочего процесса инфраструктуры как услуги</a></p>
<p><input type="checkbox"/> После создания настраиваемого рабочего процесса изменения состояния администратор арендатора или диспетчер бизнес-группы должен включить его в определенную схему элементов путем добавления настраиваемого свойства.</p>	<p><a href="#">Настройка схемы элементов для вызова рабочего процесса изменения состояния</a></p>

## Установка и настройка vRealize Automation Designer;

Можно установить vRealize Automation Designer на компьютере под управлением Windows и настроить его для обмена данными с удаленным экземпляром диспетчера моделей. Если вы используете рабочие процессы инфраструктуры как услуги для вызова рабочих процессов vRealize Orchestrator, нужно настроить экземпляр vRealize Orchestrator в качестве конечной точки.

### Установка vRealize Automation Designer

Можно установить vRealize Automation Designer на компьютере под управлением Windows и настроить его для обмена данными с удаленным экземпляром диспетчера моделей.

#### Необходимые условия для vRealize Automation Designer

vRealize Automation Designer обычно устанавливается на компьютере для разработки, а не на сервере.

#### Поддерживаемые операционные системы

Поддерживаемые операционные системы для vRealize Automation Designer указаны в *Матрица поддержки vRealize Automation* на странице документации VMware vRealize Automation.

## Требования к конфигурации системы

Чтобы узнать о потенциальных обновлениях этой информации, см. *Матрица поддержки vRealize Automation* для своей версии vRealize Automation.

- Необходимо установить .NET Framework 4.5.
- Узел vRealize Automation Designer должен иметь сетевой доступ к компонентам веб-сайта IaaS (а именно к веб-компоненту диспетчера моделей).
- Если диспетчер моделей установлен удаленно, к сертификату, используемому для веб-компонента диспетчера моделей, должно быть установлено доверие на узле vRealize Automation Designer.

## Загрузка средства установки vRealize Automation Designer

Можно загрузить программу установки vRealize Automation Designer с устройства vRealize Automation.

### Необходимые условия

- Войдите на компьютер под управлением Windows в качестве локального администратора.
- Если для установки используется Internet Explorer, убедитесь, что конфигурация усиленной безопасности не включена. См. <res://iesetup.dll/SoftAdmin.htm>.

### Процедура

1. Откройте браузер.
2. Перейдите на страницу загрузки программы установки Windows, используя имя узла (<https://vra-va-hostname.domain.name:5480/installer/>).
3. Выберите **vRealize Automation Designer**.
4. В ответ на запрос сохраните программу установки.

### Следующие шаги

[Установка vRealize Automation Designer.](#)

## Установка vRealize Automation Designer

Программа установки vRealize Automation Designer упаковывается как мастер установки Windows.

### Необходимые условия

[Загрузка средства установки vRealize Automation Designer.](#)

### Процедура

1. Перейдите в каталог, где находится загруженная программа установки.
2. Щелкните правой кнопкой мыши файл DesignCenter-Setup.exe и выберите **Запустить от имени администратора**.
3. На **странице приветствия** нажмите кнопку **Далее**.

4. Прочитайте лицензионное соглашение, установите флажок **Я принимаю условия лицензионного соглашения** и нажмите кнопку **Далее**.
5. На странице **Выборочная установка** нажмите кнопку **Далее**.
6. Укажите полное доменное имя и порт веб-экземпляра диспетчера моделей в формате *hostname:port*.  
По умолчанию используется порт 443.
7. Укажите учетные данные службы диспетчера моделей.
8. Нажмите кнопку **Далее**.  
Программа установки проверяет комбинацию узла диспетчера моделей и учетные данные, пытаясь получить доступ к этому диспетчеру. Если возвращается ошибка, необходимо обеспечить правильную комбинацию узла диспетчера моделей и учетных данных, прежде чем продолжить.
9. Щелкните элемент **Установить**.
10. Щелкните элемент **Готово**.

#### Следующие шаги

Можно запустить vRealize Automation Designer через меню «Пуск» ОС Windows, выполнив переход в каталог установки.

### Настройка конечных точек vRealize Orchestrator

Если вы используете рабочие процессы vRealize Automation для вызова рабочих процессов vRealize Orchestrator, нужно настроить экземпляр или сервер vRealize Orchestrator в качестве конечной точки.

Конечную точку vRealize Orchestrator можно связать со схемой элементов нескольких компьютеров, чтобы убедиться, что для запуска всех рабочих процессов vRealize Orchestrator компьютеров, подготовленных с использованием этой схемы элементов, применена эта конечная точка.

vRealize Automation содержит встроенный экземпляр vRealize Orchestrator по умолчанию. Рекомендуется использовать встроенный экземпляр в качестве конечной точки vRealize Orchestrator для выполнения рабочих процессов vRealize Automation в производственной или тестовой среде или создания пилотной версии.

Можно также установить подключаемый модуль на внешнем сервере vRealize Orchestrator, однако этот способ не рекомендуется для производственных сред.

#### Необходимые условия интеграции vRealize Orchestrator

Если для выполнения рабочих процессов vRealize Orchestrator, которые имеют параметры ввода или вывода типа VC:VirtualMachine, используются рабочие процессы vRealize Automation, убедитесь в наличии рабочих процессов vRealize Orchestrator для преобразования типов виртуальных машин между vRealize Orchestrator и инфраструктурой как услугой.

Необходимые рабочие процессы включены по умолчанию в vRealize Orchestrator 5.5 и более поздних версий в рамках подключаемого модуля vCenter.

Если используется vRealize Orchestrator 5.1, установите пакет интеграции vRealize Automation для vRealize Orchestrator. Загрузите пакет `com.vmware.library.vcenter.vcac-integration.package` с веб-сайта сообщества vRealize Orchestrator по адресу <https://communities.vmware.com/t5/vRealize-Orchestrator-Documents/vCloud-Automation-Center-integration-package/ta-p/2777982>.

Импортируйте пакет на каждый сервер vRealize Orchestrator, который настроен в качестве конечной точки в IaaS.

Подробнее об импорте пакетов в vRealize Orchestrator см. в документации vRealize Orchestrator.

## Создание конечной точки vRealize Orchestrator

Можно создать конечную точку vRealize Orchestrator для подключения к серверу vRealize Orchestrator.

Вы можете настроить несколько конечных точек, чтобы подключиться к разным серверам vRealize Orchestrator, однако для каждой конечной точки нужно задать приоритет.

При выполнении рабочих процессов vRealize Orchestrator vRealize Automation сначала использует конечную точку vRealize Orchestrator с наивысшим приоритетом. Если к ней невозможно подключиться, то устройство переходит к следующей конечной точке с наивысшим приоритетом, пока сервер vRealize Orchestrator не сможет выполнить рабочий процесс.

### Необходимые условия

- Войдите в службу vRealize Automation в качестве **администратора инфраструктуры как услуги**.

### Процедура

1. Выберите **Инфраструктура > Конечные точки > Конечные точки**.
2. Выберите **Создать > Оркестрация > vRealize Orchestrator**.
3. Введите имя и, при необходимости, описание.
4. Введите URL-адрес с полным квалифицированным именем или IP-адрес сервера vRealize Orchestrator и номер порта vRealize Orchestrator.

Следует использовать транспортный протокол HTTPS. Если порт не указан, используется порт по умолчанию 443.

Чтобы использовать экземпляр vRealize Orchestrator по умолчанию, который содержится в устройстве vRealize Automation, введите

**`https://vrealize-automation-appliance-hostname:443/vco`**.

5. Укажите свои учетные данные vRealize Orchestrator в текстовых полях **Имя пользователя** и **Пароль**, чтобы подключиться к конечной точке vRealize Orchestrator.

Используемые учетные данные должны иметь разрешения на выполнение любых рабочих процессов vRealize Orchestrator, которые можно вызвать из Инфраструктура как услуга

Чтобы использовать экземпляр vRealize Orchestrator по умолчанию, содержащийся в устройстве vRealize Automation, используйте имя пользователя **`administrator@vsphere.local`** и пароль администратора, указанный при настройке службы единого входа.

6. В текстовом поле **Свойство** введите целое число не меньше 1.

Чем меньше значение, тем выше приоритет.

7. (дополнительно) Выберите элемент **Свойства** и добавьте стандартные настраиваемые свойства, группы свойств или собственные определения свойств для конечной точки.
8. Нажмите кнопку **ОК**.

### Связывание конечной точки vRealize Orchestrator со схемой элементов

Можно указать определенную конечную точку vRealize Orchestrator для использования со схемой элементов.

В инфраструктуре как услуге при запуске рабочего процесса vRealize Orchestrator для любого компьютера, подготовленного из этой схемы элементов, всегда используется связанная конечная точка. Если конечная точка недостижима, рабочий процесс завершается ошибкой.


#### Необходимые условия

Войдите в службу vRealize Automation как **архитектор инфраструктуры**.

#### Процедура

1. Выберите **Проектирование > Схемы элементов**.
2. Создайте новую схему элементов или измените существующую.

При редактировании существующей схемы элементов конечная точка vRealize Orchestrator указывает на применение только новых компьютеров, подготовленных из обновленной схемы. Существующие компьютеры, подготовленные из схемы элементов, по-прежнему используют конечную точку с высшим приоритетом, если это свойство не добавлено вручную к компьютеру.

3. Нажмите значок **Свойства схемы элементов** (  ).
4. Перейдите на вкладку **Свойства**.
  - а) Нажмите в меню **Настраиваемое свойство > Создать**.
  - б) Введите значение **VMware.VCenterOrchestrator.EndpointName** в текстовом поле **Имя**.

Имя свойства задается с учетом регистра символов.
  - в) Нажмите кнопку **ОК**, чтобы сохранить данное свойство.
5. Нажмите кнопку **ОК**.

## Настройка рабочих процессов инфраструктуры как услуги с помощью vRealize Automation Designer

VMware предоставляет ряд рабочих процессов, которые можно настроить с помощью vRealize Automation Designer. К ним относятся рабочие процессы изменений состояния и операций меню.

Рабочие процессы инфраструктуры как услуги создаются с помощью Microsoft Windows Workflow Foundation 4, составной части .NET Framework 4. Дополнительные сведения о Windows Workflow Foundation и создании рабочих процессов см. в документации Microsoft. vRealize Automation также предоставляет некоторые действия vRealize Automation Designer для запуска и мониторинга рабочих процессов vRealize Orchestrator.

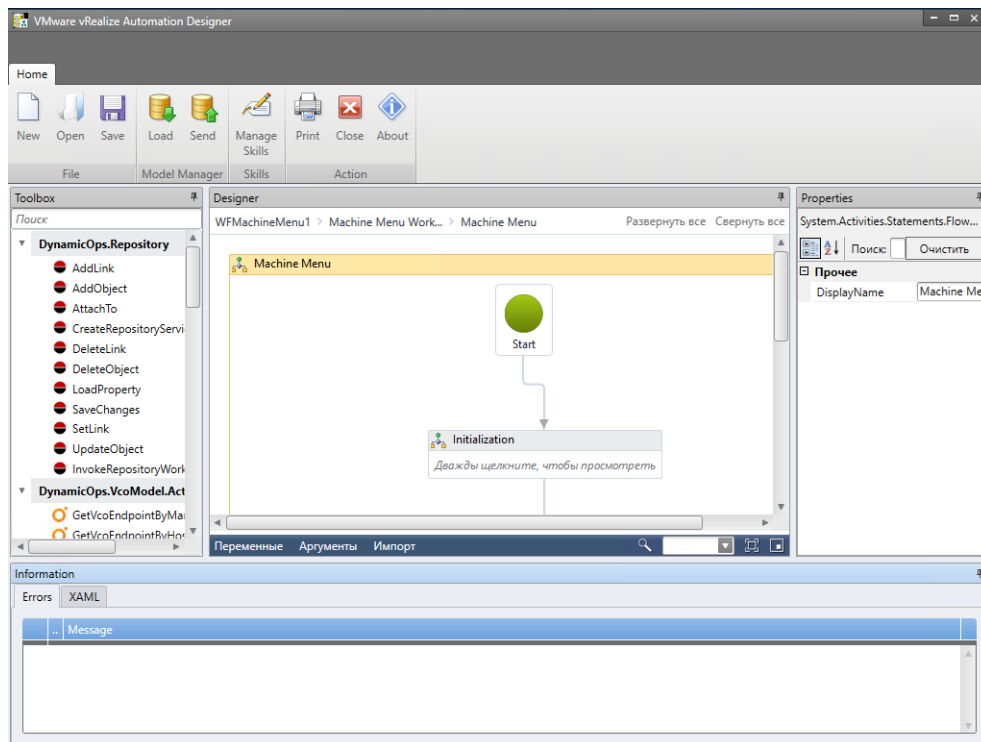
В настраиваемых шаблонах рабочих процессов, предоставляемых VMware, отражены практические рекомендации по структурированию рабочих процессов с отдельными последовательностями для инициализации, настраиваемой логики и завершения. Весь рабочий процесс упакован в блок TryCatch для обработки ошибок. Любые неперехваченные или переброшенные исключения регистрируются диспетчером Distributed Execution Manager, который выполняет рабочий процесс.

После создания настраиваемого рабочего процесса инфраструктуры как услуги автор схемы элементов должен включить рабочий процесс в определенные схемы.

## Консоль vRealize Automation Designer

В консоли vRealize Automation Designer предоставляется визуальный редактор рабочих процессов для настройки рабочих процессов IaaS.

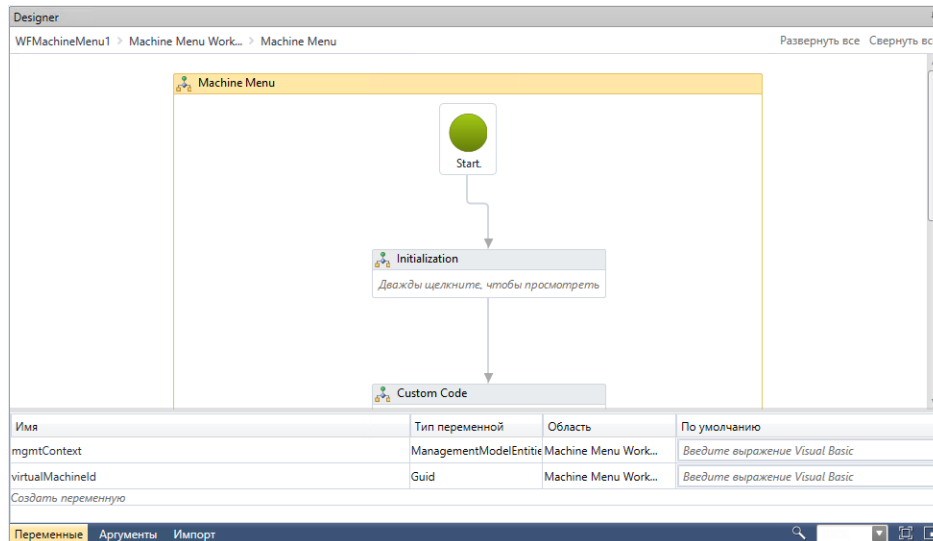
Для запуска консоли vRealize Automation Designer необходимо обладать правами локального администратора на узле vRealize Automation Designer (обычно это компьютер для разработки).



В области «Панель элементов» слева можно получить доступ к библиотеке действий рабочих процессов vRealize Automation. Действия с панели элементов можно перетащить в область «Конструктор», чтобы добавить их в рабочий процесс. В области «Свойства» отображаются настраиваемые свойства действия, выбранного в настоящее время в области «Конструктор». Данный интерфейс очень похож на конструктор рабочих процессов в Visual Studio.

Вкладки с подробными сведениями в нижней части области «Конструктор» позволяют отображать и редактировать переменные в диапазоне выбранного действия или аргументов к нему.

**Примечание** Переменные и аргументы указываются как выражения Visual Basic. Однако в названиях переменных не учитывается регистр, в то время как в названиях аргументов регистр учитывается. Подробнее о допустимых аргументах для действий рабочих процессов IaaS см. в разделе [Справка по действиям рабочих процессов vRealize Automation](#).



На вкладке «Импорт» отображаются импортированные пространства имен, из которых можно выбрать типы сущности для добавления к рабочему процессу.

В сворачиваемой области «Информация» в нижней части консоли отображаются все ошибки в настройке действий. В этой области можно получить доступ к представлению XAML рабочего процесса.

## Типы рабочих процессов инфраструктуры как услуги

Можно настроить два типа рабочих процессов с помощью vRealize Automation Designer — рабочие процессы изменения состояния и рабочие процессы операций меню.

- Рабочий процесс изменения состояния выполняется при изменении состояния основного рабочего процесса, например на определенном этапе во время подготовки нового компьютера.
- Рабочий процесс операции меню выполняется, когда пользователь выбирает параметр в меню «Действия» каталога служб или в меню компьютеров на вкладке «Инфраструктура».

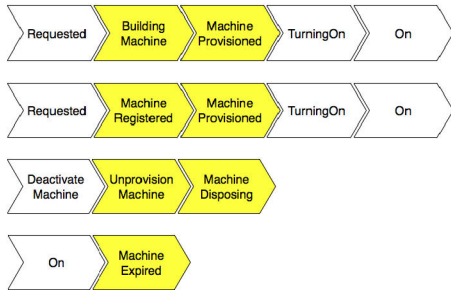
### Рабочие процессы изменения состояния

Создание рабочего процесса изменения состояния позволяет запустить рабочий процесс до того, как основной рабочий процесс инфраструктуры как услуги перейдет в определенное состояние.

Например, можно создать настраиваемые рабочие процессы для интеграции с внешней базой данных и записи информацию на различных этапах жизненного цикла компьютера.

- Создание настраиваемого рабочего процесса, который выполняется до того, как основной рабочий процесс перейдет в состояние **MachineProvisioned**, для записи такой информации, как владелец компьютера, утверждающие и т. д.
- Создание настраиваемого рабочего процесса, который выполняется до того, как компьютер перейдет в состояние **MachineDisposing**, для записи времени его уничтожения и таких данных, как использование его ресурсов при последнем сборе данных, входе в систему и т. д.

На следующих рисунках показаны базовые состояния основного рабочего процесса.



vRealize Automation Designer предоставляет настраиваемый рабочий процесс для каждого из этих состояний.

Таблица 1-18. Настраиваемые рабочие процессы изменения состояния

Состояние основного рабочего процесса	Имя настраиваемого рабочего процесса
BuildingMachine	WFStubBuildingMachine
Disposing	WFStubMachineDisposing
Expired	WFStubMachineExpired
MachineProvisioned	WFStubMachineProvisioned
RegisterMachine	WFStubMachineRegistered
UnprovisionMachine	WFStubUnprovisionMachine

### Обзор настройки рабочего процесса изменения состояния

Можно настроить рабочий процесс изменения состояния с помощью vRealize Automation Designer. Автор схемы элементов может затем включить ее для определенных схем.

Ниже приведен общий обзор этапов, необходимых для включения рабочих процессов изменения состояния.

1. Разработчик рабочего процесса настраивает один из шаблонов процесса изменения состояния с помощью vRealize Automation Designer. См. [Настройка рабочего процесса инфраструктуры как услуги](#).

Любой рабочий процесс инфраструктуры как услуги может вызвать рабочий процесс vRealize Orchestrator. Дополнительные сведения см. в статье [Использование действий рабочего процесса vRealize Orchestrator](#).



2. Администратор арендатора или диспетчер бизнес-группы настраивает схему элементов, чтобы вызвать настраиваемый рабочий процесс для компьютеров, подготовленных в этой схеме. См. [Настройка схемы элементов для вызова рабочего процесса изменения состояния](#).

## Рабочие процессы операций меню

Рабочий процесс операции меню выполняется, когда пользователь выбирает параметр меню «Действия» каталога служб или меню компьютеров на вкладке «Инфраструктура».

Например, можно создать настраиваемый рабочий процесс, который позволяет пользователю создавать билет поддержки, связанный с компьютером, путем выбора пункта «Сообщить о проблеме в службу поддержки» в меню компьютера.

vRealize Automation Designer предоставляет шаблоны для настройки рабочих процессов операций меню.

Помимо определения, рабочий процесс операции меню зависит от файла конфигурации операции, в котором определены аспекты настраиваемого параметра меню, например, текст отображения ролей, которым предоставляется доступ, и состояния компьютера, в которых доступна данная операция.

---

**Примечание** Разработчик Все как услуга может определить настраиваемые действия для любого элемента каталога с помощью Все как услуга. Для создания настраиваемых действий для компьютеров инфраструктуры как услуги, отличающихся от компьютеров, подготовленных с помощью vSphere или vCloud Director, требуется vRealize Automation 6.1 или более поздней версии.

---

## Настройка обзора рабочих процессов операций меню

Можно настроить рабочий процесс операции меню с помощью vRealize Automation Designer и служебной программы командной строки CloudUtil. Автор схемы элементов может затем включить ее для определенных схем.

Ниже приведен общий обзор этапов, необходимых для включения рабочих процессов операций меню.

1. Разработчик рабочего процесса настраивает один из шаблонов процессов операций меню с помощью vRealize Automation Designer. См. [Настройка рабочего процесса инфраструктуры как услуги](#).  
Любой рабочий процесс инфраструктуры как услуги может вызвать рабочий процесс vRealize Orchestrator. Дополнительные сведения см. в статье [Использование действий рабочего процесса vRealize Orchestrator](#).
2. Разработчик рабочего процесса настраивает операцию меню в диспетчере моделей. См. [Настройка операции меню](#).
3. Разработчик рабочего процесса регистрирует новую операцию меню с помощью каталога служб. См. [Регистрация новых операций меню с помощью каталога служб](#).
4. Администратор арендатора или диспетчер бизнес-группы настраивает схему элементов, чтобы включить операцию меню для компьютеров, подготовленных в этой схеме. См. [Настройка схемы элементов для включения рабочего процесса операций меню](#).

Если операция меню предназначена для использования в каталоге служб, необходимо предоставить соответствующие права пользователям. Дополнительные сведения см. в разделе *Администрирование арендаторов*.

## Настройка рабочего процесса инфраструктуры как услуги

vRealize Automation Designer позволяет редактировать настраиваемые рабочие процессы и рабочие процессы обновления в диспетчере моделей.

### Необходимые условия

Запустите vRealize Automation Designer.

### Процедура

1. Нажмите **Загрузить**.
2. Выберите рабочий процесс, который необходимо настроить.

Параметр	Описание
<b>WFMachineMenuN</b>	Настраиваемый рабочий процесс операции меню
<b>WFStubBuildingMachine</b>	Настраиваемый рабочий процесс изменения состояния, который выполняется перед переходом компьютера в состояние BuildingMachine
<b>WFStubMachineDisposing</b>	Настраиваемый рабочий процесс изменения состояния, который выполняется перед переходом компьютера в состояние Disposing
<b>WFStubMachineExpired</b>	Настраиваемый рабочий процесс изменения состояния, который выполняется перед переходом компьютера в состояние Expired
<b>WFStubMachineProvisioned</b>	Настраиваемый рабочий процесс изменения состояния, который выполняется перед переходом компьютера в состояние MachineProvisioned
<b>WFStubMachineRegistered</b>	Настраиваемый рабочий процесс изменения состояния, который выполняется перед переходом компьютера в состояние RegisterMachine
<b>WFStubUnprovisionMachine</b>	Настраиваемый рабочий процесс изменения состояния, который выполняется перед переходом компьютера в состояние UnprovisionMachine

3. Нажмите кнопку **OK**.

Рабочий процесс отображается в области конструктора.

4. Настройте рабочий процесс путем перетаскивания действий из панели элементов в область конструктора и выполните конфигурацию их аргументов.
5. После редактирования рабочего процесса обновите его в диспетчере моделей, нажав кнопку **Отправить**.

Рабочий процесс сохраняется и отображается в виде новой редакции в списке при следующей загрузке рабочего процесса. В любое время можно получить доступ к более ранней версии рабочего процесса. См. [Возврат к предыдущей редакции рабочего процесса](#).

## Использование действий рабочего процесса vRealize Orchestrator

Можно использовать действия vRealize Automation Designer для синхронного и асинхронного вызова рабочих процессов vRealize Orchestrator.

Конечная точка vRealize Orchestrator указана одним из следующих способов:

- `VirtualMachineId` — имя переменной, представляющей идентификатор виртуальной машины. Выбирается виртуальная машина с этим идентификатором, а значение, которое извлекается из настраиваемого свойства `VMware.VCenterOrchestrator.EndpointName` для виртуальной машины, используется в качестве имени конечной точки vRealize Orchestrator.
- `GetVcoEndpointByManagementEndpoint` возвращает значение настраиваемого свойства определенному объекту `ManagementEndpoint`. Если значение `CustomPropertyName` не задано, используется значение свойства `VMware.VCenterOrchestrator.EndpointName`.
- `GetVcoEndpointByHost` возвращает значение настраиваемого свойства определенному узлу. Если значение `CustomPropertyName` не задано, используется значение свойства `VMware.VCenterOrchestrator.EndpointName`.

### Синхронное выполнение

Действие `InvokeVcoWorkflow` вызывает рабочий процесс vRealize Orchestrator и блокирует дальнейшее выполнение его родительского рабочего процесса инфраструктуры как услуги до завершения рабочего процесса vRealize Orchestrator. Действие возвращает выходной параметр для рабочего процесса vRealize Orchestrator.

Кроме того, синхронный режим поддерживает следующее свойство:

- `WorkflowTimeout` — значение времени ожидания в секундах. Если рабочий процесс vRealize Orchestrator не завершен в течение указанного времени, генерируется исключение, но рабочий процесс не блокируется, пока ответ не возвращен. Если значение не определено или равняется нулю, время ожидания не будет активировано. Состояние рабочего процесса проверяется каждые 10 секунд в течение этого периода, если время опроса для конечной точки не изменяется на указанное значение в настраиваемом свойстве `VMware.VCenterOrchestrator.PollingInterval`.

### Асинхронное выполнение рабочих процессов

Действие `InvokeVcoWorkflowAsync` вызывает рабочий процесс vRealize Orchestrator и продолжает выполнять действия в рабочем процессе инфраструктуры как услуги, не дожидаясь завершения рабочего процесса vRealize Orchestrator.

Действие возвращает либо уникальный маркер рабочего процесса, который можно использовать для мониторинга рабочего процесса, либо ошибку, если не удалось выполнить вызов REST API, направленный на сервер vRealize Orchestrator (например, если не удастся связаться с сервером).

В этом действии доступны два дополнительных действия:

- `GetVcoWorkflowExecutionStatus` позволяет опросить рабочий процесс vRealize Orchestrator, чтобы узнать его статус.

- `WaitForVcoWorkflowCompletion` позволяет блокировать дальнейшее выполнение рабочего процесса инфраструктуры как услуги, пока рабочий процесс `vRealize Orchestrator` не будет завершен или не истечет время его ожидания. Это действие можно использовать, чтобы получить результаты рабочего процесса `vRealize Orchestrator`, выполняемого асинхронно.

## Вызов рабочего процесса `vRealize Orchestrator`

Можно использовать действие `InvokeVcoWorkflow` или `InvokeVcoWorkflowAsync` для вызова рабочего процесса `vRealize Orchestrator` из рабочего процесса инфраструктуры как услуги.

Для некоторых рабочих процессов `vRealize Orchestrator` требуется действие пользователя. Для этих рабочих процессов запрос пользователя появляется в клиенте `vRealize Orchestrator`, а не консоли `vRealize Automation`, поэтому конечному пользователю не видно в `vRealize Automation`, что рабочий процесс ожидает ввода.

Чтобы избежать рабочих процессов, блокирующих пользовательский ввод, не вызывайте рабочие процессы `vRealize Orchestrator`, требующие действия пользователя из рабочих процессов инфраструктуры как услуги.

## Процедура

1. В `vRealize Automation Designer` откройте рабочий процесс и перейдите к контексту, где необходимо вызвать рабочий процесс `vRealize Orchestrator`.
2. Перетащите действие `InvokeVcoWorkflow` или `InvokeVcoWorkflowAsync` в область конструктора.
3. Выберите рабочий процесс `vCenter Orchestrator` для выполнения.

- а) На вкладке «Общие» щелкните многоточие рядом с полем «Рабочий процесс».
- б) В диалоговом окне «Выбор рабочего процесса `vCO`» выберите рабочий процесс.
- в) Нажмите кнопку **ОК**.

В разделах «Ввод» и «Вывод» отображаются входные и выходные параметры выбранного процесса.

4. В области свойств задайте один из следующих целевых параметров.
  - `VirtualMachineId` — имя переменной, представляющей идентификатор виртуальной машины. Выбирается виртуальная машина с этим идентификатором, а значение, которое извлекается из настраиваемого свойства `VMware.VCenterOrchestrator.EndpointName` для виртуальной машины, используется в качестве имени конечной точки `vRealize Orchestrator`.
  - `VcoEndpointName` — имя конечной точки, используемой для запуска рабочего процесса. Если задано, это значение переопределяет значение `VirtualMachineId` при выборе конечной точки `vRealize Orchestrator`.
  - `WorkflowTimeout` — значение времени ожидания в секундах. Если рабочий процесс `vRealize Orchestrator` не завершен в течение указанного времени, генерируется исключение, но рабочий процесс не блокируется, пока ответ не возвращен. Если значение не определено или равняется нулю, время ожидания не будет активировано. Состояние рабочего процесса проверяется каждые 10 секунд в течение этого периода, если время опроса для конечной точки не изменяется на указанное значение в настраиваемом свойстве `VMware.VCenterOrchestrator.PollingInterval`.

## 5. Укажите параметры для рабочего процесса vRealize Orchestrator.

- Введите значения для действия в области конструктора.
- В области свойств щелкните многоточие рядом со значением **InputParameters** или **OutputParameters**, чтобы открыть диалоговое окно «Параметры». В этом диалоговом окне отображается тип инфраструктуры как услуги каждого параметра. Выделенный жирным тип параметра является обязательным.

Наведите курсор на текстовое поле для любого параметра, чтобы просмотреть всплывающую подсказку, указывающую тип vRealize Orchestrator.

При использовании действия InvokeVcoWorkflowAsync выходные параметры рабочего процесса vRealize Orchestrator отображаются с соответствующими типами для информационных целей, но для параметра в этой деятельности нельзя указать выражение.

### Следующие шаги

Чтобы получить результаты рабочего процесса, выполняемого асинхронно, выполните действие WaitForVcoWorkflowCompletion.

### Получение состояния рабочего процесса vRealize Orchestrator

Можно проверить состояние рабочего процесса vRealize Orchestrator, вызванного с помощью действия InvokeVcoWorkflowAsync, используя действие GetVcoWorkflowExecutionStatus.

### Необходимые условия

[Вызов рабочего процесса vRealize Orchestrator](#) использование действия InvokeVcoWorkflowAsync.

### Процедура

1. В vRealize Automation Designer откройте рабочий процесс, в котором использовалось действие InvokeVcoWorkflowAsync.
2. Перейдите к контексту, где необходимо проверить состояние рабочего процесса vRealize Orchestrator.
3. Перетащите действие GetVcoWorkflowExecutionStatus в область конструктора.
4. В области свойств задайте имя переменной, представляющей идентификатор виртуальной машины в VirtualMachineId..  
  
Настраиваемые рабочие процессы содержат переменную по умолчанию с именем virtualMachineId, которое устанавливается во время инициализации.
5. Создайте переменную типа DynamicOps.VcoModel.Common.VcoWorkflowExecutionToken.
6. Укажите имя переменной маркера в качестве выходного параметра executionToken в действии InvokeVcoWorkflowAsync.
7. Укажите то же самое имя переменной как для свойства WorkflowExecutionToken действия GetVcoWorkflowExecutionStatus.
8. Создайте переменную строкового типа.

9. Укажите имя строковой переменной в качестве свойства `VcoWorkflowExecutionStatus` в действии `GetVcoWorkflowExecutionStatus`.

## Результаты

При запуске рабочего процесса в качестве значения переменной `VcoWorkflowExecutionStatus` устанавливается состояние рабочего процесса `vRealize Orchestrator`.

## Получение результатов рабочего процесса vRealize Orchestrator

Если необходимо вызвать рабочий процесс `vRealize Orchestrator` в асинхронном режиме, а затем получить результаты завершенного рабочего процесса на более позднем этапе, вы можете использовать действие `WaitForVcoWorkflowCompletion`.

Действие `WaitForVcoWorkflowCompletion` блокирует рабочий процесс инфраструктуры как услуги до завершения рабочего процесса `vRealize Orchestrator` или окончания времени его ожидания. Действие возвращает результаты рабочего процесса `vRealize Orchestrator`, если он завершается успешно, сообщение об ошибке — если происходит сбой рабочего процесса, или ноль — если закончилось время рабочего процесса.

## Необходимые условия

[Вызов рабочего процесса vRealize Orchestrator](#) использование действия `InvokeVcoWorkflowAsync`.

## Процедура

1. В `vRealize Automation Designer` откройте рабочий процесс, в котором использовалось действие `InvokeVcoWorkflowAsync`.
2. Перейдите к контексту, где необходимо получить результаты рабочего процесса `vRealize Orchestrator`.
3. Перетащите действие `WaitForVcoWorkflowCompletion` в область конструктора.
4. В области свойств задайте имя переменной, представляющей идентификатор виртуальной машины в `VirtualMachineId..`  
  
Настраиваемые рабочие процессы содержат переменную по умолчанию с именем `virtualMachineId`, которое устанавливается во время инициализации.
5. Создайте переменную типа `DynamicOps.VcoModel.Common.VcoWorkflowExecutionToken`.
6. Создайте переменную типа `DynamicOps.VcoModel.Common.VcoWorkflowExecutionToken`.
7. Укажите имя переменной маркера в качестве выходного параметра `executionToken` в действии `InvokeVcoWorkflowAsync`.
8. Укажите то же самое имя переменной как для свойства `WorkflowExecutionToken` действия `WaitForVcoWorkflowCompletion`.

## 9. Получите выходные значения рабочего процесса vRealize Orchestrator.

- а) Создайте переменную типа `DynamicOps.VcoModel.Common.VcoWorkflowExecutionResult`.
- б) Укажите имя переменной результатов в качестве свойства `WorkflowOutput` в действии `WaitForVcoWorkflowCompletion`.

При запуске рабочего процесса в качестве значения переменной устанавливаются результаты рабочего процесса vRealize Orchestrator при наличии.

### Типы объектов vRealize Orchestrator и IaaS

Если в vRealize Automation Designer используется действие `InvokeVcoWorkflow` или `InvokeVcoWorkflowAsync`, свойства ввода и вывода для действия вносятся автоматически на основании выбранных параметров рабочего процесса vRealize Orchestrator.

Основные типы объектов vRealize Orchestrator преобразуются в следующие типы инфраструктуры как услуги.

Таблица 1-19. Типы объектов vRealize Orchestrator и IaaS

Тип vRealize Orchestrator	Тип IaaS
string	string
boolean	bool
number	decimal
SecureString	string
Text	string
Array/T	Array<T>
Свойства	Dictionary<string,object>
Date	DateTime
VC:VirtualMachine	VirtualMachine

**Примечание** Если используется vRealize Orchestrator 5.1, необходимо предварительно установить пакет интеграции vRealize Automation, чтобы обеспечить преобразование типов объектов `VC:VirtualMachine` в `VirtualMachine`.

Все остальные типы vRealize Orchestrator преобразовываются в тип IaaS `VcoSdkObject`.

### Настройка схемы элементов для вызова рабочего процесса изменения состояния

После создания настраиваемого рабочего процесса изменения состояния администратор арендатора или диспетчер бизнес-группы должен включить его в определенную схему элементов, добавив настраиваемое свойство.


Каждый рабочий процесс изменения состояния связан с определенным настраиваемым свойством. При выборе в компьютере состояния с соответствующим рабочим процессом изменения состояния инфраструктура как услуга проверяет наличие соответствующего настраиваемого свойства. Если такое свойство найдено, то выполняется связанный рабочий процесс. Например, если в компьютере есть настраиваемое свойство `ExternalWFStubs.MachineProvisioned`, рабочий процесс `WFStubMachineProvisioned` выполняется до перехода в состояние `MachineProvisioned`.

Настраиваемые свойства можно применить к компьютеру из ряда источников, однако, как правило, свойство для рабочего процесса изменения состояния указывается в схеме элементов, которая включает этот рабочий процесс для всех компьютеров, подготовленных на основе данной схемы.

### Необходимые условия

Войдите в службу vRealize Automation в качестве **администратора арендатора** или **диспетчера бизнес-групп**.

### Процедура

1. Выберите **Проектирование > Схемы элементов**.
2. Укажите имя схемы элементов и нажмите кнопку **Редактировать**.
3. Выберите значок **Свойства схемы элементов** (  ).
4. Перейдите на вкладку **Свойства**.
5. Нажмите **Настраиваемые свойства > Создать**.
6. В текстовом поле **Имя** введите имя настраиваемого свойства, связанного с рабочим процессом.

Имя настраиваемого рабочего процесса	Имя связанного свойства
<b>WFStubMachineProvisioned</b>	<code>ExternalWFStubs.MachineProvisioned</code>
<b>WFStubBuildingMachine</b>	<code>ExternalWFStubs.BuildingMachine</code>
<b>WFStubMachineDisposing</b>	<code>ExternalWFStubs.MachineDisposing</code>
<b>WFStubUnprovisionMachine</b>	<code>ExternalWFStubs.UnprovisionMachine</code>
<b>WFStubMachineRegistered</b>	<code>ExternalWFStubs.MachineRegistered</code>
<b>WFStubMachineExpired</b>	<code>ExternalWFStubs.MachineExpired</code>

7. Оставьте текстовое поле **Значение** пустым.  
Рабочий процесс зависит от наличия свойства, а не конкретного значения.
8. Нажмите кнопку **ОК**, чтобы сохранить данное свойство.
9. Нажмите кнопку **ОК**.



## Результаты

Рабочий процесс теперь включен для новых компьютеров, которые подготовлены из этой схемы элементов.

## Настройка рабочего процесса операции меню

После настройки рабочего процесса операции меню необходимо выполнить дополнительную настройку, после чего этот рабочий процесс будет доступен для пользователей в консоли vRealize Automation.

### Настройка операции меню

Чтобы настроить операцию меню, необходимо создать файл конфигурации операции и установить его в диспетчере моделей.

### Процедура

#### 1. Создание файла конфигурации операции файла к

Файл конфигурации операции требуется для рабочих процессов операций меню. Он определяет аспекты настраиваемого параметра меню консоли vRealize Automation, например, текст отображения данных о том, каким ролям предоставлен доступ к параметру, а также состояния компьютера, для которых доступен этот параметр.

#### 2. Установка операции в диспетчере моделей

Установите операцию в диспетчере моделей с помощью служебной программы командной строки CloudUtil.

### Следующие шаги

Если операция меню предназначена для использования в каталоге служб, ее необходимо зарегистрировать там, чтобы можно было предоставлять пользователям соответствующие права. [Регистрация новых операций меню с помощью каталога служб](#).

### Создание файла конфигурации операции файла к

Файл конфигурации операции требуется для рабочих процессов операций меню. Он определяет аспекты настраиваемого параметра меню консоли vRealize Automation, например, текст отображения данных о том, каким ролям предоставлен доступ к параметру, а также состояния компьютера, для которых доступен этот параметр.

### Процедура

#### 1. Создайте новый XML-файл.

```
<?xml version="1.0" encoding="utf-8"?>
```

#### 2. Создайте корневой элемент customOperations.

```
<customOperations xmlns="http://www.dynamicops.com/schemas/2009/OperationConfig/">
</customOperations>
```

Элемент должен задавать пространство имен XML `http://www.dynamicops.com/schemas/2009/OperationConfig/`.

3. Для каждой операции, которую необходимо определить, добавьте элемент `operation` в `customOperations`.

```
<operation name="WFMachineMenu1" displayName="Execute Machine Menu task">
</operation>
```

Элемент `operation` содержит следующие атрибуты.

Атрибут	Описание
<b>name</b>	Имя рабочего процесса, который выполняется в этой операции.
<b>displayName</b>	Описательная метка для параметра в меню компьютера.

4. Укажите роль для предоставления доступа к операции меню.

- a) Добавьте элемент `authorizedTasks`.

```
<operation name="WFMachineMenu1" displayName="Execute Machine Menu task">
  <authorizedTasks>
  </authorizedTasks>
</operation>
```

- b) Для каждой роли, которой необходимо предоставить доступ к операции, добавьте элемент `task`, например:

```
<authorizedTasks>
  <task>VRM User Custom Event</task>
  <task>VRM Support Custom Event</task>
  <task>Group Administrator Custom Event</task>
  <task>Enterprise Administrator Custom Event</task>
  <task>VRM Administrator Custom Event</task>
</authorizedTasks>
```

Допустимое содержимое элемента `task`:

Содержимое элемента	Описание
<b>VRM User Custom Event</b>	Предоставляет доступ к операции для всех пользователей.
<b>VRM Support Custom Event</b>	Предоставляет доступ к операции для поддерживаемых пользователей.
<b>Group Administrator Custom Event</b>	Предоставляет доступ к операции для диспетчеров бизнес-групп.
<b>Enterprise Administrator Custom Event</b>	Предоставляет доступ к операции для администраторов структуры
<b>VRM Administrator Custom Event</b>	Предоставляет доступ к операции только для администраторов инфраструктуры как услуги

**5.** (дополнительно) Укажите состояния компьютера, для которых доступна операция.

а) Добавьте элемент `machineStates`.

```
<operation name="WFMachineMenu1" displayName="Execute Machine Menu task">
  <machineStates>
  </machineStates>
</operation>
```

б) Для каждого состояния, в котором операция должна быть доступна, добавьте элемент `state`.

```
<machineStates>
  <state>On</state>
  <state>Off</state>
</machineStates>
```

Значением может быть любое возможное состояние компьютера. Полный список состояний компьютера см. в разделах *Конфигурация инфраструктуры как услуги для виртуальных платформ*, *Конфигурация инфраструктуры как услуги для физических компьютеров* или *Конфигурация инфраструктуры как услуги для облачных платформ*.

Если элемент опущен, операция будет доступна для всех состояний компьютера.

**Пример**

Ниже приведен пример полного файла конфигурации операции:

```
<?xml version="1.0" encoding="utf-8" ?>
<customOperations xmlns="http://www.dynamicops.com/schemas/2009/OperationConfig/">
  <operation name="WFMachineMenu1" displayName="Execute Machine Menu task">
    <authorizedTasks>
      <task>VRM User Custom Event</task>
      <task>VRM Support Custom Event</task>
      <task>Group Administrator Custom Event</task>
      <task>Enterprise Administrator Custom Event</task>
      <task>VRM Administrator Custom Event</task>
    </authorizedTasks>
    <machineStates>
      <state>On</state>
      <state>Off</state>
    </machineStates>
  </operation>
</customOperations>
```

**Установка операции в диспетчере моделей**

Установите операцию в диспетчере моделей с помощью служебной программы командной строки `CloudUtil`.

**Необходимые условия**

[Создание файла конфигурации операции](#) файла к.

## Процедура

1. Откройте окно командной строки с повышенными привилегиями.
2. Выполните команду CloudUtil.exe со следующими аргументами.

- `CloudUtil.exe Operation-Create -c <path to operation definition file>`
- Можно указать узел диспетчера моделей и запросить трассировку стека в случае ошибки (дополнительно).

```
CloudUtil.exe Operation-Create -c <path to operation definition file>
--repository <Model Manager Root URI> -v
```

## Следующие шаги

Если операция меню предназначена для использования в каталоге служб, ее необходимо зарегистрировать там, чтобы можно было предоставлять пользователям соответствующие права. [Регистрация новых операций меню с помощью каталога служб](#).

## Регистрация новых операций меню с помощью каталога служб

После установки новых операций меню разработчик рабочего процесса должен зарегистрировать их в каталоге служб, чтобы можно было предоставить соответствующие права пользователям.

## Необходимые условия

- [Настройка операции меню](#).
- На узле диспетчера модели инфраструктуры как услуги войдите в Windows в качестве локального пользователя с правами **администратора**.

## Процедура

1. Откройте окно командной строки с повышенными привилегиями.
2. Перейдите в корневой каталог установки инфраструктуры как услуги.  
Обычно применяется этот путь `C:\Program Files (x86)\VMware\VCAC`.
3. Перейдите в каталог `Server\Model Manager Data\Cafe`.
4. Выполните следующую команду:

```
Vcac-Config.exe RegisterCatalogTypes -v
```

## Следующие шаги

Администратор арендатора или диспетчер бизнес-группы должен предоставить право на новое действие, прежде чем оно станет доступно пользователям в каталоге служб. Дополнительные сведения см. в разделе *Администрирование арендаторов*.

## Настройка схемы элементов для включения рабочего процесса операций меню

Включите рабочий процесс операции меню для компьютера, подготовленного из определенной схемы элементов путем обновления конфигурации безопасности для этой схемы.

### Необходимые условия

- Войдите в vRealize Automation в качестве **администратора арендатора** или **диспетчера бизнес-групп**.
- Операция меню должна быть настроена и зарегистрирована в каталоге служб.

### Процедура

1. Выберите **Проектирование > Схемы элементов**.
2. Укажите имя схемы элементов и нажмите кнопку **Редактировать**.
3. Перейдите на вкладку **Действия**.
4. Установите флажок, соответствующий операции, которую необходимо включить.
5. Нажмите кнопку **ОК**.

### Результаты

Операция меню теперь включена для компьютеров, подготовленных из этой схемы элементов, и доступна для всех ролей пользователей, указанных в файле конфигурации операции.

### Следующие шаги

Если операция меню предназначена для использования в каталоге служб, необходимо предоставить соответствующие права пользователям. Дополнительные сведения см. в разделе *Администрирование арендаторов*.

## Возврат к предыдущей редакции рабочего процесса

В диалоговом окне **Загрузить рабочий процесс** диспетчера моделей отображаются все редакции рабочего процесса, и предоставляется доступ к журналу всех версий рабочих процессов.

Каждый раз при отправлении рабочего процесса в диспетчере моделей обновляются редакция и метка времени.

### Необходимые условия

Запустите консоль vRealize Automation Designer.

### Процедура

1. Нажмите **Загрузить**.
2. Выберите редакцию рабочего процесса, к которой необходимо вернуться.

В исходных рабочих процессах, предоставляемых VMware, содержится редакция с номером 0 (ноль).

3. Нажмите кнопку **ОК**.

4. Обновите рабочий процесс в диспетчере моделей, нажав кнопку **Отправить**.

#### Результаты

Ранняя редакция становится последней редакцией в диспетчере моделей. Например, если созданы редакции 1 и 2 для рабочего процесса, а затем загружена и сохранена редакция 0, редакции 0 и 3 становятся одинаковыми, рабочий процесс возвращается к редакции, предоставленной VMware.

## Рабочие процессы и распределенное управление

Чтобы ограничить выполнение рабочих процессов рамками определенных диспетчеров Distributed Execution Manager, можно использовать навыки.

Навык похож на тег, который можно применить к рабочим процессам и экземплярам DEM Worker. Если рабочий процесс не связан с какими-либо навыками, его может выполнять любой экземпляр DEM Worker. Если рабочий процесс связан с одним или несколькими навыками, его могут выполнять только те экземпляры DEM Worker, которые связаны со всеми аналогичными навыками.

Навыки полезны, когда для определенного рабочего процесса на узле необходимо установить DEM с определенными условиями. Например, может потребоваться ограничение рабочих процессов облачной подготовки до определенного процесса DEM, запущенного в узле с требуемым сетевым доступом к URL-адресам Amazon.

Навыки можно также использовать, чтобы связать рабочие процессы с определенным расположением центра обработки данных. Например, один диспетчер DEM можно установить в центре обработки данных в Бостоне, а другой — в центре обработки данных в Лондоне и затем с помощью навыков направлять определенные операции в один из центров.

## Связывание рабочих процессов и процессов DEM с помощью навыков

Свяжите рабочие процессы с определенным процессом DEM или набором экземпляров рабочего процесса путем добавления навыка в диспетчер моделей, а затем свяжите этот навык с одним или несколькими рабочими процессами и процессами DEM.

#### Необходимые условия

Запустите консоль vRealize Automation Designer.


#### Процедура

1. На ленте выберите **Управление навыками**.
2. В текстовом поле в верхнем левом углу диалогового окна **Управление навыками** введите имя нового навыка и нажмите кнопку «Добавить».


Имя навыка должно быть уникальным. Если имя нового навыка совпадает с именем существующего навыка, кнопка «Добавить» будет недоступна.

3. Выберите имя навыка из списка слева.

**4. Свяжите навык с одним или несколькими рабочими процессами DEM.**

- а) Щелкните значок **Добавить** () рядом с диспетчерами Distributed Execution Manager.
- б) В диалоговом окне **Выбор DEM** выберите один или несколько экземпляров рабочих процессов DEM.
- в) Нажмите кнопку **ОК**.

**5. Свяжите навык с одним или несколькими рабочими процессами.**

- а) Щелкните значок **Добавить** () рядом с разделом «Рабочие процессы».
- б) В диалоговом окне **Выбор рабочих процессов** выберите один или несколько экземпляров рабочих процессов.
- в) Нажмите кнопку **ОК**.

Рабочие процессы, связанные с этим навыком, можно выполнить только с помощью рабочих процессов DEM, которые связаны с этим навыком.

- 6.** После завершения добавления навыков, а также связывания их с рабочими процессами и процессами DEM, нажмите кнопку **ОК**, чтобы закрыть диалоговое окно **Управление навыками** и сохранить изменения в диспетчере моделей.


## Удаление связей между навыками и рабочими процессами DEM

При удалении связи между навыком и рабочим процессом DEM экземпляр этого рабочего процесса больше не сможет выполнять рабочие процессы, связанные с навыком.

### Необходимые условия

Запустите консоль vRealize Automation Designer.

### Процедура

- 1.** На ленте выберите **Управление навыками**.
- 2.** В диалоговом окне **Управление навыками** выберите имя навыка из списка слева.
- 3.** В списке диспетчеров Distributed Execution Manager выберите имя одного или нескольких экземпляров рабочего процесса DEM Worker и щелкните значок **Удалить** ()
- 4.** Нажмите кнопку **ОК**, чтобы закрыть диалоговое окно **Управление навыками** и сохранить изменения в диспетчере моделей.


## Удаление связей между навыками и рабочими процессами

При удалении связи между навыком и рабочим процессом этот рабочий процесс больше не ограничивается рабочими процессами DEM, связанным с тем же навыком.

### Необходимые условия

Запустите консоль vRealize Automation Designer.

## Процедура

1. На ленте выберите **Управление навыками**.
2. В диалоговом окне **Управление навыками** выберите имя навыка из списка слева.
3. В списке «Рабочие процессы» выберите имя одного или нескольких экземпляров рабочего процесса DEM и щелкните значок **Удалить** ().
4. Нажмите кнопку **ОК**, чтобы закрыть диалоговое окно **Управление навыками** и сохранить изменения в диспетчере моделей.


## Удаление навыка

При удалении навыка также удаляются его связи с любым рабочими процессами и процессами DEM.

### Необходимые условия

Запустите консоль vRealize Automation Designer.

## Процедура

1. На ленте выберите **Управление навыками**.
2. В диалоговом окне **Управление навыками** выберите имя навыка из списка слева.
3. Щелкните значок **Удалить** () в верхней части списка навыков.

После подтверждения удаления навыка его имя будет отображаться затемненным, чтобы указать на пометку для удаления.
4. Нажмите кнопку **ОК**, чтобы закрыть диалоговое окно **Управление навыками** и сохранить изменения в диспетчере моделей. Нажмите кнопку **Отмена**, если вы передумали удалять навык и его связи с диспетчерами DEM и рабочими процессами.

## Ссылка на команду CloudUtil

В этом разделе приведены ссылки на команды в интерфейсе командной строки CloudUtil.

CloudUtil — это интерфейс командной строки для vRealize Automation Designer. Запустите команды на компьютере под управлением Windows, на котором запущен конструктор. Место установки по умолчанию на компьютере под управлением Windows: C:\Program Files (x86)\VMware\VCAC\Design Center.

---

**Примечание** В командах CloudUtil диспетчер моделей называется repository, а диспетчер Distributed Execution Manager (DEM) — agent.

---

## Команды DEM

Команды DEM позволяют просматривать список диспетчеров Distributed Execution Manager, зарегистрированных в диспетчере моделей, а также добавлять или удалять связи между навыками и диспетчерами DEM.



## DEM-Add-Skills

Связывает навыки с зарегистрированным диспетчером Distributed Execution Manager.

### Описание

```
CloudUtil.exe DEM-Add-Skills -n|--name <Name> -s|--skills <Skills> [--repository <Model Manager Root URI>] [-v|--verbose]
```

### Аргументы DEM-Add-Skills

Аргумент	Описание
-n   - -name	Имя зарегистрированного диспетчера Distributed Execution Manager.
- -repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-s   - -skills	Список навыков с разделителями-запятые для установления связи с диспетчером Distributed Execution Manager.
-v   - -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

**Примечание** Навыки должны уже существовать в диспетчере моделей. См. раздел [Skill-Install](#).

## DEM-List

Перечисляет все диспетчеры Distributed Execution Manager, зарегистрированные в диспетчере моделей и связанные с ними навыки.

### Описание

```
CloudUtil.exe DEM-List [--repository <Model Manager Root URI>] [-v|--verbose]
```

### Аргументы DEM-List

Аргумент	Описание
- -repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-v   - -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

## DEM-Remove-Skills

Удаляет связь между навыками и диспетчером Distributed Execution Manager.

### Описание

```
CloudUtil.exe DEM-Remove-Skills -n|--name <Name> -s|--skills <Skills> [--repository <Model Manager Root URI>] [-v|--verbose]
```

## Аргументы DEM-Remove-Skills

Аргумент	Описание
-n   - -name	Имя зарегистрированного диспетчера Distributed Execution Manager.
- -repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-s   - -skills	Список навыков с разделителями-запятыми для удаления из диспетчера Distributed Execution Manager.
-v   - -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

## Команды файла

Команды файлов позволяют хранить файлы (как правило, сценарии) в диспетчере моделей и управлять ими.

### File-Export

Экспортирует файл из диспетчера моделей.

#### Описание

```
CloudUtil.exe File-Export -n|--name <Name> -o|--output <Output File> [-i|--iteration <Iteration>] [--repository <Model Manager Root URI>] [-v|--verbose]
```

#### Аргументы File-Export

Аргумент	Описание
-i   - -iteration	Строка версии файла в диспетчере моделей (дополнительно). Значение по умолчанию равно <b>0,0</b> .
-n   - -name	Понятное имя файла в диспетчере моделей.
-o   - -output	Путь для вывода файла.
- -repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-v   - -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

### File-Import

Импортирует файл в диспетчер моделей.

#### Описание

```
CloudUtil.exe File-Import -n|--name <Name> -f|--filename <File Name> [-d|--description <Description>] [-i|--iteration <Iteration>] [--repository <Model Manager Root URI>] [-v|--verbose]
```

## Аргументы File-Import

Аргумент	Описание
-d   - -description	Определение файла (дополнительно).
-f   - -filename	Путь к файлу для импорта в диспетчере моделей.
-i   - -iteration	Строка версии файла в диспетчере моделей (дополнительно). Значение по умолчанию равно <b>0,0</b> .
-n   - -name	Понятное имя для назначения файлу в диспетчере моделей.
- -repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-v   - -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

## File-List

Перечисляет все файлы, импортированные в диспетчер моделей.

### Описание

```
CloudUtil.exe File-List [--repository <Model Manager Root URI>] [-v|--verbose]
```

## Аргументы File-List

Аргумент	Описание
- -repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-v   - -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

## File-Remove-All

Удаляет все редакции указанной версии файла из диспетчера моделей.

### Описание

```
CloudUtil.exe File-Remove-All -n|--name <Name> [-i|--iteration <Iteration>]
[--repository <Model Manager Root URI>] [-v|--verbose]
```

## Аргументы File-Remove-All

### Таблица 1-20.

Аргумент	Описание
-i   - -iteration	Строка версии файла в диспетчере моделей (дополнительно). Значение по умолчанию равно <b>0,0</b> .
-n   - -name	Понятное имя файла в диспетчере моделей.

Таблица 1-20. (продолжение)

Аргумент	Описание
- -repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-v   - -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

## File-Remove-Rev

Удаляет определенную редакцию файла из диспетчера моделей.

### Описание

```
CloudUtil.exe File-Remove-Rev -n|--name <Name> -r|--revision <Revision> [-i|--iteration <Iteration>]
[--repository <Model Manager Root URI>] [-v|--verbose]
```

### Аргументы File-Export

Аргумент	Описание
-i   - -iteration	Строка версии файла в диспетчере моделей (дополнительно). Значение по умолчанию равно <b>0,0</b> .
-n   - -name	Понятное имя файла в диспетчере моделей.
-r   - -revision	Редакция файла, которую необходимо удалить.
- -repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-v   - -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

## File-Rollback

Возвращает файл в указанную редакцию в диспетчере моделей.

### Описание

```
CloudUtil.exe File-Rollback -n|--name <Name> -r|--revision <Revision> [-i|--iteration <Iteration>] [--
repository <Model Manager Root URI>] [-v|--verbose]
```

### Аргументы File-Rollback

Таблица 1-21.

Аргумент	Описание
-i   - -iteration	Строка версии файла в диспетчере моделей (дополнительно). Значение по умолчанию равно <b>0,0</b> .
-n   - -name	Понятное имя файла в диспетчере моделей.
-r   - -revision	Редакция файла, к которой необходимо вернуться.

Таблица 1-21. (продолжение)

Аргумент	Описание
- -repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-v   - -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

## File-Update

Обновляет ранее импортированный файл в диспетчере моделей до новой редакции.

### Описание

```
CloudUtil.exe File-Update -n|--name <Name> -f|--filename <File Name> [-i|--iteration <Iteration>] [--repository <Model Manager Root URI>] [-v|--verbose]
```

### Аргументы File-Update

Аргумент	Описание
-f   - -filename	Путь к обновленному файлу.
-i   - -iteration	Строка версии файла в диспетчере моделей (дополнительно). Значение по умолчанию равно 0,0.
-n   - -name	Понятное имя файла в диспетчере моделей.
- -repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-v   - -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

## Команды операций

Команды операций позволяют управлять настраиваемыми операциями в диспетчере моделей.

### Operation-Create

Создает настраиваемую операцию или набор операций, которые можно выполнить на компьютере на основе файла определения операции.

### Описание

```
CloudUtil.exe Operation-Create -c|--operationConfig <Operation Definition File> [--repository <Model Manager Root URI>] [-v|--verbose]
```

## Аргументы Operation-Create

Аргумент	Описание
-c   - -operationConfig	Путь к файлу определения операции (XML).
- -repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-v   - -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

## Operation-Delete

Удаляет настраиваемую операцию из диспетчера моделей.

### Описание

```
CloudUtil.exe Operation-Delete -n|--name <Name> [--force] [--repository <Model Manager
Root URI>] [-v|--verbose]
```

## Аргументы Operation-Delete

Аргумент	Описание
- -force	Принудительное удаление операции (дополнительно).
-n   - -name	Название настраиваемой операции в диспетчере моделей.
- -repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-v   - -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

## Operation-List

Перечисляет все настраиваемые операции в диспетчере моделей.

### Описание

```
CloudUtil.exe Operation-List [--repository <Model Manager Root URI>] [-v|--verbose]
```

## Аргументы Operation-List

Аргумент	Описание
- -repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-v   - -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

## Команды для навыков

Команды для навыков позволяют управлять навыками, связанными с рабочими процессами и диспетчерами Distributed Execution Manager.

### Skill-Install

Устанавливает навык в диспетчере моделей.

#### Описание

```
CloudUtil.exe Skill-Install -n|--name <Name> [--repository <Model Manager Root URI>] [-v|--verbose]
```

#### Аргументы Skill-Install

Аргумент	Описание
-n   -name	Имя навыка в диспетчере моделей.
-repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-v   -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

### Skill-List

Перечисляет все навыки, установленные в диспетчере моделей.

#### Описание

```
CloudUtil.exe Skill-List [--repository <Model Manager Root URI>] [-v|--verbose]
```

#### Аргументы Skill-List

Аргумент	Описание
-repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-v   -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

### Skill-Uninstall

Удаляет навык из диспетчера моделей.

#### Описание

```
CloudUtil.exe Skill-Uninstall -n|--name <Name> [--repository <Model Manager Root URI>] [-v|--verbose]
```

## Аргументы Skill-Uninstall

Аргумент	Описание
-n   -name	Имя навыка, который необходимо удалить из диспетчера моделей.
-repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-v   -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

**Примечание** Навык нельзя удалить, если он связан с диспетчером Distributed Execution Manager или рабочим процессом. См. раздел [DEM-Remove-Skills](#) или [Workflow-Remove-Skills](#).

## Команды рабочих процессов

Команды рабочих процессов позволяют управлять настраиваемыми рабочими процессами IaaS в диспетчере моделей, а также навыками, связанными с любыми рабочими процессами.

### Workflow-Add-Skills

Связывание навыков с рабочим процессом в диспетчере моделей.

```
CloudUtil.exe Workflow-Add-Skills -n|--name <Name> -s|--skills <Skills> [--repository <Model Manager Root URI>] [-v|--verbose]
```

Таблица 1-22. Аргументы Workflow-Add-Skills

Аргумент	Описание
Name	Название рабочего процесса в диспетчере моделей.
Skills	Список навыков с разделителями-запятыми, подлежащих связыванию с этим рабочим процессом.
-repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-v   -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

**Примечание** Навыки должны уже существовать в диспетчере моделей. См. раздел [Skill-Install](#).

### Workflow-List

Указание всех рабочих процессов, установленных в диспетчере моделей, и связанных с ними навыков.

```
CloudUtil.exe Workflow-List [--repository <Model Manager Root URI>] [-v|--verbose]
```



Таблица 1-23. Аргументы Workflow-List

Аргумент	Описание
-repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-v   -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

## Workflow-Remove-Skills

Удаление связи между навыками и рабочим процессом в диспетчере моделей.

### Описание

```
CloudUtil.exe Workflow-Remove-Skills -n|--name <Name> -s|--skills <Skills> [--repository
<Model Manager Root URI>] [-v|--verbose]
```

### Аргументы Workflow-Remove-Skills

Аргумент	Описание
-n   -name	Название рабочего процесса в диспетчере моделей.
-repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-s   -skills	Список навыков с разделителями-запятыми, подлежащих удалению из этого рабочего процесса.
-v   -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

## Workflow-Rollback

Возвращение рабочего процесса к определенной редакции.

### Описание

```
CloudUtil.exe Workflow-Rollback -n|--name <Name> -r|--revision <Revision> [--repository <Model
Manager Root URI>] [-v|--verbose]
```

### Аргументы Workflow-Rollback

Аргумент	Описание
-n   -name	Название рабочего процесса в диспетчере моделей.
-repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-r   -revision	Редакция рабочего процесса, к которой необходимо вернуться.
-v   -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

## Workflow-Update

Обновление настраиваемого рабочего процесса новой редакцией.

```
CloudUtil.exe Workflow-Update -f|--filename <File Name> -n|--name <Name> [-d|--description
<Description>] [--repository <Model Manager Root URI>] [-v|--verbose]
```

Таблица 1-24. Аргументы Workflow-Update

Аргумент	Описание
File Name	Путь к файлу (XAML), содержащему обновленный рабочий процесс.
Name	Название рабочего процесса, который необходимо обновить.
Description	Описание рабочего процесса (дополнительно).
- -repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-v   - -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

## Команды импорта

Команды импорта позволяют импортировать одну или несколько виртуальных машин в развертывание vRealize Automation,

### Machine-BulkRegisterExport

Создает CSV-файл данных, используемый для импорта виртуальных машин в развертывание vRealize Automation.

#### Описание

```
CloudUtil.exe Machine-BulkRegisterExport [-b|--blueprint] [-m|--managed] [-e|--exportNames] [-p|--
properties] -f|--filename <Value> [-g|--group <Value>] [-i|--ignore] [-o|--owner <Value>] [-t|--
machinetype <Value>] [-n|--resourceName <Value>] [-r|--resourceType <Value>] [--repository <Value>] [-
sn|--sourcename <Value>] [-st|--sourcetype <Value>] -u|--user <value> [-v|--verbose]
```

### Machine-BulkRegisterExport Arguments

Таблица 1-25.

Аргумент	Описание
-b   - -blueprint	Включите имя схемы элементов (дополнительно)
-e   - -exportNames	Экспорт имен вместо идентификаторов GUID (дополнительно).
-f   - -filename	Укажите имя CSV-файла данных, содержащего список имен компьютеров, например, <code>filename.csv</code> . Файл сохраняется в текущем пути по умолчанию. Можно также указать полный путь к предпочитаемому каталогу.
-g   - -group	Укажите имя бизнес-группы, например, «Проектирование» (дополнительно).

Таблица 1-25. (продолжение)

Аргумент	Описание
-i   - -ignore	Игнорируйте недействительные аргументы (дополнительно).
-m   - -managed	Экспорт управляемых виртуальных машин (дополнительно). По умолчанию установлен экспорт неуправляемых виртуальных машин.
-n   - -resourceName	Чтобы отфильтровать имена ресурсов, укажите имя вычислительных ресурсов или конечной точки (дополнительно).
-o   - -owner	Укажите владельца импортированной виртуальной машины (дополнительно), например, jsmith.
-p   - -properties	Экспорт свойств для управляемых виртуальных машин (дополнительно).
-r   - -resourceType	Для фильтра по типу ресурса укажите значение 1 для вычислительного ресурса или 2 для конечной точки (дополнительно).
- -repository	Корневой код URI диспетчера моделей (дополнительно), например, <code>http://hostname/repository</code> . Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-sn   - -sourcename	Укажите имя кластера или конечной точки (дополнительно).
-st   - -sourcetype	Укажите тип источника в качестве кластера или конечной точки (дополнительно).
-t   - -machinetype	Укажите тип компьютера для экспорта (дополнительно), например, виртуальный, физический, облачный, AppService, vApp.
-u   - -user	Укажите администратора структуры, выполняющего массовую регистрацию.
-v   - -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).

## Machine-BulkRegisterImport

Импорт одной или нескольких виртуальных машин в целевое развертывание vRealize Automation.

### Описание

```
CloudUtil.exe Machine-BulkRegisterImport [-b|--batch] [-d|--delay <value>] -f|--filename <value> [-i|--ignore] [-h|--humanreadable] -n|--name <value> [--repository <value>] [-s|--skipUser] -t|--time <value> -u|--user <value> [-v|--verbose] [-w|--whatIf]
```

## Machine-BulkRegisterImport Arguments

Таблица 1-26.

Аргумент	Описание
-b   - -batch	Размер пакета (дополнительно).
-d   - -delay	Укажите время задержки обработки в формате hh:mm:ss (дополнительно), например, 02:20:10.
-f   - -filename	Укажите имя CSV-файла данных, содержащего список имен компьютеров. Например, filename.csv.
-h   - -humanreadable	Входной файл содержит имена виртуальных машин, а не идентификаторов GUID (дополнительно).
-i   - -ignore	Игнорируйте зарегистрированные или управляемые виртуальные машины (дополнительно).
-n   - -name	Укажите имя рабочей очереди, чтобы выполнить импорт в целевой продукт vRealize Automation.
- -repository	Корневой код URI диспетчера моделей (дополнительно), например, http://hostname/repository. Значение по умолчанию задается в файле конфигурации CloudUtil в разделе <appSettings> ключа repositoryAddress.
-s   - -skipUser	Задайте для владельца компьютера значение, указанное в столбце «Владелец» в CSV-файле данных, не проверяя наличие такого пользователя (дополнительно). При выборе этого параметра можно сократить время, необходимое для импорта.
-t   - -time	Укажите время начала рабочего процесса в формате MM/DD/YYYY hh:mm GMT, например, 04/18/2014 10:01 GMT. Предполагается, что указанное время начала — местное время сервера, а не местное время рабочей станции пользователя.
-u   - -user	Укажите администратора структуры, выполняющего массовую регистрацию.
-v   - -verbose	В случае возникновения ошибки помимо сообщения об исключении выводится трассировка стека (дополнительно).
whatif	Установите проверку CSV-файла, но не импортируйте виртуальные машины (дополнительно).

## Справка по действиям рабочих процессов vRealize Automation

VMware предоставляет библиотеку действий рабочих процессов с vRealize Automation Designer для использования в рабочих процессах настройки.

**Примечание** CDK исключается, начиная с версии vRealize Automation 7.0. Для рабочих ситуаций, в которых ранее применялась функция CDK, можно использовать рабочие процессы vRealize Orchestrator.

В vRealize Automation Designer также включены пять категорий действий Windows Workflow Foundation, в частности Control Flow, Flowchart, Primitives, Collection и Error Handling.

В данном разделе предоставлена справка по действиям рабочих процессов IaaS, включенным с vRealize Automation Designer в пространства имен `DynamicOps.Repository.Activities` и `DynamicOps.Cdk.Activities`. Действия, связанные с вызовом рабочих процессов vRealize Orchestrator, описаны в разделе [Использование действий рабочего процесса vRealize Orchestrator](#).

**Примечание** В библиотеке действий IaaS диспетчер моделей обозначается как `repository`.

## DynamicOps.Repository.Activities

Пространство имен `DynamicOps.Repository.Activities` содержит действия основного рабочего процесса для рабочих процессов инфраструктуры как услуги.

**Примечание** CDK исключается, начиная с версии vRealize Automation 7.0. Для рабочих ситуаций, в которых ранее применялась функция CDK, можно использовать рабочие процессы vRealize Orchestrator.

### AddLink

Добавляет указанную ссылку в набор объектов, которые отслеживает `DataServiceContext`.

Таблица 1-27. Входные параметры действия `AddLink`

Аргумент	Тип	Описание
<code>DataServiceContext</code>	<code>RepositoryServiceContext</code>	Аргумент <code>DataServiceContext</code> , к которому необходимо добавить ссылку.
<code>Source</code>	<code>Object</code>	Исходный объект для новой ссылки.
<code>SourceProperty</code>	<code>String</code>	Имя свойства перехода в исходном объекте, который возвращает связанный объект.
<code>Target</code>	<code>Object</code>	Объект, связанный с исходным объектом по новой ссылке.

### AddObject

Добавляет указанный объект в набор объектов, которые отслеживает `DataServiceContext`.

Таблица 1-28. Входные параметры действия `AddObject`

Аргумент	Тип	Описание
<code>DataServiceContext</code>	<code>RepositoryServiceContext</code>	Аргумент <code>DataServiceContext</code> , к которому необходимо добавить объект.
<code>Instance</code>	<code>Object</code>	Объект, который необходимо отслеживать <code>DataServiceContext</code> .

### AttachTo

Сообщает `DataServiceContext` о необходимости начать отслеживание указанного ресурса.

Таблица 1-29. Входные параметры действия AttachTo

Аргумент	Тип	Описание
DataServiceContext	RepositoryServiceContext	Действие DataServiceContext, которое должно отслеживать ресурс.
Instance	Object	Ресурс, который необходимо отслеживать DataServiceContext. Ресурс прилагается в состоянии «Без изменений».

## CreateRepositoryServiceContext<T>

Создает контекст для модели, загруженной в диспетчер моделей.

При добавлении этого действия к рабочему процессу в vRealize Automation Designer необходимо выбрать класс, который наследуется из класса RepositoryServiceContext.

Таблица 1-30. Входные параметры действия CreateRepositoryServiceContext&lt;T&gt;

Аргумент	Тип	Описание
Uri	URI	Корневой код URI для использования при подключении к модели (дополнительно).
Username	String	Имя пользователя для использования при подключении к контексту (дополнительно).

Таблица 1-31. Выходные параметры действия CreateRepositoryServiceContext&lt;T&gt;

Аргумент	Тип	Описание
Result	RepositoryServiceContext	Определенный возвращаемый тип является экземпляром класса, выбранного при добавлении действия в рабочий процесс.

## DeleteLink

Изменяет состояние ссылки на удаленные элементы в списке ссылок, отслеживаемых DataServiceContext.

Таблица 1-32. Входные параметры действия DeleteLink

Аргумент	Тип	Описание
DataServiceContext	RepositoryServiceContext	Аргумент DataServiceContext, из которого необходимо удалить ссылку.
Source	Object	Исходный объект в ссылке будет помечен для удаления.
SourceProperty	String	Имя свойства перехода в исходном объекте, который используется для доступа к целевому объекту.
Target	Object	Целевой объект включен в ссылку, которая привязана к исходному объекту. Целевой объект должен иметь тип, определяемый свойством исходного объекта, или подтип.

## DeleteObject

Изменяет состояние указанного объекта, который необходимо удалить в DataServiceContext.

Таблица 1-33. Входные параметры действия DeleteObject

Аргумент	Тип	Описание
DataServiceContext	RepositoryServiceContext	Аргумент DataServiceContext, из которого необходимо удалить ресурс.
Instance	Object	Отслеживаемую сущность необходимо перевести в состояние «Удалено».

## InvokeRepositoryWorkflow

Выполняет рабочий процесс, установленный в диспетчере моделей.

Таблица 1-34. Входные параметры действия InvokeRepositoryWorkflow

Аргумент	Тип	Описание
WorkflowType	Сущность WorkflowDefinition	Рабочий процесс для выполнения.
WorkflowInputs	Dictionary<string, object>	Вводит значения в рабочий процесс (дополнительно).
CallingInstance	Сущность WorkflowInstance	Рабочий процесс, который вызывает выполняемый рабочий процесс, и к которому выполнится возврат, (дополнительно).

## LoadProperty

Загружает отложенное содержимое для указанного свойства из службы данных.

Таблица 1-35. Входные параметры действия LoadProperty

Аргумент	Тип	Описание
DataServiceContext	RepositoryServiceContext	Аргумент DataServiceContext, из которого необходимо загрузить свойство.
Instance	Object	Сущность, в которой содержится свойство для загрузки.
InstanceProperty	String	Имя свойства указанной сущности для загрузки.

## SaveChanges

Сохраняет в хранилище изменения, которые отслеживает DataServiceContext.

Таблица 1-36. Входные параметры действия SaveChanges

Аргумент	Тип	Описание
DataServiceContext	RepositoryServiceContext	Действие DataServiceContext, которое отслеживает изменения для сохранения.

## SetLink

Сообщает DataServiceContext о существовании новой связи между указанными объектами, которая представлена свойством, указанным в аргументе SourceProperty.

Таблица 1-37. Входные параметры действия SetLink

Аргумент	Тип	Описание
DataServiceContext	RepositoryServiceContext	DataServiceContext используется для уведомления о ссылке.
Source	Object	Исходный объект для новой ссылки.
SourceProperty	String	Свойство в исходном объекте, идентифицирующее целевой объект как новую ссылку.
Target	Object	Дочерний объект, включенный в новую ссылку для инициализации за счет вызова этого метода. Целевой объект должен быть подтипом типа, идентифицированного аргументом SourceProperty. Если для аргумента Target установлено значение NULL, то вызов представляет собой операцию удаления ссылки.

## UpdateObject

Изменяет состояние указанного объекта в DataServiceContext до значения «Изменено».

Таблица 1-38. Входные параметры действия UpdateObject

Аргумент	Тип	Описание
DataServiceContext	RepositoryServiceContext	Действие DataServiceContext отслеживание сущности для обновления.
Instance	Object	Отслеживаемую сущность необходимо назначить состоянию «Изменено».

## DynamicOps.Cdk.Activities

Пространство имен DynamicOps.Cdk.Activities содержит расширенные действия для рабочих процессов инфраструктуры как услуги.

**Примечание** CDK исключается, начиная с версии vRealize Automation 7.0. Для рабочих ситуаций, в которых ранее применялась функция CDK, можно использовать рабочие процессы vRealize Orchestrator.

## ExecutePowerShellScript

Выполняет сценарий PowerShell, хранящийся в диспетчере моделей под указанным именем.

Перед использованием действия ExecutePowerShellScript необходимо сначала загрузить сценарий, который требуется выполнить, в диспетчер модели с помощью команды CloudUtil File-Import.

Таблица 1-39. Входные параметры действия ExecutePowerShellScript

Аргумент	Тип	Описание
ScriptName	String	Имя в диспетчере моделей сценария для выполнения.
ScriptVersion	Object	Версия в диспетчере моделей сценария для выполнения (дополнительно). Значение по умолчанию равно 0,0.
MachineId	Guid	Если указано, загружается компьютер и все его свойства передаются в сценарий (дополнительно).



Таблица 1-39. Входные параметры действия ExecutePowerShellScript (продолжение)

Аргумент	Тип	Описание
Arguments	Dictionary<string,string>	Дополнительные аргументы для передачи в сценарий. Если указан идентификатор MachineId и существует свойство компьютера с тем же именем в качестве аргумента (без учета регистра), то значение свойства компьютера переопределяет значение аргумента.
PSModules	IEnumerable<string>	Модули загружены в среду выполнения PowerShell во время выполнения команды (дополнительно). Этот параметр доступен только в области свойств, а не конструктора.

Таблица 1-40. Выходные параметры действия ExecutePowerShellScript

Аргумент	Тип	Описание
Output	Collection<PSObject>	Выходное значение сценария (при наличии). Создает исключение в случае ошибки.

При получении сообщения об ошибке Тип `PSObject` не определен в консоли vRealize Automation Designer во время работы с выходным значением параметра `ExecutePowerShellScript` выполните следующие действия.

1. Нажмите **Импорты** в левом нижнем углу области конструктора.
2. Выберите сборку **System.Management.Automation**.

## ExecuteSshScript

Выполняет сценарий SSH, хранящийся в диспетчере моделей под указанным именем.

Перед использованием действия `ExecuteSshScript` необходимо сначала загрузить сценарий, который требуется выполнить, в диспетчер модели с помощью команды `CloudUtil File-Import`.

Таблица 1-41. Входные параметры действия ExecuteSshScript

Аргумент	Тип	Описание
ScriptName	String	Имя в диспетчере моделей сценария для выполнения.
Host	String	Имя сервера, по которому выполняется сценарий.
Username	String	Имя пользователя для использования при подключении к узлу.
Password	String	Пароль для использования при подключении к узлу.
ScriptVersion	Object	Версия в диспетчере моделей сценария для выполнения (дополнительно). Значение по умолчанию равно 0,0.
Timeout	TimeSpan	Период времени до окончания ожидания выполнения сценария (дополнительно). По умолчанию установлено значение 30 минут.

Таблица 1-42. Выходные параметры действия ExecuteSshScript

Аргумент	Тип	Описание
EnvironmentVariables	Dictionary<string, string>	Результат исключения скрипта (при наличии).

## GetMachineName

Получает имя компьютера.

Таблица 1-43. Входные параметры действия GetMachineName

Аргумент	Тип	Описание
MachineId	Guid	Компьютер, имя которого необходимо извлечь.

Таблица 1-44. Выходные параметры действия GetMachineName

Аргумент	Тип	Описание
MachineName	String	Имя компьютера, определенного в MachineId.

## GetMachineOwner

Получает имя владельца компьютера.

Таблица 1-45. Входные параметры действия GetMachineOwner

Аргумент	Тип	Описание
MachineId	Guid	Компьютер, владельца которого необходимо извлечь.

Таблица 1-46. Выходные параметры действия GetMachineOwner

Аргумент	Тип	Описание
Owner	String	Владелец компьютера, определяемый в MachineId. Значение равно NULL при отсутствии владельца.

## GetMachineProperties

Получает список настраиваемых свойств, связанных с компьютером.

Таблица 1-47. Входные параметры действия GetMachineProperties

Аргумент	Тип	Описание
MachineId	Guid	Компьютер, свойства которого необходимо извлечь.

Таблица 1-48. Выходные параметры действия GetMachineProperties

Аргумент	Тип	Описание
Properties	Dictionary<string, string>	Список свойств компьютера. Значения возвращаются расшифрованными, если они были сохранены в зашифрованном виде.

## GetMachineProperty

Получает значение указанного свойства для компьютера.

Таблица 1-49. Входные параметры действия GetMachineProperty

Аргумент	Тип	Описание
Machinelid	Guid	Компьютер, с которого извлекается свойство.
PropertyName	String	Имя свойства, значение которого необходимо вернуть.
IsRequired	bool	Если свойство не требуется и не найдено, действие создает исключение, иначе возвращает значение NULL.

Таблица 1-50. Выходные параметры действия GetMachineProperty

Аргумент	Тип	Описание
PropertyValue	String	Значение свойства, указанного в PropertyName. Возвращается расшифрованное значение, если оно было сохранено в зашифрованном виде.

## GetScriptFromName

Получает содержимое сценария, хранящегося в диспетчере моделей под указанным именем.

Таблица 1-51. Входные параметры действия GetScriptFromName

Аргумент	Тип	Описание
ScriptName	String	Имя в диспетчере моделей сценария для извлечения.
ScriptVersion	Object	Версия в диспетчере моделей сценария для извлечения (дополнительно). Значение по умолчанию равно 0,0.

Таблица 1-52. Выходные параметры действия GetScriptFromName

Аргумент	Тип	Описание
ScriptContent	String	Содержимое сценария, определенного ScriptName.

## InvokePowerShell

Выполнение команды PowerShell.

Таблица 1-53. Входные параметры действия InvokePowerShell

Аргумент	Тип	Описание
CommandText	String	Команда для выполнения.
Arguments	IEnumerable<string>	Добавляет аргументы команде (дополнительно).
Input	IEnumerable	Входной конвейер (дополнительно).
IsScript	bool	Указывает, является ли CommandText сценарием (дополнительно). По умолчанию используется значение «Ложь». Этот параметр доступен только в области свойств, а не конструктора.
Parameters	Collection	Коллекция пар имя-значение, передаваемых в качестве параметров в сценарий PowerShell (дополнительно). Этот параметр доступен только в области свойств, а не конструктора.

Таблица 1-53. Входные параметры действия InvokePowerShell (продолжение)

Аргумент	Тип	Описание
PowerShellVariables	Collection	Переменные копируются в среду выполнения PowerShell (дополнительно). Этот параметр доступен только в области свойств, а не конструктора.
PSModules	IEnumerable<string>	Модули загружены в среду выполнения PowerShell во время выполнения команды (дополнительно). Этот параметр доступен только в области свойств, а не конструктора.
Runspace	Runspace	Создание пространства выполнения PowerShell и его предоставление в этот аргумент позволяет повторно использовать одно пространство выполнения в нескольких вызовах PowerShell, что может привести к повышению производительности, (дополнительно). Этот параметр доступен только в области свойств, а не конструктора.

Таблица 1-54. Выходные параметры действия InvokePowerShell

Аргумент	Тип	Описание
Output	Collection<PSObject>	Выходное значение команды (при наличии). Создает исключение в случае ошибки.
Errors	Collection<ErrorRecord>	Ошибки, возникающие в результате исключения (при наличии).

При получении сообщения об ошибке Тип `PSObject` не определен в консоли vRealize Automation Designer во время работы с выходным значением параметра `ExecutePowerShellScript` выполните следующие действия.

1. Нажмите **Импорты** в левом нижнем углу области конструктора.
2. Выберите сборку **System.Management.Automation**.

## InvokeSshCommand

Выполняет команду SSH.

Таблица 1-55. Входные параметры действия InvokeSshCommand

Аргумент	Тип	Описание
CommandText	String	Команда для выполнения.
Host	String	Имя сервера, по которому выполняется команда.
Username	String	Имя пользователя для использования при подключении к узлу.
Password	String	Пароль для использования при подключении к узлу.
Timeout	TimeSpan	Период времени до окончания ожидания выполнения команды (дополнительно). По умолчанию установлено значение 30 минут.

Таблица 1-56. Выходные параметры действия InvokeSshCommand

Аргумент	Тип	Описание
EnvironmentVariables	Dictionary<string, string>	Выходное значение команды (при наличии). Создает исключение в случае ошибки.

## LogMachineEvent

Записывает событие компьютера в журнал пользователя, который доступен владельцу компьютера.

Таблица 1-57. Входные параметры действия LogMachineEvent

Аргумент	Тип	Описание
Machinelid	Guid	Компьютер, создающий событие для журнала.
Message	String	Сообщение для записи в журнал пользователя.
Type	String	Выберите тип сообщения из раскрывающегося списка («Информация», «Предупреждение», «Ошибка»)

## LogMessage

Записывает сведения в журнал Distributed Execution Manager.

Таблица 1-58. Входные параметры действия LogMessage

Аргумент	Тип	Описание
Message	String	Сообщение для записи в журнал DEM.
MessageCategory	String	Выберите категорию из раскрывающегося списка меню ( <b>Отладка</b> , <b>Ошибка</b> , <b>Информация</b> , <b>Трассировка</b> ) или введите настраиваемую категорию.
MessageSeverity	String	Выберите серьезность в раскрывающемся меню, которая привязана к списку «Серьезности», поставляемому в System.Diagnostics.TraceEventType.

## RunProcess

Выполняет процесс на том же компьютере, что и диспетчер DEM, выполняющий это действие.

**Примечание** vRealize Automation не может представлять интерфейс из процессов, запущенных действием RunProcess для пользователя, поэтому эти процессы должны быть не интерактивными. Чтобы не оставлять потерянные процессы на компьютере DEM, процессы должны быть также самопрекращающимися.

Таблица 1-59. Входные параметры действия RunProcess

Аргумент	Тип	Описание
Command	String	Путь к исполняемому файлу для запуска на компьютере DEM.
WorkingDirectory	String	Рабочий каталог, в котором необходимо запустить процесс (дополнительно).

Таблица 1-59. Входные параметры действия RunProcess (продолжение)

Аргумент	Тип	Описание
Arguments	String	Список аргументов командной строки для перехода к команде (дополнительно).
WaitForExit	bool	Если значение верно, перед продолжением рабочий процесс ожидает завершения процесса (дополнительно). По умолчанию используется значение «Ложь».  Этот параметр доступен только в области свойств, а не конструктора.

## SendEmail

Отправляет сообщение электронной почты на указанный набор адресов.

Таблица 1-60. Входные параметры действия SendEmail

Аргумент	Тип	Описание
To	IEnumerable<string>	Список адресов, на которые отправляется сообщение электронной почты.
From	String	Адрес, который заполняется в поле «От» сообщения электронной почты.
Subject	String	Тема сообщения электронной почты.
Body	String	Основной текст сообщения электронной почты.
Host	String	Имя узла или IP-адрес SMTP-сервера исходящей почты.
Port	Integer	SMTP-порт на сервере, указанном в узле.  Этот параметр доступен только в области свойств, а не конструктора.
CC	IEnumerable<string>	Адрес или список адресов для копирования в сообщение электронной почты (дополнительно).  Этот параметр доступен только в области свойств, а не конструктора.
Bcc	IEnumerable<string>	Адрес или список адресов для слепого копирования в сообщение электронной почты (дополнительно).  Этот параметр доступен только в области свойств, а не конструктора.
EnableSsl	bool	Указывает, требуется ли использовать SSL (дополнительно).  Этот параметр доступен только в области свойств, а не конструктора.
UserName	String	Имя пользователя, под которым необходимо выполнить проверку подлинности с помощью SMTP-сервера, указанного в Host.  Этот параметр доступен только в области свойств, а не конструктора.
Password	String	Пароль пользователя, указанного в UserName.  Этот параметр доступен только в области свойств, а не конструктора.

## SetMachineProperty

Создает или обновляет настраиваемое свойство на компьютере.

Таблица 1-61. Входные параметры действия SetMachineProperty

Аргумент	Тип	Описание
MachineId	Guid	Компьютер, на котором необходимо создать или обновить настраиваемое свойство.
PropertyName	String	Название создаваемого или обновляемого свойства
PropertyValue	String	Значение, с которым необходимо создать или обновить настраиваемое свойство.
IsEncrypted	bool	Указывает, зашифровано ли значение свойства (дополнительно).
IsHidden	bool	Указывает, является ли свойство скрытым (дополнительно).
IsRuntime	bool	Указывает, предоставляет ли запрашивающий пользователь значение свойства во время запроса, аналог пометке «Запросить пользователя» в консоли vRealize Automation, (дополнительно).

## SetWorkflowResult

Устанавливает состояние внешнего рабочего процесса со значением «Завершено» или «Неудача», которое необходимо принять на обработку в настройках ExternalWF.xml

Таблица 1-62. Входные параметры действия SetWorkflowResult

Аргумент	Тип	Описание
WorkflowId	Guid	Рабочий процесс, для которого необходимо установить состояние.
Next State	WorkflowState	Выберите в раскрывающемся меню <b>Завершено</b> или <b>Неудача</b> .