



User's Guide

SoftwarePlanner Connector for Microsoft Team Foundation Server (TFS)

This document is your guide for connecting data in
SoftwarePlanner with other software solutions.

Contents

Understanding the Integrator Capabilities.....	2
Narrative.....	2
What artifacts can be synced?	2
What are the default mappings?.....	4
SoftwarePlanner Requirements to TFS User Stories	4
SoftwarePlanner Defects to TFS Bugs	4
What are the software and hardware requirements?	4
How much does it cost?	5
Installation Instructions.....	6
How do I get started?	6
Can OpsHub do the Installation and Configuration for Us?	6
Setting up your TFS Proxy.....	6
Setting up TFS Custom Fields to Support the Integration	7
Setting up SoftwarePlanner Custom Fields to Support the Integration.....	9
Configuring the Integration using the OpsHub OIM Manager	10
Configuring System Configuration - TFS	10
System Configuration - SmartBear	10
Create Workflow	11
Create Field Mapping	12
Create Integration	12
Modify Default Field Mapping.....	15

Understanding the Integrator Capabilities

Narrative

We understand that not all teams can standardize all artifacts of their software lifecycle using SoftwarePlanner so we've partnered with [OpsHub](http://www.OpsHub.com) (<http://www.OpsHub.com>) to bring you state-of-the-art integrators that allow you to connect data in **Microsoft Team Foundation Server (TFS)** with SoftwarePlanner. This strategy allows team members to use the tool they are most comfortable with while synchronizing data stored in their tool of choice with SoftwarePlanner.

This integration works with the following editions of SoftwarePlanner:

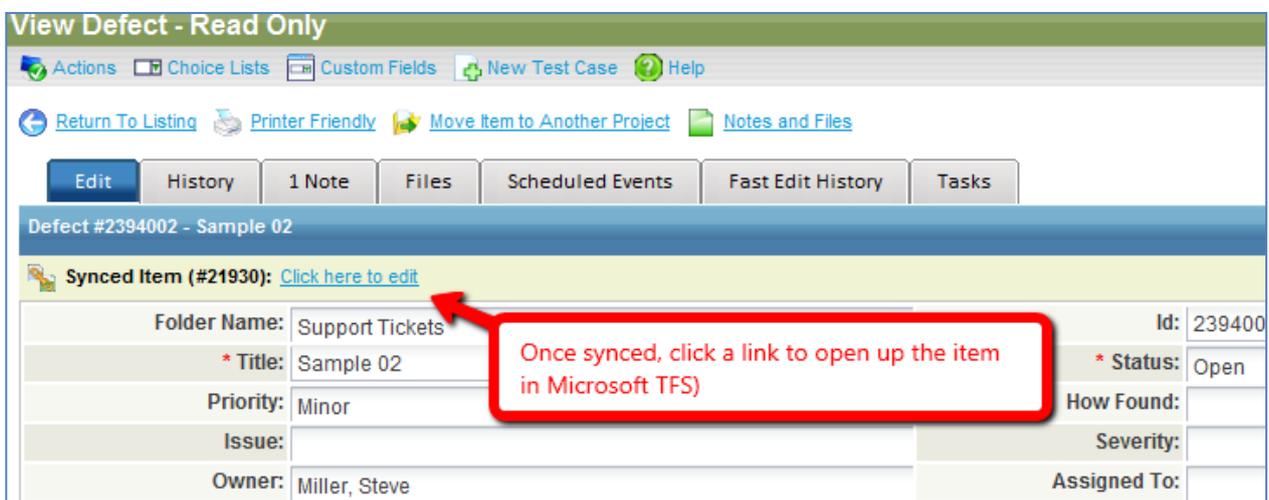
- **QAComplete** – SaaS or Enterprise
- **DevComplete** – SaaS or Enterprise
- **ALMComplete** – SaaS or Enterprise

What artifacts can be synced?

This connector can synchronize these artifacts:

- SoftwarePlanner Requirements with TFS work items (normally TFS User Story or TFS Requirement)
Movie: http://www.softwareplanner.com/Movies.asp?Topic=OpsHub_Requirements
- SoftwarePlanner Defects with TFS work items (normally TFS Bug)
Movie: http://www.softwareplanner.com/Movies.asp?Topic=OpsHub_Defects
- TFS Source Code with SoftwarePlanner Defects and/or Requirements
Movie: http://www.softwareplanner.com/Movies.asp?Topic=OpsHub_SCM

Once an artifact is synced with SoftwarePlanner, SoftwarePlanner locks the record for update inside of SoftwarePlanner, but provides a one-click link to take the user directly to that record in **Microsoft TFS**. Here is how a synced defect record might look inside of SoftwarePlanner:



View Defect - Read Only

Actions: Choice Lists, Custom Fields, New Test Case, Help

Return To Listing, Printer Friendly, Move Item to Another Project, Notes and Files

Edit, History, 1 Note, Files, Scheduled Events, Fast Edit History, Tasks

Defect #2394002 - Sample 02

Synced Item (#21930): [Click here to edit](#)

Folder Name:	Support Tickets	Id:	239400
* Title:	Sample 02	* Status:	Open
Priority:	Minor	How Found:	
Issue:		Severity:	
Owner:	Miller, Steve	Assigned To:	

Once synced, click a link to open up the item in Microsoft TFS

SCM Integration

The connector can also synchronize source code with artifacts. For example, you may want to associate requirements or defects with source code. When associating source code, you have 2 modes of operation:

1. If you are using TFS as your primary defect tracking system (and synching those to SoftwarePlanner), as you check in source code, you can link that source code to TFS defects. When the defect is synced to SoftwarePlanner, it will record this association as a note to the SoftwarePlanner defect.
2. If you are using SoftwarePlanner as your primary defect tracking system (and not synching those to TFS), you can associate source code with SoftwarePlanner defects and they show up in the traceability section of SoftwarePlanner:

Act Hours:	0	Act Start Date:	
Date Last Escalated:	Thu 08/19/2010	Est Finish Date:	
Est Hours:	0	Est Start Date:	
Est Start Date:		Est Finish Date:	

All source commits are associated with the defect(s)

Linked Items: [Link to Items](#) [Traceability Report](#)

- 1 Commit
- 1 File (D:2394002 status:Closed Here is the comment for closing this defect. - 21 - john - 8/3/2010 1:19:33 PM)
- NewDoc1.txt
- 1 Requirement

We have a more detailed document that discusses the setup of the SCM Integration, download it here:
http://www.softwareplanner.com/UsersGuide_TFS_SCM.pdf.

What are the default mappings?

This connector is setup to sync these fields by default (but you can customize it).

SoftwarePlanner Requirements to TFS User Stories

SoftwarePlanner Field	TFS Field
Title	Title
Status	State
Closed (Rejected)	Active
Closed (Completed)	Close
Approved	Resolved
----	Active(DEFAULT)
Description	Description with Acceptance criteria

SoftwarePlanner Defects to TFS Bugs

SoftwarePlanner Field	TFS Field
Title	Title
Status	State
Active	Active
Closed	Closed
Resolved	Resolved
----	Active(DEFAULT)
Description	Steps to Reproduce
Priority	Priority
1-Fix ASAP	1
2-Fix Soon	2
3-Fix If Time	3
----	1(DEFAULT)

If you want to customize your mappings, you can see a SoftwarePlanner data dictionary at http://www.softwareplanner.com/UsersGuide_DataDictionary.pdf.

What are the software and hardware requirements?

To use this connector, you must have the following hardware and software (it installs on either Windows or Linux):

- Microsoft TFS 2010
- ALMComplete, QAComplete or DevComplete version 9.5 or higher
- Windows: Windows XP, Windows Vista, Windows 7 running IIS
- Or if using Linux: Red Hat Linux
- Java 6 (JDK 1.6)
- Memory: 512mb minimum, 2gb recommended
- Databases Supported: Mysql Server with innodb (version 5.1+), Oracle 10g Release 2+, Microsoft SQL Server 2008 R2+, Microsoft SQL Server 2005
- Database Connectors Supported: Mysql Connector/J (JDBC Driver version 5.0+), Oracle JDBC Driver (version 10.2.0.4.0), MS SQL Connector (version sqljdbc_3.0)

How much does it cost?

If you purchased SoftwarePlanner (ALMComplete, QAComplete or DevComplete edition) after November 1, 2010, you receive a certain number of integration modules at no cost. An integration module is a distinct entity (defects, requirements, source control). The number of integration modules you receive at no cost depends on how many licenses of SoftwarePlanner you purchased:

- **0 – 25 users** – up to 1 integration module included at no cost
- **26 – 50 users** – up to 2 integration modules included at no cost
- **51 – 75 users** – up to 3 integration modules included at no cost
- **76 – 100 users** - up to 4 integration modules included at no cost
- **101+ Users** – up to 5 integration modules at no cost

If you purchased SoftwarePlanner prior to November 1, 2010, the cost of the first integration module is 15% of your original SoftwarePlanner purchase price.

If you need to purchase additional integration modules (after your first), they are available for 6% of your original SoftwarePlanner purchase price.

Installation Instructions

How do I get started?

To get started, you will follow these steps:

1. Watch the movies in the "[What artifacts can be synced?](#)" section to see how the integration works.
2. Download and install the OpsHub connector (Windows or Linux) here: <http://www.softwareplanner.com/downloads/OpsHubDownload.asp>
3. Configure your TFS Proxy, see the "[TFS Proxy Configuration](#)" section for more details.
4. Setup Custom Fields in SoftwarePlanner and TFS to support the integration.
5. Configure the integration(s) using the "[OpsHub OIM Manager](#)".

Can OpsHub do the Installation and Configuration for Us?

Yes, if you prefer to have OpsHub do it instead, their Professional Services team can install, configure and/or customizing the mappings. To get started with that, they will need to know what integration modules you wish to configure, custom rules for integration, and any additional required mappings. Once they have this information, they can provide you with a fixed price estimate. Email your request for this to Sandeep Jain sandeep.jain@opshub.com or call him directly at +1 650.743.7361.

Setting up your TFS Proxy

The OpsHub OIM manager requires a special proxy to be placed in front of the TFS web services. This acts as a translation layer between TFS and OpsHub and must be configured for the TFS connector to work. This assumes that you are installing and configuring this on a machine that has IIS and .NET 2.0 installed.

1. Locate the **TFSProxy_<OpsHub Version>.zip** in the installation package and extract it to your IIS location (e.g. extract zip to C:\inetpub\wwwroot\TFSWorkItemWebService).
2. Open **web.config** from TFSWorkItemWebService in notepad and change following properties:

- 2.1 Set value of "**value**" attribute of <param name="File"> element under <configuration> <log4net><appender> with the valid path to logger file that has appropriate write permission for logging. For example, we will save logs to the C:\logs\ folder so the setting will look like this:

```
<configuration>
<log4net>
  <appender name="LogFileAppender" type="log4net.Appender.FileAppender">
    <param name="File" value="C:\logs\OpsHubProxyLog.txt" />
  </appender>
</log4net>
</configuration>
```

- 2.2 Add the **<trust level="Full" />** under the <configuration> <system.web> section.

It will look like this:

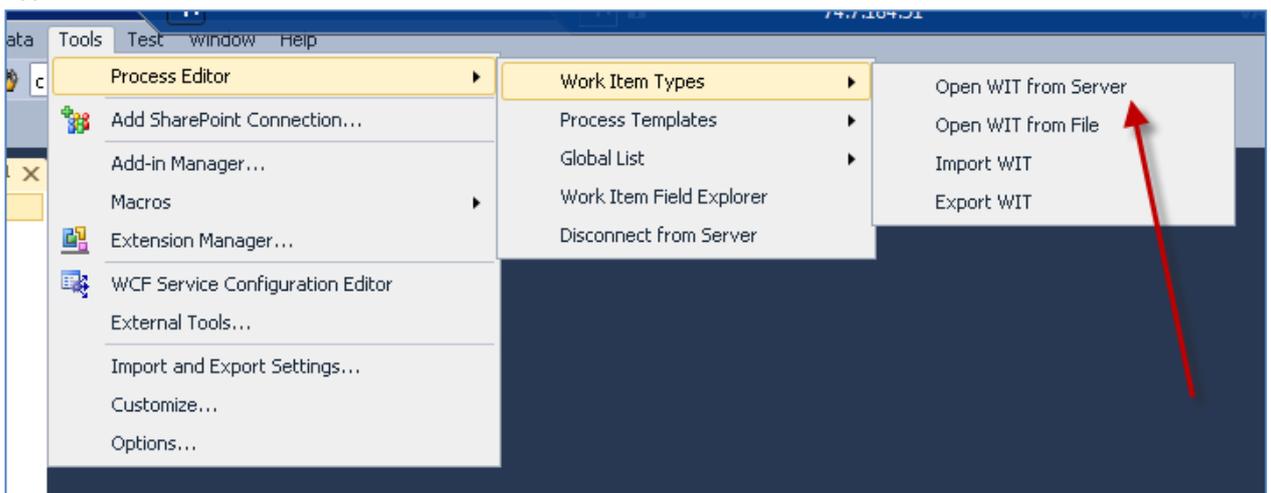
```
<configuration>
<system.web>
  <trust level="Full" />
</system.web>
</configuration>
```

3. Go to **IIS Manager** and navigate to Sites -> Default Web Site -> TFSWorkItemWebService
4. Right click on **TFSWorkItemWebService** and click on "**Convert to Application**".
5. Right click on **TFSWorkItemWebService** and click on "**Switch To Content View**".
6. From Content View Right click on **TFSWorkItemWebService.asmx** and select "Browse". It will open browser showing webservice methods.
7. From browser click on "**Test**" method. Click on **Invoke** in the browser window. It should show "**<string>WebService Call was success.</string>**".

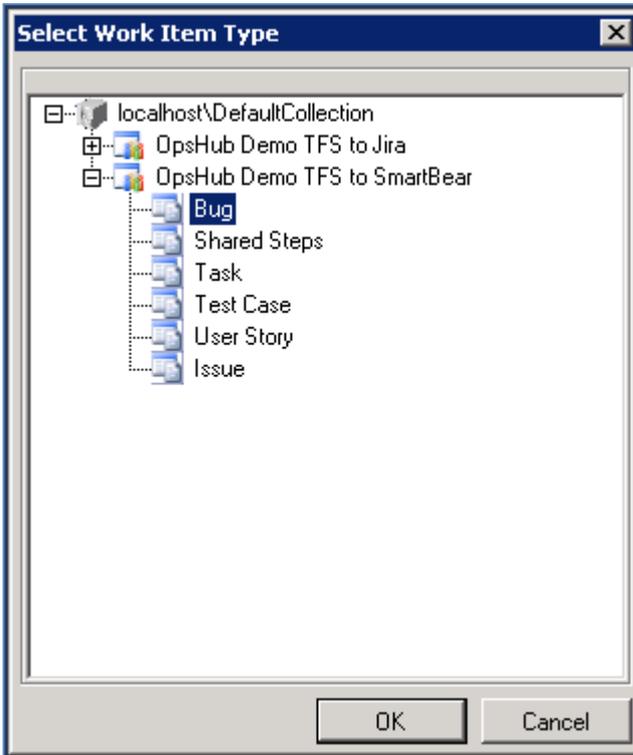
Setting up TFS Custom Fields to Support the Integration

The OpsHub OIM manager requires a few special fields to be defined on the entity that is being synced. These must be set up so that OpsHub can track the integration status of each item. To follow these steps you must have access to the TFS server with Visual Studio 2010.

1. Open Visual Studio 2010.
2. Select the '**Open WIT from Server**' menu item under the **Tools | Process Editor | Work Item Types** menu.



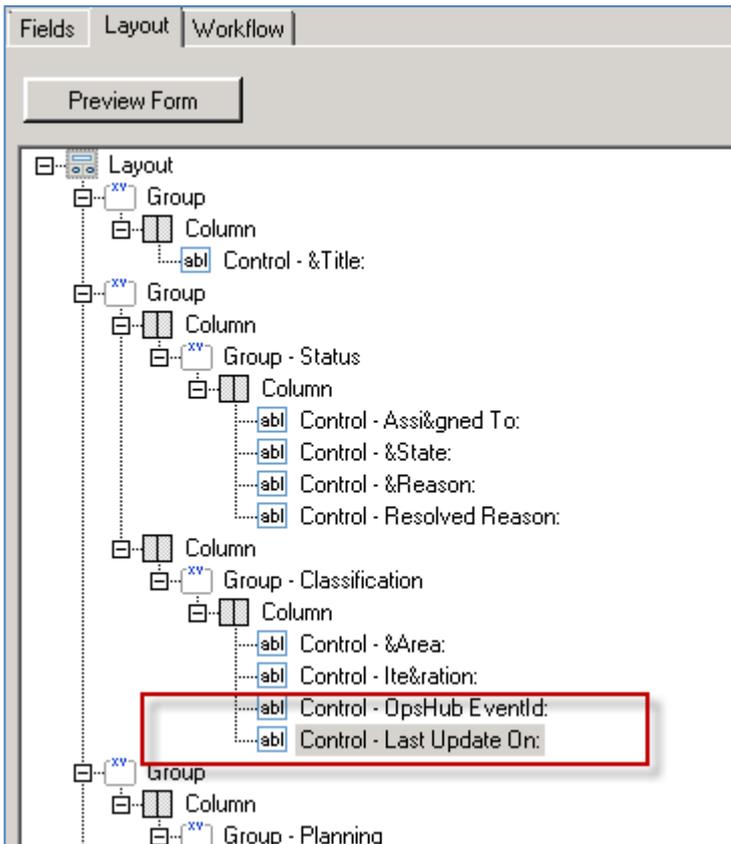
3. Select the TFS collection of the project that is to be synced.
4. Expand the project and then select the entity that is to be synced (in this case Bug).



5. Press 'OK' to open the **Work Item Type Fields** screen.
6. Create a new field.
7. Set **Name** to '**OpsHub EventId**'
8. Set **Type** to '**String**'.
9. Set **Reference Name** to '**OpsHub.EventId**'.
10. Select 'OK' to save the new field.
11. Create a new field.
12. Set **Name** to '**OpsHub LastUpdateOn**'.
13. Set **Type** to '**String**'.
14. Set **Reference Name** to '**OpsHub.LastUpdateOn**'.
15. Select 'OK' to save the new field.

ID	Integer	System.ID
Area ID	Integer	System.AreaId
Authorized As	String	System.AuthorizedAs
OpsHub EventId	String	OpsHub.EventId
OpsHub LastUpdateOn	String	OpsHub.LastUpdateOn

16. Next, add these new fields to the layout section of the entity.
17. It is recommended that you add them to the **Group - Classification** grouping.
18. On the Column node in the select group, right-click and select '**New Control**'.
19. Set **Field Name** to '**OpsHub.EventId**' (select from the dropdown list).
20. Set **Label** to '**OpsHub EventId**'.
21. On the **Column** node in the selected group, right-click and select '**New Control**'.
22. Set **Field Name** to '**OpsHub.LastUpdateOn**' (select from the dropdown list).
23. Set **Label** to '**OpsHub.Last Update On**'.



24. Be sure to click on the 'Save' icon to save all changes.

Setting up SoftwarePlanner Custom Fields to Support the Integration

The OpsHub OIM manager requires a few special fields to be defined on the entity that is being synced. These must be set up so that OpsHub can track the integration status of each item. To follow these steps you must have access to the TFS server with Visual Studio 2010.

1. Log in to SoftwarePlanner.
2. Navigate to the entity that you want to set custom fields for (in our case, we will set Defects).
3. Select a defect and go to the edit screen.
4. Select the 'Custom Fields' sub menu item.
5. Scroll down to **custom field 88** add a field called '**OH_Last_Update**' and set it to '**Text**' with a length of '**50**'.
6. In **custom field 89** add a field called '**eventId**' and set it to '**Text**' with a length of '**100**'.
7. In **custom field 90** add a field called '**Link to TFS**' and set it to '**Text**' with a length of '**200**'.

88		OH_Last_Update	Text	50
89		eventId	Text	100
90		Link to JIRA	Text	100

8. Make a note of the custom field numbers from above as that will be used in the Modify Default Field Mapping section.

Configuring the Integration using the OpsHub OIM Manager

The OpsHub OIM manager should be installed using the install package supplied. Be sure to follow all instructions included in the installation document. Once OpsHub has been installed and configured, be sure to start the OpsHub application by navigating to the **All Programs | OpsHub Application | Start OpsHub Server** menu item in the Start menu. If it is running you should see a command window spin up and it will remain open all the time. Now, you can navigate to the OpsHub OIM web based management software by following these steps.

1. Open a browser and navigate to <http://localhost:8989/OpsHubWS/>
NOTE: The OpsHubWS is case-sensitive and it is recommended you use **Firefox** or **IE7** for this.
2. Type in '**admin**' for the user and '**password**' for the corresponding password.
3. The OpsHub manager will be displayed.

Movie: Watch this movie to get an overview of the OpsHub OIM Manager:

http://www.softwareplanner.com/Movies.asp?Topic=OpsHub_OIM

Configuring System Configuration - TFS

Before setting up any integrations or workflows in the OpsHub manager we need to create the systems that will be utilized in our process and mappings. We will be setting up one for TFS and one for SmartBear. The **System Configuration** panel is located in the **Administration** tab in the **OpsHub OIM** web manager.

1. Log in to the **OpsHub OIM Manager** (see prior topic).
2. Select the '**Create System**' menu option and select '**Team Foundation Server**' from the dropdown.
3. Set **Display Name** to '**Team Foundation Server**'
4. Set **System Name** to '**tfs**'. **NOTE:** This setting must be exactly like this including case.
5. Set **Version** to '**2010**'.
6. Set **Enable OIM** to '**Yes**'.
7. Set **TFS WebService URL** to
'[http://\[YourTFSMachineName\]/TFSWorkItemWebService/TFSWorkItemWebService.asmx](http://[YourTFSMachineName]/TFSWorkItemWebService/TFSWorkItemWebService.asmx)'.
NOTE: This will vary by where you have installed your TFS web service proxy.
8. Set **TFS Server URL** to '[http:// \[YourTFSMachineName\]/tfs](http://[YourTFSMachineName]/tfs)'.
NOTE: This will vary by where you have installed your TFS server.
9. Set **TFS User Name**, **TFS User Password**, and **TFS User Domain** to the user that you want to authenticate and use for sync.
10. Set **TFS Team Collection Name** to the text value of the collection where the project resides that is used in the sync (e.g. 'DefaultCollection').
11. Set **Maximum Retry Count** to '**3**'.

System Configuration - SmartBear

Before setting up any integrations or workflows in the OpsHub manager we need to create the systems that will be utilized in our process and mappings. We will be setting up one for TFS and one

for SmartBear. The **System Configuration** panel is located in the **Administration** tab in the **OpsHub OIM** web manager.

1. Log in to the **OpsHub OIM Manager**.
2. Select the **'Create System'** menu option and select **'SmartBear'** from the dropdown.
3. Set **Display Name** to **'SmartBear'**.
4. Set **System Name** to **'SmartBear'**.
NOTE: This setting must be exactly like this including case.
5. Set **Version** to **'9.5'** or **'9.6'**.
6. Set **Enable OIM** to **'Yes'**.
7. Set **SmartBear User Name** to the **last name, first name middlename** of the corresponding user account being used in the sync (e.g. 'WebService, User').
8. Set **SmartBear User Email-Id** to the **email id of the user account** being used in the sync (e.g. 'User.WebService@abc.com').
9. Set **SmartBear User Password** to the corresponding password of the user account that is used in the sync.
10. Set **Binding URL** to **'http://ws.softwareplanner.com/psWS.asmx?wsdl'**.
NOTE: The setting value is for SaaS clients and will be different for Enterprise customers.
11. Set **SmartBear Application Code** to **'agSP'**.
NOTE: The setting value is for SaaS clients and will be different for Enterprise customers.
12. Set **Maximum Retry Count** to **'3'**.

Create Workflow

The OpsHub connector requires workflow mappings to be configured. The SmartBear installation provides some default workflow mappings and these are installed and saved in the **doc/Workflow/Common** folder. For this example we will install the *TFS Create Bug to SmartBear Defect* workflow item.

1. Log in to the **OpsHub OIM Manager**.
2. Select the **'Workflow Upload'** menu option in the **OIM Configuration** tab.
3. Create a name that is descriptive (recommended to use the same name as the file that is selected)
4. Optionally, add a **description**.
5. Select the **'Browse...'** button and select the process definition file (e.g. **'TFS Create Bug to SmartBear Defect.xml'**).
6. Next, press the **'Upload'** button.
7. Select the **Systems** that are to be used in this workflow. In this case it is SmartBear and Team Foundation Server. *NOTE: Use the CTRL key and left click on each system to select more than one item.*
8. Click on the **'Save'** button.

Create Field Mapping

The OpsHub connector requires field mappings to be configured. The SmartBear installation provides some default field mappings and these are installed and saved in the **doc/SmartBearMappings /SmartBear_TFS** folder. For this example we will install the *TFS Create Bug to SmartBear Defect* field mapping.

1. Log in to the **OpsHub OIM Manager**.
2. Select the **'Field Mapping Upload'** menu option in the **OIM Configuration** tab.
3. Create a **name** that is descriptive (recommended to use the same name as the file that is selected)
4. Optionally, add a **description**.
5. Select the **'Browse...'** button and select the process definition file (e.g. **'TFS Create Bug to SmartBear Defect.xml'**).
6. Next, press the **'Upload'** button.
7. Click on the **'Save'** button.

Create Integration

The OpsHub connector requires an integration that will tie the field mappings and workflow together on a set schedule. For this example we will configure the TFS To SmartBear Bugs that will utilize the create and update workflows and corresponding field mappings.

1. Log in to the **OpsHub OIM Manager**.
2. Select the **'Create Integration'** menu option in the **OIM Configuration** tab.
3. Create a **name** that is descriptive (recommended to use a similar name to the integration we are doing such as **'TFS to SmartBear Bugs'**).
4. Select the **Source System**. This will be determined by which system is the source system and is usually indicated by the first system name in the process workflows (in our case this is Team Foundation Server for Bugs).
5. Set Sync to **'Both'**.
6. Set **Configure Criteria** to **'No'**.
7. Set **Issue Type** to **'TFS Defect'**.
NOTE: This is just for this example, if you are doing a different source system or entity, select the appropriate entity).
8. Set the **TFS User Name**, **TFS User Password**, and **TFS User Domain** to the user who will be used in the sync process.
NOTE: Make sure this user has appropriate security to the selected entities on TFS.
9. Set **Maximum Retry Count** to **'3'**.
10. Set **Start Polling Time** to a date in the near or recent past. The OpsHub integration will use this as the start date for when data will be pulled over from the source system to the destination system (all items that were created or updated previously will not be synced).
11. Set **Associate Schedule** to a polling time that you are comfortable with. For this example we have selected 1 minute.

Create OIM Integration

*Integration Name	<input type="text" value="TFS To SmartBear"/>
*System Name	<input type="text" value="tfs"/>
*Sync	<input checked="" type="radio"/> Both (Failed and New events) <input type="radio"/> New Events <input type="radio"/> Failed Events
*Configure Criteria	<input type="text" value="No"/>
*Issue Type	<input type="text" value="TFS Defect"/>
TFS User Name	<input type="text"/>
TFS User Password	<input type="text"/>
TFS User Domain	<input type="text"/>
Maximum Retry Count	<input type="text"/>
*Start Polling Time	<input type="text" value="Wed May 11 2011"/> <div style="display: flex; gap: 5px;"> <input type="text" value="20"/> <input type="text" value="55"/> <input type="text" value="09"/> </div>
*Associate Schedule	<input type="text" value="1 Minute Schedule"/>

12. Next, **select the event type**, create/update/Common Configuration to all Events, depending on whether workflows and mappings are common or not.
13. Set the **Field Mapping** to the corresponding Mapping, in our case it will be TFS Bug To SmartBear Defect.xsl.
14. Set the **Workflow** to the corresponding workflow
15. Select the **'Set Parameter'** link and the Workflow Configuration Parameter screen should be displayed.
16. Set the **SmartBear Project Name** to corresponding **project name from drop down list** to sync with.
Note: The user that is configured in system, should have privelege to view this project.
17. Set the Software Planner **Entity Type** to **'Software Planner Defect'**.

Add Mapping, Workflow X

Field Mapping Team Foundation Defect XYZ ▼

Workflow TFS Defect To XYZ System D ▼

▼ **Set Workflow Parameter**

Team Foundation Server Entity Type(TFS Test Instance) TFS Defect ▼

Team Foundation Server Collection Name(TFS Test Instance)

Team Foundation Server User Name(TFS Test Instance)

Team Foundation Server User Domain(TFS Test Instance)

Team Foundation Server User Password(TFS Test Instance)

18. Set the Team Foundation Server **Entity Type** to 'TFS Defect'.
 19. Press the '**Save**' button.
 20. If you have an update WorkFlow and Field Mapping, repeat steps 12-19.
 21. Press the '**Submit**' button on the **Create Integration** screen.
- NOTE:** If you do not press the submit button on this page the integration will be lost.
22. The final step is to make sure it is Active. On the '**View OIM Integrations**' screen select the down arrow next to the **Edit** button and select the '**Active**' option.

View OIM Integrations

Integration Name

Status Select ▼

[Change To Advanced Search](#)

Integration Name ▶	Source System Name ▶	Source Entity ▶	Status ▶	Last Event Processed Time ▶	Last Processed Event ▶	Options
TFS Defect To SmartBear Defect	tfs	TFS Defect	Inactive	Wed May 11 20:59:52 IST 2011	735	<input type="button" value="Edit"/> ▼
SmartBear Defect To TFS defect	Jira	Jira Bug	Inactive	Wed May 11 21:00:33 IST 2011	TFS-99	<input type="button" value="Edit"/> ▼

1 to 2 of 2

Modify Default Field Mapping

The OpsHub default field mappings can be updated to change where data is being sent in one system (the default mappings and fields are shown below in this document and is different for each entity and system). To modify field mappings you can use the OIM web manager.

1. Log in to the **OpsHub OIM Manager**.
2. Navigate to **OIM Configuration | Integration | View Integrations**. Inactivate all integration which refers the Field Mapping which is required to modify.
3. Select the '**Field Mapping**' menu option in the **OIM Configuration** panel.
4. Select the '**View Field Mappings**' menu option.
5. Find the field mapping you need to change select the '**Export**' link.
6. **Note:** You can select multiple files, and then export, from the table level menu option.
7. Unzip, the files and edit in local Editor.
8. Now, to replace this, select '**Replace with new version**' option from Menu, upload this file. It will replace the current mappings with newly uploaded file.
9. Navigate to **OIM Configuration | Integration | View Integrations** , activate integrations which were inactivated in step 2.