LINK Data Dispatcher



ETFS Security

Prepared for

3M

2/2/2015

Version 1 Draft

Prepared by

**Tony Feissle**

Senior Consultant

[tfeissle@micrososoft.com](mailto:tfeissle@micrososoft.com)

Contributors

**Louis Weinstein**

|  |  |
| --- | --- |
|  | Prepared for 3M |

Revision and Signoff Sheet

Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Change reference |
| 1/9/2015 | Tony Feissle | 1.0 | Draft for review/discussion |
|  |  |  |  |
|  |  |  |  |

Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Version approved | Position | Date |
| Louis Weinstein | 1.0 | Principal Consultant | 1/9/2015 |
| Michael O’Brien |  | Software Engineer Specialist |  |
|  |  |  |  |

Table of Contents

[1 3M Recommendations 3](#_Toc410667622)

[2 Permission reference for TFS 3](#_Toc410667623)

[2.1 TFS Systems 3](#_Toc410667624)

[Tips & Tricks 4](#_Toc410667625)

[2.2 Tools Used for TFS Permission Provisioning 5](#_Toc410667626)

[3 Built-In TFS Groups 7](#_Toc410667627)

[3.1 Allow, Deny, Not set and other permission states 8](#_Toc410667628)

[3.2 Proposal to manage Process Templates 10](#_Toc410667629)

[3.3 Server-level permissions 13](#_Toc410667630)

[3.4 Collection-level permissions 14](#_Toc410667631)

[3.5 Project, test and object-level permissions 15](#_Toc410667632)

[3.6 Release Management permissions 16](#_Toc410667633)

[3.7 3M Service Accounts 17](#_Toc410667634)

[4 Permission Extraction Tool 18](#_Toc410667635)

[Appendix A – References 19](#_Toc410667636)

1. 3M Recommendations

General security recommendations for the TFS environment @ 3M

* Each environment (Dev, QA and Prod) will have each separate setup accounts
* Each environment (Dev, QA and Prod) each have their own respective service accounts
* The Build Controllers and Build Servers have a build server account
* The TFS Setup account will be the only account allowed on the TFS Administrator Console
* There will be NO specific users assigned to TFS groups. Only Active Directory (AD) groups will be added to TFS groups
* Service Accounts passwords should be set to NOT expire – if this is not possible, it is recommended a credential manager be used which will warn when the service account is about to expire. The consequences of a service password expiration include a total shutdown of the TFS service.
* TFS Administrators (Server) will be the only group that can modify any of the templates.

The specific rationale and recommendations will be outlined in the rest of the document.

1. Permission reference for TFS

Permissions determine what tasks users can and can’t do. TFS has the capabilities to manage permissions by both users and groups. This guide will draw heavily on both MSDN (the official guidance) and ALM Ranger guidance (which while based on the official guidance, can offer a different perspective).

This paper will not discuss SharePoint or SRSS security provisioning except with the recommendation that the same Active Directory groups be used to provision these services as the comparable level of permissions within TFS.

* 1. TFS Systems

There are four major TFS related systems installed at 3M:

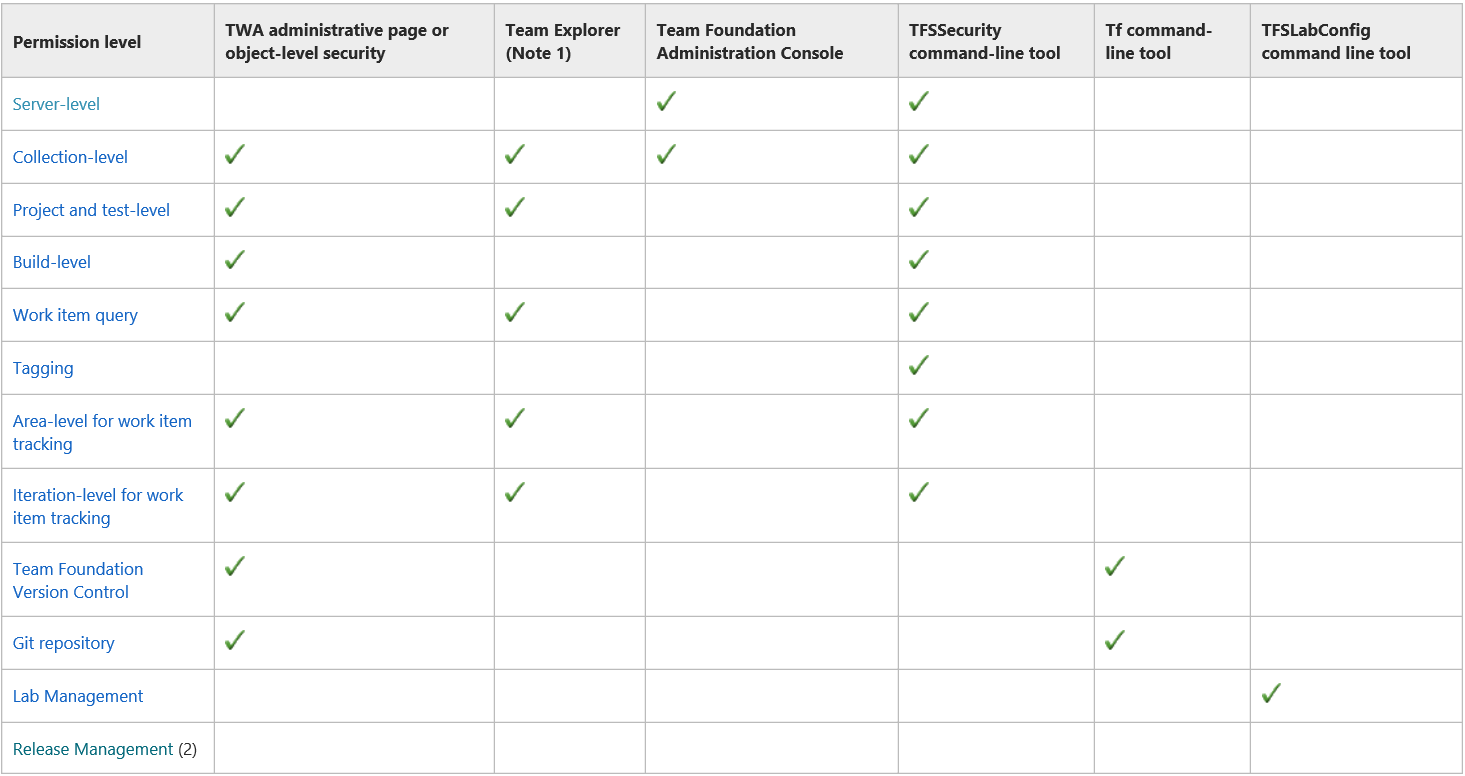
* SharePoint Foundation (for TFS) – this is **NOT** the Enterprise version
* Reports Services (SRSS) – for SQL Standard – there are no perspectives
* Team Foundation Services (TFS)
  + Interface with Windows through Team Explorer
  + Interface with Linux and OSX through Team Explorer Everywhere
* TFS Web Interface (Web Client)
* Release Management (interface via web and client applications)

It is strongly recommended that 3M use Active Directory (AD) groups to manage permissions amongst these three systems. Without AD groups, it becomes necessary to use a tool such as the [TFS Administration Tool](https://tfsadmin.codeplex.com/) (not recommended).

### Tips & Tricks

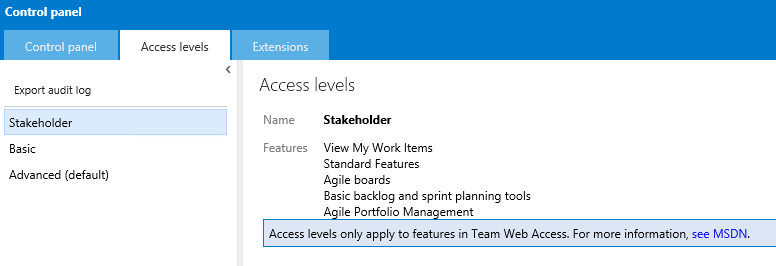
* Always use TFS Groups for permissions and contain the Active Directory groups within the respective TFS Group
* Change as few permissions as possible
* **Never** assign permissions to specific users
* **Do not allow end-users to change the process template willy-nilly (have a process!)**
* **Do not allow users to add other user ID’s directly (see the first point)**
  1. Tools Used for TFS Permission Provisioning

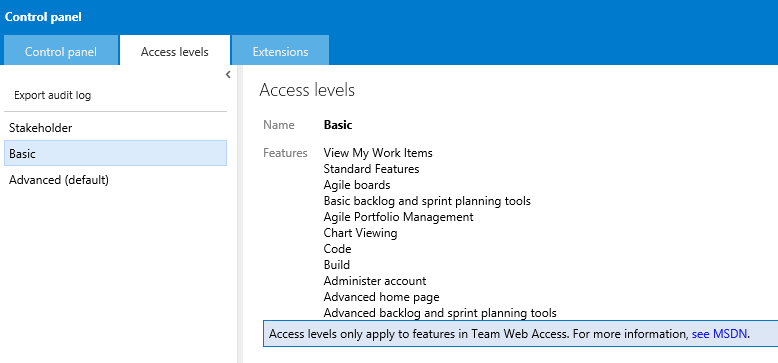
This table gives a good summary of how to provision TFS permissions:

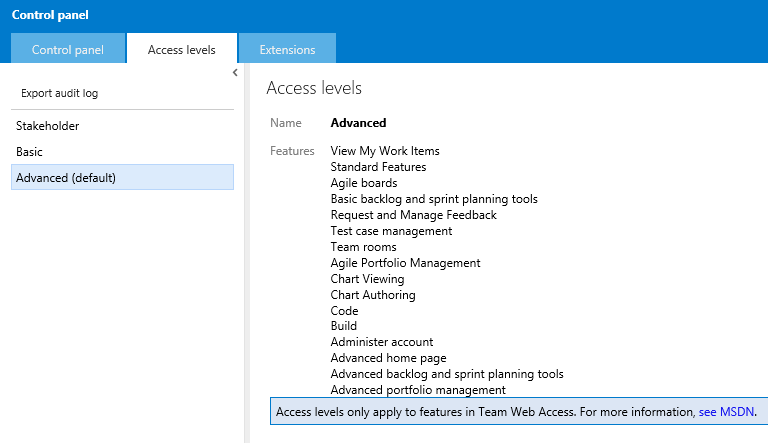


As you can see, there are six major tools used to manage the permissions of the various major parts of TFS. While this is not ideal, there is some overlap which allows the administrator the ability to choose their favorite tool.

Both the tf and tfssecurity tool are very powerful and they are accessed from the developer command prompt.







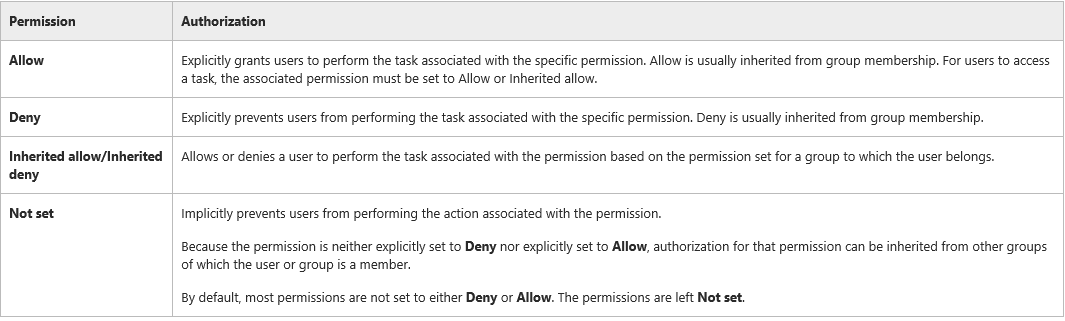
1. Built-In TFS Groups

<<need list of 3M groups>>

* 1. Allow, Deny, Not set and other permission states

TFS uses a least-permissive model for security permissions. What that means is that if a user belongs to two groups and the same permission is assigned Allow for one group and Deny for another group, Deny takes precedence over Allow. There are a few exceptions to this rule for those who belong to the Project Collection Administrator and Team Foundation Server Administrator groups.

You can specify two explicit authorization states for permissions: Deny and Allow. In addition, there are three other states: Inherited allow, Inherited deny, and Not set. Not set is an implicit Deny state.



*What this means for 3M:*

Do NOT set a Deny permission for one group and allow for another. Deny always takes precedence and this could cause unintended side-affects.

**Do’s:**

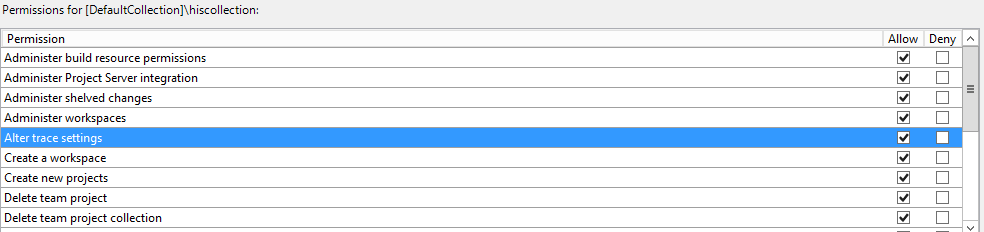
* Use Windows groups when managing lots of users.
* Consider granting the Contribute permissions to users or groups that require the ability to create and share work item queries for the project.
* When adding many teams, consider creating a Team Administrators group to TFS where you allocate a subset of the permissions available to Project Administrators.
* When adding teams, consider what permissions you want to assign to team leads, scrum masters, and other team members who may need to create and modify area paths, iteration paths, and queries.

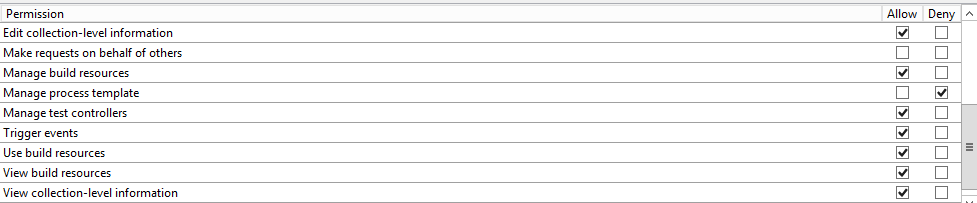
**Dont’s:**

* Don’t add accounts to the Readers group that you’ve added to the Project Administrators group. Doing so causes a Deny state to be assigned to several permissions.
* Don’t change the default assignments made to a valid users group. If you remove or set the View instance-level information permission to Deny for one of the Valid Users groups, no users in the group will be able to access the team project, collection, or deployment, depending on the group you set.
* Don’t assign permissions that are noted as ‘Assign only to service accounts’ to user accounts.
  1. Proposal to manage Process Templates

The key to making this security template work involves using Active Directory groups and controlling the “collection-level information”.

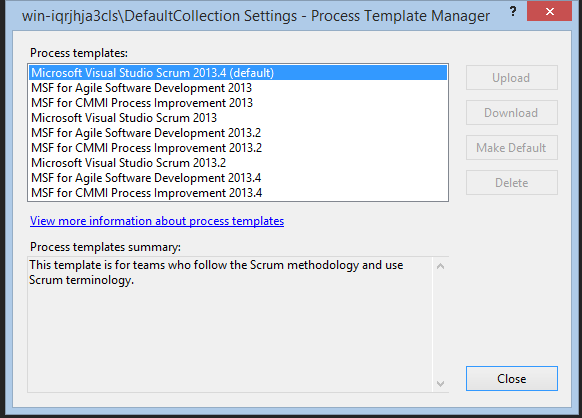
In TFS 2013, the recommended way is to setup a “HISCollection” group (for instance). This group would contain users whose sole ability would be to control the collection of team projects in HIS:







As you can see above, the HISCollection administrator is forbidden from changing the process template. This privilege is reserved for the Team Foundation Administrator only:

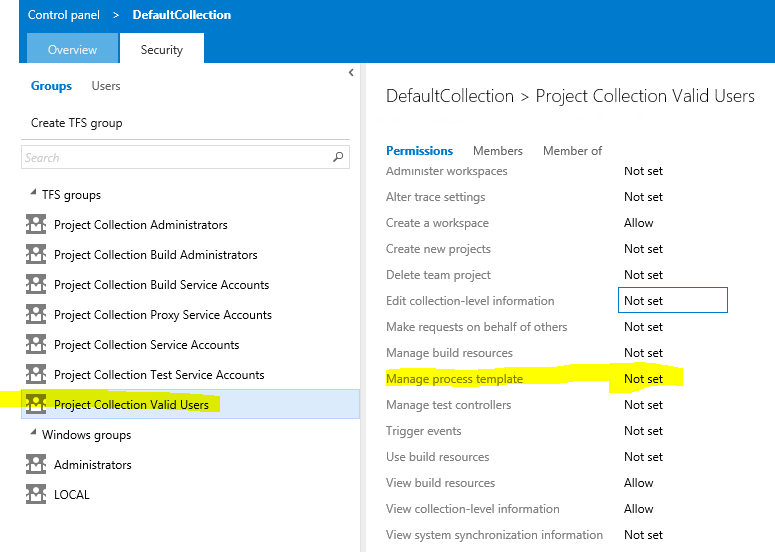


The plan involves setting up a TFS Administrator Team. This team has

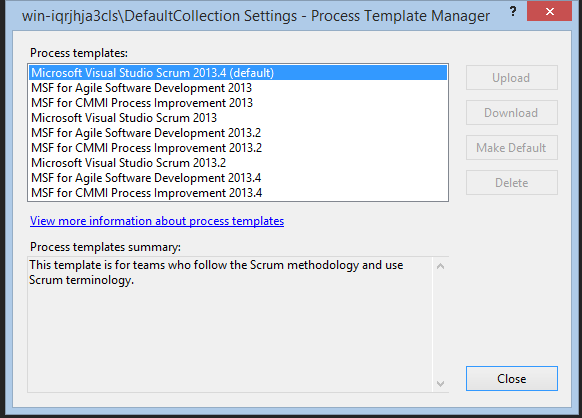
Collection-level Permissions: It is generally recommended to use the TFS Console to administer all collection level information.

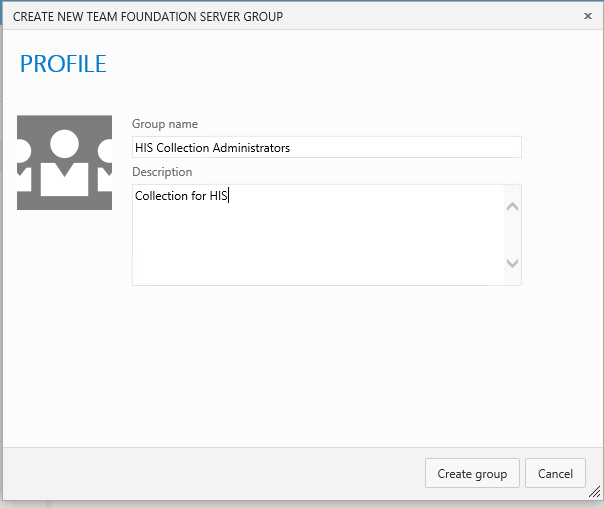
A **key** take-away is to note that the permissions in “Project Collection Administrators” **cannot** be changed. This implies that *ONLY* members of the TFS Administrator group should be members of this group

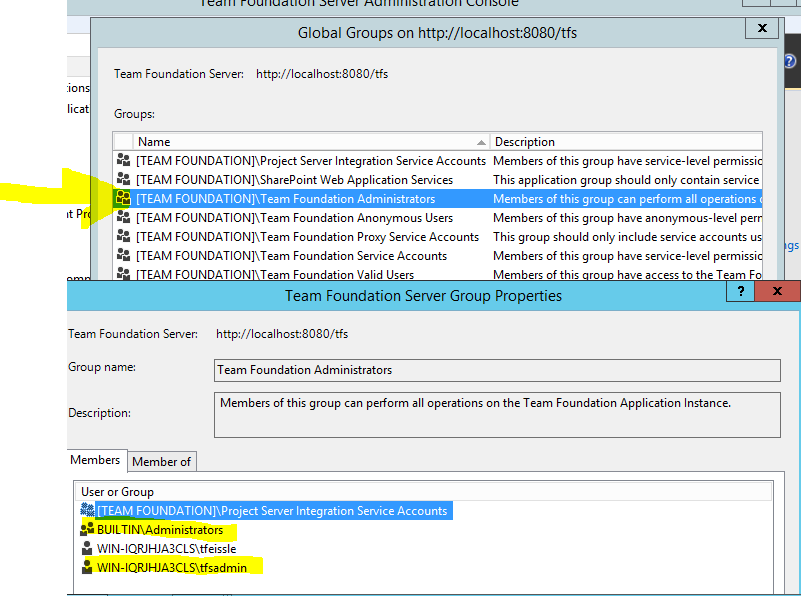
“Manage process template” will be set to **Deny**



Now we see that members of the HISCollection can no longer edit the project template:





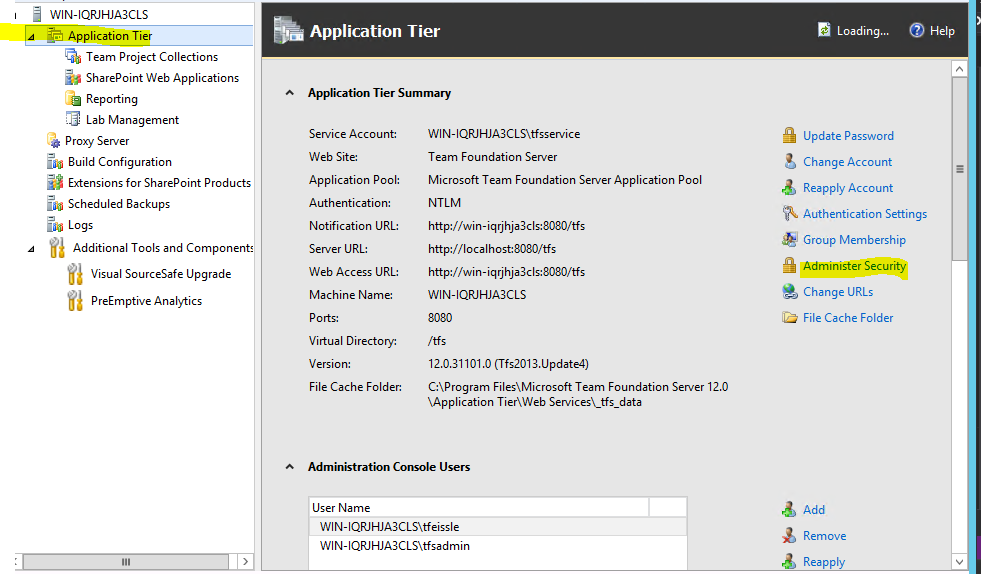


* 1. Server-level permissions

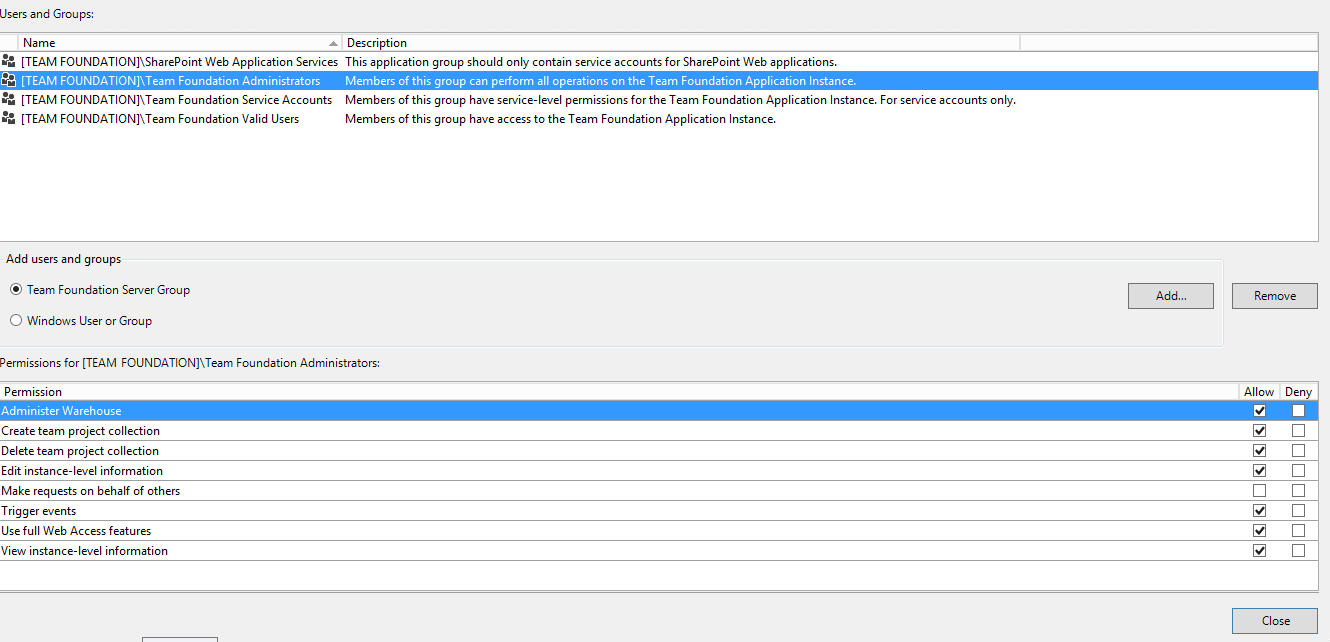
For 3M, it is **recommended** that only members of SEMS or related members of the TFS architectural team be made TFS Administrators.

Server-level permission can affect every project and collection in TFS.

These are set from the console:



As you can see. Team Foundation Administrators can perform nearly all functions:



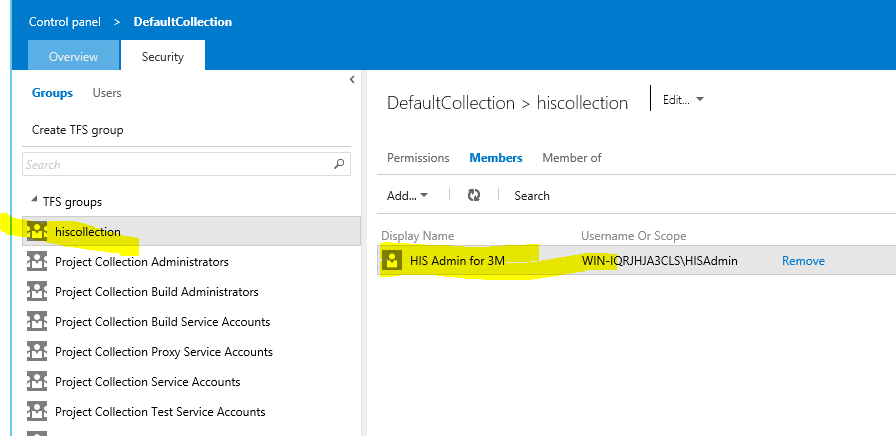
Service Accounts can additionally “Make requests on behalf of others”

* 1. Collection-level permissions

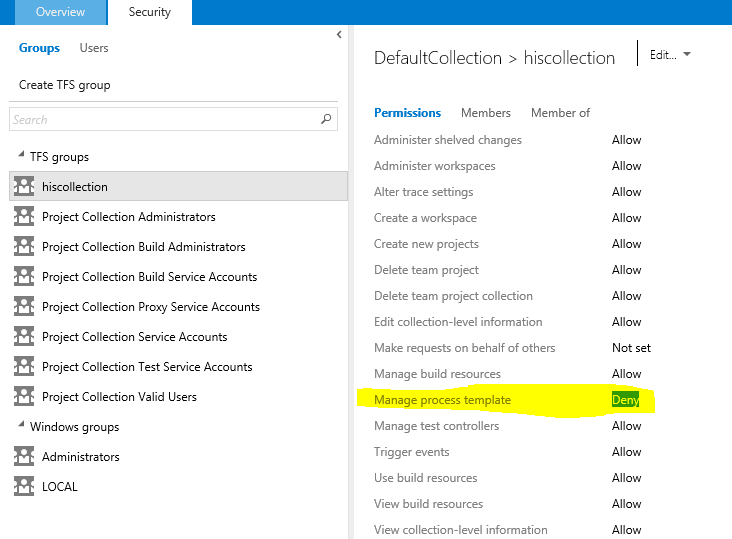
For 3M, it is recommended that various business groups: HIS, IPD, etc. be made collection administrators.

*While these users will be in charge of the security of their collection, they will not have access to the TFS administration console (on the application tier). This will only be available to the TFS administration team.*

Example:



And as you can see, the account CANNOT edit the process template:



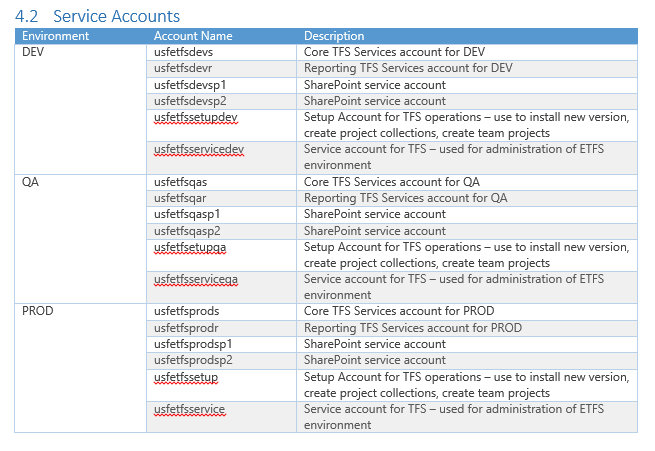
* 1. Project, test and object-level permissions

*<<need more details??>>*

* 1. Release Management permissions

*<<need more details??>>*

* 1. 3M Service Accounts



*<<need more details??>>*

1. Permission Extraction Tool

An interesting tool which has been created by the ALM Rangers allows one to view all of the permissions for a particular collection. This example shows the project level permissions for a test project:

<Name>TestGitAgile</Name>

<ProjectLevelPermissions>

<ProjectLevelPermissionsList>

<Permission>

<DisplayName>View project-level information</DisplayName>

<PermissionConstant>GENERIC\_READ</PermissionConstant>

<Value>Inherited Allow</Value>

<GroupMemberInheritance>[TestGitAgile]\TestGitAgile Team,[TestGitAgile]\Contributors,[TestGitAgile]\Project Administrators</GroupMemberInheritance>

</Permission>

<Permission>

<DisplayName>Create test runs</DisplayName>

<PermissionConstant>PUBLISH\_TEST\_RESULTS</PermissionConstant>

<Value>Inherited Allow</Value>

<GroupMemberInheritance>[TestGitAgile]\Contributors</GroupMemberInheritance>

</Permission>

<Permission>

<DisplayName>Delete test runs</DisplayName>

<PermissionConstant>DELETE\_TEST\_RESULTS</PermissionConstant>

<Value>Inherited Allow</Value>

<GroupMemberInheritance>[TestGitAgile]\Contributors</GroupMemberInheritance>

</Permission>

<Permission>

<DisplayName>View test runs</DisplayName>

<PermissionConstant>VIEW\_TEST\_RESULTS</PermissionConstant>

<Value>Inherited Allow</Value>

<GroupMemberInheritance>[TestGitAgile]\Contributors</GroupMemberInheritance>

</Permission>

<Permission>

<DisplayName>Manage test environments</DisplayName>

<PermissionConstant>MANAGE\_TEST\_ENVIRONMENTS</PermissionConstant>

<Value>Inherited Allow</Value>

<GroupMemberInheritance>[TestGitAgile]\Contributors</GroupMemberInheritance>

</Permission>

<Permission>

<DisplayName>Manage test configurations</DisplayName>

<PermissionConstant>MANAGE\_TEST\_CONFIGURATIONS</PermissionConstant>

<Value>Inherited Allow</Value>

<GroupMemberInheritance>[TestGitAgile]\Contributors</GroupMemberInheritance>

</Permission>

<Permission>

<DisplayName>Edit project-level information</DisplayName>

<PermissionConstant>GENERIC\_WRITE</PermissionConstant>

<Value>Inherited Allow</Value>

<GroupMemberInheritance>[TestGitAgile]\Project Administrators</GroupMemberInheritance>

</Permission>

</ProjectLevelPermissionsList>

</ProjectLevelPermissions>

**Example command line:**

PermissionsExtractionTool -u "tfeissle@msn.com" --collection https://tfeissle.visualstudio.com/DefaultCollection -f test.xml

## Appendix A – References

* [Permission reference for Team Foundation Server](http://msdn.microsoft.com/en-us/library/ms252587.aspx)
* [Command-line tools for TFS](http://msdn.microsoft.com/en-us/library/ms253088.aspx)
* [Changing groups and permissions with TFSSecurity](http://msdn.microsoft.com/en-us/library/ms252504.aspx)
* [Howto Change Access Levels](https://msdn.microsoft.com/en-us/library/jj159364.aspx)

**Appendix B - Upgrade Scenarios**

#### What happens after you change a template and then want to upgrade?

There is a gap in training. 3M pays outside vendors to train.

3M should stand up a software training center?

TAM will schedule and get in touch with Mike O’B.

HIS Security needs to look at security” Version Control