



RepliWeb TFS Plug-in Guide

Software Version: 1.2

January 10, 2012

January 10, 2012
Copyright © 2000-2011 by RepliWeb, Inc.

The information in this document has been compiled with care, but RepliWeb makes no warranties as to accurateness or completeness, as the software described herein may be changed or enhanced from time to time. This information does not constitute commitments or representations by RepliWeb, and is subject to change without notice. The software described in this document is furnished under license and may be used or copied only in accordance with the terms of this license. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written consent of RepliWeb Ltd. Any trademarks, trade names, service marks, or service names owned or registered by any other company and used in this document are proprietary to that company.

Please direct correspondence or inquiries to:

RepliWeb, Inc.
6441 Lyons Road
Coconut Creek
FL 33073

Phone: (954) 946-2274
Fax: (954) 337-6424
E-Mail: info@repliweb.com,
Support: <http://support.repliweb.com>
Web Site: <http://www.repliweb.com/>

Table of Contents

- Introduction 1**
 - R-1/ROSS Integration with TFS 2
 - RepliWeb TFS Plug-in High-Level Flow 3
- 1. Setting Up RepliWeb TFS Plug-in 4**
 - Installation Prerequisites..... 4
 - The Installation Procedure 4
 - Installing TFS Build Distributor 4
 - Installing TFS Events Subscription 7
 - Silent Installation of RepliWeb TFS Plug-in 10
 - The Uninstallation Procedure..... 11
 - Removing R-1/ROSS Items from Build Templates..... 12
 - Uninstalling TFS Build Distributor 14
 - Uninstalling TFS Events Subscription 14
 - Silent Uninstallation of RepliWeb TFS Plug-in..... 16
- 2. Getting Started 17**
- 3. Adding the R-1/ROSS Build Activities to a Build Template 18**
 - Adding the R-1/ROSS Build Activities to an Existing Build Template..... 18
 - Running the RepliWeb TFS Build Utility 24
 - Parameters 24
 - Adding the R-1/ROSS Build Activities to a New Build Template..... 25
- 4. Configuring R-1/ROSS Build Deployment 28**
 - R-1 Parameters..... 28
 - ROSS Parameters..... 30
 - Built-In Parameters 31
 - Basic 31
 - Advanced (ROSS Build Activity Only)..... 32
 - Example: Build Process Tab Configuration 32
- 5. Advanced Configuration Options..... 33**

Editing the R-1/ROSS Build Activity Metadata 33

Changing the Default Argument Values in the Build Template..... 35

6. Managing Build Deployments 36

 Build Process Deployment Jobs..... 36

 Initiating New Build Processes 36

 Defining Build Qualities 36

 Changing Build Qualities 37

 Build Process Deployment Rollback Jobs 39

 Defining Rollback Qualities 39

 Rolling Back Build Deployments..... 39

 Deployment Job Reports 39

7. TFS Build Deployment Variables 42

A. Scrambling Passwords 43

B. Changing Default Argument Names..... 44

Introduction

Team Foundation Server (commonly abbreviated to TFS) is a Microsoft product for Application Lifecycle Management offering source control, data collection, reporting, and project tracking, and is intended for collaborative software development projects.

RepliWeb's TFS Plug-in integrates with TFS, enabling post-build process deployment of compiled binaries (using R-1) and/or SharePoint 2010 solutions (using ROSS) from the development environment to the testing server(s). Deployment can be triggered by a specific build quality or independent of build quality.

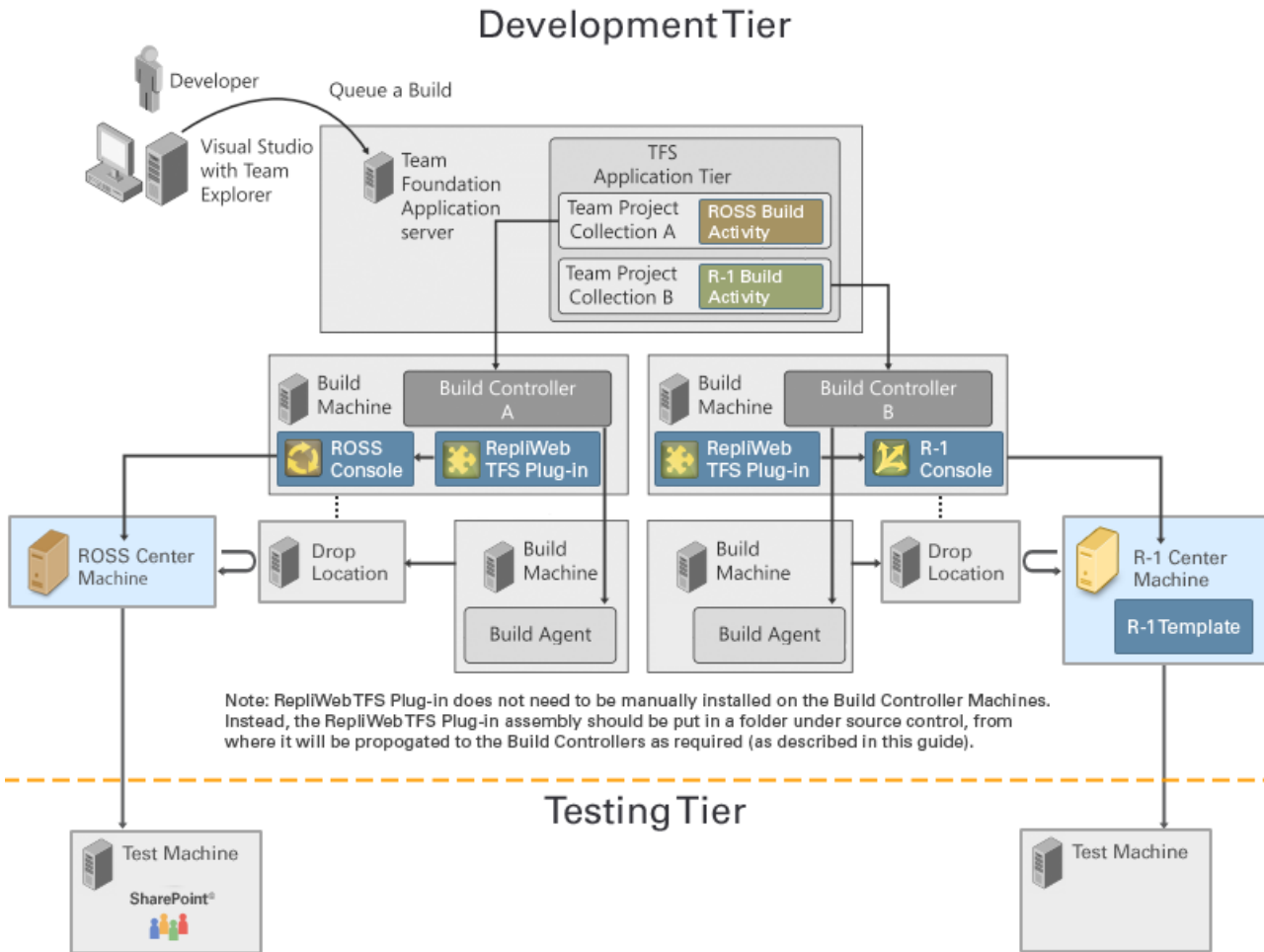
This guide explains how to install and configure RepliWeb's TFS Plug-in. Readers should possess a good working knowledge of R-1 and/or ROSS, depending on what types of files need to be deployed.

Related Documentation:

- [R-1 Installation Guide](#)
- [R-1 User Guide](#)
- [ROSS Installation Guide](#)
- [ROSS User Guide](#)

R-1/ROSS Integration with TFS

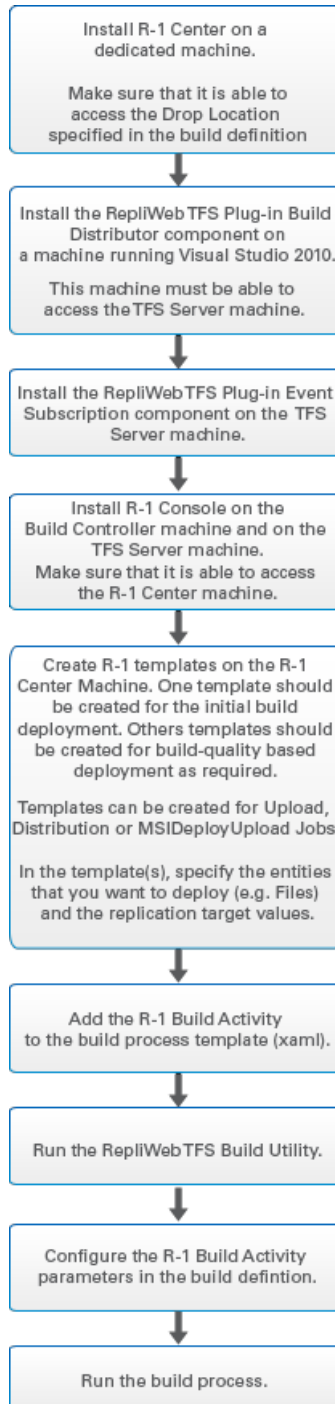
The diagram below shows how RepliWeb TFS Plug-in adds a layer of automation to Application Lifecycle Management by replicating compiled project files from the Development Tier to the Testing Tier.



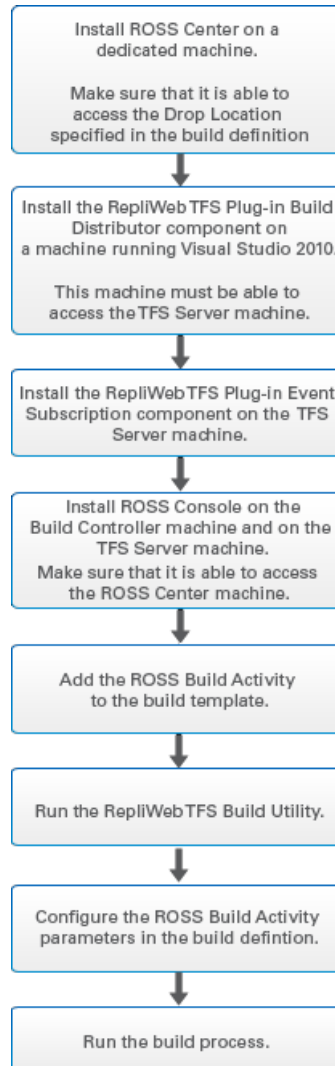
RepliWeb TFS Plug-in High-Level Flow

The workflows below provide a high-level overview of the steps required to integrate RepliWeb TFS Plug-in with TFS and configure the plug-in to deploy TFS builds at build process completion and according to the build quality selected.

Build Deployment using R-1



Build Deployment using ROSS



1. Setting Up RepliWeb TFS Plug-in

Installation Prerequisites

Before beginning the installation, make sure that the following requirements have been met:

- TFS 2010
- Microsoft Visual Studio 2010
- R-1/ROSS license that supports RepliWeb TFS Plug-in
- .Net 4 installed on all the TFS tiers.

The Installation Procedure

RepliWeb TFS Plug-in includes two components:

- **TFS Build Distributor (Required)** – Enables builds to be automatically distributed to pre-defined targets using R-1 and/or ROSS.
- **RepliWeb TFS Events Subscription** – Enables builds to be distributed to a pre-defined target based on the selected build quality (e.g. Rejected, Released, Under Investigation and so on). This feature requires the installation of the RepliWeb TFS Events Subscription Web Service, as explained in the following sections.

Note: The Web Service only needs to be installed if the Events Subscription component is installed.

Installing TFS Build Distributor

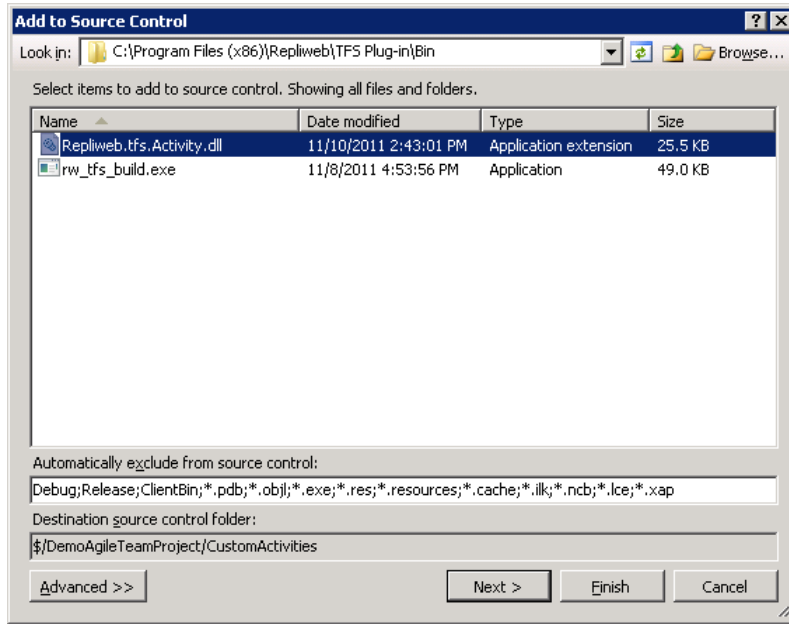
The procedure below describes how to install the **TFS Build Distributor** component. The **TFS Build Distributor** component must be installed on the Visual Studio 2010 computer.

To install TFS Build Distributor:

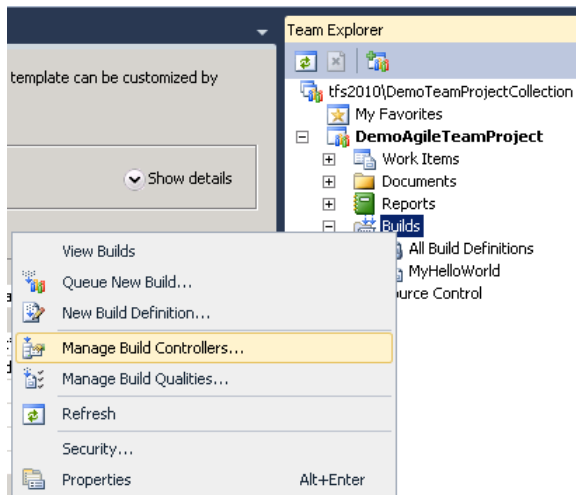
1. Install **TFS Build Distributor** on the Visual Studio 2010 (client) computer as follows:
 - a. Run the `rw_tfs_plugin_<version.build>.exe` setup file.
 - b. When the Welcome screen appears, click **Next**.
 - c. Accept the license terms and then click **Next**.
 - d. In the **Custom Setup** screen, make sure that **TFS Build Distributor** is selected. For information about installing the TFS Events Subscription component, see [TFS Events Subscription Installation](#).
 - e. Choose where you want to install **TFS Build Distributor**.
 - f. Continue clicking **Next** until the setup wizard completes.

- Restart Visual Studio 2010.
- On the client computer, open the Source Control Explorer and create a new folder called **CustomActivities**.
- Right-click the folder and select **Add Items to Folder**.

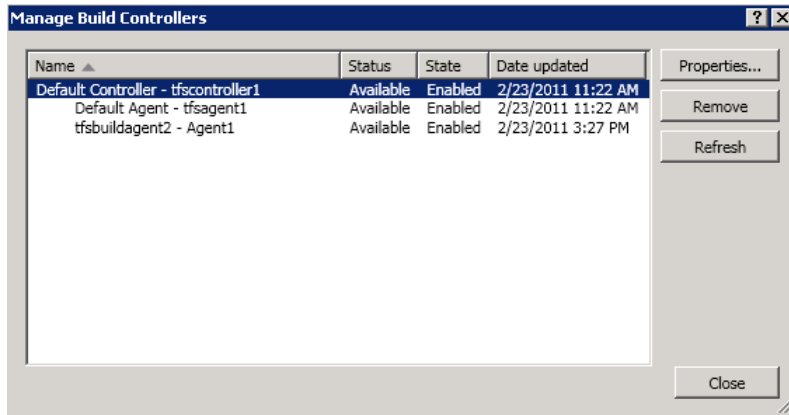
The **Add to Source Control** Window opens.



- Browse to the folder `~\RepliWeb\TFS Plug-in\Bin\` and select the file **Repliweb.tfs.Activity.dll**. Click **Finish**.
- Check in the **CustomActivities** folder.
- In the **Team Explorer** window, right-click **Builds** and then select **Manage Build Controllers**.

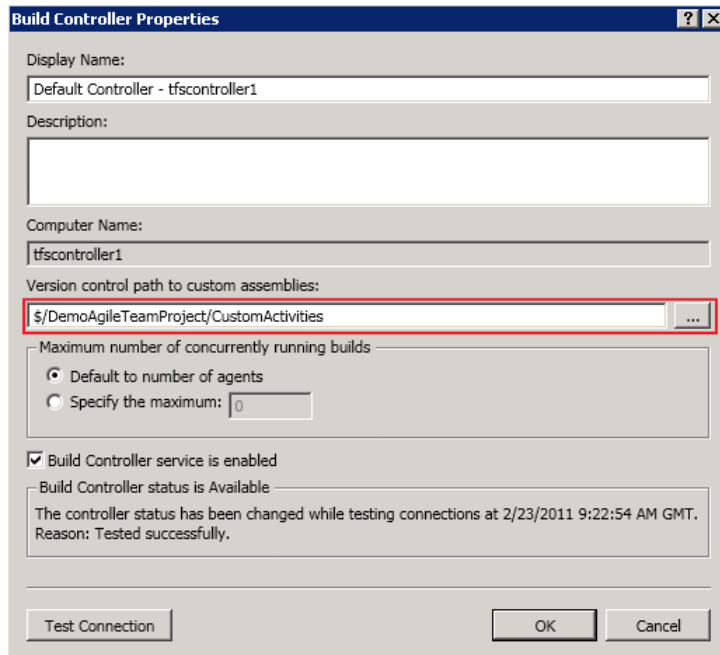


The **Manage Build Controllers** window opens.



8. Select the Build Controller and then click **Properties**.

The **Build Controller Properties** window opens.



9. In the **Version control path to custom assemblies** field, specify the path to the **CustomActivities** folder and then click **OK**.
10. Install R-1 Console and/or ROSS Console on the Build Controller computer(s) as required.

Note: Make sure that the Build Controller computer is able to access the R-1/ROSS Center computer(s).

For example, if the build process template will only include the R-1 Build Activity, then you only need to install R-1 Console on the Build Controller computer. However, if the build process template will include both the R-1 and ROSS Build Activities, then you need to install both R-1 Console and ROSS Console on the Build Controller computer.

11. Install R-1/ROSS Center on the dedicated deployment source computer (usually a drop location or SharePoint 2010 computer).

Note: Make sure that the R-1/ROSS Center computer is able to access the drop location (a UNC path).

12. Install the ROSS Edge (ROSS Build Activity) and/or the R-1 Edge (R-1 Build Activity) component on the deployment target computer(s).

Installing TFS Events Subscription

The procedure below describes how to install the **RepliWeb TFS Events Subscription Web Service**. The Web Service must be installed on the TFS Server computer.

Web Service installation consists of the following steps:

1. Install the **RepliWeb TFS Plug-in – TFS Events Subscription** component.
2. Create the Events Subscription web site.
3. Subscribe each team project collection to the Web Service.

TFS Events Subscription Component Installation

During component installation, files and commands required by the Web Service are copied to the computer.

Note: If an older version of **RepliWeb TFS Plug-in** is already installed on this computer, make sure you completely remove it before installing the new version.

To install the TFS Events Subscription component:

1. Install **TFS Events Subscription** on the TFS Server computer as follows:
 - a. Run the `rw_tfs_plugin_<version.build>.exe` setup file.
 - b. When the **Welcome** screen appears, click **Next**.
 - c. Accept the license terms and then click **Next**.
 - d. In the **Custom Setup** screen, select **TFS Events Subscription**.
 - e. Choose the installation directory for **TFS Events Subscription**.
 - f. Continue clicking **Next** until the setup wizard completes.
2. Install R-1 Console and/or the ROSS Console on the TFS Server computer(s), as required.

Note: Make sure that the TFS Server computer is able to access the R-1/ROSS Center computer(s).

For example, if only the R-1 build quality deployment feature is enabled in the build process template, then you only need to install R-1 Console on the TFS Server. However, if both the R-1 build quality deployment feature and the ROSS build quality deployment feature are enabled in the build process template, then you need to install both R-1 Console and ROSS Console on the TFS Server computer.

3. If not already installed, install R-1 Center/ROSS Center on a dedicated computer.

Note: Make sure that the R-1/ROSS Center computer is able to access the drop location (a UNC path).

4. Install the ROSS Edge (ROSS Build Activity) and/or the R-1 Edge (R-1 Build Activity) component on the target computer(s).
5. Next, create the Events Subscription web site.

Web Site Creation

This command creates the **Events Subscription** web site. This web site will host a web application for each team project collection and will provide the Web Service.

Note: Running this command requires administrative privileges on the TFS Server computer.

To install the Events Subscription web site:

1. Run a command prompt as administrator and change the working directory to:

```
~\RepliWeb\TFS Plug-In\Bin
```

Where “~” represents the installation path.

2. Create the Events Subscription web site by issuing the command below:

Syntax:

```
> rw_tfs_ws_config create_site -site=web_site_URL
```

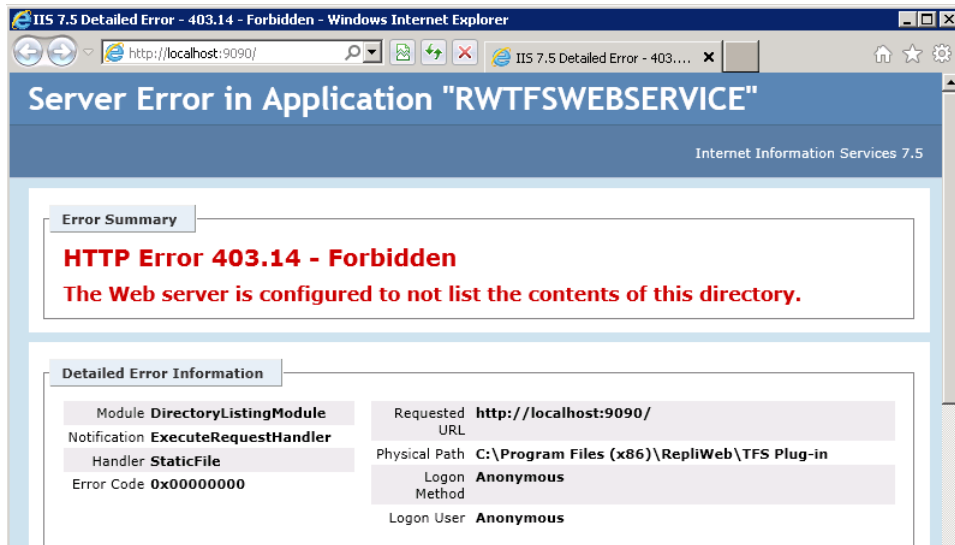
Example:

```
> rw_tfs_ws_config create_site -site=http://tfsserver:9090/RwTfsWebservice
```

3. Verify that the web site was properly created and is accessible:

- a. Open **Internet Information Services Manager**.
- b. Select the web site you created and click **Browse** in the Actions pane.

The following error should be displayed:



4. Next, subscribe one or more team project collections to the Web Service.

Parameters

| Parameter | Value |
|-----------|--|
| -site | (Relevant for <i>create_site</i> command only) The full URL of a web site that |

| | |
|--|---|
| | <p>does not already exist.</p> <p>Note: Paths with spaces must be enclosed in quotation marks. For example: <code>-site=http://tfsserver:9090/RwTfsWebservice</code></p> |
|--|---|

Web Service Subscription

This command creates the web application and application pool for the web service that will handle events from the specified team project collection. It will also subscribe BuildStatusChange events, which are raised from this team project collection to the Web Service.

Each team project collection can typically be accessed by different users and credentials. Therefore, a different web application and application pool are created for each collection, and the identity of the application pool is set to a valid user of this collection.

Once subscription completes, changing a build quality will raise an event that will invoke the Web Service to perform, in the relevant application pool, the specified deployment and update the build with the performed action. This requires the web service to have specific permissions, as explained in the `-ws_user` parameter's description in the Parameters table below.

Note: This command requires administrative privileges on the TFS Server computer.

To subscribe a team project collection to the Web Service:

1. Run a command prompt as administrator and change the working directory to:
`~\RepliWeb\TFS Plug-In\Bin`
2. Subscribe the Web Service by issuing the following command:

Syntax:

```
> rw tfs_ws_config subscribe -collection=url -ws_user=username [-ws_password=pass] [-notify_delay=n]
```

Example:

```
>rw tfs_ws_config subscribe -collection="http://tfsserver:8080/tfs/my tpc" -ws_user="comp_domain\admin" -notify_delay=60
```

3. Repeat this procedure for each team project collection that will use the Web Service.

Parameters

| Parameter | Value |
|--------------------------|---|
| <code>-collection</code> | <p>(Relevant for subscribe command) Team project collection URL.</p> <p>Note: Paths with spaces must be enclosed in quotation marks. For example: <code>-collection=" http://tfsserver:8080/tfs/my tpc"</code></p> |
| <code>-ws_user</code> | <p>(Relevant for subscribe command) The user name (or domain\username) of the web application created for this team project collection. This user will be set as the identity of</p> |

| | |
|----------------------------|--|
| | <p>the application pool.</p> <p>Note: <code>-ws_user</code> must meet the following requirements:</p> <ul style="list-style-type: none"> ▪ Belong to the Builders and/or Contributors groups of the relevant team projects of this team project collection. ▪ Have permission <i>Update Build Information</i> for the builds in these team projects. ▪ Have “Read”/“Write” permissions for the drop locations of these builds. ▪ It is also recommended to have “Read” permissions for the Event Log. |
| <code>-ws_password</code> | <p>(Relevant for subscribe command) The password of <code>-ws_user</code>.</p> <p>Note: If <code>-ws_password</code> is not specified, the command will prompt you to enter the password and will print in hidden mode (as asterisks).</p> |
| <code>-notify_delay</code> | <p>The maximum delay, in seconds, between the time when the user changes a build and when TFS invokes the Web Service.</p> <p>By default, team project collections have a delay period of 120 seconds. Use this argument if you want to shorten this delay.</p> <p>If <code>-notify_delay</code> is not specified, the delay will not be changed.</p> <p>Note: Long delays may reduce the TFS load when handling events. Therefore, in a large team project collection with many build quality change events, shortening the delay period may increase TFS load time for the team project collection.</p> |

Silent Installation of RepliWeb TFS Plug-in

RepliWeb TFS Plug-in can also be installed silently enabling you to quickly deploy it on multiple computers, should the need arise. Although the instructions below are for command line installation, the silent install can be performed using a command prompt or run as a script.

To silently install RepliWeb TFS Plug-in using qualifiers:

1. Run a command prompt as administrator and issue the following command:

```
>set RW_INSTALL_MODE=silent
```

2. Optionally, to change the default installation path (C:\Program Files (x86)\RepliWeb\TFS Plug-in), issue the following command:

Syntax:

```
>set RW_INSTALL_PARAMS=TARGETDIR=<installation dir>  
ADDLOCAL=TFS_Build_Distributor,TFS_Events_Subscription
```

Note: If no component is specified, **TFS_Build_Distributor** will be installed by default.

Example:

```
>set RW_INSTALL_PARAMS ADDLOCAL=TFS_Build_Distributor
```

3. Change the working directory to the directory containing the RepliWeb TFS Plug-in setup file and then issue the following command:

```
>rw_tfs_plugin_<version.build>.exe /s
```

Note: Please do not change the executable's filename.

To silently install RepliWeb TFS Plug-in using MSI:

1. Right-click the **rw_tfs_plugin_<version.build>.exe** file and extract the contents to a directory of your choice.

The directory should now contain the following files: **setup_launcher.exe** and **rw_tfs_plugin.msi**.

2. Run a command prompt administrator and change the working directory to the directory containing the **rw_tfs_plugin.msi** file.
3. To install without a log, issue the following command:

```
>msiexec /i rw_tfs_plugin.msi /qn
```

Note: Please do not change the executable's filename.

4. To install with a log, issue the following command:

Syntax:

```
>msiexec /i rw_tfs_plugin.msi /qn /l*v <log file path + name>  
ADDLOCAL=TFS_Build_Distributor,TFS_Events_Subscription
```

Note: If no component is specified, **TFS_Build_Distributor** will be installed by default.

Example:

```
>msiexec /i rw_tfs_plugin.msi /qn /l*v c:\rw_tfs_plugin.log  
ADDLOCAL=TFS_Build_Distributor
```

5. To install to a custom directory, add **TARGETDIR=<path>** to the command.

Example:

```
>msiexec /i rw_tfs_plugin.msi /qn TARGETDIR="C:\New Path\Install"
```

Note: Paths with spaces must be enclosed in quotation marks.

The Uninstallation Procedure

RepliWeb TFS Plug-in includes two components: **TFS Build Distributor (Required)** and **RepliWeb TFS Events Subscription**.

Removal of TFS Plug-in consists of the following steps:

1. Remove R-1/ROSS arguments and activities from TFS build templates.
2. Uninstall RepliWeb Plug-in.

TFS build Distributor – On the Architect computer, uninstall using the TFS Plug-in installer or silently.

TFS Events Subscription – On the TFS Server, remove the Web Service and then uninstall the TFS Plug-in, as described in the [Uninstalling TFS Events Subscription](#) section.

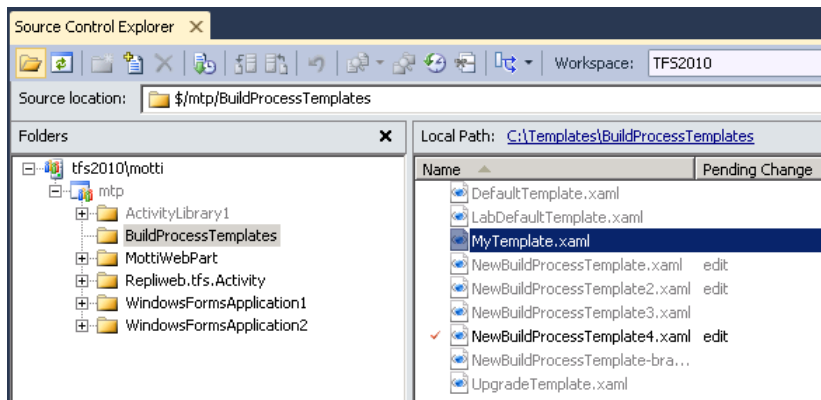
Removing R-1/ROSS Items from Build Templates

This procedure removes R-1/ROSS arguments, activities and additional RepliWeb items. If these items are not removed, they may cause unexpected behavior after TFS Plug-in is uninstalled.

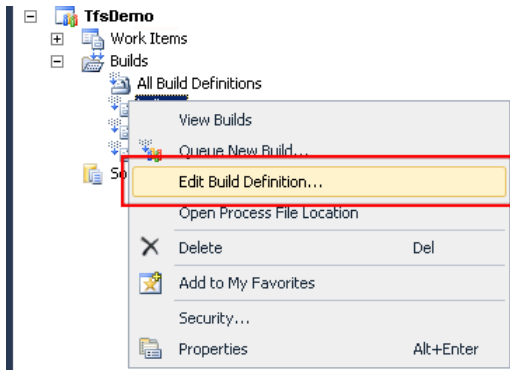
Note: This procedure must be performed for each build template where the R-1/ROSS Build Activity is installed.

To remove these items from the TFS :

1. On the Architect computer, open Visual Studio 2010.
2. Check out the template.
 - a. Open Source Control Explorer and locate the template from which you want to remove the RepliWeb Build Activity.
 - b. Right-click the template and select **Check Out for Edit**. In the **Check Out** window, click **Check Out**.

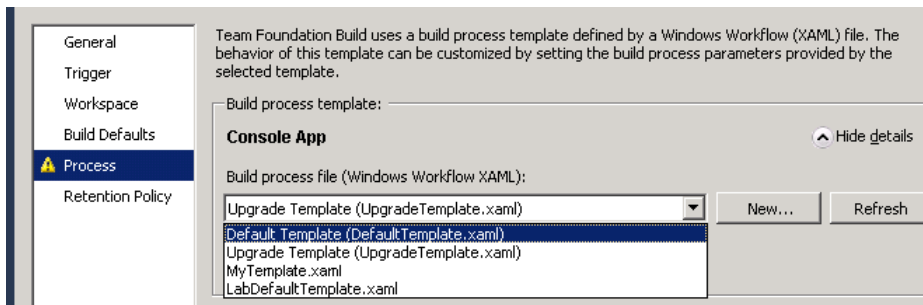


3. In the Team Explorer window, right-click the build to which the template is associated and then select **Edit Build Definition**.

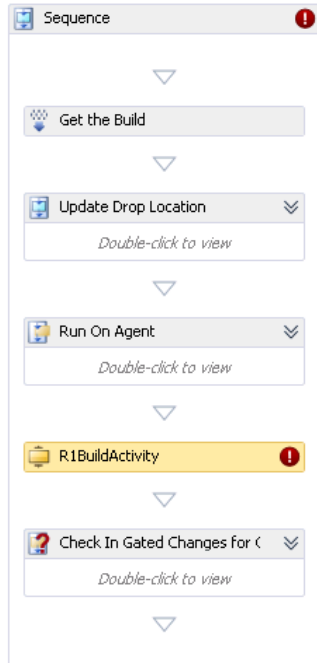


The build definition appears.

4. Click the **Process** tab, and change the Build process file to **Default Template**.



5. Click **Save**.
6. Repeat steps 3-5 for all the build definitions that use the template you checked out in step 2.
7. Remove R-1/ROSS arguments from the TFS build templates that have R-1/ROSS Build Activities installed, as described in [Running the RepliWeb TFS Build Utility](#).
8. Remove R-1/ROSS Build Activities. Repeat this procedure for each template with R-1/ROSS Build Activities.
 - a. Double-click the template file and delete the R-1/ROSS Build Activities from the template.



- b. Save the template and close it.
- c. Check in the template.
9. (Optional) Change the build process file(s) back to the original template and save.
10. Completely uninstall **RepliWeb TFS Plug-in** from the Architect computer. The plug-in can be uninstalled using the TFS Plug-in installer or [silently](#).

Uninstalling TFS Build Distributor

After removing the R-1/ROSS arguments and build activities from the template, simply uninstall TFS Plug-in from the Architect computer using the TFS Plug-in installer or [silently](#).

Uninstalling TFS Events Subscription

Uninstalling TFS Events Subscription consists of the following steps. These steps are performed on the TFS Server.

1. [Remove R-1/ROSS arguments and build activities from Build Templates.](#)
2. Unsubscribe the relevant team project collection(s) from the Web Service.
3. Once all collections are unsubscribed, delete the Events Subscription web site, which hosts these applications.
4. On the TFS Server, completely uninstall TFS Plug-in.

Note: These commands require administrative privileges on the TFS Server computer.

Unsubscribe from Web Service

This command disconnects the team project collection from the Web Service, and deletes the web application and application pool that were created for this team project collection.

To unsubscribe from the Web Service:

1. Run a command prompt as administrator and change the working directory to:

```
~\RepliWeb\TFS Plug-In\Bin
```

2. Issue the following command:

Syntax:

```
> rw_tfs_ws_config unsubscribe [-collection=url] | [-all]
```

Example:

```
> rw_tfs_ws_config unsubscribe -all
```

Parameters

| Parameter | Value |
|-------------|---|
| -collection | Team project collection URL. Note: Paths with spaces must be enclosed in quotation marks. For example: -collection="http://tfsserver:8080/tfs/my tpc" |
| -all | Unsubscribe and delete all the web applications created for all the team project collections. |

Delete Web Site

This command deletes the **Events Subscription** web site created in the [Installation](#) procedure.

To delete the web site:

1. Run a command prompt as administrator and change the working directory to:

```
~\RepliWeb\TFS Plug-In\Bin
```

2. Issue the following command:

Syntax:

```
> rw_tfs_ws_config delete_site
```

Example:

```
> rw_tfs_ws_config delete_site
```

Uninstall RepliWeb TFS Plug-in

Uninstall the TFS Plug-in. The plug-in can be uninstalled using the TFS Plug-in installer or silently, as follows.

Silent Uninstallation of RepliWeb TFS Plug-in

The plug-in can be uninstalled using qualifiers or MSI.

To silently uninstall RepliWeb TFS Plug-in using qualifiers:

1. Run a command prompt as administrator, issue the following command:

```
>set RW_INSTALL_MODE=remove
```

2. Change the working directory to the directory containing the RepliWeb TFS Plug-in setup file and then issue the following command:

```
>rw_tfs_plugin_<version.build>.exe /s
```

RepliWeb TFS Plug-in is removed from your computer.

To silently uninstall RepliWeb TFS Plug-in using MSI:

- ◆ Run a command prompt as administrator and change the working directory to the location of the `msiexec.exe` file (usually resides in `\Windows\System32`). Then issue the following command:

```
>msiexec.exe /qb- /x {DC6F49C6-FAE2-43C2-97E5-B93158D435EF}
```

where `/qb-` represents silently and `/x` represents uninstall.

2. Getting Started

This workflow guides you on how to use TFS Plug-in to manage your build process deployments.

1. Add two R-1 Upload templates to the build process. For example, “SendToTest” and “DeployProduction”.
2. Define an [Activity Deployment](#) build definition that replicates the normal build using the “SendToTest” template, and define a [Build Quality Deployment](#) build definition that replicates builds with the **Released** build quality using the “DeployProduction”.
3. [Run](#) the build.
4. [View](#) the build log and progress (during execution).
Upon completion, the build should be deployed using the “SendToTest” template.
5. Change the build quality to **Released**.
6. View the build log and progress (during execution).
The build should be deployed using the “DeployProduction” template.
7. [Define](#) two build qualities: **Rollback** and **Rollback Released**.
8. [Change](#) the build quality to **Rollback Released**.
9. View the build log and progress (during execution).
The “DeployProduction” build deployment job should be rolled back.
10. Change the build quality to **Rollback**.
11. View the build log and progress (during execution).
The “SendToTest” build deployment job should be rolled back.

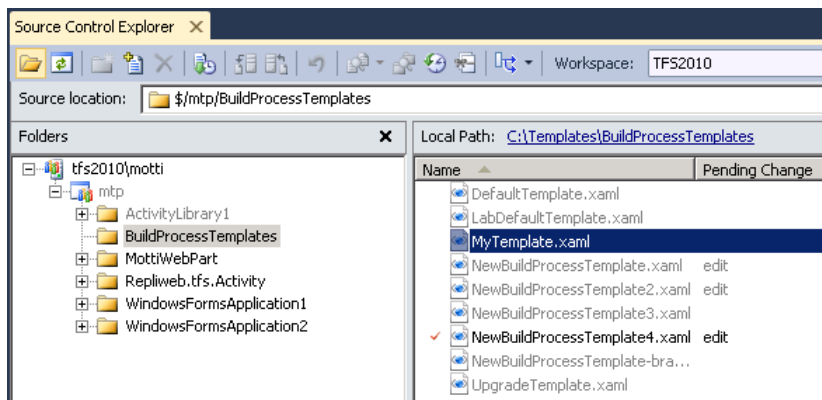
3. Adding the R-1/ROSS Build Activities to a Build Template

You can either add the R-1/ROSS Build Activities to an existing XAML file (build template) or create a new XAML file - as described in [Adding the R-1/ROSS Build Activities to a New Build Template](#) - and then add the RepliWeb Build Activity.

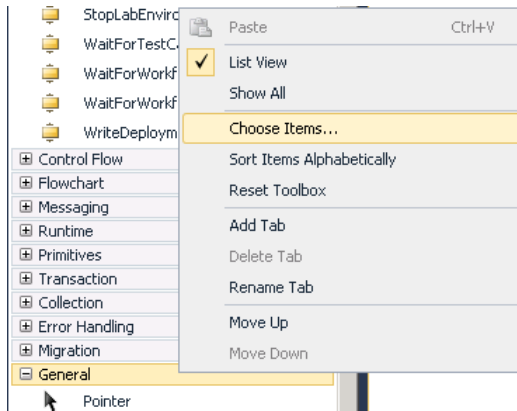
The procedures contained in this chapter explain how add the R-1/ROSS Build Activity to the same build process template. This would be necessary if you wanted to deploy binaries (using R-1) and SharePoint 2010 solutions (using ROSS) as part of the same workflow. However, if the build workflow only needs to deploy binaries, then you only need to add the R-1 Build Activity. Likewise, if the build workflow only needs to deploy SharePoint 2010 solutions, then you only need to add the ROSS Build Activity.

Adding the R-1/ROSS Build Activities to an Existing Build Template

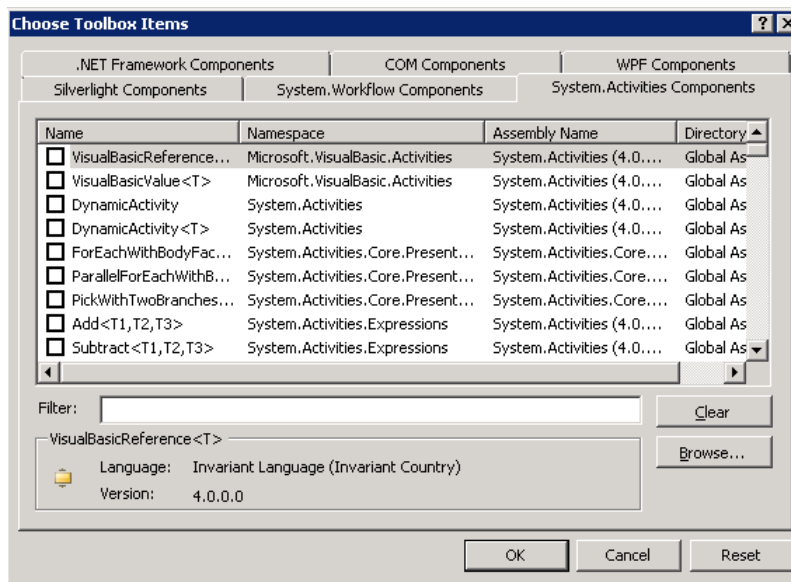
1. Open Source Control Explorer and locate the template to which you want to add the RepliWeb Build Activity.
2. Right-click the template and select **Check Out for Edit**. In the **Check Out** window, click **Check Out**.



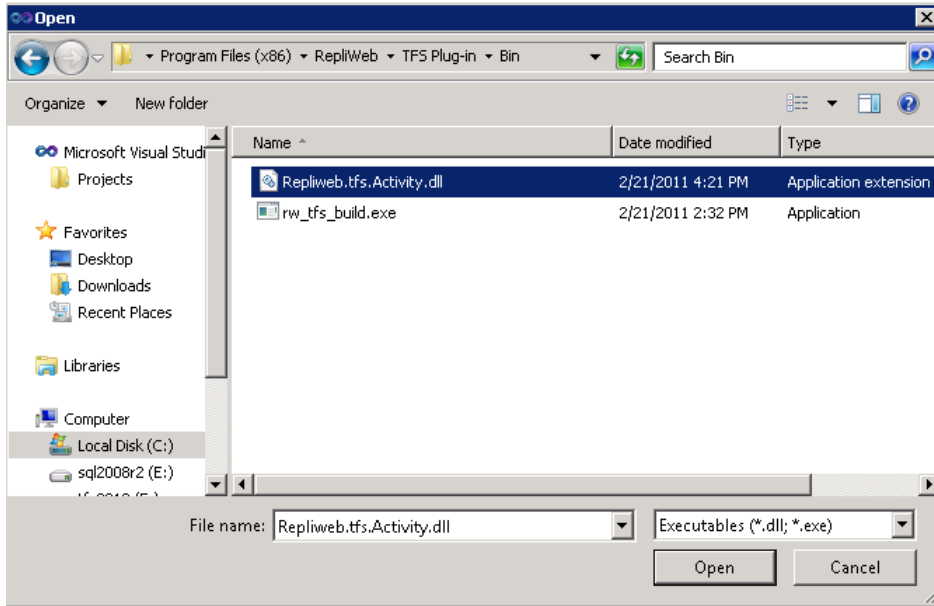
3. Double-click the template file.
4. Add the RepliWeb Build Activities to the **Toolbox** as follows:
 - a. In the **Toolbox (View → Toolbox)**, right-click the tab under which you want the RepliWeb Build Activities to be added and then select **Choose Items**.



The **Choose Toolbox Items** window opens.

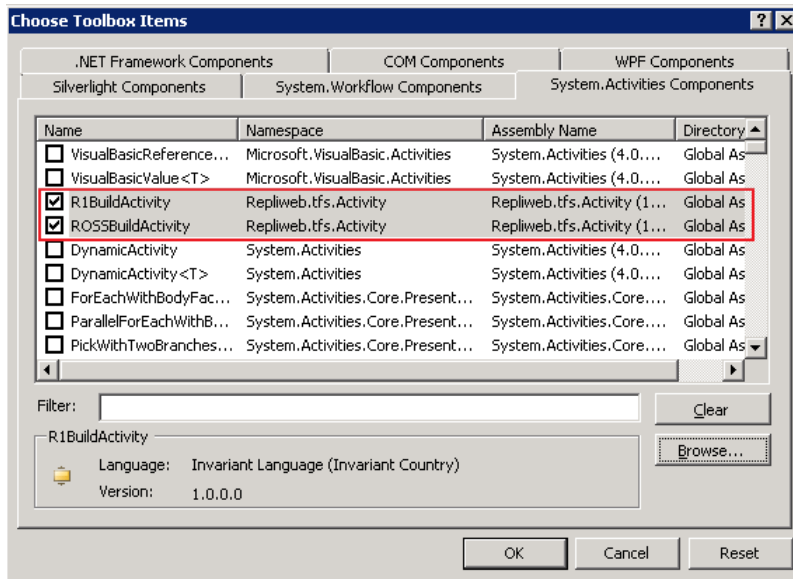


- b. Click **Browse**. The **Open** window opens.
- c. Browse to the folder containing the RepliWeb TFS Activity. If you accepted the installation default path, this folder will be: `~\RepliWeb\TFS Plug-in\Bin`.



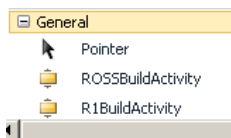
- d. Select the **Repliweb.tfs.Activity.dll** file and click **Open**.

The **R1BuildActivity** and **ROSSBuildActivity** items are added to the **System Activities Components** tab.

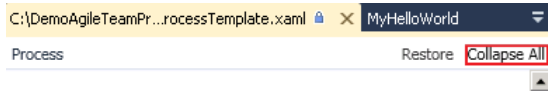


- e. Make sure that they are selected and then click **OK**.

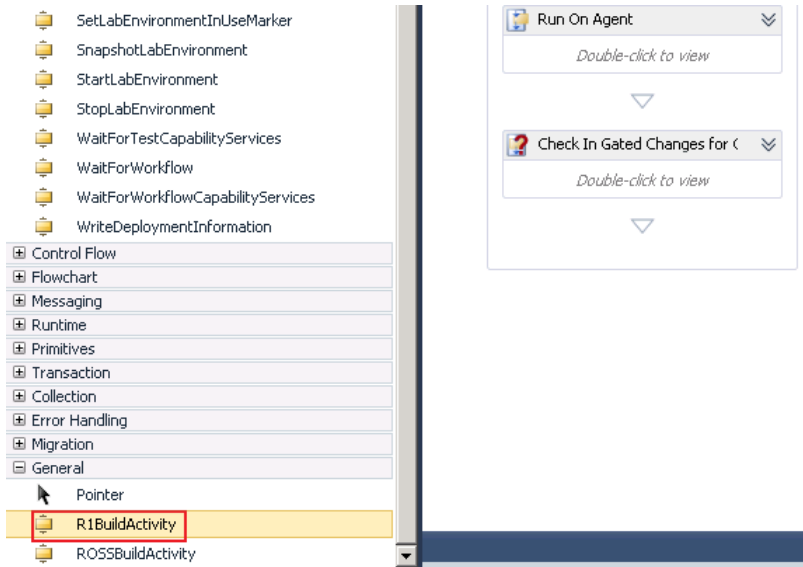
The items should now appear under their designated tab.



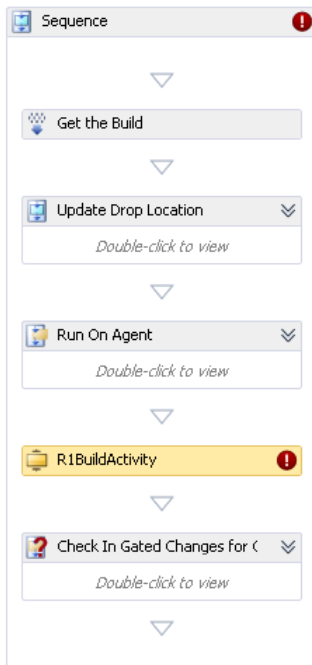
- In the top right of the build template tab, click **Collapse All**.



- Open the **Toolbox** and locate the **R1BuildActivity** item.

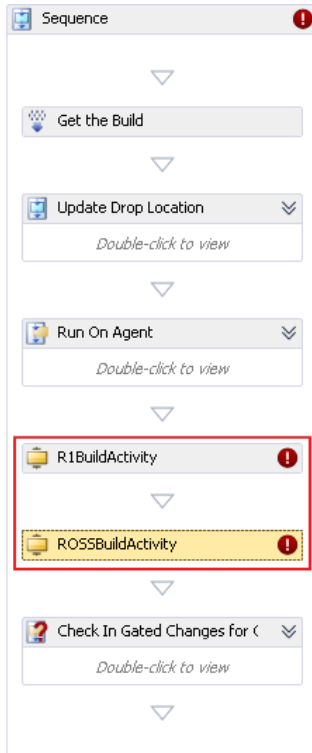


- Drag the **R1BuildActivity** item to the area below the **Run on Agent** activity.



- Repeat steps 5 and 6, but this time, locate and drag the **ROSSBuildActivity** item to the area below the **Run on Agent** activity (either above or below the **R1BuildActivity**).

Your template should now look like this:

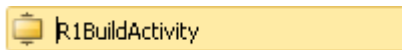


Note the exclamation mark to the right of the **R1BuildActivity** activity indicates that values for the required activity arguments have not been supplied yet. The required activity arguments and their values will be supplied at a later stage.

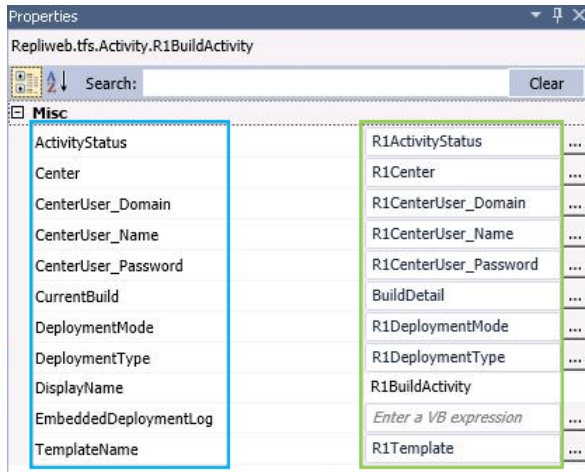
Note: The order of the RepliWeb activities is unimportant and merely determines which of the activities will be executed first.

9. Save the template and close it.
10. Run the RepliWeb Build Utility as described in [Running the RepliWeb TFS Build Utility](#).
11. Optionally, open the template and verify that a.) it contains all of the R1BuildActivity and ROSSBuildActivity arguments and b.) that each argument has been associated with its corresponding property.
 - a. To verify this for the R1BuildActivity:

Scroll down to the **R1BuildActivity** activity and click it.



The **R1BuildActivity** properties (shown in the blue rectangle) together with their corresponding template arguments (shown in the green rectangle) should now appear in the **Properties** window on the right.

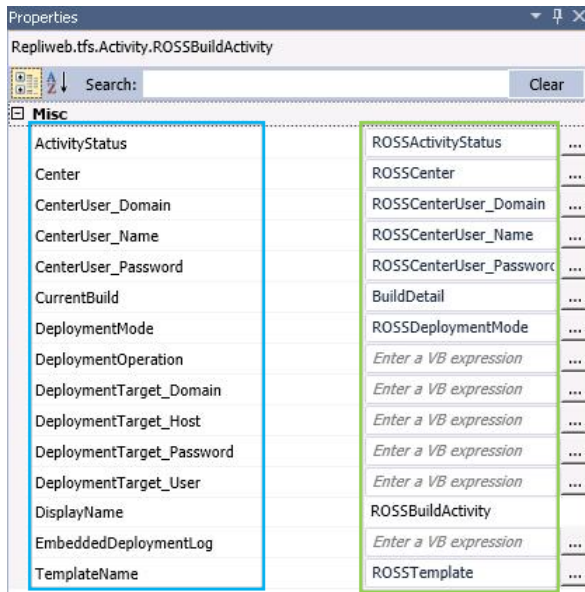


b. To verify this for the ROSSBuildActivity:

Scroll down to the **ROSSBuildActivity** activity and click it.



The **ROSSBuildActivity** properties (shown in the blue rectangle) together with their corresponding template arguments (shown in the green rectangle) should now appear in the **Properties** window on the right.



12. Check in the template.

Running the RepliWeb TFS Build Utility

The RepliWeb TFS Build Utility must be run as part of the procedure for adding R-1/ROSS activities to a build template and can optionally be run to restore a template to its previous state after removing the R-1/ROSS build activities.

When run as part of the procedure for adding R-1/ROSS activities to a build template, the utility will add R-1/ROSS activity arguments to a specified build template and then bind each argument to its corresponding R-1/ROSS Build Activity property.

Note: Before running this utility, make sure the xaml file is checked out.

Note: Before adding another activity's arguments to the same template, make sure you check in the xaml and then check it out again.

To add arguments with the RepliWeb TFS Build Utility:

1. Open a command prompt and change the working directory to:
~\RepliWeb\TFS Plug-In\Bin
2. Issue the following command:

Note: Make sure that the relevant RepliWeb Build Activity/Activities have been added to the template before issuing the command.

Syntax:

```
>rw_tfs_build add_arg -activity=[ross|r1|all] -template=<full path>
```

Example:

```
>rw_tfs_build add_arg -activity=r1 -template=c:\templates\mytemplate.xaml
```

To remove arguments with the RepliWeb TFS Build Utility:

1. Open a command prompt and change the working directory to:
~\RepliWeb\TFS Plug-In\Bin
2. Issue the following command:

Syntax:

```
>rw_tfs_build remove_arg -activity=[ross|r1|all] -template=<full path>
```

Example:

```
>rw_tfs_build remove_arg -activity=r1 -template=c:\templates\mytemplate.xaml
```

Parameters

| Parameter | Value |
|-----------|--|
| -template | The full path to the build template containing the R-1 and/or ROSS build activities. |

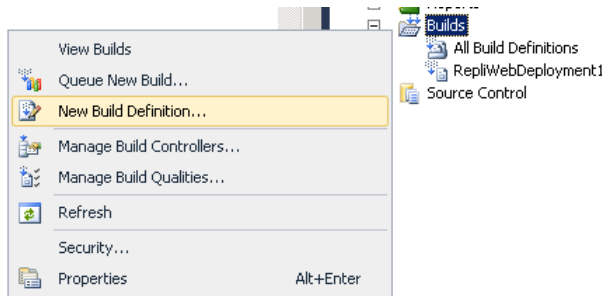
| | |
|------------------------|--|
| | <p>Note: Paths with spaces must be enclosed in quotation marks. For example: <code>-template="c:\template repository\mytemplate.xml"</code></p> |
| <code>-activity</code> | <p><code>[r1 ross all]</code></p> <p>Specify:</p> <p>r1 to add R-1 arguments to the build template.</p> <p>ross to add ROSS arguments to the build template.</p> <p>all to add both R-1 and ROSS arguments to the build template.</p> <p>Note: Make sure that the value corresponds to the RepliWeb Build Activity in the specified build template. For example, do not specify ross if the ROSS build Activity has not been added to the build template.</p> |

Adding the R-1/ROSS Build Activities to a New Build Template

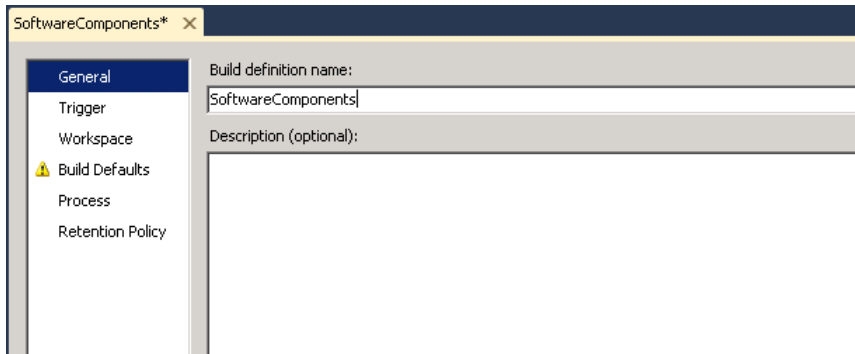
This section explains how to create a new template and add the RepliWeb Build Activities to it.

To add the R-1/ROSS Build Activities to a New Build Template:

1. Open Visual Studio 2010.
2. In the Team Explorer window, right-click **Builds** and then select **New Build Definition**.

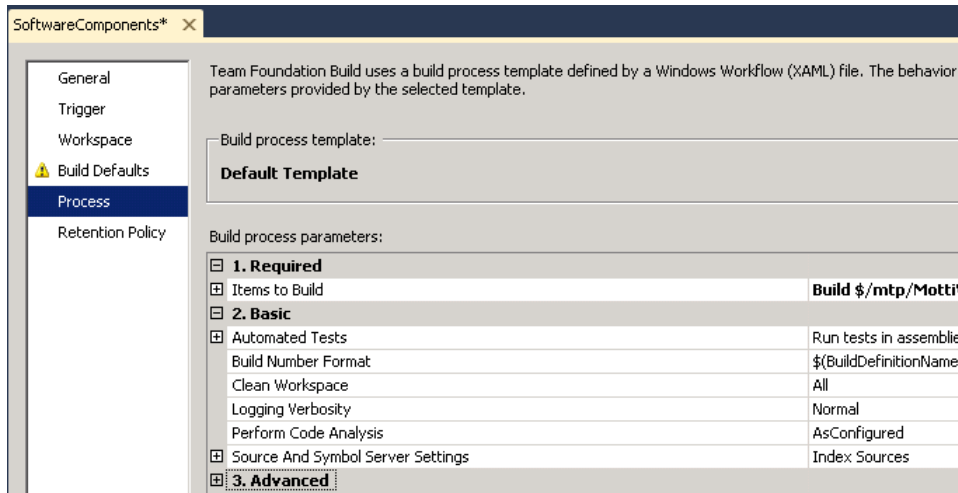


The Build Definition window opens.



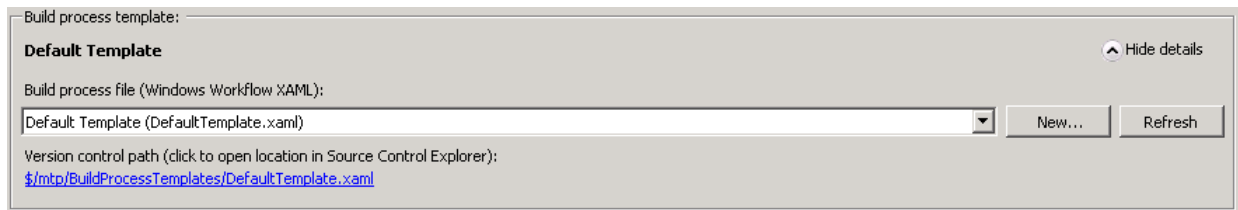
3. In the left pane, click **Process**.

The **Process** tab is displayed.



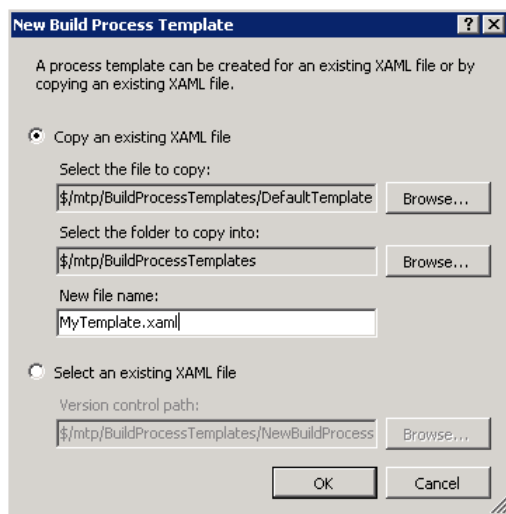
4. In the **Build process template** area, click the **Show details** button on the right.

The build process template file is revealed, together with the **New** and **Refresh** buttons.



5. Click **New**.

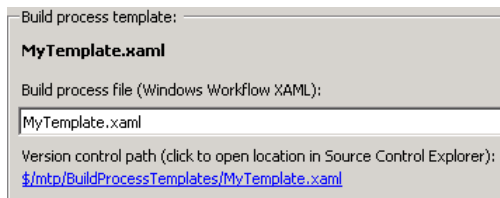
The **New Build Process Template** dialog box opens.



6. In the **New Build Process Template** dialog box:
 - a. Select **Copy an existing XAML file**,
 - b. In the **Select the file to copy** field, specify which file to copy.
 - c. In the **New file name** field, provide a name for the new template.

- d. Click **OK**.

The new template should now be displayed.



7. Click the version control path to open the location of the template in Source Control Explorer and then continue from [Adding the R-1/ROSS Build Activities to an Existing Build Template](#).

4. Configuring R-1/ROSS Build Deployment

Build Definers who wish to use the RepliWeb Build Activities need to know which values to enter in the build definition's **Process** tab. These values should be communicated to the Build Definers beforehand (e.g by email).

The tables below provide a description of each of the R-1/ROSS Build Activity parameters.

R-1 Parameters

| Parameter Name | Description |
|---|--|
| Activity Deployment | Enable/Disable R-1 build deployment. |
| Activity Deployment Mode Possible Values: Synchronous or Asynchronous Default: Synchronous | Choose Synchronous to pause the build process until the R-1 Deployment Job completes or Asynchronous to continue the build process. If you select "Asynchronous", you will not see the job's results in the build log, and you will have to track the job using the R-1 Console). Note: In "Synchronous" Deployment Mode, it is not recommended to stop or postpone a running build process once an R-1 job is running. Additionally, in situations where "Synchronous" Deployment Mode is employed and the job is in recovery, the build process may not continue until all recoveries are exhausted. |
| Activity Deployment Template | The R-1 Job Template(s) to use for deploying completed builds. Use the R-1 Console to create a template. IMPORTANT: a.) The R-1 Template can be based on Upload, Distribution or MSIDeployUpload Jobs. b.) The template must be saved in the "Common on Center" location. Note: The only settings that are required are as follows: a template name, the entities that you want to deploy (e.g. Files) and the replication target values. For detailed instructions on creating an R-1 template, please refer to the R-1 Console Help. |
| Activity Deployment Type Possible Values: Upload, Distribution or | The type of R-1 job to use to deploy the completed build. This value must be the same as the specified Activity Deployment Template's job type. |

| | |
|--------------------------------------|--|
| MSIDeployUpload | |
| Default: Upload | |
| Build Quality Deployment | Enable/Disable deployment based on changes to the build quality. |
| Build Quality Deployment Events | Specify the Build Quality event(s) that will trigger the deployment of the build, when manually set. For R-1, the deployment is done using the job properties defined in the event's corresponding template, which is specified in the Build Quality Deployment Templates parameter below. Note: Multiple entries must be separated by a comma (,). |
| Build Quality Deployment Templates | Specify an R-1 template for each of the Build Quality Deployment Events. If multiple Build Quality Deployment Events are specified, then multiple templates must also be specified (one for each build quality event). Note: Multiple entries must be separated by a comma (,). |
| Build Quality Deployment Types | Specify the job type (Upload, Distribution or MSIDeployUpload) for each of the specified Build Quality Deployment Templates. If several Build Quality Deployment Templates are specified, the order of the Build Quality Deployment Types must be the same (in terms of job type). For example, if the first Build Quality Deployment Template is based on an Upload Job, then the first Build Quality Deployment Type must be "Upload". If the second Build Quality Deployment Template is based on a Distribution Job, then the second Build Quality Deployment Type must be "Distribution", and so on. Note: Multiple entries must be separated by a comma (,). |
| R-1 Center | The hostname or IP address of the Center. Note: The Center computer must be able to access the drop location (an UNC path). The UNC feature must also be enabled in the R-1 license. |
| R-1 Center User Domain | The domain name if the user belongs to a domain. |
| R-1 Center User Name | The user under which the R-1 Deployment Job will run. IMPORTANT: This user must be granted Log on as Batch rights on the R-1 Center computer. |
| R-1 Center User Password (Scrambled) | The password associated with the user account. The password must be scrambled using the R-1 CLI. For instructions on scrambling passwords, see A. Scrambling Passwords . |
| R-1 Job Target Directory | The target directory for the R-1 job template. Note: This value will override the Target Directory in the R-1 job template, if specified. |

ROSS Parameters

| Parameter Name | Description |
|--|---|
| Activity Deployment | Enable/Disable ROSS build deployment. |
| Activity Deployment Mode Possible Values: Synchronous or Asynchronous Default: Synchronous | <p>Choose Synchronous to pause the build process until the ROSS Deployment Job completes or Asynchronous to continue the build process. If you select "Asynchronous", you will not see the job's results in the build log, and you will have to track the job using the ROSS Console).</p> <p>Note: In "Synchronous" Deployment Mode, it is not recommended to stop or postpone a running build process once a ROSS job is running. Additionally, in situations where "Synchronous" Deployment Mode is employed and the job is in recovery, the build process may not continue until all recoveries are exhausted.</p> |
| Activity Deployment Template | <p>The ROSS Job Template(s) to use for deploying completed builds. Use the ROSS Console to create a template.</p> <p>IMPORTANT:</p> <ul style="list-style-type: none"> a.) The ROSS Template must be based on a Solution Deployment Upload. b.) The template must be saved in the "Common on Center" location. <p>Note: The only settings that are required are as follows: a template name and the replication target values.</p> <p>For detailed instructions on creating a ROSS template, please refer to the ROSS Console Help.</p> |
| Build Quality Deployment | (Relevant for Build Quality Deployment) Choose whether or not to enable deploying each build according to the Build Quality specified. |
| <p>Note: In the parameters below, multiple values must be separated by a comma (,). When multiple values are specified, the order of the parameter values must correspond to each other. For example, if the first Build Quality Deployment Event specified is "Rejected", then the first Build Quality Deployment Operation specified must be the operation you want to perform in the event that the build quality is changed to "Rejected" (e.g. "Deploy"). If the second Build Quality Deployment Event specified is "Release", then the second Build Quality Deployment Operation specified must be the operation you want to perform if the build quality is changed to "Release Candidate" (e.g. "Upgrade"), and so on. This is also true when specifying multiple targets: The first host specified must correspond to the first username specified which in turn must correspond to the first password specified, and so on.</p> | |
| Build Quality Deployment Events | Specify the Build Quality event(s) that will trigger the deployment of the build, when manually set. Each build will be deployed according to the corresponding ROSS deployment job properties (see below). |
| Build Quality Deployment Templates | Specify a ROSS template for each of the Build Quality Deployment Events. If multiple Build Quality Deployment Events are specified, then multiple templates must also be specified (one for each build quality event). |

| | |
|---------------------------------------|---|
| ROSS Center | The hostname or IP address of the Center. Note: The Center computer must be able to access the drop location (an UNC path). The UNC feature must also be enabled in the ROSS license. |
| ROSS Center User Domain | The domain name if the user belongs to a domain. |
| ROSS Center User Name | The user under which the ROSS Deployment Job will run. IMPORTANT: This user must be granted Log on as Batch rights on the ROSS Center computer. |
| ROSS Center User Password (Scrambled) | The password associated with the user account. The password must be scrambled using the ROSS CLI. For instructions on scrambling passwords, see A. Scrambling Passwords . |

Built-In Parameters

In order for the ROSS Build Activity to function properly, the **MSBuild Arguments** parameter must be configured according to the [description below](#). Although not required, we also suggest changing the **Clean Workspace** parameter as [recommended below](#).

Basic

| Parameter Name | Description |
|-----------------|---|
| Clean Workspace | <p>For reasons of efficiency, we strongly recommend changing the default value of the Clean Workspace parameter (ALL) to None. If you leave the value set as ALL, the TFS Agent will create a new list of sources and binaries each time the build process runs. In this case, the files in the drop location will always have different dates than the files on the target, even though the actual content of the files may not have changed at all.</p> <p>If the file names in the drop location are the same as the file names on the target, R-1/ROSS compares the dates of the files in the drop location with the dates of the files on the target computer and only copies the files whose dates have changed. If the Clean Workspace parameter is set to ALL, the dates of the files in the drop location will always be different and, as a result, R-1/ROSS will always copy all of the files to the target computer. However, when the Clean Workspace parameter is set to None, the TFS Agent will only update sources that have changed and only compile binaries whose sources have changed. In this case, even though all the files will be copied to the drop location, R-1/ROSS will only copy files whose dates have changed to the target computer.</p> |

Advanced (ROSS Build Activity Only)

| Parameter Name | Description |
|-------------------|---|
| MSBuild Arguments | Specify the following argument: /p:IsPackaging=True This argument instructs the ROSS Build Activity to deploy the compiled solutions as .wsp files. |

Example: Build Process Tab Configuration

This is an example of a TFS build process configured to be deployed by R-1 at build process completion (using **TFS Build Distributor**) and according to “Released” and “Ready for Deployment” build qualities (using **TFS Events Subscription**).

| Process | |
|-------------------------------------|--|
| Retention Policy | Build process parameters: |
| <input checked="" type="checkbox"/> | 1. Required |
| <input checked="" type="checkbox"/> | Items to Build Build \$/TfsDemo/ConsoleApp/ConsoleApp.sln with defa |
| <input checked="" type="checkbox"/> | 2. Basic |
| <input checked="" type="checkbox"/> | Automated Tests Run tests in assemblies matching ***test*.dll |
| | Build Number Format \$(BuildDefinitionName)_\$(Date:yyyyMMdd)\$(Rev:.r) |
| | Clean Workspace None |
| | Logging Verbosity Normal |
| | Perform Code Analysis AsConfigured |
| <input checked="" type="checkbox"/> | Source And Symbol Server Settings Index Sources |
| <input checked="" type="checkbox"/> | 3. Advanced |
| <input checked="" type="checkbox"/> | 4. Misc |
| | R1EmbeddedDeploymentLog Detailed |
| <input checked="" type="checkbox"/> | 5. R-1 Build Activity |
| | Activity Deployment Enabled |
| | Activity Deployment Mode Synchronous |
| | Activity Deployment Template TfsR1Upload |
| | Activity Deployment Type Upload |
| | Build Quality Deployment Enabled |
| | Build Quality Deployment Events Released,Ready for Deployment |
| | Build Quality Deployment Templates R1Upoad_released,TfsR1Upload_ready |
| | Build Quality Deployment Types Upload,Upload |
| | R-1 Center sp2010d02.dev.repliweb.com |
| | R-1 Center User Domain rwdev |
| | R-1 Center User Name sp2010d02fa |
| | R-1 Center User Password (Scrambled) 415OPHV4R45B5600UMFR3K10L0 |

5. Advanced Configuration Options

This chapter describes how to change the R-1/ROSS Build Activity argument display names and how to change the default argument values in the build template.

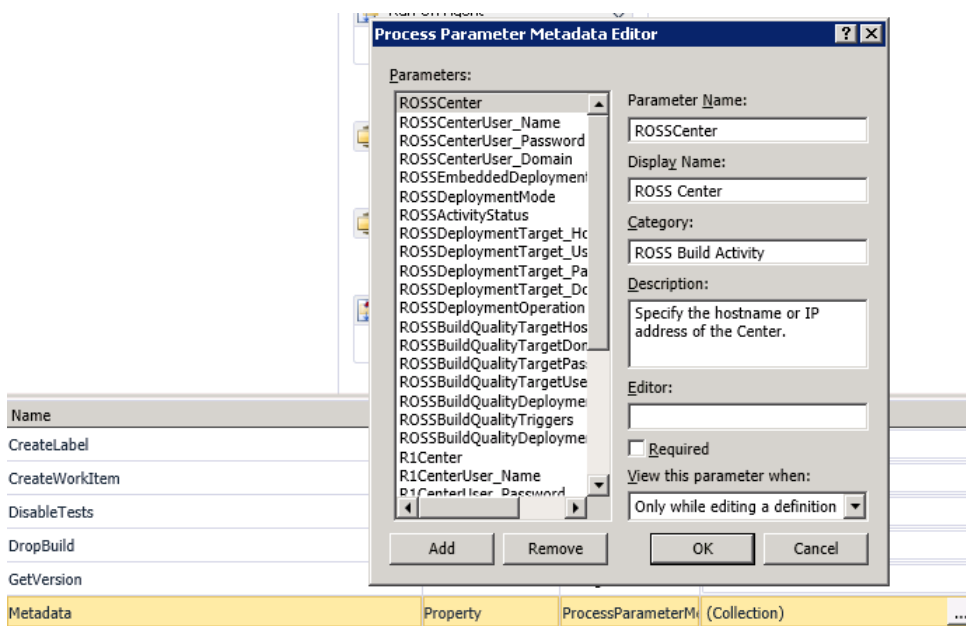
Editing the R-1/ROSS Build Activity Metadata

Each of the R-1/ROSS Build Activities arguments or parameters has a user-friendly display name. The R-1/ROSS Build Activities parameters in the Build Definition's **Process** tab are shown using their display name.

| Process | |
|---------------------------------------|--|
| Retention Policy | Build process parameters: |
| 1. Required | |
| Items to Build | Build \$/TeamProject2/VisualWebPart3/VisualWebPart3.sln wi |
| 2. Basic | |
| Automated Tests | Run tests in assemblies matching ***test*.dll |
| Build Number Format | \$(BuildDefinitionName)_\$(Date:yyyyMMdd)\$(Rev:.r) |
| Clean Workspace | All |
| Logging Verbosity | Normal |
| Perform Code Analysis | AsConfigured |
| Source And Symbol Server Settings | Index Sources |
| 3. Advanced | |
| 4. R-1 Build Activity | |
| Activity Deployment | Enabled |
| Activity Deployment Mode | Synchronous |
| Activity Deployment Template | r1_upload |
| Activity Deployment Type | Upload |
| Build Quality Deployment | Disabled |
| Build Quality Deployment Events | |
| Build Quality Deployment Templates | |
| Build Quality Deployment Types | |
| R-1 Center | 10.54.63.189 |
| R-1 Center User Domain | comp_domain |
| R-1 Center User Name | admin |
| R-1 Center User Password (Scrambled) | 8HJE350KM4352 |
| R-1 Job Target Directory | C:\R-1\BuildTarget |
| 5. ROSS Build Activity | |
| Activity Deployment | Disabled |
| Activity Deployment Mode | Synchronous |
| Activity Deployment Template | ROSS_Rel |
| Build Quality Deployment | Enabled |
| Build Quality Deployment Events | Released,Rejected |
| Build Quality Deployment Templates | ROSS_Rel,ROSS_Rej |
| ROSS Center | 10.0.56.197 |
| ROSS Center User Domain | comp_domain |
| ROSS Center User Name | admin |
| ROSS Center User Password (Scrambled) | 7KRYS4F7Q54D1 |

To see a list of all the parameters and their display names (and optionally edit them), locate the **Metadata** argument in the build template and click the browse button to the right of the default value (Collection).

The **Process Parameter Metadata Editor** opens.



Selecting a parameter reveals its metadata. The parameter metadata consists of the parameter **Display Name**, the **Category** under which the parameter will appear (in the build definition **Process** tab) and a **Description** of the parameter. You can optionally change the display name and the description.

Important: In order to change the parameter names, you first need to perform the procedure described in [B. Changing Default Argument Names.](#)

Changing the Default Argument Values in the Build Template

The [RepliWeb TFS Build Utility](#) adds R-1/ROSS Build Activity arguments to the build template and then binds those arguments to the R-1/ROSS Build Activity properties. Some of these arguments have default values that you can optionally change. You can also specify default values for arguments that do not have a default value. Most of the default values are strings. All strings need to be enclosed in quotation marks. For example, to set a default value for the **Center** argument, simply type "<Center IP>" in the **Default value** column where **Center IP** is the IP address of the R-1 Center.

| Name | Direction | Argument type | Default value |
|----------------------------|-----------|----------------|--|
| R1Center | In | String | 12.5.47.32 |
| R1CenterUser_Domain | In | String | Enter a VB expression |
| R1CenterUser_Name | In | String | Enter a VB expression |
| R1CenterUser_Password | In | String | Enter a VB expression |
| R1DeploymentMode | In | DeploymentMode | Repliweb.tfs.Activity.DeploymentMode |
| R1DeploymentType | In | DeploymentType | Repliweb.tfs.Activity.DeploymentType. |
| ROSSBuildQualityDeployment | In | ActivityStatus | Repliweb.tfs.Activity.ActivityStatus.Dis |

The default values for the following arguments are selected from drop-down lists and not specified as strings: DeploymentMode, EmbeddedDeploymentLog and R1ActivityStatus.

To change these default values:

1. Double-click the default value.
The value field becomes editable.
2. To reveal the possible default values, delete the last letter of the value (as shown below).

| Name | Direction | Argument type | Default value |
|--------------------------------------|-----------|-----------------------|--|
| R1CenterUser_Password | In | String | Enter a VB expression |
| R1DeploymentMode | In | DeploymentMode | Repliweb.tfs.Activity.DeploymentMode |
| R1DeploymentType | In | DeploymentType | Repliweb.tfs.Activity.DeploymentType. |
| R1EmbeddedDeploymentLog | In | EmbeddedDeploymentLog | Repliweb.tfs.Activity.EmbeddedDeploymentLog |
| R1Template | In | String | Enter a VB expression |
| ROSSActivityStatus | In | ActivityStatus | Repliweb.tfs.Activity.ActivityStatus.Enabled |
| ROSSBuildQualityDeployment | In | ActivityStatus | Repliweb.tfs.Activity.ActivityStatus.Enabled |
| ROSSBuildQualityDeploymentOperations | In | String | Enter a VB expression |

3. Double-click the desired value in the pop-up menu and then press [Enter].
The new value is displayed:

| | | | |
|--------------------|----|----------------|--|
| ROSSActivityStatus | In | ActivityStatus | Repliweb.tfs.Activity.ActivityStatus.Enabled |
|--------------------|----|----------------|--|

4. Check in the template.

6. Managing Build Deployments

This chapter explains how to manage your build deployments, including configure build qualities, initiate build deployment jobs, and roll back build deployment jobs.

Build Process Deployment Jobs

RepliWeb TFS Plug-in deploys the TFS build according to the build deployment operation:

- If **Activity Deployment** is enabled, the plug-in automatically deploys completed build processes to the target based on the *Activity Deployment Template* specified.
- If **Build Quality Deployment** is enabled, build processes are deployed whenever a defined build quality is manually selected, based on the *Build Quality Deployment Template* defined for that build quality. Multiple build qualities can be defined, each with its own job Template.

For more information about Build Deployment parameters, see [3. Configuring R-1/ROSS Build Deployment](#).

Initiating New Build Processes

Note: This procedure can only be performed by an Administrator user.

To initiate a new build process:

1. Open Visual Studio 2010.
2. In the Team Explorer window, right-click the desired **Build** and select **Queue New Build**.

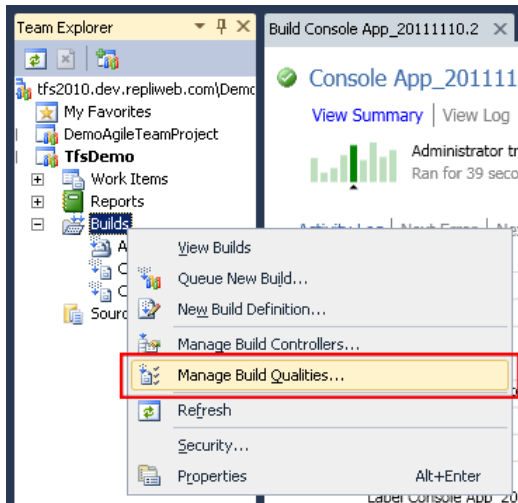
Defining Build Qualities

This procedure explains how to define build qualities.

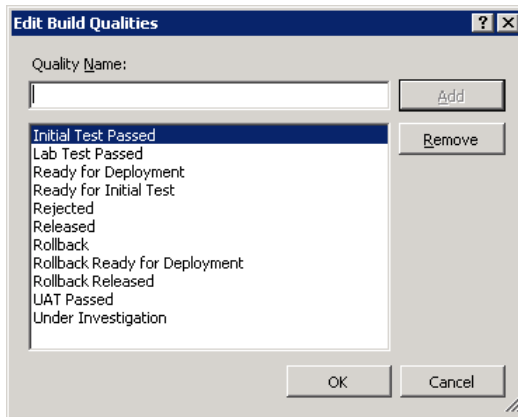
Note: This procedure can only be performed by an Administrator user.

To define build qualities:

1. Open Visual Studio 2010.
2. In the Team Explorer window, right-click **Builds** and then select **Manage Build Qualities**.



The **Edit Build Qualities** window appears.



3. Specify a new build quality and click **Add**.
4. Click **OK**.

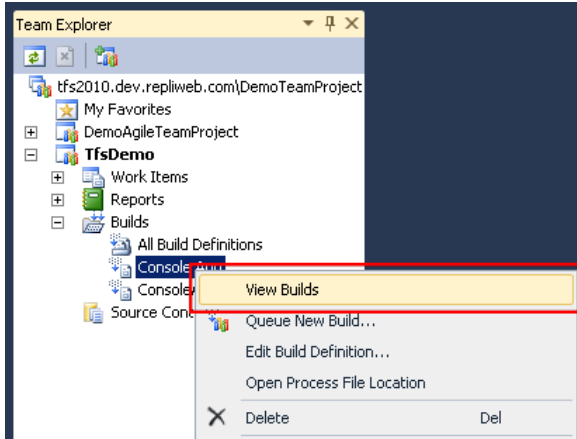
Changing Build Qualities

Use this procedure to change the build quality of a build process.

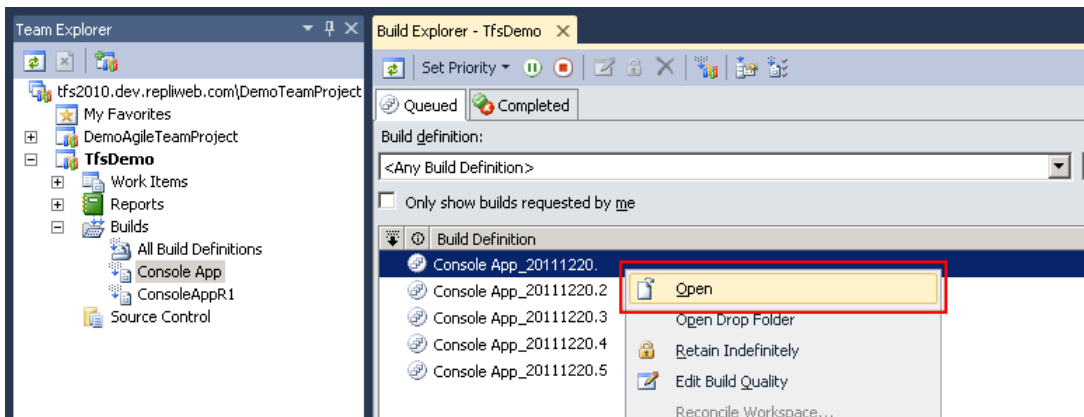
Note: This procedure can only be performed by a user with the *Edit Build Quality* permission.

To change a build quality:

1. Open Visual Studio 2010.
2. In the Team Explorer window, right click the desired build and select **View Builds**.

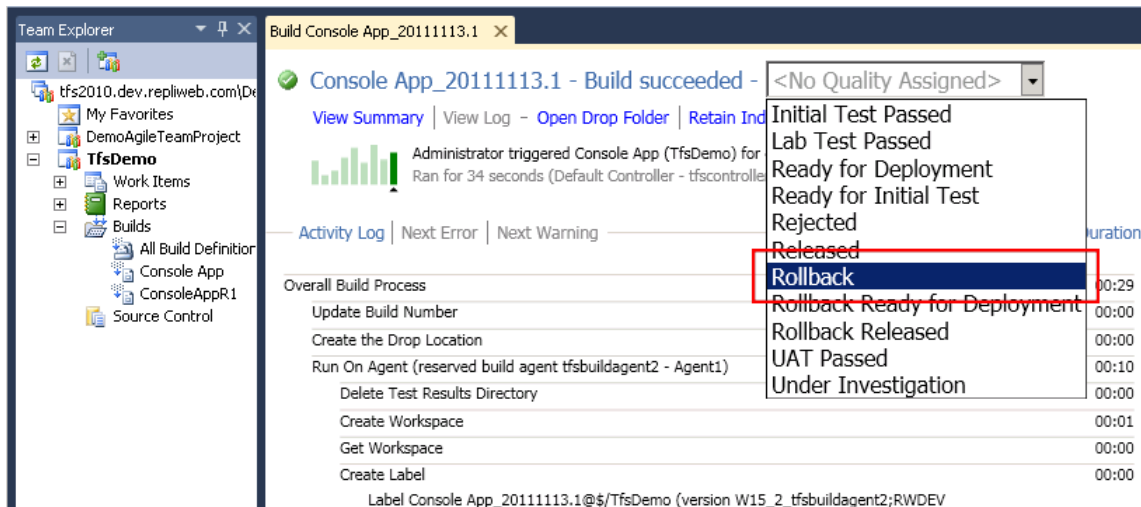


3. In the **Build Explorer** tab, click the **Completed** tab and open the desired build.



4. From the drop-down list, select the appropriate build quality.

For example, “Rollback”:



TFS Plug-in triggers the build’s deployment accordingly.

5. Click **[F5]** to refresh. The quality change is reported in the build’s log along with a link to the R-1/ROSS Deployment job report.

Build Process Deployment Rollback Jobs

Build Deployment Rollback is a two-step process. First, the desired Rollback qualities need to be defined; then, the build process build quality must be manually changed to the desired Rollback quality. The build that is linked to the selected build quality is then rolled back accordingly.

Note: Make sure the relevant R-1/ROSS build deployment template is Rollback enabled. Also, since Rollback uses build qualities, make sure [Events Subscription](#) is fully installed on the TFS Server and enabled in the build process.

Defining Rollback Qualities

This procedure explains how to define build qualities that trigger Rollback.

Note: This procedure can only be performed by a team project administrator.

To define Rollback build qualities:

- ◆ Use the [Defining Build Qualities](#) procedure to define the desired Rollback build qualities.
To roll back the deployment job of an Activity Deployment build process, add the “**Rollback**” build quality deployment event.
To roll back a deployment job that was triggered by a specific Build Quality Deployment event, make sure to add an additional build quality event with the same name but prefixed with the word “Rollback”. For example, to roll back the “Released” Build Quality deployment job, specify “**Rollback Released**”.

Rolling Back Build Deployments

Use this procedure to roll back build process deployments.

Note: This procedure can only be performed by a user with the *Edit Build Quality* permission.

To roll back a build deployment job:

1. Follow the procedure described in the [Changing Build Qualities](#) section and select the appropriate Rollback quality.
2. Click **[F5]** to refresh. The Rollback job’s report appears in a link at the bottom of the Deployment Log when Rollback completes.

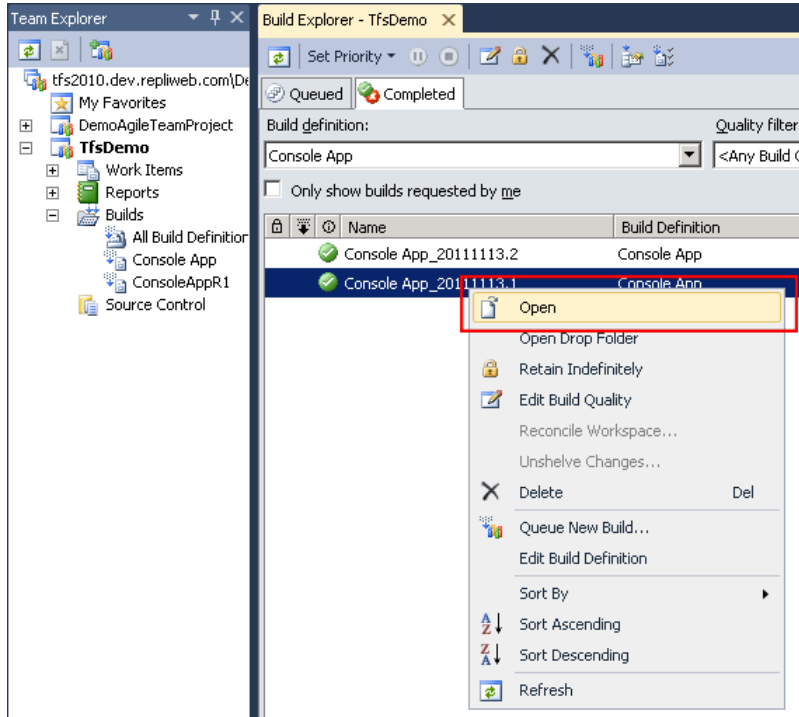
Build deployment rollback applies to all the Edges in the job.

Deployment Job Reports

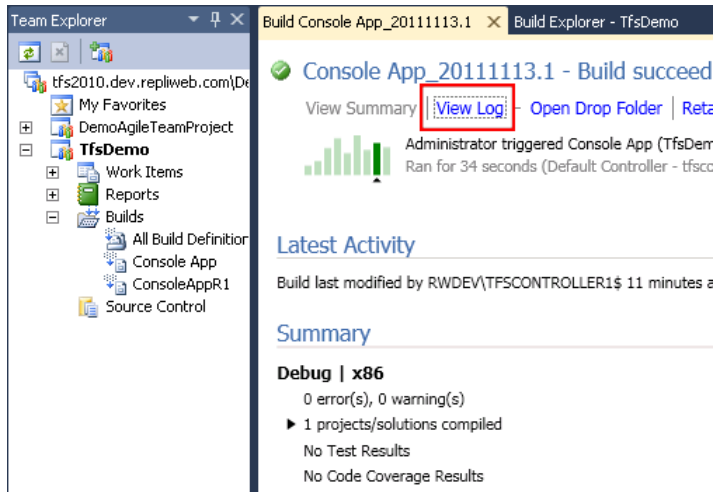
R-1/ROSS Build Deployment Job reports appear as links in the build process log (highlighted below) and are updated in real time with the jobs’ progress. Additionally, each Build Quality change appears in the TFS Build Log with a link to its R-1/ROSS Deployment Job report.

To view a Deployment Job’s Log:

1. Open Visual Studio 2010.
2. In the Team Explorer window, right-click **Builds** and select **View Builds**.
3. In the Build Explorer tab, right click the desired build and select **Open**.



4. In the Build Summary tab, click the **View Log** link.



The build’s **Deployment Job Log** appears.

The deployment job steps appear in the build log during execution.

The log also contains a link “Deployment job”. Clicking this link opens the job’s report, enabling you to track the job’s progress.

Build Console App_20111113.1 | Build Explorer - TfsDemo

Console App_20111113.1 - Build succeeded - UAT Passed

[View Summary](#) | [View Log](#) - [Open Drop Folder](#) | [Retain Indefinitely](#) | [Delete Build](#)

Administrator triggered Console App (TfsDemo) for changeset 322
 Ran for 34 seconds (Default Controller - tfscontroller1), completed 19.1 minutes ago

[Activity Log](#) | [Next Error](#) | [Next Warning](#) [Duration](#)

| Activity | Duration |
|---|----------|
| Microsoft.TeamFoundation.Build.Server.Logger.dll; "Verbosity=Normal;" | |
| Built \$/TfsDemo/ConsoleApp/ConsoleApp.sln for default targets. | 00:00 |
| Built \$/TfsDemo/ConsoleApp/ConsoleApp/ConsoleApp.csproj for default targets. | 00:00 |
| MSBuild Log File | |
| Associate Changesets and Work Items | 00:01 |
| Analyzing labels Console App_20111110.5 and Console App_20111113.1. | |
| Get Impacted Tests, Index Sources and Publish Symbols | 00:01 |
| Get Impacted Tests | 00:00 |
| Index Sources | 00:00 |
| C:\Program Files\Microsoft Team Foundation Server 2010\Tools\srctool.exe "C:\Builds\2\TfsDemo\Console App\Binaries\ConsoleApp.pdb" -r | |
| C:\Program Files\Microsoft Team Foundation Server 2010\Tools\pdbstr.exe -w -p:"C:\Builds\2\TfsDemo\Console App\Binaries\ConsoleApp.pdb" -i:C:\Windows\ServiceProfiles\NetworkService\AppData\Local\Temp\tmpCE40.tmp -s:srcsrv | |
| Skipped execution of Revert Workspace because of BuildReason Manual | |
| Copy Files to Drop Location | 00:00 |
| R1 Post-build Deployment | 00:18 |
| R-1 Deployment | |
| Connecting to R-1 Center 'sp2010d02.dev.repliweb.com' | 00:00 |
| Submitting R-1 replication job | 00:00 |
| Deployment log | |
| Deployment succeeded | 00:00 |
| Skipped execution of Check In Gated Changes because of BuildReason Manual | |

7. TFS Build Deployment Variables

The following variables can be used in R-1 and ROSS TFS Build Deployment jobs. For details defining R-1 or ROSS job variables, see the *R-1 User Guide* or *ROSS User Guide*, respectively.

All R-1/ROSS Jobs

| | |
|----------------|------------------------------|
| TFSBUILDNAME | The “Build Definition Name”. |
| TFSBUILDNUMBER | The “Build Number”. |

R-1 MSI Deployment Jobs

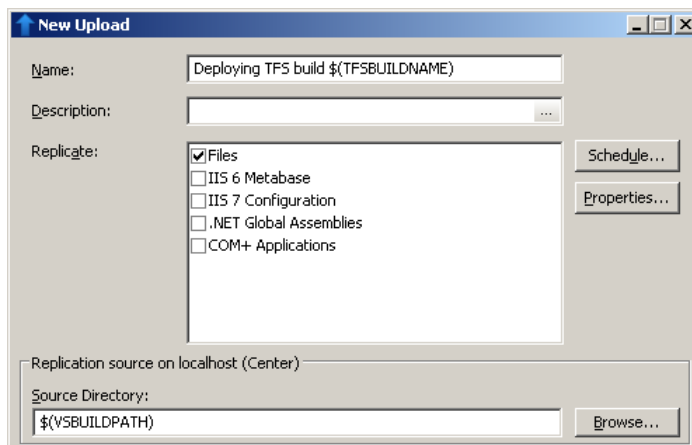
| | |
|-------------|--|
| VSBUILDPATH | The path to the “Build Drop Location”. |
|-------------|--|

To define variables in the TFS Build Deployment job template:

1. In the job definition on the Center, specify the variable(s) in the appropriate field(s) along with the other settings and save as a Template.
2. In the [TFS Build deployment](#), specify this Template to be used for deploying the TFS build (in the **Activity Deployment Template** or **Build Quality Deployment Templates** parameters).

Variables should appear as follows: “\$(VARIABLE)”

For example, R-1 Upload job with TFSBUILDNAME and VSBUILDPATH:



3. Run a new build process.

Once Build Deployment is triggered, the TFS build is deployed using a job instance that is based on this template.

A. Scrambling Passwords

Scrambling passwords protects systems against unauthorized access since scrambled passwords can only be used within the context of R-1 or ROSS jobs.

The following passwords must be specified in scrambled format:

- R-1 Center User Password
- ROSS Center User Password
- ROSS Deployment Target Password

To scramble the R-1 Center and Edge passwords:

1. Open a command prompt and change the working directory to:
~\RepliWeb\R1\Console\Bin
2. Issue the following command:
>r1 scramble -password=<password_to_scramble>
The password will be encrypted.

Example:

```
C:\Program Files\RepliWeb\R1\Console\Bin>r1 scramble -password=guest
Your encrypted password is: L9CFMSD294ACQIBSSR70NCKJ80
```

To scramble the ROSS Center and Target passwords:

1. Open a command prompt and change the working directory to:
~\RepliWeb\ROSS\Console\Bin
2. Issue the following command:
>ross scramble -password=<password_to_scramble>
The password will be encrypted.

Example:

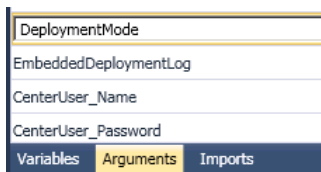
```
C:\Program Files\RepliWeb\ROSS\Console\Bin>ross scramble -password=guest
Your encrypted password is: L9CFMSD294ACQIBSSR70NCKJ80
```

B. Changing Default Argument Names

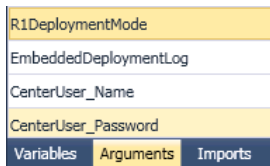
Although not recommended, you can manually change the default names of the arguments added with the RepliWeb TFS Build utility. If you change the default argument names, you must also bind the new names to their respective R-1/ROSS Build Activity property.

The following example shows how to change the **DeploymentMode** argument name and then bind the new name to its corresponding property. The same procedure should be followed for each of the **R1BuildActivity** arguments that you want to change.

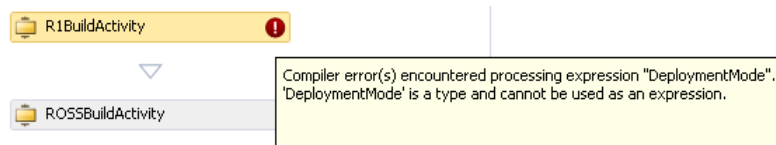
1. Click the **Arguments** button at the bottom of the build template and then double-click the argument name that you want to change.



2. Change the name and then press [Enter].

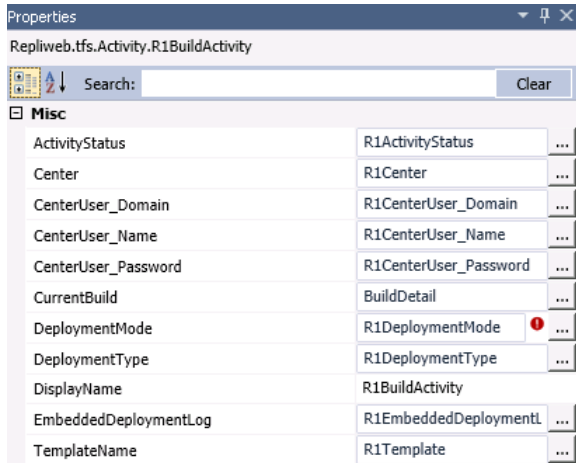


3. In the build template, scroll down to the **R1BuildActivity** activity.



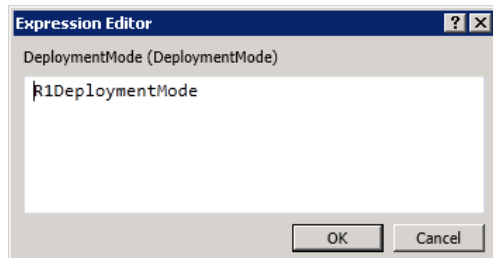
Note the exclamation mark which tells us that a problem has been encountered. The error message (revealed by hovering the mouse cursor over the exclamation mark) informs us that the problem is related to the argument name that we just changed. To resolve the problem, we need to map the new name to its corresponding property.

4. Click the **R1BuildActivity** activity to reveal the activity properties (note the exclamation mark next to the right of `DeploymentMode` argument).



- Click the browse button to the right of the **DeploymentMode** argument.

The Expression Editor opens.



- Enter the new argument name, in this case R1DeploymentMode, and then click **OK**.

The argument is now bound to its corresponding **R1BuildActivity** property (and all exclamation marks disappear).



IMPORTANT: Each of the argument names also has a display name (shown in the **Process** tab of the build definition settings). In order to preserve an argument's display name, you also need to update the actual argument name in the Process Parameter Metadata Editor. For further details, see

[Editing the R-1/ROSS Build Activity](#) Metadata.