

Installation Guide

Release Management for Visual Studio 2013

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

1	OVERVIEW	4
1.1	INTRODUCTION	4
1.2	DEPLOYMENT OVERVIEW	4
1.2.1	<i>Release Management Client for Visual Studio 2013</i>	<i>4</i>
1.2.2	<i>Release Management Server for Team Foundation Server 2013</i>	<i>4</i>
1.2.3	<i>Microsoft Deployment Agent 2013.....</i>	<i>4</i>
1.2.4	<i>Deployment Agent Tools</i>	<i>4</i>
1.2.5	<i>Mechanism between Release Management Server and Deployment Agents</i>	<i>5</i>
1.2.6	<i>Additional Components</i>	<i>5</i>
2	SYSTEM PREREQUISITES	7
2.1	RELEASE MANAGEMENT SERVER FOR TEAM FOUNDATION SERVER 2013.....	7
2.2	RELEASE MANAGEMENT CLIENT FOR VISUAL STUDIO 2013	7
2.3	MICROSOFT DEPLOYMENT AGENT 2013	7
3	CONFIGURATION	9
3.1	ACCOUNTS REQUIRED	9
3.2	TFS CONNECTION SERVICE ACCOUNT	9
3.3	ADDITIONAL PREPARATION.....	10
4	INSTALLATION.....	11
4.1	RELEASE MANAGEMENT SERVER FOR TEAM FOUNDATION SERVER 2013.....	11
4.2	RELEASE MANAGEMENT CLIENT	14
4.3	DEPLOYMENT AGENT	15
5	UPGRADE FROM TRIAL TO FULL VERSION.....	17
6	INITIAL RELEASE MANAGEMENT CONFIGURATION	17
7	CONFIGURING TEAM FOUNDATION BUILD [OPTIONAL]	18

1 OVERVIEW

1.1 Introduction

This document contains information and guidance for installing release management components. It details the deployment scenarios supported for the different components. This guide assumes that the supported version of Team Foundation Server has already been installed and configured ready for use.

1.2 Deployment Overview

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

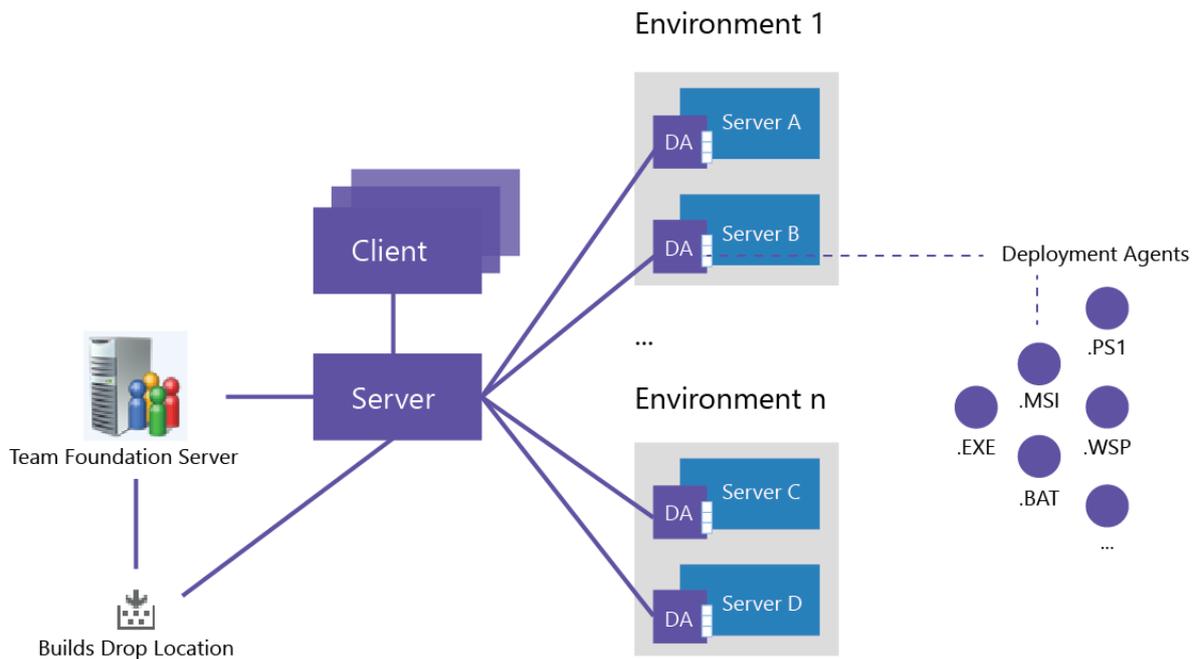


Figure 1 –Release Management components deployment overview

1.2.1 Release Management Client for Visual Studio 2013

The Release Management Client component is a Windows Presentation Foundation (WPF) application that serves as the main interface point to manage release management release information. This is the component that provides the user interface with which all users interact.

1.2.2 Release Management Server for Team Foundation Server 2013

The Release Management Server component is the heart of release management. It is a combination of Web and Windows Services that expose contracts used by all other RM components.

1.2.3 Microsoft Deployment Agent 2013

The Deployment Agent component is a service that needs to be installed on the Target Servers of the Environments where the application components will be hosted.

1.2.4 Deployment Agent Tools

The deployment agent Tools component assists 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
- Moving files to specific locations

Microsoft Deployment Agent extendable model

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

1.2.5 Mechanism between Release Management Server and Deployment Agents

The mechanism that allows the deployment agent to communicate with the Server is a one way pull mechanism from the deployment agents to the server.

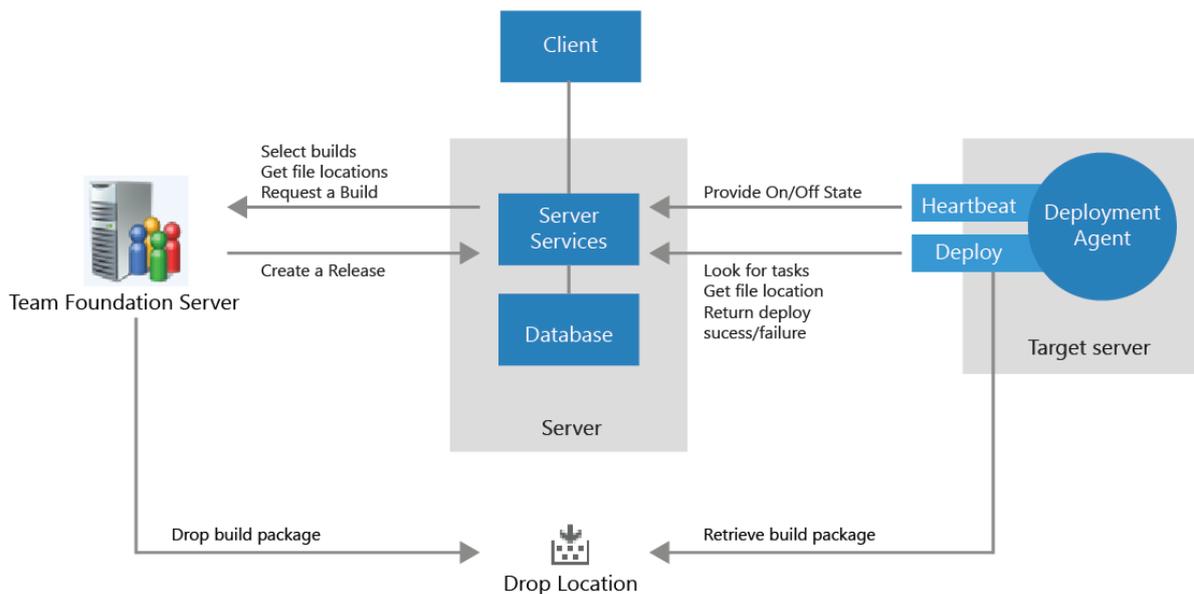


Figure 2 –Release Management components Communication Mechanisms

The deployment agent actually executes two threads on a pre-set and configurable timer. The heartbeat provides the Server with the state of the deployment agent in order for the Server to detect if the deployment agent is no longer available and act accordingly. The second thread monitors the Server for any deployment instruction to execute. If a deployment is queued, the deployment agent then requests the details to the server and will retrieve the build package from the provided drop location. It will then install the component and send back the deployment status to the server.

This mechanism allows Release Management components to operate without having to open a port on the Target Servers since the Release Management Server may not have access to the target environment; which is fairly common in environments closer to production.

1.2.6 Additional Components

In order for it to integrate with TFS, a few additional components are required:

1. The Release Management Client needs to be installed on the Team Foundation Build server. This is required in order for Team Foundation Build to be able to create releases

automatically as part of its execution. This is especially practical for continuous integration and deployment.

2. A custom Release Management build template provides a few operations that are needed to be included in the Build definition of an application that is deployed by Release Management. The build template can be added to the TFS server and used as is or can alternatively be manually merged with an existing build template if the standard template has been customized.

2 SYSTEM PREREQUISITES

2.1 Release Management Server for Team Foundation Server 2013

- **Processor:** 1 GHz Pentium processor or equivalent (Minimum); 2GHz Pentium processor or equivalent (Recommended).
- **RAM:** 1024 MB (Minimum); 2048 MB (Recommended).
- **Hard Disk:** On a clean machine, up to 2.2 GB of available space may be required. This is due to the dependency on .NET. Database can grow up to 1 GB per year (could be as low as 10 MB - depending on usage)
- **Display:** 1024 x 768 high color, 16-bit colors (Minimum); 1280 x 1024 high color, 32-bit (Recommended).
- **Supported platform:**
 - Windows Server 2008 R2 SP1
 - Windows Server 2012
 - Windows Server 2012 R2
- **Database requirements:** Microsoft SQL Server 2008 or 2008 R2 or 2012.

2.2 Release Management Client for Visual Studio 2013

- **Processor:** 1 GHz Pentium processor or equivalent (Minimum); 2GHz Pentium processor or equivalent (Recommended).
- **RAM:** 512 MB (Minimum); 1024 MB (Recommended).
- **Hard Disk:** On a clean machine, up to 2.2 GB of available space may be required. This is due to the dependency on .NET.
- **Display:** 1024 x 768 high color, 16-bit colors (Minimum); 1280 x 1024 high color, 32-bit (Recommended).
- **Supported platforms:**
 - Windows 7 SP1
 - Windows 8
 - Windows 8.1
 - Windows Server 2008 R2 SP1
 - Windows Server 2012
 - Windows Server 2012 R2

2.3 Microsoft Deployment Agent 2013

- **Processor:** 400 MHz Pentium processor or equivalent (Minimum); 1GHz Pentium processor or equivalent (Recommended).
- **RAM:** 256 MB (Minimum); 512 MB (Recommended).
- **Hard Disk:** On a clean machine, up to 2.2 GB of available space may be required. This is due to the dependency on .NET. More space might be required depending on the size of the supporting files for the components to be deployed (application size).
- **Supported platforms:**
 - Vista (latest service pack, PowerShell 2.0)
 - Windows 7 SP1
 - Windows 8

- Windows 8.1
- Windows Server 2008 R2 SP1
- Windows Server 2012
- Windows Server 2012 R2
- **Display:** 1024 x 768 high color, 16-bit colors (Minimum); 1280 x 1024 high color, 32-bit (Recommended).

3 CONFIGURATION

The installer provides automation of all configurations for the majority of systems/environments. However, in some cases, more configurations need to be done manually. This section describes those.

3.1 Accounts Required

Account	Role	Requirement
Release Management Server installation user	Identity under which the Server is installed	Local administrator rights sysadmin rights on the SQL Server
Release Management Client installation user	Identity under which the Client is installed on the end-user computer	Local administrator rights
Deployment Agent installation user	Identity under which the deployment agent is installed on the target Server	Local administrator rights
Server application pool identity	Identity under which the Server services run	Needs to be the TFS Service account (often named TFSService) Login rights to SQL Server and db_owner of release management database Service User of Release Management Write Access to the install folder (to enable logging)
Deployment agent's Windows Service Identity	Identity under which the deployment agent service runs	Local administrator rights Read Access to TFS drop location Service User of Release Management
Monitor's Windows Service Identity	Identity under which the Release Management Monitor service runs	Needs to be the TFS Service account (often named TFSService) Service User of Release Management
Release Management Client user	End-user that creates releases, configure components and/or TFS connections	Domain User Access to TFS (with appropriate CAL) Write Access to Release Management install folder (to enable logging)
Release Management Web Client user (workflow-only user)	End-user of Release Management that does not create releases, configure components or TFS connections	Domain User
Release Management User in TFS	Identity configured in Release Management to access TFS	This user needs to be granted the "Make requests on behalf of others" permission at collection level within TFS.

3.2 TFS Connection Service Account

When creating a TFS Connection in Release Management, a release management service account needs to be specified. This account is used to perform TFS related task. Because this account has collection level rights in TFS and normal users of Release Management should not be displayed Team Project they don't normally have access to, Release Management does requests to TFS on

behalf of the currently logged in user. For that reason, the account used as a TFS Connection credential needs to have the permission **"Make requests on behalf of others"** set to enabled at collection level.

The screenshot shows the TFS Control Panel interface. At the top, there is a navigation bar with 'Control Panel > QaCollection' and tabs for 'overview' and 'security'. Below this, there are tabs for 'groups' and 'users', and a search box. A list of users is shown, with 'TFS Service' selected. On the right, the 'permissions' tab is active for 'TFS Service'. A table lists various permissions and their status:

Permission	Status
Administer build resource permissions	inherited allow
Administer Project Server integration	inherited allow
Administer shelved changes	inherited allow
Administer workspaces	inherited allow
Alter trace settings	not set
Create a workspace	inherited allow
Create new projects	not set
Delete team project	not set
Delete team project collection	inherited allow
Edit collection-level information	inherited allow
Make requests on behalf of others	allow
Manage build resources	inherited allow
Manage process template	not set
Manage test controllers	not set

3.3 Additional Preparation

The following elements need to be known prior to installation of the Release Management Server for Team Foundation Server 2013:

1. SMTP Server (to send out email) – Server name, port, username/password (if needed, it can be left blank on most configurations).
2. Web Services port for the Release Management Server. A port needs to be opened for inbound communication from the Deployment Agents, Release Management Client and Web client.

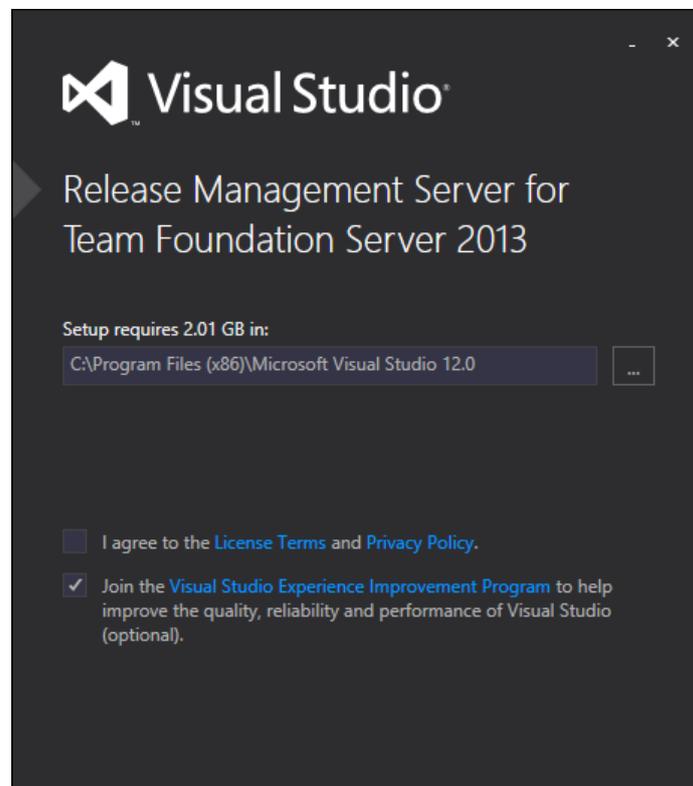
4 INSTALLATION

4.1 Release Management Server for Team Foundation Server 2013

Server component installation

During the installation of the Release Management Server, the user doing the install needs to be a **local administrator** on the machine and have **sysadmin** rights on the SQL Server.

1. Execute the provided setup file "**rm_Server.exe**". This will launch the setup of **Release Management Server for TFS 2013**. Select the destination folder.
 - a. Review and accept License terms, Privacy Policy.



2. Click Install to start the installation process. It will take a few minutes.
3. Restart your machine if prompted.
4. Configure the Release Management server.
 - a. **Account:** Specify the credential to run the Release Management Server. The default option is to run this as NetworkService. Optionally you can specify a specific user who should be part of the local Administrators group

This will be the account with which Release Management Services will be running with. This includes the release management Monitoring service and the application pools.

- b. **Web Service:** Default port is 1000. The web service URI is <http://<Computer Name:Port>/ReleaseManagement>

The Web app will be hosted on the port mentioned above.

- 5. **Database:** Specify a database server instance to use for release management. The release management Server can run on any version of SQL Server 2008, 2008 R2 and 2012
- 6. You can check if the connection works by connecting to the database server as the Interactive User. Service identity account will be used as the database owner and used to connect from the release management Server to the Database.

The Database Test link validates that the interactive user has the necessary rights to complete the database installation.

Configure Release Management Server

Identity for Release Management services

Network service

Account:

Password: [Test](#)

Release Management Server settings

HTTP HTTPS

Web service port:

URL:

Database server: [Test](#)

Example: MyDBServer\SQLEXPRESS

Database: ReleaseManagement

You can download SQL Server Express from [here](#).

[About](#) [Apply settings](#) [Close](#)

- 7. Configuration that happens behind the scene when the user chooses **Apply Settings**.
 - a. Account – A release management service identity is created as a valid release management service user in the database.

b. Database –

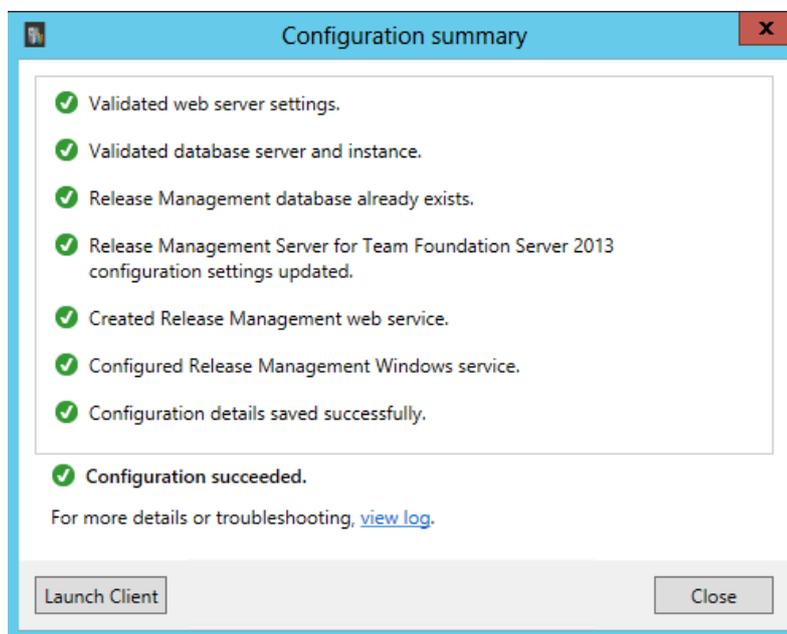
- Database named “ReleaseManagement” is created.
- release management service identity is created as db_owner.
- Interactive user is created as a valid release management super user.

c. Web Server

- If needed, Internet Information Services IIS 8.0 or IIS 7.5 is installed and configured as per the Operating System of the Release Management Server. If not started, the Windows service for IIS [World Wide Web Publishing Service] is configured
- Release Management application pool is created.
- Release Management web service is created.
- Release Management service identity is set as the application pool identity.

d. Windows Service

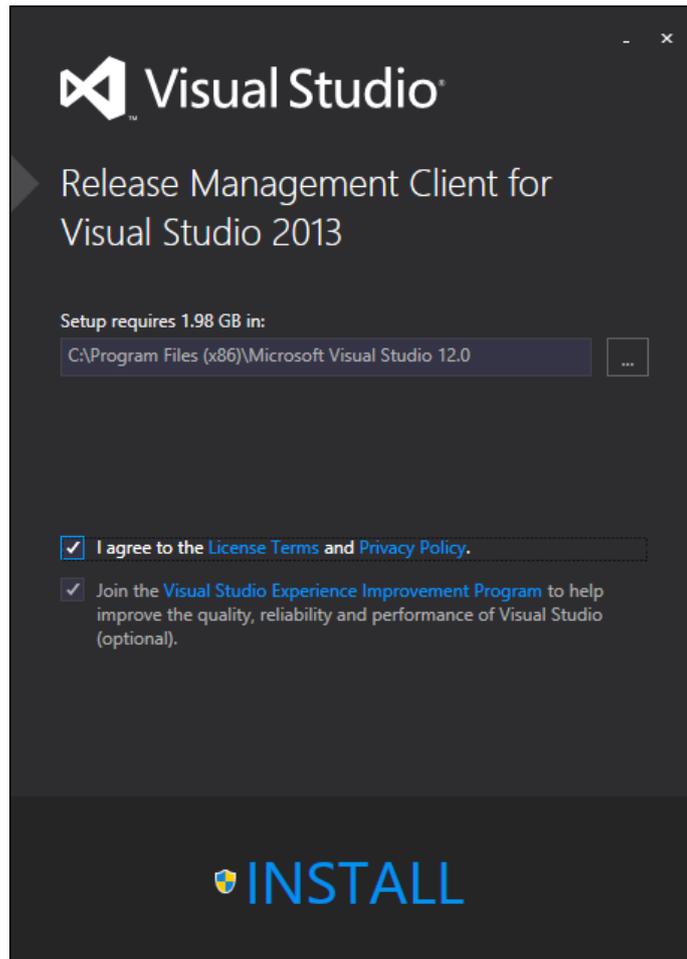
- The Monitoring service is started, and the Release management service identity is set as the Monitoring Service RunAs account.



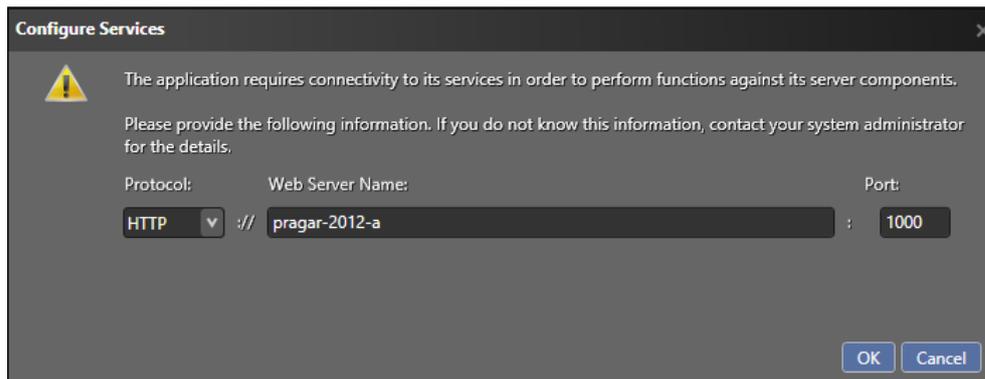
e.

4.2 Release Management Client

1. Execute the provided setup file “rm_Client.exe”. This will launch the setup of Release Management Client for Visual Studio 2013.
 - a. Select the destination folder.
 - b. Review and accept License terms, Privacy Policy.
 - c. Click Install to start the installation process. It will take a few minutes.

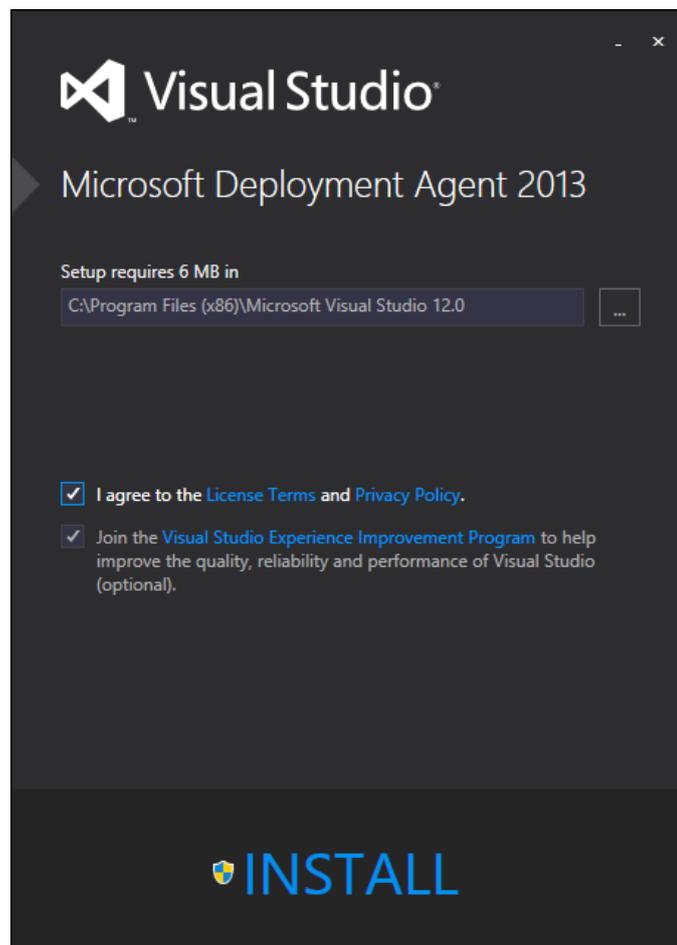


2. After the installation of the client, launch the Release Management Client from the programs folder.
3. Configure the client to connect to the Release Management Server.



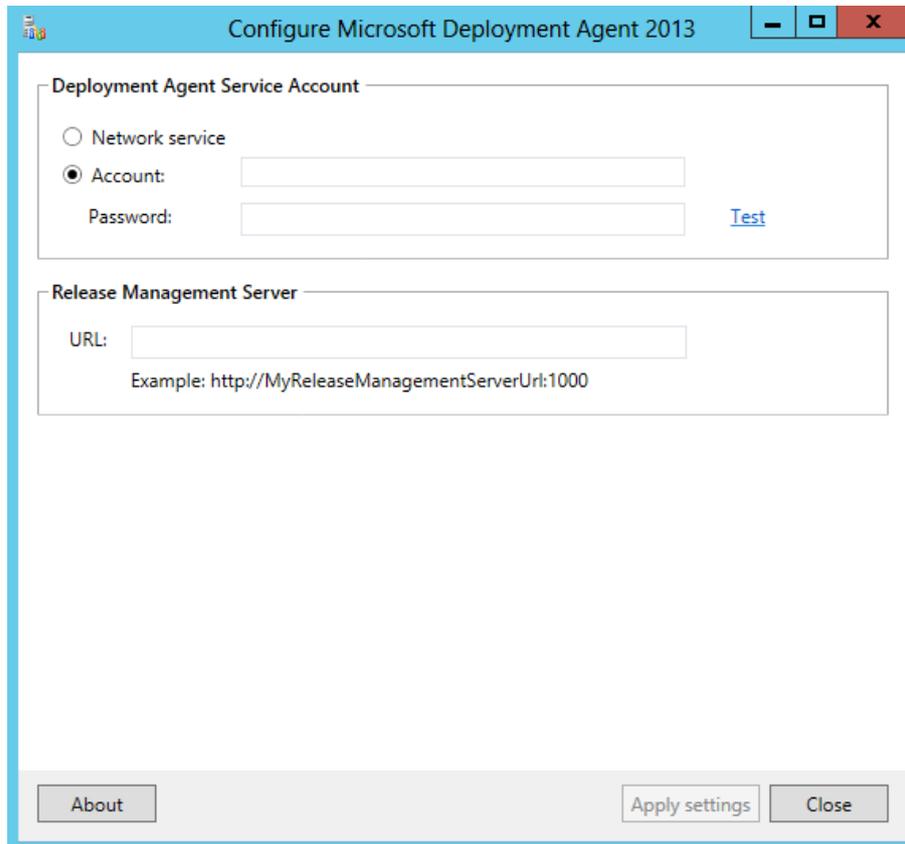
4.3 Deployment Agent

1. Execute the provided setup file “rm_Deployment*.exe”. This will launch the setup of Microsoft Deployment Agent 2013.
2. Review and accept License terms, Privacy Policy.
3. Click Install to start the installation process. It will take a few minutes.



4. Configure the Deployment Agent.

- a. Configure the Deployment Agent service by specifying an identity. The default here is to run as NetworkService. Optionally, a specific user credential can be provided, which is expected to have required permissions to do application deployments locally.
- b. Specify the URL of the Release Management Server.



5. Requirements:

- a. Interactive user should be local admin.
- b. Interactive user should have the privilege to create a user in the Release Management Server
- c. Release Management Server URL should be valid and accessible by the interactive user.

The following tasks are performed:

- a. If not already existing, the Deployment Agent Service account user is created in the Release Management Server.
- b. Configuration data is saved to the Release Management Server database.

5 UPGRADE FROM TRIAL TO FULL VERSION

Release Management for Visual Studio 2013 has three components (server, client and deployment agent) and you should strive to use the same version of all these components throughout your entire deployment. If you have the trial version installed and you upgrade, you should upgrade all components before you start to manage any releases

Note:

Trail version of Release Management components for Visual Studio 2013 have to be uninstalled prior upgrading to full version

6 INITIAL RELEASE MANAGEMENT CONFIGURATION

At this stage, the Release Management Server for Team Foundation Server 2013 and Release Management Client for Visual Studio 2013 should be up and running. In order to have the Release Management Server communicating with the Deployment Agent, additional configuration steps are required to be performed in Release Management:

In the Administration section:

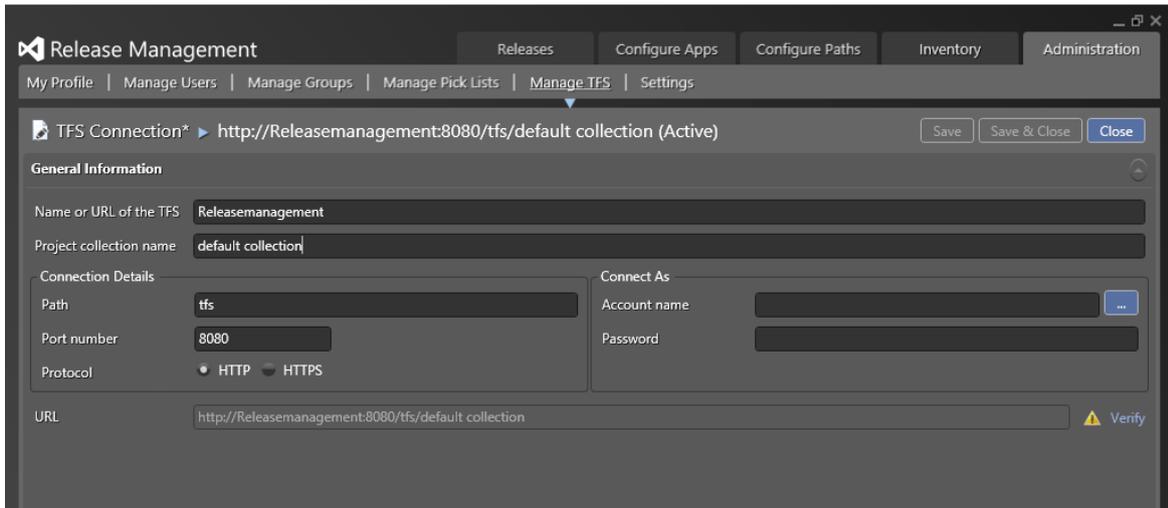
Configuring the Deployment Agent:

1. For each identity under which the deployment agents run, a corresponding user needs to be added in Release Management. This user needs to be set up as a 'Service User'.
2. Create the Servers on which Deployment Agents have been installed. The **Scan for New** feature can also be used to expedite the creation of Servers.

Refer to the user guide for information about how to use the Deployment Agents for your release.

Configuring Team Foundation Server:

Add a TFS Connection in order to establish the connection with TFS and be able to select builds to release. For each collection that the components belong to, there has to be a dedicated connection setup with TFS at that collection level. The user has the option to specify user credentials to use while connecting to the Team project collection.



7 CONFIGURING TEAM FOUNDATION BUILD [OPTIONAL]

The last element to configure is to add the Release Management Team Build template. Add ReleaseDefaultTemplate.xaml (for TFS 2010) - or ReleaseDefaultTemplate.11.1.xaml (for TFS 2012 & TFS 2013) to your TFS following the standard TFS procedure.