

User Guide

Release Management for Visual Studio 2013

ABOUT THIS GUIDE

The User Guide for the release management features is for administrators and users. The following related documents for release management are also available:

- Installation guide, which describe setup and configuration of release management components.
- Release Notes, which describe the new and changed features and fixed problems in the latest version of the software

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2013 Microsoft. All rights reserved.

Microsoft, Visual Studio® and Windows Azure are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.

Contents

ABOUT THIS GUIDE	2
1 OVERVIEW	8
1.1 INTRODUCTION	8
1.2 MAIN COMPONENTS	9
1.2.1 Release Management Client for Visual Studio 2013.....	9
1.2.2 Release Management Server for Team Foundation Server 2013.....	9
1.2.3 Microsoft Deployment Agent 2013	9
1.2.4 Tools	10
2 CONFIGURATION.....	11
2.1 FIRST TIME CONFIGURATION	11
2.2 CONFIGURE SYSTEM SETTINGS	11
2.3 CONFIGURE USERS.....	12
2.4 CONFIGURE GROUPS	13
2.4.1 “Everyone”	13
2.4.2 Add AD Groups.....	13
2.4.3 Add TFS Groups	13
2.4.4 Edit Groups.....	13
2.5 CONFIGURE TFS CONNECTIONS	14
2.6 CONFIGURE PICK LISTS.....	14
2.6.1 Configuring Stage Types.....	14
2.6.2 Configure Technology Types	15
2.7 CONFIGURE ACTIONS AND TOOLS	15
2.7.1 Create a new Action.....	15
2.7.2 Create a new Category from an Action	15
2.7.3 Create a new Tool from an Action	15
2.8 CONFIGURE ENVIRONMENTS AND SERVERS.....	16
2.8.1 Create a new Environment	16
2.8.2 Create a new Server from an Environment.....	16
2.8.3 Scan for new Servers.....	18
2.9 CONFIGURING RELEASE PATHS.....	19
2.10 CONFIGURING RELEASE TEMPLATES AND COMPONENTS	20
2.10.1 Create a new Release Template.....	20
2.10.2 Create a new Component from a Release Template.....	20

2.10.3	<i>Configure Component in a Release Template</i>	23
2.10.4	<i>Configure Security in a Release Template.....</i>	23
2.11	CONFIGURE RELEASES	23
2.11.1	<i>Open a Release Template</i>	23
2.11.2	<i>Create Release from Release Template</i>	24
2.12	DEPLOY A RELEASE	25
2.12.1	<i>Open a Draft Release</i>	25
2.12.2	<i>Start a Release.....</i>	25
2.12.3	<i>Accept Deployment in Stage.....</i>	25
2.12.4	<i>Retry a Failed Deployment.....</i>	25
2.12.5	<i>Validate Deployment</i>	26
2.12.6	<i>Approve Release</i>	26
2.13	TRIGGER A RELEASE FROM TEAM BUILD	26
2.13.1	<i>Pre-requisites</i>	26
2.13.2	<i>Visual Studio 2010, Visual Studio 2012 and Visual Studio 2013</i>	27
2.14	INCLUDE RELEASE MANAGEMENT CUSTOM BUILD LOGIC INTO A BUILD TEMPLATE.....	27
2.14.1	<i>Add arguments to custom Build Template</i>	27
2.14.2	<i>Updating Metadata Argument.....</i>	28
3	RELEASE TEMPLATES	31
3.1	OVERVIEW	31
3.2	PROPERTIES	31
3.3	DEPLOYMENT SEQUENCE.....	32
3.4	CONFIGURATION VARIABLES.....	34
3.5	COPYING A STAGE TO ANOTHER.....	36
3.6	LOCKING MECHANISM	36
3.7	ROLLBACK AND ROLLBACK ALWAYS MECHANISM	37
3.7.1	<i>Example 1</i>	37
3.7.2	<i>Example 2</i>	38
4	RELEASES.....	39
4.1	OVERVIEW	39
4.2	PROPERTIES	39
4.3	DEPLOYMENT SEQUENCE.....	40
4.4	DEPLOYMENT LOG.....	41
4.5	PENDING APPROVAL REQUESTS.....	41
5	RELEASE EXPLORER.....	42
5.1	OVERVIEW	42
5.2	OPERATIONS	43

5.2.1	<i>Viewing Component Details</i>	43
5.2.2	<i>Release Stage</i>	44
5.2.3	<i>Approving a Release</i>	44
5.2.4	<i>Rejecting a Release</i>	45
5.2.5	<i>Reassigning a Release</i>	46
5.3	PREVIOUSLY APPROVED RELEASES	46
APPENDIX		47
A1.	SECURITY MANAGEMENT	48
	RELEASE PATH SECURITY	48
	RELEASE TEMPLATE SECURITY	50
	RELEASE SECURITY	54
	STAGE TYPE SECURITY	57
	ENVIRONMENT SECURITY	59
	SERVER SECURITY	60
	INVENTORY SECURITY	60
	ACTION AND COMPONENT SECURITY	61
A2.	ACTIONS CATALOG	62
	WINDOWS AZURE	62
	<i>Start a Windows Azure VM</i>	62
	<i>Stop a Windows Azure VM</i>	62
	IIS	62
	<i>Create Application Pool</i>	62
	<i>Configure Application Pool</i>	64
	<i>Remove Application Pool</i>	65
	<i>Recycle Application Pool</i>	65
	<i>Start Application Pool</i>	65
	<i>Stop Application Pool</i>	66
	<i>Create Web Site</i>	66
	<i>Configure Web Site</i>	66
	<i>Remove Web Site</i>	67
	<i>Start Web Site</i>	67
	<i>Stop Web Site</i>	67
	<i>Restart Web Site</i>	68
	<i>Create Web Application</i>	68
	<i>Configure Web Application</i>	68
	<i>Remove Web Application</i>	69
	<i>Create Virtual Directory</i>	69

<i>Configure Virtual Directory</i>	<i>69</i>
<i>Remove Virtual Directory</i>	<i>70</i>
MS-SQL	70
<i>Create SQL Database</i>	<i>70</i>
<i>Drop SQL Database.....</i>	<i>70</i>
<i>Backup SQL Database</i>	<i>71</i>
<i>Restore SQL Database.....</i>	<i>71</i>
WINDOWS OS	71
<i>Copy File or Folder</i>	<i>71</i>
<i>Create Folder.....</i>	<i>71</i>
<i>Delete File(s) or Folder</i>	<i>71</i>
<i>Move File or Folder.....</i>	<i>72</i>
<i>Rename File or Folder.....</i>	<i>72</i>
<i>Modify File(s) or Folder Attributes</i>	<i>72</i>
<i>Modify File(s) or Folder Owner</i>	<i>73</i>
<i>Create Environment Variable</i>	<i>73</i>
<i>Modify Environment Variable.....</i>	<i>73</i>
<i>Delete Environment Variable.....</i>	<i>74</i>
<i>Create Windows Registry Key.....</i>	<i>74</i>
<i>Modify Windows Registry Key</i>	<i>74</i>
<i>Delete Windows Registry Key</i>	<i>75</i>
<i>Create Windows Registry Value</i>	<i>75</i>
<i>Modify Windows Registry Value</i>	<i>76</i>
<i>Delete Windows Registry Value</i>	<i>77</i>
<i>Run Command Line</i>	<i>77</i>
<i>Run Command Line as User</i>	<i>77</i>
<i>Kill Windows Process</i>	<i>77</i>
WINDOWS SERVICES	78
<i>Create Service.....</i>	<i>78</i>
<i>Configure Service</i>	<i>79</i>
<i>Configure Service Credentials.....</i>	<i>79</i>
<i>Delete Service</i>	<i>79</i>
<i>Start Service.....</i>	<i>80</i>
<i>Stop Service.....</i>	<i>80</i>
<i>Restart Service</i>	<i>80</i>
INI FILE	80
<i>Modify Section Name</i>	<i>80</i>

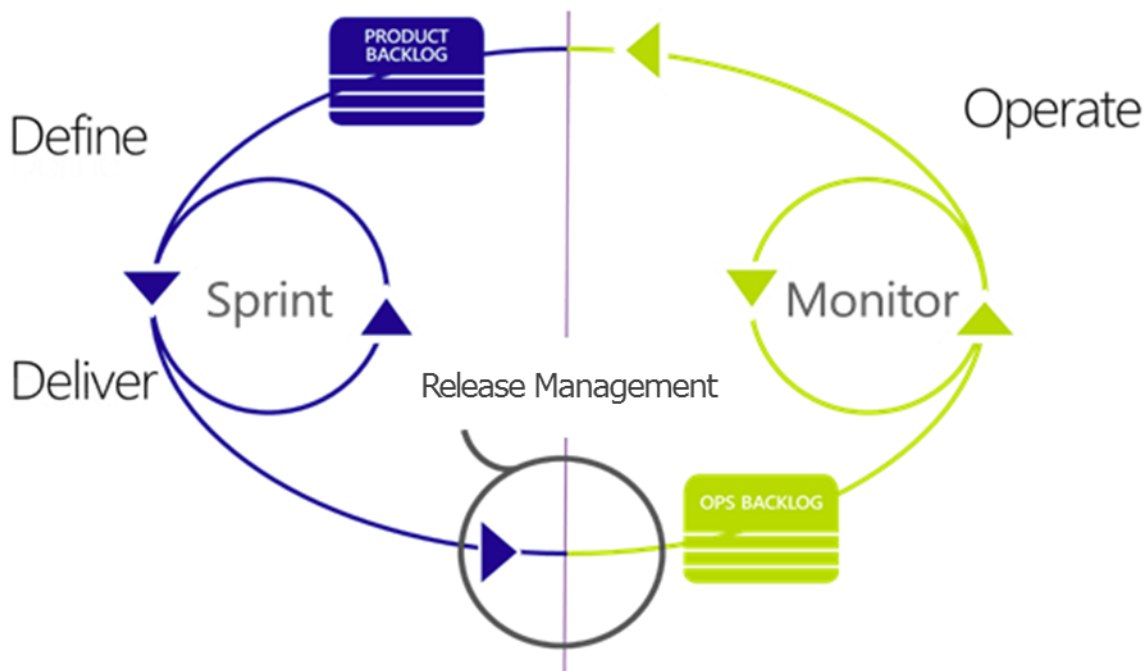
<i>Remove Section</i>	81
<i>Create Key and Value</i>	81
<i>Modify Key and Value</i>	81
<i>Remove Key and Value</i>	82
A3. TOOLS CATALOG	83
DATABASE DEPLOYMENT AGENT – EXECUTE SCRIPT	83
REPORTING SERVICES DEPLOYMENT AGENT	83
MSI DEPLOYMENT AGENT	84
XCOPY DEPLOYMENT AGENT	85
COMMAND LINE RUNNER	85
WINDOWS REGISTRY MANAGER	86
WINDOWS COMMON IO	88
WINDOWS SERVICES MANAGER	89
DACPAC DATABASE DEPLOYMENT AGENT	90
IIS DEPLOYMENT AGENT	91
WINDOWS AZURE VM MANAGER	95
XBAP DEPLOYER	96
MICROSOFT TEST MANAGER (MTM) COMMAND-LINE UTILITY	96
INI FILE MANAGER	99
WINDOWS PROCESS	100

1 OVERVIEW

1.1 Introduction

Release management is a continuous deployment solution that makes release cycles repeatable, visible and more efficient by automating deployments through every environment from Team Foundation Server (TFS) until production. With pre-defined release paths, release management triggers deployments upon approval, assembles all the components of your application, moves them to the target servers and installs all of them in one transaction. Once the installation is successful, release management can execute automated tests or data generation scripts specified for your application. The same steps are repeated until the application is approved and goes to the next environment.

Based on a business-approval workflow and a flexible and centralized configuration, release management is an orchestration platform that improves coordination and communication between development, operations and quality assurance to decrease issues inherent to it such as inefficiency, errors, frustration, high costs and delays.



1.2 Main Components

The following figure shows the main components that require deployment in order for release management components to function effectively.

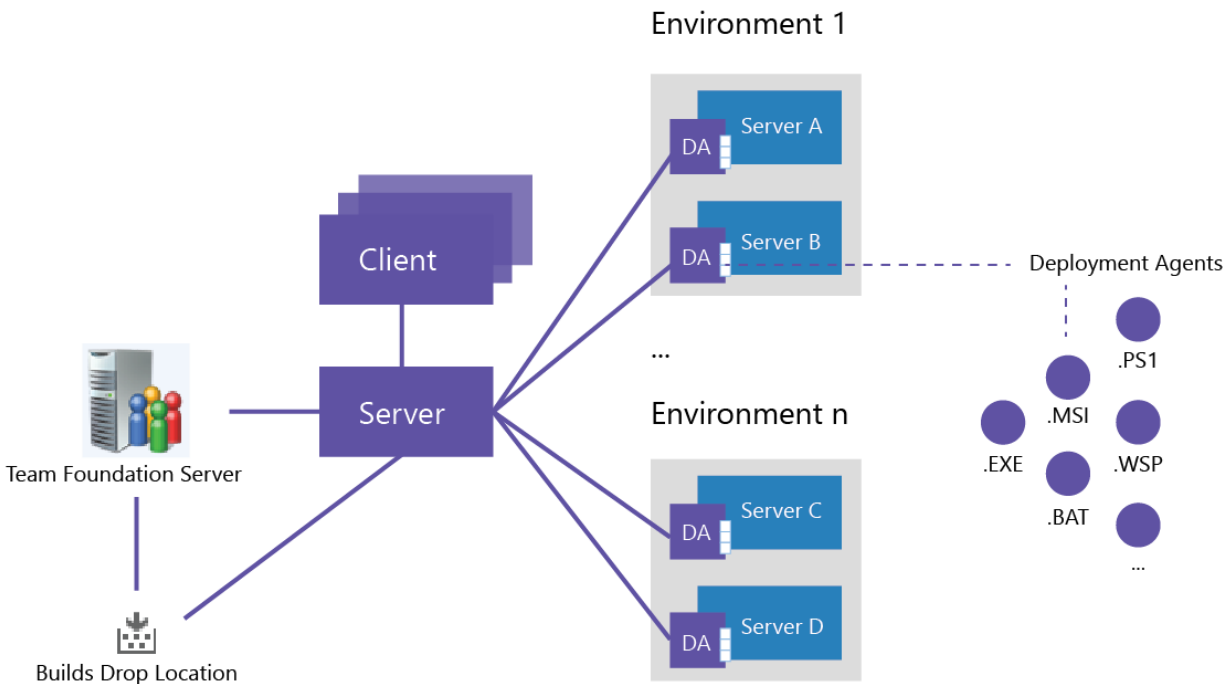


Figure 1 –Release Management Deployment Overview

1.2.1 Release Management Client for Visual Studio 2013

There are two Client components. The Windows client is a Windows Presentation Foundation (WPF) application that serves as the main interface point to manage release information. This is the component with which most users interact. The Web client is used to act on Approval Requests. This is the interface to which users are directed when following links in e-mail notifications.

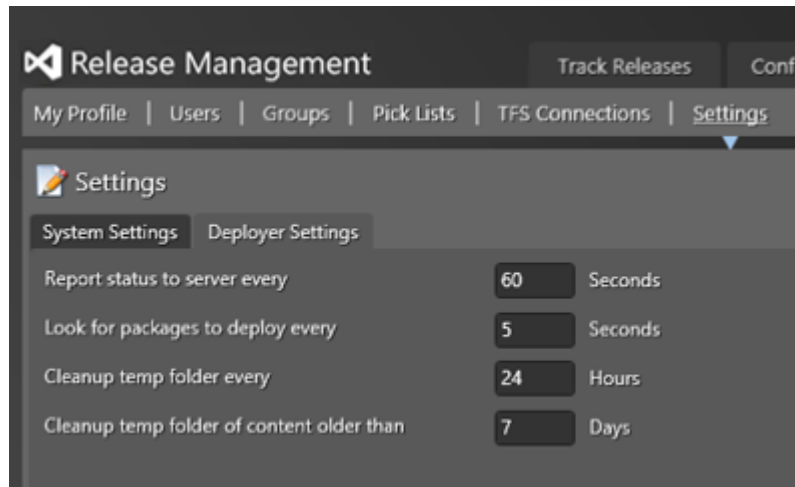
1.2.2 Release Management Server for Team Foundation Server 2013

The Server component is the heart of release management components. It is a combination of Web and Windows Services that expose contracts used by all other components. The server component also contains a SQL Server database.

1.2.3 Microsoft Deployment Agent 2013

The deployment agent component is a Windows service that lives on the Target Servers where your application components managed by release management are installed.

The deployment agent can be configured in the Client by choosing **Administration, Settings, Deployer Settings**.



The following options are available:

- **Report status to server every x seconds** – Used to define the interval for running the **Heartbeat** service. The Heartbeat service is used to communicate with the Server to report the status of the deployment agent and run various tests.
- **Look for packages to deploy every x seconds** – Used to define the interval for running the **Deployment** service. The Deployment service will look for new packages to deploy.
- **Cleanup temp folder every x hours** – Used to define an interval for running the **Cleanup** service. The Cleanup service is used to clear unneeded content from the deployment agent temp folder.
- **Cleanup temp folder of content older than x days** – Used to define the age of the folder contents to clean. Any folders, inside the deployment agent temp folder, older than this option will be removed from disk.

1.2.4 Tools

The Tools are components that assist in the execution of various deployment scenarios, such as:

- Installing a version of a component to a specific environment
- Uninstalling a previous version of a component before a re-deployment
- Deploying reports to Microsoft SQL Reporting Services
- Moving files to specific locations

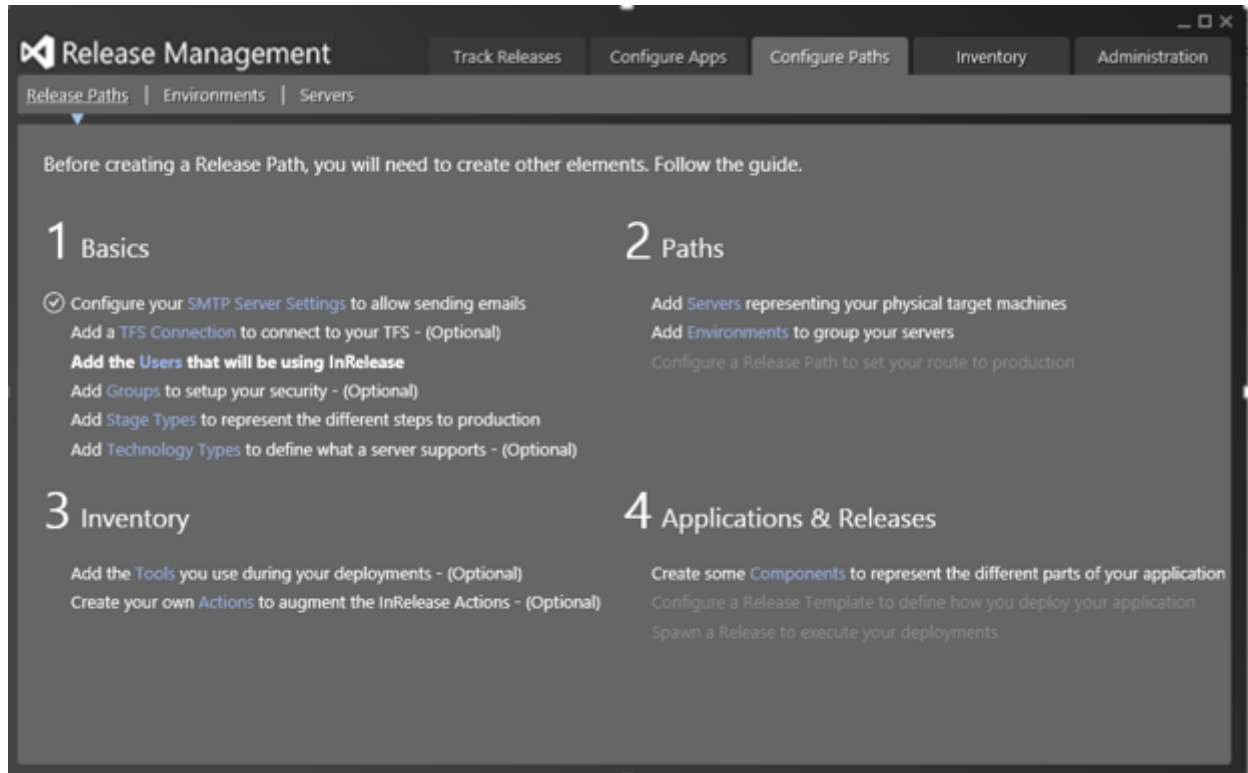
Deployment agent Extensible Model

The deployment agent uses an open and extensible model that allows for the integration of any type of “tool” that can be executed as a process. The Tool can be in the form of a batch file (.bat or .cmd), a PowerShell script (.ps1), an executable (.exe), etc.

2 CONFIGURATION

2.1 First Time Configuration

Using release management features for the first time can seem a bit difficult to grasp. In order to help with these first steps, there is a built-in guide. Each time you get to a screen to create an object which needs other steps to be completed (pre-requisites), the guide will be presented to let you know what you are missing.



Here's how to read this screen:

- The steps should be done in order from section 1 to 4
- Each line is a configuration step that needs to be completed
- A line starting with a check mark means that step has already been done at least once
- Some steps are optional (shown at the end of the line)
- The line in bold is the next mandatory step to execute
- A dimmed line means that step cannot be done just yet

2.2 Configure System Settings


Release Managers have the option to configure global system settings. This can be done by navigating to **Administration, Settings, System Settings**.

The following information can be configured in this section.

- **Maximum Resource File Size for Tools** – Maximum size (in KB) of a resource attached to a Tool.
- **Default Component Deployment Timeout** – Default deployment timeout used in new components.
- **TFS-Triggered Deployment Timeout** – Maximum time allowed to complete a deployment that has been triggered by a TFS build. If the threshold is reached, the deployment is considered a failure.
- **AD/TFS-Based Group Refresh Interval** – Refresh interval (in minutes) to automatically update Groups linked to AD or TFS groups. A value of 0 turns off the auto-refresh functionality. The size and complexity of your AD forest will impact the time it takes to completely refresh all synchronized groups. So, a larger value may work best.
- **SMTP Server Configuration** – Information required to connect to the local SMTP server in order for release management to send out email notifications.
 - **Server Name** – Fully qualified address of the SMTP server to use.
 - **Port** – Optional. Port to be used when connecting to the SMTP server to send email notifications.
 - **User Name** – Optional. Name of the identity to be used to authenticate against the SMTP services.
 - **User Password** – Password of the identity to be used to authenticate against the SMTP services. Mandatory if **User Name** is specified. Otherwise, optional.
 - **Sender Address** – Email address as it will appear in the email client. No replies are expected to be sent to this address. Therefore, the email or text can be anything.
- **Product License Information** – Displays current license information for server and client.

2.3 Configure Users

All users requiring access to the release management server must be configured. This can be done in two ways: manually or as part of group imported from TFS or Active Directory (AD). This section describes the manual steps to add and configure a user.

1. In the main menu, choose **Administration**. Then, in the sub-menu, choose **Manage Users**. The list of active users will appear.
2. Choose the **New** button in the Users list title bar. A form allowing you to define a new user will appear.
3. Choose the **ellipsis** button () beside the **Windows Account** textbox. A **Select User** dialog window will appear.
4. Find the Windows account you want to add and choose the **OK** button. The Name, Windows Account and Email fields will automatically be updated with information from Active Directory.
5. If the user is a release manager, select **Yes** to the field **Release Manager?**
6. If you want the user to receive email notifications, select **Yes** to the field **Receive Email Notification?**
7. Choose the **Save & Close** button in the User form title bar to persist your changes. The new user will now appear in the list of users.

2.4 Configure Groups

2.4.1 “Everyone”

The group named **Everyone** is a reserved group automatically added by the system. Some security parameters may be modified within this group. Please be aware that since the permissions are evaluated to “most permissive”, you will have to remove permissions to this group if you want to control them manually within other groups (for example: if you do not remove **Can Create Release Template** from Everyone and remove it from a specific group, all users will still be able to create a Release Template even if they are part of that specific group).

2.4.2 Add AD Groups

1. In the main menu, choose **Administration**. Then, in the sub-menu, choose **Manage Groups**. The list of active groups will appear.
2. Choose the arrow to the right of the **New** button to display more options. Choose the **New From AD** option. This will display the **Select Groups** dialog. This dialog is the common AD object selection dialog. Navigate through it to find the AD Groups you would like to synchronize with the release management Server.
3. Choose **OK** to confirm your AD Groups selection. The process will start and all members of the selected AD Groups will be created, all the selected AD Groups will have a corresponding RM Group created and these will be set to **Synched**. This will allow you to refresh their contents at a later time.

Note: The release management server monitors groups that are synchronized to AD and can automatically refresh their content based on a configurable interval. This interval can be configured under **Administration, Settings**.

2.4.3 Add TFS Groups

1. In the main menu, choose **Administration**. Then, in the sub-menu, choose **Manage Groups**. The list of active groups will appear.
2. Choose the arrow to the right of the **New** button to display more options. Choose the **New From TFS** option. This will display the **Available TFS Groups** dialog, allowing you to select the **Connection**, **Collection**, and **Team Project**.

Note: If the **New From TFS** option is disabled, it may be because there is no valid TFS Connection configured. Refer to section [Configure TFS Connections](#) for details.

3. Select at least one TFS Group and choose **Add** to start the process. All members of the selected TFS Groups will be created, all the selected TFS Groups will have a corresponding RM Group created and these will be set to **Synched**. This will allow you to refresh their contents at a later time.

Note: RM monitors groups that are synchronized to TFS and can automatically refresh their content based on a configurable interval. This interval can be configured under **Administration, Settings**.

2.4.4 Edit Groups

1. In the main menu, choose **Administration**. Then, in the sub-menu, choose **Manage Groups**. The list of active groups will appear.

2. Double-click the group you want to start editing.
3. Edit the group name and description.
4. In the **Security** tab, you can configure permissions for this Group to perform different actions in the system.
5. In the **Stage Type Security** tab, you can identify which Stage Type this Group has access to with its specific permissions.
6. Choose the **Save & Close** button in the Security Groups form title bar to persist your changes.

2.5 Configure TFS Connections

1. In the main menu, choose **Administration**. Then, in the sub-menu, choose **Manage TFS**. The list of TFS Connections currently configured in RM will be shown.
2. Choose the **New** button on the top of the list of TFS Connections.
3. Enter your **TFS URL** in the **Name or URL of the TFS** field.

Note: As the TFS field properties are modified, the end result is shown lower in the field **URL**.

4. Select the version of TFS, its Path, port number and protocol.
5. Next, the user to connect to TFS needs to be selected. The connection to TFS will be done using the supplied account **on behalf** of the currently logged in user (to preserve correct rights to TFS).

Note: To know which user to use, check this [article](#).

6. When all information is entered, choose **Verify** to validate the connection. If the connection succeeds, a green checkmark appears and the **Save** button is enabled.
7. Finally, in the title bar, choose the **Save** button to persist your changes.

2.6 Configure Pick Lists

2.6.1 Configuring Stage Types

Stage types are the logical steps required to bring a build from development to production (Development, QA, UAT, Volumetric tests, Production...).

1. In the main menu, choose **Administration**. Then, in the sub-menu, choose **Pick Lists**. The list of all available pick lists will appear.
2. In the list of pick lists, choose **Stage Type**. The list of all stage types will appear on the right side.
3. To edit a stage type, choose its description and modify it. To add a new stage type, choose the **Add** button and enter a description.
4. Choose the **Save** button to persist your changes.

2.6.2 Configure Technology Types

Technology types are informational values that allow the user to identify what kinds of technologies are used in your environments and servers. That information is not required in the process of release management.

1. In the main menu, choose **Administration**. Then, in the sub-menu, choose **Manage Pick Lists**. The list of all available pick lists will appear.
2. In the list of pick lists, choose **Technology Type**. The list of all technology types will appear on the right side.
3. To edit a technology type, choose its description and modify it. To add a new technology type, choose the **Add** button and enter a description.
4. Choose the **Save** button to persist your changes.

2.7 Configure Actions and Tools

2.7.1 Create a new Action

1. In the main menu, choose **Inventory**. Then, in the sub-menu, choose **Actions**. The list of active actions will appear.
2. Choose the **New** button in the Action list title bar. A form allowing you to define a new action will appear.
3. Enter the action name in the **Name** field.
4. You can choose a category from the **Categories** drop down list.
5. You can also select a tool from the **Tools** drop down list or create a new tool by choosing the **New** hyperlink. You can also edit existing custom tools by choosing the **Edit** hyperlink. It is not possible to edit any tools provided by release management.

2.7.2 Create a new Category from an Action

1. Choose the **New** hyperlink beside the **Categories** drop down list.
2. A popup will open so you can enter a category name.
3. Choose **OK** when you are done and the newly created category will appear in the drop down list and will be selected.

2.7.3 Create a new Tool from an Action

1. Choose the **New** hyperlink beside the Tools drop down list.
2. A form allowing you to define a new tool will appear.
3. Enter the tool name in the **Name** field.
4. Enter the command in the **Command** field.
5. Enter the arguments in **Arguments** field. Parameters will be generated automatically based on the tokens (start with `__` and ends with `__`) found in the arguments field.

6. In the **Resources** section, you can enter one or more resources (executables, scripts, etc.). These resources will then be available to be used in the command field described previously.

2.8 Configure Environments and Servers

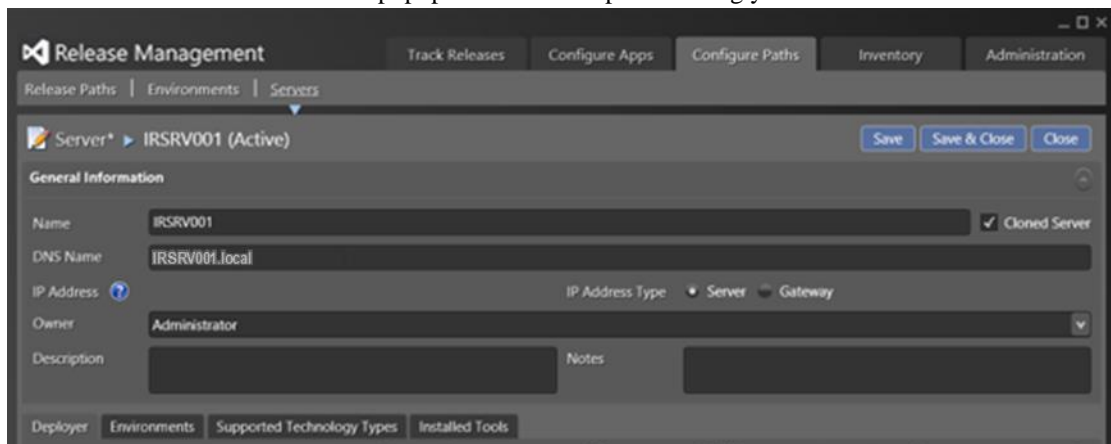
2.8.1 Create a new Environment

1. In the main menu, choose **Configure Paths**. Then, in the sub-menu, choose **Environments**. The list of active environments will appear.
2. Choose the **New** button in the Environments list title bar. A form allowing you to define a new environment will appear.
3. Enter the environment name in the **Name** field.
4. Select the environment owner from the **Owner** dropdown.
5. Go to the **Supported Technology Types** tab and select all technologies that are supported by the environment.
6. Go to the **Stage Type Security** tab and select all stage types that can use this environment.

2.8.2 Create a new Server from an Environment

1. Go to the **Servers** tab. This tab shows the list of servers associated to the environment. You have the choice of linking a new server or an existing server to the environment.

Choose the **New** button on top of the list of servers in order to create a new server that will be associated to the environment. A form within a popup window will open allowing you to define the new server.



The screenshot shows the 'Release Management' application interface. The 'Servers' tab is active, displaying a list of servers. The 'Server' button is highlighted, and the configuration form for 'IRSRV001 (Active)' is shown. The form includes the following fields and controls:

- Name:** IRSRV001
- DNS Name:** IRSRV001.local
- IP Address:** (with a help icon)
- IP Address Type:** Server (selected), Gateway
- Owner:** Administrator (dropdown menu)
- Description:** (text area)
- Notes:** (text area)
- Cloned Server:** (checkbox, checked)
- Buttons:** Save, Save & Close, Close

At the bottom of the form, there are tabs for 'Deployer', 'Environments', 'Supported Technology Types', and 'Installed Tools'.

2. Enter the server name in the **Name** field.
3. Enter the server fully qualified name in the **DNS Name** field.
4. Indicate whether or not this is a **Cloned Server**.

Cloned Server

This flag indicates whether or not the server is cloned. That is, it is a copy of another server. Cloned servers are special in that their IP Address must be static. This is the only way to uniquely identify them from other copies with the same characteristics.

Although servers can be registered in the system manually, we recommend they are configured using the **Scan for New** feature (see next section).

5. Select the server owner from the **Owner** dropdown.
6. Select the **IP Address Type**; either Server or Gateway. This will allow release management to recognize the deployment agent based on its DNS Name and the IP Address that it communicates.

IP Address Type

There are two ways a deployment agent can indicate its uniqueness in a network.

Server – indicates that the IP Address is that of the Server within its local network. This is the actual server's IP address and it is usually considered internal to the network to which the server belongs.

Gateway – indicates that the IP Address is that of the Gateway behind which the Server resides. This is commonly used when servers are behind firewalls or are using Network Address Translation (NAT) .

The IP Address Type is most used when dealing with “cloned” environments where the servers may be exact images but appear to be different to the outside world.

Select the IP Address Type that uniquely identifies your server configurations within your deployment agent network.

7. Select the **Drop Location Access** mechanism.

Drop Location Access

There are two mechanisms for a Deployment Agent to obtain files from the drop location.

The first one – *Directly using UNC Path* – is by accessing directly the drop location server via a UNC path. The account running the deployment agent must be granted security access to the drop location. Using this mechanism makes the file transfer very fast.

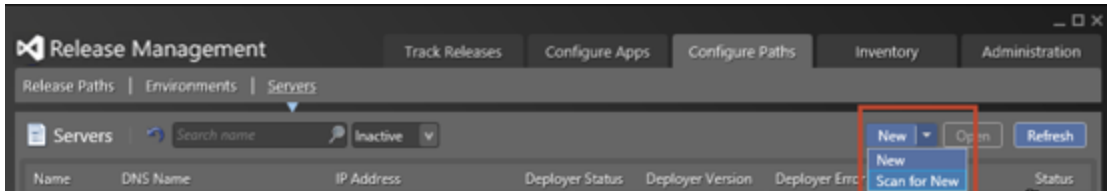
The second one – *Through Release Management Server over HTTP* – permits the deployment agent to be on a separate domain than the server where the drop location is. It is the responsibility of the release management server to get the files from the drop location (using a UNC path). The files are then transferred to the deployment agent over HTTP. This means that the account running the ReleaseManagementAppPool application pool used by the web services of the release management server needs to be granted security access to the drop location. The file transfer rates will be slower in this scenario.

8. Go to the Supported Technology Types tab and select all technologies that are supported by the server.
9. Choose the **Save and Close** button in the Server form title bar. The new server will now appear in the list of servers.
10. Repeat step 6 to 11 to associate more servers to the environment.
11. Choose the **Save & Close** button in the Environment form title bar to persist your changes. The new environment will now appear in the list of environments.

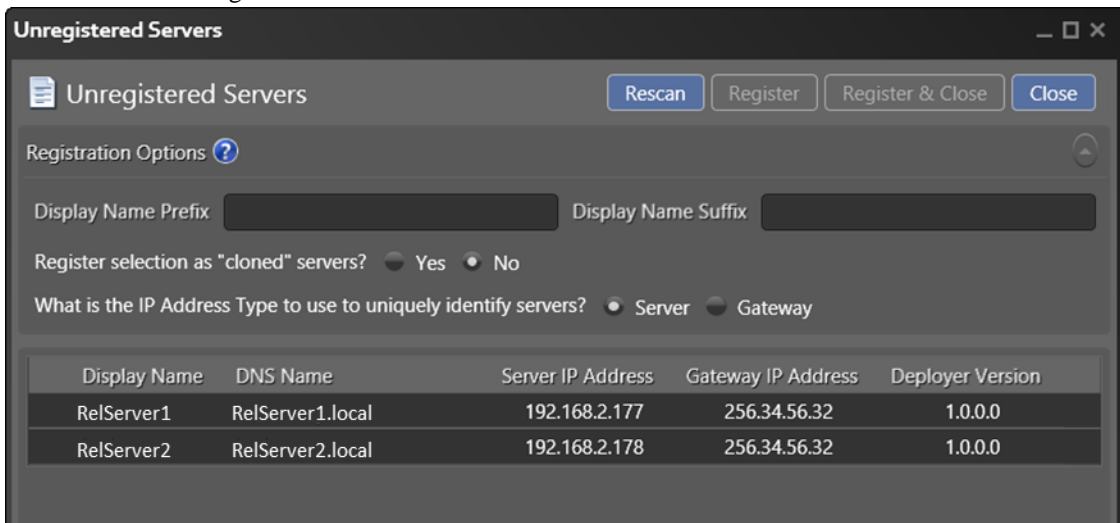
2.8.3 Scan for new Servers

If you have installed deployment agents before creating the corresponding Servers in release management, you can “scan” for them. Each deployment agent communicates its presence and shares certain details with the release management Server. The list of Unregistered Servers can be viewed from the Servers List view.

1. In the main menu, choose **Configure Paths**. Then, in the sub-menu, choose **Servers**. The list of active servers will appear.
2. Choose the down arrow of the **New** button. A menu will appear. Choose **Scan for New**. The **Unregistered Servers** dialog will appear.



3. In the **Unregistered Servers** dialog, select any servers you would like to register. Choose **Register** to register the selected servers and stay in the dialog. Choose **Register & Close** to register the selected servers and dismiss the dialog.



4. Choose **Rescan** to update the list with more unregistered servers.
5. When dealing with “cloned” environments where server characteristics are re-used, you can use the following options:
 - a. Specify the **Display Name Prefix** and/or **Display Name Suffix** to append information to the **Name** property of the Server. This will be shown under the **Display Name** column in the view. Changes will impact the selected servers only. This way you can do what-if scenarios before registering the new servers. Validation will apply where possible prior to processing the selected servers.
 - b. Indicate whether or not the servers to register will be treated as “cloned” servers. Selecting **Yes** will set a flag in the corresponding server profile. This can be changed later, if required. Selecting **No** will treat the server in a standard way. That will allow the server to use dynamic IP addresses. For cloned servers the IP address must be static.

- c. Select whether to use the **Server** or **Gateway IP Address** to uniquely identify the Server across multiple networks.

2.9 Configuring Release Paths

Release paths are used to define the full path for distributing the release for diverse scenarios. For example, the release path for a customer emergency might not be the same as the one for standard development. This also allows the definition of who are the acceptance, deployment, validation and approvers of each stage.

1. In the main menu, choose **Configure Paths**. Then, in the sub-menu, choose **Release Paths**. The list of active release paths will appear.
2. Choose the **New** button in the Release Paths list title bar. A form allowing you to define the new release path will appear.
3. Enter the release path name and description in the **Name** and **Description** fields respectively.
4. Go to the **Stages** tab. This tab shows the list of stages defining the release path.
5. Choose the **Add** button to create a new stage. A new stage will appear at the end of the list. It is possible to change the order of the stages using the **Move Left** and the **Move Right** buttons.
6. Select the stage type from the **Stage** dropdown. This will also load the Environments allowed for the selected stage type. If the logged user does not have access to the **Edit Approvals and Environment** permission or is not a release manager, he will not be able to modify the information of the stage.
7. Select the environment associated to the stage from the **Environment** dropdown.
8. Each stage is composed of 3 sequential steps (an **Acceptance** step, a **Deployment** step and an **Approval** step). Here is how to configure each of these steps:
 - I. **Acceptance** step: During the acceptance step, the user selected as the approver in the **Approver** dropdown will have the responsibility to approve or reject the deployment of the release in the stage. This step can be automated by selecting the **Automated** check box.
 - II. **Deployment** step: The deployment step is composed of 2 parts: the deployment of the components and the validation of the deployment which start after all components have been deployed successfully. During the deployment part, the user selected as the approver in the **Approver** dropdown will have the responsibility of the deployment and will receive email about execution and failure of the deployment. During the validation part, the user selected as the validator in the **Validator** dropdown will have the responsibility to perform basic tests to validate that the components have been deployed correctly.
 - III. **Approval** step: During the approval step, the different users added to the **Approvers** list will have the responsibility to approve or reject the release. Those approvals are not sequential; they are done in parallel.
9. Repeat step 5 to 8 to add more stages to the release path.
10. Go to the **Security** tab and select all groups that will have access to this release path in **View** or **Edit**. You can also decide to delegate management of the Release Path security to a group by using the **Manage Security** check box.

11. Choose the **Save & Close** button in the Release Path form title bar to persist your changes. The new release path will now appear in the list of active release paths.

2.10 Configuring Release Templates and Components

2.10.1 Create a new Release Template

A Release Template is the definition of the workflow used for releasing an application.

1. In the main menu, choose **Configure Apps**. Then, in the sub-menu, choose **Release Templates**. The list of active release templates will appear.
2. Choose the **New** button in the Release Templates list title bar. A form allowing you to define the new release template will appear.
3. Enter the **General Information** and configure **Security**. This information can be edited at a later time by selecting **Properties**.
4. As part of the **General Information** you will need to select a **Release Path** and a **Build Definition**.

Note: The Release Path dropdown will only contain the release paths that are active, allowed to the logged in user (depending on its group(s) security). The Build Definition selection will only appear if the system has been configured with at least one TFS Connection and is required only if one of the components to deploy uses the “Builds with application” as its source.

5. When all required information is entered accordingly, choose the **Create** button to generate a new Release Template. Be aware that this new Release Template will be inactive by default.
6. After the release template is created, it is always possible to access the **General Information** and **Security** by choosing the **Properties** hyperlink on the top left of the Release template.
7. Define the deployment activities to perform in the **Deployment Sequence** section by using the elements in the **Toolbox**. Be aware that any action or component can be executed only inside a server. To add or create new Components, see the next section.
8. Activate the Release template so it can be used to create releases.

2.10.2 Create a new Component from a Release Template

Components describe every part needed to deploy a complete application.

1. In the toolbox, choose the **Components** node and open its context menu. Choose **Add** from the context menu. This will present a list of components available for inclusion, and it also provides the ability to create new components.
2. Choose the **New** button on top of the list of components in order to associate a new component to the release template. A form within a popup window will appear allowing you to define the new component.
3. Enter the component name in the **Name** field.
4. Go to the **Source** tab.

5. Specify how the component gets the files used during the deployment by selecting one of the following 3 options:
 - i. **Builds with application:** When 'Builds with application' is selected, the component is integrated with Team Build and inherits the Team Project and Build Definition from the release template. At release time, those fields will be used to provide a selection of builds from which a specific package to release will be selected.
 - ii. **Builds independently:** When 'Builds independently' is selected, the Team Project and Build Definition need to be provided. At release time, those fields will be used to provide a selection of builds from which a specific package to release will be selected.
 - iii. **Builds externally:** When 'Build externally' is selected, a path needs to be supplied to get the base package location. At release time, a subfolder (corresponding to a package version) will need to be entered so that the full package location path can be derived.
6. Go to the **Deployment** tab. This tab allows you to configure how your component will be installed at the time of deployment on the target server.

What are Tools?

A **Tool** assists in the execution of various deployment scenarios that are often encountered. Tools come in the form of a batch file (.bat or .cmd), a PowerShell script (.ps1), an executable (.exe), etc.

7. Select a tool from the **Tool** dropdown. The Command and Argument fields will automatically be updated with values configured in the selected tool.

If you decide to use a custom tool, you will need to manually enter the installation command and its argument in the **Command** and **Argument** fields. You must also make sure that any files referenced by your command will be available in your package location at the time of deployment.

8. Go to the **Configuration Variables** tab. This tab shows the list of configuration variables used by the component.

9. Select the configuration variable replacement mode from the **Variable Replacement Mode** dropdown.

Variable Replacement Mode

The Variable Replacement Mode determines when the configuration variable tokens will be replaced with their values during a deployment. The following options are available: Never, Before Installation, After Installation and, Before and After Installation.

- *Only in Command*
The variables will only be replaced in the installation command if any
- *Before Installation*
The variables will be replaced in the installation command and before executing the installation command
- *After Installation*
The variables will be replaced in the installation command and after executing the installation command
- *Before and After Installation*
The variables will be replaced in the installation command, before and after executing the installation command

MSI deployment agent example

Using an MSI Deployer, the mode has to be *After Installation*. The reason is that only after executing the MSI that we have the config files extracted and available to the Deployer to replace.

XCopy deployment agent example

Using the XCopy deployment agent, the mode can be either *Before Installation* or *After Installation*. The reason is when the variables are replaced before execution of the xcopy command, the files being copied will already contain the right values instead of the token. When they are replaced after, the final files get scanned and modified with correct values.

Note that when using *After Installation*, only the files contained in the *Installation Path* folder (supplied by a mandatory configuration variable) will be replaced.

File Extension Filter

When the selection is *Before Installation*, *After Installation* or *Before and After installation*, the File Extension Filter needs to be provided. This tells the deployment agent which files needs to be scanned when replacing the variables. It uses a matching pattern to find the files. If multiple patterns are needed, separate them with a semi-colon (;).

Example of Pattern	Deployment Agent Action
.	will scan all files
*.config	will scan all files ending with .config
MyApp.MyModule.exe.config	will scan only that specific file
.config;.xml	will scan all files ending either with .config or .xml

10. Choose the **Add** button on top of the list of configuration variables in order to create a new configuration variable for the component. Enter the configuration variable name and description in the **Name** and **Description** column respectively. Repeat as necessary.
11. Choose the **Save and Close** button in the Component form title bar. The new component will now appear in the list of components.
12. Repeat the steps in this section to add more components to the release template.

2.10.3 Configure Component in a Release Template

1. If at least one component associated to the release template is of type **Builds with application**, you can override its default package location. This is done by accessing the component context menu in the toolbox and selecting **Override Package Location**.
2. Go to the **Configuration Variables** panel at the bottom of the view. This panel shows a table to set the value of the configuration variables for all stages of the selected release path.
3. Fill the configuration variables table for all components.
4. Choose the **Save & Close** button in the Release Template form title bar to persist your changes. The new release template appears in the list of release templates.

2.10.4 Configure Security in a Release Template

1. Go to the **Security** section in the **Properties** editor. This section shows the list of security groups used by the release template.
2. Choose the **Link Existing** button on the top of the list of security.
3. A form within a popup window will open allowing you to select the active group. Select item(s) and choose the **Add** button on the top of the list.
4. In the **Security** tab select the **Can Release** permission to allow all members of this group the release permission for this release template. It is also possible to define which group can **View** or **Edit** the release template or decide to delegate management of the Release Template security to a group by using the **Manage Security** check box.
5. Choose the **Close** button in the **Properties** editor, followed by a **Save** or **Save & Close** to persist your changes.

2.11 Configure Releases

2.11.1 Open a Release Template

We are assuming in this section that a release template has already been configured.

1. In the main menu, choose **Configure Apps**. Then, in the sub-menu, choose **Release Templates**. The list of active release templates will be displayed.
2. Double-click the release template for which you need to create a new release. The release template will open.

Note: If you cannot find the release template in the list, it may be because its status is inactive. This is the default status when creating a new release template. Change the status filter in the Release Templates list title bar from **Active** to **Inactive**.

2.11.2 Create Release from Release Template

Releases are instances constructed from the definition created during the configuration of a release template.

3. Choose the **New Release** button in the Release Template form title bar. A form will appear to configure the new release.

Note: If the New Release button is disabled, it may be because the release template status is set to Inactive. In that case, activate the release template by choosing the **Activate** button in the release template form title bar. Another reason could be that you are not part of a group that has the rights to create releases for this release template.

4. Enter the release name in the **Name** field.
5. If at least one component is of type **Builds with application**, select the build that will be used for all components of this type.

Note: There are three ways to select a build.

1. **By choosing the Latest link:** This will fetch the **latest build** for the selected build definition that is in the status **Succeeded** or **Partially Succeeded**.
2. **By manually entering a build:** Simply manually enter the build in the textbox. It will get validated on **Save** or **Start**.
3. **By choosing the Select... link:** This will open a popup with the today's build preloaded. Other filters are available to select a build from a different date.

6. If components need to be configured at this time, you will see a section in the form where you can complete the component configuration. The **Components** section helps you determine the locations of the files supporting the deployment of the different components. This is referred to as the Package Location.
7. Select **Start** to immediately start the release. Or select **Create** to persist the changes and allow you to do further editing of other data such as configuration variable values. Selecting **Cancel** will conclude the creation process, discarding any configuration changes you may have made.
8. Go to the **Configuration Variables** panel. The values seen here come from those existing in the release template at the moment the release was created. They can be modified for this release. However, the changes will only apply to the current release (they won't be reflected back to the release template or other releases).

Note: In a release template, any modification to components and/or their configuration variable values will not impact already existing releases.

2.12 Deploy a Release

2.12.1 Open a Draft Release

We are assuming in this section that a release has already been configured.

1. In the main menu, choose **Track Releases**. Then, in the sub-menu, choose **Releases**. The list of **In Progress** releases will appear.
2. Change the status filter in the Releases list title bar from **In Progress** to **Draft**.
3. Double-click the release you want to start.

2.12.2 Start a Release

We are assuming in this section that the release status is **Draft**.

4. Choose the **Start** button in the Release form title bar. The status of the release will change to **In Progress**.

Note: If the release status changes to **Rejected** after you choose the Start button, it may be because the **Accept Deployment** step is automated and the release automatically jumped to the **Deploy** step which failed. Check the **View Log** section to review what happened.

5. Context will be immediately placed in the **View Log** section, where you will be presented with a log of all events associated to the release. If any actions are required of you, a **My Approval Requests** section will appear and the context will be automatically switched to it. At this point, you should be able to see an **Accept Deployment** step in **Pending** status. The release will stay in this state until the owner of the **Accept Deployment** step accepts or rejects the deployment.

2.12.3 Accept Deployment in Stage

We are assuming in this section that the **Accept Deployment** step status is **Pending**.


6. If you are the owner (Approver) of the **Accept Deployment** step, go to the **Action Requests** tab. The action requests list will contain a **Validate Deployment** action request.

Note: The action requests list shows a list of all the Approve/Reject actions you need to perform for the current release.

7. Select the **Accept Deployment** action from the list and choose the **Approve** button on top of the action request list.
8. Go back to the **Log** tab. The status of the Accept Deployment step will now be set to **Done** and a new **Deploy** step with status **Pending** is now visible.

2.12.4 Retry a Failed Deployment

We are assuming in this section that the release status is **Rejected** because the deployment step failed.

9. Go to the **Log** tab. The first entry in the log will be the last deployment attempt.
10. Choose the ellipsis button () in the **Details** column of the rejected **Deploy** step. The Deployment Log popup window will appear. The deployment log will help you determine why the deployment failed.
11. Make the necessary adjustments to the release configuration and/or its component.

Note: When debugging to find the reason why a deployment failed, it can be helpful to execute the installation command (which can be found in the Log) manually in a command prompt on the targeted server.

12. Choose the **Retry Failed Deployment** button in the Release form title bar.
13. Go back to the **Log** tab and verify that the deployment was successful.

2.12.5 Validate Deployment

We are assuming that the release has been deployed successfully into the current stage environment.

14. If you are the owner (Approver) of the **Validate Deployment** step, go to the **Action Requests** tab. The action requests table will contain a **Validate Deployment** action request.
15. Once you have completed the deployment of the component, select the **Validate Deployment** action from the list and choose the **Approve** button on top of the action request list.

2.12.6 Approve Release

We are assuming in this section that the **Approve Release** step status is **Pending** and that the release path for the current stage have approvers defined.

16. If you are the owner (Approver) of the Approve Release step, go to the **Action Requests** tab. The action requests list should contain an **Approve Release** action request.
17. Select the **Approve Release** action from the list and choose the **Approve** button on top of the action request list.

Note: The release will not move to the next stage until **all** approvers have approved the release.

2.13 Trigger a Release from Team Build

The current support for triggering releases from Team Build is done using a custom Build Process Template.

2.13.1 Pre-requisites

To be able to start a Release from a build, a release management Client needs to be installed on the machine(s) running the TFS build agent. Configure the Client to point to the release management Server. In order to trigger a release from a build, the release template must first be configured correctly. From the release template detail view select “**Can Trigger a Release from a Build?**” found under the **Properties** editor.

By selecting this flag we are telling release management to accept requests from a Team Foundation Build to start a release. If this flag is not set, the request coming from a Team Foundation Build will simply be ignored and the build will fail.

Restrictions

Only release templates for which **all** its components have been configured to “**Builds with application**” or “**Builds externally**” will be deployed in this way. In other words, if a release template is using a component that “**Builds Independently**” it cannot be deployed automatically from a build.

The **Acceptance Step** and the **Deployment Step** of the first stage of the associated release path must be set to **Automated**.

2.13.2 Visual Studio 2010, Visual Studio 2012 and Visual Studio 2013

When using Visual Studio 2010, Visual Studio 2012, or Visual Studio 2013, release management provides its own template (select the proper version of the Build template corresponding to your TFS Version). This build template contains the logic needed to trigger a Release from a build.

This template also contains the logic to tokenize your configuration files. This logic assumes that in your solution, you have two versions of your configuration files. One version is your normal configuration file used during local development, and the other is a corresponding file that has the same content, except that instead of having local values for your variables, tokens have been put there. The build activity will swap those two files before doing the build, so that we end up with the tokenized version of the configuration files in the drop location.

Here is an example of how to achieve this: For example, if your solution contains a file called web.config, then you will need to copy that file (and keep them in sync), and name it web.config.token. Your web.config file will stay the way it is now (and that is what is going to be used when you run the app locally). The web.config.token will contain tokens instead of values.

So, as an example, let's say you have in your web.config this line:

```
<add key="SMTPHostServer" value="mysmtp.myserver.domain"/>
```

Then, you would have this line in your web.config.token:

```
<add key="SMTPHostServer" value="__SMTPHostServer__"/>
```

2.14 Include Release Management Custom Build Logic into a Build Template

To be able to provide some of its functionality, release management provides a TFS Build Template with additional logic. If you currently use the DefaultTemplate provided by TFS, you can simply replace it with the ReleaseDefaultTemplate and you will be up and running. However, if you have modified the DefaultTemplate to add your own logic, you will need to merge the build logic to yours.

Note that this is only needed if you want to take advantage of one of the following features:

- Automatically trigger releases from a build
- Use the file swapping technique to get tokenized file in the drop location

Here is a step by step procedure to get the logic into your Build Template.

2.14.1 Add arguments to custom Build Template

Open ReleaseDefaultTemplate.xaml (TFS 2010) - or ReleaseDefaultTemplate.11.1.xaml (TFS 2012 & TFS 2013). Those files can be found on a computer where release management components have been installed at this default location: *C:\Program Files (x86)\Microsoft Visual Studio 12.0\ReleaseManagement\bin*. Find the last 3 arguments and add them to your customized Build Template:

Name	Direction	Argument type	Default value
ConfigurationsToRelease	In	PlatformConfigurationList	Enter a VB expression
ReleaseBuild	In	Boolean	False
ReleaseTargetStage	In	String	Enter a VB expression
Create Argument			

*** If you are using the DefaultTemplate.xaml from TFS 2012, you need to add a "DropBuild" argument. In TFS 2010, the "DropBuild" argument already exists in the arguments list.

Name	Direction	Argument type	Default value
DropBuild	In	Boolean	True
Create Argument			
Variables Arguments Imports			

2.14.2 Updating Metadata Argument

Open the argument Metadata.

Name	Direction	Argument type	Default value
DisableTests	In	Boolean	False
GetVersion	In	String	Enter a VB expression
PrivateDropLocation	In	String	Enter a VB expression
Verbosity	In	BuildVerbosity	Microsoft.TeamFoundation.Build.Workflow.BuildVerbosity.Normal
Metadata	Property	ProcessParameterMetadataCollection	(Collection)
SupportedReasons	Property	BuildReason	All
BuildProcessVersion	Property	String	11.0
ConfigurationsToRelease	In	PlatformConfiguration	Enter a VB expression
ReleaseBuild	In	Boolean	Enter a VB expression
ReleaseTargetStage	In	String	Enter a VB expression

Add the 3 entries found in Metadata collection of ReleaseDefaultTemplate.xaml file.

2.1 - Configurations to Release

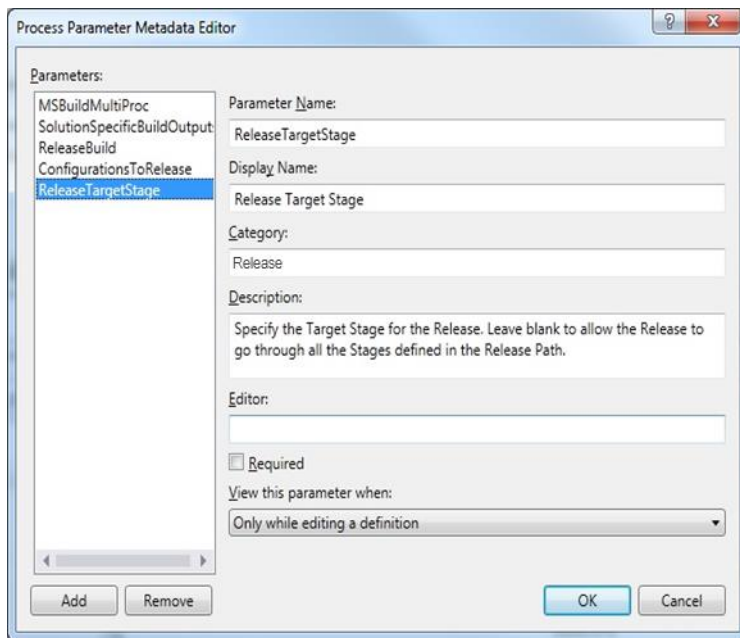
Leave the ConfigurationsToRelease property blank.

2.2 - Release Build

Copy and paste this description: **Set this to True to indicate you want this build to be released.**

2.3 - Release Target Stage

Copy and paste this description: **Specify the Target Stage for the Release. Leave blank to allow the Release to go through all the Stages defined in the Release Path.**

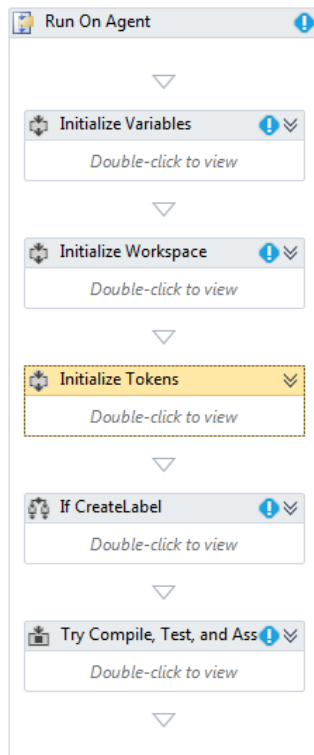


3 - Initialize Tokens logic

From the **ReleaseDefaultTemplate.xaml**, navigate to

Process > Sequence > Run On Agent

and copy the **Initialize Tokens** sequence. Paste it in your Build Template after the sequence **Initialize Workspace** and before the If condition **If CreateLabel**.

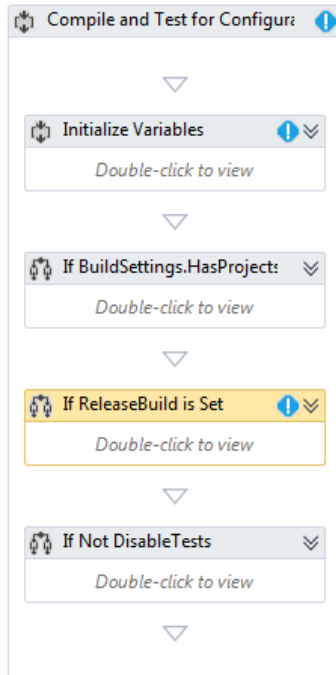


4 - Release Build logic

From the **ReleaseDefaultTemplate.xaml**, navigate to

Process > Sequence > Run On Agent > Try Compile, Test... > Sequence > Compile, Test, an... >
> Try Compile and Test > Compile and Test > For Each Configur... > Compile and Test ...

and copy the **If ReleaseBuild is Set** sequence. Paste it in your Build Template after the sequence **If BuildSettings.HasProjects** and before the sequence **If Not Disable Tests**.



5 - Save

Save your Build Template.

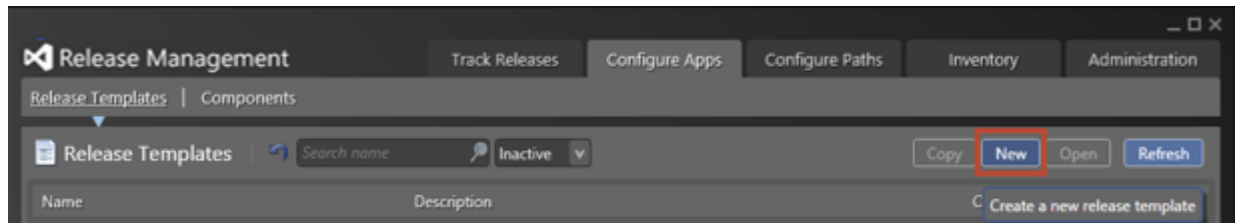
Note: When using the functionality to automatically trigger Releases from a Build, you will need to install Release Management Client for Visual Studio 2013 on your TFS Build Server. Also, after installation, restart the TFS Build Services.

3 RELEASE TEMPLATES

3.1 Overview

The Release Template allows for the definition of the sequence used to orchestrate the deployment of your application components. It also allows for the selection of all sorts of actions to be executed during a deployment process. The sequence can be configured independently for every stage defined in the associated release path.

A Release Template can be created from the **Release Templates** view by selecting the **New** option.



3.2 Properties

The Release Template **Properties** dialog appears allowing you to specify details related to your application. You can create as many Release Templates as needed.

The 'Properties' dialog box is divided into two main sections: 'General Information' and 'Security'.
General Information:
- Name: 'New Release Template 1'
- Description: (empty text field)
- Release Path: '< Select >' (dropdown menu)
- Build Definition: '<Click Edit to Select>' (button) with an 'Edit' link.
- A checkbox labeled 'Can Trigger a Release from a Build?' is present.
Security:
- Buttons: 'Open', 'Create', 'Link Existing', 'Remove'.
- Warning: 'Changes to security will only take effect when users restart their application.'
- Table:

Name	Description	View	Edit	Can Release
Everyone		✓	✓	✓

- Bottom buttons: 'Create', 'Cancel'.

Because a Deployment Sequence is defined per Stage, a Release Path must be selected. The Release Path is where the Stages are defined.

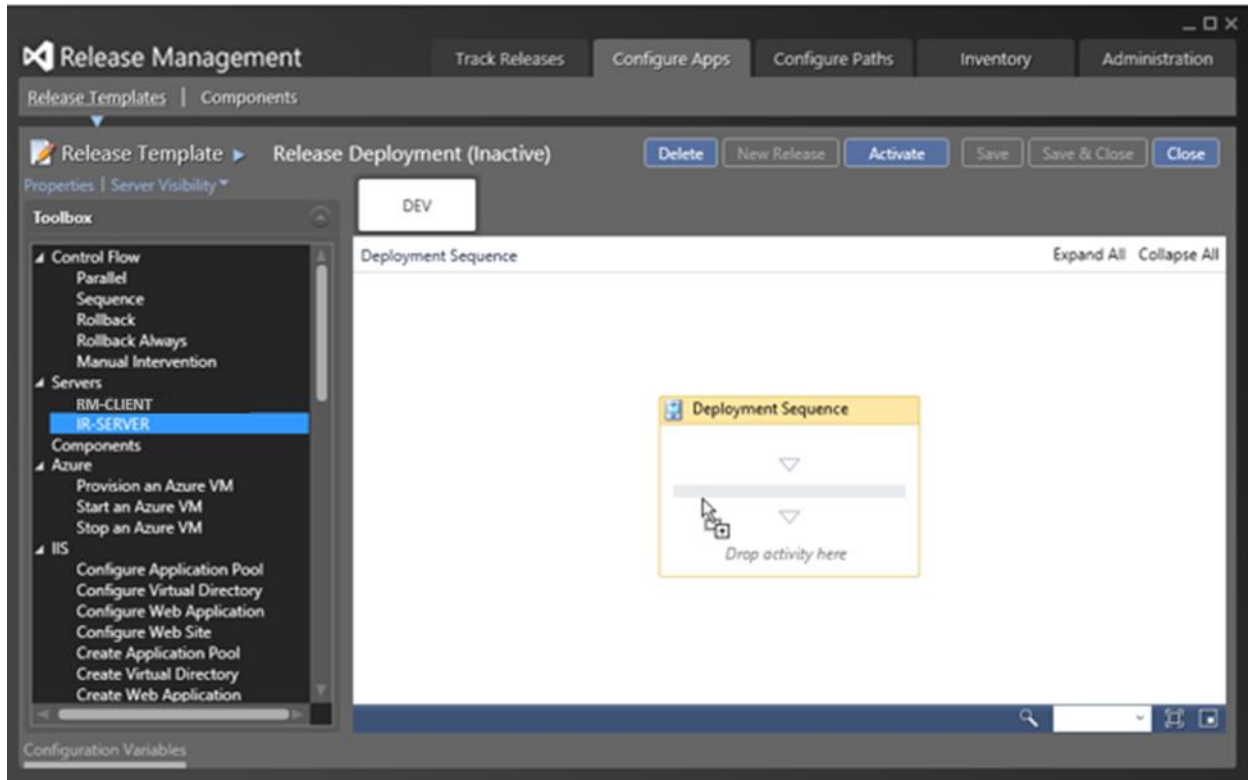
If your Release Management is configured with at least one Team Foundation Server connection, the **Build Definition** field will appear. When visible, this field becomes mandatory.

This is also where you would specify if the release template can be used to release from a Team Foundation Build by setting the flag **Can Trigger a Release from a Build?**

In addition, this is the place where **Security** can be configured to limit access to the release template.

3.3 Deployment Sequence

Following the creation of a Release Template, you are placed in the Deployment Sequence view. The Toolbox contains a list of all activities that can be added to the sequence. This can be done via a simple drag-and-drop operation.



The Toolbox is organized into categories. There are static / system categories. The System Categories are **Control Flow**, **Servers**, and **Components**. The remaining Categories are extracted from the Actions. When you edit an Action, you can associate it to a Category.

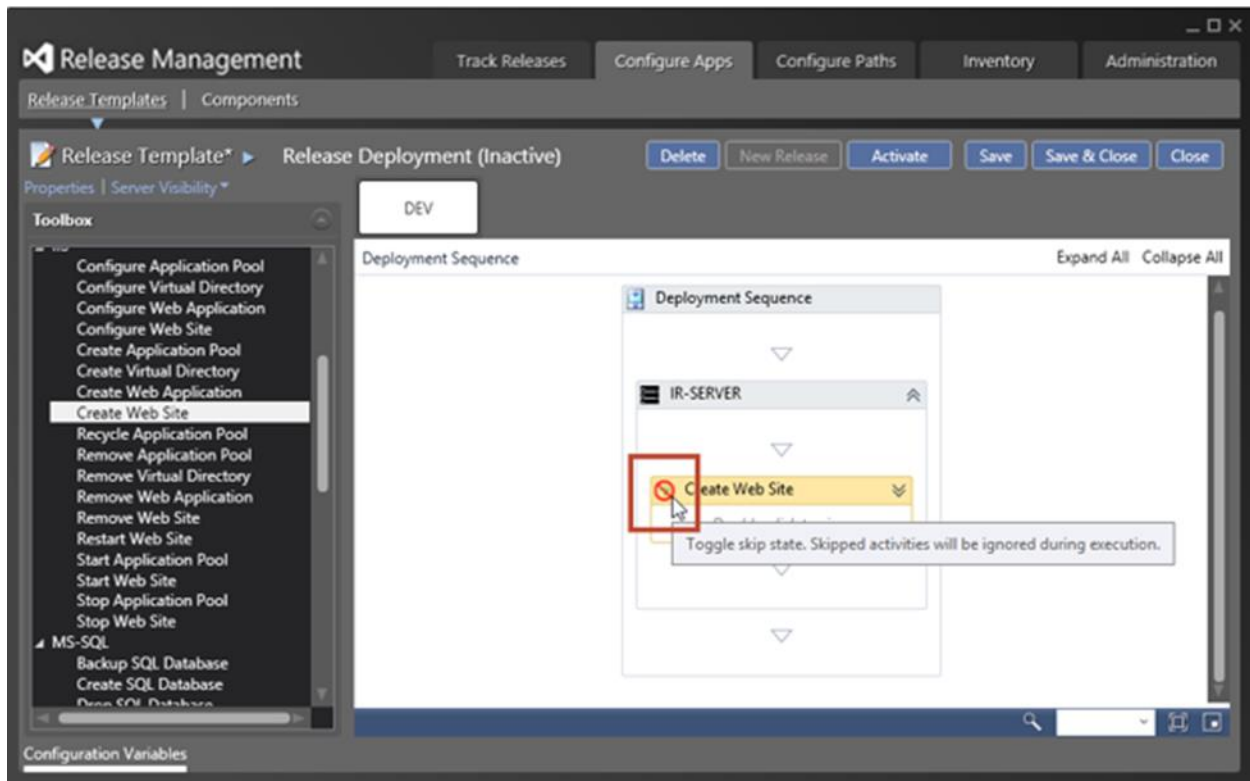
Release management comes pre-packaged with a large number of ready-to-use Actions. All of these are documented in the Appendix sections at the end of this user guide.

The Deployment Sequence view has some basic rules to remember:

- Each Stage has its own Deployment Sequence.
- Actions and Components can only be added to Servers.
- Servers can be organized in Sequence or Parallel activities.
- Titles of Parallel, Sequence, and Manual Intervention activities can be edited; all others are read-only.
- Server activities must have at least one child activity such as a Component or an Action.
- Rollback activities must have be preceded by at least one Component or Action.
- Rollback and Rollback Always activities must have at least one child activity such as a Component or an Action.

If any of the preceding rules is not met, it will not be possible to activate the Release Template and therefore Releases cannot be created using this Release Template.

In addition, only Component or Action activities can be temporarily disabled or skipped from execution. This can be done by choosing the activity's icon image.

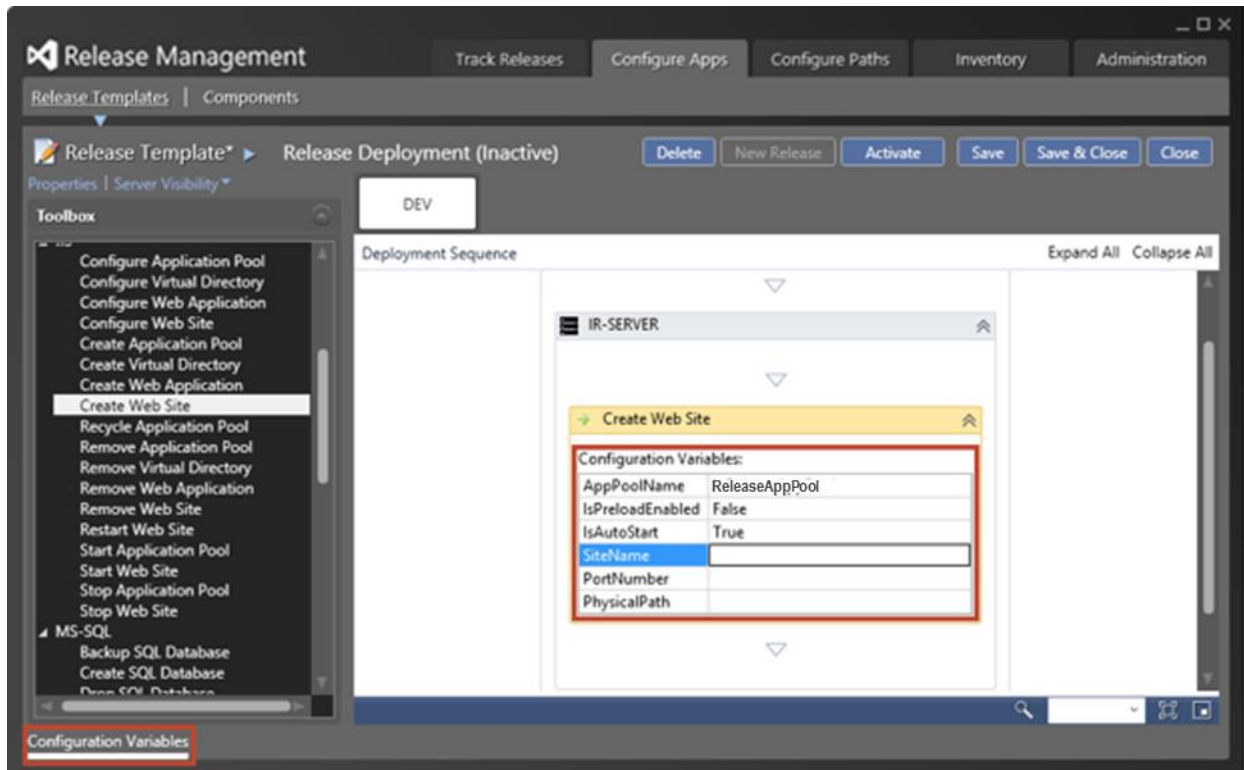


Skipping a Component or Action can also be done when defining a Release. Therefore, the Release Template can be the default, while the Release can be the override for such a feature.

3.4 Configuration Variables

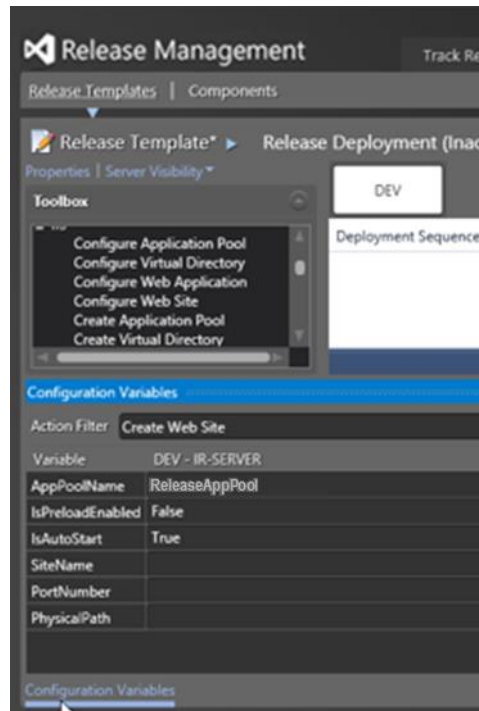
Components and Actions can be dynamically configured by providing values specific to a Stage. This is done via Configuration Variables, which are defined during the configuration of a Component / Action.

Specifying the values of a Configuration Variable can be done in 2 ways: Inline, within a Component / Action activity inside the Deployment Sequence; or globally, within the Configuration Variables panel at the bottom, left of the view.



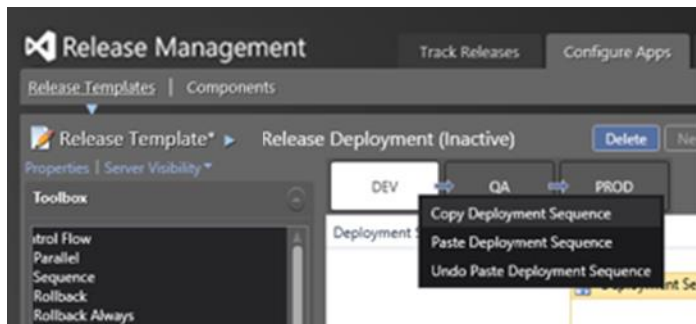
These two input mechanisms are synchronized. So, specifying a value using one method would automatically impact the other representation of the same data.

The Configuration Variables panel is useful when you want to compare values across multiple Stages. You can also use it to manage the values (i.e. copy / paste).

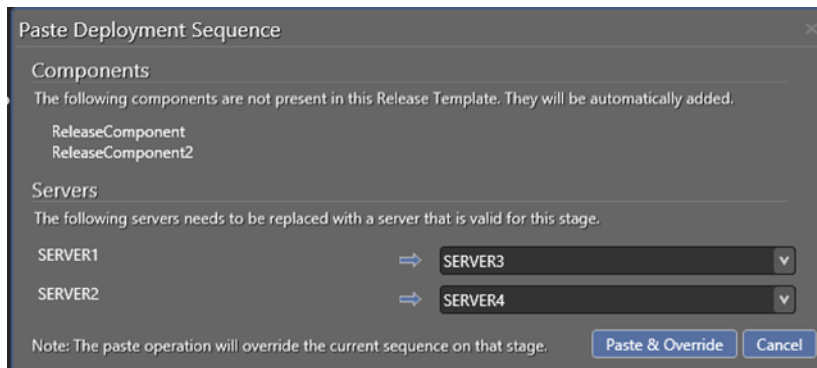


3.5 Copying a stage to another

It is possible to copy the entire deployment sequence of a stage onto another one. This is done by using the contextual menu of the stage. To access it, right-click the stage to copy and then right-click again on the destination stage to paste it. It is also possible to undo the last paste to recover the previous deployment sequence of the destination stage:



Since the allowed servers are usually different from stage to stage, when pasting, a new window will be shown to identify the equivalence in the servers. Also, if copying from one release template to another, it could be possible that the destination release template is missing some of the component being copied. If it is the case, the new window will show the list of component that will automatically be added to the destination release template. Be aware that components added in this way will not be removed if the paste is undone.

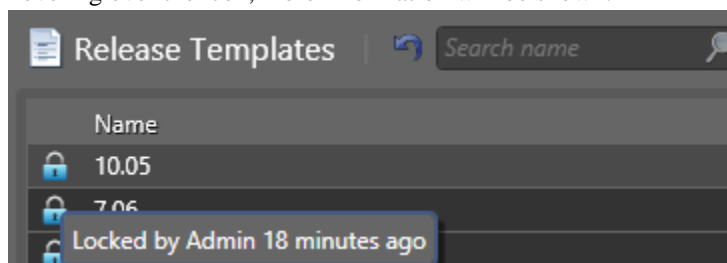


3.6 Locking Mechanism

To prevent having two users editing a release template at the same time and hence overwriting each other's work, a locking mechanism has been put in place. As soon as a user opens a release template, the template is locked. Other users will not be able to edit this release template until the first user leaves that release template (with or without saving).

Users will be informed that a release template is locked to another user in two ways.

1. The list of release template shows a small icon to the left of the name to indicate that it is locked. When hovering over the icon, more information will be shown.



2. When a user opens a release template, he will be informed that it is locked by someone else. The user can still open the release template, but it will be put in read-only mode.

Sometimes, different users will want to work on the same release template. One way to achieve it would be to make a copy of the release template, edit that copy and either save it for future edits or copy/paste the relevant information from a deployment sequence to the original release template. The copied release template can be deleted once it is not needed anymore.

The release managers have the option of removing any locks that a user may have. This can be useful if a user has left RM running with a release template opened.

3.7 Rollback and Rollback Always Mechanism

A deployment sequence has two modes of execution: Normal and Rollback. During normal execution, all activities that are not in a **Rollback** or **Rollback Always** container are executed. During rollback execution, only the activities that are within a **Rollback** or **Rollback Always** will be executed.

Rollback and **Rollback Always** are containers in which users can add any actions/components/manual interventions as needed. The user is responsible to build its rollback mechanism (although preexisting actions can be used).

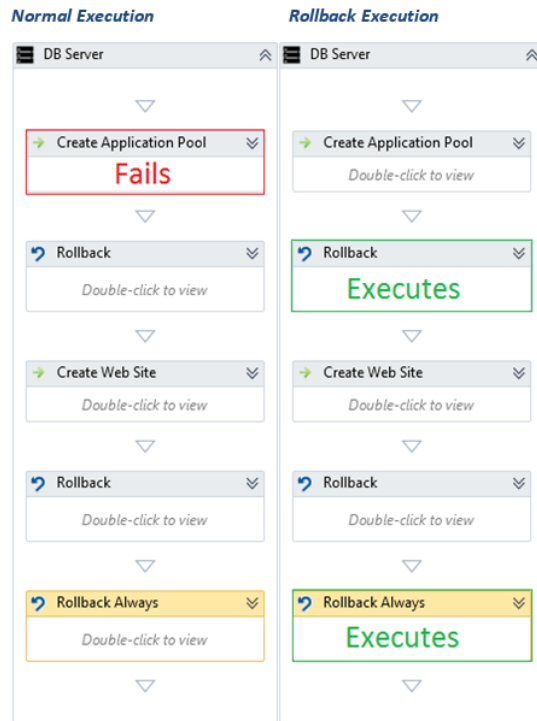
When an error happens during the Normal execution of a deployment sequence, the sequence terminates (parallel activities finish their execution). If there is at least one rollback activity to be executed, the deployment sequence is restarted (from the top) in mode Rollback. In that mode, only the activities that have been identified (automatically during the Normal execution) will be executed.

A rollback activity of type **Rollback** will be identified to be executed when at least one normal activity position before that rollback activity have been executed during the normal execution phase.

A rollback activity of type **Rollback Always** will be identified to be executed if any normal activity fails during the normal execution phase.

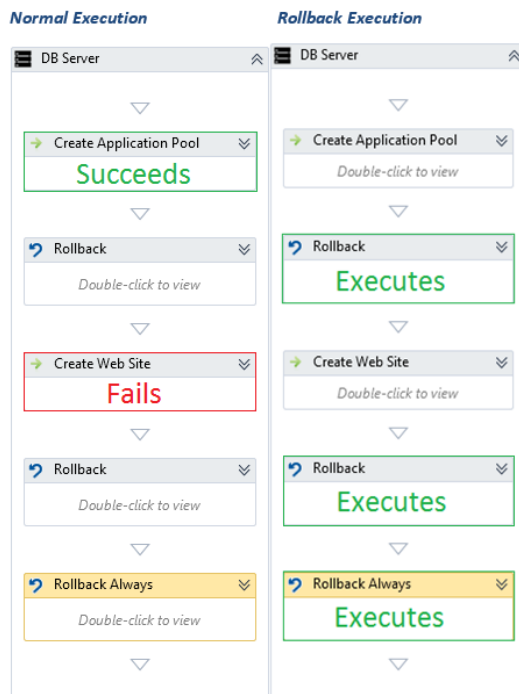
3.7.1 Example 1

During normal execution, if 'Create Application Pool' fails, activities contained in first Rollback and activities contained in Rollback Always are marked as to be executed during the rollback mode execution.



3.7.2 Example 2

During normal execution, if 'Create Application Pool' succeeds, then 'Create Web Site' fails, activities contained in first Rollback, second Rollback and Rollback Always are marked as to be executed during the rollback mode execution.

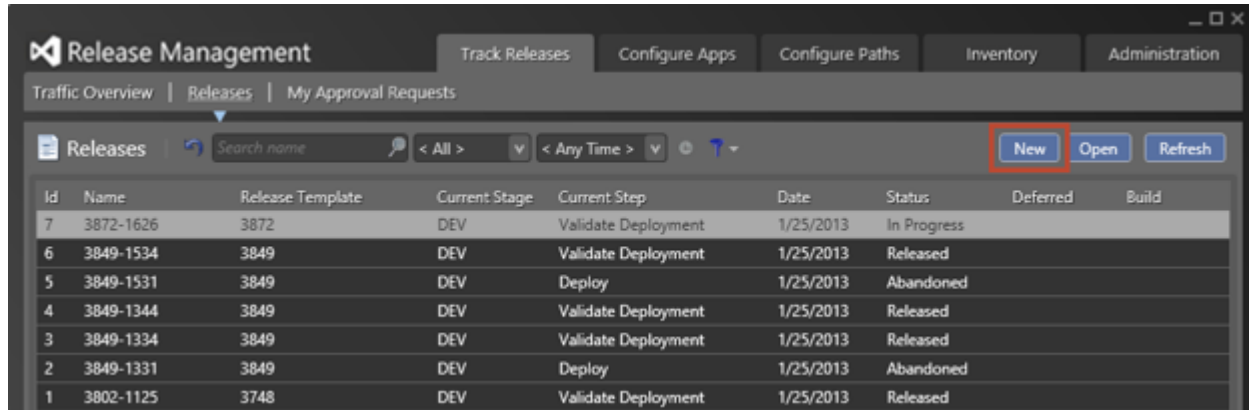


4 RELEASES

4.1 Overview

Releases represent the physical act of deploying your application through a Release Path until the Target Stage is reached. Throughout this process, release management performs a number of checks to ensure that components are only installed on a Server when the owner of the corresponding Environment allows it. A Release is based on a Release Template.

A Release can be created from the **Releases** view under **Track Releases** by selecting the **New** option. You can also create a Release directly from inside a **Release Template** by selecting the **New Release** option.

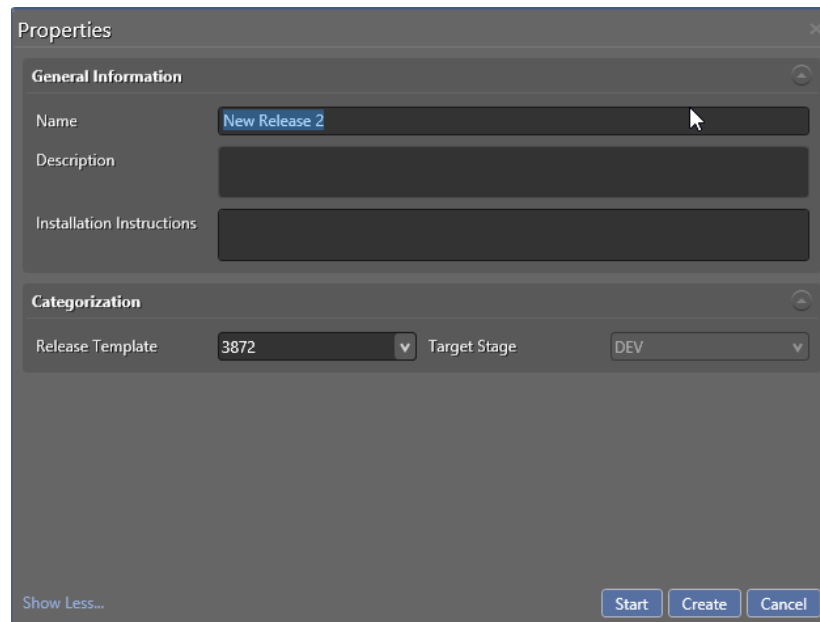


The screenshot shows the 'Release Management' application interface. The 'Track Releases' tab is active. Below the navigation bar, there's a 'Releases' section with a search bar and filters. The 'New' button is highlighted with a red box. Below this is a table of releases.

Id	Name	Release Template	Current Stage	Current Step	Date	Status	Deferred	Build
7	3872-1626	3872	DEV	Validate Deployment	1/25/2013	In Progress		
6	3849-1534	3849	DEV	Validate Deployment	1/25/2013	Released		
5	3849-1531	3849	DEV	Deploy	1/25/2013	Abandoned		
4	3849-1344	3849	DEV	Validate Deployment	1/25/2013	Released		
3	3849-1334	3849	DEV	Validate Deployment	1/25/2013	Released		
2	3849-1331	3849	DEV	Deploy	1/25/2013	Abandoned		
1	3802-1125	3748	DEV	Validate Deployment	1/25/2013	Released		

4.2 Properties

When a new Release is created you are presented with the Release **Properties** dialog. This is where you can specify the name of your Release, a Description, and some Installation Instructions.



The screenshot shows the 'Properties' dialog box for a new release. It has two main sections: 'General Information' and 'Categorization'. In 'General Information', the 'Name' field is filled with 'New Release 2'. In 'Categorization', the 'Release Template' is set to '3872' and the 'Target Stage' is set to 'DEV'. There are 'Start', 'Create', and 'Cancel' buttons at the bottom.

General Information	
Name	New Release 2
Description	
Installation Instructions	

Categorization	
Release Template	3872
Target Stage	DEV

If the Release is created from the Release Template, the corresponding property is set. If the Release is created directly from the Releases list, then you must first select the Release Template based on which the Release will execute.

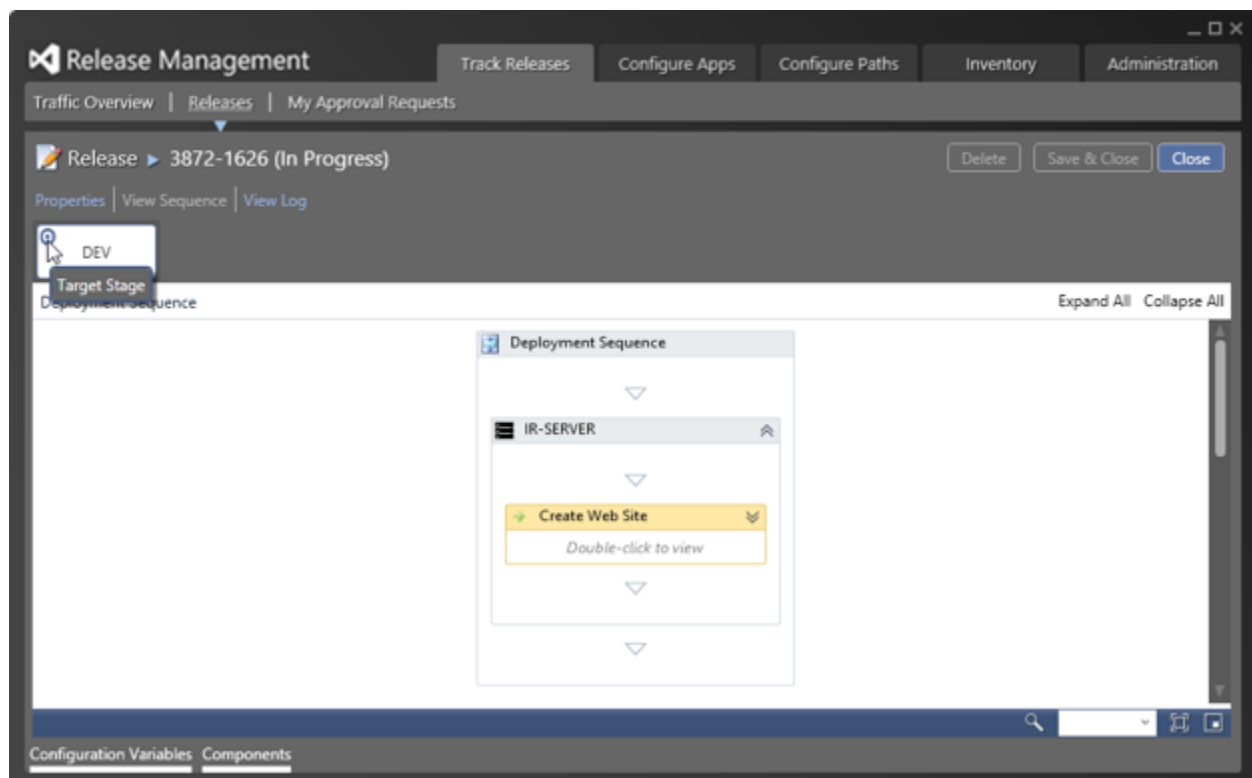
When selecting a Release Template that is associated with a Release Path that contains more than one Stage, you can then also select the Target Stage. In cases where the Release Path contains multiple Stages, the Target Stage allows you to override the default behavior of publishing a Release through all Stages. You may, for example, only want to publish your Release up to your User Acceptance Test (UAT) Stage but not all the way to Production.

A Release that has been successfully published all the way to its Target Stage, can later be re-configured to continue the process to a Stage later in the Release Path.

4.3 Deployment Sequence

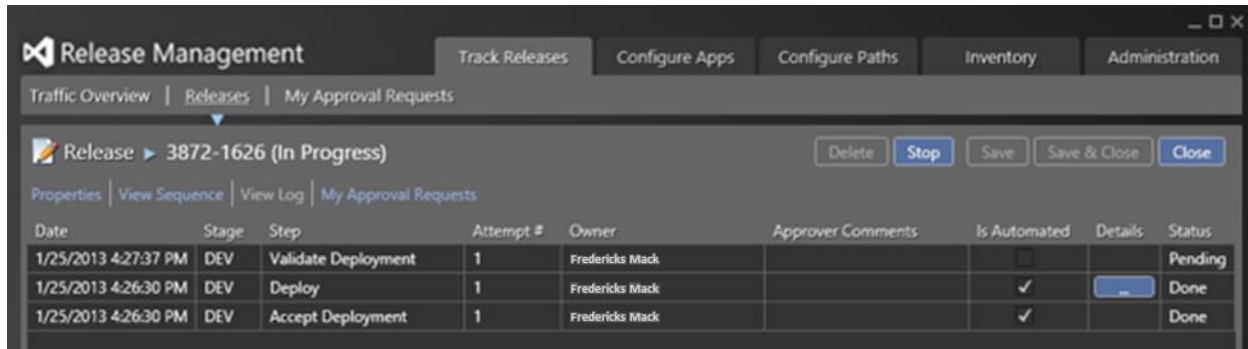
A Release Deployment Sequence can be viewed by selecting the **View Sequence** option. The layout of the Deployment Sequence is the same as in the Release Template. However, the actual sequence is not editable. Configuration Variables, skipping activities, and some of the activity titles are editable. Editing of a Release is controlled by the Release's current State.

The Target Stage is depicted by a “bulls-eye” icon over the corresponding Stage. You can change it again via the Properties dialog.



4.4 Deployment Log

The Deployment Log can be accessed by selecting the **View Log** option. In this view you get all the already completed or currently processing steps. Setting the **Include Future Steps** flag causes the view to show steps that have yet to be executed.



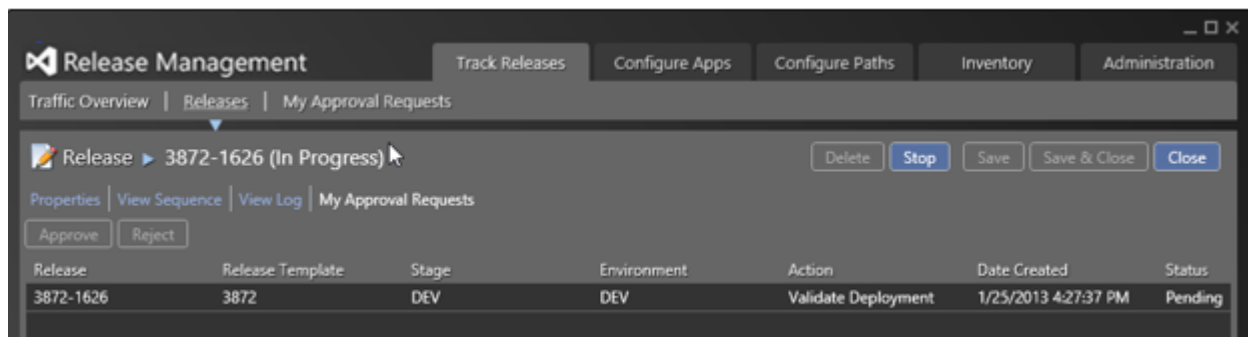
The screenshot shows the 'Release Management' application window. The 'Track Releases' tab is active. Under the 'Releases' sub-tab, the release '3872-1626 (In Progress)' is selected. The 'View Log' option is chosen, displaying a table of deployment steps.

Date	Stage	Step	Attempt #	Owner	Approver Comments	Is Automated	Details	Status
1/25/2013 4:27:37 PM	DEV	Validate Deployment	1	Fredericks Mack		<input type="checkbox"/>		Pending
1/25/2013 4:26:30 PM	DEV	Deploy	1	Fredericks Mack		<input checked="" type="checkbox"/>	Details	Done
1/25/2013 4:26:30 PM	DEV	Accept Deployment	1	Fredericks Mack		<input checked="" type="checkbox"/>		Done

Selecting **Details** (button under the Details column) displays the next level information for the corresponding Deployment Log Entry. Not all entries have more details to show.

4.5 Pending Approval Requests

Whenever you, as a release management user, are required to act on an Approval Request, the **My Approval Request** option will appear. In this view you will see all pending items on which you need to act. Selecting the entry in the list, followed by **Approve** or **Reject** will display the required dialog to allow you to provide details for your action.



The screenshot shows the 'Release Management' application window. The 'Track Releases' tab is active. Under the 'My Approval Requests' sub-tab, the release '3872-1626 (In Progress)' is selected. The 'Approve' and 'Reject' buttons are visible. A table lists the pending approval requests.

Release	Release Template	Stage	Environment	Action	Date Created	Status
3872-1626	3872	DEV	DEV	Validate Deployment	1/25/2013 4:27:37 PM	Pending

5 RELEASE EXPLORER

5.1 Overview

For approvers, a web client called **Release Explorer** is available. To access your **Release Explorer**, request the URL from your application administrator. On a default installation, the URL would be in the form **http[s]://<Release ManagementServer>[:port]/ReleaseManagement**.

In the **Release Explorer**, you have the option to

- **Approve** and/or **Reject** a Release,
- **Reassign** a Release,
- See the **Components** used in a Release,
- See the **Current Stage** and **Step** of a Release, and
- See the list of **Approved/Rejected** Releases.


Releases Explorer

To approve



Releases

Approve Reject Reassign

- My releases
To approve
Previously approved

2265  waiting for acceptance
6 days ago

Build Triggered: 06/05/2013 4:03:21 ... Lead Develop...
InRelease 3.x

5.2 Operations

5.2.1 Viewing Component Details

To view the Component Details for a Release,

1. Choose the **Component** button. This will open the Component view.
2. In the Component view you will see a list of all the **components** used in your current release with their build associations.

Lead Develop...
7 comp.

7 Components

Server

Release_Main_CI_20130506.1_3.3.0.3561

Post Action: Add...

Release_Main_CI_20130506.1_3.3.0.3561

Post Action: Chan...

Release_Main_CI_20130506.1_3.3.0.3561

Post Action: Chan...

Release_Main_CI_20130506.1_3.3.0.3561

Client

Release_Main_CI_20130506.1_3.3.0.3561

Deployment Agent

Release_Main_CI_20130506.1_3.3.0.3561

Backup database

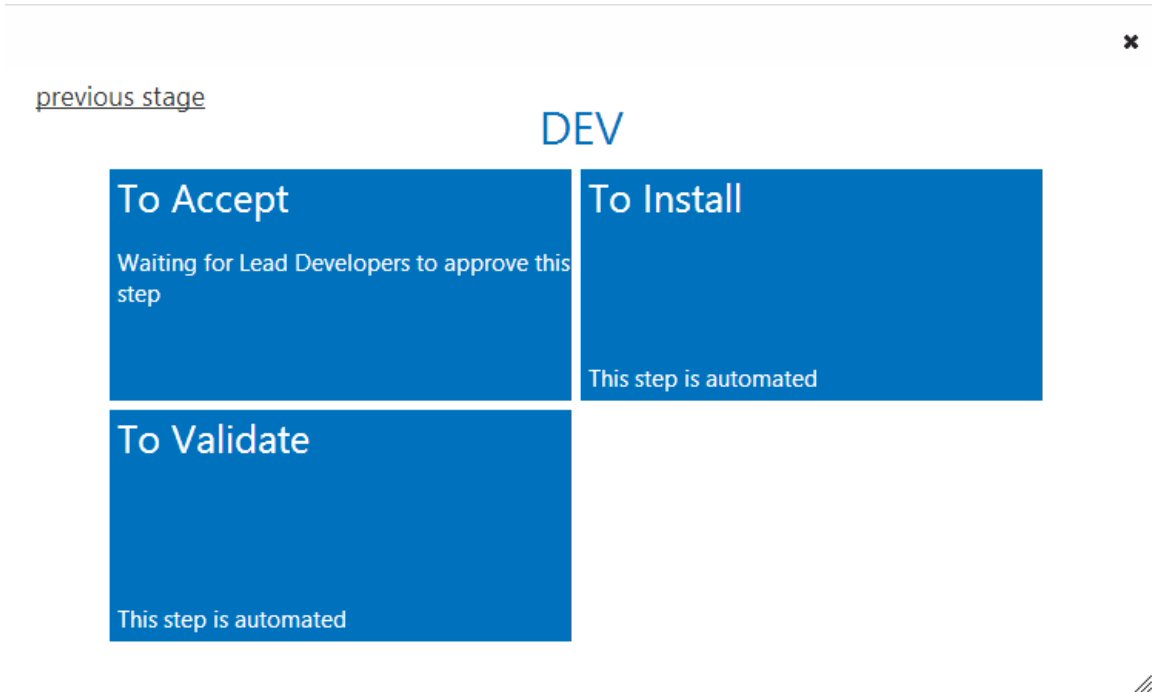
Release_Main_CI_20130506.1_3.3.0.3561

5.2.2 Release Stage

To retrieve the details of a Release Stage:



1. Choose the blue square, which indicates the active stage. This will open the Current Stage view.
2. The Current Stage view shows all the steps that have been or will be executed.



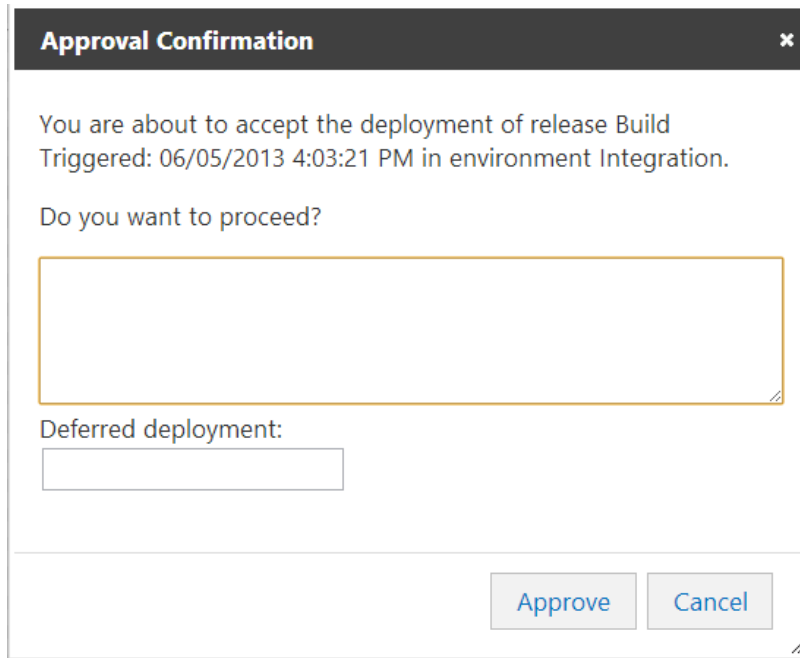
3. Selecting **previous stage** will navigate to the previous stage. Similarly, selecting **next stage** will navigate to the next available stage in the Release Path.

5.2.3 Approving a Release

To approve a Release:

1. Select the Release you want to approve by choosing it.
2. Click the **Approve** button. The **Approval Confirmation** dialog appears.
3. On the **Approval Confirmation** dialog, observe the following:
 - a. The **Do you want to proceed?** text box is used to put a comment for the action.
 - b. The **Defer deployment** field allows you to schedule a deployment to be triggered at a later point in time.

4. Approve the Release by choosing the **Approve** button.

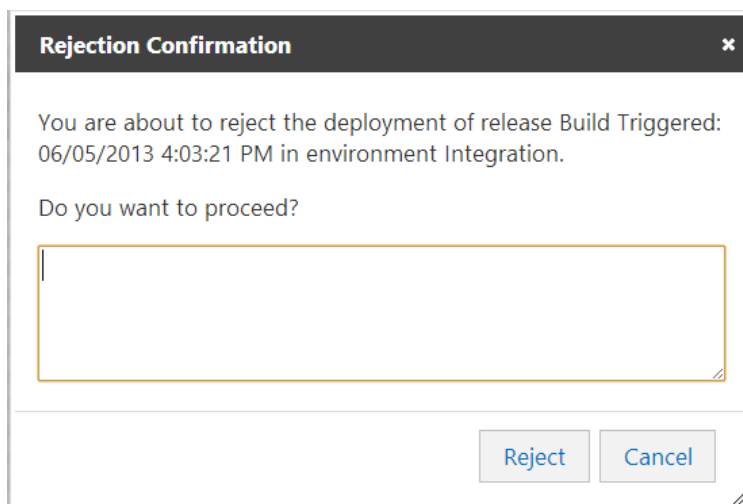


The dialog box has a dark header bar with the title "Approval Confirmation" and a close button (X). The main content area contains the text "You are about to accept the deployment of release Build Triggered: 06/05/2013 4:03:21 PM in environment Integration." followed by the question "Do you want to proceed?". Below this is a large, empty rectangular text box with a yellow border. Underneath the text box is the label "Deferred deployment:" followed by a smaller, empty rectangular text box. At the bottom right of the dialog are two buttons: "Approve" and "Cancel".

5.2.4 Rejecting a Release

To reject a Release:

1. Select the Release you want to reject by choosing it.
2. Choose the **Reject** button. The **Rejection Confirmation** dialog appears.
3. On the **Rejection Confirmation** dialog, observe the following:
 - a. The **Do you want to proceed?** text box is used to put a comment for the action. Reject your release by choosing the Reject button.
4. Reject the Release by choosing the **Reject** button.

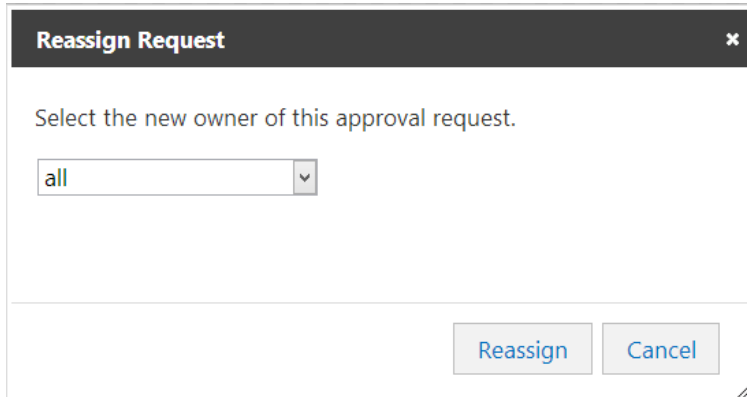


The dialog box has a dark header bar with the title "Rejection Confirmation" and a close button (X). The main content area contains the text "You are about to reject the deployment of release Build Triggered: 06/05/2013 4:03:21 PM in environment Integration." followed by the question "Do you want to proceed?". Below this is a large, empty rectangular text box with a yellow border. At the bottom right of the dialog are two buttons: "Reject" and "Cancel".

5.2.5 Reassigning a Release

To reassign a Release:

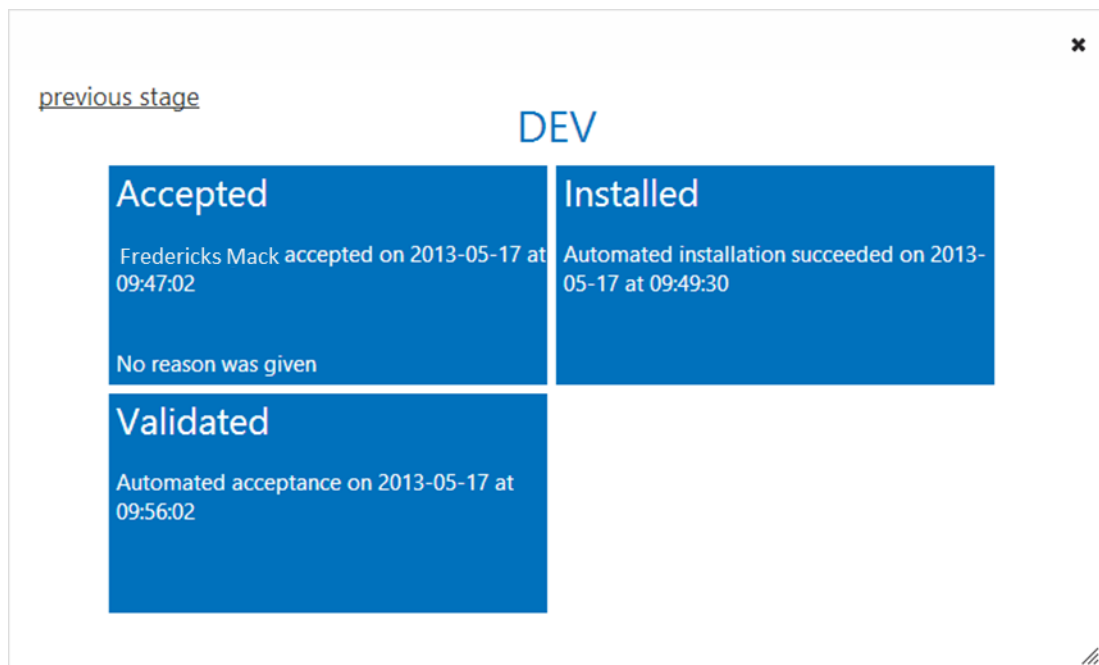
1. Select the release you want to **reassign** by choosing it.
2. Choose the **Reassign** button. The Reassign Request dialog appears.
3. Choose the **User/Group** to whom you want to reassign the release.
4. Reassign your approval by choosing the **Reassign** button.



The image shows a 'Reassign Request' dialog box. It has a dark header bar with the title 'Reassign Request' and a close button (X). Below the header, the text 'Select the new owner of this approval request.' is displayed. Underneath this text is a dropdown menu currently showing 'all'. At the bottom right of the dialog, there are two buttons: 'Reassign' and 'Cancel'. A small double-slash icon is visible in the bottom right corner of the dialog area.

5.3 Previously Approved Releases

The **Previously Approved** view shows a list of all previously acted on releases for a specific user. This view is intended to allow a user to see what is happening to a release he has participated in.



The image shows a 'Previously Approved Releases' view. It has a light gray background and a close button (X) in the top right corner. In the top left, there is a link labeled 'previous stage'. In the top center, the text 'DEV' is displayed. Below this, there are three blue rectangular boxes representing release stages. The first box is titled 'Accepted' and contains the text 'Fredericks Mack accepted on 2013-05-17 at 09:47:02' and 'No reason was given'. The second box is titled 'Installed' and contains the text 'Automated installation succeeded on 2013-05-17 at 09:49:30'. The third box is titled 'Validated' and contains the text 'Automated acceptance on 2013-05-17 at 09:56:02'. A small double-slash icon is visible in the bottom right corner of the view area.

Appendix

A1. SECURITY MANAGEMENT

Release Path Security

Introduction

The release path can be secured to control who can view, edit or manage security on them. Security definition of the release path is a “most permissive” kind of security. For example, if a user is in 2 different groups and a release path is accessible for one of the 2 group and not for the other group, this user will have access to this release path. Also, keep in mind that if a user is a release manager, he will have access to view and edit all release paths. The manage security permission is used to delegate security management to other groups in the system.

Prerequisites

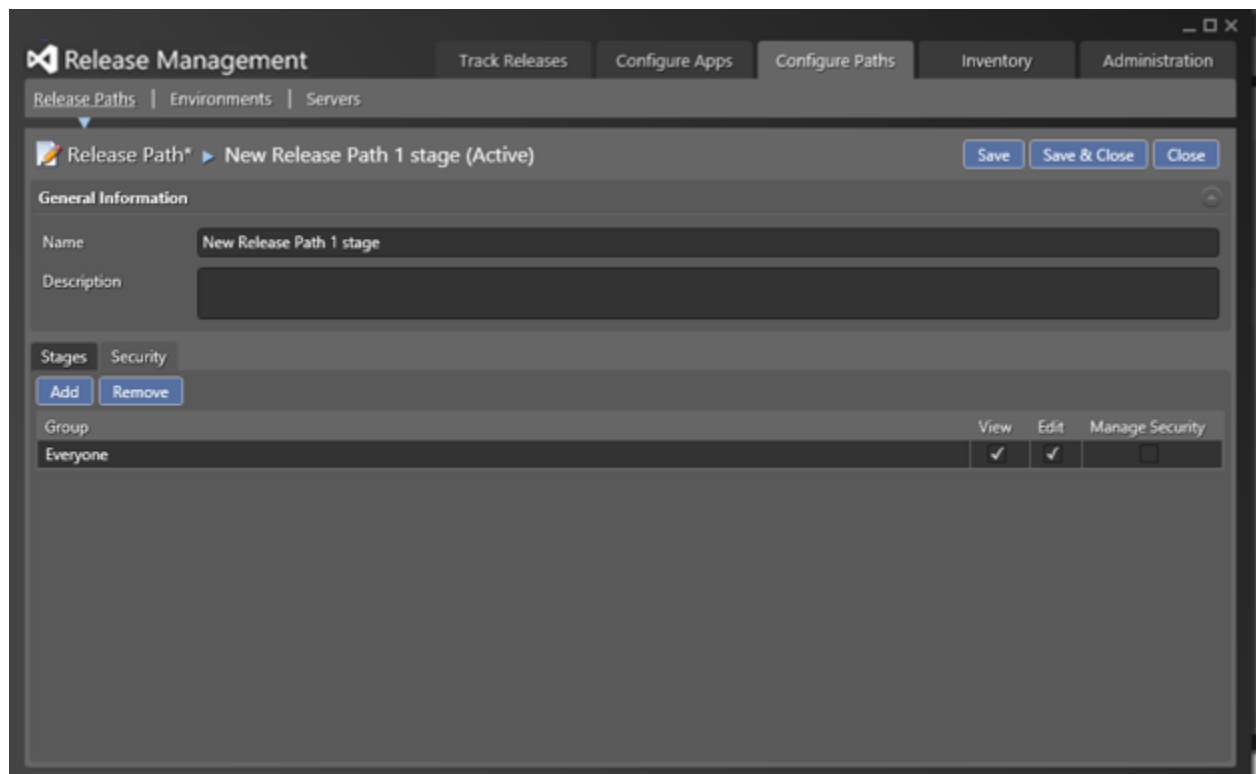
Here are some prerequisites in order to use this security level:

- In release management, Security groups, Stages and Application Categories are defined.

Define Security

Define “View” “Edit” and “Can Manage Security” through Release Paths

1. Navigate to **Configure Paths, Release Paths**.
2. **Open** or create a **New Release Path**
3. Choose the **Security** tab.

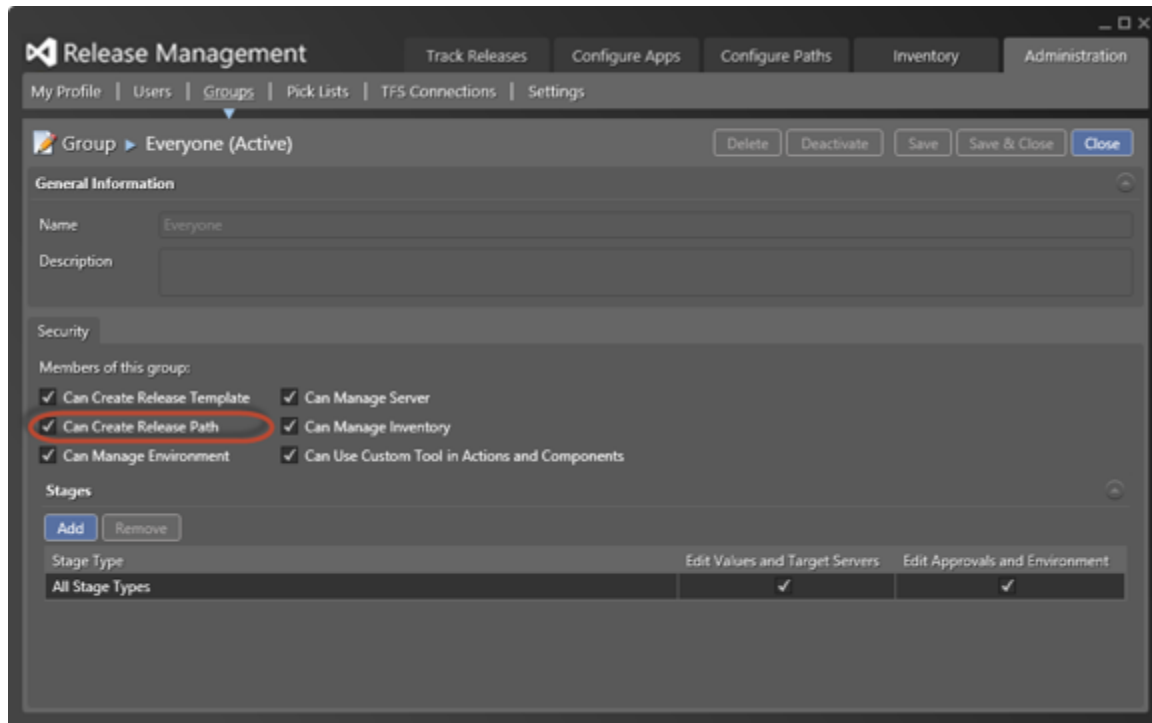


4. By default, the system’s reserved group named **Everyone** will have **View**, **Edit** and **Manage Security** permissions. Since the security is “most permissive,” you will have to clear the **View**, **Edit** and **Manage Security** permissions of this group to be able to control manually over other groups.

5. Choose the **Add** button to access the group lookup and select one or more group (hold CTRL key to select more than one).
6. Select **View**, **Edit** or **Manage Security** permissions as needed for that group. Note that if a group has the **Edit** permission, he will have access to the **View** as well.
7. Choose the **Save & Close** button of the release path to save all security changes.

Define “Can Create Release Path” Security through Groups

8. Navigate to **Administration, Groups**.
9. **Open** or create a **New** group
10. Choose the **Security** tab.

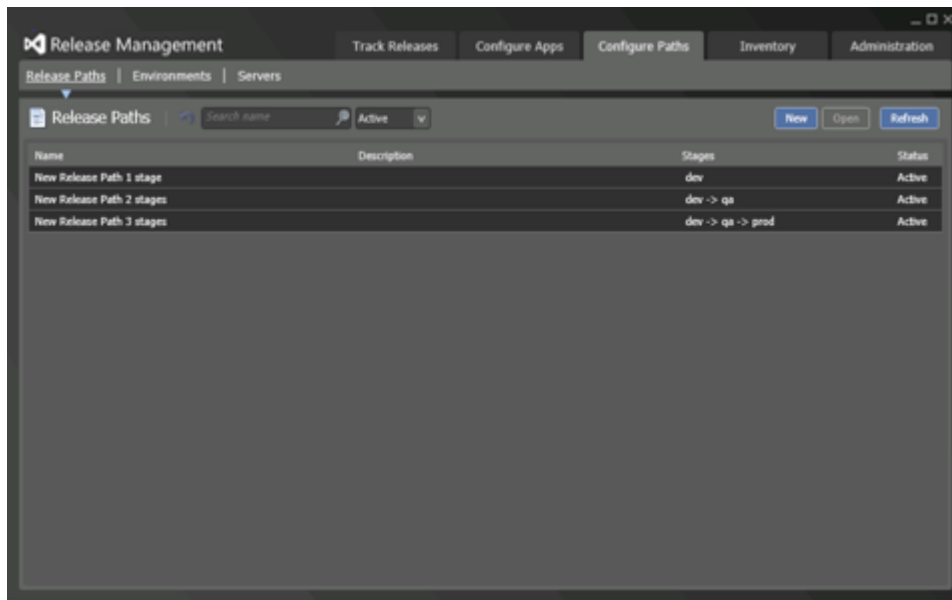


11. By default, the system's reserved group named **Everyone** will be checked with **Can Create Release Path** permissions. Since the security is “most permissive,” you will have to clear the **Can Create Release Path** permission of this group to be able to control manually over other groups.
12. Choose the **Add** button to access the group lookup and select one or more group (hold CTRL key to select more than one).
13. Select **Can Create Release Path** permissions as needed for that group.
14. Choose the **Save & Close** button of the group to save all security changes.

What is affected?

Release Paths

Only Release Paths accessible with the **View** or **Edit** permission for the user will be visible in the list. When a user opens a Release Path which is not allowed in Edit, it will be in Read-Only and saving will not be allowed. If the user is a release manager, all release paths will be visible and editable. Also, if the user is not part of a group that has access to the **Can Create Release Path** security, the **New** button will not be accessible. The security tab is only available when the current user is a release manager or if the current user is member of at least one group who has the **Manage Security** permission.



Release Template Security

Introduction

The release template can be secured to control who can view, edit or manage security on them. Security definition of the release template is a “most permissive” kind of security. For example, if a user is in 2 different groups and a release template is accessible for one of the 2 group and not for the other group, this user will have access to this release template. Also, keep in mind that if a user is a release manager, he will have access to view, edit and manage security on all release templates. The manage security permission is used to delegate security management to other groups in the system.

Prerequisites

Here are some prerequisites in order to use this security level:

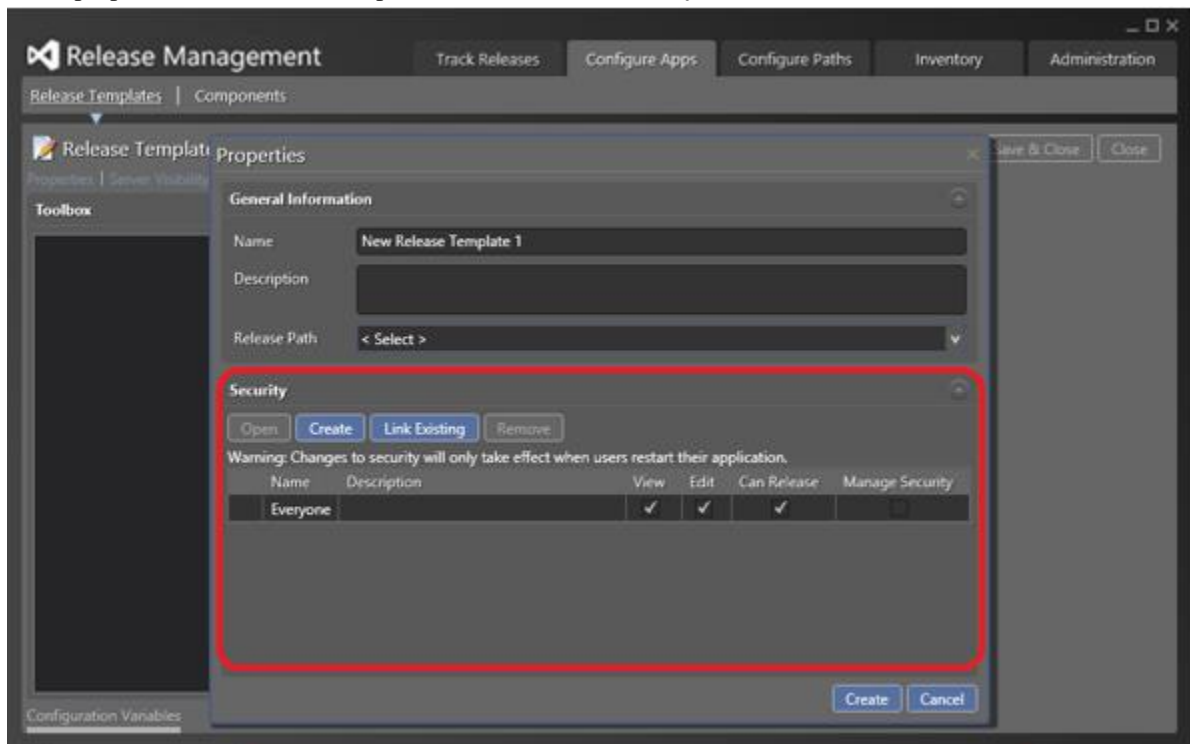
- Security groups, Application Categories and Release Paths are defined.

Define Security

Define Security through Release Template

1. Navigate to **Configure Apps, Release Templates**.
2. **Open** or create a **New** Release Template (it is also possible to create a copy of an existing one).

3. In the properties of the release template, look under the **Security** section.

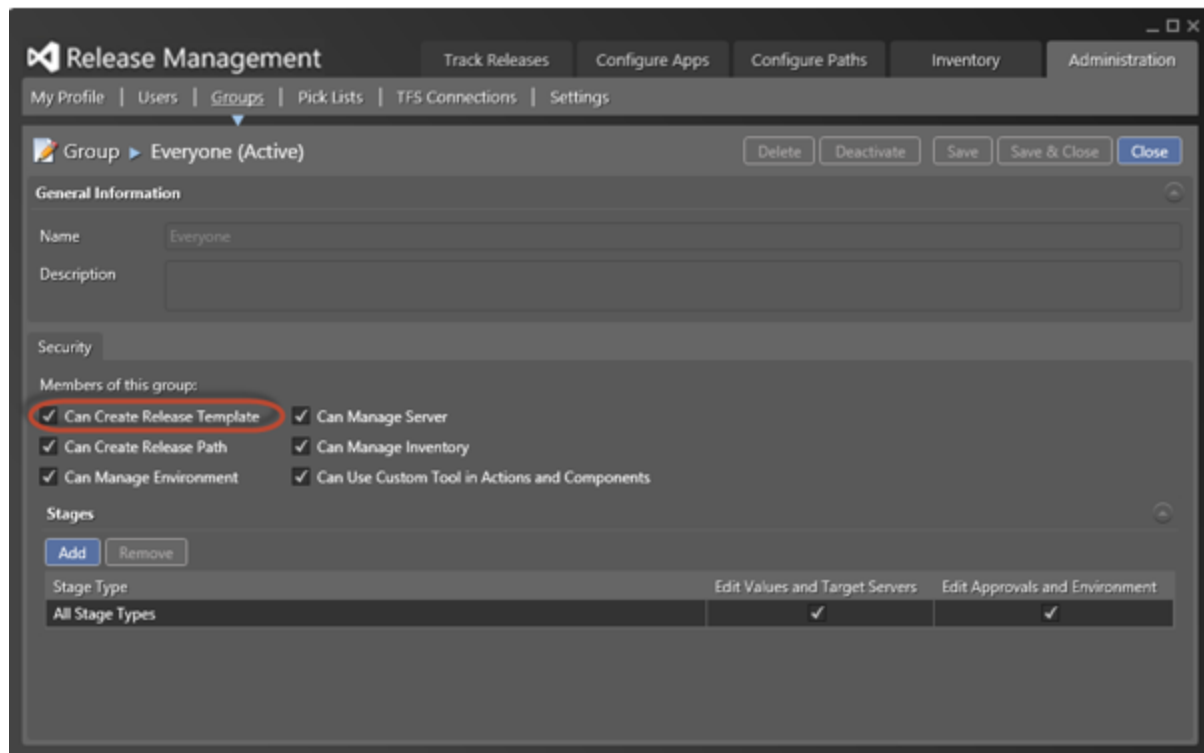


4. By default, the system's reserved group named **Everyone** will be checked with **View**, **Edit**, **Can Release** and **Manage Security** permissions. Since the security is "most permissive," you will have to clear the **View**, **Edit**, **Can Release** and **Manage Security** permissions of this group to be able to control manually over other groups.
5. Choose the **Add** button to access the group lookup and select one or more group (hold CTRL key to select more than one).
6. Select **View**, **Edit**, **Can Release** and **Manage Security** permissions as needed for that group. Note that if a group has the **Edit** permission, he will have access to the **View** as well.
7. Choose the **Save & Close** button of the release path to save all security changes.

Define Create Security through Groups

15. Navigate to **Administration, Groups**.
16. **Open** or create a **New** group

17. Choose the **Security** tab.

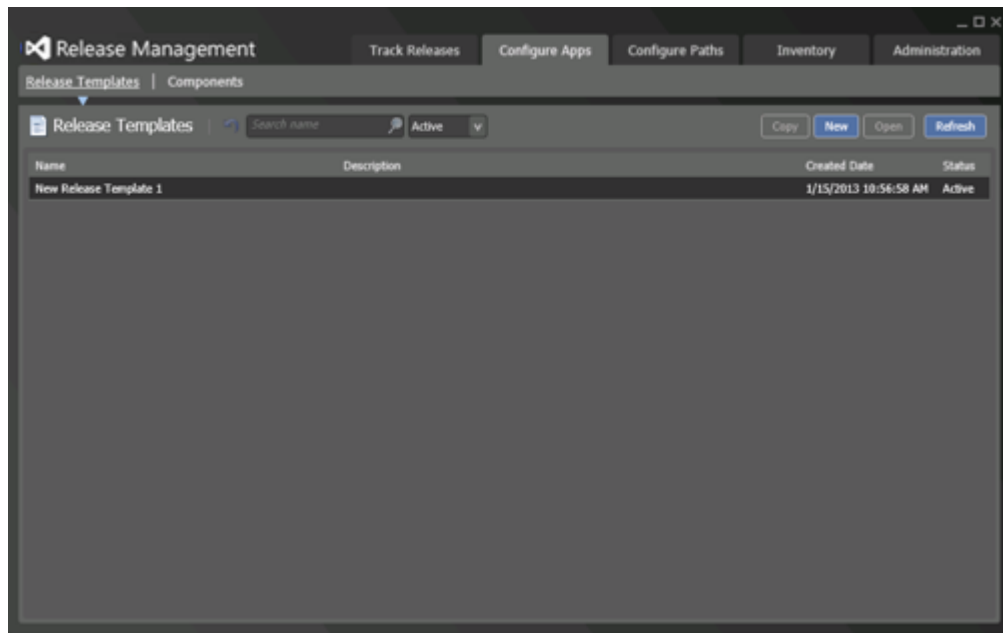


18. By default, the system's reserved group named **Everyone** will have **Can Create Release Template** permissions. Since the security is "most permissive," you will have to clear the **Can Create Release Template** permission of this group to be able to control manually over other groups.
19. Choose the **Add** button to access the group lookup and select one or more group (hold CTRL key to select more than one).
20. Choose **Can Create Release Template** permissions as needed for that group.
21. Choose the **Save & Close** button of the group to save all security changes.

What is affected?

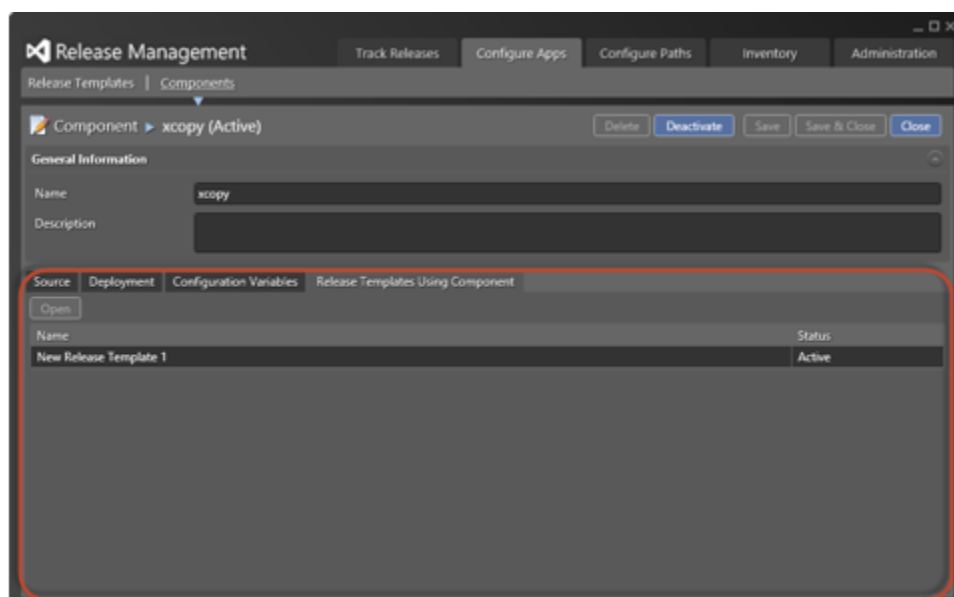
Release Templates

Only release templates accessible with the **View** or **Edit** permission for the user will be visible in the list. When a user opens a release template which is not allowed in Edit, it will be in Read-Only and saving will not be allowed. If the user is a release manager, all release templates will be visible and editable. Also, if the user is not part of a group that has access to the **Can Create Release Template** security, the **New** button will not be accessible. The security section is only available when the current user is a release manager or if the current user is member of at least one group who has the **Manage Security** permission.




Components

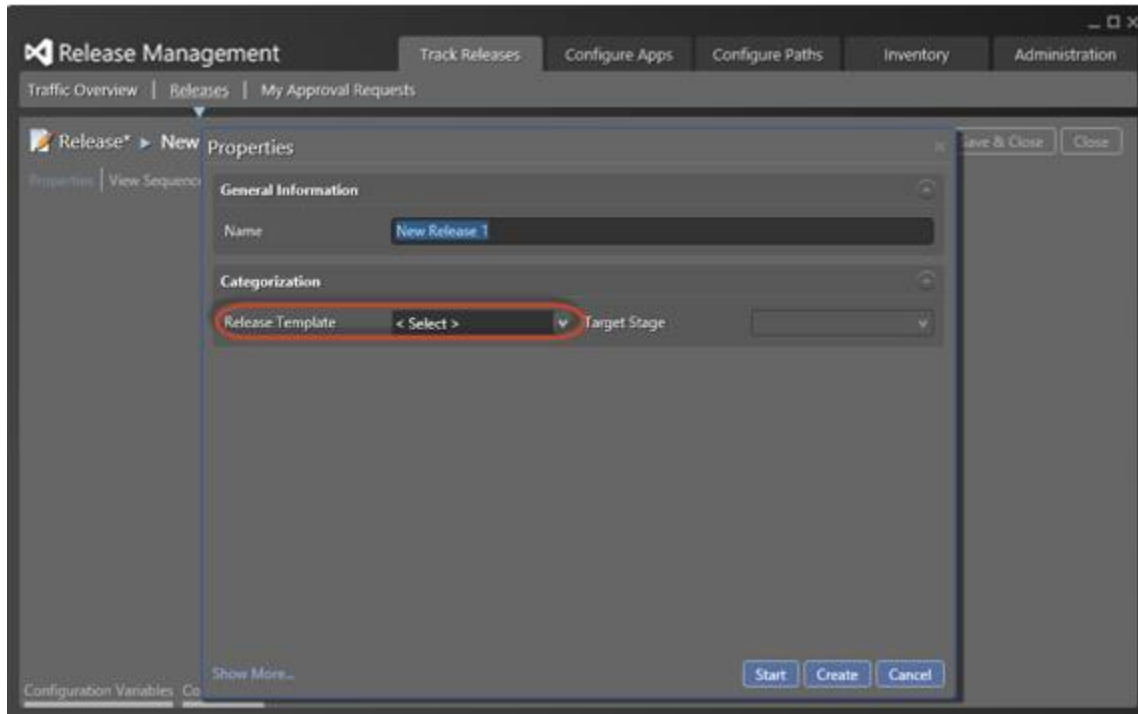
In the **Release Templates Using Component**, all release templates of the component will be available but the **Open** button will not be available if the selected release template is not **View** or **Edit** for the user. If the release template is **View** but not **Edit** for the user, the Open button will be available but no modification will be allowed in the opened release template. If the user is a release manager, all release templates will be visible and editable.



Track Release

From the release list's advanced filter (button ) , only the release template available in View or Edit will be available.

When creating a new release, only **Release Templates** configured with the **Can Release** permission for the user will be visible in the dropdown list of the **Categorization** tab. If the user is a release manager, all release templates will be visible.



Release Security

Introduction

The releases can be secured to control who can view them. Security definition of the releases is a “most permissive” kind of security. For example, if a user has access to the release path of the release and not to the release template, this user will have access to the release. Also, keep in mind that if a user is a release manager, he will have access to view and edit all releases.

Prerequisites

Here are some prerequisites in order to use this security level:

- all data required to create a release is defined.

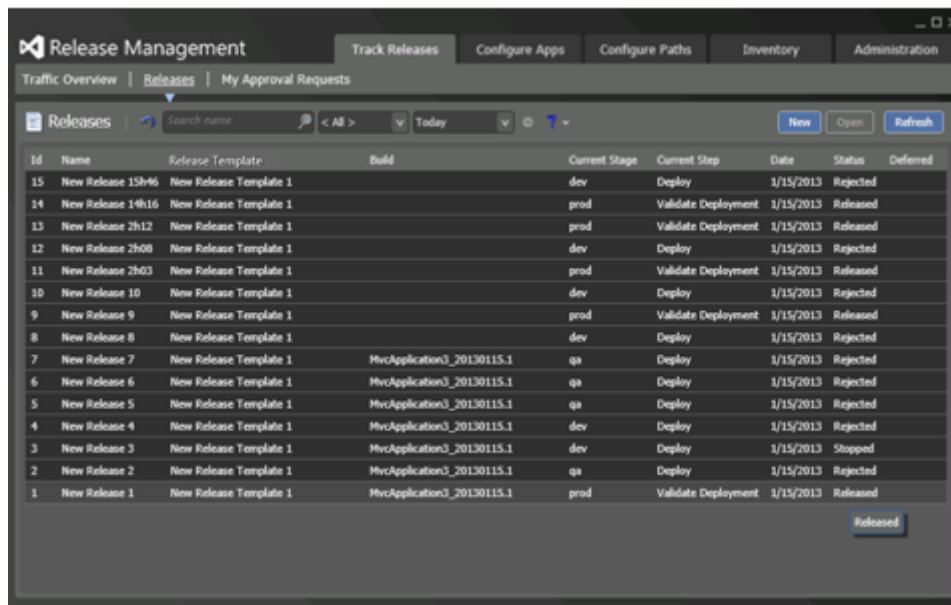
Define Security

1. All releases containing a release template accessible to the user will be accessible.
2. All releases having the user or a group containing the user will be accessible.
3. All releases will be accessible to a release manager.

What is affected?

Track Releases

A release will be visible for a user only if this user has access to its release template or is an approver or is a release manager.

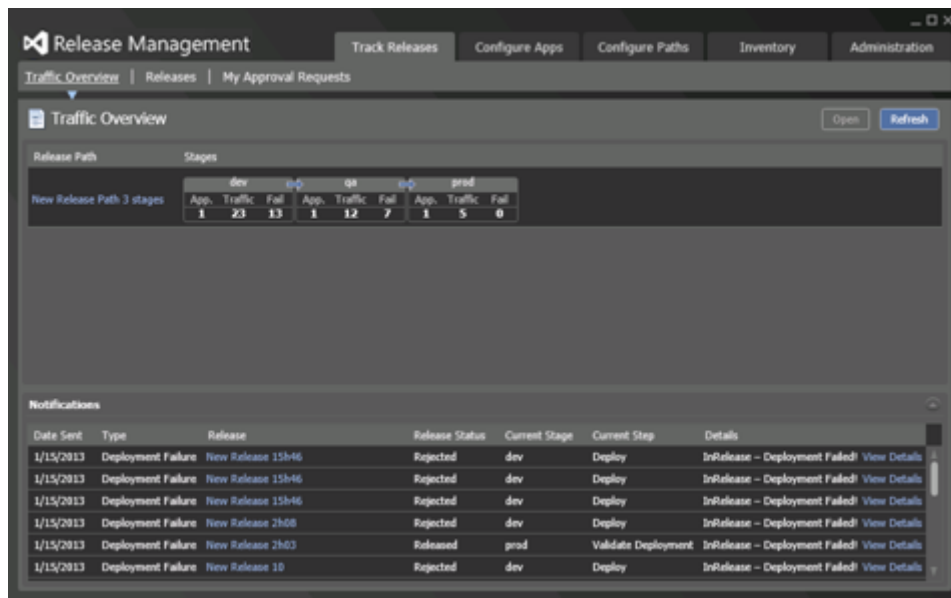


The screenshot shows the 'Track Releases' tab in the Release Management interface. It displays a table of releases with columns: Id, Name, Release Template, Build, Current Stage, Current Step, Date, Status, and Deferred. The table lists 15 releases, with statuses ranging from 'Rejected' to 'Released'. A 'Released' button is visible at the bottom right of the table.

Id	Name	Release Template	Build	Current Stage	Current Step	Date	Status	Deferred
15	New Release 15h46	New Release Template 1		dev	Deploy	1/15/2013	Rejected	
14	New Release 14h16	New Release Template 1		prod	Validate Deployment	1/15/2013	Released	
13	New Release 2h12	New Release Template 1		prod	Validate Deployment	1/15/2013	Released	
12	New Release 2h08	New Release Template 1		dev	Deploy	1/15/2013	Rejected	
11	New Release 2h03	New Release Template 1		prod	Validate Deployment	1/15/2013	Released	
10	New Release 10	New Release Template 1		dev	Deploy	1/15/2013	Rejected	
9	New Release 9	New Release Template 1		prod	Validate Deployment	1/15/2013	Released	
8	New Release 8	New Release Template 1		dev	Deploy	1/15/2013	Rejected	
7	New Release 7	New Release Template 1	HvcApplication3_20130115.1	qa	Deploy	1/15/2013	Rejected	
6	New Release 6	New Release Template 1	HvcApplication3_20130115.1	qa	Deploy	1/15/2013	Rejected	
5	New Release 5	New Release Template 1	HvcApplication3_20130115.1	qa	Deploy	1/15/2013	Rejected	
4	New Release 4	New Release Template 1	HvcApplication3_20130115.1	dev	Deploy	1/15/2013	Rejected	
3	New Release 3	New Release Template 1	HvcApplication3_20130115.1	dev	Deploy	1/15/2013	Stopped	
2	New Release 2	New Release Template 1	HvcApplication3_20130115.1	qa	Deploy	1/15/2013	Rejected	
1	New Release 1	New Release Template 1	HvcApplication3_20130115.1	prod	Validate Deployment	1/15/2013	Released	

Release Traffic Overview

When consulting the release history, releases will be visible for the user only if this user has access to its release template or is an approver or is a release manager.



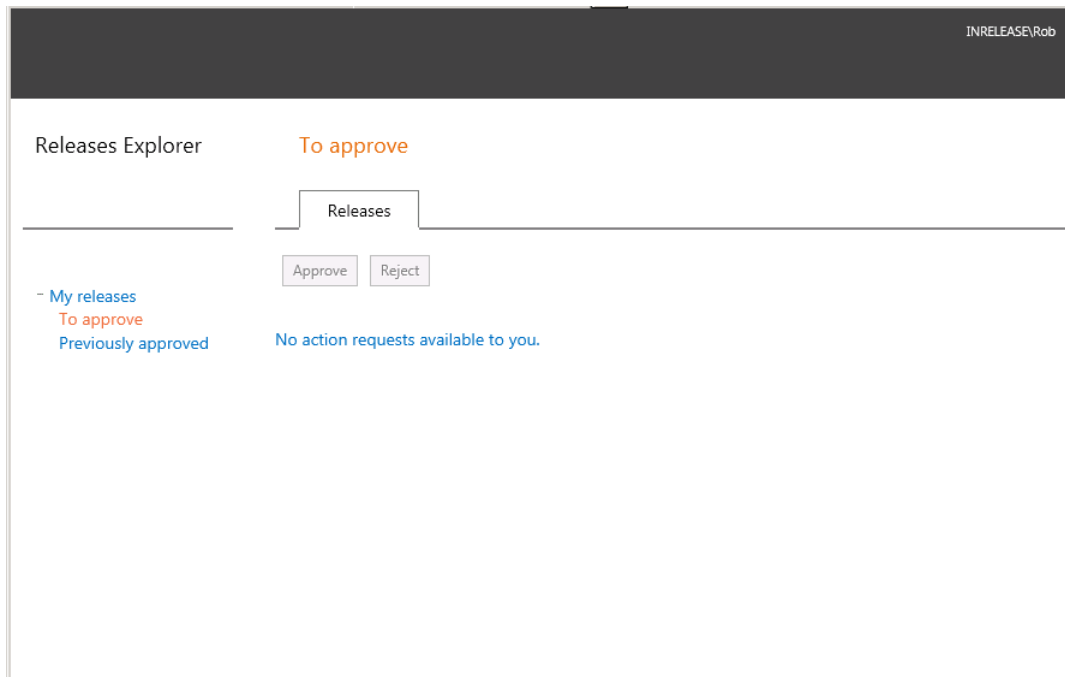
The screenshot shows the 'Traffic Overview' tab in the Release Management interface. It displays a 'Release Path' section with stages (dev, qa, prod) and a table of traffic data. Below this is a 'Notifications' section with a table of release events.

dev			qa			prod		
App.	Traffic	Fail	App.	Traffic	Fail	App.	Traffic	Fail
1	23	13	1	12	7	1	5	0

Date Sent	Type	Release	Release Status	Current Stage	Current Step	Details
1/15/2013	Deployment Failure	New Release 15h46	Rejected	dev	Deploy	InRelease - Deployment Failed! View Details
1/15/2013	Deployment Failure	New Release 15h46	Rejected	dev	Deploy	InRelease - Deployment Failed! View Details
1/15/2013	Deployment Failure	New Release 15h46	Rejected	dev	Deploy	InRelease - Deployment Failed! View Details
1/15/2013	Deployment Failure	New Release 2h08	Rejected	dev	Deploy	InRelease - Deployment Failed! View Details
1/15/2013	Deployment Failure	New Release 2h03	Released	prod	Validate Deployment	InRelease - Deployment Failed! View Details
1/15/2013	Deployment Failure	New Release 10	Rejected	dev	Deploy	InRelease - Deployment Failed! View Details

Release Explorer (Web)

A release will be visible for a user only if this user has access to its release template or is an approver or is a release manager.



Stage Type Security

Introduction

The Stage Type can be secured to control who can edit the Configuration Variables, the Target Servers, the Approver Selection and which Environments can be selected. Security definition of the Stage Type is “most permissive.” For example, if a user is in 2 different groups and **Edit** is enabled in one of the 2 groups but not in the other group, the user will be considered to have the rights to edit. Also, keep in mind that if a user is a Release Manager, he will have access to view and edit Configuration Variables of any Stage Type.

Prerequisites

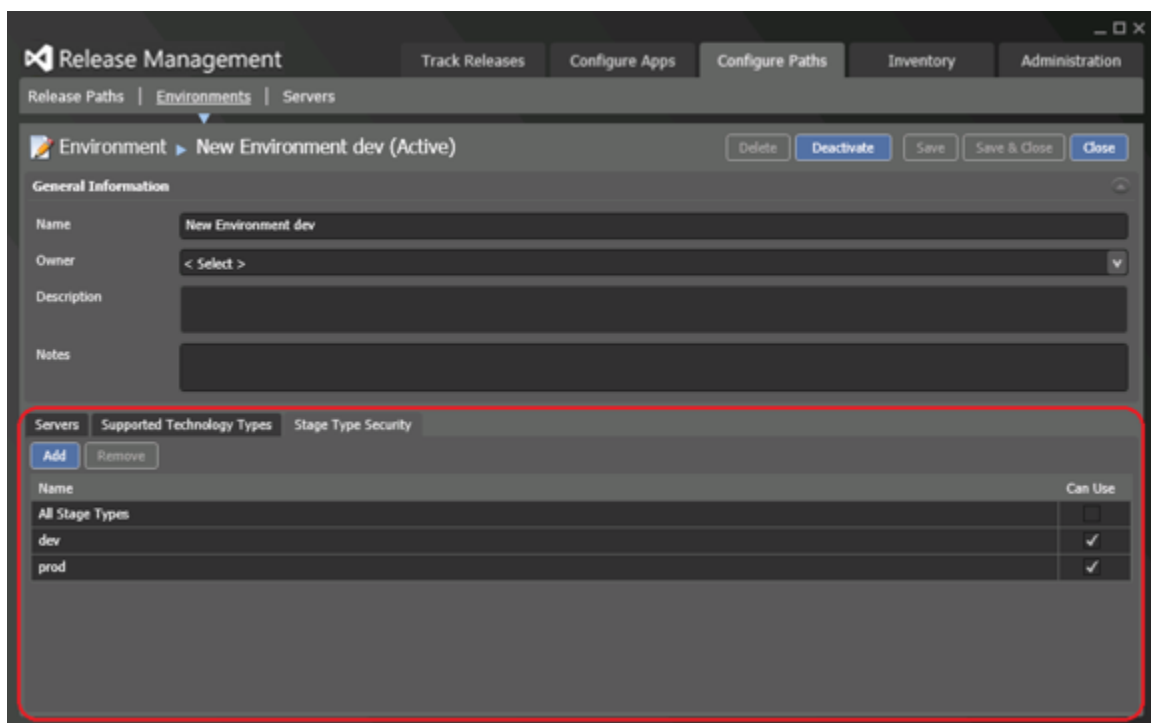
Here are some prerequisites in order to use this security level:

- You must log in using a Release manager account since Security Groups are only available to them.

Define Security

Allow limitation of stage types used by environment

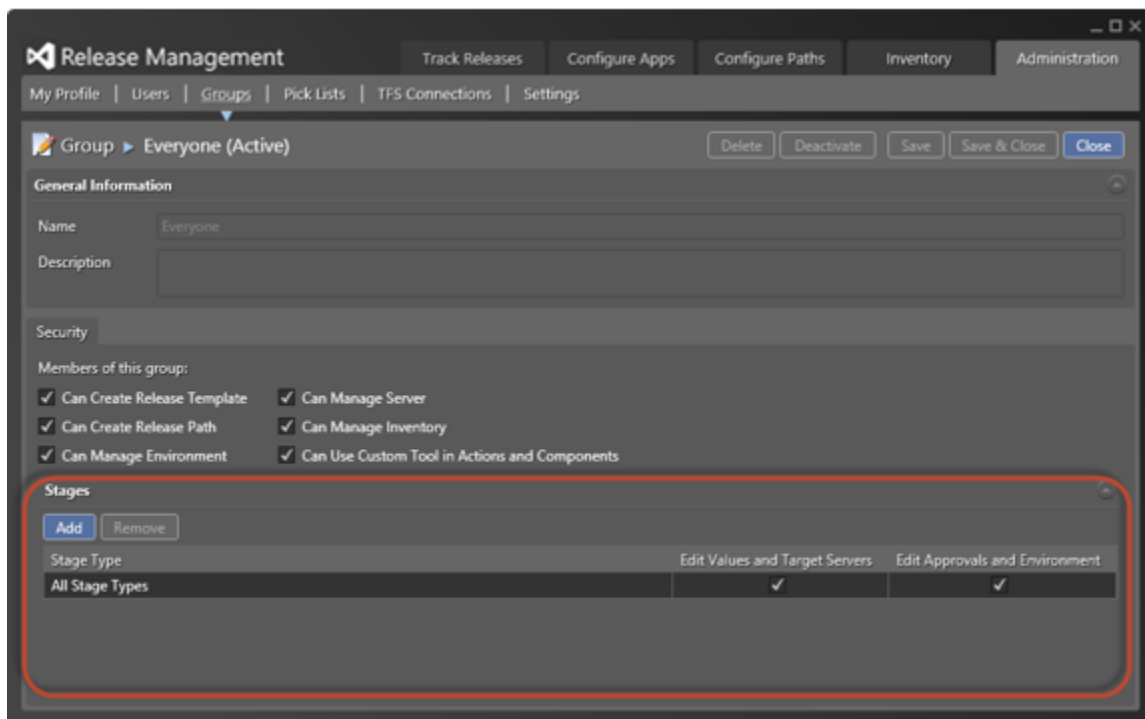
1. Navigate to **Configure Paths, Environments**.
2. **Open** or create a **New** Environment.
3. Choose **Stage Type Security** tab.



4. By default, the system’s reserved stage type named **All Stage Types** will be selected with **Can Use** permission. Since the security is “most permissive,” you will have to clear the **Can Use** permission of this stage type to be able to control it manually over other stage types.
5. Choose the **Add** button to access the stage type lookup and select one or more stage type (hold CTRL key to select more than one).
6. Select **Can Use** permission as needed for that stage type.
7. Choose the **Save** or **Save & Close** button of the environment to save all security changes.

Control variables, target servers, approvals and environment permissions

1. Navigate to **Administration, Groups**.
2. By default, the system's reserved group named **Everyone** will have access to both **Edit Values and Target Servers** and **Edit Approvals and Environment** permissions for **All Stage Types**. Since the security is "most permissive," you will have to clear the **Edit Values and Target Servers** and **Edit Approvals and Environment** permissions of this group for **All Stage Types** to be able to control them manually over other groups and stage types.
3. **Open** or create a **New** group.
4. Choose **Stage Type Security** tab.



5. By default, the system's reserved stage type named **All Stage Types** will be checked with **Edit Values and Target Servers** and **Edit Approvals and Environment** permissions. Since the security is "most permissive," you will have to clear the **Edit Values and Target Servers** and **Edit Approvals and Environment** permissions of this stage type to be able to control them manually over other stage types.
6. Choose the **Add** button to access the stage type lookup and select one or more stage type (hold CTRL key to select more than one).
7. Select **Edit Values and Target Servers** and **Edit Approvals and Environment** permissions as needed for that stage type.
8. Choose the **Save** or **Save & Close** button of the group to save all security changes.

What is affected?

Release Paths

Only environments allowed by the selected stage type will be available in the **Environment** dropdown list. Also, if the user does not have access to **Edit Approvals and Environment** permission, all stage information will be read-only. If the user is a release manager, all information will be editable.

The screenshot shows the 'Release Management' console with the 'Release Paths' tab selected. A new release path is being configured. The 'Stages' tab is active, showing two stage configurations: DEV and QA. Each stage has a set of configuration options. The QA stage configuration is highlighted with a red border. The configuration options for each stage include: Stage (DEV/QA), Environment (DEV/QA), Acceptance Step (Automated checkbox), Approver (Admin dropdown), Deployment Step (Owner dropdown), Validation Step (Automated checkbox), Validator (Admin dropdown), and Approval Step (Approver(s) field).

Release Templates

If the user does not have access to the **Edit Values and Target Servers** permission for a stage type, the information of the Deployment Configuration and Configuration Variables tabs will be read-only for that stage type. If the user is a release manager, all information will be editable.

Releases

If the user does not have access to the **Edit Values and Target Servers** permission for a stage type, the information of the Deployment Sequence and Configuration Variables panel will be read-only for that stage type. If the user is a release manager, all information will be editable.

Environment Security

Introduction

An Environment can be secured to control who can manage them (view, create or edit). Security definition of the Environment is “most permissive.” For example, if a user is in 2 different groups and **Can Manage Environment** is enabled for one of the 2 groups but not in the other group, the user will be considered to have the rights to manage the environments. Also, keep in mind that if a user is a Release Manager, he is able to manage any Environment.

Prerequisites

Here are some prerequisites in order to use this security level:

- You must login using a Release Manager account; Security Groups are only available to Release Managers.

Define Security

1. Navigate to **Administration, Groups**.
2. **Open** or create a **New Group**.
3. In the **Security** section, select or clear the **Can Manage Environment** check box.
4. Be sure to remove the **Can Manage Environment** permission to the **Everyone** group or every user will be able to manage all environments.

What is affected?

When a user does not have the **Can Manage Environment** permission, the menu **Configure Paths → Environments** is hidden.

Server Security

Introduction

A Server can be secured to control who can manage them (view, create or edit). Security definition of the Server is “most permissive.” For example, if a user is in 2 different groups and **Can Manage Server** is enabled in one of the 2 groups but not in the other group, the user will be considered to have the rights to manage the servers. Also, keep in mind that if a user is a Release Manager, he will be able to manage any server.

Prerequisites

Here are some prerequisites in order to use this security level:

- You must login using a Release Manager account since Groups are only available to Release Managers.

Define Security

1. Navigate to **Administration, Groups**.
2. **Open** or create a **New Group**.
3. In the **Security** section, select or clear the **Can Manage Server** check box.
4. Be sure to remove the **Can Manage Server** permission to the **Everyone** group or every user will be able to manage all servers.

What is affected?

When a user does not have the **Can Manage Server** permission, the menu **Configure Paths, Servers** is hidden.

Inventory Security

Introduction

The Inventory (actions and tools) section can be secured to control who can manage its contents (view, create or edit). Security definition of the Inventory is “most permissive.” For example, if a user is in 2 different groups and **Can Manage Inventory** is enabled in one of the 2 groups but not in the other group, the user will be considered to have the rights to manage all inventory. Also, keep in mind that if a user is a Release Manager, he will always be able to manage the inventory.

Prerequisites

Here are some prerequisites in order to use this security level:

- You must login using a Release Manager account since Inventory is only available to Release Managers.

Define Security

1. Navigate to **Administration, Groups**.
2. **Open** or create a **New Group**.
3. In the **Security** section, select or clear the **Can Manage Inventory** check box.
4. Be sure to remove the **Can Manage Inventory** permission to the **Everyone** group or every user will be able to manage the inventory.

What is affected?

When a user does not have the **Can Manage Inventory** permission, the menu **Inventory** is hidden.

Action and Component Security

Introduction

The Actions and Components can be secured to control who can use custom tools when defining them (tools defined manually in the Component without the use of the tool inventory). Security definition of the Actions and Components is “most permissive.” For example, if a user is in 2 different groups and **Can Use Custom Tool in Actions and Components** is enabled in one of the 2 groups but not in the other group, the user will be considered to have the right to use custom tools in the Actions and Components. Also, keep in mind that if a user is a Release Manager, he will always be able to use custom tool.

Prerequisites

Here are some prerequisites in order to use this security level:

- You must login using a Release Manager account since defining the security for Actions and Components are only available to Release Managers.

Define Security

1. Navigate to **Administration, Groups**.
2. **Open** or create a **New Group**.
3. In the **Security** section, select or clear the **Can Use Custom Tool in Actions and Components** check box.
4. Be sure to remove the **Can Use Custom Tool in Actions and Components** permission to the **Everyone** group or every user will still be able to use custom tool.

What is affected?

In **Inventory, Actions**, when a user does not have the **Can Use Custom Tool in Actions and Components** permission, the fields **Command** and **Arguments** will not be editable when the option No Tool is selected.

In **Configure Apps, Components**, when a user does not have the **Can Use Custom Tool in Actions and Components** permission, the fields **Command** and **Arguments** will not be editable when the option No Tool is selected.

A2. ACTIONS CATALOG

Windows Azure

Start a Windows Azure VM

This action allows users to start an existing VM in Windows Azure. It will fail if the VM does not exist or is not accessible. The configuration variables of this action are:

Parameters	Mandatory?	Description
ServiceName	Yes	The name of the Windows Azure service into which the VM exists.
Name	Yes	The name of the Windows Azure VM to start.

Stop a Windows Azure VM

This action allows users to stop an existing VM in Windows Azure. It will fail if the VM does not exist or is not accessible. The configuration variables of this action are:

Parameters	Mandatory?	Description
ServiceName	Yes	The name of the Windows Azure service into which the VM exists.
Name	Yes	The name of the Windows Azure VM to stop.

IIS

Create Application Pool

This action allows users to create a new application pool under IIS. Under IIS 7.0, 7.5, 8.0 and 8.5, if the application pool already exists, it will be reconfigured to match the defined configuration variable values (undefined configuration variables will not affect the actual configuration of the application pool). Under IIS versions prior to 7.0, if the application pool already exists, the deployment will fail. The configuration variables of this action are:

Parameters	Mandatory?	Description
AppPoolName	Yes	The name of the application pool to create or configure.
Is32BitsAllowed	No	Flag that indicates if the application pool allows 32-bit applications. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5. The allowed values are: <ul style="list-style-type: none">- 1 or True: 32-bit applications are allowed- 0 or False: 32-bit applications are not allowed

Parameters	Mandatory?	Description
IdentityUserDomain	No	Domain of the user to use as the identity of the application pool. This parameter is allowed only for IIS 7.0, 7.5, 8.0, and 8.5. If not defined and the "IdentityUserName" is defined, the user will be considered as a local user (i.e. .\localuser).
IdentityUserName	No	User name to use as the identity of the application pool. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5. If not defined, "IdentityUserDomain" and "IdentityUserPassword" will be ignored and the default value of IIS will be used (i.e. ApplicationPoolIdentity).
IdentityUserPassword	No	The password of the user to use as the identity of the application pool. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5.
DotNetVersion	No	The .NET Framework version to use for the application pool. This parameter is allowed only for IIS 7.5 8.0 and 8.5. The allowed values are: v1.0, v1.1, v2.0, v4.0
PipeLineMode	No	The managed pipeline mode to use in the application pool. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5. The allowed values are: <ul style="list-style-type: none"> - Classic - Integrated
ProcessIdleTimeOut	No	The number of minutes a process can be idle in the application pool. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5.
RecycleKbMemory	No	The maximum number of Kb of memory to be used before the application pool is recycled. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5.
RecycleAfterMinutes	No	A fixed number of minutes after which the application pool is recycled. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5.
RecycleFixedTime	No	A fixed time at which the application pool is recycled. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5. The required format is HH:MM (i.e. 23:58 or 01:23).
StartMode	No	The start mode to be used for the application pool. This parameter is allowed only for 8.0 and 8.5. The allowed values are: <ul style="list-style-type: none"> - OnDemand - AlwaysRunning

Parameters	Mandatory?	Description
IsAutoStart	No	<p>Flag that indicate if the application pool must be started automatically. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5. The allowed values are:</p> <ul style="list-style-type: none"> - 1 or True: The application pool will be started automatically when Windows starts. - 0 or False: The application pool will not be started automatically when Windows starts.

Configure Application Pool

This action allows users to configure an existing application pool under IIS, 7.5, 8.0 and 8.5. It will fail a deployment if the application pool does not exist. The configuration variables of this action are:

Parameters	Mandatory?	Description
AppPoolName	Yes	The name of the application pool to configure.
Is32BitsAllowed	No	<p>Flag that indicate if the application pool must allow 32-bit applications. The allowed values are:</p> <ul style="list-style-type: none"> - 1 or True: 32-bit applications are allowed - 0 or False: 32-bit applications are not allowed
IdentityUserDomain	No	Domain of the user to use as the identity of the application pool. If not defined and the "IdentityUserName" is defined, the user will be considered as a local user (i.e. .\localuser).
IdentityUserName	No	User name to use as the identity of the application pool. If not defined, "IdentityUserDomain" and "IdentityUserPassword" will be ignored and the default value of IIS will be used (i.e. ApplicationPoolIdentity).
IdentityUserPassword	No	The password of the user to use as the identity of the application pool.
DotNetVersion	No	The .NET Framework version to use for the application pool. The allowed values are: v1.0, v1.1, v2.0, v4.0
PipeLineMode	No	<p>The managed pipeline mode to use in the application pool. The allowed values are:</p> <ul style="list-style-type: none"> - Classic - Integrated
ProcessIdleTimeOut	No	The number of minutes a process can be idle in the application pool.

Parameters	Mandatory?	Description
RecycleKbMemory	No	The maximum number of Kb of memory that be used before the application pool is recycled.
RecycleAfterMinutes	No	A fixed number of minutes after which the application pool is recycled.
RecycleFixedTime	No	A fixed time at which the application pool is recycled. The required format is HH:MM (i.e. 23:58 or 01:23).
StartMode	No	The start mode to be used for the application pool. This parameter is allowed only for IIS 8.0 and 8.5 The allowed values are: <ul style="list-style-type: none"> - OnDemand - AlwaysRunning
IsAutoStart	No	Flag that indicate if the application pool must be started automatically. The allowed values are: <ul style="list-style-type: none"> - 1 or True: The application pool will be started automatically when Windows start. - 0 or False: The application pool will not be started automatically when Windows start.

Remove Application Pool

This action allows users to delete an existing application pool under IIS 7.0, 7.5, 8.0 and 8.5. It will NOT fail a deployment if the application pool does not exist. The configuration variables of this action are:

Parameters	Mandatory?	Description
AppPoolName	Yes	The name of the application pool to remove.

Recycle Application Pool

This action allows users to recycle an existing application pool under IIS 7.0, 7.5, 8.0 and 8.5. It will fail a deployment if the application pool does not exist. The configuration variables of this action are:

Parameters	Mandatory?	Description
AppPoolName	Yes	The name of the application pool to recycle.

Start Application Pool

This action allows users to start an existing application pool under IIS 7.0, 7.5, 8.0 and 8.5. It will fail a deployment if the application pool does not exist but will not fail if the application pool is already started. The configuration variables of this action are:

Parameters	Mandatory?	Description
AppPoolName	Yes	The name of the application pool to start.

Stop Application Pool

This action allows users to stop an existing application pool under IIS 7.0, 7.5, 8.0 and 8.5. It will fail a deployment if the application pool does not exist but will not fail if the application pool is already stopped. The configuration variables of this action are:

Parameters	Mandatory?	Description
AppPoolName	Yes	The name of the application pool to stop.

Create Web Site

This action allows users to create a new web site under IIS 7.0, 7.5, 8.0 and 8.5. If the web site already exists, it will be reconfigured to match the defined configuration variable values (undefined configuration variables will not affect the actual configuration of the web site). Under IIS versions earlier than 7.0, if the web site already exists, the deployment will fail. The configuration variables of this action are:

Parameters	Mandatory?	Description
SiteName	Yes	The name of the web site to create or configure.
PortNumber	Yes	The port number to use for the new / modified web site.
PhysicalPath	Yes	The complete physical path of the web site.
AppPoolName	No	The name of the application pool to use for the new / modified web site.
IsPreloadEnabled	No	Flag that indicate if the preload functionality is enabled for the web site. This parameter is allowed only for 8.0 and 8.5. The allowed values are: <ul style="list-style-type: none"> - 1 or True: Preload is enabled - 0 or False: Preload is disabled
IsAutoStart	No	Flag that indicate if the web site must be started automatically when Windows starts. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5. The allowed values are: <ul style="list-style-type: none"> - 1 or True: Preload is enabled - 0 or False: Preload is disabled

Configure Web Site

This action allows users to configure an existing web site under IIS 7.0, 7.5, 8.0 and 8.5. It will fail a deployment if the web site does not exist. The configuration variables of this action are:

Parameters	Mandatory?	Description
SiteName	Yes	The name of the web site to configure.
PortNumber	No	The port number to use for the web site to configure.
PhysicalPath	No	The complete physical path of the web site to configure.
AppPoolName	No	The name of the application pool to use for the web site to configure.
IsPreloadEnabled	No	Flag that indicate if the preload functionality is enabled for the web site. The allowed values are: <ul style="list-style-type: none"> - 1 or True: Preload is enabled - 0 or False: Preload is disabled
IsAutoStart	No	Flag that indicate if the web site must be started automatically when Windows starts. The allowed values are: <ul style="list-style-type: none"> - 1 or True: Preload is enabled - 0 or False: Preload is disabled

Remove Web Site

This action allows users to remove an existing web site under IIS 7.0, 7.5, 8.0 and 8.5. It will NOT fail a deployment if the web site does not exist. The configuration variables of this action are:

Parameters	Mandatory?	Description
SiteName	Yes	The name of the web site to remove.

Start Web Site

This action allows users to start an existing web site under IIS 7.0, 7.5, 8.0 and 8.5. It will fail a deployment if the web site does not exist but will not fail if the web site is already started. The configuration variables of this action are:

Parameters	Mandatory?	Description
SiteName	Yes	The name of the web site to start.

Stop Web Site

This action allows users to stop an existing web site under IIS 7.0, 7.5, 8.0 and 8.5. It will fail a deployment if the web site does not exist but will not fail if the web site is already stopped. The configuration variables of this action are:

Parameters	Mandatory?	Description
SiteName	Yes	The name of the web site to stop.

Restart Web Site

This action allows users to restart an existing web site under IIS 7.0, 7.5, 8.0 and 8.5. It will fail a deployment if the web site does not exist but will not fail if the web site is already stopped. The configuration variables of this action are:

Parameters	Mandatory?	Description
SiteName	Yes	The name of the web site to restart.

Create Web Application

This action allows users to create a new web application under IIS 7.0, 7.5, 8.0 and 8.5. If the web application already exists, it will be reconfigured to match the defined configuration variable values (undefined configuration variables will not affect the actual configuration of the web application). Under IIS versions earlier than 7.0, if the web application already exists, the deployment will fail. The configuration variables of this action are:

Parameters	Mandatory?	Description
WebAppName	Yes	The name of the web application to create or configure.
WebSiteName	No	The name of the web site under which the web application resides. If not defined, the Default Web Site will be used.
PhysicalPath	Yes	The complete physical path of the web application.
AppPoolName	No	The name of the application pool to use for the web application to create or configure.
IsPreloadEnabled	No	Flag that indicate if the preload functionality is enabled for the web application. This parameter is allowed only for IIS 8.0 and 8.5. The allowed values are: <ul style="list-style-type: none">- 1 or True: Preload is enabled- 0 or False: Preload is disabled

Configure Web Application

This action allows users to configure an existing web application under IIS 7.0, 7.5, 8.0 and 8.5. It will fail a deployment if the web application does not exist. The configuration variables of this action are:

Parameters	Mandatory?	Description
WebAppName	Yes	The name of the web application to configure.
WebSiteName	No	The name of the web site under which the web application resides. If not defined, the Default Web Site will be used.
PhysicalPath	No	The complete physical path of the web application.
AppPoolName	No	The name of the application pool to use for the web application to configure.

Parameters	Mandatory?	Description
IsPreloadEnabled	No	Flag that indicate if the preload functionality is enabled for the web application. The allowed values are: <ul style="list-style-type: none"> - 1 or True: Preload is enabled - 0 or False: Preload is disabled

Remove Web Application

This action allows users to remove an existing web application under IIS 7.0, 7.5, 8.0 and 8.5. It will NOT fail a deployment if the web application does not exist. The configuration variables of this action are:

Parameters	Mandatory?	Description
WebAppName	Yes	The name of the web application to remove.
WebSiteName	No	The name of the web site under which the web application resides. If not defined, the Default Web Site will be used.

Create Virtual Directory

This action allows users to create a new virtual directory under IIS 7.0, 7.5, 8.0 and 8.5. If the virtual directory already exists, it will be reconfigured to match the defined configuration variable values (undefined configuration variables will not affect the actual configuration of the virtual directory). The configuration variables of this action are:

Parameters	Mandatory?	Description
VirtualDirectoryName	Yes	The name of the virtual directory to create or configure.
SiteName	No	The name of the web site under which the virtual directory resides. If not defined, the Default Web Site will be used.
WebApplicationName	No	The name of the web application under which the virtual directory resides. If not defined, the Root web application of the web site will be used.
PhysicalPath	Yes	The complete physical path of the virtual directory.

Configure Virtual Directory

This action allows users to configure an existing virtual directory under IIS 7.0, 7.5, 8.0 and 8.5. It will fail a deployment if the virtual directory does not exist. The configuration variables of this action are:

Parameters	Mandatory?	Description
VirtualDirectoryName	Yes	The name of the virtual directory to configure.
SiteName	No	The name of the web site under which the virtual directory resides. If not defined, the Default Web Site will be used.

Parameters	Mandatory?	Description
WebApplicationName	No	The name of the web application under which the virtual directory resides. If not defined, the Root web application of the web site will be used.
PhysicalPath	No	The complete physical path of the virtual directory.

Remove Virtual Directory

This action allows users to remove an existing virtual directory under IIS 7.0, 7.5, 8.0, and 8.5. It will NOT fail a deployment if the virtual directory does not exist. The configuration variables of this action are:

Parameters	Mandatory?	Description
VirtualDirectoryName	Yes	The name of the virtual directory to remove.
SiteName	No	The name of the web site under which the virtual directory resides. If not defined, the Default Web Site will be used.
WebApplicationName	No	The name of the web application under which the virtual directory resides. If not defined, the Root web application of the web site will be used.

MS-SQL

Create SQL Database

This action allows users to create a new SQL Server database. It will fail if the database already exists. The configuration variables of this action are:

Parameters	Mandatory?	Description
ServerName	Yes	The server name of the SQL database.
DatabaseName	Yes	The SQL database name to create.

Drop SQL Database

This action allows users to drop a SQL Server database. It will fail if the database does not exist. The configuration variables of this action are:

Parameters	Mandatory?	Description
ServerName	Yes	The server name of the SQL database.
DatabaseName	Yes	The SQL database name to drop.

Backup SQL Database

This action allows users to backup an existing SQL Server database. The configuration variables of this action are:

Parameters	Mandatory?	Description
ServerName	Yes	The server name where the SQL database resides.
DatabaseName	Yes	The SQL database name to backup.
Path	Yes	The path and file name where to put the backup file (i.e. \\server\filename.bak).

Restore SQL Database

This action allows users to restore a SQL Server database by using an existing backup file. The configuration variables of this action are:

Parameters	Mandatory?	Description
DatabaseName	Yes	The SQL database name to restore.
Source	Yes	The path and file name where the backup file resides (i.e. \\server\filename.bak).

Windows OS

Copy File or Folder

This action allows users to copy a file(s) or a folder and its content from one location to another. The configuration variables of this action are:

Parameters	Mandatory?	Description
SourceFileFolder	Yes	The source file(s) or folder to be copied. Wildcards can be used (i.e. c:\temp*.log).
DestinationFileFolder	Yes	The destination path where to copy.

Create Folder

This action allows users to create a new folder structure. Any non-existent parent directory of the folder to create will be created as well. The configuration variables of this action are:

Parameters	Mandatory?	Description
FolderName	Yes	The folder name and path to be created

Delete File(s) or Folder

This action allows deletion of a file(s) or a folder. The configuration variables of this action are:

Parameters	Mandatory?	Description
FileFolderName	Yes	The file(s) or folder name and path to delete. Wildcards can be used (i.e. c:\temp*.log).

Move File or Folder

This action allows moving a file(s) or a folder from one location to another. The configuration variables of this action are:

Parameters	Mandatory?	Description
FileFolderName	Yes	The source file(s) or folder to be moved. Wildcards cannot be used (i.e. c:\temp*.log).
DestinationName	Yes	The destination path where to move.

Rename File or Folder

This action allows renaming a file or a folder. The configuration variables of this action are:

Parameters	Mandatory?	Description
FileFolderName	Yes	The source file(s) or folder to be renamed. Wildcards cannot be used (i.e. c:\temp*.log).
DestinationName	Yes	The new name to be given to the file or folder.

Modify File(s) or Folder Attributes

This action allows modification of the attributes of a file(s) or folder. The configuration variables of this action are:

Parameters	Mandatory?	Description
FileFolderName	Yes	The source file(s) or folder to be modified. Wildcards can be used (i.e. c:\temp*.log).
ReadOnly	No	Allow the user to set or reset the file(s) or folder read-only attribute. <ul style="list-style-type: none"> - + : Set the flag - - : Reset the flag - Any other value : Let the flag as it is
Archive	No	Allow the user to set or reset the file(s) or folder archive attribute. <ul style="list-style-type: none"> - + : Set the flag - - : Reset the flag - Any other value : Let the flag as it is

Parameters	Mandatory?	Description
System	No	Allow the user to set or reset the file(s) or folder system attribute. <ul style="list-style-type: none"> - + : Set the flag - - : Reset the flag - Any other value : Let the flag as it is
Hidden	No	Allow the user to set or reset the file(s) or folder hidden attribute. <ul style="list-style-type: none"> - + : Set the flag - - : Reset the flag - Any other value : Let the flag as it is

Modify File(s) or Folder Owner

This action allows modification of the owner of a file(s) or folder. The configuration variables of this action are:

Parameters	Mandatory?	Description
FileFolderName	Yes	The source file(s) or folder to be modified. Wildcards can be used (i.e. c:\temp*.log).
OwnerDomain	No	Represent the domain of the new owner to affect to the file(s) or folder. If not defined, the user specified will be considered as a local one (i.e. ".\localuser")
OwnerName	Yes	Represent the name of the new owner to affect to the file(s) or folder.

Create Environment Variable

This action allows creation of an environment variable at system level. If the environment variable already exists at system level, its value will be replaced with the new one. The configuration variables of this action are:

Parameters	Mandatory?	Description
VariableName	Yes	The name of the variable to be created / modified
VariableValue	No	The value to give to the added / modified variable. If no value is set, the variable will be deleted.

Modify Environment Variable

This action allows modification of an environment variable at system level. If the environment variable does not exist at system level, it will be created. The configuration variables of this action are:

Parameters	Mandatory?	Description
VariableName	Yes	The name of the variable to be created / modified

Parameters	Mandatory?	Description
VariableValue	No	The new value to give to the variable. If no value is set, the variable will be deleted.

Delete Environment Variable

This action allows deletion of an environment variable at system level. It will not fail if the environment variable does not exist. The configuration variables of this action are:

Parameters	Mandatory?	Description
VariableName	Yes	The name of the variable to be deleted

Create Windows Registry Key

This action allows creating a new registry key. It will not fail if the registry key already exists. The configuration variables of this action are:

Parameters	Mandatory?	Description
RegistryKey	Yes	The complete name of the registry key to create (i.e. HKLM\Software\NEWKEY).
OperatingSystemType	No	The operating system type in which to create the registry key. The allowed values are: <ul style="list-style-type: none"> - 32: The key will be created in the 32-bit registry. If not specified, this will be the default. - 64: The key will be created in the 64-bit registry.

Modify Windows Registry Key

This action allows modifying the name or location of a registry key. It will fail if the registry key does not exist. The configuration variables of this action are:

Parameters	Mandatory?	Description
RegistryKey	Yes	The complete name of the registry key to modify (i.e. HKLM\Software\OLDKEY).
NewRegistryKey	Yes	The complete name of the destination registry key (i.e. HKLM\Software\NEWKEY).
OperatingSystemType	No	The operating system type in which to modify the registry key. The allowed values are: <ul style="list-style-type: none"> - 32: The key will be modified in the 32-bit registry. If not specified, this will be the default. - 64: The key will be modified in the 64-bit registry.

Delete Windows Registry Key

This action allows deleting an existing registry key. It will fail if the registry key does not exist. The configuration variables of this action are:

Parameters	Mandatory?	Description
RegistryKey	Yes	The complete name of the registry key to delete (i.e. HKLM\Software\KEYTODELETE).
OperatingSystemType	No	The operating system type in which to delete the registry key. The allowed values are: <ul style="list-style-type: none">- 32: The key will be deleted in the 32-bit registry. If not specified, this will be the default.- 64: The key will be deleted in the 64-bit registry.

Create Windows Registry Value

This action allows creating a new registry value. If the registry key specified does not exist, it will be created. If the registry value already exists, its data and type will be replaced. The configuration variables of this action are:

Parameters	Mandatory?	Description
RegistryKey	Yes	The complete name of the registry key that contains the value (i.e. HKLM\Software\MYKEY).
RegistryValue	Yes	The name of the registry value to be added / modified.
ValueType	No	The type of data of the registry value. The allowed values are: <ul style="list-style-type: none">- REG_SZ : String (default value if not specified)- REG_QWORD_LITTLE_ENDIAN : A 64-bit number in little-endian format- REG_QWORD : A 64-bit number- REG_NONE : No value- REG_MULTI_SZ : Multi-line string (separated by \0)- REG_LINK : Symbolic Link- REG_EXPAND_SZ : String with unexpanded references to environment variables- REG_DWORD_BIG_ENDIAN : A 32-bit number in big-endian format- REG_DWORD_LITTLE_ENDIAN : A 32-bit number in little-endian format- REG_DWORD : A 32-bit number- REG_BINARY : Binary data in any form
RegistryData	No	The data to affect to the registry value.

Parameters	Mandatory?	Description
OperatingSystemType	No	<p>The operating system type in which to create the registry value. The allowed values are:</p> <ul style="list-style-type: none"> - 32: The value will be created in the 32-bit registry. If not specified, this will be the default. - 64: The value will be created in the 64-bit registry.

Modify Windows Registry Value

This action allows modifying an existing registry value. If the registry key or the registry value specified does not exist, they will be created. The configuration variables of this action are:

Parameters	Mandatory?	Description
RegistryKey	Yes	The complete name of the registry key that contains the value (i.e. HKLM\Software\MYKEY).
RegistryValue	Yes	The name of the registry value to be added / modified.
ValueType	No	<p>The type of data of the registry value. The allowed values are:</p> <ul style="list-style-type: none"> - REG_SZ : String (default value if not specified) - REG_QWORD_LITTLE_ENDIAN : A 64-bit number in little-endian format - REG_QWORD : A 64-bit number - REG_NONE : No value - REG_MULTI_SZ : Multi-line string (separated by \0) - REG_LINK : Symbolic Link - REG_EXPAND_SZ : String with unexpanded references to environment variables - REG_DWORD_BIG_ENDIAN : A 32-bit number in big-endian format - REG_DWORD_LITTLE_ENDIAN : A 32-bit number in little-endian format - REG_DWORD : A 32-bit number - REG_BINARY : Binary data in any form
RegistryData	No	The data to affect to the registry value.
OperatingSystemType	No	<p>The operating system type in which to modify the registry value. The allowed values are:</p> <ul style="list-style-type: none"> - 32: The value will be modified in the 32-bit registry. If not specified, this will be the default. - 64: The value will be modified in the 64-bit registry.

Delete Windows Registry Value

This action allows deleting an existing registry value. It will fail if the registry value does not exist. The configuration variables of this action are:

Parameters	Mandatory?	Description
RegistryKey	Yes	The complete name of the registry key that contains the value (i.e. HKLM\Software\MYKEY).
RegistryValue	Yes	The name of the registry value to be deleted.
OperatingSystemType	No	The operating system type in which to delete the registry value. The allowed values are: <ul style="list-style-type: none">- 32: The value will be deleted in the 32-bit registry. If not specified, this will be the default.- 64: The value will be deleted in the 64-bit registry.

Run Command Line

This action allows execution of a command line. The command line will use the deployment agent identity to run. The configuration variables of this action are:

Parameters	Mandatory?	Description
FilePath	Yes	The complete location and the executable name.
Arguments	No	The arguments required to run the command line.

Run Command Line as User

This action allows execution of a command line as a specific user. The configuration variables of this action are:

Parameters	Mandatory?	Description
FilePath	Yes	The complete location and the executable name.
Arguments	No	The arguments required to run the command line.
UserDomain	No	The domain of the user to use to run the executable. If not defined, the user specified will be considered as a local one (i.e. ".\localuser").
UserName	Yes	The name of the user to use to run the executable.
UserPassword	No	The password of the user to use to run the executable.

Kill Windows Process

This tool allows users to kill a Windows Process. The configuration variables of this tool are:

Parameter	Variable	Mandatory?	Description
-command	N/A	Yes	Fixed value <code>./WindowsProcess.ps1</code> .
-ProcessName		Yes	The name of the Windows process to kill.
-UserName		No	The user account running the process. If not specified, the process will be killed for all users.
-IsKillProcessTree		No	Flag that indicates if the process tree must be killed. Allowed values are: <ul style="list-style-type: none"> - 1: Kill process tree - 0 or empty: Do not kill process tree

Windows Services

Create Service

This action allows users to create a new Windows service. It will fail if the service to create already exists. The configuration variables of this action are:

Parameters	Mandatory?	Description
ServiceName	Yes	Name of the service to create.
BinPath	Yes	Binary path and name of the service.
DisplayName	No	Display name that will be used for this service.
Description	No	Description that will be given to the service.
UserName	No	User name that will be used for the credentials to start the service (must be in the form <code>DOMAIN\USER</code>). If provided, the parameter Password must also be provided. If not provided, the LocalSystem account will be used by default.
Password	No	Password of the user name specified to start the service.
StartMode	No	Start mode of the service. The allowed values are: <ul style="list-style-type: none"> - Manual: Service is started only manually - Automatic: Service will be started automatically next time the computer is restarted. This is the default if not specified. - Disabled: Service is disabled

Configure Service

This action allows users to configure an existing Windows service. It will fail if the service to configure does not exist. The configuration variables of this action are:

Parameters	Mandatory?	Description
ServiceName	Yes	Name of the service to configure.
BinPath	No	Binary path and name of the service.
DisplayName	No	Display name that will be used for this service.
Description	No	Description that will be given to the service.
UserName	No	User name that will be used for the credentials to start the service (must be in the form DOMAIN\USER). If provided, the parameter Password must also be provided. To reset the credentials to the LocalSystem, enter LocalSystem as the user and "" as the password.
Password	No	Password of the user name specified to start the service.
StartMode	No	Start mode of the service. The allowed values are: <ul style="list-style-type: none">- Manual: Service is started only manually- Automatic: Service will be started automatically next time the computer is restarted. This is the default if not specified.- Disabled: Service is disabled

Configure Service Credentials

This action allows users to configure an existing Windows service with new credentials. It will fail if the service to configure does not exist. The configuration variables of this action are:

Parameters	Mandatory?	Description
ServiceName	Yes	Name of the service to configure.
UserName	Yes	User name that will be used for the credentials to start the service (must be in the form DOMAIN\USER). If provided, the parameter Password must also be provided. To reset the credentials to the LocalSystem, enter LocalSystem as the user and "" as the password.
Password	No	Password of the user name specified to start the service.

Delete Service

This action allows users to delete an existing Windows service. It will not fail if the service to delete does not exist. The configuration variables of this action are:

Parameters	Mandatory?	Description
ServiceName	Yes	Name of the service to delete.

Start Service

This action allows users to start an existing Windows service. It will fail if the service to configure does not exist but will not fail if it is already started. The configuration variables of this action are:

Parameters	Mandatory?	Description
ServiceName	Yes	Name of the service to start.

Stop Service

This action allows users to stop an existing Windows service. It will fail if the service to configure does not exist but will not fail if it is already stopped. The configuration variables of this action are:

Parameters	Mandatory?	Description
ServiceName	Yes	Name of the service to stop.

Restart Service

This action allows users to restart an existing Windows service. It will fail if the service to configure does not exist but will not fail if it is stopped. The configuration variables of this action are:

Parameters	Mandatory?	Description
ServiceName	Yes	Name of the service to restart.

INI File

Modify Section Name

This action allows users to change the name of a section in an INI file specified file. If the INI file specified does not exist, it will be created. If no section corresponds to the name to be changed, the action will still succeed. The configuration variables of this action are:

Parameters	Mandatory?	Description
INIFileName	Yes	The INI file name and path (if required)
SectionName	No	The section name to be modified. If not specified, the root section will be renamed (all keys that are at the beginning of the file before the first section is considered in the root).

Parameters	Mandatory?	Description
NewSectionName	No	The new section name to affect. If not specified, the keys of the old section will be put under the root section (all keys that are at the beginning of the file before the first section is considered in the root).

Remove Section

This action allows users to remove a section and all its keys and values in an INI file. If the INI file specified does not exist, the action will fail. If no section corresponds to the name to be removed, the action will still succeed. The configuration variables of this action are:

Parameters	Mandatory?	Description
INIFileName	Yes	The INI file name and path (if required)
SectionName	No	The section name to be removed. If not specified, the root section will be removed (all keys that are at the beginning of the file before the first section is considered in the root).

Create Key and Value

This action allows users to create a new key with a value in a specific section of an INI file. If the INI file specified does not exist, it will be created. If no section corresponds to the one specified, it will be created. If the key already exists in the specified section, its value will be replaced. The configuration variables of this action are:

Parameters	Mandatory?	Description
INIFileName	Yes	The INI file name and path (if required)
SectionName	No	The section name where the key is to be added. If not specified, the root section will be considered (all keys that are at the beginning of the file before the first section is considered in the root).
KeyName	Yes	The name of the key to add.
KeyValue	No	The value to affect to the new key.

Modify Key and Value

This action allows users to modify an existing key with its value in a specific section of an INI file. If the INI file specified does not exist, it will be created. If no section corresponds to the one specified, it will be created. If the key does not exist in the specified section, it will be added. The configuration variables of this action are:

Parameters	Mandatory?	Description
INIFileName	Yes	The INI file name and path (if required)

Parameters	Mandatory?	Description
SectionName	No	The section name where the key is to be modified. If not specified, the root section will be considered (all keys that are at the beginning of the file before the first section is considered in the root).
KeyName	Yes	The name of the key to modify.
KeyValue	No	The new value to affect to the key. If not specified and the NewKeyName is specified, the system will use the actual value. If not specified and the NewKeyName is not specified, the value will be emptied.
NewKeyName	No	The new name to give to the key. If not specified, the key name will not be changed.

Remove Key and Value

This action allows users to remove a key and its values from a specific section in an INI file. If the INI file specified does not exist, the action will fail. If no section or key corresponds to the name to be removed, the action will still succeed. If all keys from a section are removed, the section will be removed as well. The configuration variables of this action are:

Parameters	Mandatory?	Description
INIFileName	Yes	The INI file name and path (if required)
SectionName	No	The section name where the key is to be deleted. If not specified, the root section will be considered (all keys that are at the beginning of the file before the first section is considered in the root).
KeyName	Yes	The name of the key to remove.

A3. TOOLS CATALOG

Database Deployment Agent – Execute Script

This tool allows users to execute a SQL query in a specific database. It is based on the standard sqlcmd executable distributed with SQL Server. There are a lot of possible parameters for this executable but we will list only the one used for this tool. Please refer to the sqlcmd documentation for further information about this executable. The configuration variables of this tool are:

Parameter	Variable	Mandatory?	Description
-S	ServerName	Yes	The server name of the SQL database.
-i	ScriptName	Yes	The name of the script to be executed.
-b	N/A	No	This parameter is a switch. If present, it means that if an error occurs, the batch will be aborted.

Reporting Services Deployment Agent

This tool allows users to deploy reporting services reports and related object. It is based on a custom release management executable. The configuration variables of this tool are:

Parameter	Variable	Mandatory?	Description
-s	serverUrl	Yes	The complete URL of the reporting services server.
-l	itemLocation	Yes if multiple items	The location of the items. If specified, the folder must be provided (-f).
-f	folder	Yes if multiple items	The folder of the items. If specified, the item location must be provided (-l).
-t		Yes	The item type to be processed. The allowed values are: <ul style="list-style-type: none">- Report- Picture
-sp		No	List of search patterns to use to filter the items. Each search pattern must be separated by a comma or a semicolon (i.e. *.bmp;*.jpg). If not specified, the search pattern will be defaulted following the selected item type (-t): <ul style="list-style-type: none">- Report: *.rdl- Picture: *.bmp, *.gif, *.jpg, *.jpeg, *.png

Parameter	Variable	Mandatory?	Description
-i		Yes if single items	Item file. If specified, the item name must be provided (-n). This parameter is used when a single item must be deployed. If so, the item location (-l) and folder (-f) parameters must not be present.
-n		Yes if single items	Item name. If specified, the item file must be provided (-i). This parameter is used when a single item must be deployed. If so, the item location (-l) and folder (-f) parameters must not be present.
-log		No	A log file with this given name will be generated in the Physical Directory.
-o	N/A	No	This parameter is a switch. If present, it means that the destination will be overwritten if it already exists.
-ds ds1,...,dsN			Comma-delimited fully qualified name of all data sources to associate to the item. This option only applies to reports. The data source can be in the format [name:] reference where name is the name of the data source as specified in the report where the references are to be replaced. And reference is the fully qualified reference to the data source in the target server to use for the report.

MSI Deployment Agent

This tool allows users to deploy an MSI and will uninstall the previous installation of the product prior to installing the new one. It is based on a custom release management PowerShell script. The configuration variables of this tool are:

Parameter	Variable	Mandatory?	Description
-command	N/A	Yes	Fixed value ./msiexec.ps1.
-MsiFileName	Installer	Yes	The complete path (if required) and name of the MSI file to install.
-MsiCustomArgs	MsiCustomArgs	No	The custom arguments to use to install the MSI. These arguments will be passed directly to the msiexec executable. If no custom arguments are needed, the parameter will have to be removed from the Arguments.

Parameter	Variable	Mandatory?	Description
-PrevProductName		No	Allow the user to specify the previous product name to be uninstalled prior to installing the MSI. By default, the system will use the product name in the MSI to install.

XCOPY Deployment Agent

This tool allows users to copy a package location to a specific destination. It is based on a custom release management batch script. The configuration variables of this tool are:

Parameter	Variable	Mandatory?	Description
	N/A	Yes	Fixed value indicating that all the source location must be copied
	Installation Path	Yes	The destination of the copy.

Command Line Runner

This tool allows users to run a specific executable. It is based on a custom release management PowerShell script. The configuration variables of this tool are:

Parameter	Variable	Mandatory?	Description
-command	N/A	Yes	Fixed value ./RunCommandLine.ps1.
-FilePath	FilePath	Yes	The complete path (if required) and file name of the executable to be run.
-Arguments	Arguments	No	The arguments needed when the executable is run.
-UserDomain	UserDomain	No	The domain of the user to use if the executable has to be run under a specific identity. If not provided, user specified will be considered as local (i.e. .\localuser).
-UserName	UserName	No	The name of the user to use if the executable has to be run under a specific identity.
-UserPassword	UserPassword	No	The password of the user to use if the executable has to be run under a specific identity.

Windows Registry Manager

This tool allows users to manipulate registry information in Windows. It is based on a custom release management PowerShell script. The configuration variables of this tool are:

Parameter	Variable	Mandatory?	Description
-command	N/A	Yes	Fixed value <code>./ManageWindowsRegistry.ps1</code> .
-Action	Action	Yes	The action to be performed. The allowed values are: <ul style="list-style-type: none">- CreateKey : Create a new registry key- ModifyKey : Modify a registry key or its data- DeleteKey : Delete a registry key with all its values and sub keys- CreateValue : Create a new registry value located under a key- ModifyValue : Modify a registry value located under a key- DeleteValue : Delete a registry value located under a key
-OperatingSystemType		No	Indicates if the key / value is accessed in the 32-bit or 64 bit registry- If not provided, the default is 32-bit.
-RegistryKey	RegistryKey	Yes	Path and name of the registry key (i.e. <code>HKLM\Software\MyKey</code>).
-RegistryValue		No*	Registry value to which changes will be applied (i.e. <code>MyValue</code>). Required for <code>CreateValue</code> , <code>ModifyValue</code> and <code>DeleteValue</code> .

Parameter	Variable	Mandatory?	Description
-ValueType		No	<p>Type of the value being affected. The accepted values are:</p> <ul style="list-style-type: none"> - REG_SZ : String (default value if not specified) - REG_QWORD_LITTLE_ENDIAN : A 64-bit number in little-endian format - REG_QWORD : A 64-bit number - REG_NONE : No value - REG_MULTI_SZ : Multi-line string (separated by \0) - REG_LINK : Symbolic Link - REG_EXPAND_SZ : String with unexpanded references to environment variables - REG_DWORD_BIG_ENDIAN : A 32-bit number in big-endian format - REG_DWORD_LITTLE_ENDIAN : A 32-bit number in little-endian format - REG_DWORD : A 32-bit number - REG_BINARY : Binary data in any form
-RegistryData		No	The data to apply to either the value (if using CreateValue or ModifyValue) or to the default value of the key (if using CreateKey or ModifyKey).
-RegistryKeyDestination		No*	<p>The registry key name to use when using the ModifyKey action to rename a registry key. For example:</p> <ul style="list-style-type: none"> - RegistryKey = HKLM\Software\MyKey - RegistryKeyDestination = HKLM\Software\MyModifiedKey <p>Required for ModifyKey.</p>

Windows Common IO

This tool allows users to perform common IO under Windows. It is based on a custom release management PowerShell script. The configuration variables of this tool are:

Parameter	Variable	Mandatory?	Description
-command	N/A	Yes	Fixed value ./ManageWindowsIO.ps1.
-Action	Action	Yes	Indicate the action to be performed: <ul style="list-style-type: none">- Create : Create a new folder- Delete : Delete file(s) or folder- Rename : Rename a file or folder- Move : Move a file or folder- Attrib : Modify the attributes (including ownership) of file(s) or folder
-FileFolderName	FileFolderName	Yes	The file or folder path and name to affect.
-DestinationName		No*	The destination name (for rename action) or the folder path and name (for move action). Required for Rename and Move actions.
-ReadOnly		No	Allow the user to set or reset the file(s) or folder read-only attribute. <ul style="list-style-type: none">- + : Set the flag- - : Reset the flag- Any other value : Let the flag as it is
-Archive		No	Allow the user to set or reset the file(s) or folder archive attribute. <ul style="list-style-type: none">- + : Set the flag- - : Reset the flag- Any other value : Let the flag as it is
-System		No	Allow the user to set or reset the file(s) or folder system attribute. <ul style="list-style-type: none">- + : Set the flag- - : Reset the flag- Any other value : Let the flag as it is
-Hidden		No	Allow the user to set or reset the file(s) or folder hidden attribute. <ul style="list-style-type: none">- + : Set the flag- - : Reset the flag- Any other value : Let the flag as it is

Parameter	Variable	Mandatory?	Description
-OwnerDomain		No	Represent the domain of the new owner to affect to the file or folder.
-OwnerName		No	Represent the name of the new owner to affect to the file or folder.

Windows Services Manager

This tool allows users to manipulate Windows services. It is based on a custom release management PowerShell script. The configuration variables of this tool are:

Parameter	Variable	Mandatory?	Description
-command	N/A	Yes	Fixed value ./ManageWindowsServices.ps1.
-Action	Action	Yes	Indicate the action to be performed: <ul style="list-style-type: none"> - Create : Create a new service - Config : Configure an existing service - Delete : Delete an existing service - Start : Start a service - Stop : Stop a service - Restart : Stop and start the service
-ServiceName	ServiceName	Yes	Name of the service to affect.
-ServerName		No	The name of the server where to perform the action. If not specified, the current server will be used.
-BinPath		No*	Binary path and name of the service. Required for Create action, optional for Config action and ignored by all others.
-DisplayName		No	Display name that will be used for this service. Optional for Create and Config actions and ignored by all others.
-Description		No	Description that will be given to the service. Optional for Create and Config actions and ignored by all others.

Parameter	Variable	Mandatory?	Description
-UserName		No	User name that will be used for the credentials to start the service (must be in the form DOMAIN\USER). If provided, the parameter – Password must also be provided. If not provided, the LocalSystem account will be used by default. Optional for Create and Config actions and ignored by all others. To reset the credentials to the LocalSystem, enter LocalSystem as the user and "" as the password.
-Password		No	Password of the user name specified to start the service. Optional for Create and Config actions and ignored by all others.
-StartMode		No	Start mode of the service. By default, the service will be created with Automatic. The choices are: <ul style="list-style-type: none"> - Manual : Service is started only manually - Automatic : Service will be started automatically next time the computer is restarted - Disabled : Service is disabled Optional for Create and Config actions and ignored by all others.

DACPAC Database Deployment Agent

This tool allows users to deploy a DACPAC package. It is based on a SQL server command line utility. There are a lot of possible parameters for this executable but we will list only the one used for this tool. Please refer to the sqlpackage documentation for further information about this utility. The configuration variables of this tool are:

Parameter	Variable	Mandatory?	Description
/Action:		Yes	Fixed value Publish meaning we want to publish the package.
/SourceFile:	FileName	Yes	The complete path (if required) and file name of the DACPAC.
/TargetServerName:	ServerName	Yes	The name of the SQL Server where to publish.

Parameter	Variable	Mandatory?	Description
/TargetDatabaseName:	DatabaseName	Yes	The name of the SQL Server database where to publish.

IIS Deployment Agent

This tool allows users to manipulate IIS objects and settings. It is based on a custom release management executable. The configuration variables of this tool vary depending of the type of actions to perform.

For application pool actions:

Parameter	Variable	Mandatory?	Description
-Action		Yes	Indicate the action to be performed: <ul style="list-style-type: none"> - CreateApplicationPool - ConfigureApplicationPool - DropApplicationPool - StartApplicationPool - StopApplicationPool - RecycleApplicationPool
-ap		Yes	Application Pool name.
-apAllow32Bit		No	Flag that indicate if the application pool must allow 32 bits application. This parameter is allowed only for IIS 7.0, 7.5, and 8.0, 8.5. The allowed values are: <ul style="list-style-type: none"> - 1 or True: 32 bits application are allowed - 0 or False: 32 bits application are not allowed
-apIdentUserDomain		No	Domain of the user to use as the identity of the application pool. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5. If not defined and the "IdentityUserName" is defined, the user will be considered as a local user (i.e. .\localuser).
-apIdentUserName		No	User name to use as the identity of the application pool. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5. If not defined, "IdentityUserDomain" and "IdentityUserPassword" will be ignored and the default value of IIS will be used (i.e. ApplicationPoolIdentity).

Parameter	Variable	Mandatory?	Description
-apIdentUserPassword		No	The password of the user to use as the identity of the application pool. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5.
-apNetVers		No	The .NET Framework version to use for the application pool. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5. The allowed values are: v1.0, v1.1, v2.0, v4.0
-apPipelineMode		No	The managed pipeline mode to use in the application pool. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5. The allowed values are: <ul style="list-style-type: none"> - Classic - Integrated
-apProcessIdleTimeout		No	The number of minutes a process can be idle in the application pool. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5.
-apRecycleKbMemory		No	The maximum number of Kb of memory that be used before the application pool is recycled. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5.
-apRecycleMinutes		No	A fixed number of minutes after which the application pool is recycled. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5.
apRecycleSpecificTime		No	A fixed time at which the application pool is recycled. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5. The required format is HH:MM (i.e. 23:58 or 01:23).
-apStartMode		No	The start mode to be used for the application pool. This parameter is allowed only for IIS 8.0 and 8.5. The allowed values are: <ul style="list-style-type: none"> - OnDemand - AlwaysRunning

Parameter	Variable	Mandatory?	Description
-AutoStart		No	Flag that indicate if the application pool must be started automatically. This parameter is allowed only for IIS 7.0, 7.5, 8.0 and 8.5. The allowed values are: <ul style="list-style-type: none"> - 1 or True: The application pool will be started automatically when Windows start. - 0 or False: The application pool will not be started automatically when Windows start.
-log		No	

For Web Site actions:

Parameter	Variable	Mandatory?	Description
-Action		Yes	Indicate the action to be performed: <ul style="list-style-type: none"> - CreateWebSite - ConfigureWebSite - DropWebSite - StartWebSite - StopWebSite - RestartWebSite
-sn		Yes	Web site name.
-port		No*	The port number of the web site. Mandatory for create action and optional in configure.
-pd		No*	The full path routing to the location where the application was published. Mandatory for create action and optional in configure.
-ap		No	Name of the application pool. If not defined when creating, the default application pool will be used. Optional in Create and Configure actions.
-EnablePreload		No	Allowed on IIS 7.0, 7.5, 8.0 and 8.5. Flag that indicates if the web site must be preloaded. Optional in Create and Configure actions. Allowed values are: <ul style="list-style-type: none"> - 1 or True: Preload enabled - 0 or false: Preload disabled

Parameter	Variable	Mandatory?	Description
-AutoStart		No	Allowed on IIS 7.0, 7.5, 8.0 and 8.5. Flag that indicates if the web site must be automatically started. Optional in Create and Configure actions. Allowed values are: <ul style="list-style-type: none"> - 1 or True: Start Automatically - 0 or false: Do not Start Automatically
-log		No	A log file with this given name will be generated in the Physical Directory.

For Web Application actions:

Parameter	Variable	Mandatory?	Description
-Action		Yes	Indicate the action to be performed: <ul style="list-style-type: none"> - CreateWebApplication - ConfigureWebApplication - DropWebApplication
-ws		Yes	Name of the application to create.
-pd		No*	The full path routing to the location where the application was published. Mandatory for create action and optional in configure.
-sn		No	Web site name. If not specified, the system will use Default web site when creating a new application. Optional in Create and Configure actions.
-ap		No	Name of the application pool. If not defined when creating, the default application pool will be used. Optional in Create and Configure actions.
-EnablePreload		No	Allowed on IIS 7.0, 7.5, 8.0 and 8.5. Flag that indicates if the web site must be preloaded. Optional in Create and Configure actions. Allowed values are: <ul style="list-style-type: none"> - 1 or True: Preload enabled - 0 or false: Preload disabled
-log		No	A log file with this given name will be generated in the Physical Directory.

For Virtual Directory actions:

Parameter	Variable	Mandatory?	Description
-Action		Yes	Indicate the action to be performed: <ul style="list-style-type: none"> - CreateVirtualDirectory - ConfigureVirtualDirectory - DropVirtualDirectory
-vd		Yes	Name of the virtual directory.
-pd		No*	The full path of the virtual directory. Mandatory for create action and optional in configure.
-sn		No	Web site name. If not specified, the system will use Default web site when creating a new virtual directory. Optional in Create and Configure actions.
-ws		No	Web application name. If not specified, the system will use the root web application when creating a new virtual directory. Optional in Create and Configure actions.
-log		No	A log file with this given name will be generated in the Physical Directory.

Windows Azure VM Manager

This tool allows users to manage a VM that resides in Windows Azure. It is based on a custom release management PowerShell script. The configuration variables of this tool are:

Parameter	Variable	Mandatory?	Description
-command	N/A	Yes	Fixed value ./ControlAzureVM.ps1.
-Command		Yes	Indicate the action to be performed: <ul style="list-style-type: none"> - Start : Start the VM - Stop : Stop the VM
-ServiceName	ServiceName	Yes	The name of the Windows Azure service into which the VM exists.
-Name	Name	Yes	The name of the Windows Azure VM.

XBAP Deployer

This tool allows users to deploy XBAP application. It is based on a custom release management executable. The configuration variables of this tool are:

Parameter	Variable	Mandatory?	Description
-pn	ProjectName	Yes	The XBAP project name to be deployed.
-d	Installation Path	Yes	The path where to install the XBAP application.
-ml		No	The complete path to the mage utility executable. If not defined, the tool will use the one provided as a resource in the tool.
-pl		No	The complete path where the XBAP project to deploy is located. If not specified, the tool will use the package location where it is executed.
-url		No	If specified, the system will replace the URL of the Setup.exe file to the one specified.
-cf		No	If specified, this will be used as the certification name when updating the manifest.
-pwd		No	If specified, this will be used as the certification password when updating the manifest.
-log		No	A log file with this given name will be generated in the Physical Directory.

Microsoft Test Manager (MTM) Command-Line Utility

This tool allows users to create and launch automated tests run on MTM (Microsoft Test Manager). It is based on a custom release management PowerShell script. The configuration variables of this tool are:

Parameter	Variable	Mandatory?	Description
-command	N/A	Yes	Fixed value ./TcmExec.ps1.
-Title	TestRunTitle	Yes	The name that will be used when creating the test run.
-PlanId	PlanId	Yes	The identifier of the Teat Plan under which the tests must run (Unique ID as defined in Test Manager).

Parameter	Variable	Mandatory?	Description
-SuiteId	SuiteId	Yes	The identifier of the Test Plan under which the tests must run (Unique ID as defined in Test Manager).
-ConfigId	ConfigId	Yes	The identifier of the Test Configuration under which the tests must run (Unique ID as defined in Test Manager).
-Collection	TfsCollection	Yes	The Team Foundation Server Collection URL for which the Automated Tests will execute.
-TeamProject	TeamProject	Yes	The name of the Team Project in which the Automated Tests were configured.
-TestEnvironment	TestEnvironment	Yes	The Test Environment in which the Tests are to be executed (the test environment is associated to a corresponding test controller).
-BuildDirectory	BuildDirectory	No*	The location of the Automated Tests. In MTM 2010, this parameter is required. In MTM 2012, 2013 this parameter is still supported, but users should preferably use BuildDefinition and BuildNumber parameters.
-BuildDefinition		No*	Allows the user to specify the build definition in which the Automated Tests are included. This parameter can only be used with MTM 2012. If defined, the BuildNumber parameter must also be specified. If not specified, the field "build," visible in the details of a test run will not be affected by the test run.

Parameter	Variable	Mandatory?	Description
-BuildNumber		No*	Allows the user to specify the build number whose drop location will contain the Automated Tests. This parameter can only be used under MTM 2012. If defined, the BuildDefinition parameter must also be specified. If not specified, the field "build," visible in the details of a test run, will not be affected by the test run.
-SettingsName		No	Allows the user to specify the settings to use for the test run. If not specified, the default one will be used.
-TestRunWaitDelay		No	Allows the user to specify the delay, in seconds, between each call to the test controller. This is required to allow the test run to complete before processing the test results. If not specified, it will default to 10 seconds.
-InconclusiveFailsTests	N/A	No	When this flag is set, the existence of inconclusive tests will fail the deployment. By default, an inconclusive test will not fail the deployment.
RemoveIncludeParameter	N/A	No	When this flag is set, the /include parameter will not be appended to the test creation command. This means that any tests that have a status different than Active will not be included in the test run.

If the Automated Tests are included in the same build process as the application being deployed, it is possible to use Metadata as the variable value and it will be automatically replaced at release time. The following is the currently available Metadata that can be used with Configuration Variables.

Variable	Variable value Metadata
BuildDirectory	\$(PackageLocation)
BuildNumber	\$(BuildNumber)
BuildDefinition	\$(BuildDefinition)

TFSCollection	\$(TfsUrlWithCollection)
TeamProject	\$(TeamProject)

INI File Manager

This tool allows users to alter the content of an INI file and can create a new INI file if specified. It is based on a custom release management executable. The configuration variables of this tool are:

Parameter	Variable	Mandatory?	Description
-Action		Yes	Indicate the action to be performed: <ul style="list-style-type: none"> - ModifySection - RemoveSection - CreateKeyValue - ModifyKeyValue - RemoveKeyValue
-INIFile	INIFileName	Yes	The INI file name and path (if required)
-SectionName	SectionName	No	The section name to affect. If not specified, the root section will be considered (all keys that are at the beginning of the file before the first section is considered in the root).
-NewSectionName		No	The new section name. If not specified, the root section will be considered (all keys that are at the beginning of the file before the first section is considered in the root). Required for ModifySection action.
-KeyName		No	The key name to affect. Required for CreateKeyValue, ModifyKeyValue and RemoveKeyValue.
-KeyValue		No	The value to affect to the key. When using the ModifyKeyValue action and this parameter is not specified and the NewKeyName is specified, the system will use the actual value. If not specified and the NewKeyName is not specified, the value will be emptied.

Parameter	Variable	Mandatory?	Description
-NewKeyName		No	The new name to give to the key. Used only in the ModifyKeyValue action.
-CreateFileIfNotExists		No	Indicate to the tool that a file must be created if it does not exist.

Windows Process

This tool allows users to kill a Windows Process. The configuration variables of this tool are:

Parameter	Variable	Mandatory?	Description
-command	N/A	Yes	Fixed value ./WindowsProcess.ps1.
-ProcessName		Yes	The name of the Windows process to kill.
-UserName		No	The user account running the process. If not specified, the process will be killed for all users.
-IsKillProcessTree		No	Flag that indicates if the process tree must be killed. Allowed values are: <ul style="list-style-type: none"> - 1: Kill process tree - 0 or empty: Do not kill process tree