

image not found or type unknown



Windows PowerShell Workflow — нововведение Windows Management Framework версии 3.0. В широком смысле, рабочий процесс — специализированная разновидность сценариев Windows PowerShell. Он описывает набор операций, одни из которых должны выполняться в определенной последовательности, тогда как другие могут выполняться параллельно.

Windows PowerShell Workflow устанавливается вместе с Windows Server 2012 и Windows 8. Он также доступен для Windows 7, Windows Server 2008 и Windows Server 2008 R2. Для запуска рабочего процесса обязательно нужна одна из этих ОС. Однако адресатом рабочего процесса, т. е. системой, в который запускаются задания, может быть любая версия Windows, в зависимости от того, какое именно задание вы пытаетесь выполнить.

При написании рабочих процессов используется более или менее стандартный синтаксис Windows PowerShell. Однако существует несколько несовместимостей рабочих процессов и традиционных сценариев Windows PowerShell. Windows PowerShell Workflow активно использует функции удаленного выполнения (remoting) Windows PowerShell. (Ознакомьтесь с моим бесплатным руководством «Secrets of PowerShell Remoting», если вам нужно обучение основам удаленного выполнения или рекомендации по настройке.)

Как функционирует рабочий процесс

Заглянув внутрь (а в последующих статьях этой серии мы туда заглянем и поглубже), вы обнаружите, что рабочий процесс Windows PowerShell, на самом деле, выполняется не Windows PowerShell. Вместо этого, Windows PowerShell преобразовывает ваш сценарий в XAML и передает его Windows Workflow Foundation (WWF), являющейся частью Microsoft .NET Framework. WWF принимает XAML-данные и уже по-настоящему выполняет процесс.

То, что рабочий процесс выполняется WWF, а не Windows PowerShell, дает кое-какие интересные возможности и вносит не так уж мало сложностей. Например, в WWF предусмотрены контрольные точки для отслеживания хода рабочего процесса. Поэтому, если компьютер, на котором выполняется рабочий процесс, по каким-либо причинам завершит работу (например, его выключат), то после повторного запуска

компьютера выполнение процесса продолжится с того места, на котором оно остановилось.

Кроме того, можно вручную приостанавливать и возобновлять рабочие процессы. Например, у вас может быть рабочий процесс, который выполняет определенный набор заданий до момента, когда требуется ручное вмешательство. Вы можете сделать так, чтобы в этот момент он приостанавливал работу и автоматически отправлял сообщение об изменении статуса по электронной почте. После выполнения ручных операций можно возобновить рабочий процесс и позволить ему выполняться дальше.

Если нужно, рабочие процессы могут выполнять операции параллельно. Например, если у вас имеется набор независимых заданий, которые можно выполнять в любом порядке, то их можно запустить примерно в одно и то же время. Это сократит общее время выполнения рабочего процесса, а значит, повысит эффективность. Кроме того, WWF отслеживает каждый этап рабочего процесса. Это значит, что можно сгенерировать подробный журнал аудита с данными о том, какие задания выполнялись, очень полезный при устранении неполадок.

Рабочие процессы не могут делать все, что делают сценарии Windows PowerShell. Как вы узнаете из этой серии статей, все, что нельзя преобразовать в нечто «понятное» WWF, выходит за их границы. Конечно, имеется ряд функций Windows PowerShell, которые не преобразовываются. Таким образом, рабочий процесс не является ни подмножеством, ни надмножеством Windows PowerShell. Это пересекающееся множество: рабочие процессы имеют кое-какие дополнительные возможности по сравнению с Windows PowerShell, и наоборот. Такое несоответствие — один из наиболее сложных аспектов рабочих процессов.

Не единственно возможное решение

Рабочий процесс не является единственно верным решением любой задачи. Ажиотаж вокруг рабочих процессов подпитывается группой Microsoft, отвечающей за продукт, и многочисленными независимыми энтузиастами, создающими впечатление, что все следует реализовывать в виде рабочих процессов. Не поддавайтесь их убеждениям, поскольку у рабочих процессов значительно более высокая кривая обучения. Кроме того, они сложнее для устранения неполадок и отладки, и необходимы не всегда.

Например, если нужно просто параллельно запустить задание на нескольких компьютерах, можно сделать это с помощью обычного удаленного выполнения.

Можно также воспользоваться фоновыми заданиями в стиле второй версии Windows PowerShell (которые остались доступными и в версии 3).

Если нужно просто выполнить последовательность заданий, которая перезапустит целевой компьютер и дождется, когда он возобновит работу, чтобы продолжить работу сценария, можно сделать это с помощью обновленной в версии 3 команды Restart-Computer (поддерживающей параметр -Wait). Хотя, как вы увидите, рабочие процессы действительно предоставляют мощные и в чем-то уникальные возможности, они не обязательно являются самым простым вариантом решения любой возможной задачи.

Анатомия рабочего процесса

Как и функция или командлет, рабочий процесс — это тип команды Windows PowerShell. После создания его, как правило, просто запускают аналогично любой другой команде. Кроме того, можно выполнять его по расписанию или передать на удаленный компьютер и запустить на нем. Во время выполнения Windows PowerShell преобразовывает его в формат WWF и обращается к WWF, чтобы по-настоящему запустить его.

Как и другие типы команд, рабочие процессы принимают параметры, позволяющие настраивать их поведение во время выполнения. Имеются различные типовые параметры, определяющие то или иное поведение процессов.

В рабочем процессе можно определить набор операций (activities), выполняемых в определенной последовательности или параллельно друг другу. Можно комбинировать последовательности и параллельные блоки столько раз, сколько требуется. Тонкость в том, что каждая операция должна представлять собой нечто «понятное» WWF. И здесь уже действительно начинаются сложности.

Группа разработки Windows PowerShell подготовила «версии» большинства основных командлетов Windows PowerShell для WWF. Это означает, что можно использовать эти командлеты прямо в рабочих процессах. При преобразовании в формат WWF такие командлеты замещаются их WWF-эквивалентами и работают ожидаемым образом.

В идеальном мире каждый командлет Windows PowerShell имел бы эквивалент в WWF. Тогда рабочие процессы выглядели бы во многом аналогично знакомым сценариям Windows PowerShell, которые мы пишем уже много лет. Однако мы живем в несовершенном мире. На самом деле, большинство командлетов не имеет

WWF-эквивалентов. Это особенно верно в отношении командлетов, относящихся к Exchange Server, SharePoint и Windows Server 2012 (но не относится к базовым командлетам Windows PowerShell).

В WWF нет встроенных средств выполнения таких командлетов. Их приходится обертывать тем, что «понятно» WWF: InlineScript. Эта специальная операция, по сути, заключается в том, что WWF просто запускает Windows PowerShell. Содержимое операции InlineScript передается PowerShell.

Каждая операция InlineScript — независимый, изолированный и отсоединенный объект. Операции InlineScript не могут совместно использовать одну и ту же информацию за исключением данных, хранящихся в каком-либо доступном для всех них месте (например, базе данных).

Это значительно усложняет написание рабочих процессов. Однако в настоящее время в основном придется иметь дело именно с такими операциями, поскольку большинство командлетов Windows PowerShell пока что не имеет WWF-версий.