

# TFS 2010 Beta2 to RC Upgrade Guide (v1.02)

---

## Contents

Revision History .....	2
Introduction.....	3
1: Upgrade paths (Which one do I follow?) .....	4
In-Place Upgrade .....	4
Migration Upgrade .....	4
Which path to choose? .....	5
Scenario 1: Testing Upgrade .....	5
Scenario 2: Production Upgrade .....	5
Scenario 3: Production Upgrade with quick fall back plan .....	5
2: Considerations before attempting upgrading .....	6
SharePoint Integration .....	6
SharePoint extensions configuration on SharePoint servers in a farm may fail .....	6
Detaching and attaching collections and upgrading .....	7
Upgrading Beta2 Clients .....	7
Visual Studio 2008 + GDR update.....	7
Turkish Collation bug.....	7
Team Foundation Server Proxy Does Not Work in the RC Release .....	8
Deleting Web Services cache directory .....	8
3: Additional steps required before upgrading to RC .....	9
1. Verifying current environment .....	9
2. Configure Reporting Database .....	9
3. Disable SQL Compression .....	9
4. Reporting – Handling partially configured TFS Reporting .....	9
5. Team Lab pre upgrade changes.....	10
6. Backup Test Controller configuration file .....	10
4: TFS Upgrade.....	11
1. Ensure you have a backup of existing SQL databases (TFS, Sharepoint and Reporting) .....	11
2. Uninstall Team Foundation Server 2010 Beta2 components (in-place upgrade).....	11
3. If using a separate SharePoint farm uninstall TFS 2010 Beta2 extensions and install TFS 2010 RC extensions	

4.	Run the TFS 2010 RC setup and select the <i>Upgrade from Previous Version Wizard</i> .....	11
5.	Uninstall and reinstall the additional components located on separate machines .....	11
5:	Additional steps required after upgrade to RC finishes .....	12
1.	Team Lab post upgrade changes .....	12
2.	Updating existing project collections to use new RC process templates .....	12
3.	Updating existing project collections to use new Build templates .....	12
4.	Re-apply Filters on Reports and Excel workbooks.....	12
6:	Troubleshooting Upgrade Failures .....	14
Appendix:	.....	15

## Revision History

Version	Change	Date
1.00	Initial publication	02/03/2010
1.01	Addition of SharePoint extensions configuration note in Section 2	02/09/2010
1.02	Addition of guidance to clear cache directory prior to running upgrade	02/17/2010

# Introduction

While every attempt has been made to ensure a smooth upgrade process from previous versions to Microsoft Team Foundation Server 2010, there are some specific action items you should follow when upgrading from Beta2 to the Release Candidate (RC). Please read this document before starting the upgrade to the RC. Note that some of the steps in this document are conditional based on the Beta2 configuration you are currently running.

The document is divided into six sections:

1. Choosing an upgrade path – since some of the steps only apply to “Migration” upgrades, we feel it important to define the two types of upgrade paths available to you and some scenarios for when you would select one path over the other.
2. Considerations before attempting upgrading – PLEASE READ.
3. Additional steps required BEFORE upgrading to RC.
4. Upgrading to RC.
5. Additional steps required AFTER upgrading to RC.
6. Troubleshooting Upgrade failures.

DRAFT

# 1: Upgrade paths (Which one do I follow?)

As in upgrading from TFS 2005 or TFS 2008 there are two paths to choose from when upgrading from Beta2 to RC: “In-Place” upgrade and “Migration” upgrade.

## In-Place Upgrade

An In-Place upgrade is defined as an upgrade where you use the same set of hardware that is running the current TFS version. In this scenario you will be uninstalling the current TFS version and then installing TFS and running the Upgrade Wizard.

### Examples:

1. Single Server - you are currently running TFS2010 Beta2 on a single server with SQL Server 2008, SharePoint 2007 and Reporting Services/Analysis Services components. You would uninstall TFS and then install/upgrade TFS to TFS 2010.
2. Multi Server - you are using separate servers for TFS, SQL Server and SharePoint. You would uninstall TFS on the TFS Server and then install/upgrade TFS to TFS 2010 on that same server.

### High level upgrade steps (See the TFS Install guide for more details):

1. Backup existing SQL databases
2. Uninstall Team Foundation Server components using Add/Remove Programs
3. If using a separate SharePoint farm, uninstall existing Team Foundation Server Extensions for WSS and install TFS 2010 Extensions for WSS
4. Run the TFS 2010 setup and select the *Upgrade from Previous Version Wizard*.

## Migration Upgrade

A Migration upgrade is defined as an upgrade where you using a separate duplicate set of hardware to perform the upgrade. In this scenario you will be installing the pre-req components, copying the existing TFS related databases over and then installing TFS and running the Upgrade Wizard.

### Examples:

1. Single Server - you are currently running TFS on a single server where all components are on this same server. For a migration upgrade you would set up a new separate server with the pre-req components (SQL Server, SharePoint, AS/RS), back up the TFS, SharePoint and Reporting databases from the existing server, restore them on the new server and then run the install/upgrade of TFS to TFS 2010.
2. Multi Server – you are using separate servers for TFS, SQL Server and SharePoint. You would setup new servers for each separate component, backup the appropriate databases, restore them to the new servers and then run the install/upgrade of TFS to TFS 2010.

### High level upgrade steps (See the TFS Install guide for more details):

1. Backup existing SQL databases (TFS, SharePoint content, Warehouse and Reporting databases)
2. On new hardware, install necessary TFS pre-reqs including OS, IIS, SQL Server 2008, SQL Reporting Services and Analysis Services and Windows SharePoint Services.
3. Copy over backups from step #1 and restore databases on new hardware. NOTE: Be sure to restore them with the same file names they had on the original server.
4. Configure SharePoint and SQL Reporting to connect and use these restored databases
5. If using a separate SharePoint farm, uninstall existing Team Foundation Server Extensions for WSS and install TFS 2010 Extensions for WSS
6. Run the TFS 2010 setup and select the *Upgrade from Previous Version Wizard*.

## Which path to choose?

There are essentially three scenarios to choose from which will determine which upgrade path you follow:

### Scenario 1: Testing Upgrade

In this scenario you want to test upgrade on a separate set of hardware while leaving the existing system running. For this scenario you follow the **Migration Upgrade path**. In addition you will also need to change the TFS database "stamp" on this new upgraded instance. All TFS instances have a GUID that identifies them (regardless of what url you use to access them). Your clients will get confused if there are two TFS instances with the same GUID. To change the TFS GUID follow these commands:

1. Open a cmd window as admin on the AT
2. Change to the directory: "%programfiles%\Microsoft Team Foundation Server 2010\Tools" and run the following commands.
  - a. iisreset /stop
  - b. tfsconfig changeserverid /sqlinstance:<dataTierName> /databasename:Tfs\_Configuration
  - c. tfsconfig registerdb /sqlinstance:<dataTierName> /databaseName:Tfs\_Configuration
  - d. iisreset /start
  - e. net start tfsjobagent

You may also want to disable SharePoint and Reporting Services on the trial upgrade (using either the upgrade wizard to uncheck the options or from the TFS Admin Console after the fact), OR you could choose to copy the SharePoint and Reporting Services databases over to the trial instance before upgrade.

### Scenario 2: Production Upgrade

In this scenario you are ready to upgrade your existing production system and plan to use the existing hardware. For this scenario you follow the **In-Place Upgrade path**. (Note that in this scenario that because you have to uninstall the existing version of TFS and the TFS 2010 upgrade is changing the existing data, "failing back" to that previous version of TFS will involve restoring the TFS databases from backup and re-installing the previous TFS version.)

### Scenario 3: Production Upgrade with quick fall back plan

In this scenario you are going to move to new hardware at the same time as upgrading your production system, AND you will keep the existing version of TFS running on the old hardware until the upgrade completes successfully. For this scenario you follow the **Migration Upgrade path**. (Note that the benefit of this scenario is that you can "fall back" and resume using the previous version of TFS on the old hardware if the upgrade fails.)

## 2: Considerations before attempting upgrading

This section includes some late breaking information that is IMPORTANT to read prior to upgrading from Beta2.

### SharePoint Integration

You may run into an issue during upgrade if you configured SharePoint integration with Team Foundation Server Beta2 using non-default values. The following scenarios are affected by this issue:

- You do not have SharePoint installed on the same server as TFS.
- OR
- You do have SharePoint installed on the same server as TFS and one or more of the following is true:
    - Your SharePoint web application was configured on a port other than 80.
    - Your SharePoint Central Administration site was configured on a port other than 17012.
    - You set up SharePoint to use https rather than http.

If you fall into at least one of the above categories and you select “Use existing SharePoint settings” during upgrade, you will encounter an error message during the verification step similar to the following:

“TF250020: The following SharePoint Web application is not valid: <http://YourSharePointURL/>. Verify that you have the correct URL.”

If your SharePoint web application was configured on a port other than 80, but your SharePoint Central Administration site was configured on port 17012 and you are using http rather than https, then you can work around this error by creating a SharePoint web application on port 80 (the verification step is erroneously looking for a SharePoint site on port 80).

You can also change your SharePoint Central Administration port by running “stsadm -o setadminport -port 17012”.

Otherwise, to work around this issue, you must uncheck the “Configure SharePoint” checkbox during upgrade. If you do so, you will have to manually hook up your existing SharePoint integration points to your team project collections and team projects after upgrading. This involves:

- Manually entering in your SharePoint web application and central administration site details in the TFS Administration Console.
- Manually entering in the corresponding SharePoint site details for each team project collection in the TFS Administration Console.
- Manually entering in the corresponding SharePoint site details for each team project in the TFS Administration Console.”

### SharePoint extensions configuration on SharePoint servers in a farm may fail

Installation will succeed, but the SharePoint extensions configuration wizard may fail with the following error:

[Error @12:34:34.204] !Verify Error!: Windows SharePoint Services 3.0 is configured to be installed but the following URL refers to a remote computer: <http://servername/>. The installation wizard can only install Windows SharePoint Services 3.0 on the same computer as Team Foundation Server.

This is a bug in the RC and will be fixed in RTM, however to workaround this issue please follow these steps:

- Install the extensions, but don't configure it. (Or, if you did configure it, and it gave you the error, ignore it).

- Open regedit. Go to [HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Shared Tools\Web Server Extensions\12.0\WSS]
- Take note of the existing setting for "CentralAdministrationURL" as you'll need to set it back to that after the TFS SP extensions are configured.
- Modify the setting for "CentralAdministrationURL" to include the local machine name instead of the actual host name.
- Run "TFSMGMT.exe Configure".
- Click on the "Extensions for SharePoint Products" node.
- Click on the "Configure Installed Features" link.
- Run the Extensions for SharePoint Products configuration wizard.
- Click Configure.
- When the configuration is complete, don't click the "Close" button yet.
- Go back to regedit and restore the original value of "CentralAdministrationURL" under [HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Shared Tools\Web Server Extensions\12.0\WSS].
- Go back to the Extensions for SharePoint Products wizard and click Close.
- The Team Foundation Admin Console should start and be ready to use.

### Detaching and attaching collections and upgrading

In order for a project collection to be successfully upgraded from Beta2 to RC the collection(s) must be "attached" and on-line as a Beta 2 collection(s) prior to the upgrade. "Detached" Beta2 collections cannot be "attached" to an RC or RTM instance and cannot be upgraded separately from the rest of the TFS instance. The Attach/Detach command is used for topology changes and moving of collections and is not used during the upgrade process. Please see the TFS Administration guide for more information on the Attach/Detach command.

### Upgrading Beta2 Clients

You need to upgrade Visual Studio Beta2 clients to the RC version. We do not officially support using Beta2 clients against Team Foundation Server RC instances. Functions and features may or may not work correctly when using Beta2 clients against a RC server. Please see the Visual Studio RC documentation for information on upgrading clients.

### Visual Studio 2008 + GDR update

Visual Studio 2008 clients using the 2008 Forward Compat Update (GDR) works with the RC server.

### Turkish Collation bug

If your Beta 2 configuration of TFS used SQL Server with a Turkish collation, users were unable to execute filtered and/or sorted tree queries of work items. In RC, the scope of this bug has increased to include any type of tree query. Thus, if you upgrade from Beta 2 to RC in this scenario, your users will not be able to execute any type of tree query of work items. This will affect the following scenarios:

- Executing tree queries from within Visual Studio, Team Web Access, and Microsoft Excel. This includes the tree queries that are shipped in the out-of-the box Agile and CMMI templates, such as Iteration Backlog queries.
- Exporting work items to Microsoft Project.
- Getting work items in Microsoft Project.

This bug has been fixed for RTM.

## Team Foundation Server Proxy Does Not Work in the RC Release

In the RC release, Team Foundation Server Proxy will not service download requests from users who connect to it. Users who attempt to use the proxy will get an http 500 or 404 error message, and all requests will be directed to the main instance of Team Foundation Server. Users will also see messages in the console output or Visual Studio output window indicating the proxy was not used.

Use Team Foundation Server Proxy from the Beta 2 milestone or Team Foundation Server Proxy from the previous release (Visual Studio Team System 2008 Team Foundation Server).

## Deleting Web Services cache directory

We have found that upgrades can take a long time when the TFS Web Services cache directory contains many objects. You can empty the following cache directory prior to upgrading your system ([\Program Files\Microsoft Team Foundation Server 2010\Application Tier\Web Services\\\_tfs\\_data](#)) which will shorten the upgrade duration. However, it is not possible to predict how much this will shorten the upgrade duration

DRAFT

# 3: Additional steps required before upgrading to RC

## 1. Verifying current environment

- Verify that all active environments are running properly prior to starting the upgrade process. This includes Team Foundation Server, Proxy, Build and Test integration components.
- It may be helpful to have a list of your environment including machine names, accounts for all your components

## 2. Configure Reporting Database - If you are doing a migration upgrade (backing up SQL databases and restoring them to another server) **and you are using TFS Reporting** you need to also restore the ReportServer and ReportServerTempDB databases and then use SQL Server *Reporting Services Configuration Manager* to configure Reporting Services to use these restored databases.

- Use SQL Server Management Studio to restore the ReportServer and ReportServerTempDB to your destination server
- Start SQL Server Reporting Services Configuration Manager from Start->Microsoft SQL Server 2008->Configuration Tools menu
- Connect to the report server instance
- Select the Database node and click on "Change Database"
- Follow the Report Server Database Configuration Wizard and select the "Choose an existing report server database" option
- Point to the restored ReportServer database and complete the wizard inputs

## 3. Disable SQL Compression - If you are doing a migration upgrade (backing up SQL databases and restoring them to another server) **and moving from SQL 2008 Enterprise Edition to SQL 2008 Express Edition** as part of the Beta2 to RC upgrade you need to run the following SQL script to disable SQL compression as it isn't supported in SQL Express.

Note this script needs to be run before backing up the Beta2 databases on the server running SQL Enterprise.

- Run the "Disable SQL Compression" SQL script in the appendix of this document on **each** TFS project collection database the TFS Beta2 SQL Server
- Backup the TFS Beta2 databases (Tfs\_Configuration and Tfs project collection databases)
- After you have successfully backed up the TFS Beta2 databases you may turn on SQL compression again if you want to continue running the existing Beta2 server by executing the stored procedure created above with different value:  
`exec prc_EnablePrefixCompression @online = 0, @disable = 0`
- Restore the backed up TFS databases onto the server running SQL 2008 Express Edition.

## 4. Reporting – Handling partially configured TFS Reporting – If you have partially configured reporting in Beta2 you must either fully configure or un-configure reporting before upgrade.

- If you have not fully configured all three reporting components (Warehouse database, Analysis Services database and Report Server) via the Reporting node in the TFS Administrative Console you will need to either complete the configuration of all three components or un-configure all three components prior to starting the upgrade.

- You can select the 'Edit' button on the Reporting node to change the configuration.
5. **Team Lab pre upgrade changes** – If you had *configured Team Lab during Beta1 and then upgraded to Beta2* please see the Team Lab Management [blog](#) for additional information on upgrading Team Lab.
  6. **Backup Test Controller configuration file** - If you using Team Lab you should back up the Test Controller configuration file. You can restore this file after the upgrade completes and after you have installed the RC Test Controller to preserve the Team Lab configuration. (See section below for installing the Test Controller and restoring the configuration file)
    - Take a backup of the XML file C:\Program Files\Microsoft\Visual Studio 10.0\Common7\IDE\QTControllerConfig.xml
    - In the case of a 64 bit machine the location will be C:\Program Files (x86)\Microsoft\Visual Studio 10.0\Common7\IDE\QTControllerConfig.xml.

DRAFT

## 4: TFS Upgrade

1. **Ensure you have a backup of existing SQL databases (TFS, Sharepoint and Reporting)**
2. **Uninstall Team Foundation Server 2010 Beta2 components (in-place upgrade)**
  - Use Add/Remove programs from Control Panel to uninstall TFS Components via “Microsoft Team Foundation Server 2010”
  - Select “Uninstall” to remove all TFS components from each server where Beta2
  - NOTE: You may see a program in the Control Panel called “Microsoft Team Foundation Server 2010 Object Model”. This is an object model used by products connecting to TFS (e.g. Visual Studio and Team Lab). This component will be uninstalled or updated by the installation program for those products. You should not uninstall this component during the TFS 2010 upgrade process.
3. **If using a separate SharePoint farm uninstall TFS 2010 Beta2 extensions and install TFS 2010 RC extensions**
4. **Run the TFS 2010 RC setup and select the [Upgrade from Previous Version Wizard](#)**
5. **Uninstall and reinstall the additional components located on separate machines**
  - Team Foundation Build Services from Control Panel via “Microsoft Team Foundation Server 2010”
  - Test Agent/Controller from Control Panel via “Microsoft Visual Studio Test Controller 2010”
  - Test Controller (if you are using Team Lab):
    - Take a backup of the XML file C:\Program Files\Microsoft\Visual Studio 10.0\Common7\IDE\QTControllerConfig.xml (in the case of a 64 bit machine the location will be C:\Program Files (x86)\Microsoft\Visual Studio 10.0\Common7\IDE\QTControllerConfig.xml. This will ensure that you can go back to the original state if you want to rollback the upgrade. Alternatively, if the controller is running on a VM, take a snapshot before you continue.
    - Uninstall the Beta2 Test controller and Install RC Test controller in the same folder, so it will use the old settings.
    - Before configuring the test controller and connecting it to TFS either:
      - Copy the QTControllerConfig.xml and override the file that came from the installation.
      - If you are trying to use a new machine for the new controller, use the same machine name. Otherwise you will need to go to all the test agents that were connected to this controller. For Lab Environment, you can stop them and edit the controller in the capabilities dialog, and then start the LE again and Lab will configure the agents automatically for you.
    - Walk through the default options during Install. Configure the controller with the upgraded TFS.

## 5: Additional steps required after upgrade to RC finishes

1. **Team Lab post upgrade changes** – If you had *configured Team Lab during Beta1 and then upgraded to Beta2* please see the Team Lab Management [blog](#) for additional information on upgrading Team Lab.
2. **Updating existing project collections to use new RC process templates** – The RC process templates will not be provisioned to existing project collections during upgrade. Follow the steps below to provision the RC process templates to existing project collections:
  1. Create a new team project collection
  2. Connect to the project collection and download the RC process templates using the Process Template Manager
  3. For each existing team project collection, connect and upload the RC process templates using the Process Template Manager.
3. **Updating existing project collections to use new Build templates** – When upgrading from Beta 2 to RC/RTM, your build definitions may have issues due to changes introduced into the default (or upgrade) build process templates between the Beta 2 and RC/RTM releases. Replace (or reconcile) your Beta 2 build process templates with those that ship in the RC/RTM release. You can do this by creating a new team project and copying the build process templates into your upgraded team project. If you've made customizations to your build process templates, you'll want to merge the RC/RTM build process templates with your customized versions.
4. **Re-apply Filters on Reports and Excel workbooks** - Some filters in reports and Excel workbooks may have to be re-applied after upgrading from Beta 2 to RC.

Reports using Area Hierarchy, Iteration Hierarchy, Work Item Tree Hierarchy, Version Control File Hierarchy, Test Suite Hierarchy, Build and Build Source Project File Hierarchy as a filter on a report filtering on specific values will need to be re-applied after upgrade.

- . The keys associated with these dimensions and hierarchies in the cube are modified during upgrade, so filters will not find the specific members that are selected and will get reset to (All). These filters will need to be applied again.
- a. The following Excel pivot table report is an example where the Area filter, highlighted in red, will need to be re-applied after upgrade:

	A	B
1	Team Project Hierarchy	Dev10
2	Work Item.Work Item Type	Dev10 Bug
3	Work Item.State	Active
4	Work Item.Sub Status	(Multiple Items)
5	Work Item.Area Hierarchy	Reporting Platform
6		
7	<b>Work Item Count</b>	
8		20

DRAFT

## 6: Troubleshooting Upgrade Failures

If upgrade fails, please see the detailed log files generated during the upgrade process. The “Complete” screen will contain links to the log files.

After common (disk space capacity limits, network timeouts) upgrade failures are resolved, you can attempt to restart the upgrade by launching the TFS Administration Console, navigating to the Team Project Collections node and then to the Status tab and then clicking on the “Rerun Job” link.

DRAFT

# Appendix:

## Disable SQL Compression script:

```
-- first query if there are any entries using database compression
-- this is the 'before' query
select *

from sys.partitions

where data_compression_desc = 'PAGE'
go

-- next drop and replace the sproc to disable compression on a beta1 DB
IF EXISTS (SELECT * FROM sysobjects WHERE type = 'P' AND name = 'prc_EnablePrefixCompression')
BEGIN
    DROP PROCEDURE prc_EnablePrefixCompression
END
GO

PRINT 'Creating Procedure prc_EnablePrefixCompression'
GO

CREATE PROCEDURE prc_EnablePrefixCompression
    @online BIT,
    @disable BIT = 0

AS

SET NOCOUNT ON

DECLARE @targetCompression NVARCHAR(60)
DECLARE @tableName SYSNAME
DECLARE @indexName SYSNAME
DECLARE @sql NVARCHAR(MAX)

DECLARE @status INT
DECLARE @rowCount INT

DECLARE @indexes TABLE (
    TableName SYSNAME,
    IndexName SYSNAME
)

-- Initialize the ProcedureName for error messages.
DECLARE @procedureName SYSNAME
SELECT @procedureName = @@SERVERNAME + '..' + db_name() + '..' + object_name(@@PROCID)

IF (CONVERT(VARCHAR(MAX), SERVERPROPERTY('Edition')) NOT LIKE '%Enterprise%'
    AND CONVERT(VARCHAR(MAX), SERVERPROPERTY('Edition')) NOT LIKE '%Developer%')
BEGIN
    PRINT 'Prefix compression not enabled as not running the right edition.'
    RETURN
END

INSERT @indexes (TableName, IndexName)
VALUES (N'tbl_Branch', N'PK_tbl_Branch'),
    (N'tbl_BranchMapping', N'PK_tbl_BranchMapping'),
    (N'tbl_BranchMapping', N'IX_tbl_BranchMapping_ServerItem'),
    (N'tbl_LocalVersion', N'PK_tbl_LocalVersion'),
    (N'tbl_LocalVersion', N'IX_tbl_LocalVersion_WorkspaceId_LocalParent_Child'),
    (N'tbl_Lock', N'PK_tbl_Lock'),
```

```

(N'tbl_PendingChange', N'PK_tbl_PendingChange'),
(N'tbl_PendingChange', N'IX_tbl_PendingChange_TargetParentPath_ChildItem'),
(N'tbl_PendingChangeRecursive', N'PK_tbl_PendingChangeRecursive'),
(N'tbl_PendingChangeRecursive', N'IX_tbl_PendingChangeRecursive_WorkspaceId_TargetParentPath_TargetChildItem'),
(N'tbl_PendingChangeRecursive', N'IX_tbl_PendingChangeRecursive_SourceParentPath_SourceChildItem'),
(N'tbl_PendingMerge', N'PK_tbl_PendingMerge'),
(N'tbl_PendingRollback', N'PK_tbl_PendingRollback'),
(N'tbl_Version', N'PK_tbl_Version'),
(N'tbl_Version', N'IX_tbl_Version_ItemId_VersionFrom'),
(N'tbl_Version', N'IX_tbl_Version_VersionFrom'),
(N'tbl_VersionedItem', N'PK_tbl_VersionedItem'),
(N'tbl_VersionedItem', N'IX_tbl_VersionedItem_ParentPath_ChildItem'),
(N'tbl_WorkingFolder', N'PK_tbl_WorkingFolder'),
(N'tbl_WorkspaceMapping', N'PK_tbl_WorkspaceMapping')

SET @status = @@ERROR

IF (@status <> 0)
BEGIN
    RAISERROR (500004, 16, -1, @procedureName, @status, N'INSERT', N'@indexes')
    RETURN 500004
END

IF (@disable = 0)
BEGIN
    SET @targetCompression = N'PAGE'
END
ELSE
BEGIN
    SET @targetCompression = N'NONE'
END

DECLARE IndexCursor CURSOR LOCAL FAST_FORWARD FOR
SELECT i.TableName,
       si.name
FROM   @indexes i
JOIN   sys.indexes si
ON     si.object_id = OBJECT_ID(i.TableName)
       AND si.name = i.IndexName
WHERE  EXISTS (
        SELECT *
        FROM  sys.partitions sp
        WHERE sp.object_id = si.object_id
              AND sp.index_id = si.index_id
              AND sp.data_compression_desc <> @targetCompression
      )

OPEN IndexCursor
FETCH NEXT FROM IndexCursor INTO @tableName, @indexName

WHILE (@@FETCH_STATUS = 0)
BEGIN
    SET @sql = 'ALTER INDEX ' + QUOTENAME(@indexName) + ' ON ' + QUOTENAME(@tableName) + ' REBUILD '

    SET @sql = @sql + N' WITH (DATA_COMPRESSION=' + @targetCompression

    IF (@online = 1)
    BEGIN
        SET @sql = @sql + ', ONLINE=ON'
    END

    SET @sql = @sql + ')'

```

```

EXEC @status = sp_executesql @sql

IF (@status <> 0)
BEGIN
    RAISERROR (500004, 16, -1, @procedureName, @status, N'REBUILD INDEX', @indexName)
    RETURN 500004
END

    FETCH NEXT FROM IndexCursor INTO @tableName, @indexName
END

CLOSE IndexCursor
DEALLOCATE IndexCursor

RETURN 0
GO

GRANT EXEC ON prc_EnablePrefixCompression TO TFSEXECROLE
GO

-- Now run the new sproc
exec prc_EnablePrefixCompression @online = 0, @disable =1
go

-- Alter the one entry that gets dropped from Beta1->Beta2 upgrade and did not get fixed by the above sproc
ALTER INDEX IX_tbl_Lock_LockStatus_FullPath ON tbl_Lock REBUILD WITH (DATA_COMPRESSION=NONE)
go

-- Finally run the query again to confirm there are no more entries using data compress.
-- This is the 'after' query
select *

from sys.partitions

where data_compression_desc = 'PAGE'
go

```