TFS 2013 Migration Guidance – Parking and Public Safety

Purpose

This document is provided to give you more in-depth details regarding the migration of the cloud hosted-TFS server to our on-premise TFS 2013 server. This will start **Friday, May 2**. This migration is an overall initiative to eventually consolidate all existing TFS servers across MVSS into one consolidated enterprise solution.

Update

During the first migration attempt, there were no full service Microsoft tools available for us to export the existing service-hosted TFS projects to an on-premise server. This tool is now available and will be used to migrate the entire Team Project Collections for both the Parking and Public Safety groups. The good news is that this migration should maintain complete data integrity – unlike the previous migration attempts where several artifacts, e.g. test artifacts and build results, were not able to be migrated. As stated before, on any migration on any system, nothing is perfect. This Microsoft migration tool is the one and only release that will be available for us to move the TFS projects. It has been tested by Microsoft, and we have their full support for the data integrity (unlike other the Integration Tools and custom tooling). Our time window is limited to May 7, so if there are issues, and the migration must be rolled back, the next attempt will happen quickly. During the migration, we will try to minimize the amount of interruption to your work.

Overall Schedule

**Start**: The migration will start on **Friday, May 2 at 7:00pm CST**. Please excuse that we are providing you this information at the very last moment. Here are the main action items for you before the migration, and some of these will be explained in greater detail later in the document:

* + All pending changes must be either checked in, or shelved in TFS. Unless you will cause a major break or instability in your application(s), go ahead and check in or undo any pending check-outs in version control that you may have.
  + Even if you are somehow able to get into the old system during the migration, PLEASE do NOT touch anything. We will do our best to keep everyone out, but there will be times when the system will be up. If you need or forgot to do something, please contact **Everett Taylor** immediately. He *may* be able retrieve it for you during the migration window in case of emergency.

**Finish**: Our migration window will close at **Monday, May 5 at 5:00am CST**. The new environment will be back online and ready for you to use. We will broadcast emails along the way to let you know the status, and where to go to connect when you come back on Monday.

What to Expect

Here are the high level migration steps:

1. Prepare version control for migration. This will include you performing an undo check-out on non-essential work in version control on the old system. More on this later.
2. Migrate the entire Team Project collections from the online hosted service to the on-premise server.
3. Validate imported data.
4. Contact necessary team members to verify migration results.
5. Notify all users of new connection location.
6. Open system for access.

**If the migration fails:**

In the event that something catastrophic happens during the move and prevents us from completing the migration, you will be redirected back to old system, and we will do emergency overnight attempts again on Monday and Tuesday night.

Version control

**Pending changes:**

**ALL** pending changes have to be released before the migration. **THIS IS CRITICAL**. You will have to recreate a new workspace linked to the on-premise server, so any pending work you have will be LOST when you delete the old workspace on your workstation. If you would cause a catastrophic break, or destabilize the application code, shelve your pending changes. After shelving, undo your pending changes to release the locks in version control (see more on this below). In the event that you forget to check in or shelve something, you will still have a local copy on your workstation UNTIL YOU DELETE THE OLD WORKSPACE, which you will be doing first thing on Monday during setup of your new workspace linked to the new server. If you would like, copy the local files to a different folder just for safekeeping BEFORE destroying the old workspace. When you delete the old workspace, it will delete all the local files before you map the new workspace to the same local folder.

Another shortcut for small changes that you have made could be to comment out whatever code changes that would break the application when you check in:

1. Comment out breaking changes.
2. Check in the files.
3. After the migration, check the files back out to your new workspace and uncomment your changes.

**Shelving:**

The complete migration of the team project collection also means that shelvesets will be preserved in version control. Shelving is the best option for preserving pending changes that would break your application if you checked in. After you shelve, remember to undo pending changes on your files to release the file locks on the server side.

For detailed information on how to use shelving, see: <http://msdn.microsoft.com/en-us/library/ms181403(v=vs.110).aspx> . These instructions vary greatly depending on which version of Visual Studio you are using (see the versions at the top of the page), but just remember to look for an option to preserve pending changes locally, and DO NOT preserve locally. In older versions, you have to manually Undo Pending Changes on your files after shelving.

**More on source control binding:**

Some Visual Studio files, like solution files, have source control bindings embedded in the files. ***Most*** of the time, the bindings will update to point to the new server connection information. But this is not always the case. For more information on this, see the basics on binding: <http://msdn.microsoft.com/en-us/library/0eh3790h(v=vs.90).aspx>

There is some new intelligence in the later versions of Visual Studio where it will detect that it cannot connect to the bound version control server, and will give you an opportunity to change the binding to the new server. You will see a dialog where you can change the server path (pay attention to the server paths, look at the How to Connect information later in the document for the server binding path). Otherwise, you may have to look under File > Source Control > Change Source Control to reestablish the bindings. Contact [Everett Taylor](mailto:etaylor@mmm.com) if you have issues with solution or project bindings.

Reports

Good news. The work items in the project will be pushed into the on premise OLAP data warehouse periodically, and there are default reports (like bugs, tests, etc.) that will available to you. For more information on these reports, see: <http://msdn.microsoft.com/en-us/library/dd380647.aspx>

Dashboards

Initially, the SharePoint dashboards (you may have seen these mentioned in the previous MSDN documentation will not be available. A SharePoint instance will put into place later, and these will become available.

Testing

**Test plans, suites, and cases:**

All test artifacts should be intact post-migration. This includes all including test plans, test suites, test cases, test runs and results.

Builds

**Builds results:**

All build results should be intact post-migration.

**Build definitions:**

The complete build definitions cannot be migrated (again, see the previous section about build differences). **However**, if you have .xaml build process templates or custom build activity code, we will reuse as much as possible. The DevOps team will be assisting with creating new build definitions to match the old behavior, and in some cases, we may already have more advanced or elegant tooling available to use for common activities like assembly versioning or deployment. Also, there is a large file share that we use for build drops, so more disk space will be available for longer build retention policies.

**Build servers:**

New dedicated build servers for Parking and Public Safety have been built and are ready. We will work with you when the server is ready to get the build definitions and automated builds back into place as soon as possible to minimize interruption.

Post-migration steps for you

**How to connect:**

The new server is located at <http://tfs13.archon-tech.com:8080/tfs>. This is not a hyperlink, this is the URL that you will need to point Visual Studio and Microsoft Test Manager to in order to access the new server. First, you will configure a combination of Credentials Manager and Internet Explorer settings to prevent authentication prompting for your archon-tech domain account:

1. Go to Control Panel > Credentials Manager

1. Add a Windows Credential:



1. Add a Credential like this:



1. Open Internet Explorer, go to Tools > Internet Options (how you get there depends on the version of browser (the gear here):





1. Select the Security tab, then select Local Intranet, then click Sites:



1. Click Advanced:



1. Add an entry for all archon-tech servers, this should keep it from prompting you and will go back to Credential Manager to pull the user and password, make sure the "Require server verification checkbox is turned off":



1. Click Close, Ok, Ok.
2. Test by going to <http://tfs13.archon-tech.com:8080/tfs/web>
3. It shouldn’t prompt you.

**NOTE: If you change your archon-tech password (via Time Reporting, etc.), make sure to go back to Credentials Manager and UPDATE YOUR PASSWORD.**

Your Team Project Collection will be either Parking or Public Safety, and the Team Project will be the same project that you were currently working on. For basic instructions on how to connect to a new server, see: <http://www.youtube.com/watch?v=X4BaJ_BUjSw>

**Firewall sessions:**

For access to some archon-tech servers (TFS 2013 in this case), a session must be opened through the firewall by going to <https://fwtx-proc-a.mmm.com:1443/netaccess/loginuser.html>. Use your ‘A’ account to open a new session in the firewall. We have configured the firewall to allow for port 8080 to allow Visual Studio communication through the firewall without a session, but there have been some recent issues with the firewall, so you may need to open a session. If you have issues with this, please submit a Salesforce ticket to Operations at: <https://www.clicktools.com/survey?iv=30guqed2bwa2w>

**Web access:**

To access to web interface for the new instance, go to <http://tfs13.archon-tech.com:8080/tfs/web>

There are a couple of feature differences between the service hosted TFS instance and the on-premise, but after conversations with some of the end users, the probability that you will notice is extremely low.

**Down-level IDE compatibility:**

For some previous development environments, like Visual Studio 2010 or 2012, there may be service packs or forward compatibility updates that will need to be applied to work correctly with the server. Also, for some of the new functionality (like Team Rooms, etc.), these may or may not be available for the desired IDE. For more detailed compatibility information, see: <http://msdn.microsoft.com/en-us/library/dd997788.aspx>

**Workspace Configuration:**

1. After the migration, you will not be able to connect to the old server. Remove the server from your list of available servers in Team Explorer.
2. Remove your old workspace and clear your version control cache:
   1. Open a command prompt, and go to C:\Program Files (x86)\[Visual Studio Version]\Common7\IDE. Here are the folders for each of the versions:
      1. Visual Studio 2008 – Microsoft Visual Studio 9.0
      2. Visual Studio 2010 – Microsoft Visual Studio 10.0
      3. Visual Studio 2012 – Microsoft Visual Studio 11.0
      4. Visual Studio 2013 – Microsoft Visual Studio 12.0
   2. Run: **tf workspaces**. Look for an entry for the old server that looks something like this (it will have visualstudio.com in the URL):

Collection: <https://mmm-parking.visualstudio.com/defaultcollection>

* 1. Run: **tf workspaces /remove:\* /collection:https://mmm-parking.visualstudio.com/defaultcollection** (the collection parameter should match your online-hosted instance, depending on which team you are on)
  2. Clear the files and folders out the old folder that you were using.
  3. Run tf workspaces again. Look for the entries to the old server. You should not see any.
  4. Open your version of Visual Studio, and depending on the version apply the latest update possible:

VS 2008 SP1: No direct Team Explorer support.  You have to use the MSSCCI Provider for TFS 2013. You can find it here: <http://visualstudiogallery.msdn.microsoft.com/06c8e056-7f77-4a5c-9b8b-49318c143df8>

VS 2010:   
Install SP1: <http://www.microsoft.com/en-us/download/details.aspx?id=23691>   
Apply Forward Compatibility GDR: <http://www.microsoft.com/en-us/download/details.aspx?id=29082>   
  
VS 2012:   
Requires latest Visual Studio 2012 update.   
  
For compatibility information for all previous versions of TFS, see:   
<http://msdn.microsoft.com/en-us/library/dd997788.aspx>

1. Set up a new workspace that is linked to the on-premise server. For information on this, see: <http://msdn.microsoft.com/en-us/library/cc138514(v=vs.110).aspx> . This information varies based on which version of Visual Studio, so select the correct version at the top of the page.

Corrections

In the event that you find errors in this document, please contact Everett Taylor so that corrections can be made and an updated document can be redistributed. The corrections should be highlighted in red text.

Additional Issues

If you encounter any severe issues, like missing work items or version control files, these should be reported to [Everett Taylor](mailto:etaylor@mmm.com) immediately. We will attempt to correct these as soon as possible. If Everett is not showing as Available on IM, email is the next best option. If you do not receive a timely response, email us-mvss-devops, and we will get back to you. For a final attempt, please call Everett Taylor at (214) 870-9888.