

spiraPlan[®]

spiraTest[®]

spiraTeam[®]

**SpiraTest / SpiraTeam / SpiraPlan[®] | External Bug-Tracking
Integration Guide**

Inflectra Corporation

Date: July 8th, 2019

inflectra[®]

Contents

Introduction	1
1. Setting up Data Synchronization	2
2. Using SpiraTeam with JIRA 5+	9
3. Using SpiraTeam with JIRA 3 / 4....	34
4. Using SpiraTest with Bugzilla	48
5. Using SpiraTest with MS-TFS	60
6. Using SpiraTest with FogBugz	84
7. Using SpiraTeam with Mantis.....	96
8. Using SpiraTeam with ClearQuest107	
9. Using SpiraTeam with IBM RTC... 119	
10. Using Spira with Axosoft 14+	127
11. Using SpiraTeam with Redmine . 138	
12. Using Spira with VersionOne.....	149
13. Using Spira with OnTime 11	159
14. Using Spira with GitHub	172
15. Using Spira with GitLab	176
16. Using Spira with ServiceNow	181

Introduction

SpiraTest® provides an integrated, holistic Quality Assurance (QA) management solution that manages requirements, tests and incidents in one environment, with complete traceability from inception to completion.

SpiraTeam® is an integrated Application Lifecycle Management (ALM) system that manages your project's requirements, releases, test cases, issues and tasks in one unified environment. SpiraTeam® contains all of the features provided by SpiraTest® - our highly acclaimed quality assurance system and SpiraPlan® - our agile-enabled project management solution.

However, many organizations may be already using other bug-tracking systems and not want to have to migrate all their users over to SpiraTeam. Therefore SpiraPlan, SpiraTest and SpiraTeam are capable of integrating with a variety of commercial and open-source bug-tracking systems.

This guide outlines how to integrate and use SpiraTest, SpiraPlan and SpiraTeam in conjunction with other external Bug/Issue Tracking systems.

This guide assumes that the reader is familiar with both SpiraTeam and the appropriate tool being discussed. For information regarding how to use SpiraTeam, please refer to the *SpiraTeam User Manual*.

Each of the sections covers a different tool so we recommend using the table of contents on the left to locate the tool you're looking to either integrate or migrate from, then read the installation and usage instructions.

1. Setting up Data Synchronization

This section outlines the general data synchronization configuration to use any of the supported bug trackers with SpiraTest, SpiraPlan or SpiraTeam (hereafter referred to as Spira).

► Please read this section first, before performing the configuration steps specific to your bug-tracker.

The built-in data-synchronization service that comes with Spira, allows the quality assurance team to manage their requirements and test cases in Spira, execute their test runs, and then have the new defects/bugs generated during the run be automatically loaded into an external bug-tracker. Once the incidents are loaded into the external bug-tracker, the development team can then manage the lifecycle of these defects/bugs in their chosen tool, and have the status changes be reflected back in Spira.

In addition, any issues logged directly into the external bug-tracker will get imported into Spira as either new incidents or new requirements (depending on their type) so that they can be used as part of the planning and testing lifecycle.

There are three possible deployment options for the Spira data synchronization:

1. You have both Spira and the External Bug Tracker cloud-hosted
2. You have Spira installed on-premise (External Bug Tracker can be either)
3. You have Spira cloud-hosted, but the External Bug Tracker installed on-premise

We shall provide the configuration steps for each option:

1.1. Spira & External Tool Cloud Hosted

When you sign up for Spira as a cloud-hosted subscription, you have the option of including the Spira DataSync service as an add-on feature to the subscription:

Add Optional Features

Please select which of the FREE optional features you'd like to add to your trial:

- Spira DataSync** - Data Synchronization Services for Spira
- TaraVault** - Secure Source Code Management and Hosting

Make sure you include the 'Spira DataSync' add-on with your subscription. Once your subscription is provisioned, you will be able to configure the connection to Spira by going to your secure Customer Area on our website:

My Subscriptions

This is where you can view all of your SaaS hosted subscriptions with Inflectra. All of our subscriptions include software, hosting and technical support.

Product	Type	Site URL	End Date	Status	#	Actions
SpiraTeam	Trial	https://demo.spiraservice.net/ams1	30-Jun-2016	Active	n/a	<button>Purchase</button> <button>Customize</button>
Spira-DataSync	Trial	https://demo.spiraservice.net/ams1	30-Jun-2016	Active	1	<button>Configure</button>

Click on the 'Configure' button associated with the **Spira-DataSync** addon row:

< **Data Synchronization** [https://demo.spiraservice.net/ams1]

Please enter in a login/password that will be used to connect to your instance of SpiraTeam:

SpiraTeam URL*: <https://demo.spiraservice.net/ams1>

Login*:

Password*:

Active

Enter a login and password that can connect to your Spira instance. This user needs to be a member of the project(s) that will be synchronized with the external bug-tracker and needs to have at least **Incident create/modify/view** permissions and **Release create/modify/view** permissions in these projects.

Click on the 'Test' button to verify the credentials, and once they validate, make sure the 'Active' flag is checked and then click 'Save'. You have now configured the synchronization.

You should now see a list of the plugins currently configured in your Spira instance:

Plugin Name	Status	Last Sync Date	Operations
BugzillaDataSync	Failure	6/21/2016 5:09 PM	<input type="button" value="Manage"/>
FogBugzDataSync	Failure	6/21/2016 5:09 PM	<input type="button" value="Manage"/>
JiraDataSync	Warning	6/21/2016 5:09 PM	<input type="button" value="Manage"/>
MsTfsDataSync	Warning	6/21/2016 5:09 PM	<input type="button" value="Manage"/>
			<input type="button" value="Manage All"/>

If you click on any of the 'Manage' buttons you will be taken to your Spira instance where you can complete the plugin configuration:

Data Synchronization

SpiraTeam is able to communicate with a variety of external systems and applications using its background data synchronization service and library of available plug-ins.

This page allows you to view, add and modify the list of plug-ins, make changes to their configuration, and verify the status of the most recent synchronization. In addition, you can force a particular plug-in to perform a complete data-synchronization by resetting its date of last sync:

Plug-In	Data Mapping	Last Sync Date	Status	Operations
BugzillaDataSync	<input type="button" value="View Project Mappings"/>	6/21/2016 1:09:56 PM	Failure	<input type="button" value="Force Re-Sync"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="View Errors"/>
FogBugzDataSync	<input type="button" value="View Project Mappings"/>	6/21/2016 1:09:56 PM	Failure	<input type="button" value="Force Re-Sync"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="View Errors"/>
JiraDataSync	<input type="button" value="View Project Mappings"/>	6/21/2016 1:09:56 PM	Warning	<input type="button" value="Force Re-Sync"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="View Errors"/>
MsTfsDataSync	<input type="button" value="View Project Mappings"/>	6/21/2016 1:09:56 PM	Warning	<input type="button" value="Force Re-Sync"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="View Errors"/>

The steps for configuring each plugin are specific to each external bug-tracking tool. Please refer to the appropriate section in this document for the tool you are using.

1.2. Spira Installed On-Premise

With Spira installed on-premise, there is a built-in Windows® service that is installed with Spira that is not running by default, but is available for data-synchronization.

The steps that need to be performed to configure integration are as follows:

- Download appropriate plug-in for Spira from our website
- Configure the DataSync Service
- Start the service and proceed to the plugin specific section of this manual

1.2.1. Download the Data-Sync Plug-In

Go to the Inflectra website and open up the page that lists the various downloads available for Spira (<http://www.inflectra.com/SpiraTeam/Downloads.aspx>). Listed on this page will be the data-synchronization plug-In for your desired bug-tracking tool. Right-click on this link and save the Zip compressed folder to the hard-drive of the server where Spira is installed.

Open up the compressed folder and extract the DLL assembly files and place them in the C:\Program Files (x86)\SpiraTeam\Bin folder (it may be SpiraTest or SpiraPlan depending on which product you're running). This folder should already contain the DataSyncService.exe and DataSyncService.exe.config files that are the primary files used for managing the data synchronization between Spira and other systems.

1.2.2. Configuring the Synchronization Service

To configure the integration service, please open up the DataSyncService.exe.config file located in C:\Program Files (x86)\SpiraTeam\Bin with a text editor such as Notepad. Once open, it should look like:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <sectionGroup name="applicationSettings" type="System.Configuration.ApplicationSettingsGroup, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" >
      <section name="Inflectra.SpiraTest.DataSyncService.Properties.Settings"
type="System.Configuration.ClientSettingsSection, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" requirePermission="false" />
    </sectionGroup>
  </configSections>
  <applicationSettings>
    <Inflectra.SpiraTest.DataSyncService.Properties.Settings>
      <setting name="PollingInterval" serializeAs="String">
        <value>600000</value>
      </setting>
      <setting name="WebServiceUrl" serializeAs="String">
        <value>http://localhost/SpiraTeam</value>
      </setting>
      <setting name="Login" serializeAs="String">
        <value>fredbloggs</value>
      </setting>
      <setting name="Password" serializeAs="String">
        <value>fredbloggs</value>
      </setting>
      <setting name="EventLogSource" serializeAs="String">
        <value>SpiraTeam Data Sync Service</value>
      </setting>
      <setting name="TraceLogging" serializeAs="String">
        <value>False</value>
      </setting>
    </Inflectra.SpiraTest.DataSyncService.Properties.Settings>
  </applicationSettings>
</configuration>
```

The sections that need to be verified and possibly changed are marked in yellow above. You need to check the following information:

- The polling interval allows you to specify how frequently the data-synchronization service will ask Spira and the external system for new data updates. The value is specified in milliseconds and we recommend a value no smaller than 5 minutes (i.e. 300,000ms). The larger the number, the longer it will take for data to be synchronized, but the lower the network and server overhead.
- The base URL to your instance Spira. It is typically of the form <http://<server name>/SpiraTeam>. Make sure that when you enter this URL on a browser on the server itself, the application login page appears.
- A valid login name and password to your instance of Spira. This user needs to be a member of the project(s) that will be synchronized with the external bug-tracker and needs to have at least **Incident create/modify/view** permissions and **Release create/modify/view** permissions in these projects.

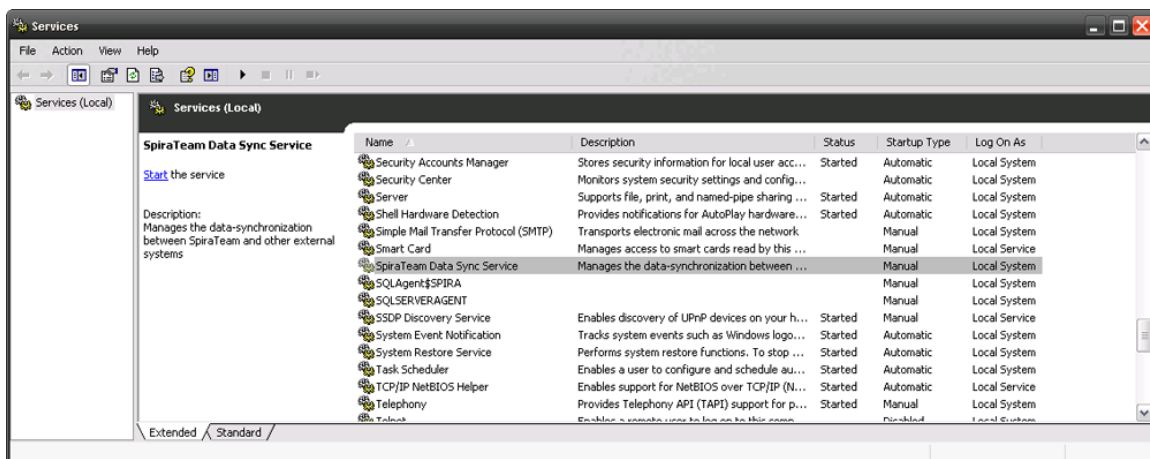
Once you have made these changes, please refer to the section in this document that covers the specific bug-tracking tool you will be integrating with.

Note: If you are using the MS-TFS plugin on premise, you will also need to switch over your IIS application pool running Spira to “Enable 32-bit Applications. You will also need to download the recompiled 32-bit version of the DataSyncService.exe application from our support knowledge base - [KB14 - Using SpiraTeam Data Synchronization with TFS on a 64-bit system](#).

1.2.3. Starting the Data-Synchronization Service

When Spira is installed, a Windows Service – SpiraTeam Data Sync Service – is installed along with the web application. However to avoid wasting system resources, this service is initially set to run manually. To ensure continued synchronization of SpiraTeam with the external tool, we recommend starting the service and setting its startup-type to Automatic.

To make these changes, open up the Windows Control Panel, click on the “Administrative Tools” link, and then choose the Services option. This will bring up the Windows Service control panel:



Click on the 'SpiraTeam Data Sync Service' entry and click on the link to start the service. Then right-click the service entry and choose the option to set the startup type to 'Automatic'. This will ensure that synchronization continues after a reboot of the server.

1.3. Spira Cloud Hosted, External Tool On-Premise

The Desktop Data Synchronization utility (hereafter referred to as the “Desktop DataSync”) is a standalone utility that can be used to run the various Data Synchronization Plugins without a server installation of Spira.

This is useful where you have your SpiraTeam instance cloud hosted, but the external tool is locally installed behind your firewall.

1.3.1. Installation

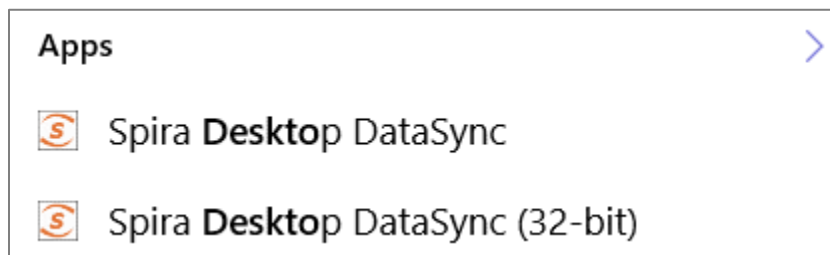
To obtain the Desktop DataSync, go to the Inflectra website and under the “Downloads and Add-Ons” section you will find a Windows Installation (MSI) package that will install the Desktop DataSync onto your computer. The installer will install both a 64-bit version of the Desktop Data Sync and a 32-bit version. You should use the 64-bit version for all plugins except the Microsoft TFS plugin which will require the 32-bit version.



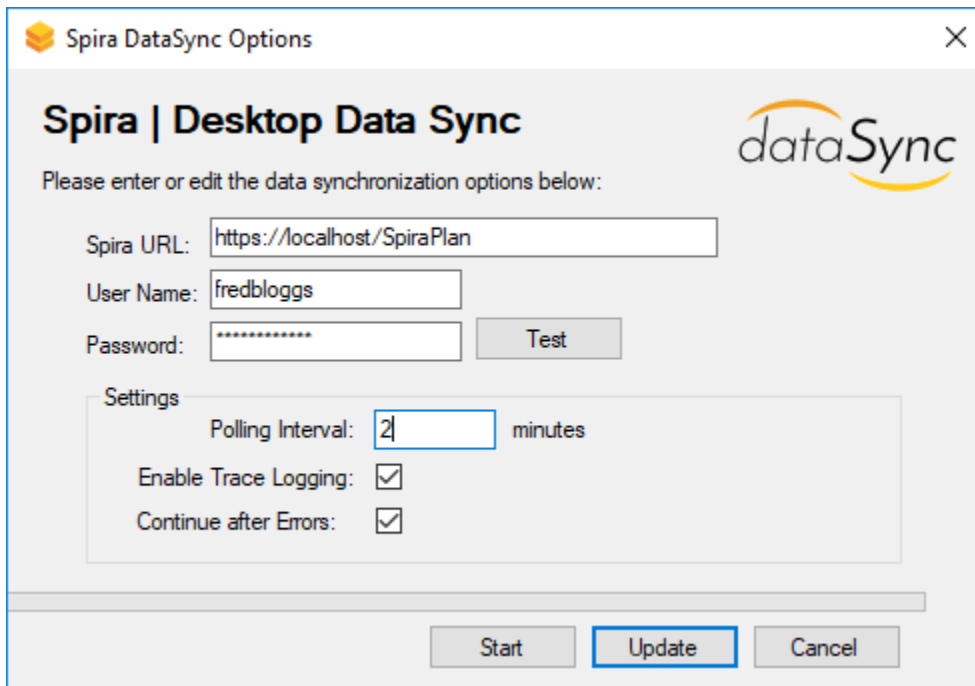
Next you need to download the appropriate plug-in(s) for the various bug-trackers (as described in the appropriate section of this document) and place the assemblies (DLL files) into the same folder that contains the DesktopDataSync.exe application.

1.3.2. Usage

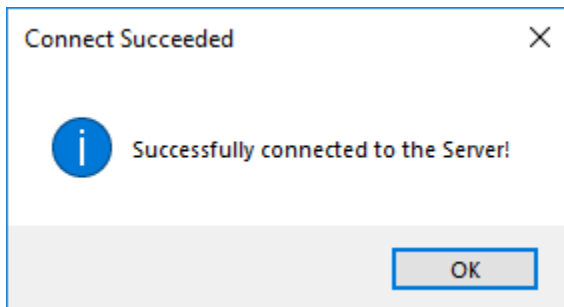
Once you have downloaded and installed the application and appropriate plug-ins, go to Start > Programs > Inflectra > Desktop DataSync to launch the application.



This will bring up the main options window of the application:

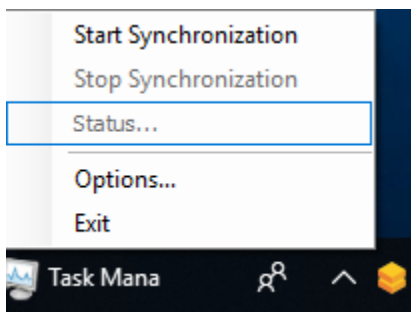


You should then enter the URL, login and password to your Spira installation and click [Test]. Assuming that this information is correct, you will see a confirmation message:



Now you should complete the configuration by setting the Polling Interval (how often the utility will synchronize data between Spira and the external system) and whether Trace Logging is enabled (useful when verifying your data mapping, but will fill up the application log, so leave unchecked for production use). Then click the [Update] button to save your settings or [Start] to save your settings and start synchronization immediately.

Once the Options window closes, the application will remain active in the system tray of your computer:



You can then use the right-click context menu to start synchronization, stop synchronization, view the status (if synchronization is running) or exit the application altogether.

During synchronization, any errors will be logged to the Windows Application Event Log and you can use those logs to diagnose any issues connecting to the external bug-tracker or any data mapping configuration changes that need to be made.

2. Using SpiraTeam with JIRA 5+

This section outlines how to use SpiraTest, SpiraPlan or SpiraTeam (hereafter referred to as SpiraTeam) in conjunction with the JIRA issue/bug tracking system version 5.0 and later. The built-in integration service allows the quality assurance team to manage their requirements and test cases in SpiraTeam, execute test runs in SpiraTest, and then have the new incidents generated during the run be automatically loaded into JIRA. Once the incidents are loaded into JIRA as issues, the development team can then manage the lifecycle of these issues in JIRA, and have the status changes in JIRA be reflected back in SpiraTeam.

In addition, any issues logged directly into JIRA will get imported into SpiraTeam as either new incidents or new requirements (depending on their type) so that they can be used as part of the planning and testing lifecycle.

► **STOP! Please make sure you have first read the Instructions in Section 1 before proceeding!**

2.1. Configuring the Plug-In

This section outlines how to configure the integration service to export incidents into JIRA, import new issues from JIRA and pick up subsequent status changes in JIRA and have them update SpiraTeam. It assumes that you already have a working installation of SpiraTest, SpiraPlan or SpiraTeam and a working installation of JIRA.

The following versions of SpiraTeam and JIRA are supported:

- The JIRA 5.x plugin supports JIRA 5.0 or later and SpiraTeam v4.0 or later
- The JIRA 4.x plugin supports JIRA 4.0 or later and SpiraTeam v3.0 or later (see section 3)
- The JIRA 3.x plugin supports JIRA 3.0 or later and SpiraTeam v2.3 or later (see section 3)

If you have an earlier version of SpiraTeam, you will need to upgrade to at least v2.3 before trying to integrate with JIRA.

The next step is to configure the plug-in within SpiraTeam so that the system knows how to access the JIRA server. To start the configuration, please open up SpiraTeam in a web browser, log in using a valid account that has System-Administration level privileges and click on the System > Data Synchronization administration option from the left-hand navigation:

Plug-In	Data Mapping	Last Sync Date	Status	Operations
Jira JiraDataSync	View Project Mappings	2/24/2017 12:56:00 PM	Success	Force Re-Sync Edit Delete View Errors

This screen lists all the plug-ins already configured in the system. Depending on whether you chose the option to include sample data in your installation or not, you will see either an empty screen or a list of sample data-synchronization plug-ins.

If you already see an entry for **JiraDataSync** you should click on its “Edit” link. If you don’t see such an entry in the list, please click on the [Add] button instead. In either case you will be taken to the following screen where you can enter or modify the JIRA Data-Synchronization plug-in:

Jira Edit Plug-In

[← Back to Data Synchronization Home](#)

Please enter/edit the following information for the plug-in. The exact information that needs to be entered is specific to each plug-in, and you should refer to the documentation for the plug-in when entering/changing it:

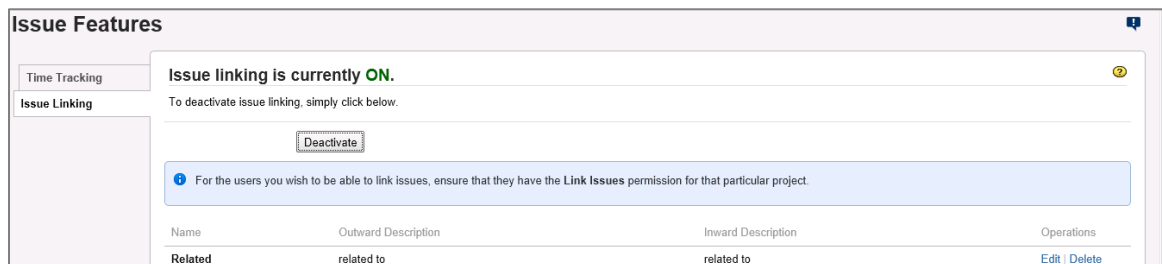
Name: *	<input type="text" value="JiraDataSync"/> <small>The physical name of the plugin (e.g. JiraDataSync) that matches the DLL being used</small>
Caption:	<input type="text" value="Jira"/> <small>The display name of the plugin (e.g. Internal Jira) that can be used to distinguish between multiple instances of the same plugin type</small>
Description:	<input type="text" value="This plug-in allows incidents in the system to be synchronized with the JIRA issue-tracking system"/> <small>The long description of the plugin (e.g. 'Connects to the in-house bug-tracker')</small>
Connection Info: *	<input type="text" value="http://dalek:8080"/> <small>The URL or other connection string used to connect to the external bug-tracker</small>
Login: *	<input type="text" value="syncuser"/>
Password:	<input type="password" value="*****"/> <small>The login and password used to connect to Jira</small>
Time Offset: *	<input type="text" value="0"/> hours <small>The number of hours to look back for changes to synchronization. Normally should be set to zero (0) unless you find changes are not being synchronized</small>
Auto-Map Users	<input type="radio"/> No
Custom 01:	<input type="text"/>
Custom 02:	<input type="text"/>
Custom 03:	<input type="text"/>
Custom 04:	<input type="text" value="5,6"/>
Custom 05:	<input type="text" value="Related"/>
	<input type="button" value="Save"/> <input type="button" value="Cancel"/>

You need to fill out the following fields for the JIRA Plug-in to operate correctly:

- **Name** – this needs to be set to **JiraDataSync**.
- **Caption** – this is the display name of the plugin. Normally you can use something generic such as “Jira”, however if you have multiple JIRA instances you might want to name it something specific such as “Jira External”. If you don’t enter a value, the display name will be “JiraDataSync”
- **Description** – this should be set to a description of the plug-in. This is an optional field that is used for documentation purposes and is not actually used by the system.
- **Connection Info** – this should be the full URL to the JIRA installation being connected to (including any custom port numbers). Entering this URL into a web browser should bring up the JIRA login page.
 - ▷ For JIRA cloud customers, it is typically of the form:
<https://mycompany.atlassian.net>
 - ▷ For JIRA server customers, it is typically of the form:
<http://myserver:8080>
- **Login** – this should be set to a valid login to the JIRA installation. The login needs to have permissions to create and view issues and versions within JIRA.
- **Password** – this should be set to the password of the login specified above.
- **Time Offset** – normally this should be set to zero, but if you find that issues being changed in JIRA are not being updated in SpiraTeam, try increasing the value as this will tell the data-synchronization plug-in to add on the time offset (in hours) when comparing date-time stamps.

Also if your JIRA installation is running on a server set to a different time-zone, then you should add in the number of hours difference between the servers' time-zones here.

- **Auto-Map Users** – This changes the way that the plugin maps users in SpiraTeam to those in JIRA:
 - ▷ **Auto-Map = True**
With this setting, all users in SpiraTeam need to have the same username as those in JIRA. If this is the case then you do not need to perform the user-mapping task outlined in section 2.2.2. This is a big time-saver if you can guarantee that all usernames are the same in both systems.
 - ▷ **Auto-Map = False**
With this setting, users in SpiraTeam and JIRA are free to have different usernames because you specify the corresponding JIRA name for each user as outlined in section 2.2.2.
- **Custom 01** – This is used to specify a JIRA custom property that should be mapped to the built-in SpiraTeam Incident Severity field (which does not exist in JIRA). This can be left empty for now and will be discussed below in section 2.2.
- **Custom 02** – This should be set to the word “True” if you want to have the new issues submitted to JIRA be submitted using a specified SecurityLevel. If you’re not using the security level feature of JIRA, leave the field blank.
- **Custom 03** – This determines how the synchronization of incidents works:
 - ▷ **Default (leave blank)** – By default, SpiraTeam will log new incidents in JIRA that were originally created in SpiraTeam. It will also log new issues from JIRA in SpiraTeam. Updates will only occur from JIRA → SpiraTeam.
This is the recommended option for most users.
 - ▷ **“True”** – If you enter the word “True” in this setting, the plugin will restrict synchronization to only loading new incidents from SpiraTeam > JIRA and updating existing items. This is useful if you want to prevent existing issues in JIRA from being loaded into SpiraTeam.
 - ▷ **“Both”** – If you enter the word “Both” in this setting, the plugin will allow full bidirectional synchronization of new incidents/issues and also updates to existing incidents/issues in both SpiraTeam and JIRA. This option should only be used if you have a well-defined set of workflows that make sense in both systems, and that do not conflict.
This option is not recommended for novice users.
- **Custom 04** – This should be set to a comma-separated list of IDs of any JIRA issue types that you want to be synchronized with SpiraTeam requirements instead of incidents. If you leave this blank, all JIRA issue types will be synchronized with incidents.
- **Custom 05** – This should be set to the name of the JIRA issue link type that you want SpiraTeam incident associations to use. If you leave this blank, incident associations in SpiraTeam will not be imported into JIRA. You can get the list of issue link types from the following screen in JIRA:



Note: For most users, we recommend leaving Custom 01 – Custom 04 blank.

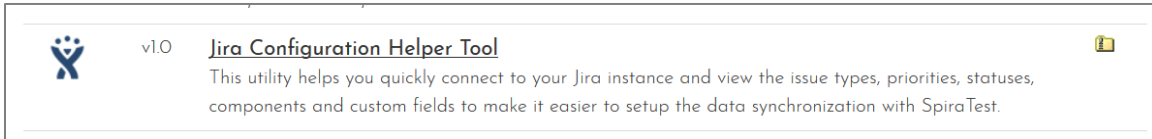
2.2. Configuring the Data Mapping

Next, you need to configure the data mapping between SpiraTeam and JIRA. This allows the various projects, users, releases, incident types, statuses, priorities and custom property values used in the two applications to be related to each other. This is important, as without a correct mapping, there is no way for the integration service to know that an “Enhancement” in SpiraTeam is the same as a “New Feature” in JIRA (for example).

The following mapping information needs to be setup in SpiraTeam:

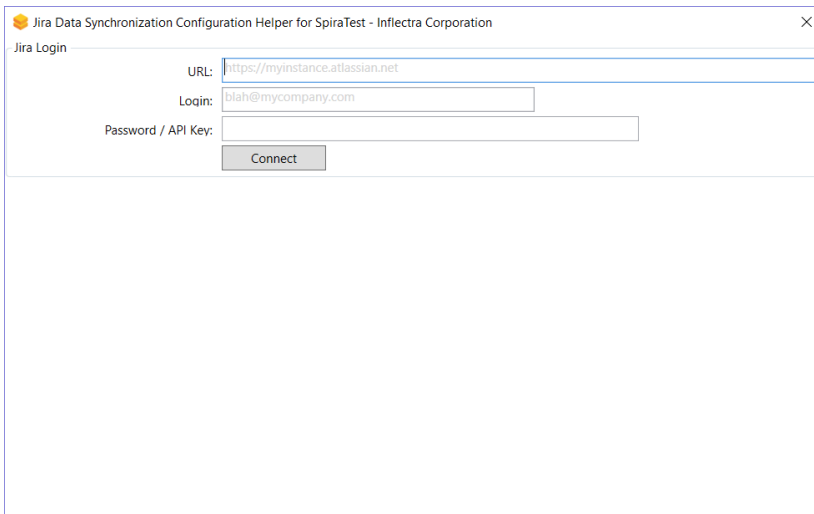
- The mapping of the project identifiers for the projects that need to be synchronized
- The mapping of users in the system
- The mapping of releases (equivalent to JIRA versions) in the system
- The mapping of the various standard fields in the system
- The mapping of the various custom properties in the system

Each of these is explained in turn below. However, to make the data mapping process easier, we have a helpful utility that will help you connect to your JIRA instance (both cloud or server) and determine the matching IDs for the various fields in JIRA:



You can download it from this URL: <https://www.inflectra.com/Downloads/JiraConfigurationHelper.zip>

Once you have downloaded and unzipped the program, run the **JiraConfigurationHelper.exe** and the following screen will be displayed:

A screenshot of the 'Jira Data Synchronization Configuration Helper for SpiraTest - Inflectra Corporation' dialog box. The title bar includes the application name and a close button. The dialog is titled 'Jira Login' and contains three input fields: 'URL:' with the value 'https://myinstance.atlassian.net', 'Login:' with the value 'blah@mycompany.com', and 'Password / API Key:'. A 'Connect' button is located below the input fields.

Enter in the URL, login and password/API Key for your JIRA instance and click **Connect**:

Description	IconUrl	Name
A problem which impairs or prevents the functions of the product.	http://dalek:8080/images/icons/issuetypes/bug.png	Bug
A new feature of the product, which has yet to be developed.	http://dalek:8080/images/icons/issuetypes/newfeature.png	New Feature
A task that needs to be done.	http://dalek:8080/images/icons/issuetypes/task.png	Task
An improvement or enhancement to an existing feature or task.	http://dalek:8080/images/icons/issuetypes/improvement.png	Improvement
Agile User Story	http://dalek:8080/images/icons/issuetypes/genericissue.png	User Story

Choose the project in JIRA that you will be connecting to SpiraTest and then the list of issue types, issue statuses, issue priorities, components, versions and custom fields will be displayed. We will be using this tool later on when you want to get some of the ID values to populate in SpiraTest.

2.2.1. Configuring the Project Mapping

From the data synchronization administration page, you need to click on the “View Project Mappings” hyperlink next to the JIRA plug-in name. This will take you to the data-mapping home page for the currently selected project:

Jira Project Data Mapping JIRA DEMO Project

← Back to Data Synchronization Home ↻ Change Project

To configure this project for data mapping with the 'Jira' plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name:* JIRA DEMO Project

External Key:* DEMO

Active:* Yes

Save

If the project name does not match the name of the project you want to configure the data-mapping for, click on the “(Change Project)” hyperlink to change the current project.

To enable this project for data-synchronization with JIRA, you need to enter:

- **External Key** – This should be set to the name of the project **Key** in JIRA. Typically, this is a short acronym for the project:

Projects						
Project list						Add Project
Name	Key	URL	Project Lead	Default Assignee	Operations	
Demonstration	DEMO	No URL	Administrator	Project Lead	Edit Delete	

Active Flag – Set this to ‘Yes’ so that SpiraTeam knows that you want to synchronize data for this project. Once the project has been completed, setting the value to “No” will stop data synchronization, reducing network utilization.

Click [Update] to confirm these settings. Once you have enabled the project for data-synchronization, you can now enter the other data mapping values outlined below.

Note: *Once you have successfully configured the project, when creating a new project, you should choose the option to “Create Project from Existing Project” rather than “Use Default Template” so that all the project mappings get copied across to the new project.*

2.2.2. Configuring the User Mapping

To configure the mapping of users in the two systems, you need to go to Administration > Users > View Edit Users, which will bring up the list of users in the system. Then click on the “Edit” button for a particular user that will be editing issues in JIRA:

Fred Bloggs Edit User

Please review the information listed below and make any changes if necessary. Once you have finished, click [Save/Insert] to confirm them.

Save Cancel

Dates for Latest Activity by User

Last Lockout: - Last Activity: 3/27/2014 1:31:00 PM Last Login: 3/27/2014 1:29:4

User Information

First Name*: Fred Department: QA
Middle Initial: Organization:

Membership and Mapping

Project Membership Data Mapping

Jira ID: fredbloggs

Click on the ‘Data Mapping’ tab to list all the configured data-synchronization plug-ins for this user. In the text box next to the JIRA Data-Sync plug-in you need to enter the login for this username in JIRA. This will allow the data-synchronization plug-in to know which user in SpiraTeam match which equivalent user in JIRA. Click [Save] once you’ve entered the appropriate login name. You should now repeat for the other users who will be active in both systems.

If you have set the “Auto-Map Users” option in the JIRA plugin, you can skip this section completely.

2.2.3. Configuring the Release Mapping

When the data-synchronization service runs, when it comes across a release/iteration in SpiraTeam that it has not seen before, it will create a corresponding “Version” in JIRA. Similarly, if it comes across a new Version in JIRA that it has not seen before, it will create a new Release in SpiraTeam. Therefore, when

using both systems together, it is recommended that you only enter new Releases/Versions in one system and let the data-synchronization service add them to the other system.

However, you may start out with the situation where you already have pre-existing Releases/Version in both systems that you need to associate in the data-mapping. If you don't do this, you may find that duplicates get created when you first enable the data-synchronization service. Therefore, for any Releases/Iterations that already exist in BOTH systems please navigate to Planning > Releases and click on the Release/Iteration in question. Make sure you have the 'Overview' tab visible and expand the "Details" section of the release/iteration:

Release 1.0 Release 1.0 [RL:21]

Name: Release 1.0

Overview Incidents Reqs & Tasks Test Cases Test Runs Attachments History

Details

Status: In Progress Creator: System Administrator Version #: 1.0

Creation Date: 4/11/2013 4:58:18 PM Start Date: 4/10/2013 End Date: 4/15/2013

Resources: 1 Non-Working: 0 person days Plan Effort: 32.0h

Owner: -- None -- Last Updated: 2/23/2017 3:46:10 PM Type: Major Release

Available Effort: 32.0h

Jira ID: 10000

In addition to the standard fields and custom properties, you will see an additional text property called "Jira ID" that is used to store the mapped external identifier for the equivalent Version in JIRA. You need to locate the ID of the equivalent version in JIRA, enter it into this text-box and click [Save]. You should now repeat for all the other pre-existing releases.

The JIRA ID can be found using the Jira Configuration Helper using the **Versions** tab:

Jira Data Synchronization Configuration Helper for SpiraTest - Inflectra Corporation

Jira Login

URL: http://dalek:8080

Login: administrator

Password / API Key:

Connect

Data Mapping Helper

Project: DEMO - Demonstration

Name	Description	ReleaseDate	Archived	Released	Overdue	Project	Id	IdString
1.0	Version 1.0		<input type="checkbox"/>	<input type="checkbox"/>			10000	10000
1.1	Version 1.1	4/9/2013 12:00:00 AM	<input type="checkbox"/>	<input type="checkbox"/>	True		10001	10001
2.0	Version 2.0		<input type="checkbox"/>	<input type="checkbox"/>			10100	10100

2.2.4. Configuring the Standard Field Mapping

Now that the projects, user and releases have been mapped correctly, we need to configure the standard incident and requirement fields. To do this, go to Administration > System > Data Synchronization and click on the "View Project Mappings" for the JiraDataSync plug-in entry:

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact Type	Standard Fields	Custom Properties
Requirement	Component Importance Status Type	JIRA Key
Test Case	Component Priority Status Type	
Incident	Component Priority Severity Status Type	Custom Date Free Text Field Multi Select Field Single Select List Custom User Environment Resolution JIRA Key Radio Value
Release		
Task	Component Priority Status Type	

From this screen, you need to click on Priority, Severity, Component, Status and Type in turn to configure the incident field mappings. If you're using the option to have JIRA also synchronize some issue types as requirements, then you'll need to also configure the Requirement Importance, Type, Component and Status fields.

a) Incident Type

Click on the "Type" hyperlink under Incident Standard Fields to bring up the Incident type mapping configuration screen:

Edit Standard Field Data Mapping JIRA DEMO Project

[← Back to Project Mappings](#)
[↻ Change Project](#)

To configure the data mappings for this field, please fill in the external key values for the Jira plug-in next to each SpiraTeam value:

Artifact Type: **Incident**
Field Name: **Type**

Type	Active	External Key <small>Jira</small>	Primary
Bug	Yes	<input type="text" value="1"/>	<input checked="" type="checkbox"/> Yes
Change Request	Yes	<input type="text" value="4"/>	<input checked="" type="checkbox"/> Yes
Enhancement	Yes	<input type="text" value="2"/>	<input checked="" type="checkbox"/> Yes
Incident	Yes	<input type="text" value="1"/>	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No
Issue	Yes	<input type="text"/>	<input checked="" type="checkbox"/> Yes
Limitation	Yes	<input type="text"/>	<input checked="" type="checkbox"/> Yes
New Value	No	<input type="text"/>	<input checked="" type="checkbox"/> Yes
Risk	Yes	<input type="text"/>	<input checked="" type="checkbox"/> Yes
Task	Yes	<input type="text" value="3"/>	<input checked="" type="checkbox"/> Yes
Training	Yes	<input type="text"/>	<input checked="" type="checkbox"/> Yes

The table lists each of the incident types available in SpiraTeam and provides you with the ability to enter the matching JIRA issue type ID for each one. You can map multiple SpiraTeam fields to the same JIRA fields (e.g. Bug and Incident in SpiraTeam are both equivalent to Bug in JIRA), in which case only one of

the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from JIRA > SpiraTeam).

The JIRA ID can be found by using the **Issue Types** tab of the Jira configuration helper:

IconUrl	Name	SubTask	Fields	Id	IdString
http://dalek:8080/images/icons/issuetypes/bug.png	Bug	<input type="checkbox"/>	(Collection)	1	1
http://dalek:8080/images/icons/issuetypes/newfeature.png	New Feature	<input type="checkbox"/>	(Collection)	2	2
http://dalek:8080/images/icons/issuetypes/task.png	Task	<input type="checkbox"/>	(Collection)	3	3
http://dalek:8080/images/icons/issuetypes/improvement.png	Improvement	<input type="checkbox"/>	(Collection)	4	4
http://dalek:8080/images/icons/issuetypes/genericissue.png	User Story	<input type="checkbox"/>	(Collection)	6	6

b) Incident Status

Click on the “Status” hyperlink under Incident Standard Fields to bring up the Incident status mapping configuration screen:

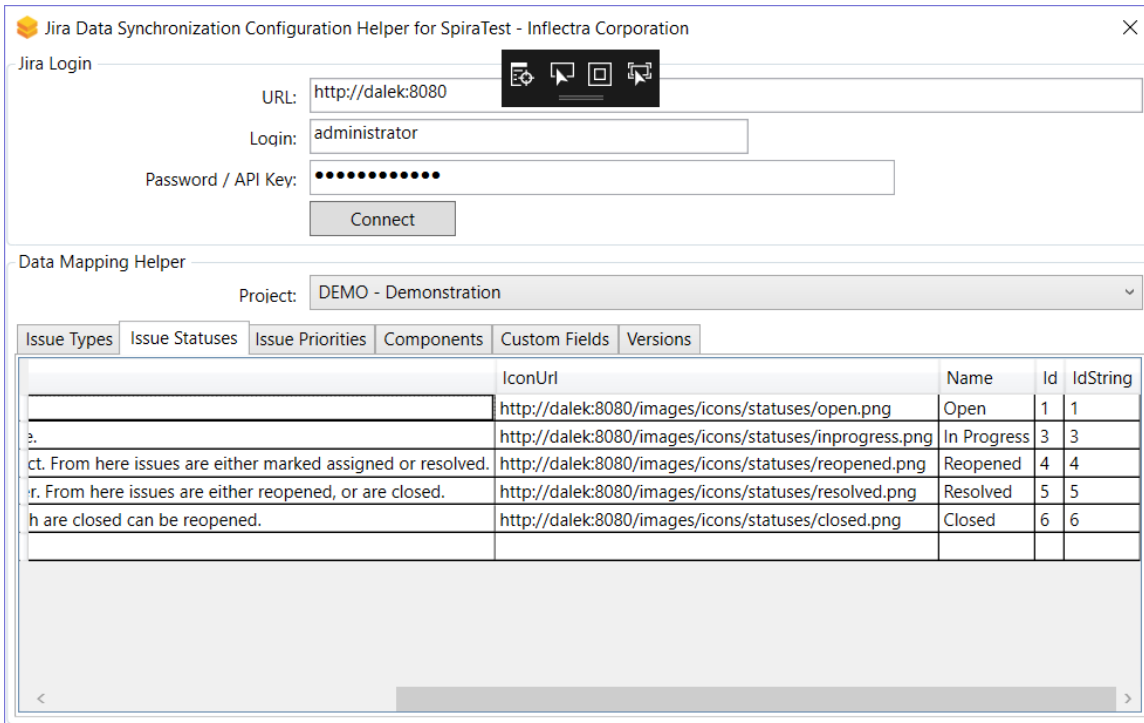
Status	Active	External Key Jira	Primary
Assigned	Yes	3	<input checked="" type="radio"/> Yes <input type="radio"/> No
Closed	Yes	6	<input checked="" type="radio"/> Yes <input type="radio"/> No
Duplicate	Yes		<input checked="" type="radio"/> Yes <input type="radio"/> No
New	Yes	1	<input type="radio"/> Yes <input checked="" type="radio"/> No
Not Reproducible	Yes		<input checked="" type="radio"/> Yes <input type="radio"/> No
Open	Yes	1	<input checked="" type="radio"/> Yes <input type="radio"/> No
Reopen	Yes	4	<input checked="" type="radio"/> Yes <input type="radio"/> No
Resolved	Yes	5	<input checked="" type="radio"/> Yes <input type="radio"/> No

The table lists each of the incident statuses available in SpiraTeam and provides you with the ability to enter the matching JIRA issue status ID for each one. You can map multiple SpiraTeam fields to the same JIRA fields (e.g. New and Open in SpiraTeam are both equivalent to Open in JIRA), in which case

only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from JIRA > SpiraTeam).

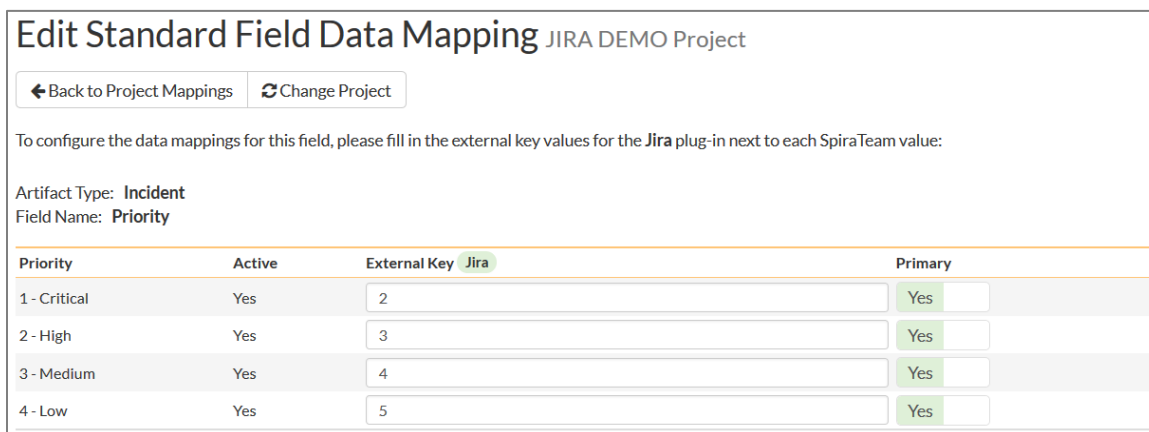
We recommend that you always point the New and Open statuses inside SpiraTeam to point to the ID for "Open" inside JIRA and make Open in SpiraTeam the Primary status of the two. This is recommended so that as new incidents in SpiraTeam get synched over to JIRA, they will get switched to the Open status in JIRA which will then be synched back to "Open" in SpiraTeam. That way you'll be able to see at a glance which incidents have been synched with JIRA and those that haven't.

The JIRA ID can be found by using the **Issue Statuses** tab of the Jira configuration helper:



c) Incident Priority

Click on the "Priority" hyperlink under Incident Standard Fields to bring up the Incident Priority mapping configuration screen:



The table lists each of the incident priorities available in SpiraTeam and provides you with the ability to enter the matching JIRA priority ID for each one. You can map multiple SpiraTeam fields to the same

JIRA fields, in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from JIRA > SpiraTeam).

The JIRA ID can be found by using the **Issue Priorities** tab of the Jira configuration helper:

self	iconUrl	name	id
http://dalek:8080/rest/api/2/priority/1	http://dalek:8080/images/icons/priorities/blocker.png	Blocker	1
http://dalek:8080/rest/api/2/priority/2	http://dalek:8080/images/icons/priorities/critical.png	Critical	2
http://dalek:8080/rest/api/2/priority/3	http://dalek:8080/images/icons/priorities/major.png	Major	3
http://dalek:8080/rest/api/2/priority/4	http://dalek:8080/images/icons/priorities/minor.png	Minor	4
http://dalek:8080/rest/api/2/priority/5	http://dalek:8080/images/icons/priorities/trivial.png	Trivial	5

d) Incident Component (Optional)

Click on the “Component” hyperlink under Incident Standard Fields to bring up the Incident component mapping configuration screen:

Component	Active	External Key Jira	Primary
Component 1	Yes	10000	<input checked="" type="checkbox"/>
Component 2	Yes	10001	<input checked="" type="checkbox"/>
Component 3	Yes	10100	<input checked="" type="checkbox"/>

The table lists each of the components available in SpiraTeam and provides you with the ability to enter the matching JIRA component ID for each one. You can map multiple SpiraTeam fields to the same JIRA fields, in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from JIRA > SpiraTeam).

The JIRA ID can be found by using the **Components** tab of the Jira configuration helper:

:

e) Incident Severity (Optional)

Click on the “Severity” hyperlink under Incident Standard Fields to bring up the Incident severity mapping configuration screen:

Severity	Active	External Key Jira	Primary
1 - Critical	Yes	Critical	<input checked="" type="checkbox"/>
2 - High	Yes	High	<input checked="" type="checkbox"/>
3 - Medium	Yes	Medium	<input checked="" type="checkbox"/>
4 - Low	Yes	Low	<input checked="" type="checkbox"/>

Unlike the other incident standard fields, JIRA doesn't actually have a built-in field for storing the severity of an issue, so if you want to be able to see the SpiraTeam incident severity in JIRA, you'll need to create a JIRA custom list field to store the different severity values. If you don't want to synchronize severity values with JIRA, you can skip the rest of this section.

Once you have created a custom field in JIRA to contain the list of severity values, you need to now populate the above table **with the name** (Not the ID) of the severity custom property values inside JIRA and click Update. Secondly you need to go to the Plug-in configuration screen:

Custom 01:	<input type="text" value="10001"/>
Custom 02:	<input type="text"/>
Custom 03:	<input type="text"/>
Custom 04:	<input type="text" value="5,6"/>
Custom 05:	<input type="text" value="Related"/>

On this screen you need to enter the ID of the custom field that you're using to store severities in JIRA and populate the **Custom 01 property with this value** (see above). The ID can be found by using the Custom Fields tab of the Jira Configuration Helper:

Jira Data Synchronization Configuration Helper for SpiraTest - Inflectra Corporation ✕

Jira Login

URL:

Login:

Password / API Key:

Data Mapping Helper

Project:

Issue Types | Issue Statuses | Issue Priorities | Components | **Custom Fields** | Versions

Name	Id
User Custom	10101
Custom Date	10100
Radios	10200
Severity	10300
Multi Select Field	10001
Single Select List	10002
Free Text Field	10000

e) Requirement Status (Optional)

Click on the “Status” hyperlink under Requirement Standard Fields to bring up the Requirement status mapping configuration screen:

Edit Standard Field Data Mapping JIRA DEMO Project

[← Back to Project Mappings](#)
[↻ Change Project](#)

To configure the data mappings for this field, please fill in the external key values for the Jira plug-in next to each SpiraTeam value:

Artifact Type: **Requirement**
Field Name: **Status**

Status	Active	External Key Jira	Primary
Accepted	Yes	<input type="text" value="1"/>	<input checked="" type="checkbox"/>
Completed	Yes	<input type="text"/>	<input checked="" type="checkbox"/>
Developed	Yes	<input type="text" value="6"/>	<input checked="" type="checkbox"/>
In Progress	Yes	<input type="text" value="3"/>	<input checked="" type="checkbox"/>
Obsolete	Yes	<input type="text"/>	<input checked="" type="checkbox"/>
Planned	Yes	<input type="text" value="5"/>	<input checked="" type="checkbox"/>
Rejected	Yes	<input type="text"/>	<input checked="" type="checkbox"/>
Requested	Yes	<input type="text"/>	<input checked="" type="checkbox"/>
Tested	Yes	<input type="text"/>	<input checked="" type="checkbox"/>
Under Review	Yes	<input type="text"/>	<input checked="" type="checkbox"/>

The table lists each of the requirement statuses available in SpiraTeam and provides you with the ability to enter the matching JIRA issue status ID for each one. You can map multiple SpiraTeam fields to the same JIRA fields, in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from JIRA > SpiraTeam).

The JIRA ID can be found by using the **Issue Statuses** tab of the Jira configuration helper:

Jira Data Synchronization Configuration Helper for SpiraTest - Inflectra Corporation ✕

Jira Login

URL:

Login:

Password / API Key:

Data Mapping Helper

Project:

	IconUrl	Name	Id	IdString
	http://dalek:8080/images/icons/statuses/open.png	Open	1	1
	http://dalek:8080/images/icons/statuses/inprogress.png	In Progress	3	3
ct. From here issues are either marked assigned or resolved.	http://dalek:8080/images/icons/statuses/reopened.png	Reopened	4	4
r. From here issues are either reopened, or are closed.	http://dalek:8080/images/icons/statuses/resolved.png	Resolved	5	5
h are closed can be reopened.	http://dalek:8080/images/icons/statuses/closed.png	Closed	6	6

f) Requirement Importance (Optional)

Click on the "Importance" hyperlink under Requirement Standard Fields to bring up the Requirement Importance mapping configuration screen:

Edit Standard Field Data Mapping JIRA DEMO Project

[← Back to Project Mappings](#) [↻ Change Project](#)

To configure the data mappings for this field, please fill in the external key values for the **Jira** plug-in next to each SpiraTeam value:

Artifact Type: **Requirement**

Field Name: **Importance**

Importance	Active	External Key	Jira	Primary
1 - Critical	Yes	<input type="text" value="2"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2 - High	Yes	<input type="text" value="3"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3 - Medium	Yes	<input type="text" value="4"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4 - Low	Yes	<input type="text" value="5"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

The table lists each of the requirement importances available in SpiraTeam and provides you with the ability to enter the matching JIRA priority ID for each one. You can map multiple SpiraTeam fields to the same JIRA fields, in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from JIRA > SpiraTeam).

The JIRA ID can be found by using the **Issue Priorities** tab of the Jira configuration helper:

self	iconUrl	name	id
http://dalek:8080/rest/api/2/priority/1	http://dalek:8080/images/icons/priorities/blocker.png	Blocker	1
http://dalek:8080/rest/api/2/priority/2	http://dalek:8080/images/icons/priorities/critical.png	Critical	2
http://dalek:8080/rest/api/2/priority/3	http://dalek:8080/images/icons/priorities/major.png	Major	3
http://dalek:8080/rest/api/2/priority/4	http://dalek:8080/images/icons/priorities/minor.png	Minor	4
http://dalek:8080/rest/api/2/priority/5	http://dalek:8080/images/icons/priorities/trivial.png	Trivial	5

g) Requirement Type (Optional)

Click on the "Type" hyperlink under Requirement Standard Fields to bring up the Requirement type mapping configuration screen:

Edit Standard Field Data Mapping JIRA DEMO Project

[← Back to Project Mappings](#)
[↻ Change Project](#)

To configure the data mappings for this field, please fill in the external key values for the Jira plug-in next to each SpiraTeam value:

Artifact Type: **Requirement**
Field Name: **Type**

Type	Active	External Key	Jira	Primary
Design Element	Yes			<input checked="" type="checkbox"/>
Feature	Yes			<input checked="" type="checkbox"/>
Need	Yes			<input checked="" type="checkbox"/>
Package	Yes			<input checked="" type="checkbox"/>
Quality	Yes			<input checked="" type="checkbox"/>
Use Case	Yes	5		<input checked="" type="checkbox"/>
User Story	Yes	6		<input checked="" type="checkbox"/>

The table lists each of the requirement types available in SpiraTeam and provides you with the ability to enter the matching JIRA issue type ID for each one. You can map multiple SpiraTeam fields to the same JIRA fields (e.g. Use Case and User Story in SpiraTeam are both equivalent to User Story in JIRA), in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from JIRA > SpiraTeam).

The JIRA ID can be found by using the **Issue Types** tab of the Jira configuration helper:

IconUrl	Name	SubTask	Fields	Id	IdString
http://dalek:8080/images/icons/issuetypes/bug.png	Bug	<input type="checkbox"/>	(Collection)	1	1
http://dalek:8080/images/icons/issuetypes/newfeature.png	New Feature	<input type="checkbox"/>	(Collection)	2	2
http://dalek:8080/images/icons/issuetypes/task.png	Task	<input type="checkbox"/>	(Collection)	3	3
http://dalek:8080/images/icons/issuetypes/improvement.png	Improvement	<input type="checkbox"/>	(Collection)	4	4
http://dalek:8080/images/icons/issuetypes/genericissue.png	User Story	<input type="checkbox"/>	(Collection)	6	6

h) Requirement Component (Optional)

Click on the "Component" hyperlink under Requirement Standard Fields to bring up the Requirement component mapping configuration screen:

Edit Standard Field Data Mapping JIRA DEMO Project

[← Back to Project Mappings](#)
[↻ Change Project](#)

To configure the data mappings for this field, please fill in the external key values for the Jira plug-in next to each SpiraTeam value:

Artifact Type: **Requirement**
Field Name: **Component**

Component	Active	External Key <small>Jira</small>	Primary
Component 1	Yes	<input type="text" value="10000"/>	<input checked="" type="checkbox"/>
Component 2	Yes	<input type="text" value="10001"/>	<input checked="" type="checkbox"/>
Component 3	Yes	<input type="text" value="10100"/>	<input checked="" type="checkbox"/>

The table lists each of the components available in SpiraTeam and provides you with the ability to enter the matching JIRA component ID for each one. You can map multiple SpiraTeam fields to the same JIRA fields, in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from JIRA > SpiraTeam).

The JIRA ID can be found by using the **Components** tab of the Jira configuration helper:

:

Name	Description	Id	IdString
Component 1	The first component	10000	10000
Component 2	The second component	10001	10001
Component 3	The third component	10100	10100

2.2.5. Configuring the Custom Property Mapping

Now that the various SpiraTeam standard fields have been mapped correctly, we need to configure the custom property mappings. This is used for both custom properties in SpiraTeam that map to custom fields in JIRA and also for custom properties in SpiraTeam that are used to map to standard fields in JIRA (Environment, Resolution and SecurityLevel) that don't exist in SpiraTeam.

From the View/Edit Project Data Mapping screen, you need to click on the name of the Incident or Requirement Custom Property that you want to add data-mapping information for. We will consider the four different types of mapping that you might want to enter:

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact Type	Standard Fields	Custom Properties
Requirement	Component Importance Status Type	JIRA Key
Test Case	Component Priority Status Type	
Incident	Component Priority Severity Status Type	Custom Date Free Text Field Multi Select Field Single Select List Custom User Environment Resolution JIRA Key Radio Value
Release		
Task	Component Priority Status Type	

a) Scalar Custom Properties

This refers to custom properties that have a simple user-entered value and don't need to have their specific options mapped between SpiraTeam and JIRA. All of the custom property types except **List** and **Multi-List** fall into this category (e.g. Text, Date, User, Boolean, Decimal, Integer, etc.)

Click on the hyperlink of the scalar custom property under Incident/Requirement Custom Properties to bring up the custom property mapping configuration screen. For scalar custom properties, there will be no values listed in the lower half of the screen.

Edit Custom Property Data Mapping JIRA DEMO Project

[← Back to Project Mappings](#) [↻ Change Project](#)

To configure the data mappings for this custom property, please fill in the **Jira** external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Artifact Type:* Incident

Name:* Free Text Field Text

External Key:*

You need to lookup the ID of the custom field in JIRA that matches this custom property in SpiraTeam. Once you have entered the id of the custom field, click [Update].

The ID can be found by using the Custom Fields tab of the Jira Configuration Helper:

Jira Data Synchronization Configuration Helper for SpiraTest - Inflectra Corporation

Jira Login

URL:

Login:

Password / API Key:

Data Mapping Helper

Project:

Issue Types | Issue Statuses | Issue Priorities | Components | Custom Fields | Versions

Name	Id
User Custom	10101
Custom Date	10100
Radios	10200
Severity	10300
Multi Select Field	10001
Single Select List	10002
Free Text Field	10000

b) List Custom Properties

This refers to custom properties that are either of type **List** or **Multi-List**. Click on the hyperlink of the list custom property under Incident/Requirement Custom Properties to bring up the custom property mapping configuration screen. For list custom properties there will be a textbox for both the custom field itself and a mapping table for each of the custom property values that need to be mapped:

Edit Custom Property Data Mapping JIRA DEMO Project

To configure the data mappings for this custom property, please fill in the Jira external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Artifact Type:* Incident

Name:* Multi Select Field **MultiList**

External Key:*

Custom Value Mapping

Value	Active	External Key Jira
Value 1	Yes	<input type="text" value="Value 1"/>
Value 2	Yes	<input type="text" value="Value 2"/>
Value 3	Yes	<input type="text" value="Value 3"/>
Value 4	Yes	<input type="text" value="Value 4"/>

First you need to lookup the ID of the custom field in JIRA that matches this custom property in SpiraTeam. This should be entered in the 'External Key' field below the name of the custom property. The ID can be found by using the Custom Fields tab of the Jira Configuration Helper:

Jira Data Synchronization Configuration Helper for SpiraTest - Inflectra Corporation

Jira Login

URL:

Login:

Password / API Key:

Data Mapping Helper

Project:

Issue Types | Issue Statuses | Issue Priorities | Components | **Custom Fields** | Versions

Name	Id
User Custom	10101
Custom Date	10100
Radios	10200
Severity	10300
Multi Select Field	10001
Single Select List	10002
Free Text Field	10000

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the full name (not the id this time) of the custom field value as specified in JIRA:

Fields

Custom Fields

Field Configurations

Field Configuration...

Configure Custom Field: Multi Select Field

Below are the Custom Field Configuration schemes for this custom field. Schemes are applicable for various issue types in a particular context. You can configure a custom field differently for each project context or in a global context. Moreover, project level schemes will over-ride global ones.

- [Add new context](#)
- [View Custom Fields](#)

Default Configuration Scheme for Multi Select Field

Default configuration scheme generated by JIRA

Applicable contexts for scheme: [Edit Configuration](#)

Issue type(s):
Global (all issues)

Default Value: [Edit Default Value](#)

Options: [Edit Options](#)

- Value 1
- Value 2
- Value 3
- Value 4

c) JIRA's Resolution Field

If you would like the values of the JIRA '**Resolution**' field to be synchronized back to SpiraTeam, then you will need to fill out this section. You first need to create an incident custom property in SpiraTeam of type 'LIST' that contains the various resolution names that exist inside JIRA.

Then click on the hyperlink of this new list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen:

Edit Custom Property Data Mapping JIRA DEMO Project

[← Back to Project Mappings](#) [↻ Change Project](#)

To configure the data mappings for this custom property, please fill in the **Jira** external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Artifact Type:* Incident
Name:* Resolution **List**
External Key:*

Custom Value Mapping

Value	Active	External Key Jira
Cannot Reproduce	Yes	<input type="text" value="5"/>
Fixed	Yes	<input type="text" value="1"/>
Won't Fix	Yes	<input type="text" value="2"/>

First you need to enter the word “Resolution” as the External Key of the custom property. This tells the data-sync plug-in that the custom property in SpiraTeam should be mapped to built-in Resolution field in JIRA.

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the JIRA ID of the various Resolutions that are configured in JIRA. The external ID can be found by looking at the URL inside JIRA which choosing to View/Edit the resolution name/description.

d) JIRA's Environment Field

If your instance of JIRA requires that all new issues are submitted with an ‘Environment’ description specified, then you will need to fill out this section. You first need to create an incident custom property in SpiraTeam of type ‘TEXT’ that will be used to store the environment description within SpiraTeam.

Then click on the hyperlink of this new list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen:

Edit Custom Property Data Mapping JIRA DEMO Project

[← Back to Project Mappings](#) [↻ Change Project](#)

To configure the data mappings for this custom property, please fill in the **Jira** external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Artifact Type:* Incident
Name:* Environment **Text**
External Key:*

All you need to do on this screen is enter the word “**Environment**” in the External Key textbox and the data-sync plug-in will know that this custom property is mapped to the built-in Environment field in JIRA.

e) JIRA's Security Level Field

If your instance of JIRA requires that all new issues are submitted with a 'Security Level' then you will need to fill out this section. You first need to create an incident custom property in SpiraTeam of type 'LIST' that contains the various security levels that exist inside JIRA.

Then click on the hyperlink of this new list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen.

First you need to enter the word "**SecurityLevel**" as the External Key of the custom property. This tells the data-sync plug-in that the custom property in SpiraTeam should be mapped to built-in Security Level field in JIRA.

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the JIRA ID of the various Security Levels that are configured in JIRA. The external ID can be found by looking at the URL inside JIRA which choosing to View/Edit the security level name/description.

f) JIRA's Issue Key Field

It can be convenient to create a SpiraTeam custom property to store the JIRA Issue Key (the ID used to identify an issue in JIRA). This allows you to display a list of incidents in SpiraTest and see the corresponding JIRA ID in the same list. You first need to create an incident custom property in SpiraTeam of type 'TEXT' that will be used to store the JIRA issue key within SpiraTeam.

Then click on the hyperlink of this new list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen:

The screenshot shows a web interface titled "Edit Custom Property Data Mapping" for the "JIRA DEMO Project". At the top, there are two buttons: "← Back to Project Mappings" and "↻ Change Project". Below these is a paragraph of instructions: "To configure the data mappings for this custom property, please fill in the Jira external key values for the custom property below. If this is a custom list property, you also need to fill in the external key for each of the custom property values as well:". The form contains three rows of input fields: "Artifact Type:" with the value "Incident", "Name:" with the value "JIRA Key Text", and "External Key:" with the value "JiralssueKey". At the bottom left of the form is an orange "Update" button.

All you need to do on this screen is enter the word "**JiralssueKey**" in the External Key textbox and the data-sync plug-in will know that this custom property is mapped to the built-in Issue Key field in JIRA.

Once you have updated the various mapping sections, you are now ready to use the integration.

2.3. Using SpiraTeam with JIRA

Now that the integration service has been configured and the service started, initially any incidents created in SpiraTeam for the specified projects will be imported into JIRA and any existing issues in JIRA will get loaded into SpiraTeam as either incidents or requirements (depending on your configuration).

At this point we recommend opening the Windows Event Viewer and choosing the Application Log. In this log any error messages raised by the SpiraTeam Data Sync Service will be displayed. If you see any error messages at this point, we recommend immediately stopping the SpiraTeam service and checking the various mapping entries. If you cannot see any issues with the mapping information, we recommend sending a copy of the event log message(s) to Inflectra customer services (support@inflectra.com) who will help you troubleshoot the problem.

To use SpiraTeam with JIRA on an ongoing basis, we recommend the following general processes be followed:

- When running tests in SpiraTest or SpiraTeam, defects found should be logged through the Test Execution Wizard as normal.
- Developers can log new defects into either SpiraTeam or JIRA. In either case they will get loaded into the other system.
- Once created in one of the systems and successfully replicated to the other system, the incident should not be modified again inside SpiraTeam
- At this point, the incident should not be acted upon inside SpiraTeam and all data changes to the issue should be made inside JIRA. To enforce this, you should modify the workflows set up in SpiraTeam so that the various fields are marked as inactive for all the incident statuses other than the “New” status. This will allow someone to submit an incident in SpiraTeam, but will prevent them making changes in conflict with JIRA after that point.
- As the issue progresses through the customized JIRA workflow, changes to the type of issue, changes to its status, priority, description and resolution will be updated automatically in SpiraTeam. In essence, SpiraTeam acts as a read-only viewer of these incidents.
- You are now able to perform test coverage and incident reporting inside SpiraTest/SpiraTeam using the test cases managed by SpiraTest/SpiraTeam and the incidents managed on behalf of SpiraTest/SpiraTeam inside JIRA.

2.4. Using the Jira Cloud Connector

Once you have the data-synchronization established between SpiraTest and JIRA, we have an additional Atlassian marketplace connector that you can use (see <https://marketplace.atlassian.com/apps/1218742/spiratest-app-for-jira>):

The screenshot displays the Jira interface with a 'SpiraTest Connection Settings' dialog box open. The dialog box contains the following information:

- SpiraTest Connection Settings**
- Make Sure You Are Using SpiraTest 5.0 or Later**
- Information: "This add-on connects to version 5.0+ of SpiraTest (or SpiraTeam, or SpiraPlan). You will also need to configure the data-sync between SpiraTest and Jira. This is done from within SpiraTest and you can read more about that on our website."
- SpiraTest URL***:
- Note: "For Jira to access SpiraTest it must be publicly accessible"
- Username***:
- API key / RSS Token***:
- Save** button

In the background, a Jira issue page is visible, showing a 'SpiraTest Test Coverage' widget with a green progress bar and the text "2 Tests: 1 (50%) passed." Other visible elements include 'votes: 0', 'Watchers: 1 Stop watching this issue', and 'Activity' tabs (All, Comments, Work log, History, Activity).

You can install the connector by following these instructions:

1. Log into your Jira instance as an admin.
2. Click the admin dropdown and choose Add-ons. The Find new apps or Find new add-ons screen loads.
3. Locate SpiraTest app for Jira.

4. Click Install to download and install your app.
5. Click Close in the “Installed and ready to go” dialog.
6. Now you need to configure the add-on to connect to your SpiraTest instance.

SpiraTest Connection Settings

Make Sure You Are Using SpiraTest 5.0 or Later

This add-on connects to version 5.0+ of SpiraTest (or SpiraTeam, or SpiraPlan). You will also need to configure the data-sync between SpiraTest and Jira. This is done from within SpiraTest and you can read more about that [on our website](#).

SpiraTest URL *
For Jira to access SpiraTest it must be publicly accessible

Username *

API key / RSS Token *

Please enter the following information:

- **SpiraTest URL:** this needs to be the base URL for your SpiraTest instance, typically of the form:
 - <https://mysite.spiraservice.net>
 - <https://demo.spiraservice.net/mysite>
- **Username:** This is the login you use to connect to SpiraTest
- **API Key / RSS Token:** This is the RSS Token / API key for the user name you specified.

You can get the SpiraTest API Key from within the User Profile screen of SpiraTest:

User Name/ID: fredbloggs [US:2]

First Name: * Fred

Middle Initial: _____

Last Name: * Bloggs

Enable RSS Feeds: Yes

RSS Token: (7A05FD06-83C3-4436-B37F-51BCF0060483)

Department: QA

Organization: _____

Start Page: * My Page

3. Using SpiraTeam with JIRA 3 / 4

This section outlines how to use SpiraTest, SpiraPlan or SpiraTeam (hereafter referred to as SpiraTeam) in conjunction with the JIRA issue/bug tracking system versions 3.0 – 4.0. The built-in integration service allows the quality assurance team to manage their requirements and test cases in SpiraTeam, execute test runs in SpiraTest, and then have the new incidents generated during the run be automatically loaded into JIRA. Once the incidents are loaded into JIRA as issues, the development team can then manage the lifecycle of these issues in JIRA, and have the status changes in JIRA be reflected back in SpiraTeam.

In addition, if you are using JIRA 4.x or higher, any issues logged directly into JIRA will get imported into SpiraTeam so that they can be linked to test cases and requirements.

► **STOP! Please make sure you have first read the Instructions in Section 1 before proceeding!**

3.1. Configuring the Plug-In

The next step is to configure the plug-in within SpiraTeam so that the system knows how to access the JIRA server. To start the configuration, please open up SpiraTeam in a web browser, log in using a valid account that has System-Administration level privileges and click on the System > Data Synchronization administration option from the left-hand navigation:

Data Synchronization

SpiraTeam is able to communicate with a variety of external systems and applications using its background data synchronization service and library of available plug-ins.

This page allows you to view, add and modify the list of plug-ins, make changes to their configuration, and verify the status of the most recent synchronization. In addition, you can force a particular plug-in to perform a complete data-synchronization by resetting its date of last sync:

[Refresh](#) [+ Add](#)

Plug-In	Data Mapping	Last Sync Date	Status	Operations
BugzillaDataSync	View Project Mappings	6/21/2016 1:09:56 PM	Failure	Force Re-Sync Edit Delete View Errors
FogBugzDataSync	View Project Mappings	6/21/2016 1:09:56 PM	Failure	Force Re-Sync Edit Delete View Errors
JiraDataSync	View Project Mappings	6/21/2016 1:09:56 PM	Warning	Force Re-Sync Edit Delete View Errors
MsTfsDataSync	View Project Mappings	6/21/2016 1:09:56 PM	Warning	Force Re-Sync Edit Delete View Errors

This screen lists all the plug-ins already configured in the system. Depending on whether you chose the option to include sample data in your installation or not, you will see either an empty screen or a list of sample data-synchronization plug-ins.

If you already see an entry for **JiraDataSync** you should click on its “Edit” link. If you don’t see such an entry in the list, please click on the [Add] button instead. In either case you will be taken to the following screen where you can enter or modify the JIRA Data-Synchronization plug-in:

Edit Plug-In | JiraDataSync

[<< Back to Data Synchronization Home](#)

Please enter/edit the following information for the plug-in. The exact information that needs to be entered is specific to each plug-in, and you should refer to the documentation for the plug-in when entering/changing it:

Name*:

Description:

Connection Info*:

Login*:

Password:

Time Offset*:

Auto-Map Users

Custom 01:

Custom 02:

Custom 03:

Custom 04:

Custom 05:

You need to fill out the following fields for the JIRA Plug-in to operate correctly:

- **Name** – this needs to be set to **JiraDataSync**. This needs to match the name of the plug-in DLL assembly that was copied into the C:\Program Files\SpiraTeam\Bin folder (minus the .dll file extension). If you renamed the JiraDataSync.dll file for any reason, then you need to change the name here to match.
- **Description** – this should be set to a description of the plug-in. This is an optional field that is used for documentation purposes and is not actually used by the system.
- **Connection Info** – this should be the full URL to the JIRA installation's web-service API. This is typically <http://<jira server name>/rpc/soap/jirasoapervice-v2>.
- **Login** – this should be set to a valid login to the JIRA installation. The login needs to have permissions to create and view issues and versions within JIRA.
- **Password** – this should be set to the password of the login specified above.
- **Time Offset** – normally this should be set to zero, but if you find that issues being changed in JIRA are not being updated in SpiraTeam, try increasing the value as this will tell the data-synchronization plug-in to add on the time offset (in hours) when comparing date-time stamps. Also if your JIRA installation is running on a server set to a different time-zone, then you should add in the number of hours difference between the servers' time-zones here.

The remaining fields work differently depending on which version of the plugin you are using (JIRA 3.x or JIRA 4.x):

a) JIRA 3.x Plugin

Please fill out the fields as follows:

- **Auto-Map Users** – this is not currently used and can be ignored.

- **Custom 01** – This is used to specify a JIRA custom property that should be mapped to the built-in SpiraTeam Incident Severity field (which does not exist in JIRA). This can be left empty for now and will be discussed below in section 3.2.
- **Custom 02 – 05** – these are not currently used by the plug-in and should be left blank.

b) JIRA 4.x Plugin

Please fill out the fields as follows:

- **Auto-Map Users** – This changes the way that the plugin maps users in SpiraTeam to those in JIRA:
 - ▷ **Auto-Map = True**
With this setting, all users in SpiraTeam need to have the same username as those in JIRA. If this is the case then you do not need to perform the user-mapping task outlined in section 3.2.2. This is a big time-saver if you can guarantee that all usernames are the same in both systems.
 - ▷ **Auto-Map = False**
With this setting, users in SpiraTeam and JIRA are free to have different usernames because you specify the corresponding JIRA name for each user as outlined in section 3.2.2.
- **Custom 01** – This is used to specify a JIRA custom property that should be mapped to the built-in SpiraTeam Incident Severity field (which does not exist in JIRA). This can be left empty for now and will be discussed below in section 3.2.
- **Custom 02** – This should be set to the word “True” if you want to have the new issues submitted to JIRA be submitted using a specified SecurityLevel. If you’re not using the security level feature of JIRA, leave the field blank.
- **Custom 03** – This should be set to the word “True” if you want to have the plugin restrict synchronization to only loading new incidents from SpiraTeam > JIRA and updating existing items. This is useful if you want to prevent existing issues in JIRA from being loaded into SpiraTeam. Leave blank if you want the plugin to synchronize normally.
- **Custom 04** – This should be set to the word “True” if you want to have the plugin copy file attachments from SpiraTeam > JIRA. This can use additional system resources and may fail if the files are too large for JIRA’s API to handle. Leave the field blank if you want the default behavior – which is to not synchronize attachments.
- **Custom 05** - When you click “Force Resync” inside SpiraTeam it will attempt to resynchronize all incidents/issues from 1/1/1900. Sometimes that causes the JIRA API to timeout or exceed the maximum allowed number of results if there are a large number of existing issues in JIRA.

You can set this field to a specific year (e.g. 1995) or year and month (e.g. 2010-11) to restrict how far back the system will look for existing issues. If you leave this field blank it will use the default value of “1900-01”.

Note: For most users, we recommend leaving Custom 01 – Custom 05 blank.

3.2. Configuring the Data Mapping

Next, you need to configure the data mapping between SpiraTeam and JIRA. This allows the various projects, users, releases, incident types, statuses, priorities and custom property values used in the two applications to be related to each other. This is important, as without a correct mapping, there is no way for the integration service to know that an “Enhancement” in SpiraTeam is the same as a “New Feature” in JIRA (for example).

The following mapping information needs to be setup in SpiraTeam:

- The mapping of the project identifiers for the projects that need to be synchronized
- The mapping of users in the system
- The mapping of releases (equivalent to JIRA versions) in the system
- The mapping of the various standard fields in the system
- The mapping of the various custom properties in the system

Each of these is explained in turn below:

3.2.1. Configuring the Project Mapping

From the data synchronization administration page, you need to click on the “View Project Mappings” hyperlink next to the JIRA plug-in name. This will take you to the data-mapping home page for the currently selected project:

View/Edit Project Data Mapping | Library Information System [\(Change Project\)](#)
[<< Back to Data Synchronization Home](#)

To configure this project for data mapping with the [JiraDataSync](#) plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name: Library Information System

External Key:*

Active:* ▼

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Incident	Priority Severity Status Type	Notes Operating System
Release		Notes Operating System
Task	Priority Status	

If the project name does not match the name of the project you want to configure the data-mapping for, click on the “(Change Project)” hyperlink to change the current project.

To enable this project for data-synchronization with JIRA, you need to enter:

- **External Key** – This should be set to the name of the project token in JIRA. Typically this is a three-letter acronym for the project.
- **Active Flag** – Set this to ‘Yes’ so that SpiraTeam knows that you want to synchronize data for this project. Once the project has been completed, setting the value to “No” will stop data synchronization, reducing network utilization.

Click [Update] to confirm these settings. Once you have enabled the project for data-synchronization, you can now enter the other data mapping values outlined below.

Note: Once you have successfully configured the project, when creating a new project, you should choose the option to “Create Project from Existing Project” rather than “Use Default Template” so that all the project mappings get copied across to the new project.

3.2.2. Configuring the User Mapping

To configure the mapping of users in the two systems, you need to go to Administration > Users > View Edit Users, which will bring up the list of users in the system. Then click on the “Edit” button for a particular user that will be editing issues in JIRA:

Add/Edit User | Fred Bloggs

Please review the information listed below and make any changes if necessary. Once you have finished, click [Update/Insert] to confirm them.

First Name*:

Middle Initial:

Last Name*:

User Name*:

Email Address*:

System Administrator*:

Active User*:

BugzillaDataSync ID:

FogBugzDataSync ID:

JiraDataSync ID:

MsTfsDataSync ID:

LDAP Managed User

LDAP DN:

Change Password

New Password:

Confirm Password:

You will notice that below the Active flag for the user is a list of all the configured data-synchronization plug-ins. In the text box next to the JIRA Data-Sync plug-in you need to enter the login for this username in JIRA. This will allow the data-synchronization plug-in to know which user in SpiraTeam match which equivalent user in JIRA. Click [Update] once you’ve entered the appropriate login name. You should now repeat for the other users who will be active in both systems.

3.2.3. Configuring the Release Mapping

When the data-synchronization service runs, when it comes across a release/iteration in SpiraTeam that it has not seen before, it will create a corresponding “Version” in JIRA. Similarly if it comes across a new Version in JIRA that it has not seen before, it will create a new Release in SpiraTeam. Therefore when using both systems together, it is recommended that you only enter new Releases/Versions in one system and let the data-synchronization service add them to the other system.

However you may start out with the situation where you already have pre-existing Releases/Version in both systems that you need to associate in the data-mapping. If you don’t do this, you may find that duplicates get created when you first enable the data-synchronization service. Therefore for any Releases/Iterations that already exist in BOTH systems please navigate to Planning > Releases and click on the Release/Iteration in question. Make sure you have the ‘Overview’ tab visible and expand the “Details” section of the release/iteration:

Release: 1.0 - 1.0 [RL:000021]

Name: 1.0

Overview Incidents # Reqs & Tasks # Test Cases Test Runs # Attachments History

Details

Version #: 1.0 Creator*: System Administrator

Start/End Date: 4/10/2013 - 4/15/2013 Creation Date: 4/11/2013 4:58:18 PM

Resources: 1 Active*: Yes

Non-Working: 0 person days Iteration*: No

Plan Effort: 32.00 Available Effort: 32.00

JiraDataSync ID: 10000

In addition to the standard fields custom properties configured for Releases, you will see an additional text property called “**JiraDataSync ID**” that is used to store the mapped external identifier for the equivalent Version in JIRA. You need to locate the ID of the equivalent version in JIRA, enter it into this text-box and click [Save]. You should now repeat for all the other pre-existing releases.

*Note: The JIRA ID can be found by looking at the URL inside JIRA when choosing to View/Edit the version name/description. The URL will include the section: **id=X** where X is the numeric ID of the version inside JIRA.*

3.2.4. Configuring the Standard Field Mapping

Now that the projects, user and releases have been mapped correctly, we need to configure the standard incident fields. To do this, go to Administration > System > Data Synchronization and click on the “View Project Mappings” for the JiraDataSync plug-in entry:

View/Edit Project Data Mapping | Library Information System [\(Change Project\)](#)

[<< Back to Data Synchronization Home](#)

To configure this project for data mapping with the **JiraDataSync** plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name: Library Information System

External Key:*

Active:*

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Incident	Priority Severity Status Type	Notes Operating System
Release		Notes Operating System
Task	Priority Status	

From this screen, you need to click on Priority, Severity, Status and Type in turn to configure their values:

a) Incident Type

Click on the “Type” hyperlink under Incident Standard Fields to bring up the Incident type mapping configuration screen:

Field Value	Active	External Key	Primary
Bug	Yes	<input type="text" value="1"/>	Yes ▼
Change Request	Yes	<input type="text" value="4"/>	Yes ▼
Enhancement	Yes	<input type="text" value="2"/>	Yes ▼
Incident	Yes	<input type="text" value="1"/>	No ▼
Issue	Yes	<input type="text" value="1"/>	No ▼
Limitation	Yes	<input type="text" value="1"/>	No ▼
Risk	Yes	<input type="text" value="1"/>	No ▼
Training	Yes	<input type="text" value="1"/>	No ▼

[► Update](#)

The table lists each of the incident types available in SpiraTeam and provides you with the ability to enter the matching JIRA issue type ID for each one. You can map multiple SpiraTeam fields to the same JIRA fields (e.g. Bug and Incident in SpiraTeam are both equivalent to Bug in JIRA), in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from JIRA > SpiraTeam).

*Note: The JIRA ID can be found by looking at the URL inside JIRA when choosing to View/Edit the Issue Type. The URL will include the section: **id=X** where X is the numeric ID of the Issue Type inside JIRA.*

b) Incident Status

Click on the "Status" hyperlink under Incident Standard Fields to bring up the Incident status mapping configuration screen:

Field Value	Active	External Key	Primary
Assigned	Yes	<input type="text" value="3"/>	Yes ▼
Closed	Yes	<input type="text" value="6"/>	Yes ▼
Duplicate	Yes	<input type="text" value="1"/>	No ▼
New	Yes	<input type="text" value="1"/>	No ▼
Not Reproducible	Yes	<input type="text" value="1"/>	No ▼
Open	Yes	<input type="text" value="1"/>	Yes ▼
Reopen	Yes	<input type="text" value="4"/>	Yes ▼
Resolved	Yes	<input type="text" value="5"/>	Yes ▼

[► Update](#)

The table lists each of the incident statuses available in SpiraTeam and provides you with the ability to enter the matching JIRA issue status ID for each one. You can map multiple SpiraTeam fields to the same JIRA fields (e.g. New and Open in SpiraTeam are both equivalent to Open in JIRA), in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from JIRA > SpiraTeam).

We recommend that you always point the New and Open statuses inside SpiraTeam to point to the ID for “Open” inside JIRA and make Open in SpiraTeam the Primary status of the two. This is recommended so that as new incidents in SpiraTeam get synched over to JIRA, they will get switched to the Open status in JIRA which will then be synched back to “Open” in SpiraTeam. That way you’ll be able to see at a glance which incidents have been synched with JIRA and those that haven’t.

*Note: The JIRA ID can be found by looking at the URL inside JIRA when choosing to View/Edit the Issue Status. The URL will include the section: **id=X** where X is the numeric ID of the Issue Status inside JIRA.*

c) Incident Priority

Click on the “Priority” hyperlink under Incident Standard Fields to bring up the Incident Priority mapping configuration screen:

Field Value	Active	External Key	Primary
1 - Critical	Yes	<input type="text" value="2"/>	Yes ▼
2 - High	Yes	<input type="text" value="3"/>	Yes ▼
3 - Medium	Yes	<input type="text" value="4"/>	Yes ▼
4 - Low	Yes	<input type="text" value="5"/>	Yes ▼

The table lists each of the incident priorities available in SpiraTeam and provides you with the ability to enter the matching JIRA priority ID for each one. You can map multiple SpiraTeam fields to the same JIRA fields, in which case only one of the two values can be listed as Primary = Yes as that’s the value that’s used on the reverse synchronization (from JIRA > SpiraTeam).

*Note: The JIRA ID can be found by looking at the URL inside JIRA when choosing to View/Edit the Priority. The URL will include the section: **id=X** where X is the numeric ID of the Priority inside JIRA.*

d) Incident Severity (Optional)

Click on the “Severity” hyperlink under Incident Standard Fields to bring up the Incident severity mapping configuration screen:

Field Value	Active	External Key	Primary
1 - Critical	Yes	<input type="text" value="Critical"/>	Yes ▼
2 - High	Yes	<input type="text" value="High"/>	Yes ▼
3 - Medium	Yes	<input type="text" value="Medium"/>	Yes ▼
4 - Low	Yes	<input type="text" value="Low"/>	Yes ▼

Unlike the other incident standard fields, JIRA doesn’t actually have a built-in field for storing the severity of an issue, so if you want to be able to see the SpiraTeam incident severity in JIRA, you’ll need to create a JIRA custom list field to store the different severity values. If you don’t want to synchronize severity values with JIRA, you can skip the rest of this section.

Once you have created a custom field in JIRA to contain the list of severity values, you need to now populate the above table with the name (Not the ID) of the severity custom property values inside JIRA and click Update. Secondly you need to go to the Plug-in configuration screen:

Edit Plug-In | JiraDataSync
[<< Back to Data Synchronization Home](#)

Please enter/edit the following information for the plug-in. The exact information that needs to be entered is specific to each plug-in, and you should refer to the documentation for the plug-in when entering/changing it:

Name*:

Description:

Connection Info*:

Login*:

Password:

Time Offset*:
 Auto-Map Users

Custom 01:

Custom 02:

Custom 03:

Custom 04:

Custom 05:

On this screen you need to enter the ID of the custom field that you're using to store severities in JIRA and populate the Custom 01 property with this value (see above). *Note: The ID can be found by looking at the URL inside JIRA when choosing to View/Edit the Custom Field. The URL will include the section: **id=X** where X is the numeric ID of the Custom Field inside JIRA.*

3.2.5. Configuring the Custom Property Mapping

Now that the various SpiraTeam standard fields have been mapped correctly, we need to configure the custom property mappings. This is used for both custom properties in SpiraTeam that map to custom fields in JIRA and also for custom properties in SpiraTeam that are used to map to standard fields in JIRA (Component, Environment, Resolution and SecurityLevel) that don't exist in SpiraTeam.

From the View/Edit Project Data Mapping screen, you need to click on the name of the Incident Custom Property that you want to add data-mapping information for. We will consider the four different types of mapping that you might want to enter:

View/Edit Project Data Mapping | Library Information System [\(Change Project\)](#)

[<< Back to Data Synchronization Home](#)

To configure this project for data mapping with the [JiraDataSync](#) plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name: Library Information System

External Key:*

Active:*

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Incident	Priority Severity Status Type	Environment Notes Component Operating System
Release		Notes Operating System
Task	Priority Status	

a) Text Custom Properties

Click on the hyperlink of the text custom property under Incident Custom Properties to bring up the custom property mapping configuration screen. For text custom properties there will be no values listed in the lower half of the screen.

Edit Custom Property Data Mapping | Library Information System [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the JiraDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Name: Notes

External Key:

Custom Value Mapping

Property Value	Active	External Key
<input type="button" value="Update"/>		

You need to lookup the ID of the custom field in JIRA that matches this custom property in SpiraTeam. Once you have entered the id of the custom field, click [Update].

*Note: The ID can be found by looking at the URL inside JIRA when choosing to View/Edit the Custom Field. The URL will include the section: **id=X** where X is the numeric ID of the Custom Field inside JIRA.*

b) List Custom Properties

Click on the hyperlink of the list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen. For list custom properties there will be a textbox for both the custom field itself and a mapping table for each of the custom property values that need to be mapped:

Edit Custom Property Data Mapping | Library Information System [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the JiraDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Name: Operating System

External Key:

Custom Value Mapping

Property Value	Active	External Key
Windows 2000	Yes	<input type="text" value="Windows 2000"/>
Windows 2003	Yes	<input type="text" value="Windows 2003"/>
Windows NT 4.0	No	<input type="text"/>
Windows Vista	Yes	<input type="text" value="Windows Vista"/>
Windows XP	Yes	<input type="text" value="Windows XP"/>

First you need to lookup the ID of the custom field in JIRA that matches this custom property in SpiraTeam. This should be entered in the 'External Key' field below the name of the custom property. *Note: The ID can be found by looking at the URL inside JIRA when choosing to View/Edit the Custom Field. The URL will include the section: **id=X** where X is the numeric ID of the Custom Field inside JIRA.*

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the full name (not the id this time) of the custom field value as specified in JIRA.

c) JIRA's Component Field

If your instance of JIRA requires that all new issues are submitted with a 'Component' then you will need to fill out this section. You first need to create an incident custom property in SpiraTeam of type 'LIST' that contains the various component names that exist inside JIRA.

Then click on the hyperlink of this new list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen:

Edit Custom Property Data Mapping | Library Information System [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the JiraDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Name: Component

External Key:

Custom Value Mapping

Property Value	Active	External Key
Component 1	Yes	<input type="text" value="10001"/>
Component 2	Yes	<input type="text" value="10010"/>
Component 3	Yes	<input type="text" value="10020"/>

First you need to enter the word “**Component**” as the External Key of the custom property. This tells the data-sync plug-in that the custom property in SpiraTeam should be mapped to built-in Component field in JIRA.

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the JIRA ID of the various Components that are configured in JIRA. The external ID can be found by looking at the URL inside JIRA which choosing to View/Edit the component name/description.

d) JIRA's Resolution Field

If you would like the values of the JIRA ‘**Resolution**’ field to be synchronized back to SpiraTeam, then you will need to fill out this section. You first need to create an incident custom property in SpiraTeam of type ‘LIST’ that contains the various resolution names that exist inside JIRA.

Then click on the hyperlink of this new list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen:

Edit Custom Property Data Mapping | Sample JIRA Project [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the JiraDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Name: JIRA-Resolution

External Key:

Custom Value Mapping

Property Value	Active	External Key
Cannot Reproduce	Yes	<input type="text" value="5"/>
Fixed	Yes	<input type="text" value="1"/>
Won't Fix	Yes	<input type="text" value="2"/>

First you need to enter the word “Resolution” as the External Key of the custom property. This tells the data-sync plug-in that the custom property in SpiraTeam should be mapped to built-in Resolution field in JIRA.

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the JIRA ID of the various Resolutions that are configured in JIRA. The external ID can be found by looking at the URL inside JIRA which choosing to View/Edit the resolution name/description.

e) JIRA's Environment Field

If your instance of JIRA requires that all new issues are submitted with an 'Environment' description specified, then you will need to fill out this section. You first need to create an incident custom property in SpiraTeam of type 'TEXT' that will be used to store the environment description within SpiraTeam.

Then click on the hyperlink of this new list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen:

Edit Custom Property Data Mapping | Library Information System ([Change Project](#))

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the JiraDataSync external key values for the custom property below. If this is a custom list property, you also need to fill in the external key for each of the custom property values as well:

Name: Environment

External Key:

[Update](#)

Custom Value Mapping

Property Value	Active	External Key
----------------	--------	--------------

[Update](#)

All you need to do on this screen is enter the word "**Environment**" in the External Key textbox and the data-sync plug-in will know that this custom property is mapped to the built-in Environment field in JIRA.

f) JIRA's Security Level Field (JIRA 4.x Plug-In Only)

If your instance of JIRA requires that all new issues are submitted with a 'Security Level' then you will need to fill out this section. You first need to create an incident custom property in SpiraTeam of type 'LIST' that contains the various security levels that exist inside JIRA.

Then click on the hyperlink of this new list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen.

First you need to enter the word "**SecurityLevel**" as the External Key of the custom property. This tells the data-sync plug-in that the custom property in SpiraTeam should be mapped to built-in Security Level field in JIRA.

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the JIRA ID of the various Security Levels that are configured in JIRA. The external ID can be found by looking at the URL inside JIRA which choosing to View/Edit the security level name/description.

g) JIRA's Issue Key Field (JIRA 4.x Plug-In Only)

It can be convenient to create a SpiraTeam custom property to store the JIRA Issue Key (the ID used to identify an issue in JIRA). This allows you to display a list of incidents in SpiraTest and see the corresponding JIRA ID in the same list. You first need to create an incident custom property in SpiraTeam of type 'TEXT' that will be used to store the JIRA issue key within SpiraTeam.

Then click on the hyperlink of this new list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen:

Edit Custom Property Data Mapping | Sample Application One [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the JiraDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Name: JIRA Key

External Key:

[> Update](#)

Custom Value Mapping

Property Value	Active	External Key

[> Update](#)

All you need to do on this screen is enter the word “**JiralssueKey**” in the External Key textbox and the data-sync plug-in will know that this custom property is mapped to the built-in Issue Key field in JIRA.

Once you have updated the various mapping sections, you are now ready to start using the system.

3.3. Using SpiraTeam with JIRA

Now that the integration service has been configured and the service started, initially any incidents created in SpiraTeam for the specified projects will be imported into JIRA and if using JIRA 4.x, any existing issues in JIRA will get loaded into SpiraTeam

At this point we recommend opening the Windows Event Viewer and choosing the Application Log. In this log any error messages raised by the SpiraTeam Data Sync Service will be displayed. If you see any error messages at this point, we recommend immediately stopping the SpiraTeam service and checking the various mapping entries. If you cannot see any issues with the mapping information, we recommend sending a copy of the event log message(s) to Inflectra customer services (support@inflectra.com) who will help you troubleshoot the problem.

To use SpiraTeam with JIRA on an ongoing basis, we recommend the following general processes be followed:

- When running tests in SpiraTest or SpiraTeam, defects found should be logged through the Test Execution Wizard as normal.
- Developers using JIRA 4.x can log new defects into either SpiraTeam or JIRA. In either case they will get loaded into the other system.
- Once created in one of the systems and successfully replicated to the other system, the incident should not be modified again inside SpiraTeam
- At this point, the incident should not be acted upon inside SpiraTeam and all data changes to the issue should be made inside JIRA. To enforce this, you should modify the workflows set up in SpiraTeam so that the various fields are marked as inactive for all the incident statuses other than the “New” status. This will allow someone to submit an incident in SpiraTeam, but will prevent them making changes in conflict with JIRA after that point.
- As the issue progresses through the customized JIRA workflow, changes to the type of issue, changes to its status, priority, description and resolution will be updated automatically in SpiraTeam. In essence, SpiraTeam acts as a read-only viewer of these incidents.
- You are now able to perform test coverage and incident reporting inside SpiraTest/SpiraTeam using the test cases managed by SpiraTest/SpiraTeam and the incidents managed on behalf of SpiraTest/SpiraTeam inside JIRA.

4. Using SpiraTest with Bugzilla

This section outlines how to use SpiraTest in conjunction with the open-source Bugzilla bug tracking system. The built-in integration service allows the quality assurance team to manage their requirements and test cases in SpiraTest, execute test runs in SpiraTest, and then have the new incidents generated during the run be automatically loaded into Bugzilla. Once the incidents are loaded into Bugzilla as bugs, the development team can then manage the lifecycle of these bugs in Bugzilla, and have the status changes in Bugzilla be reflected back in SpiraTest.

In addition, if you are using Bugzilla 4.x or higher, any issues logged directly into Bugzilla will get imported into SpiraTeam so that they can be linked to test cases and requirements.

► **STOP! Please make sure you have first read the Instructions in Section 1 before proceeding!**

4.1. Configuring the Plug-In

The next step is to configure the plug-in within SpiraTeam so that the system knows how to access the Bugzilla server. To start the configuration, please open up SpiraTeam in a web browser, log in using a valid account that has System-Administration level privileges and click on the System > Data Synchronization administration option from the left-hand navigation:

Data Synchronization

SpiraTeam is able to communicate with a variety of external systems and applications using its background data synchronization service and library of available plug-ins.

This page allows you to view, add and modify the list of plug-ins, make changes to their configuration, and verify the status of the most recent synchronization. In addition, you can force a particular plug-in to perform a complete data-synchronization by resetting its date of last sync:

[Refresh](#) [+ Add](#)

Plug-In	Data Mapping	Last Sync Date	Status	Operations
BugzillaDataSync	View Project Mappings	6/21/2016 1:09:56 PM	Failure	Force Re-Sync Edit Delete View Errors
FogBugzDataSync	View Project Mappings	6/21/2016 1:09:56 PM	Failure	Force Re-Sync Edit Delete View Errors
JiraDataSync	View Project Mappings	6/21/2016 1:09:56 PM	Warning	Force Re-Sync Edit Delete View Errors
MsTfsDataSync	View Project Mappings	6/21/2016 1:09:56 PM	Warning	Force Re-Sync Edit Delete View Errors

This screen lists all the plug-ins already configured in the system. Depending on whether you chose the option to include sample data in your installation or not, you will see either an empty screen or a list of sample data-synchronization plug-ins.

If you already see an entry for **BugzillaDataSync** you should click on its “Edit” link. If you don’t see such an entry in the list, please click on the [Add] button instead. In either case you will be taken to the following screen where you can enter or modify the Bugzilla Data-Synchronization plug-in:

Edit Plug-In | BugzillaDataSync

[<< Back to Data Synchronization Home](#)

Please enter/edit the following information for the plug-in. The exact information that needs to be entered is specific to each plug-in, and you should refer to the documentation for the plug-in when entering/changing it:

Name*:	<input type="text" value="BugzillaDataSync"/>
Description:	<input type="text" value="This plug-in allows incidents in the system to be synchronized with the Bugzilla bug-tracking system"/>
Connection Info*:	<input type="text" value="http://landfill.bugzilla.org/bugzilla-3.0-branch/xmlrpc.cgi"/>
Login*:	<input type="text" value="bugzillauser@mycompany.com"/>
Password:	<input type="password" value="*****"/>
Time Offset*:	<input type="text" value="0"/>
	<input type="checkbox"/> Auto-Map Users
Custom 01:	<input type="text" value="False"/>
Custom 02:	<input type="text" value="True"/>
Custom 03:	<input type="text"/>
Custom 04:	<input type="text"/>
Custom 05:	<input type="text"/>
	<input type="button" value="► Update"/>

You need to fill out the following fields for the Bugzilla Plug-in to operate correctly:

- **Name** – this needs to be set to **BugzillaDataSync**. This needs to match the name of the plug-in DLL assembly that was copied into the C:\Program Files\SpiraTeam\Bin folder (minus the .dll file extension). If you renamed the BugzillaDataSync.dll file for any reason, then you need to change the name here to match.
- **Description** – this should be set to a description of the plug-in. This is an optional field that is used for documentation purposes and is not actually used by the system.
- **Connection Info** – this should be the full URL to the Bugzilla installation's web-service API. This is typically <http://<Bugzilla server name>/xmlrpc.cgi>
- **Login** – this should be set to a valid login to the Bugzilla installation – typically an email address. The login needs to have permissions to create and view bugs within Bugzilla.
- **Password** – this should be set to the password of the login specified above.
- **Time Offset** – normally this should be set to zero, but if you find that issues being changed in Bugzilla are not being updated in SpiraTeam, try increasing the value as this will tell the data-synchronization plug-in to add on the time offset (in hours) when comparing date-time stamps. Also if your Bugzilla installation is running on a server set to a different time-zone, then you should add in the number of hours difference between the servers' time-zones here.
- **Auto-Map Users** – this is not currently used by the Bugzilla data-sync plug-in and can be ignored.

- **Custom 01** – When connecting to Bugzilla, sometimes the connection gets dropped by the server without notifying the plug-in. This happens when using HTTP 1.1 Keep-Alive connections. If you set this property to “**False**”, it will tell the plug-in to **not-use** HTTP keep-alives when connecting to Bugzilla, otherwise set it to “**True**”.
- **Custom 02** – When connecting to a Bugzilla instance that is running under HTTPS (SSL) this custom property can be set to determine if the plug-in should verify that the SSL certificate is a trusted root certificate. Set to “**True**” if you are using an SSL certificate that was issued by a trusted Certification Authority, and set to “**False**” if you are using a self-signed certificate.
- **Custom 03 – 05** – these are not currently used by the Bugzilla data-sync plug-in and can be left blank.

4.2. Configuring the Data Mapping

Next, you need to configure the data mapping between SpiraTeam and Bugzilla. This allows the various projects, users, releases, incident types, statuses, priorities and custom property values used in the two applications to be related to each other. This is important, as without a correct mapping, there is no way for the integration service to know that an “Duplicate” incident in SpiraTeam is the same as an “UNCONFIRMED” bug in Bugzilla (for example).

The following mapping information needs to be setup in SpiraTeam:

- The mapping of the project identifiers for the projects that need to be synchronized
- The mapping of users in the system
- The mapping of releases (equivalent to Bugzilla versions) in the system
- The mapping of the various standard fields in the system
- The mapping of the various custom properties in the system

Each of these is explained in turn below:

4.2.1. Configuring the Project Mapping

From the data synchronization administration page, you need to click on the “View Project Mappings” hyperlink next to the Bugzilla plug-in name. This will take you to the data-mapping home page for the currently selected project:

View/Edit Project Data Mapping | Library Information System [\(Change Project\)](#)

[<< Back to Data Synchronization Home](#)

To configure this project for data mapping with the [BugzillaDataSync](#) plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name: Library Information System

External Key:*

Active:*

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Incident	Priority Severity Status Type	Resolution Notes Component Operating System Hardware
Release		Notes Operating System
Task	Priority Status	

If the project name does not match the name of the project you want to configure the data-mapping for, click on the “(Change Project)” hyperlink to change the current project.

To enable this project for data-synchronization with Bugzilla, you need to enter:

- **External Key** – This should be set to the name of the equivalent Product in Bugzilla.
- **Active Flag** – Set this to ‘Yes’ so that SpiraTeam knows that you want to synchronize data for this project. Once the project has been completed, setting the value to “No” will stop data synchronization, reducing network utilization.

Click [Update] to confirm these settings. Once you have enabled the project for data-synchronization, you can now enter the other data mapping values outlined below.

Note: Once you have successfully configured the project, when creating a new project, you should choose the option to “Create Project from Existing Project” rather than “Use Default Template” so that all the project mappings get copied across to the new project.

4.2.2. Configuring the User Mapping

To configure the mapping of users in the two systems, you need to go to Administration > Users > View Edit Users, which will bring up the list of users in the system. Then click on the “Edit” button for a particular user that will be editing issues in Bugzilla:

Add/Edit User | Fred Bloggs

Please review the information listed below and make any changes if necessary. Once you have finished, click [Update/Insert] to confirm them.

First Name*:

Middle Initial:

Last Name*:

User Name*:

Email Address*:

System Administrator*:

Active User*:

BugzillaData Sync ID:

LDAP Managed User

LDAP DN:

Change Password

New Password:

Confirm Password:

You will notice that below the Active flag for the user is a list of all the configured data-synchronization plug-ins. In the text box next to the Bugzilla Data-Sync plug-in you need to enter the login for this username in Bugzilla. This will allow the data-synchronization plug-in to know which user in SpiraTeam match which equivalent user in Bugzilla. Click [Update] once you've entered the appropriate login name. You should now repeat for the other users who will be active in both systems.

4.2.3. Configuring the Release Mapping

Now that the projects and users have been mapped correctly, we need to configure the mapping between Releases/Iterations in SpiraTeam and Versions in Bugzilla. To do this, please navigate to Planning > Releases and click on the Release/Iteration in question. Make sure you have the 'Overview' tab visible and expand the "Details" section of the release/iteration:

Release: 3.0.0.0 - Release 3.0 [RL:000020]

Name:

Overview Incidents Reqs & Tasks Test Cases Test Runs Attachments History

Details

Version #: Creator*:

Start/End Date: - Creation Date: 6/5/2013 5:49:11 PM

Resources: Active*:

Non-Working: person days Iteration*:

Plan Effort: 184.00 Available Effort: 184.00

BugzillaDataSync ID:

In addition to the standard fields and custom properties configured for Releases, you will see an additional text property called "**BugzillaDataSync ID**" that is used to store the mapped external identifier for the equivalent Version in Bugzilla. You need to enter the name of the equivalent version in Bugzilla, enter it into this text-box and click [Save]. You should now repeat for all the other releases and iterations in the project.

If you are using the plugin for Bugzilla 4.x then any Versions that have already been created in Bugzilla will be automatically imported into SpiraTeam if they do not already exist in SpiraTeam and they have not already been mapped.

4.2.4. Configuring the Standard Field Mapping

Now that the projects, user and releases have been mapped correctly, we need to configure the standard incident fields. To do this, go to Administration > System > Data Synchronization and click on the “View Project Mappings” for the BugzillaDataSync plug-in entry:

View/Edit Project Data Mapping | Library Information System [\(Change Project\)](#)

[<< Back to Data Synchronization Home](#)

To configure this project for data mapping with the **BugzillaDataSync** plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name: Library Information System

External Key:*

Active:* ▼

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Incident	Priority Severity Status Type	Resolution Notes Component Operating System Hardware
Release		Notes Operating System
Task	Priority Status	

From this screen, you need to click on Priority, Severity and Status in turn to configure their values:

a) Incident Status

Click on the “Status” hyperlink under Incident Standard Fields to bring up the Incident status mapping configuration screen:

Edit Standard Field Data Mapping | Library Information System [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this field, please fill in the external key values for the **BugzillaDataSync** plug-in next to each SpiraTeam value:

Field Value	Active	External Key	Primary
Assigned	Yes	<input type="text" value="ASSIGNED"/>	<input type="button" value="Yes"/> ▼
Closed	Yes	<input type="text" value="CLOSED"/>	<input type="button" value="Yes"/> ▼
Duplicate	Yes	<input type="text" value="UNCONFIRMED"/>	<input type="button" value="No"/> ▼
New	Yes	<input type="text" value="NEW"/>	<input type="button" value="No"/> ▼
Not Reproducible	Yes	<input type="text" value="UNCONFIRMED"/>	<input type="button" value="Yes"/> ▼
Open	Yes	<input type="text" value="NEW"/>	<input type="button" value="Yes"/> ▼
Reopen	Yes	<input type="text" value="REOPENED"/>	<input type="button" value="Yes"/> ▼
Resolved	Yes	<input type="text" value="RESOLVED"/>	<input type="button" value="Yes"/> ▼

The table lists each of the incident statuses available in SpiraTeam and provides you with the ability to enter the matching Bugzilla bug status for each one. You can map multiple SpiraTeam fields to the same Bugzilla fields (e.g. New and Open in SpiraTeam are both equivalent to NEW in Bugzilla), in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from Bugzilla > SpiraTeam).

We recommend that you always point the New and Open statuses inside SpiraTeam to point to the NEW status inside Bugzilla and make Open in SpiraTeam the Primary status of the two. This is recommended so that as new incidents in SpiraTeam get synched over to Bugzilla, they will get switched to the NEW status in Bugzilla which will then be synched back to "Open" in SpiraTeam. That way you'll be able to see at a glance which incidents have been synched with Bugzilla and those that haven't.

b) Incident Priority

Click on the "Priority" hyperlink under Incident Standard Fields to bring up the Incident Priority mapping configuration screen:

Edit Standard Field Data Mapping | Library Information System [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this field, please fill in the external key values for the **BugzillaDataSync** plug-in next to each SpiraTeam value:

Field Value	Active	External Key	Primary
1 - Critical	Yes	<input type="text" value="P1"/>	Yes ▼
2 - High	Yes	<input type="text" value="P2"/>	Yes ▼
3 - Medium	Yes	<input type="text" value="P3"/>	Yes ▼
4 - Low	Yes	<input type="text" value="P4"/>	Yes ▼

The table lists each of the incident priorities available in SpiraTeam and provides you with the ability to enter the matching Bugzilla priority for each one. You can map multiple SpiraTeam fields to the same Bugzilla fields, in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from Bugzilla > SpiraTeam).

c) Incident Severity

Click on the "Severity" hyperlink under Incident Standard Fields to bring up the Incident severity mapping configuration screen:

Edit Standard Field Data Mapping | Library Information System [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this field, please fill in the external key values for the **BugzillaDataSync** plug-in next to each SpiraTeam value:

Field Value	Active	External Key	Primary
1 - Critical	Yes	<input type="text" value="critical"/>	Yes ▼
2 - High	Yes	<input type="text" value="major"/>	Yes ▼
3 - Medium	Yes	<input type="text" value="normal"/>	Yes ▼
4 - Low	Yes	<input type="text" value="minor"/>	Yes ▼

[▶ Update](#)

The table lists each of the incident severities available in SpiraTeam and provides you with the ability to enter the matching Bugzilla severity for each one. You can map multiple SpiraTeam fields to the same Bugzilla fields, in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from Bugzilla > SpiraTeam).

4.2.5. Configuring the Custom Property Mapping

Now that the various SpiraTeam standard fields have been mapped correctly, we need to configure the custom property mappings. This is used for custom properties in SpiraTeam that are used to map to standard fields in Bugzilla (Component, Hardware, Operating System and Resolution) that don't exist in SpiraTeam. You need to make sure that you have first added custom lists in SpiraTeam that contain the list of Components, Hardware platforms and Operating Systems used in Bugzilla and that you have setup those lists as Custom Properties on the Incident artifact type.

Once that's done, from the View/Edit Project Data Mapping screen, you need to click on the name of the Incident Custom Property that you want to add data-mapping information for. We will consider the four different types of mapping that you might want to enter in turn:

View/Edit Project Data Mapping | Library Information System [\(Change Project\)](#)

[<< Back to Data Synchronization Home](#)

To configure this project for data mapping with the [BugzillaDataSync](#) plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name: Library Information System

External Key:*

Active:* ▼

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Incident	Priority Severity Status Type	Resolution Notes Component Operating System Hardware
Release		Notes Operating System
Task	Priority Status	

a) Bugzilla's Component Field

If your instance of Bugzilla requires that all new bugs are submitted with a 'Component' then you will need to fill out this section. You first need to create an incident custom property in SpiraTeam of type 'LIST' that contains the various component names that exist inside Bugzilla.

Then click on the hyperlink of this new list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen:

Edit Custom Property Data Mapping | Library Information System [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the JiraDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Name: Component

External Key:

Custom Value Mapping

Property Value	Active	External Key
Component 1	Yes	<input type="text" value="10001"/>
Component 2	Yes	<input type="text" value="10010"/>
Component 3	Yes	<input type="text" value="10020"/>

First you need to enter the word "Component" as the External Key of the custom property. This tells the data-sync plug-in that the custom property in SpiraTeam should be mapped to built-in Component field in Bugzilla.

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the Bugzilla name of the various Components that are configured in Bugzilla.

b) Bugzilla's Operating System Field

If your instance of Bugzilla requires that all new issues are submitted with an 'Operating System' then you will need to fill out this section. You first need to create an incident custom property in SpiraTeam of type 'LIST' that contains the various operating system names that exist inside Bugzilla.

Then click on the hyperlink of this new list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen:

Edit Custom Property Data Mapping | Library Information System [\(Change Project\)](#)
[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the BugzillaDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Name: Operating System
External Key:

Custom Value Mapping

Property Value	Active	External Key
Windows 2000	Yes	<input type="text" value="Windows 2000"/>
Windows 2003	Yes	<input type="text" value="Windows 2003"/>
Windows NT 4.0	No	<input type="text"/>
Windows Vista	Yes	<input type="text" value="Windows Vista"/>
Windows XP	Yes	<input type="text" value="Windows XP"/>

First you need to enter the word "OperatingSystem" as the External Key of the custom property. This tells the data-sync plug-in that the custom property in SpiraTeam should be mapped to built-in Operating System field in Bugzilla.

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the Bugzilla name of the various Operating System values that are configured in Bugzilla.

c) Bugzilla's Hardware Field

If your instance of Bugzilla requires that all new issues are submitted with a 'Hardware' value then you will need to fill out this section. You first need to create an incident custom property in SpiraTeam of type 'LIST' that contains the various hardware platform names that exist inside Bugzilla.

Then click on the hyperlink of this new list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen:

Edit Custom Property Data Mapping | Library Information System [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the BugzillaDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Name: Hardware

External Key:

Custom Value Mapping

Property Value	Active	External Key
HP	Yes	<input type="text" value="HP"/>
Sun	Yes	<input type="text" value="Sun"/>

First you need to enter the word “Hardware” as the External Key of the custom property. This tells the data-sync plug-in that the custom property in SpiraTeam should be mapped to built-in Hardware field in Bugzilla.

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the Bugzilla name of the various Hardware platforms that are configured in Bugzilla.

d) Bugzilla’s Resolution Field (Optional)

When incidents in SpiraTeam are updated with changes made in Bugzilla, the value of the Bugzilla resolution field (FIXED, INVALID, WONTFIX, LATER, REMIND, DUPLICATE, WORKSFORME, MOVED, DEPLOY) is used to populate the Resolution/Comments text box within SpiraTeam.

However the Resolution/Comments field in SpiraTeam cannot be displayed in the incident list page as it’s a long text-field, so if you would like to be able to see the list of Bugzilla Resolution codes displayed in a list, it is necessary to add a TEXT custom property to Incidents that can be used to store this returned value and then be filtered in the list. The rest of this section describes how to map this text custom property so that it picks up the Resolution field values from Bugzilla.

To configure the mapping, click on the hyperlink of this new text custom property under Incident Custom Properties to bring up the custom property mapping configuration screen:

Edit Custom Property Data Mapping | Library Information System [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the BugzillaDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Name: Resolution

External Key:

Custom Value Mapping

Property Value	Active	External Key
----------------	--------	--------------

All you need to do on this screen is enter the word “Resolution” in the External Key textbox and the data-sync plug-in will know that this custom property is mapped to the built-in Resolution field in Bugzilla.

Once you have updated the various mapping sections, you are now ready to use the synchronization.

4.3. Using SpiraTest with Bugzilla

Now that the integration service has been configured and the service started, initially any incidents created in SpiraTeam for the specified projects will be imported into Bugzilla. At this point we recommend opening the Windows Event Viewer and choosing the Application Log. In this log any error messages raised by the Data Synchronization service will be displayed. If you see any error messages at this point, we recommend immediately stopping the service and checking the various mapping entries. If you cannot see any issues with the mapping information, we recommend sending a copy of the event log message(s) to Inflectra customer services (support@inflectra.com) who will help you troubleshoot the problem.

To use SpiraTeam with Bugzilla on an ongoing basis, we recommend the following general processes be followed:

- When running tests in SpiraTeam, defects found should be logged through the ‘Add Incident’ option as normal.
- Once an incident has been created during the running of the test, it will now be populated across into Bugzilla as a bug. It will be populated with the information captured in SpiraTeam.
- At this point, the incident should not be acted upon inside SpiraTeam, and all data changes to the issue should be made inside Bugzilla. To enforce this, you can modify the workflows set up in SpiraTeam so that the various fields are marked as inactive for all the incident statuses other than the “New” status. This will allow someone to submit an incident in SpiraTeam, but will prevent them making changes in conflict with Bugzilla after that point.
- As the issue progresses through the Bugzilla workflow, changes to the status, priority, severity, and resolution will be updated automatically in SpiraTeam. In essence, SpiraTeam acts as a read-only viewer of these incidents.
 - ▷ If you are using the plugin for Bugzilla 4.x, changes to the hardware, operating system and component will also be synchronized back into SpiraTeam. In addition, any comments added to the bug in Bugzilla 4.x will get added to the corresponding incident in SpiraTeam
- You are now able to perform test coverage and incident reporting inside SpiraTeam using the test cases managed by SpiraTeam and the incidents managed on behalf of SpiraTeam inside Bugzilla.

5. Using SpiraTest with MS-TFS

This section outlines how to use SpiraTest, SpiraPlan or SpiraTeam (hereafter referred to as SpiraTeam) in conjunction with the work item tracking functionality of Microsoft Visual Studio Team Services (MS-VSTS), Visual Studio Online (VSO), or Microsoft Team Foundation Server (TFS) hereafter referred to as TFS.

The built-in integration service allows the quality assurance team to manage their requirements and test cases in SpiraTeam, execute test runs in SpiraTest, and then have the new incidents generated during the run be automatically loaded into TFS. Once the incidents are loaded into TFS as work items, the development team can then manage the lifecycle of these work items in TFS, and have the status changes in TFS be reflected back in SpiraTeam.

Similarly, as the requirements are decomposed into discrete project tasks in SpiraTeam, the integration service will automatically load these new tasks into TFS as task work items where the development team can manage their lifecycle, with schedule and progress changes in TFS being reflected back in SpiraTeam.

► **STOP! Please make sure you have first read the Instructions in Section 1 before proceeding!**

► **Note: Only the MS-TFS 2012+ Plug-In is Available on the Inflectra Cloud-Based DataSync Service.**

5.1. Configuring the Plug-In

The next step is to configure the plug-in within SpiraTeam so that the system knows how to access the TFS server. To start the configuration, please open up SpiraTeam in a web browser, log in using a valid account that has System-Administration level privileges and click on the System > Data Synchronization administration option from the left-hand navigation:

Plug-In	Data Mapping	Last Sync Date	Status	Operations
Microsoft TFS MsTfsDataSync	View Project Mappings	11/25/2014 10:40:15 AM	Success	Force Re-Sync Edit Delete View Errors

This screen lists all the plug-ins already configured in the system. Depending on whether you chose the option to include sample data in your installation or not, you will see either an empty screen or a list of sample data-synchronization plug-ins.

If you already see an entry for **MsTfsDataSync** you should click on its “Edit” link. If you don’t see such an entry in the list, please click on the [Add] button instead. In either case you will be taken to the following screen where you can enter or modify the TFS Data-Synchronization plug-in:

Microsoft TFS Edit Plug-In

[← Back to Data Synchronization Home](#)

Please enter/edit the following information for the plug-in. The exact information that needs to be entered is specific to each plug-in, and you should refer to the documentation for the plug-in when entering/changing it.

Name:
The physical name of the plugin (e.g. JiraDataSync) that matches the DLL being used

Caption:
The display name of the plugin (e.g. Internal Jira) that can be used to distinguish between multiple instances of the same plugin type

Description:
The long description of the plugin (e.g. 'Connects to the in-house bug-tracker')

Connection Info:
The URL or other connection string used to connect to the external bug-tracker

Login:
The login and password used to connect to Microsoft TFS

Password:
The login and password used to connect to Microsoft TFS

Time Offset: hours
The number of hours to look back for changes to synchronization. Normally should be set to zero (0) unless you find changes are not being synchronized

Auto-Map Users Yes

Custom 01:

Custom 02:

Custom 03:

Custom 04:

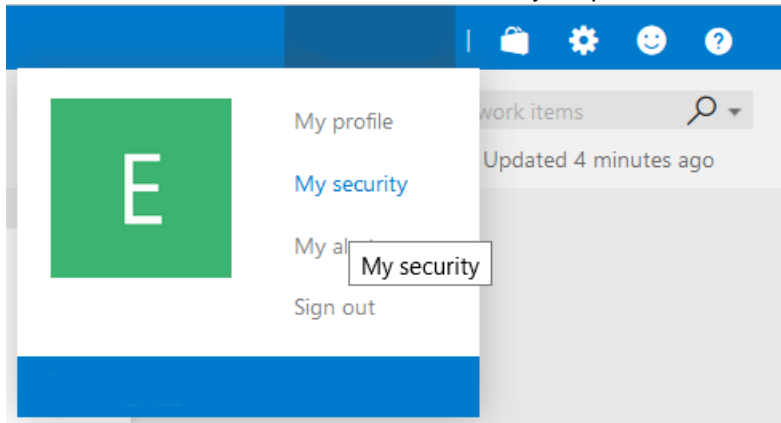
Custom 05:

You need to fill out the following fields for the TFS Plug-in to operate correctly:

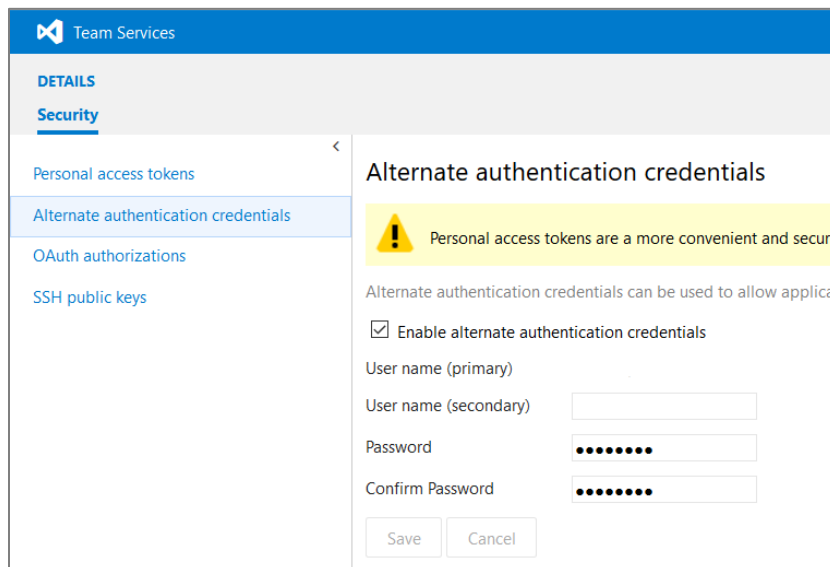
- **Name** – this needs to be set to **MsTfsDataSync**.
- **Caption** – this is the display name of the plugin. Normally you can use something generic such as “Microsoft TFS”, however if you have multiple TFS instances you might want to name it something specific such as “TFS External”. If you don’t enter a value, the display name will be “MsTfsDataSync”
- **Description** – this should be set to a description of the plug-in. This is an optional field that is used for documentation purposes and is not actually used by the system.
- **Connection Info** – this should be the URL that you use for connecting Visual Studio to the Team Foundation Server.
 - ▷ **For Visual Studio Online**, it is of the format <https://mycompany.visualstudio.com>. See our special [KB213 regarding Visual Studio Online](#).
 - ▷ **For TFS 2010+** it is of the format <http://servername:8080/tfs/collectionname> where “collectionname” is the name of the project collection you’re integrating with.
 - ▷ **For TFS 2005/2008** it is of the format <http://servername:8080>.
- **Login** – this should be set to a valid user that has permissions to access the TFS installation. The login needs to have permissions to create and view work items and iterations within TFS. Note: Do not include the Windows Active Directory Domain in this field if you are using a Windows domain user.
- **Password** – this should be set to the password of the user specified above.

- **Time Offset** – normally this should be set to zero, but if you find that work items being changed in TFS are not being updated in SpiraTeam, try increasing the value as this will tell the data-synchronization plug-in to add on the time offset (in hours) when comparing date-time stamps. Also if your TFS installation is running on a server set to a different time-zone, then you should add in the number of hours difference between the servers' time-zones here.

If you are using Visual Studio Online (VSO) instead of a local TFS instance, you will need to login to the instance of Visual Studio Online and access your profile:



Click on the Security Option. Then in the user security menu, choose the option to enter 'Alternate Credentials'. You need to now enter a login and password that SpiraTeam will use to connect to VSO:



The remaining fields work differently depending on which version of the TFS plugin you are using (2012, 2010 or 2005/2008):

a) TFS 2012 Plugin

Please fill out the fields as follows:

- **Auto-Map Users** – This changes the way the plugin maps users in SpiraTeam to those in TFS:
 - ▷ **Auto-Map = True**
With this setting, all users in SpiraTeam need to have the same username as those in TFS. If this is the case then you do not need to perform the user-mapping task outlined in

section 5.2.2. This is a big time-saver if you can guarantee that all usernames are the same in both systems.

▷ **Auto-Map = False**

With this setting, users in SpiraTeam and TFS are free to have different usernames because you specify the corresponding TFS name for each user as outlined in 5.2.2.

- **Custom 01** – This is used to specify the Windows Active Directory Domain that the Windows user specified above is a member of. If you are running TFS on a Windows workgroup, just use the server name and make sure that the Windows user above is a user on that server itself. If you are using a Window Live ID, just leave this field blank
- **Custom 02** – This field should contain a comma-separated list of work item types that you want to synchronize as SpiraTeam Tasks as opposed to Incidents. Normally you would want to list at least the TFS “Task” work item type in this field.
- **Custom 03** – If you would like the system to display the SpiraTeam artifact ID (e.g. IN5 for incidents or TK36 for tasks) in a custom field inside TFS, you should just enter the name of the appropriate TFS field from your process template (e.g. `Spira.IncidentId`) and then when the incident or task is added to TFS, the corresponding SpiraTeam ID will be added to that field of the work item.
- **Custom 04** - Depending on your TFS process template, the data-synchronization plugin may not be allowed to set the detector of the incident inside TFS. If you would like the system to display the detector of the incident (as recorded in SpiraTeam) in a custom field inside TFS, you should just enter the name of the appropriate TFS field from your process template (e.g. `Spira.Detector`) and then when the incident is added to TFS, the corresponding detector's name will be added to that field of the work item.
- **Custom 05** – This field should contain a comma-separated list of work item types that you want to synchronize as SpiraTeam Requirements as opposed to Incidents. Normally you would want to list at least the TFS “User Story” work item type in this field.

b) TFS 2010 Plugin

Please fill out the fields as follows:

- **Auto-Map Users** – this is not used by this version of the plugin and can be ignored.
- **Custom 01** – This is used to specify the Windows Active Directory Domain that the Windows user specified above is a member of. If you are running TFS on a Windows workgroup, just use the server name and make sure that the Windows user above is a user on that server itself.
- **Custom 02** – This is used to specify if you want to synchronize Incidents, Tasks or Both. By default this field is blank, meaning synchronize both types of artifact. However if you enter in “Incidents” into this field it will tell the system to only synchronize incidents, and if you enter “Tasks” it will tell the system to only synchronize tasks.
- **Custom 03** – If you would like the system to display the SpiraTeam artifact ID (e.g. IN5 for incidents or TK36 for tasks) in a custom field inside TFS, you should just enter the name of the appropriate TFS field from your process template (e.g. `Spira.IncidentId`) and then when the incident or task is added to TFS, the corresponding SpiraTeam ID will be added to that field of the work item.
- **Custom 04** - Depending on your TFS process template, the data-synchronization plugin may not be allowed to set the detector of the incident inside TFS. If you would like the system to display the detector of the incident (as recorded in SpiraTeam) in a custom field inside TFS, you should just enter the name of the appropriate TFS field from your process template (e.g. `Spira.Detector`) and then when the incident is added to TFS, the corresponding detector's name will be added to that field of the work item.
- **Custom 05** – this is not used by this version of the plugin and can be ignored.

c) TFS 2005/2008 Plugin

Please fill out the fields as follows:

- **Auto-Map Users** – this is not used by this version of the plugin and can be ignored.
- **Custom 01** – This is used to specify the Windows Active Directory Domain that the Windows user specified above is a member of. If you are running TFS on a Windows workgroup, just use the server name and make sure that the Windows user above is a user on that server itself.
- **Custom 02** – This is used to specify if you want to synchronize Incidents, Tasks or Both. By default this field is blank, meaning synchronize both types of artifact. However if you enter in “*Incidents*” into this field it will tell the system to only synchronize incidents, and if you enter “*Tasks*” it will tell the system to only synchronize tasks.
- **Custom 03** – this is not used by this version of the plugin and can be ignored.
- **Custom 04** – this is not used by this version of the plugin and can be ignored.
- **Custom 05** – this is not used by this version of the plugin and can be ignored.

5.2. Configuring the Data Mapping

Next, you need to configure the data mapping between SpiraTeam and TFS. This allows the various projects, users, releases, incident types, statuses, priorities and custom property values used in the two applications to be related to each other. This is important, as without a correct mapping, there is no way for the integration service to know that a “Not Reproducible” incident in SpiraTeam is the same as a “Closed + Cannot Reproduce” bug work item in TFS (for example).

The following mapping information needs to be setup in SpiraTeam:

- The mapping of the project identifiers for the projects that need to be synchronized
- The mapping of *users* in the system
- The mapping of *releases* (equivalent to TFS iterations) in the system
- The mapping of the various standard *incident* fields in the system
- The mapping of the various custom *incident* properties in the system
- The mapping of the various standard *requirement* fields in the system (if synching requirements)
- The mapping of the various custom *requirement* properties in the system (if synching requirements)
- The mapping of the various standard *task* fields in the system (if synching tasks)
- The mapping of the various custom *task* properties in the system (if synching tasks)

Note: If using SpiraTest, you do not need to setup the last two sets of mappings as Tasks are not available in SpiraTest.

5.2.1. Configuring the Project Mapping

From the data synchronization administration page, you need to click on the “View Project Mappings” hyperlink next to the TFS plug-in name. This will take you to the data-mapping home page for the currently selected project:

Microsoft TFS Project Data Mapping TFS Test Project 1

[← Back to Data Synchronization Home](#)
[↻ Change Project](#)

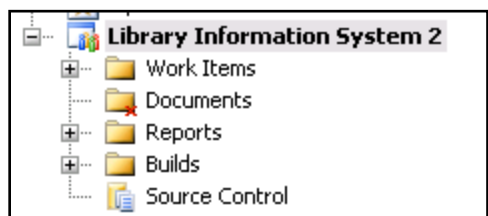
To configure this project for data mapping with the 'Microsoft TFS' plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name: TFS Test Project 1
External Key:
Active: Yes

If the project name does not match the name of the project you want to configure the data-mapping for, click on the “(Change Project)” hyperlink to change the current project.

To enable this project for data-synchronization with TFS, you need to enter:

- **External Key** – This should be set to the name of the project in TFS as visible from the Visual Studio Team Explorer:



- **Active Flag** – Set this to ‘Yes’ so that SpiraTeam knows that you want to synchronize data for this project. Once the project has been completed, setting the value to “No” will stop data synchronization, reducing network utilization.

Click [Update] to confirm these settings. Once you have enabled the project for data-synchronization, you can now enter the other data mapping values outlined below.

Note: *Once you have successfully configured the project, when creating a new project, you should choose the option to “Create Project from Existing Project” rather than “Use Default Template” so that all the project mappings get copied across to the new project.*

5.2.2. Configuring the User Mapping

To configure the mapping of users in the two systems, you need to go to Administration > Users > View Edit Users, which will bring up the list of users in the system. Then click on the “Edit” button for a particular user that will be editing work items in TFS:

System Administrator Edit User

Please review the information listed below and make any changes if necessary. Once you have finished, click [Save/Insert] to confirm them.

Save Cancel

Dates for Latest Activity by User

Last Lockout: - Last Activity: 2/24/2017 3:17:50 PM Last Login: 2/24/2017 3:08:46 PM Last Password Change: -

User Information

First Name:* System Department:
 Middle Initial: Organization:
 Last Name:* Administrator Email Address:* administrator@mycompany.com
 User Name:* administrator

Membership and Mapping

Project Membership Data Mapping

Microsoft TFS ID: Administrator

You will notice that in the special **Data Mapping** tab, there is a list of all the configured data-synchronization plug-ins. In the text box next to the TFS Data-Sync plug-in you need to enter the full name of this Windows User (not the login). This is the name of the user as they appear inside work items within TFS:

Assigned to: Administrator

Rank:

This will allow the data-synchronization plug-in to know which user in SpiraTeam match which equivalent user in TFS. Click [Update] once you've entered the appropriate login name. You should now repeat for the other users who will be active in both systems.

If you have set the "Auto-Map Users" option in the TFS 2012 plugin, you can skip this section completely.

5.2.3. Configuring the Release Mapping

When the data-synchronization service runs, when it comes across a release/iteration in SpiraTeam that it has not seen before, it will create a corresponding "Iteration" in TFS. Similarly if it comes across a new Iteration in TFS that it has not seen before, it will create a new Release/Iteration in SpiraTeam. Therefore when using both systems together, it is recommended that you only enter new Releases/Iterations in one system and let the data-synchronization service add them to the other system.

However you may start out with the situation where you already have pre-existing Releases/Iterations in both systems that you need to associate in the data-mapping. If you don't do this, you may find that duplicates get created when you first enable the data-synchronization service. Therefore for any Releases/Iterations that already exist in BOTH systems please navigate to Planning > Releases and click on the Release/Iteration in question. Make sure you have the 'Overview' tab visible and expand the "Details" section of the release/iteration:

Iteration 1 Release TFS-1 [RL:21]

Name:

Overview Incidents * Reqs & Tasks * Test Cases Test Runs * Attachments History

Details

Status: **In Progress** Creator: TFS DataSync Version #: TFS-1

Creation Date: 8/15/2013 4:20:03 PM Start Date: 8/14/2013 End Date: 8/19/2013

Resources: 1 Non-Working: 0 person days Plan Effort: 32.0h

Owner: -- None -- Last Updated: 8/15/2013 4:20:03 PM Type: Major Release

Available Effort: 32.0h

Microsoft TFS ID:

In addition to the standard fields and custom properties configured for Releases, you will see an additional text property called “**MsTfsDataSync ID**” that is used to store the mapped external identifier for the equivalent Version in TFS. You need to locate the ID of the equivalent Iteration in TFS, enter it into this text-box and click [Save]. You should now repeat for all the other pre-existing releases.

The TFS Iteration ID is not visible in the TFS user interface, but can instead be located by opening up the SQL Server that it’s installed on, opening the ‘TfsWorkItemTracking’ database (in TFS 2010 it will named after your project collection instead) and locating the ‘TreeNodes’ table:

ID	TypeID	Name	ParentID
0	-41	\	0
1	-42	Library Information System 2	0
2	-43	Project Iteration	1
3	-44	Iteration 1	2
4	-44	Iteration 2	2
5	-44	Iteration 0	2
6	-43	Project Structure	1
7	-44	Iteration 3	2
8	-45	Iteration 3.1	7
9	-45	Iteration 3.2	7

Once you have found the matching Iteration (by name), the numeric value stored in the ID column (the one on the left) is the value that needs to get added as the MsTfsDataSync ID inside SpiraTeam.

5.2.4. Configuring the Standard Incident Field Mapping

Now that the projects, user and releases have been mapped correctly, we need to configure the standard incident fields. To do this, go to Administration > System > Data Synchronization and click on the “View Project Mappings” for the MsTfsDataSync plug-in entry:

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact Type	Standard Fields	Custom Properties
Requirement	Component Importance Status Type	Area Risk
Test Case	Component Priority Status Type	
Incident	Component Priority Severity Status Type	Area Stack Rank Triage
Release		
Task	Component Priority Status Type	Area Discipline

From this screen, you need to click on Priority, Severity, Status and Type in turn to configure their values:

a) Incident Type

Click on the “Type” hyperlink under Incident Standard Fields to bring up the Incident type mapping configuration screen:

Edit Standard Field Data Mapping TFS Test Project 1

[← Back to Project Mappings](#)
[↻ Change Project](#)

To configure the data mappings for this field, please fill in the external key values for the **Microsoft TFS** plug-in next to each SpiraTeam value:

Artifact Type: **Incident**
Field Name: **Type**

Type	Active	External Key Microsoft TFS	Primary
Bug	Yes	<input type="text" value="Bug"/>	<input checked="" type="checkbox"/> Yes
Change Request	Yes	<input type="text"/>	<input checked="" type="checkbox"/> Yes
Enhancement	Yes	<input type="text"/>	<input checked="" type="checkbox"/> Yes
Incident	Yes	<input type="text" value="Bug"/>	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No
Issue	Yes	<input type="text" value="Issue"/>	<input checked="" type="checkbox"/> Yes
Limitation	Yes	<input type="text"/>	<input checked="" type="checkbox"/> Yes
Risk	Yes	<input type="text"/>	<input checked="" type="checkbox"/> Yes
Training	Yes	<input type="text"/>	<input checked="" type="checkbox"/> Yes

The table lists each of the incident types available in SpiraTeam and provides you with the ability to enter the matching TFS work item type name for each one. To make this easier, we recommend that inside the Administration > Edit Incident Statuses screen you first make all incident types inactive except *Risk*, *Issue* and *Bug* since only those types make sense to synchronize with TFS.

b) Incident Status

Click on the “Status” hyperlink under Incident Standard Fields to bring up the Incident status mapping configuration screen:

Edit Standard Field Data Mapping TFS Test Project 1

[← Back to Project Mappings](#) [↻ Change Project](#)

To configure the data mappings for this field, please fill in the external key values for the **Microsoft TFS** plug-in next to each SpiraTeam value:

Artifact Type: **Incident**
Field Name: **Status**

Status	Active	External Key Microsoft TFS	Primary
Assigned	Yes	<input type="text" value="Active+Accepted"/>	<input checked="" type="checkbox"/>
Closed	Yes	<input type="text" value="Resolved+Completed"/>	<input checked="" type="checkbox"/>
Duplicate	Yes	<input type="text" value="Resolved+Duplicate"/>	<input checked="" type="checkbox"/>
New	Yes	<input type="text" value="Active+New"/>	<input checked="" type="checkbox"/>
Not Reproducible	Yes	<input type="text" value="Resolved+Rejected"/>	<input checked="" type="checkbox"/>
Open	Yes	<input type="text" value="Active+Accepted"/>	<input type="checkbox"/> No
Reopen	Yes	<input type="text" value="Active+Reactivated"/>	<input checked="" type="checkbox"/>
Resolved	Yes	<input type="text" value="Resolved-Fixed"/>	<input checked="" type="checkbox"/>

The table lists each of the incident statuses available in SpiraTeam and provides you with the ability to enter the matching TFS work item **State + Reason** for each one. Since TFS uses separate State (Active, Resolved, Closed) and Reason (Fixed, Duplicate, Not Fixed, etc.) codes and SpiraTeam uses a single status code, you need to concatenate the TFS State and Reason together with a 'plus' (+) sign so that the system knows that the incident status in SpiraTeam corresponds to that specific combination.

You can map multiple SpiraTeam fields to the same TFS fields (e.g. New and Open in SpiraTeam are both equivalent to 'Active+New' in TFS), in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from TFS > SpiraTeam).

We recommend that you always point the New and Open statuses inside SpiraTeam to point to the "Active+New" TFS state+reason, and make Open in SpiraTeam the Primary status of the two. This is recommended so that as new incidents in SpiraTeam get synched over to TFS, they will get switched to the "Active+New" status in TFS which will then be synched back to "Open" in SpiraTeam. That way you'll be able to see at a glance which incidents have been synched with TFS and those that haven't.

c) Incident Priority

Click on the "Priority" hyperlink under Incident Standard Fields to bring up the Incident Priority mapping configuration screen:

Edit Standard Field Data Mapping TFS Test Project 1

[← Back to Project Mappings](#) [↻ Change Project](#)

To configure the data mappings for this field, please fill in the external key values for the **Microsoft TFS** plug-in next to each SpiraTeam value:

Artifact Type: **Incident**
Field Name: **Priority**

Priority	Active	External Key Microsoft TFS	Primary
1 - Critical	Yes	<input type="text" value="1"/>	<input checked="" type="checkbox"/>
2 - High	Yes	<input type="text" value="2"/>	<input checked="" type="checkbox"/>
3 - Medium	Yes	<input type="text" value="3"/>	<input checked="" type="checkbox"/>
4 - Low	Yes	<input type="text" value="4"/>	<input checked="" type="checkbox"/>

[Save](#)

The table lists each of the incident priorities available in SpiraTeam and provides you with the ability to enter the matching TFS priority value for each one. To make this easier, we recommend that inside the Administration > Edit Incident Priorities screen you first make any statuses not used in TFS inactive in SpiraTeam.

d) Incident Severity (TFS 2012 plugin only)

Click on the “Severity” hyperlink under Incident Standard Fields to bring up the Incident Severity mapping configuration screen:

Edit Standard Field Data Mapping TFS Test Project 1

[← Back to Project Mappings](#)
[↻ Change Project](#)

To configure the data mappings for this field, please fill in the external key values for the **Microsoft TFS** plug-in next to each SpiraTeam value:

Artifact Type: **Incident**
Field Name: **Severity**

Severity	Active	External Key Microsoft TFS	Primary
1 - Critical	Yes	<input type="text" value="1 - Critical"/>	<input checked="" type="checkbox"/>
2 - High	Yes	<input type="text" value="2 - High"/>	<input checked="" type="checkbox"/>
3 - Medium	Yes	<input type="text" value="3 - Medium"/>	<input checked="" type="checkbox"/>
4 - Low	Yes	<input type="text" value="4 - Low"/>	<input checked="" type="checkbox"/>

The table lists each of the incident severities available in SpiraTeam and provides you with the ability to enter the matching TFS severity value for each one. To make this easier, we recommend that inside the Administration > Edit Incident Severities screen you first make any statuses not used in TFS inactive in SpiraTeam.

5.2.5. Configuring the Incident Custom Property Mapping

Now that the various SpiraTeam standard incident fields have been mapped correctly, we need to configure the custom property mappings. This is used for both custom properties in SpiraTeam that map to custom fields in TFS and also for custom properties in SpiraTeam that are used to map to standard fields in TFS (e.g. Area) that don't exist in SpiraTeam.

From the View/Edit Project Data Mapping screen, you need to click on the name of the Incident Custom Property that you want to add data-mapping information for:

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact Type	Standard Fields	Custom Properties
Requirement	Component Importance Status Type	Area Risk
Test Case	Component Priority Status Type	
Incident	Component Priority Severity Status Type	Area Stack Rank Triage
Release		
Task	Component Priority Status Type	Area Discipline

a) TFS's Area Field

First you need to go to Administration > Edit Custom Lists and create a new custom list that contains all the different Areas that are being used in TFS.

Then you need to go to Administration > Edit Custom Properties and add a new *list* custom property onto the Incident artifact type called 'Area' and link it to the Area custom list you created in the previous step. This will now be available for mapping.

Now, back in the data-mapping page, click on the 'Area' hyperlink under Incident Custom Properties to bring up the custom property mapping configuration screen:

Edit Custom Property Data Mapping TFS Test Project 1

[← Back to Project Mappings](#) [↻ Change Project](#)

To configure the data mappings for this custom property, please fill in the **Microsoft TFS** external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Artifact Type: Incident

Name: Area [List](#)

External Key:

Custom Value Mapping

Value	Active	External Key <small>Microsoft TFS</small>
Administration	Yes	<input type="text" value="8"/>
Customization	Yes	<input type="text" value="9"/>
Management	Yes	<input type="text" value="10"/>
Reporting	Yes	<input type="text" value="11"/>

First you need to enter the word "Area" as the External Key of the custom property. This tells the data-sync plug-in that the custom property in SpiraTeam should be mapped to built-in Area field in TFS.

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the ID of the various Areas that are configured in TFS. The TFS Area ID is not visible in the TFS user interface, but can instead be located by opening up the SQL Server that it's installed on, opening the '[TfsWorkItemTracking](#)' database (in TFS 2010 and later it will named after your project collection instead) and locating the '[TreeNodes](#)' table:

ID	TypeID	Name
10	-44	Area 1
11	-44	Area 2
12	-44	Area 3

Once you have found the matching Area (by name), the numeric value stored in the ID column (the one on the left) is the value that needs to get added as the External Key inside SpiraTeam.

b) TFS Custom Fields

If the custom field in TFS is a list field, first you need to go to Administration > Edit Custom Lists in SpiraTeam and create a new custom list that contains all the different values that are being used in TFS.

Then for both list-fields and value-fields you need to go to Administration > Edit Custom Properties and add a new custom property onto the Incident artifact type with the name of the appropriate TFS field (e.g.

Triage, Rank, etc.) and if a list-field, link it to the custom list you created in the previous step. The custom property will now be available for data-mapping.

Now, back in the data-synchronization data-mapping page, click on the hyperlink under Incident Custom Properties that corresponds to the custom property to bring up the custom property mapping configuration screen:

Edit Custom Property Data Mapping TFS Test Project 1

[← Back to Project Mappings](#) [↻ Change Project](#)

To configure the data mappings for this custom property, please fill in the **Microsoft TFS** external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Artifact Type: Incident

Name: Triage List

External Key:

Custom Value Mapping

Value	Active	External Key Microsoft TFS
Approved	Yes	<input type="text" value="Approved"/>
Investigate	Yes	<input type="text" value="Investigate"/>

First you need to enter the full **Reference Name** of the TFS field as the External Key of the custom property. This tells the data-sync plug-in that the custom property in SpiraTeam should be mapped to this specific field in TFS. To see a list of fields and their reference names, you can run the following SQL query against your TFS database:

```
SELECT Name, ReferenceName FROM Fields ORDER BY Name
```

We have included a list of fields in the Agile process template in section 5.5 of this guide as a helpful reference.

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the name of the field values as they appear in TFS as the External Key.

5.2.6. Configuring the Standard Task Field Mapping

Now that the projects, user, releases and incident fields have been mapped correctly, we need to configure the standard task fields. To do this, go to Administration > System > Data Synchronization and click on the “View Project Mappings” for the MsTfsDataSync plug-in entry:

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact Type	Standard Fields	Custom Properties
Requirement	Component Importance Status Type	Area Risk
Test Case	Component Priority Status Type	
Incident	Component Priority Severity Status Type	Area Stack Rank Triage
Release		
Task	Component Priority Status Type	Area Discipline

From this screen, you need to click on Priority and Status in turn to configure their values:

a) Task Status

Click on the “Status” hyperlink under Task Standard Fields to bring up the Task status mapping configuration screen:

Edit Standard Field Data Mapping TFS Test Project 1

[← Back to Project Mappings](#)
[↻ Change Project](#)

To configure the data mappings for this field, please fill in the external key values for the **Microsoft TFS** plug-in next to each SpiraTeam value:

Artifact Type: **Task**
Field Name: **Status**

Status	Active	External Key <small>Microsoft TFS</small>	Primary
Blocked	Yes	<input type="text" value="Removed"/>	<input checked="" type="checkbox"/> Yes
Completed	Yes	<input type="text" value="Closed"/>	<input checked="" type="checkbox"/> Yes
Deferred	Yes	<input type="text" value="Removed"/>	<input type="checkbox"/> No
In Progress	Yes	<input type="text" value="Active"/>	<input checked="" type="checkbox"/> Yes
Not Started	Yes	<input type="text" value="New"/>	<input checked="" type="checkbox"/> Yes

The table lists each of the task statuses available in SpiraTeam and provides you with the ability to enter the matching TFS work item State for each one. Unlike the mapping for incidents (see above) SpiraTeam does not track the reason codes associated with the tasks in MS TFS, so you only need to map the State names from TFS with the task status names.

You can map multiple SpiraTeam fields to the same TFS fields (e.g. Blocked, Completed and Deferred in SpiraTeam are all equivalent to State = Closed in TFS), in which case only one of the values can be listed as Primary = Yes as that’s the value that’s used on the reverse synchronization (from TFS > SpiraTeam).

b) Task Priority (TFS 2012 Plugin Only)

Click on the “Priority” hyperlink under Task Standard Fields to bring up the Task Priority mapping configuration screen:

Edit Standard Field Data Mapping TFS Test Project 1

[← Back to Project Mappings](#) [↻ Change Project](#)

To configure the data mappings for this field, please fill in the external key values for the **Microsoft TFS** plug-in next to each SpiraTeam value:

Artifact Type: **Task**
Field Name: **Priority**

Priority	Active	External Key Microsoft TFS	Primary
1 - Critical	Yes	<input type="text" value="1"/>	<input checked="" type="checkbox"/>
2 - High	Yes	<input type="text" value="2"/>	<input checked="" type="checkbox"/>
3 - Medium	Yes	<input type="text" value="3"/>	<input checked="" type="checkbox"/>
4 - Low	Yes	<input type="text" value="4"/>	<input checked="" type="checkbox"/>

The table lists each of the task priorities available in SpiraTeam and provides you with the ability to enter the matching TFS priority value for each one.

5.2.7. Configuring the Task Custom Property Mapping

Now that the various SpiraTeam standard task fields have been mapped correctly, we need to configure the custom property mappings. This is used for both custom properties in SpiraTeam that map to custom fields in TFS and also for custom properties in SpiraTeam that are used to map to standard fields in TFS (e.g. Area) that don't exist in SpiraTeam.

From the View/Edit Project Data Mapping screen, you need to click on the name of the Task Custom Property that you want to add data-mapping information for:

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact Type	Standard Fields	Custom Properties
Requirement	Component Importance Status Type	Area Risk
Test Case	Component Priority Status Type	
Incident	Component Priority Severity Status Type	Area Stack Rank Triage
Release		
Task	Component Priority Status Type	Area Discipline

a) TFS's Area Field

First you need to go to Administration > Edit Custom Lists and create a new custom list that contains all the different Areas that are being used in TFS.

Then you need to go to Administration > Edit Custom Properties and add a new *list* custom property onto the Task artifact type called 'Area' and link it to the Area custom list you created in the previous step. This will now be available for mapping.

Now, back in the data-mapping page, click on the 'Area' hyperlink under Task Custom Properties to bring up the custom property mapping configuration screen:

Edit Custom Property Data Mapping TFS Test Project 1

[← Back to Project Mappings](#) [↻ Change Project](#)

To configure the data mappings for this custom property, please fill in the **Microsoft TFS** external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Artifact Type:
Name:
External Key:

Custom Value Mapping

Value	Active	External Key
Administration	Yes	<input type="text" value="8"/>
Customization	Yes	<input type="text" value="9"/>
Management	Yes	<input type="text" value="10"/>
Reporting	Yes	<input type="text" value="11"/>

First you need to enter the word “Area” as the External Key of the custom property. This tells the data-sync plug-in that the custom property in SpiraTeam should be mapped to built-in Area field in TFS.

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the ID of the various Areas that are configured in TFS. The TFS Area ID is not visible in the TFS user interface, but can instead be located by opening up the SQL Server that it’s installed on, opening the ‘TfsWorkItemTracking’ database (in TFS 2010 and later it will named after your project collection instead) and locating the ‘TreeNodes’ table:

ID	TypeID	Name
10	-44	Area 1
11	-44	Area 2
12	-44	Area 3

Once you have found the matching Area (by name), the numeric value stored in the ID column (the one on the left) is the value that needs to get added as the External Key inside SpiraTeam.

b) TFS Custom Fields

If the custom field in TFS is a list field, first you need to go to Administration > Edit Custom Lists in SpiraTeam and create a new custom list that contains all the different values that are being used in TFS.

Then for both list-fields and value-fields you need to go to Administration > Edit Custom Properties and add a new custom property onto the Task artifact type with the name of the appropriate TFS field (e.g. Discipline, Stack Rank, etc.) and if a list-field, link it to the custom list you created in the previous step. The custom property will now be available for data-mapping.

Now, back in the data-synchronization data-mapping page, click on the hyperlink under Task Custom Properties that corresponds to the custom property to bring up the custom property mapping configuration screen:

Edit Custom Property Data Mapping TFS Test Project 1

[← Back to Project Mappings](#) [↻ Change Project](#)

To configure the data mappings for this custom property, please fill in the **Microsoft TFS** external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Artifact Type:* Task
Name:* Discipline [List](#)
External Key:*

Custom Value Mapping

Value	Active	External Key Microsoft TFS
Architecture	Yes	<input type="text" value="Architecture"/>
Development	Yes	<input type="text" value="Development"/>
Project Management	Yes	<input type="text" value="Project Management"/>
Release Management	Yes	<input type="text" value="Release Management"/>
Requirements	Yes	<input type="text" value="Requirements"/>
Test	Yes	<input type="text" value="Test"/>

First you need to enter the full **Reference Name** of the TFS field as the External Key of the custom property. This tells the data-sync plug-in that the custom property in SpiraTeam should be mapped to this specific field in TFS. To see a list of fields and their reference names, you can run the following SQL query against your TFS database:

```
SELECT Name, ReferenceName FROM Fields ORDER BY Name
```

We have included a list of fields in the Agile process template in section 5.5 of this guide as a helpful reference.

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the name of the field values as they appear in TFS as the External Key.

5.2.8. Configuring the Standard Requirement Field Mapping (2012 Plugin Only)

Now that the projects, user, releases, incident and task fields have been mapped correctly, we need to configure the standard requirement fields. To do this, go to Administration > System > Data Synchronization and click on the “View Project Mappings” for the MsTfsDataSync plug-in entry:

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact Type	Standard Fields	Custom Properties
Requirement	Component Importance Status Type	Area Risk
Test Case	Component Priority Status Type	
Incident	Component Priority Severity Status Type	Area Stack Rank Triage
Release		
Task	Component Priority Status Type	Area Discipline

From this screen, you need to click on Importance and Status in turn to configure their values:

a) Requirement Status

Click on the “Status” hyperlink under Requirement Standard Fields to bring up the Requirement status mapping configuration screen:

Edit Standard Field Data Mapping TFS Test Project 1

[← Back to Project Mappings](#)
[↻ Change Project](#)

To configure the data mappings for this field, please fill in the external key values for the **Microsoft TFS** plug-in next to each SpiraTeam value:

Artifact Type: **Requirement**
Field Name: **Status**

Status	Active	External Key Microsoft TFS	Primary
Accepted	Yes	Accepted	<input checked="" type="checkbox"/>
Completed	Yes		<input checked="" type="checkbox"/>
Developed	Yes	Closed	<input checked="" type="checkbox"/>
In Progress	Yes	Active	<input checked="" type="checkbox"/>
Obsolete	Yes	Removed	<input checked="" type="checkbox"/>
Planned	Yes	Ready	<input checked="" type="checkbox"/>
Rejected	Yes	Rejected	<input checked="" type="checkbox"/>
Requested	Yes	New	<input checked="" type="checkbox"/>
Tested	Yes		<input checked="" type="checkbox"/>
Under Review	Yes	Design	<input checked="" type="checkbox"/>

The table lists each of the requirement statuses available in SpiraTeam and provides you with the ability to enter the matching TFS work item State for each one. Unlike the mapping for incidents (see above) SpiraTeam does not track the reason codes associated with the requirements in MS TFS, so you only need to map the State names from TFS with the requirement status names.

You can map multiple SpiraTeam fields to the same TFS fields, in which case only one of the values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from TFS > SpiraTeam).

b) Requirement Importance

Click on the “Importance” hyperlink under Requirement Standard Fields to bring up the Requirement Importance mapping configuration screen:

Edit Standard Field Data Mapping TFS Test Project 1

← Back to Project Mappings
↻ Change Project

To configure the data mappings for this field, please fill in the external key values for the **Microsoft TFS** plug-in next to each SpiraTeam value:

Artifact Type: **Requirement**
Field Name: **Importance**

Importance	Active	External Key	Microsoft TFS	Primary
1 - Critical	Yes		1	<input checked="" type="checkbox"/>
2 - High	Yes		2	<input checked="" type="checkbox"/>
3 - Medium	Yes		3	<input checked="" type="checkbox"/>
4 - Low	Yes		4	<input checked="" type="checkbox"/>

The table lists each of the requirement importance values available in SpiraTeam and provides you with the ability to enter the matching TFS work item priority value for each one.

5.2.9. Configuring the Requirement Custom Property Mapping (2012 Plugin Only)

Now that the various SpiraTeam standard requirement fields have been mapped correctly, we need to configure the custom property mappings. This is used for both custom properties in SpiraTeam that map to custom fields in TFS and also for custom properties in SpiraTeam that are used to map to standard fields in TFS (e.g. Area) that don't exist in SpiraTeam.

From the View/Edit Project Data Mapping screen, you need to click on the name of the Requirement Custom Property that you want to add data-mapping information for:

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact Type	Standard Fields	Custom Properties
Requirement	Component Importance Status Type	Area Risk
Test Case	Component Priority Status Type	
Incident	Component Priority Severity Status Type	Area Stack Rank Triage
Release		
Task	Component Priority Status Type	Area Discipline

a) TFS's Area Field

First you need to go to Administration > Edit Custom Lists and create a new custom list that contains all the different Areas that are being used in TFS.

Then you need to go to Administration > Edit Custom Properties and add a new *list* custom property onto the Requirement artifact type called 'Area' and link it to the Area custom list you created in the previous step. This will now be available for mapping.

Now, back in the data-mapping page, click on the 'Area' hyperlink under Requirement Custom Properties to bring up the custom property mapping configuration screen:

Edit Custom Property Data Mapping TFS Test Project 1

← Back to Project Mappings
↻ Change Project

To configure the data mappings for this custom property, please fill in the **Microsoft TFS** external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Artifact Type: Requirement

Name: Area List

External Key:

Custom Value Mapping

Value	Active	External Key Microsoft TFS
Administration	Yes	<input style="width: 100%;" type="text" value="8"/>
Customization	Yes	<input style="width: 100%;" type="text" value="9"/>
Management	Yes	<input style="width: 100%;" type="text" value="10"/>
Reporting	Yes	<input style="width: 100%;" type="text" value="11"/>

First you need to enter the word "Area" as the External Key of the custom property. This tells the data-sync plug-in that the custom property in SpiraTeam should be mapped to built-in Area field in TFS.

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the ID of the various Areas that are configured in TFS. The TFS Area ID is not visible in the TFS user interface, but can instead be located by opening up the SQL Server that it's installed on, opening the 'TfsWorkItemTracking' database (in TFS 2010 and later it will named after your project collection instead) and locating the 'TreeNodes' table:

ID	TypeID	Name
10	-44	Area 1
11	-44	Area 2
12	-44	Area 3

Once you have found the matching Area (by name), the numeric value stored in the ID column (the one on the left) is the value that needs to get added as the External Key inside SpiraTeam.

b) TFS Custom Fields

If the custom field in TFS is a list field, first you need to go to Administration > Edit Custom Lists in SpiraTeam and create a new custom list that contains all the different values that are being used in TFS.

Then for both list-fields and value-fields you need to go to Administration > Edit Custom Properties and add a new custom property onto the Requirement artifact type with the name of the appropriate TFS field (e.g. Risk, Stack Rank, etc.) and if a list-field, link it to the custom list you created in the previous step. The custom property will now be available for data-mapping.

Now, back in the data-synchronization data-mapping page, click on the hyperlink under Requirement Custom Properties that corresponds to the custom property to bring up the custom property mapping configuration screen:

Edit Custom Property Data Mapping TFS Test Project 1

[← Back to Project Mappings](#)
[↻ Change Project](#)

To configure the data mappings for this custom property, please fill in the **Microsoft TFS** external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Artifact Type: Requirement
Name: Risk List
External Key:

Custom Value Mapping

Value	Active	External Key <small>Microsoft TFS</small>
1 - High	Yes	<input type="text"/>
2 - Medium	Yes	<input type="text"/>
3 - Low	Yes	<input type="text"/>

First you need to enter the full **Reference Name** of the TFS field as the External Key of the custom property. This tells the data-sync plug-in that the custom property in SpiraTeam should be mapped to this specific field in TFS. To see a list of fields and their reference names, you can run the following SQL query against your TFS database:

```
SELECT Name, ReferenceName FROM Fields ORDER BY Name
```

We have included a list of fields in the Agile process template in section 5.5 of this guide as a helpful reference.

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the name of the field values as they appear in TFS as the External Key.

Once you have updated the various mapping sections, you are now ready to start the service.

5.3. Using SpiraTeam with TFS

Now that the integration service has been configured and the service started, initially any incidents already created in SpiraTeam for the specified projects will be imported into TFS and any requirements, tasks or bugs already created in TFS will be imported into SpiraTeam. At this point we recommend opening the Windows Event Viewer and choosing the Application Log. In this log any error messages raised by the SpiraTeam Data Sync Service will be displayed. If you see any error messages at this point, we recommend immediately stopping the SpiraTeam service and checking the various mapping entries. If you cannot see any work items with the mapping information, we recommend sending a copy of the event log message(s) to Inflectra customer services (support@inflectra.com) who will help you troubleshoot the problem.

To use SpiraTeam with TFS on an ongoing basis, we recommend the following general processes be followed:

- When running tests in SpiraTest or SpiraTeam, defects found should be logged through the Test Execution Wizard as normal.
 - ▷ Once an incident has been created during the running of the test, it will now be populated across into TFS as a work item of type corresponding to the types setup in the incident type mappings.

- ▷ At this point, the incident can be worked on in either system, with changes being synchronized to the other system. However in general we recommend that the QA/Testing team use SpiraTeam and the development team use TFS. E.g. the developers will mark the bugs as resolved in MSTs once they have completed fixing them and the QA team will either reopen or close them in SpiraTeam once they have had a change to verify the resolution.
- ▷ You are now able to perform test coverage and incident reporting inside SpiraTest/SpiraTeam using the test cases managed by SpiraTest/SpiraTeam and the incidents managed collaboratively between SpiraTest/SpiraTeam and TFS.
- ▷ You can create project requirements and associated tasks in either SpiraTeam or TFS, however the synchronization service is only unidirectional for requirements and tasks, so when you create or update a requirement or task in TFS, the change will be reflected in SpiraTeam, but not the other way around.

5.4. Troubleshooting

In most cases once you have started the service, once it's up and running you will not see any error or warning messages from the Data-Sync service. However, if you have new users created in SpiraTeam that have not been mapped to users in TFS, when you assign incidents, requirements or tasks to those items, you may see warning messages in the Event Viewer letting you know which users needs to be mapped.

5.5. TFS Field Reference

The following fields are available in TFS for data-mapping when using the TFS agile process template:

Display Name	Reference Name
Accepted By	Microsoft.VSTS.CodeReview.AcceptedBy
Accepted Date	Microsoft.VSTS.CodeReview.AcceptedDate
Activated By	Microsoft.VSTS.Common.ActivatedBy
Activated Date	Microsoft.VSTS.Common.ActivatedDate
Activity	Microsoft.VSTS.Common.Activity
Application Launch Instructions	Microsoft.VSTS.Feedback.ApplicationLaunchInstructions
Application Start Information	Microsoft.VSTS.Feedback.ApplicationStartInformation
Application Type	Microsoft.VSTS.Feedback.ApplicationType
Area ID	System.Areald
Area Level 1	System.AreaLevel1
Area Level 2	System.AreaLevel2
Area Level 3	System.AreaLevel3
Area Level 4	System.AreaLevel4
Area Level 5	System.AreaLevel5
Area Level 6	System.AreaLevel6
Area Level 7	System.AreaLevel7
Area Path	System.AreaPath
Assigned To	System.AssignedTo
Associated Context	Microsoft.VSTS.CodeReview.Context
Associated Context Code	Microsoft.VSTS.CodeReview.ContextCode
Associated Context Owner	Microsoft.VSTS.CodeReview.ContextOwner
Associated Context Type	Microsoft.VSTS.CodeReview.ContextType
Attached File Count	System.AttachedFileCount

Attached Files	System.AttachedFiles
Authorized As	System.AuthorizedAs
Authorized Date	System.AuthorizedDate
Automated Test Id	Microsoft.VSTS.TCM.AutomatedTestId
Automated Test Name	Microsoft.VSTS.TCM.AutomatedTestName
Automated Test Storage	Microsoft.VSTS.TCM.AutomatedTestStorage
Automated Test Type	Microsoft.VSTS.TCM.AutomatedTestType
Automation status	Microsoft.VSTS.TCM.AutomationStatus
BIS Links	System.BISLinks
Changed By	System.ChangedBy
Changed Date	System.ChangedDate
Changed Set	System.ChangedSet
Closed By	Microsoft.VSTS.Common.ClosedBy
Closed Date	Microsoft.VSTS.Common.ClosedDate
Closed Status	Microsoft.VSTS.CodeReview.ClosedStatus
Closed Status Code	Microsoft.VSTS.CodeReview.ClosedStatusCode
Closing Comment	Microsoft.VSTS.CodeReview.ClosingComment
Completed Work	Microsoft.VSTS.Scheduling.CompletedWork
Created By	System.CreatedBy
Created Date	System.CreatedDate
Description	System.Description
Due Date	Microsoft.VSTS.Scheduling.DueDate
External Link Count	System.ExternalLinkCount
Finish Date	Microsoft.VSTS.Scheduling.FinishDate
Found In	Microsoft.VSTS.Build.FoundIn
History	System.History
Hyperlink Count	System.HyperLinkCount
ID	System.Id
InAdminOnlyTreeFlag	System.InAdminOnlyTreeFlag
InDeletedTreeFlag	System.InDeletedTreeFlag
Integration Build	Microsoft.VSTS.Build.IntegrationBuild
Issue	Microsoft.VSTS.Common.Issue
Iteration ID	System.IterationId
Iteration Level 1	System.IterationLevel1
Iteration Level 2	System.IterationLevel2
Iteration Level 3	System.IterationLevel3
Iteration Level 4	System.IterationLevel4
Iteration Level 5	System.IterationLevel5
Iteration Level 6	System.IterationLevel6
Iteration Level 7	System.IterationLevel7
Iteration Path	System.IterationPath
Link Type	System.Links.LinkType
Linked Files	System.LinkedFiles
Local Data Source	Microsoft.VSTS.TCM.LocalDataSource
Node Name	System.NodeName
Node Type	System.NodeType
Not a field	System.NotAField
Original Estimate	Microsoft.VSTS.Scheduling.OriginalEstimate

Parameters	Microsoft.VSTS.TCM.Parameters
PersonID	System.PersonId
Priority	Microsoft.VSTS.Common.Priority
ProjectID	System.ProjectId
Rating	Microsoft.VSTS.Common.Rating
Reason	System.Reason
Related Link Count	System.RelatedLinkCount
Related Links	System.RelatedLinks
Remaining Work	Microsoft.VSTS.Scheduling.RemainingWork
Repro Steps	Microsoft.VSTS.TCM.ReproSteps
Resolved By	Microsoft.VSTS.Common.ResolvedBy
Resolved Date	Microsoft.VSTS.Common.ResolvedDate
Resolved Reason	Microsoft.VSTS.Common.ResolvedReason
Rev	System.Rev
Reviewed By	Microsoft.VSTS.Common.ReviewedBy
Revised Date	System.RevisedDate
Risk	Microsoft.VSTS.Common.Risk
Severity	Microsoft.VSTS.Common.Severity
Stack Rank	Microsoft.VSTS.Common.StackRank
Start Date	Microsoft.VSTS.Scheduling.StartDate
State	System.State
State Change Date	Microsoft.VSTS.Common.StateChangeDate
State Code	Microsoft.VSTS.Common.StateCode
Steps	Microsoft.VSTS.TCM.Steps
Story Points	Microsoft.VSTS.Scheduling.StoryPoints
System Info	Microsoft.VSTS.TCM.SystemInfo
Tags	System.Tags
Team Project	System.TeamProject
TF Server	System.TFServer
Title	System.Title
Tree	System.Tree
Watermark	System.Watermark
WEF_BD66C4E18FB54884A18B2299E91ADE1B_Extension Marker	WEF_BD66C4E18FB54884A18B2299E91ADE1B_System.ExtensionMarker
WEF_BD66C4E18FB54884A18B2299E91ADE1B_Kanban Column	WEF_BD66C4E18FB54884A18B2299E91ADE1B_Kanban.Column
Work Item Form	System.WorkItemForm
Work Item FormID	System.WorkItemFormId
Work Item Type	System.WorkItemType
WorkItem	System.WorkItem
WorkItemLink	System.WorkItemLink
WorkItemTypeExtension	System.WorkItemTypeExtension

For a full list of the available TFS fields in the different process templates, please refer to:
<http://msdn.microsoft.com/en-us/library/vstudio/dd997792.aspx>

6. Using SpiraTest with FogBugz

This section outlines how to use SpiraTest, SpiraPlan or SpiraTeam (hereafter referred to as SpiraTeam) in conjunction with the FogBugz issue/bug tracking system. The built-in integration service allows the quality assurance team to manage their requirements and test cases in SpiraTeam, execute test runs in SpiraTest, and then have the new incidents generated during the run be automatically loaded into FogBugz.

Once the incidents are loaded into FogBugz as cases, the development team can then manage the lifecycle of these cases in FogBugz, and have the status changes in FogBugz be reflected back in SpiraTeam. In addition, any cases logged into FogBugz will get imported into SpiraTeam so that they can be linked to test cases and requirements.

► **STOP! Please make sure you have first read the Instructions in Section 1 before proceeding!**

6.1. Configuring the Plug-In

The next step is to configure the plug-in within SpiraTeam so that the system knows how to access the FogBugz server. To start the configuration, please open up SpiraTeam in a web browser, log in using a valid account that has System-Administration level privileges and click on the System > Data Synchronization administration option from the left-hand navigation:

Data Synchronization

SpiraTeam is able to communicate with a variety of external systems and applications using its background data synchronization service and library of available plug-ins.

This page allows you to view, add and modify the list of plug-ins, make changes to their configuration, and verify the status of the most recent synchronization. In addition, you can force a particular plug-in to perform a complete data-synchronization by resetting its date of last sync.

Refresh + Add

Plug-In	Data Mapping	Last Sync Date	Status	Operations
Bugzilla BugzillaDataSync	View Project Mappings	-	Not Run	Reset Sync Edit Delete View Errors
FogBugz FogBugzDataSync	View Project Mappings	-	Not Run	Reset Sync Edit Delete View Errors
Jira JiraDataSync	View Project Mappings	-	Not Run	Reset Sync Edit Delete View Errors
Microsoft TFS MsTfsDataSync	View Project Mappings	-	Not Run	Reset Sync Edit Delete View Errors

This screen lists all the plug-ins already configured in the system. Depending on whether you chose the option to include sample data in your installation or not, you will see either an empty screen or a list of sample data-synchronization plug-ins.

If you already see an entry for **FogBugzDataSync** you should click on its “Edit” link. If you don’t see such an entry in the list, please click on the [Add] button instead. In either case you will be taken to the following screen where you can enter or modify the FogBugz Data-Synchronization plug-in:

FogBugz Edit Plug-In

[← Back to Data Synchronization Home](#)

Please enter/edit the following information for the plug-in. The exact information that needs to be entered is specific to each plug-in, and you should refer to the documentation for the plug-in when entering/changing it:

Name:
The physical name of the plugin (e.g. JiraDataSync) that matches the DLL being used

Caption:
The display name of the plugin (e.g. Internal Jira) that can be used to distinguish between multiple instances of the same plugin type

Description:
The long description of the plugin (e.g. 'Connects to the in-house bug-tracker')

Connection Info:
The URL or other connection string used to connect to the external bug-tracker

Login:

Password:
The login and password used to connect to FogBugz

You need to fill out the following fields for the FogBugz Plug-in to operate correctly:

- **Name** – this needs to be set to **FogBugzDataSync**. This needs to match the name of the plug-in DLL assembly that was copied into the C:\Program Files\SpiraTeam\Bin folder (minus the .dll file extension). If you renamed the FogBugzDataSync.dll file for any reason, then you need to change the name here to match.
- **Caption** – this is the display name of the plugin, typically just “FogBugz”. If you have multiple instances of FogBugz, they could have different captions.
- **Description** – this should be set to a description of the plug-in. This is an optional field that is used for documentation purposes and is not actually used by the system.
- **Connection Info** – this should be the URL that you use to access your instance of FogBugz (e.g. https://mycompany.fogbugz.com)
- **Login** – this should be set to a valid login to the FogBugz installation. The login needs to have permissions to create and view cases and versions within FogBugz.
- **Password** – this should be set to the password of the login specified above.

Time Offset: hours
The number of hours to look back for changes to synchronization. Normally should be set to zero (0) unless you find changes are not being synchronized

Auto-Map Users:

Custom 01:

Custom 02:

Custom 03:

Custom 04:

Custom 05:

- **Time Offset** – normally this should be set to zero, but if you find that cases being changed in FogBugz are not being updated in SpiraTeam, try increasing the value as this will tell the data-synchronization plug-in to add on the time offset (in hours) when comparing date-time stamps. Also if your FogBugz installation is running on a server set to a different time-zone, then you should add in the number of hours difference between the servers' time-zones here.
- **Auto-Map Users** – this is not currently used by the FogBugz data-sync plug-in and can be ignored.
- **Custom 01** – When connecting to FogBugz, sometimes the connection gets dropped by the server without notifying the plug-in. This happens when using HTTP 1.1 Keep-Alive connections. If you set this property to “**False**”, it will tell the plug-in to **not-use** HTTP keep-alives when connecting to FogBugz, otherwise set it to “**True**”.
- **Custom 02** – When connecting to a FogBugz instance that is running under HTTPS (SSL) this custom property can be set to determine if the plug-in should verify that the SSL certificate is a trusted root certificate. Set to “**True**” if you are using an SSL certificate that was issued by a trusted Certification Authority, and set to “**False**” if you are using a self-signed certificate.
- **Custom 03** – Normally all rich text (HTML) descriptions in SpiraTeam are converted into plain text when added to FogBugz. However, more recent version of FogBugz can now support rich text. So if you have rich-text enabled in your instance of FogBugz, you should enter the word “**True**” in Custom 03 to enable rich text description transfer.
- **Custom 04** – Normally you can leave this blank. However if you want to prevent the plugin from getting new cases from FogBugz (that did not originate in SpiraTest), set it to “**False**”.
- **Custom 05** – this is not currently used by the FogBugz data-sync plug-in and can be left blank.

6.2. Configuring the Data Mapping

Next, you need to configure the data mapping between SpiraTeam and FogBugz. This allows the various projects, users, releases, incident types, statuses, priorities and custom property values used in the two applications to be related to each other. This is important, as without a correct mapping, there is no way for the integration service to know that an “Enhancement” in SpiraTeam is the same as a “Feature” in FogBugz (for example).

The following mapping information needs to be setup in SpiraTeam:

- The mapping of the project identifiers for the projects that need to be synchronized
- The mapping of users in the system
- The mapping of releases (equivalent to FogBugz releases/fix-fors) in the system
- The mapping of the various standard fields in the system
- The mapping of the various custom properties in the system

Each of these is explained in turn below:

6.2.1. Configuring the Project Mapping

From the data synchronization administration page, you need to click on the “View Project Mappings” hyperlink next to the FogBugz plug-in name. This will take you to the data-mapping home page for the currently selected project:

FogBugz Project Data Mapping Library Information System

[← Back to Data Synchronization Home](#)
[↻ Change Project](#)

To configure this project for data mapping with the 'FogBugz' plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name*: Library Information System
 External Key*:
 Active*: Yes

If the project name does not match the name of the project you want to configure the data-mapping for, click on the “(Change Project)” hyperlink to change the current project.

To enable this project for data-synchronization with FogBugz, you need to enter:

- **External Key** – This should be set to the ID of the project in FogBugz. This can be found by navigating to Settings > Projects in FogBugz:

Edit	Delete	Project	Client/Department	Primary Contact	Report	Description
		Inbox	Internal	Adam		Inbox for FogBugz Autosort
		Sample Project	Internal	Adam		
		Create New Project				

Then hover the mouse over the project name. The project ID will be displayed in the URL line as ixProject=X where X is the numeric ID of the project.

- **Active Flag** – Set this to ‘Yes’ so that SpiraTeam knows that you want to synchronize data for this project. Once the project has been completed, setting the value to “No” will stop data synchronization, reducing network utilization.

Click [Update] to confirm these settings. Once you have enabled the project for data-synchronization, you can now enter the other data mapping values outlined below.

Note: Once you have successfully configured the project, when creating a new project, you should choose the option to “Create Project from Existing Project” rather than “Use Default Template” so that all the project mappings get copied across to the new project.

6.2.2. Configuring the User Mapping

To configure the mapping of users in the two systems, you need to go to Administration > Users > View Edit Users, which will bring up the list of users in the system. Then click on the “Edit” button for a particular user that will be editing cases in FogBugz:

Add/Edit User | Fred Bloggs

Please review the information listed below and make any changes if necessary. Once you have finished, click [Update/Insert] to confirm them.

First Name*:

Middle Initial:

Last Name*:

User Name*:

Email Address*:

System Administrator*:

Active User*:

FogBugzData Sync ID:

LDAP Managed User

LDAP DN:

Change Password

New Password:

Confirm Password:

You will notice that below the Active flag for the user is a list of all the configured data-synchronization plug-ins. In the text box next to the FogBugz Data-Sync plug-in you need to enter the ID of this user in FogBugz. This will allow the data-synchronization plug-in to know which user in SpiraTeam match which equivalent user in FogBugz. The ID can be found in FogBugz by going to Settings > Users:

Edit	Name	Email	Schedule	Status	Phone	Description
	Fred Bloggs	fredbloggs@inflectra.com		Active		No email
	joesmith	joesmith@inflectra.com		Active		
	System Administrator	webmaster@inflectra.com		Active		Site Admin
Add New Person						

Then hover the mouse over the user's name. The user ID will be displayed in the URL line as ixPerson=X where X is the numeric ID of the user.

Back in SpiraTeam, click [Update] once you've entered the appropriate user ID in the mapping box. You should now repeat for the other users who will be active in both systems.

6.2.3. Configuring the Release Mapping

When the data-synchronization service runs, when it comes across a release/iteration in SpiraTeam that it has not seen before, it will create a corresponding Release/Fix-For in FogBugz. Similarly if it comes across a new Release/Fix-For in FogBugz that it has not seen before, it will create a new Release in SpiraTeam. Therefore when using both systems together, it is recommended that you only enter new Releases/Versions in one system and let the data-synchronization service add them to the other system.

However you may start out with the situation where you already have pre-existing Releases/Versions in both systems that you need to associate in the data-mapping. If you don't do this, you may find that duplicates get created when you first enable the data-synchronization service. Therefore for any Releases/Iterations that already exist in BOTH systems please navigate to Planning > Releases and click on the Release/Iteration in question. Make sure you have the 'Overview' tab visible and expand the "Details" section of the release/iteration:

Release: 2.0B - Version 2.0B [RL:000022]

Name: Version 2.0B

Overview Incidents Reqs & Tasks Test Cases Test Runs Attachments History #

Details

Version #: 2.0B Creator*: System Administrator

Start/End Date: 7/6/2013 8/6/2013 Creation Date: 6/5/2013 6:00:56 PM

Resources: 1 Active*: Yes

Non-Working: 0 person days Iteration*: No

Plan Effort: 176.00 Available Effort: 176.00

FogBugzDataSync ID: 8

In addition to the standard fields and custom properties configured for Releases, you will see an additional text property called “**FogBugzDataSync ID**” that is used to store the mapped external identifier for the equivalent Release in FogBugz. You need to locate the ID of the equivalent Release in FogBugz, enter it into this text-box and click [Save]. You should now repeat for all the other pre-existing releases.

The FogBugz Release ID can be found by going to Settings > Projects and viewing the releases:

Releases: (this project)	Edit	Version	Assignable	Date
		1.0.0.0(1)	Yes	5/30/2009
		v2.0B	Yes	6/8/2009
		v2.0C	Yes	6/15/2009
		Create New Release		

A list of past and upcoming releases or versions specific to this project (for example "Beta", "Version 2.0", etc.)

Then hover the mouse over the release name. The release ID will be displayed in the URL line as ixFixFor=X where X is the numeric ID of the release.

6.2.4. Configuring the Standard Field Mapping

Now that the projects, user and releases have been mapped correctly, we need to configure the standard incident fields. To do this, go to Administration > System > Data Synchronization and click on the “View Project Mappings” for the FogBugzDataSync plug-in entry:

Incident	Component	Computer
	Priority	Version
	Severity	Area
	Status	Parent Case ID
	Type	

From this screen, you need to click on Priority, Status and Type in turn to configure their values:

a) Incident Type

Click on the “Type” hyperlink under Incident Standard Fields to bring up the Incident type mapping configuration screen:

Artifact Type: Incident
Field Name: Type

Type	Active	External Key	FogBugz	Primary
Bug	Yes	1	<input type="text"/>	<input checked="" type="checkbox"/>
Change Request	Yes		<input type="text"/>	<input checked="" type="checkbox"/>
Enhancement	Yes	2	<input type="text"/>	<input checked="" type="checkbox"/>
Incident	Yes	1	<input type="text"/>	<input checked="" type="checkbox"/>
Issue	Yes	3	<input type="text"/>	<input checked="" type="checkbox"/>
Limitation	Yes		<input type="text"/>	<input checked="" type="checkbox"/>
Risk	Yes		<input type="text"/>	<input checked="" type="checkbox"/>
Training	Yes		<input type="text"/>	<input checked="" type="checkbox"/>

The table lists each of the incident types available in SpiraTeam and provides you with the ability to enter the matching FogBugz case category ID for each one. You can map multiple SpiraTeam fields to the same FogBugz fields (e.g. Bug and Incident in SpiraTeam are both equivalent to Bug in FogBugz), in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from FogBugz > SpiraTeam).

The values for the category ID are fixed for FogBugz and should be:

Category Name	Category ID
Bug	1
Feature	2
Inquiry	3

So, depending on which types have been configured in SpiraTeam, you'll need to adjust the mapping so that the appropriate SpiraTeam types correspond to the equivalent FogBugz category.

b) Incident Status

Click on the "Status" hyperlink under Incident Standard Fields to bring up the Incident status mapping configuration screen:

Artifact Type: Incident
Field Name: Status

Status	Active	External Key	FogBugz	Primary
Assigned	Yes	1	<input type="text"/>	<input checked="" type="checkbox"/>
Closed	Yes	15	<input type="text"/>	<input checked="" type="checkbox"/>
Duplicate	Yes	4	<input type="text"/>	<input checked="" type="checkbox"/>
New	Yes	1	<input type="text"/>	<input checked="" type="checkbox"/>
Not Reproducible	Yes	3	<input type="text"/>	<input checked="" type="checkbox"/>
Open	Yes	1	<input type="text"/>	<input checked="" type="checkbox"/>
Reopen	No	1	<input type="text"/>	<input checked="" type="checkbox"/>
Resolved	Yes	2	<input type="text"/>	<input checked="" type="checkbox"/>

The table lists each of the incident statuses available in SpiraTeam and provides you with the ability to enter the matching FogBugz case status ID for each one. You can map multiple SpiraTeam fields to the

same FogBugz fields (e.g. New, Open, Assigned, and Reopen in SpiraTeam are all equivalent to Active in FogBugz), in which case only one of the four values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from FogBugz > SpiraTeam).

We recommend that you always point the New, Open, Assigned and Reopen statuses inside SpiraTeam to point to the ID for "Assigned" inside FogBugz and make Assigned in SpiraTeam the Primary status of the four. This is recommended so that as new incidents in SpiraTeam get synched over to FogBugz, they will get switched to the Active status in FogBugz which will then be synched back to "Assigned" in SpiraTeam. That way you'll be able to see at a glance which incidents have been synched with FogBugz and those that haven't.

You also might want to consider changing the statuses in SpiraTeam to match the 16 discrete statuses in FogBugz to make things easier for your users. In which case you'll need to create the new statuses and configure the workflow (as described in the *SpiraTeam Administration Guide*).

The status IDs in FogBugz are fixed and should be:

Status ID	Status Name
1	Active
2	Resolved (Fixed)
3	Resolved (Not Reproducible)
4	Resolved (Duplicate)
5	Resolved (Postponed)
6	Resolved (Won't Fix)
7	Resolved (By Design)
8	Resolved (Implemented)
9	Resolved (Won't Implement)
10	Resolved (Already Exists)
11	Resolved (Responded)
12	Resolved (Won't Respond)
13	Resolved (SPAM)
14	Resolved (Waiting For Info)
15	Resolved (Completed)
16	Resolved (Canceled)

In addition to these statuses, FogBugz also has the concept of a 'Closed' case which is one where the case has been assigned to the special Closed user (user id 1). If you want to map a SpiraTeam status to this special closed status, for the external key just enter 'Closed' instead of a numeric ID and that will tell the plug-in to associate that SpiraTest status with the special condition of a FogBugz case that is assigned to the 'closed' user.

c) Incident Priority

Click on the "Priority" hyperlink under Incident Standard Fields to bring up the Incident Priority mapping configuration screen:

Field Value	Active	External Key	Primary
1 - Must Fix	Yes	<input type="text" value="1"/>	Yes ▼
2 - Must Fix	Yes	<input type="text" value="2"/>	Yes ▼
3 - Must Fix	Yes	<input type="text" value="3"/>	Yes ▼
4 - Fix If Time	Yes	<input type="text" value="4"/>	Yes ▼
5 - Fix If Time	Yes	<input type="text" value="5"/>	Yes ▼
6 - Fix If Time	Yes	<input type="text" value="6"/>	Yes ▼
7 - Don't Fix	Yes	<input type="text" value="7"/>	Yes ▼

The table lists each of the incident priorities available in SpiraTeam and provides you with the ability to enter the matching FogBugz priority ID for each one. You can map multiple SpiraTeam fields to the same FogBugz fields, in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from FogBugz > SpiraTeam).

Since both applications allow you to customize the priority list, we recommend that you modify the list in both systems to be the same and then map them one to one as this will be easier for users to understand. In the example above, we have switched over SpiraTeam to match the priorities in FogBugz, but you could do it the other way around as well.

The FogBugz Priority IDs can be found by going to Settings > Priorities and viewing the priorities:

Priority

You can edit the text associated with each priority entry.

Priority Number	Priority Label	Default
1	<input type="text" value="Must Fix"/>	<input type="radio"/>
2	<input type="text" value="Must Fix"/>	<input type="radio"/>
3	<input type="text" value="Must Fix"/>	<input checked="" type="radio"/>
4	<input type="text" value="Fix If Time"/>	<input type="radio"/>
5	<input type="text" value="Fix If Time"/>	<input type="radio"/>
6	<input type="text" value="Fix If Time"/>	<input type="radio"/>
7	<input type="text" value="Don't Fix"/>	<input type="radio"/>

The priority ID is the "priority number" value displayed in the left hand column.

6.2.5. Configuring the Custom Property Mapping

Now that the various SpiraTeam standard fields have been mapped correctly, we need to configure the custom property mappings. This is used for custom properties in SpiraTeam that are used to map to standard fields in FogBugz (Computer, Version and Area) that don't exist in SpiraTeam.

From the View/Edit Project Data Mapping screen, you need to click on the name of the Incident Custom Property that you want to add data-mapping information for. We will consider the three different types of mapping that you typically will want to enter:

 Incident	Component	Computer
	Priority	Version
	Severity	Area
	Status	Parent Case ID
	Type	

a) FogBugz's Computer Field

You first need to create an incident custom property in SpiraTeam of type 'TEXT' that will be used to store the Computer description within SpiraTeam.

Then click on the hyperlink of this new text custom property under Incident Custom Properties to bring up the custom property mapping configuration screen:

Artifact Type: ★ Incident

Name: ★ Computer Text

External Key: ★

Update

All you need to do on this screen is enter the word "Computer" in the External Key textbox and the data-sync plug-in will know that this custom property is mapped to the built-in Computer field in FogBugz.

b) FogBugz's Version Field

You first need to create an incident custom property in SpiraTeam of type 'TEXT' that will be used to store the Version description within SpiraTeam.

Then click on the hyperlink of this new text custom property under Incident Custom Properties to bring up the custom property mapping configuration screen:

Artifact Type: ★ Incident

Name: ★ Version Text

External Key: ★

Update

All you need to do on this screen is enter the word “Version” in the External Key textbox and the data-sync plug-in will know that this custom property is mapped to the built-in Version field in FogBugz.

c) FogBugz’s Area Field

You first need to create an incident custom property in SpiraTeam of type ‘LIST’ that will be used to store the list of project areas within SpiraTeam. You will need to create a new custom list to store the different possible values of area and then use that list when creating the new custom property.

Then back on the Data Mapping page, click on the hyperlink of this new list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen:

Artifact Type: Incident

Name: Area List

External Key: Area

Custom Value Mapping

Value	Active	External Key FogBugz
Code	Yes	1
Documentation	Yes	3
Miscellaneous	Yes	4
User Interface	Yes	2

Update

First you need to enter the word “Area” as the External Key of the custom property. This tells the data-sync plug-in that the custom property in SpiraTeam should be mapped to built-in Area field in FogBugz.

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the FogBugz ID of the various Areas that are configured in FogBugz. The FogBugz Area ID can be found by going to Settings > Projects and viewing the areas in the project:

Edit	Delete	Area	Primary Contact
		Code	Project's Primary Contact
		Documentation	Project's Primary Contact
		Miscellaneous	Project's Primary Contact
		User Interface	Project's Primary Contact
		Create New Area	

Each project is divided into functional areas. You must have at least one area per project.

Then hover the mouse over the area name. The area ID will be displayed in the URL line as ixArea=X where X is the numeric ID of the area.

d) FogBugz’s Parent Case Field

FogBugz lets you link a new case with an existing ‘parent’ case. You can make this possible from within SpiraTeam by simply creating a new custom text property and mapping to the special **External Key = Parent:**

Artifact Type:*	Incident
Name:*	Parent Case ID Text
External Key:*	<input type="text" value="Parent"/>
Update	

Users will then enter the FogBugz ID of an existing case when they log a new SpiraTeam incident and the data-synchronization system will know how to associate the two cases.

Once you have updated the various mapping sections, you are now ready to use the synchronization.

6.3. Using SpiraTeam with FogBugz

Now that the integration service has been configured and the service started, initially any incidents created in SpiraTeam for the specified projects will be imported into FogBugz and any existing cases in FogBugz will get loaded into SpiraTeam. At this point we recommend opening the Windows Event Viewer and choosing the Application Log. In this log any error messages raised by the SpiraTeam Data Sync Service will be displayed. If you see any error messages at this point, we recommend immediately stopping the SpiraTeam service and checking the various mapping entries. If you cannot see any cases with the mapping information, we recommend sending a copy of the event log message(s) to Inflectra customer services (support@inflectra.com) who will help you troubleshoot the problem.

To use SpiraTeam with FogBugz on an ongoing basis, we recommend the following general processes be followed:

- When running tests in SpiraTest or SpiraTeam, defects found should be logged through the Test Execution Wizard as normal.
- Developers using FogBugz can log new defects into either SpiraTeam or FogBugz. In either case they will get loaded into the other system.
- Once created in one of the systems and successfully replicated to the other system, the incident should not be modified again inside SpiraTeam. Since FogBugz is considered the master system for incidents/cases, all data changes to the case should be made inside FogBugz. To enforce this, you should modify the workflows set up in SpiraTeam so that the various fields are marked as inactive for all the incident statuses other than the “New” status. This will allow someone to submit an incident in SpiraTeam, but will prevent them making changes in conflict with FogBugz after that point.
- As the case progresses through the FogBugz workflow, changes to the type of case, changes to its status, priority, description and resolution will be updated automatically in SpiraTeam. In essence, SpiraTeam acts as a read-only viewer of these incidents.
- You are now able to perform test coverage and incident reporting inside SpiraTest/SpiraTeam using the test cases managed by SpiraTest/SpiraTeam and the incidents managed on behalf of SpiraTest/SpiraTeam inside FogBugz.

7. Using SpiraTeam with Mantis

This section outlines how to use SpiraTest, SpiraPlan or SpiraTeam (hereafter referred to as SpiraTeam) in conjunction with the Mantis issue tracking system. The built-in integration service allows the quality assurance team to manage their requirements and test cases in SpiraTeam, execute test runs in SpiraTest, and then have the new incidents generated during the run be automatically loaded into Mantis.

Once the incidents are synchronized into Mantis, the development team can then manage the issues in Mantis and have the status changes and additional notes entered in Mantis be reflected back in SpiraTeam. In addition, any new issues logged into mantis will get imported into SpiraTeam so that they can be linked to test cases and requirements.

► **STOP! Please make sure you have first read the Instructions in Section 1 before proceeding!**

7.1. Configuring the Plug-In

The next step is to configure the plug-in within SpiraTeam so that the system knows how to access the Mantis server. To start the configuration, open up SpiraTeam in a web browser, log in using a valid account that has System-Administration level privileges and click on the System > Data Synchronization administration option from the left-hand navigation:

Data Synchronization ([Refresh](#))

SpiraTeam is able to communicate with a variety of external systems and applications using its background data synchronization service and library of available plug-ins.

This page allows you to view, add and modify the list of plug-ins, make changes to their configuration, and verify the status of the most recent synchronization. In addition, you can force a particular plug-in to perform a complete data-synchronization by resetting its date of last sync:

Plug-In	Data Mapping	Last Sync Date	Status	Operations
MantisDataSync	> View Project Mappings	-	Not Run	> Force Re-Sync Edit Delete

[► Add](#)

This screen lists all the plug-ins already configured in the system. Depending on whether you chose the option to include sample data in your installation or not, you will see either an empty screen or a list of sample data-synchronization plug-ins.

If you already see an entry for **MantisDataSync** you should click on its “Edit” link. If you don’t see such an entry in the list, please click on the [Add] button instead. In either case you will be taken to the following screen where you can enter or modify the Mantis Data-Synchronization plug-in:

Edit Plug-In | MantisDataSync
[<< Back to Data Synchronization Home](#)

Please enter/edit the following information for the plug-in. The exact information that needs to be entered is specific to each plug-in, and you should refer to the documentation for the plug-in when entering/changing it.

Name*:

Description:

Connection Info*:

Login*:

Password:

Time Offset*:

Auto-Map Users

Custom 01:

Custom 02:

Custom 03:

Custom 04:

Custom 05:

You need to fill out the following fields for the Mantis Plug-in to operate correctly:

- **Name** – this needs to be set to **MantisDataSync**. This needs to match the name of the plug-in DLL assembly that was copied into the C:\Program Files\SpiraTeam\Bin folder (minus the .dll file extension). If you renamed the MantisDataSync.dll file for any reason, then you need to change the name here to match.
- **Description** – this should be set to a description of the plug-in. This is an optional field that is used for documentation purposes and is not actually used by the system.
- **Connection Info** – this should be the URL that you use to access your instance of Mantis (e.g. https://www.mycompany.com/bugs)
- **Login** – this should be set to a valid login to the Mantis installation. The login needs to have permissions to create and view issues and versions within Mantis for the projects that you will be syncing to SpiraTeam.
- **Password** – this should be set to the password of the login specified above.
- **Time Offset** – The time offset between the two servers, if the Mantis server is on a different server than SpiraTeam. For example, if the Mantis server's clock is set to Pacific Standard Time (PST) and the SpiraTeam server is set to Eastern Standard Time (EST), the Mantis server would be three hours behind SpiraTeam, so you would need to put **-3** into this field.
- **Auto-Map Users** – If enabled and a mapped user is not found between the two systems, a search will be made comparing logins between SpiraTeam and Mantis for matching UserIDs. If one is found, then that user will be used. If not enabled and a match is not found, then the UserID used will be the connecting user for the Data Sync. (The SpiraTeam User for issues coming into SpiraTeam, and the Mantis Login for issues imported into Mantis.)
- **Custom 01** – This field specifies whether or not a Resolution item in SpiraTeam, or a Note item in Mantis will be created when an issue is created in either system for a new issue. Valid values are **True** or **False**. Default (or blank) is **True**.
- **Custom 02** – This field indicates whether or not to convert Carriage Returns and spaces in Mantis issues when synchronizing them into SpiraTeam. If enabled, then carriage returns will be converted to HTML breaks, and multiple spaces will be converted to non-breaking spaces to preserve formatting when importing into SpiraTeam. If disabled, then carriage returns and spaces will be left as-is. Valid values are **True** or **False**. Default (or blank) is **True**.

- **Custom 03** – This field is only used when ‘Auto-Map Users’ is enabled and for Incidents synchronized from SpiraTeam into Mantis. If enabled, and the Auto-Map User did not find a user with a matching Login ID, then the Login ID will be set to the User in Spira, even if that user may not exist in Mantis. Depending on Mantis configuration, the user may be accepted, or it may default back to the Mantis UserID that the Data Sync runs under. Valid values are **True** or **False**. Default (or blank) is **False**.
- **Custom 04** – If enabled, this option specifies whether or not to append the “Additional Information” and “Steps To Reproduce” fields to the end of the Description field in Spira. During transfer of new issues from Mantis to SpiraTeam, the Description field in SpiraTeam will consist of the Description field in Mantis appended by the Additional Information field in Mantis, and finally the Steps To Reproduce field in Mantis. If this option is disabled, only the Description will be transferred over. Valid values are **True** or **False**. Default (or blank) is **False**.
- **Custom 05** – This is not currently used by the MantisDataSync, and can be left blank.

7.2. Configuring the Data Mapping

Next, you need to configure the data mapping between SpiraTeam and Mantis. This allows the various projects, users, releases, incident types, statuses, priorities and custom property values used in the two applications to be related to each other. This is important, as without a correct mapping, there is no way for the integration service to know that an “Enhancement” in SpiraTeam is the same as a “Feature” in Mantis (for example).

The following mapping information needs to be setup in SpiraTeam:

- The linking between the project in SpiraTeam and the project in Mantis.
- The linking of users between the two systems.
- The linking of releases between the two systems.
- The linking of standard SpiraTeam fields to Mantis fields.
- The linking of custom SpiraTeam fields to Mantis custom fields.

Each of these is explained in turn below:

7.2.1. Configuring the Project Mapping

While working in the project you want to map, from the data synchronization administration page you need to click on the “View Project Mappings” hyperlink next to the Mantis plug-in name. This will take you to the data-mapping overview page:

View/Edit Project Data Mapping | Sample Application Two [\(Change Project\)](#)
[<< Back to Data Synchronization Home](#)

To configure this project for data mapping with the **MantisDataSync** plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name: Sample Application Two

External Key:*

Active:* ▼

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Incident	Priority Severity Status Type	
Release		
Task	Priority Status	

If the project name does not match the name of the project you want to configure the data-mapping for, click on the “(Change Project)” hyperlink to change the current project.

To enable this project for data-synchronization with Mantis, you need to enter:

- **External Key** – This should be set to the ID of the project in Mantis. To get the ID of the Project in Mantis, log in as an administrator and go to Manage -> Manage Projects:

Projects <input type="button" value="Create New Project"/>		
Name ▲	Status	Enabled
Library System	development	X
Sample Application 2	development	X

Then hover the mouse over the project name. The project ID will be displayed in the URL line as **project_id=X** where **X** is the numeric ID of the project.

- **Active Flag** – Set this to ‘Yes’ so that SpiraTeam knows that you want to synchronize data for this project. Once the project has been completed, setting the value to “No” will stop data synchronization, reducing network utilization.

Click [Update] to confirm these settings. Once you have enabled the project for data-synchronization, you can now enter the other data mapping values outlined below.

Note: Once you have successfully configured the project, when creating a new project, you should choose the option to “Create Project from Existing Project” rather than “Use Default Template” so that all the project mappings get copied across to the new project, if you are going to want to Sync the new project up to Mantis.

7.2.2. Configuring the User Mapping

To configure the mapping of users in the two systems, you need to go to Administration > Users > View Edit Users, which will bring up the list of users in the system. Then click on the “Edit” button for a particular user that will be editing issues in Mantis:

Add/Edit User | Fox Michaels

Please review the information listed below and make any changes if necessary. Once you have finished, click [Update/Insert] to confirm them.

First Name*:

Middle Initial:

Last Name*:

User Name*:

Email Address*:

System Administrator*: Yes ▼

Active User*: Yes ▼

MantisDataSync ID:

LDAP Managed User

LDAP DN:

Change Password

New Password:

Confirm Password:

You will notice that below the Active flag for the user is a list of all the configured data-synchronization plug-ins. In the text box next to the MantisDataSync ID you need to enter the Login ID of this user in Mantis. If you have the “Automap Users” checkbox enabled in the MantisDataSync plugin, then if no link is created, the system will scan for a matching Login ID from both systems and use a match. (If you then do not have Custom3 set to “False”, then for data going into Mantis the User ID will be forced to that of the User ID in SpiraTeam.)

Once you have entered the Mantis Login ID in, click [Update]. You should now repeat for the other users who will be active in both systems.

7.2.3. Configuring the Release Mapping

When the data-synchronization service runs and it comes across a release in SpiraTeam (or a Version in Mantis) that it has not linked before, it will create a corresponding entry in the other system. When starting out a new project, it is recommended that you let the MantisDataSync handle creation of the releases/versions in either system, and then edit the information once the link is made.

In cases where you are syncing up two existing projects in both systems, it is advised that you link any existing releases that exist in both systems manually, and then only create new releases in one system. To link a release in SpiraTeam up to a version in Mantis, please navigate to Planning > Releases and click on the Release/Iteration in question. Make sure you have the ‘Overview’ tab visible and expand the “Details” section of the release/iteration:

Release: 3.0.0.1 - New Release [RL:000023]

Name: Beta 1

Overview Incidents Reqs & Tasks Test Cases Test Runs Attachments History

▼ Details

Version #:	1.0.2.8	Creator*:	System Administrator
Start/End Date:	8/7/2013 - 9/7/2013	Creation Date:	6/5/2013 6:04:46 PM
# Resources:	1	Active*:	Yes
Non-Working:	0 person days	Iteration*:	No
Plan Effort:	184.00	Available Effort:	184.00
MantisDataSync ID:	Beta1 - 1.0.2.8		

In addition to the standard fields and custom properties configured for Releases, you will see an additional text property called “**MantisDataSync ID**” that is used to store the mapped external identifier for the equivalent Release in Mantis. The Mantis ID of a version is the string that is in the

The Mantis Release ID can be found by going to **Manage -> Manage Projects -> Versions** and viewing a release’s details:

Edit Project Version

Version	Beta1 - 1.0.2.8
Date Order	2009-12-10 13:34:24
Description	
Released	<input checked="" type="checkbox"/>

Update Version

The Mantis Release ID is the highlighted text field. Copy and paste this into the field in SpiraTeam. Depending on your regional settings in both applications, this field will likely be case-sensitive.

For versions imported into Mantis from SpiraTeam, the Version will have an “(S)” appended to the name, and for versions in SpiraTeam imported from Mantis the version field of the Release will have “(M)” appended to the name.

7.2.4. Configuring the Standard Field Mapping

Now that the projects, user and releases have been mapped correctly, we need to configure the standard incident fields. To do this, go to Administration > System > Data Synchronization and click on the “View Project Mappings” for the MantisDataSync plug-in entry:

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Incident	Priority Severity Status Type	Operating System
Release		
Task	Priority Status	

From this screen, you need to set up the Priority, Severity, Status, and Type fields:

a) Incident Type

The Incident Type field is optional and can be linked to the Mantis Category selection.

If you do not link values, then all issues being imported into SpiraTeam from Mantis will be set to the Default Type (as specified in the “View/Edit Types” screen), and issues going from SpiraTeam into Mantis will be assigned to the first Category in the list. (Usually Mantis orders them alphabetically, but this may change depending on your installation. If you do not have any Categories set-up, then issues will not transfer over and error messages will be logged.) For existing issues, updates to this field will not be transferred.

Field Value	Active	External Key	Primary
Bug	Yes	<input type="text" value="development"/>	<input type="button" value="Yes"/> ▼
Change Request	Yes	<input type="text"/>	<input type="button" value="Yes"/> ▼
Enhancement	Yes	<input type="text"/>	<input type="button" value="Yes"/> ▼
Incident	Yes	<input type="text" value="security"/>	<input type="button" value="Yes"/> ▼
Issue	Yes	<input type="text"/>	<input type="button" value="Yes"/> ▼
Limitation	Yes	<input type="text"/>	<input type="button" value="Yes"/> ▼
New Value	Yes	<input type="text"/>	<input type="button" value="Yes"/> ▼
Risk	Yes	<input type="text"/>	<input type="button" value="Yes"/> ▼
Training	Yes	<input type="text" value="documentation"/>	<input type="button" value="Yes"/> ▼

The table lists each of the incident types available in SpiraTeam and provides you with the ability to enter the matching Mantis Category for each one. The value to put in External ID is the Category text:

Edit Project Category

Category

Assigned To

The Mantis Release ID is the highlighted text field. Copy and paste this into the field in SpiraTeam. Depending on your regional settings in both applications, this field will likely be case-sensitive.

You can map multiple SpiraTeam fields to the same Mantis fields (e.g. Bug and Incident in SpiraTeam are both equivalent to category “development” in Mantis). In a situation like this, enter in the Mantis category in both Bug and Incident external keys, and decide which one will be primary. For issues coming from Mantis into SpiraTeam, the one marked Primary will be used, and for issues being created in Mantis, the same category will be used to create the issue.

b) Incident Status

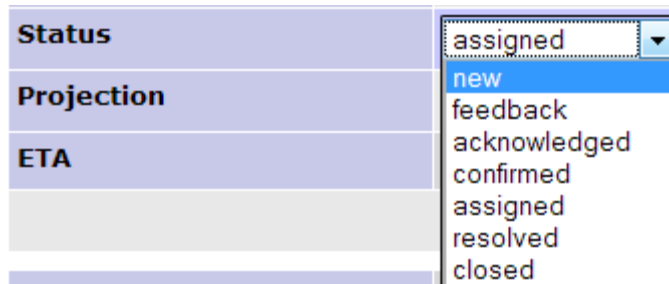
The Incident Status is an optional field to be linked to the Mantis field by the same name.

If you do not link values, then defaults will be used. For issues coming from Mantis into SpiraTeam, incidents will be marked as ‘New’ (as defined by the “View/Edit Status” in Administration), and for issues being transferred to Mantis, the default is ‘new’. Note that if an issue has an Owner in SpiraTeam, then the default for the new issue in Mantis is ‘assigned’. For existing issues, updates to the field will not be transferred over.

Field Value	Active	External Key	Primary
Assigned	Yes	assigned	Yes ▼
Closed	Yes	closed	Yes ▼
Duplicate	Yes		Yes ▼
New	Yes	new	Yes ▼
Not Reproducible	Yes	feedback	Yes ▼
Open	Yes	acknowledged	Yes ▼
Reopen	Yes		Yes ▼
Resolved	Yes		Yes ▼

[Update](#)

The table lists each of the incident types available in SpiraTeam and provides you with the ability to enter the matching Mantis Category for each one. The values to put in External Key is any one of the Status values in Mantis. By default in Mantis, the available statuses are:



The Mantis values are in the highlighted text field. Type these into the External Key field in SpiraTeam. Depending on your regional settings in both applications, this field will likely be case-sensitive.

You can map multiple SpiraTeam fields to the same Mantis fields, just like the Incident Type above.

c) Incident Priority & Severity

The Incident Priority and Severity are optional fields that are linked to Mantis fields by the same name.

If you do not link values, then defaults will be used. For issues coming from Mantis into SpiraTeam, incidents will leave those fields undefined (unset). For issues coming from SpiraTeam into Mantis, the

default priority of 'normal' and severity of 'minor' is used. For existing issues, updates to the field will not be transferred over.

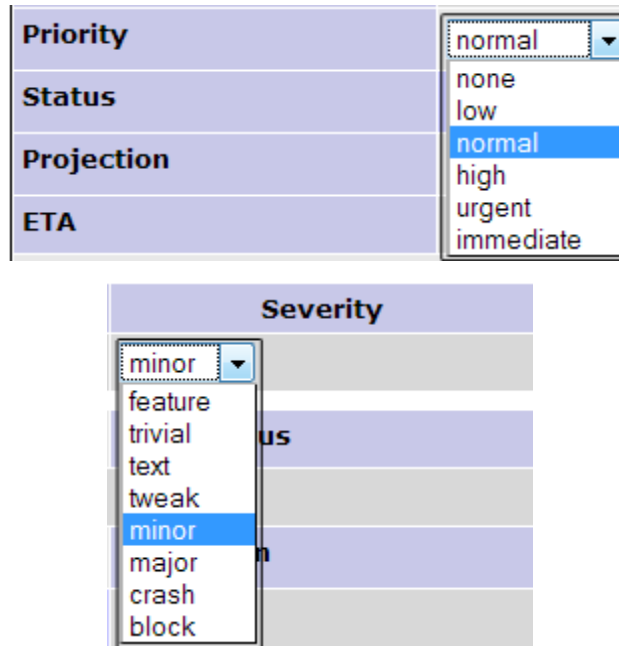
Field Value	Active	External Key	Primary
1 - Critical	Yes	immediate	Yes ▼
2 - High	Yes	high	Yes ▼
3 - Medium	Yes	normal	Yes ▼
4 - Low	Yes	none	Yes ▼

▶ Update

Field Value	Active	External Key	Primary
1 - Critical	Yes	crash	Yes ▼
2 - High	Yes	major	Yes ▼
3 - Medium	Yes	minor	Yes ▼
4 - Low	Yes	trivial	Yes ▼

▶ Update

The table lists each of the priorities available in SpiraTeam and provides you with the ability to enter the matching Mantis priority for each one. (The table for Severities has the same functionality.) The values to put in External Key are any one of the Priority (or Severity) values in Mantis. By default in Mantis, the available values are:



The Mantis values are in the highlighted fields above. Type these into the External Key field in SpiraTeam. Depending on your regional settings in both applications, this field will likely be case-sensitive.

You can map multiple SpiraTeam fields to the same Mantis fields, just like described in Incident Type above.

7.2.5. Configuring the Custom Property Mapping

Now that the various SpiraTeam standard fields have been mapped correctly, we need to configure the custom property mappings. At the moment, only custom fields in Mantis can be linked to custom fields in SpiraTeam.

From the View/Edit Project Data Mapping screen, you need to click on the name of the Incident Custom Property that you want to add data-mapping information for. Both field types in SpiraTeam can be linked up to any of the supported field types in Mantis. Linking between the two systems is done in text values only – that means that if you have a SpiraTeam custom list, then the values that will be put into Mantis will be the strings of the list. The same works for moving fields back from Mantis. Rules for linking different field types up are as follows:

SpiraTeam ‘List’ to Mantis ‘Enum’, ‘List’, or ‘Multiselection’: For linking these types of fields together, the available values must match. For example, if you have “Windows” as an item in your list in SpiraTeam, then in the associated field in Mantis, “Windows” must be an available item as well. In instances where there is no match, then the default will be used in either system. On a Multiselection-type field, for importing back into SpiraTeam, only the first (top) selected value will be stored.

SpiraTeam ‘List’ to Mantis ‘Numeric’, ‘Float’, ‘Date’, ‘Text’, or ‘Email’: In this case, the text value of the SpiraTeam list will be assigned to the Mantis field, and values must be exact. For example, if you linked a SpiraTeam List to a Mantis Date field, the value for the List must be a valid date, like “1/1/2010”. If any value fails the Mantis validation, the value will be ignored and the custom field will be set blank or to default. When transferring a value back from Mantis into SpiraTeam, the text must equal an available item in the custom list, or the field will be left blank.

SpiraTeam ‘Text’ to Mantis ‘Numeric’, ‘Float’, ‘Date’, ‘Text’, or ‘Email’: In this case, text will be copied over as-is. Note that in some special cases, like the number, date, and e-mail fields, Mantis may apply formatting or verification on values transferred over.

SpiraTeam ‘Text’ to Mantis ‘Enum’, ‘List’, or ‘Multiselection’: When pulling data from Mantis, the SpiraTeam custom field will be translated as the field in Mantis displays. However, when transferring data to Mantis, if the text in the SpiraTeam field does not match an available item in the lists, then Mantis may leave the field blank or set it to the default value.

a) Mapping custom fields

For a SpiraTeam test field, all you need to do is link the custom field to the custom field in Mantis. To do this, click on the name of the custom field under the “Custom Properties” header in the MantisDataSync Project Mappings, and you will see a screen allowing you to enter the External Key:

[insert screenshot of custom map text prop screen with mapping for below]

In the External Key field, put the name of your custom field in Mantis:

Edit custom field	
Name	OS Platform
Type	List
Possible Values	Windows Linux Mac
Default Value	Windows

Once you have updated the various mapping sections, you are now ready to use the service.

7.3. Using SpiraTeam with Mantis

Now that the integration service has been configured and the service started, initially any incidents created in SpiraTeam for the specified projects will be imported into Mantis and any existing issues in Mantis will get loaded into SpiraTeam. After the first sync, we recommend opening the Windows Event Viewer and viewing the Application Log. Any errors (unable to connect messages, invalid required field mappings) and warnings (incomplete field mappings) will be displayed. If on Server 2008/Vista or later, you can filter by the Application name "MantisDataSync". If you see any error messages (or warning messages that you want to correct before continuing) at this point, we recommend immediately stopping the SpiraTeam service and checking the various mapping entries. If you cannot see any issues with the mapping information, we recommend sending a copy of the event log message(s) to Inflectra customer services (support@inflectra.com) who will help you troubleshoot the problem.

To use SpiraTeam with Mantis on an ongoing basis, we recommend the following general processes be followed:

- When running tests in SpiraTest or SpiraTeam, defects found should be logged through the Test Execution Wizard as normal.
- Developers using Mantis can log new defects into either SpiraTeam or Mantis. In either case they will get loaded into the other system.
- Once created in one of the systems and successfully replicated to the other system, the incident should not be modified again inside SpiraTeam. Since Mantis is considered the **master system** for incidents/issues, all data changes to the issue should be made inside Mantis. To enforce this, you should modify the workflows set up in SpiraTeam so that the various fields are marked as inactive for all the incident statuses other than the "New" status. This will allow someone to submit an incident in SpiraTeam, but will prevent them from making changes in conflict with Mantis after that point.
- As the issue progresses in Mantis, changes to the type of issue, changes to its status, priority, description and resolution will be updated automatically in SpiraTeam. In essence, SpiraTeam acts as a read-only viewer of these incidents.
- You are now able to perform test coverage and incident reporting inside SpiraTeam using the test cases managed by SpiraTeam and the incidents managed on behalf of SpiraTeam inside Mantis.

8. Using SpiraTeam with ClearQuest

This section outlines how to use SpiraTest, SpiraPlan or SpiraTeam (hereafter referred to as SpiraTeam) in conjunction with the ClearQuest defect tracking system. The built-in integration service allows the quality assurance team to manage their requirements and test cases in SpiraTeam, execute test runs in SpiraTest, and then have the new incidents generated during the run be automatically loaded into ClearQuest. Once the incidents are loaded into ClearQuest as defects, the development team can then manage the lifecycle of these defects in ClearQuest, and have the status changes in ClearQuest be reflected back in SpiraTeam. In addition, any issues logged directly into ClearQuest will get imported into SpiraTeam so that they can be linked to test cases and requirements.

► **Note: The ClearQuest Plug-In is Not Available on the Inflectra Cloud-Based DataSync Service.**

8.1. Configuring the Integration Service

This section outlines how to configure the integration service to export incidents into ClearQuest and pick up subsequent status changes in ClearQuest and have them be updated in SpiraTeam. It assumes that you already have a working installation of SpiraTest, SpiraPlan or SpiraTeam (v3.0 or higher) and a working installation of IBM Rational ClearQuest 7.0 or higher.

If you have an earlier version of SpiraTeam, you will need to upgrade to at least v3.0 before trying to integrate with ClearQuest.

The steps that need to be performed to configure integration with ClearQuest are as follows:

- Download the latest ClearQuest Data-Sync plug-in for SpiraTeam from our website
- Setup the plug-in in SpiraTeam to point to the correct instance of ClearQuest
- Configure the data field mappings between SpiraTeam and ClearQuest
- Start the service and verify data transfer

8.1.1. Download the ClearQuest Plug-In

Go to the Inflectra website and open up the page that lists the various downloads available for SpiraTeam (<http://www.inflectra.com/SpiraTeam/Downloads.aspx>). Listed on this page will be the ClearQuest Plug-In for SpiraTeam. Right-click on this link and save the Zip compressed folder to the hard-drive of the server where SpiraTeam is installed.

Open up the compressed folder and extract the ClearQuestDataSync.dll file and place it in the C:\Program Files\SpiraTeam\Bin folder (it may be SpiraTest or SpiraPlan depending on which product you're running). This folder should already contain the DataSyncService.exe and DataSyncService.exe.config files that are the primary files used for managing the data synchronization between SpiraTeam and other systems.

You will then need to install the ClearQuest client application itself onto the SpiraTeam server. This is needed because the ClearQuest plugin communicates with the ClearQuest API which is part of the ClearQuest client installation. The SpiraTeam plugin will use a single ClearQuest user license when it connects to ClearQuest.

If you do not have an on-premise installation of SpiraTeam, but instead are using a hosted subscription provided by Inflectra (or a third party company) you will not have access to the DataSyncService background service. In such situations, you should use the Desktop DataSync application instead. This application is described in Appendix 1 and can be used instead of the server-based DataSyncService.

8.1.2. Configuring the Service

To configure the integration service, please open up the DataSyncService.exe.config file located in C:\Program Files\SpiraTeam\Bin with a text editor such as Notepad. Once open, it should look like:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <sectionGroup name="applicationSettings" type="System.Configuration.ApplicationSettingsGroup,
System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" >
      <section name="Inflectra.SpiraTest.DataSyncService.Properties.Settings"
type="System.Configuration.ClientSettingsSection, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" requirePermission="false" />
    </sectionGroup>
  </configSections>
  <applicationSettings>
    <Inflectra.SpiraTest.DataSyncService.Properties.Settings>
      <setting name="PollingInterval" serializeAs="String">
        <value>600000</value>
      </setting>
      <setting name="WebServiceUrl" serializeAs="String">
        <value>http://localhost/SpiraTeam</value>
      </setting>
      <setting name="Login" serializeAs="String">
        <value>fredbloggs</value>
      </setting>
      <setting name="Password" serializeAs="String">
        <value>fredbloggs</value>
      </setting>
      <setting name="EventLogSource" serializeAs="String">
        <value>SpiraTeam Data Sync Service</value>
      </setting>
      <setting name="TraceLogging" serializeAs="String">
        <value>False</value>
      </setting>
    </Inflectra.SpiraTest.DataSyncService.Properties.Settings>
  </applicationSettings>
</configuration>
```

The sections that need to be verified and possibly changed are marked in yellow above. You need to check the following information:

- The polling interval allows you to specify how frequently the data-synchronization service will ask SpiraTeam and the external system for new data updates. The value is specified in milliseconds and we recommend a value no smaller than 5 minutes (i.e. 300,000ms). The larger the number, the longer it will take for data to be synchronized, but the lower the network and server overhead.
- The base URL to your instance SpiraTeam. It is typically of the form http://<server name>/SpiraTeam. Make sure that when you enter this URL on a browser on the server itself, the application login page appears.
- A valid login name and password to your instance of SpiraTeam. This user needs to be a member of the project(s) that will be synchronized with ClearQuest and needs to have at least **Incident create/modify/view** permissions and **Release create/modify/view** permissions in these projects.

Once you have made these changes, save the file and proceed to the next stage.

8.1.3. Configuring the Plug-In

The next step is to configure the plug-in within SpiraTeam so that the system knows how to access the ClearQuest server. To start the configuration, please open up SpiraTeam in a web browser, log in using a

valid account that has System-Administration level privileges and click on the System > Data Synchronization administration option from the left-hand navigation:

Data Synchronization ([Refresh](#))

SpiraTeam is able to communicate with a variety of external systems and applications using its background data synchronization service and library of available plug-ins.

This page allows you to view, add and modify the list of plug-ins, make changes to their configuration, and verify the status of the most recent synchronization. In addition, you can force a particular plug-in to perform a complete data-synchronization by resetting its date of last sync:

Plug-In	Data Mapping	Last Sync Date	Status	Operations
ClearQuestDataSync	> View Project Mappings	9/13/2011 4:02:19 PM	Success	> Force Re-Sync Edit Delete

[Add](#)

This screen lists all the plug-ins already configured in the system. Depending on whether you chose the option to include sample data in your installation or not, you will see either an empty screen or a list of sample data-synchronization plug-ins.

If you already see an entry for **ClearQuestDataSync** you should click on its “Edit” link. If you don’t see such an entry in the list, please click on the [Add] button instead. In either case you will be taken to the following screen where you can enter or modify the ClearQuest Data-Synchronization plug-in:

Edit Plug-In | ClearQuestDataSync

[<< Back to Data Synchronization Home](#)

Please enter/edit the following information for the plug-in. The exact information that needs to be entered is specific to each plug-in, and you should refer to the documentation for the plug-in when entering/changing it:

Name*:

Description:

Connection Info*:

Login*:

Password:

Time Offset*:

Auto-Map Users

You need to fill out the following fields for the ClearQuest Plug-in to operate correctly:

- **Name** – this needs to be set to **ClearQuestDataSync**. This needs to match the name of the plug-in DLL assembly that was copied into the C:\Program Files\SpiraTeam\Bin folder (minus the .dll file extension). If you renamed the ClearQuestDataSync.dll file for any reason, then you need to change the name here to match.
- **Description** – this should be set to a description of the plug-in. This is an optional field that is used for documentation purposes and is not actually used by the system.
- **Connection Info** – this should be the name of the ClearQuest *master database*. In most installations this is simply called “MASTR”.
- **Login** – this should be set to a valid login for your ClearQuest installation. The login needs to have permissions to create and view defects within ClearQuest.
- **Password** – this should be set to the password of the login specified above.
- **Time Offset** – normally this should be set to zero, but if you find that issues being changed in ClearQuest are not being updated in SpiraTeam, try increasing the value as this will tell the data-synchronization plug-in to add on the time offset (in hours) when comparing date-time stamps.

Also if your ClearQuest installation is running on a server set to a different time-zone, then you should add in the number of hours difference between the servers' time-zones here.

- **Auto-Map Users** – this is not currently used and can be ignored.
- **Custom 01** – This should be set to the word “True” if you want to have the plugin restrict synchronization to only loading new incidents from SpiraTeam > ClearQuest and updating existing items. This is useful if you want to prevent existing issues in ClearQuest from being loaded into SpiraTeam. Leave blank if you want the plugin to synchronize normally.
- **Custom 02** – Sometimes you don't want all the incidents in SpiraTeam to be added to ClearQuest. You can optionally enter a filter definition in this box to restrict the incidents that get synchronized. The filter uses the following syntax:

```
[Property]=[Value|*]:[Property]=[Value|*]
```

For example, to limit the incidents to only have those where List01 = 5 and Text08 = “Hello” and Text05 is not blank you would use:

```
List01=5:Text08=Hello:Text05=*
```

- **Custom 03** – ClearQuest doesn't have a built-in Detected in Release field. If you would like to map a custom ClearQuest field to the SpiraTeam Detected in Release, simply enter in the name of the ClearQuest field here.
- **Custom 04** – ClearQuest doesn't have a built-in Resolved in Release field. If you would like to map a custom ClearQuest field to the SpiraTeam Resolved in Release, simply enter in the name of the ClearQuest field here.
- **Custom 05** – This is the optional “DBset” value, when you have installations with more than one database set. If you have a single database set you can just leave this blank.

If you enter a field name in either Custom 03 or Custom 04, you will need to also map the various releases in SpiraTeam to their corresponding equivalent field value in ClearQuest. To do that, click on Planning > Releases and choose a specific release. Then in the “ClearQuest DataSync ID” field under the “Custom Properties” tab you need to enter the name of the equivalent ClearQuest release.

8.2. Configuring the Data Mapping

Next, you need to configure the data mapping between SpiraTeam and ClearQuest. This allows the various projects, users, incident statuses, priorities, severities and custom property values used in the two applications to be related to each other. This is important, as without a correct mapping, there is no way for the integration service to know that a “New” item in SpiraTeam is equivalent to a “Submitted” item in ClearQuest (for example).

The following mapping information needs to be setup in SpiraTeam:

- The mapping of the project identifiers for the projects that need to be synchronized
- The mapping of users in the system
- The mapping of the various standard fields in the system
- The mapping of the various custom properties in the system

Each of these is explained in turn below:

8.2.1. Configuring the Project Mapping

From the data synchronization administration page, you need to click on the “View Project Mappings” hyperlink next to the ClearQuest plug-in name. This will take you to the data-mapping home page for the currently selected project:

ClearQuestDataSync Project Data Mapping | Sample Application Two [\(Change Project\)](#)

[<< Back to Data Synchronization Home](#)

To configure this project for data mapping with the 'ClearQuestDataSync' plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name: Sample Application Two

External Key:*

Active:* ▼

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Incident	Priority Severity Status Type	Product Resolution
Release		
Task	Priority Status	

If the project name does not match the name of the project you want to configure the data-mapping for, click on the “(Change Project)” hyperlink to change the current project.

To enable this project for data-synchronization with ClearQuest, you need to enter:

- **External Key** – This should be set to the name of the project database in ClearQuest that will be mapped to the specific SpiraTeam project.
- **Active Flag** – Set this to ‘Yes’ so that SpiraTeam knows that you want to synchronize data for this project. Once the project has been completed, setting the value to “No” will stop data synchronization, reducing network utilization.

Click [Update] to confirm these settings. Once you have enabled the project for data-synchronization, you can now enter the other data mapping values outlined below.

Note: *Once you have successfully configured the project, when creating a new project, you should choose the option to “Create Project from Existing Project” rather than “Use Default Template” so that all the project mappings get copied across to the new project.*

8.2.2. Configuring the User Mapping

To configure the mapping of users in the two systems, you need to go to Administration > Users > View Edit Users, which will bring up the list of users in the system. Then click on the “Edit” button for a particular user that will be editing issues in ClearQuest:

Add/Edit User | Joe P Smith

Please review the information listed below and make any changes if necessary. Once you have finished, click [Update/Insert] to confirm them.

First Name*:

Middle Initial:

Last Name*:

User Name*:

Department:

Email Address*:

Emails Enabled: ▾

System Administrator*: ▾

Active User*: ▾

RSS Token: Enable RSS Feeds

Project Membership Passwords & Security **Data Mapping**

ClearQuestDataSync ID:

You will notice that in the Data Mapping tab for the user, there is a list of all the configured data-synchronization plug-ins. In the text box next to the ClearQuest Data-Sync plug-in you need to enter the login for this username in ClearQuest. This will allow the data-synchronization plug-in to know which user in SpiraTeam match which equivalent user in ClearQuest. Click [Update] once you've entered the appropriate login name. You should now repeat for the other users who will be active in both systems.

8.2.3. Configuring the Standard Field Mapping

Now that the projects, user and releases have been mapped correctly, we need to configure the standard incident fields. To do this, go to Administration > System > Data Synchronization and click on the "View Project Mappings" for the ClearQuestDataSync plug-in entry:

ClearQuestDataSync Project Data Mapping | Sample Application Two [\(Change Project\)](#)
[<< Back to Data Synchronization Home](#)

To configure this project for data mapping with the 'ClearQuestDataSync' plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name: Sample Application Two
External Key:*
Active:*

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Incident	Priority Severity Status Type	Product Resolution
Release		
Task	Priority Status	

From this screen, you need to click on Priority, Severity and Status in turn to configure their values:

a) Incident Status

Click on the "Status" hyperlink under Incident Standard Fields to bring up the Incident status mapping configuration screen:

Edit Standard Field Data Mapping | Sample Application Two [\(Change Project\)](#)
[<< Back to Project Mappings](#)

To configure the data mappings for this field, please fill in the external key values for the ClearQuestDataSync plug-in next to each SpiraTeam value:

Field Value	Active	External Key	Primary
Assigned	Yes	<input type="text" value="Assigned"/>	<input type="button" value="Yes"/>
Closed	Yes	<input type="text" value="Closed"/>	<input type="button" value="Yes"/>
Duplicate	Yes	<input type="text" value="Duplicate"/>	<input type="button" value="Yes"/>
New	Yes	<input type="text" value="Submitted"/>	<input type="button" value="Yes"/>
Not Reproducible	Yes	<input type="text"/>	<input type="button" value="Yes"/>
Open	Yes	<input type="text" value="Opened"/>	<input type="button" value="Yes"/>
Reopen	No	<input type="text"/>	<input type="button" value="Yes"/>
Resolved	Yes	<input type="text" value="Resolved"/>	<input type="button" value="Yes"/>

The table lists each of the incident statuses available in SpiraTeam and provides you with the ability to enter the matching ClearQuest issue status name for each one. You can map multiple SpiraTeam fields to the same ClearQuest fields (e.g. Open and Reopen in SpiraTeam are both equivalent to Opened in ClearQuest), in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from ClearQuest > SpiraTeam).

b) Incident Priority

Click on the “Priority” hyperlink under Incident Standard Fields to bring up the Incident Priority mapping configuration screen:

Edit Standard Field Data Mapping | Sample Application Two [\(Change Project\)](#)
[<< Back to Project Mappings](#)

To configure the data mappings for this field, please fill in the external key values for the **ClearQuestDataSync** plug-in next to each SpiraTeam value:

Field Value	Active	External Key	Primary
1 - Critical	Yes	<input type="text" value="1-Resolve Immediately"/>	Yes ▼
2 - High	Yes	<input type="text" value="2-Give High Attention"/>	Yes ▼
3 - Medium	Yes	<input type="text" value="3-Normal Queue"/>	Yes ▼
4 - Low	Yes	<input type="text" value="4-Low Priority"/>	Yes ▼

[► Update](#)

The table lists each of the incident priorities available in SpiraTeam and provides you with the ability to enter the matching ClearQuest priority name for each one. You can map multiple SpiraTeam fields to the same ClearQuest fields, in which case only one of the two values can be listed as Primary = Yes as that’s the value that’s used on the reverse synchronization (from ClearQuest > SpiraTeam).

c) Incident Severity

Click on the “Severity” hyperlink under Incident Standard Fields to bring up the Incident severity mapping configuration screen:

Edit Standard Field Data Mapping | Sample Application Two [\(Change Project\)](#)
[<< Back to Project Mappings](#)

To configure the data mappings for this field, please fill in the external key values for the **ClearQuestDataSync** plug-in next to each SpiraTeam value:

Field Value	Active	External Key	Primary
1 - Critical	Yes	<input type="text" value="1-Critical"/>	Yes ▼
2 - High	Yes	<input type="text" value="2-Major"/>	Yes ▼
3 - Medium	Yes	<input type="text" value="3-Average"/>	Yes ▼
4 - Low	Yes	<input type="text" value="4-Minor"/>	Yes ▼

[► Update](#)

The table lists each of the incident severities available in SpiraTeam and provides you with the ability to enter the matching ClearQuest severity name for each one. You can map multiple SpiraTeam fields to the same ClearQuest fields, in which case only one of the two values can be listed as Primary = Yes as that’s the value that’s used on the reverse synchronization (from ClearQuest > SpiraTeam).

8.2.4. Configuring the Custom Property Mapping

Now that the various SpiraTeam standard fields have been mapped correctly, we need to configure the custom property mappings. This is used for both custom properties in SpiraTeam that map to custom fields in ClearQuest and also for custom properties in SpiraTeam that are used to map to standard fields in ClearQuest (e.g. Project, Resolution) that don't exist in SpiraTeam.

From the View/Edit Project Data Mapping screen, you need to click on the name of the Incident Custom Property that you want to add data-mapping information for. We will consider the two different types of mapping that you might want to enter:

ClearQuestDataSync Project Data Mapping | Sample Application Two [\(Change Project\)](#)

[<< Back to Data Synchronization Home](#)

To configure this project for data mapping with the 'ClearQuestDataSync' plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name: Sample Application Two

External Key:*

Active:* ▼

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Incident	Priority Severity Status Type	Product Resolution
Release		
Task	Priority Status	

a) Text Custom Properties

Click on the hyperlink of the text custom property under Incident Custom Properties to bring up the custom property mapping configuration screen. For text custom properties there will be no values listed in the lower half of the screen.

Edit Custom Property Data Mapping | Library Information System [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the JiraDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Name: Notes

External Key:

Custom Value Mapping

Property Value	Active	External Key
----------------	--------	--------------

You need to lookup the ID of the custom field in ClearQuest that matches this custom property in SpiraTeam. Once you have entered the id of the custom field, click [Update].

Note: The ID can be found by looking at the URL inside ClearQuest when choosing to View/Edit the Custom Field. The URL will include the section: **id=X** where X is the numeric ID of the Custom Field inside ClearQuest.

b) List Custom Properties

Click on the hyperlink of the list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen. For list custom properties there will be a textbox for both the custom field itself and a mapping table for each of the custom property values that need to be mapped:

Edit Custom Property Data Mapping | Sample Application Two [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the ClearQuestDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

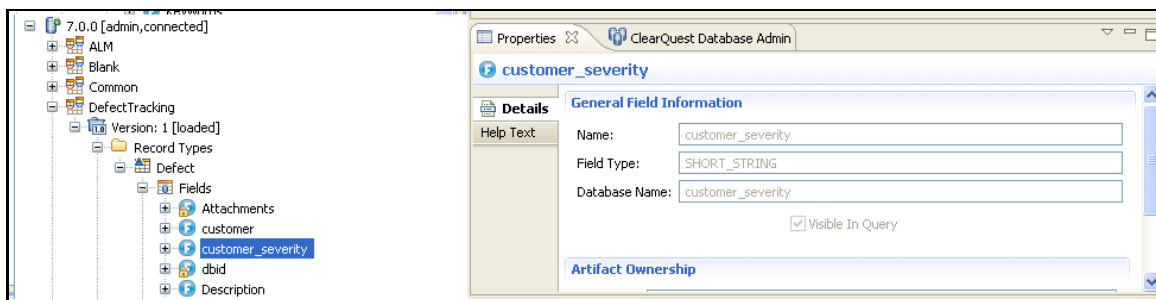
Name: Resolution

External Key:

Custom Value Mapping

Property Value	Active	External Key
Duplicate	Yes	<input type="text" value="Duplicate"/>
Enhancement Request	Yes	<input type="text" value="Enhancement Request"/>
Fixed	Yes	<input type="text" value="Fixed"/>
Fixed Indirectly	Yes	<input type="text" value="Fixed Indirectly"/>
Functions as Designed	Yes	<input type="text" value="Functions as Designed"/>

First you need to lookup the name of the field in ClearQuest that matches this custom property in SpiraTeam. This should be entered in the 'External Key' field below the name of the custom property. The easiest way to determine this is to use the *ClearQuest Designer* which lets you see all the fields associated with a ClearQuest defect:



Next for each of the Property Values in the table (in the lower half of the page) you need to enter the full name of the possible field values as displayed in the ClearQuest client application.

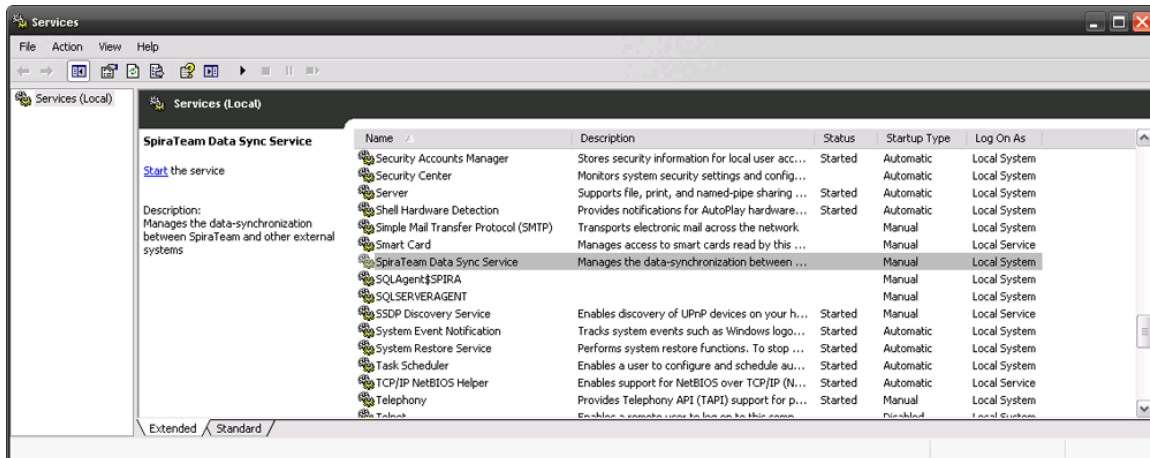
Once you have updated the various mapping sections, you are now ready to start the service.

8.3. Enabling the Data-Synchronization

8.3.1. Starting the Service

When SpiraTeam is installed, a Windows Service – SpiraTeam Data Sync Service – is installed along with the web application. However to avoid wasting system resources, this service is initially set to run manually. To ensure continued synchronization of SpiraTeam with ClearQuest, we recommend starting the service and setting its startup-type to Automatic.

To make these changes, open up the Windows Control Panel, click on the “Administrative Tools” link, and then choose the Services option. This will bring up the Windows Service control panel:



Click on the ‘SpiraTeam Data Sync Service’ entry and click on the link to start the service. Then right-click the service entry and choose the option to set the startup type to ‘Automatic’. This will ensure that synchronization continues between SpiraTeam and ClearQuest after a reboot of the server.

8.3.2. Using SpiraTeam with ClearQuest

Now that the integration service has been configured and the service started, initially any incidents created in SpiraTeam for the specified projects will be imported into ClearQuest and any existing defects in ClearQuest will get loaded into SpiraTeam

At this point we recommend opening the Windows Event Viewer and choosing the Application Log. In this log any error messages raised by the SpiraTeam Data Sync Service will be displayed. If you see any error messages at this point, we recommend immediately stopping the SpiraTeam service and checking the various mapping entries. If you cannot see any issues with the mapping information, we recommend sending a copy of the event log message(s) to Inflectra customer services (support@inflectra.com) who will help you troubleshoot the problem.

To use SpiraTeam with ClearQuest on an ongoing basis, we recommend the following general processes be followed:

- When running tests in SpiraTest or SpiraTeam, defects found should be logged through the Test Execution Wizard as normal.
- Developers using ClearQuest can log new defects into either SpiraTeam or ClearQuest. In either case they will get loaded into the other system.
- Once created in one of the systems and successfully replicated to the other system, the incident should not be modified again inside SpiraTeam
- At this point, the incident should not be acted upon inside SpiraTeam and all data changes to the issue should be made inside ClearQuest. To enforce this, you should modify the workflows set up

in SpiraTeam so that the various fields are marked as inactive for all the incident statuses other than the “New” status. This will allow someone to submit an incident in SpiraTeam, but will prevent them making changes in conflict with ClearQuest after that point.

- As the issue progresses through the customized ClearQuest workflow, changes to the type of issue, changes to its status, priority, description and resolution will be updated automatically in SpiraTeam. In essence, SpiraTeam acts as a read-only viewer of these incidents.
- You are now able to perform test coverage and incident reporting inside SpiraTest/SpiraTeam using the test cases managed by SpiraTest/SpiraTeam and the incidents managed on behalf of SpiraTest/SpiraTeam inside ClearQuest.

9. Using SpiraTeam with IBM RTC

This section outlines how to use SpiraTest, SpiraPlan or SpiraTeam (hereafter referred to as SpiraTeam) in conjunction with the IBM Rational Team Concert (hereafter referred to as RTC) work item tracking system. The built-in integration service allows the quality assurance team to manage their requirements and test cases in SpiraTeam, execute test runs in SpiraTest, and then have the new incidents generated during the run be automatically loaded into RTC.

Once the incidents are loaded into RTC as work items, the development team can then manage the lifecycle of these work items in RTC, and have the status changes in RTC be reflected back in SpiraTeam. In addition, any issues logged directly into RTC will get imported into SpiraTeam so that they can be linked to test cases and requirements.

► **STOP! Please make sure you have first read the Instructions in Section 1 before proceeding!**

9.1. Configuring the Plug-In

The next step is to configure the plug-in within SpiraTeam so that the system knows how to access the RTC server. To start the configuration, please open up SpiraTeam in a web browser, log in using a valid account that has System-Administration level privileges and click on the System > Data Synchronization administration option from the left-hand navigation:

Data Synchronization [\(Refresh\)](#)

SpiraTeam is able to communicate with a variety of external systems and applications using its background data synchronization service and library of available plug-ins.

This page allows you to view, add and modify the list of plug-ins, make changes to their configuration, and verify the status of the most recent synchronization. In addition, you can force a particular plug-in to perform a complete data-synchronization by resetting its date of last sync:

Plug-In	Data Mapping	Last Sync Date	Status	Operations
BugzillaDataSync	> View Project Mappings	-	Failure	> Force Re-Sync Edit Delete
FogBugzDataSync	> View Project Mappings	-	Failure	> Force Re-Sync Edit Delete
JiraDataSync	> View Project Mappings	-	Failure	> Force Re-Sync Edit Delete
MSTfsDataSync	> View Project Mappings	-	Failure	> Force Re-Sync Edit Delete
RtcDataSync	> View Project Mappings	5/31/2012 1:35:37 PM	Success	> Force Re-Sync Edit Delete

[► Add](#)

This screen lists all the plug-ins already configured in the system. Depending on whether you chose the option to include sample data in your installation or not, you will see either an empty screen or a list of sample data-synchronization plug-ins.

If you already see an entry for **RtcDataSync** you should click on its “Edit” link. If you don’t see such an entry in the list, please click on the [Add] button instead. In either case you will be taken to the following screen where you can enter or modify the RTC Data-Synchronization plug-in:

Edit Plug-In | RtcDataSync

[<< Back to Data Synchronization Home](#)

Please enter/edit the following information for the plug-in. The exact information that needs to be entered is specific to each plug-in, and you should refer to the documentation for the plug-in when entering/changing it:

Name*:

Description:

Connection Info*:

Login*:

Password:

Time Offset*:

Auto-Map Users

Custom 01:

Custom 02:

Custom 03:

Custom 04:

Custom 05:

You need to fill out the following fields for the RTC Plug-in to operate correctly:

- **Name** – this needs to be set to **RtcDataSync**. This needs to match the name of the plug-in DLL assembly that was copied into the C:\Program Files\SpiraTeam\Bin folder (minus the .dll file extension). If you renamed the RtcDataSync.dll file for any reason, then you need to change the name here to match.
- **Description** – this should be set to a description of the plug-in. This is an optional field that is used for documentation purposes and is not actually used by the system.
- **Connection Info** – this should be the base URL for connecting to your instance of RTC (for example <https://servername:9443/ccm>).
- **Login** – this should be set to a valid login for your RTC installation. The login needs to have permissions to create and view work items within RTC.
- **Password** – this should be set to the password of the login specified above.
- **Time Offset** – normally this should be set to zero, but if you find that issues being changed in RTC are not being updated in SpiraTeam, try increasing the value as this will tell the data-synchronization plug-in to add on the time offset (in hours) when comparing date-time stamps. Also if your RTC installation is running on a server set to a different time-zone, then you should add in the number of hours difference between the servers' time-zones here.
- **Auto-Map Users** – this is not currently used and can be ignored.
- **Custom 01** – this is not currently used and can be ignored
- **Custom 02** – this is not currently used and can be ignored
- **Custom 03** – this is not currently used and can be ignored
- **Custom 04** – this is not currently used and can be ignored
- **Custom 05** – this is not currently used and can be ignored

9.2. Configuring the Data Mapping

Next, you need to configure the data mapping between SpiraTeam and RTC. This allows the various projects, users, incident statuses, priorities, severities and custom property values used in the two applications to be related to each other. This is important, as without a correct mapping, there is no way for the integration service to know that a “New” item in SpiraTeam is equivalent to a “New” item in RTC (for example).

The following mapping information needs to be setup in SpiraTeam:

- The mapping of the project identifiers for the projects that need to be synchronized
- The mapping of the various standard fields in the system
- The mapping of the various custom properties in the system

Each of these is explained in turn below:

9.2.1. Configuring the Project Mapping

From the data synchronization administration page, you need to click on the “View Project Mappings” hyperlink next to the RTC plug-in name. This will take you to the data-mapping home page for the currently selected project:

RtcDataSync Project Data Mapping | Sample Application Two [\(Change Project\)](#)

[<< Back to Data Synchronization Home](#)

To configure this project for data mapping with the **RtcDataSync** plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name: Sample Application Two

External Key:*

Active:*

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Incident	Priority Severity Status Type	
Release		
Task	Priority Status	

If the project name does not match the name of the project you want to configure the data-mapping for, click on the “(Change Project)” hyperlink to change the current project.

To enable this project for data-synchronization with RTC, you need to enter:

- **External Key** – This should be set to the display name of the project in RTC that will be mapped to the specific SpiraTeam project.
- **Active Flag** – Set this to ‘Yes’ so that SpiraTeam knows that you want to synchronize data for this project. Once the project has been completed, setting the value to “No” will stop data synchronization, reducing network utilization.

Click [Update] to confirm these settings. Once you have enabled the project for data-synchronization, you can now enter the other data mapping values outlined below.

Note: Once you have successfully configured the project, when creating a new project, you should choose the option to “Create Project from Existing Project” rather than “Use Default Template” so that all the project mappings get copied across to the new project.

9.2.2. Configuring the Standard Field Mapping

Now that the projects, user and releases have been mapped correctly, we need to configure the standard incident fields. To do this, go to Administration > System > Data Synchronization and click on the “View Project Mappings” for the RtcDataSync plug-in entry:

RtcDataSync Project Data Mapping | Sample Application Two [\(Change Project\)](#)

[<< Back to Data Synchronization Home](#)

To configure this project for data mapping with the **RtcDataSync** plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name: Sample Application Two

External Key:*

Active:* ▼

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Incident	Priority Severity Status Type	
Release		
Task	Priority Status	

From this screen, you need to click on Status and Type in turn to configure their values:

a) Incident Status

Click on the “Status” hyperlink under Incident Standard Fields to bring up the Incident status mapping configuration screen:

Edit Standard Field Data Mapping | Sample Application Two [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this field, please fill in the external key values for the **RtcDataSync** plug-in next to each SpiraTeam value:

Field Value	Active	External Key	Primary
Assigned	Yes	<input type="text"/>	Yes ▼
Closed	Yes	<input type="text" value="Complete"/>	Yes ▼
Duplicate	Yes	<input type="text"/>	Yes ▼
New	Yes	<input type="text" value="New"/>	Yes ▼
Not Reproducible	Yes	<input type="text" value="Invalidate"/>	Yes ▼
Open	Yes	<input type="text" value="Start Working"/>	Yes ▼
Reopen	No	<input type="text"/>	Yes ▼
Resolved	Yes	<input type="text" value="Complete"/>	No ▼

The table lists each of the incident statuses available in SpiraTeam and provides you with the ability to enter the matching RTC work item status name for each one. You can map multiple SpiraTeam fields to the same RTC fields (e.g. Closed and Resolved in SpiraTeam are both equivalent to Complete in RTC), in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from RTC > SpiraTeam).

b) Incident Type

Click on the "Type" hyperlink under Incident Standard Fields to bring up the Incident type mapping configuration screen:

Edit Standard Field Data Mapping | Sample Application Two [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this field, please fill in the external key values for the **RtcDataSync** plug-in next to each SpiraTeam value:

Field Value	Active	External Key	Primary
Bug	Yes	<input type="text" value="Task"/>	Yes ▼
Change Request	Yes	<input type="text" value="Task"/>	No ▼
Enhancement	Yes	<input type="text" value="Task"/>	No ▼
Incident	Yes	<input type="text" value="Task"/>	No ▼
Issue	Yes	<input type="text" value="Task"/>	No ▼
Limitation	Yes	<input type="text" value="Task"/>	No ▼
Risk	Yes	<input type="text" value="Task"/>	No ▼
Training	Yes	<input type="text" value="Task"/>	No ▼

[Update](#)

The table lists each of the incident types available in SpiraTeam and provides you with the ability to enter the matching RTC work item type name for each one. You can map multiple SpiraTeam fields to the same RTC fields, in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from RTC > SpiraTeam).

9.2.3. Configuring the Custom Property Mapping

Now that the various SpiraTeam standard fields have been mapped correctly, we need to configure the custom property mappings. This is used to associate custom properties in SpiraTeam that map to custom fields in RTC. From the View/Edit Project Data Mapping screen, you need to click on the name of the Incident Custom Property that you want to add data-mapping information for.

We will consider the two different types of mapping that you might want to enter:

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Incident	Priority Severity Status Type	Approved? Archived
Release		
Task	Priority Status	

a) Text Custom Properties

Click on the hyperlink of the text custom property under Incident Custom Properties to bring up the custom property mapping configuration screen. For text custom properties there will be no values listed in the lower half of the screen.

Edit Custom Property Data Mapping | Sample Application Two [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the RtcDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Name: Approved?

External Key:

Custom Value Mapping

Property Value	Active	External Key
----------------	--------	--------------

You need to obtain the fully qualified XML name of the custom field in RTC that matches this custom property in SpiraTeam from the RTC documentation. Once you have entered the name of the custom field, click [Update].

b) List Custom Properties

Click on the hyperlink of the list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen. For list custom properties there will be a textbox for both the custom field itself and a mapping table for each of the custom property values that need to be mapped:

Edit Custom Property Data Mapping | Sample Application Two ([Change Project](#))

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the RtcDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Name: Archived

External Key:

Custom Value Mapping

Property Value	Active	External Key
False	Yes	<input type="text" value="false"/>
True	Yes	<input type="text" value="true"/>

First you need to obtain the fully qualified XML name of the field in RTC that matches this custom property in SpiraTeam. This should be entered in the 'External Key' field below the name of the custom property. Then you need enter the possible values in RTC for the custom property, mapping each one to the corresponding SpiraTeam custom property value.

Once you have updated the various mapping sections, you are now ready to use the service.

9.3. Using SpiraTeam with RTC

Now that the integration service has been configured and the service started, initially any incidents created in SpiraTeam for the specified projects will be imported into RTC and any existing work items in RTC will get loaded into SpiraTeam

At this point we recommend opening the Windows Event Viewer and choosing the Application Log. In this log any error messages raised by the SpiraTeam Data Sync Service will be displayed. If you see any error messages at this point, we recommend immediately stopping the SpiraTeam service and checking the various mapping entries. If you cannot see any issues with the mapping information, we recommend sending a copy of the event log message(s) to Inflectra customer services (<http://www.inflectra.com/Support>) who will help you troubleshoot the problem.

To use SpiraTeam with RTC on an ongoing basis, we recommend the following general processes be followed:

- When running tests in SpiraTest or SpiraTeam, defects found should be logged through the Test Execution Wizard as normal.
- Developers using RTC can log new work items into either SpiraTeam or RTC. In either case they will get loaded into the other system.
- Once created in one of the systems and successfully replicated to the other system, the incident should not be modified again inside SpiraTeam
- At this point, the incident should not be acted upon inside SpiraTeam and all data changes to the issue should be made inside RTC. To enforce this, you should modify the workflows set up in SpiraTeam so that the various fields are marked as inactive for all the incident statuses other than the "New" status. This will allow someone to submit an incident in SpiraTeam, but will prevent them making changes in conflict with RTC after that point.

- As the issue progresses through the customized RTC workflow, changes to the type of work item, changes to its status, description and custom fields will be updated automatically in SpiraTeam. In essence, SpiraTeam acts as a read-only viewer of these incidents.

You are now able to perform test coverage and incident reporting inside SpiraTest/SpiraTeam using the test cases managed by SpiraTest/SpiraTeam and the incidents managed on behalf of SpiraTest/SpiraTeam inside RTC.

10. Using Spira with Axosoft 14+

This section outlines how to use SpiraTest, SpiraPlan or SpiraTeam (hereafter referred to as SpiraTeam) in conjunction with the Axosoft defect tracking system (formerly known as OnTime). The built-in integration service allows the quality assurance team to manage their requirements and test cases in SpiraTeam, execute test runs in SpiraTest, and then have the new incidents generated during the run be automatically loaded into Axosoft.

Once the incidents are loaded into Axosoft as defects, the development team can then manage the lifecycle of these defects in Axosoft, and have the status changes in Axosoft be reflected back in SpiraTeam.

▶ **STOP! Please make sure you have first read the Instructions in Section 1 before proceeding!**

10.1. Configuring the Plug-In

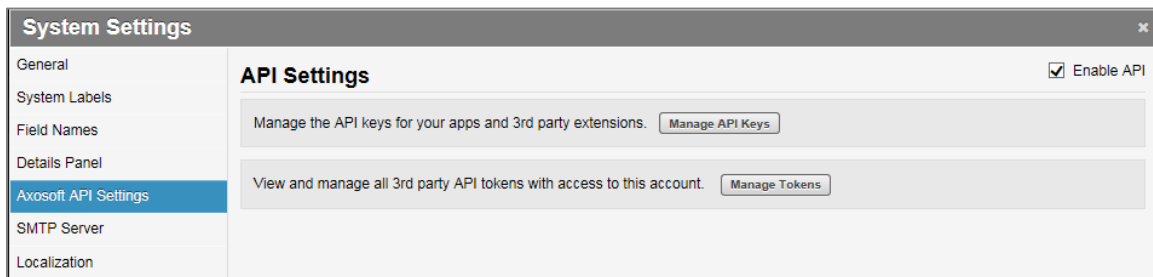
This section outlines how to configure the integration service to export incidents into Axosoft and pick up subsequent status changes in Axosoft and have them update SpiraTeam. It assumes that you already have a working installation of SpiraTest, SpiraPlan or SpiraTeam v4.0 or later and a working installation of Axosoft 14 or later (either hosted in the cloud or on-premise). If you have an earlier version of SpiraTeam, you will need to upgrade to at least v4.0 before trying to integrate with Axosoft.

The steps that need to be performed to configure integration with Axosoft are as follows:

- ▶ Enable the REST API in Axosoft
- ▶ Setup the plug-in in SpiraTeam to point to the correct instance of Axosoft
- ▶ Configure the data field mappings between SpiraTeam and Axosoft
- ▶ Start synchronization and verify data transfer

10.1.1. Enable the REST API in Axosoft

First you will need to login to your instance of Axosoft and click on Tools > System Options. Then click on the 'Axosoft API Settings' section:



Check the box to 'Enable API' and then click on the [Manage API Keys] button:

On this screen you will need to enter the name of the application you are creating an API key for (e.g. “Spira”) and then record the following two pieces of information:

- Client ID
- Client Secret

You will need these later on. Then click Save.

The Axosoft Client Secret is a long hash that will be of the form:

ykk8WD3eYfMJ6WbV1HtkutJv_w9jS2ah1tSbwqs-408Gp0_cPh5wTnjwfqPLN3-oCSHPVG5tpFkETHBgxUBKbXaTzzVqYtKC9_S

10.1.2. Configuring the Plug-In

The next step is to configure the plug-in within SpiraTeam so that the system knows how to access the Axosoft server. To start the configuration, please open up SpiraTeam in a web browser, log in using a valid account that has System-Administration level privileges and click on the System > Data Synchronization administration option from the left-hand navigation:

Plug-In	Data Mapping	Last Sync Date	Status	Operations
Axosoft On Time OnTimeDataSync	View Project Mappings	2/28/2018 2:21:47 PM	Success	Reset Sync Edit Delete View Errors

This screen lists all the plug-ins already configured in the system. Depending on whether you chose the option to include sample data in your installation or not, you will see either an empty screen or a list of sample data-synchronization plug-ins.

If you already see an entry for **AxosoftDataSync** you should click on its “Edit” link. If you don’t see such an entry in the list, please click on the [Add] button instead. In either case you will be taken to the following screen where you can enter or modify the Axosoft Data-Synchronization plug-in:

Axosoft On Time Edit Plug-In

[← Back to Data Synchronization Home](#)

Please enter/edit the following information for the plug-in. The exact information that needs to be entered is specific to each plug-in, and you should refer to the documentation for the plug-in when entering/changing it:

Name*	<input type="text" value="OnTimeDataSync"/>
	The physical name of the plugin (e.g. JiraDataSync) that matches the DLL being used
Caption:	<input type="text" value="Axosoft On Time"/>
	The display name of the plugin (e.g. Internal Jira) that can be used to distinguish between multiple instances of the same plugin type
Description:	<input type="text"/>
	The long description of the plugin (e.g. 'Connects to the in-house bug-tracker')
Connection Info*:	<input type="text" value="https://mysite.axosoft.com"/>
	The URL or other connection string used to connect to the external bug-tracker
Login*:	<input type="text" value="myuser@blah.com"/>
Password:	<input type="password" value="*****"/>
	The login and password used to connect to Axosoft On Time

Time Offset*:	<input type="text" value="0"/> hours
	The number of hours to look back for changes to synchronization. Normally should be set to zero (0) unless you find changes are not being synchronized
Auto-Map Users	<input type="text" value="No"/>
Custom 01:	<input type="text" value="a99d704d-77b0-4bf5-8418-986fa86934ae"/>
Custom 02:	<input type="text" value="1-4bf5-8418-986fa86934aea99d704d-77b0-4bf5-8418-986fa86934aea99d704d-77b0-4bf5-8418-986fa86934ae"/>
Custom 03:	<input type="text"/>
Custom 04:	<input type="text"/>
Custom 05:	<input type="text"/>
	<input type="button" value="Save"/> <input type="button" value="Cancel"/>

You need to fill out the following fields for the Axosoft Plug-in to operate correctly:

- **Name** – this needs to be set to **AxosoftDataSync**. This needs to match the name of the plug-in DLL assembly that was copied into the C:\Program Files\SpiraTeam\Bin folder (minus the .dll file extension). If you renamed the AxosoftDataSync.dll file for any reason, then you need to change the name here to match.
- **Description** – this should be set to a description of the plug-in. This is an optional field that is used for documentation purposes and is not actually used by the system.
- **Connection Info** – this should be the full URL to Axosoft. This is typically something like: <https://mysite.axosoft.com>.
- **Login** – this should be set to the login that you use to access Axosoft through its web interface

- **Password** – this should be set to the password that you use to access Axosoft through its web interface
- **Time Offset** – normally this should be set to zero, but if you find that defects being changed in Axosoft are not being updated in SpiraTeam, try increasing the value as this will tell the data-synchronization plug-in to add on the time offset (in hours) when comparing date-time stamps.
- **Auto-Map Users** – This changes the way that the plugin maps users in SpiraTeam to those in Axosoft:
 - ▷ **Auto-Map = True**
With this setting, all users in SpiraTeam need to have the same username as those in Axosoft. If this is the case then you do not need to perform the user-mapping task outlined in section 10.2.2. This is a big time-saver if you can guarantee that all usernames are the same in both systems.
 - ▷ **Auto-Map = False**
With this setting, users in SpiraTeam and Axosoft are free to have different usernames because you specify the corresponding Axosoft login for each user as outlined in section 10.2.2.
- **Custom 01** – This should contain the **Client ID** value from the Axosoft API Key screen
- **Custom 02** – This should contain the Axosoft **Client Secret** that you obtained from the Axosoft API Key Screen.
- **Custom 03-05** – these are not currently used by the Axosoft data-sync plug-in and can be left blank.

10.2. Configuring the Data Mapping

Next, you need to configure the data mapping between SpiraTeam and Axosoft. This allows the various projects, users, releases, incident statuses, priorities, severities and custom property values used in the two applications to be related to each other. This is important, as without a correct mapping, there is no way for the integration service to know that an “Open” incident in SpiraTeam is the same as an “Open” defect in Axosoft (for example).

The following mapping information needs to be setup in SpiraTeam:

- The mapping of the project identifiers for the projects that need to be synchronized
- The mapping of users in the system
- The mapping of releases in the system
- The mapping of the various standard fields in the system
- The mapping of the various custom properties in the system

Each of these is explained in turn below:

10.2.1. Configuring the Project Mapping

From the data synchronization administration page, you need to click on the “View Project Mappings” hyperlink next to the Axosoft plug-in name. This will take you to the data-mapping home page for the currently selected project:

OnTimeDataSync Project Data Mapping | OnTime Sample Project [\(Change Project\)](#)

[<< Back to Data Synchronization Home](#)

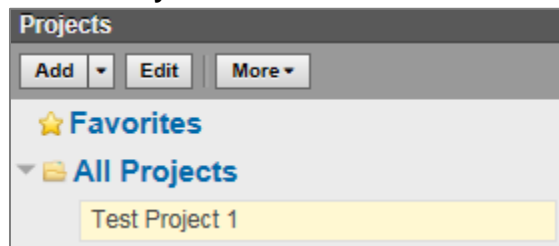
To configure this project for data mapping with the 'OnTimeDataSync' plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name:* OnTime Sample Project
External Key:*
Active:*

If the project name does not match the name of the project you want to configure the data-mapping for, click on the “(Change Project)” hyperlink to change the current project.

To enable this project for data-synchronization with Axosoft, you need to enter:

- **External Key** – This should be set to the **name** of the project token in Axosoft:



If you have a sub-project, you need to include both the parent and sub-project names separated by a forward slash (/), e.g. MyProject/SubProject1.

- **Active Flag** – Set this to 'Yes' so that SpiraTeam knows that you want to synchronize data for this project. Once the project has been completed, setting the value to “No” will stop data synchronization, reducing network utilization.

Click [Update] to confirm these settings. Once you have enabled the project for data-synchronization, you can now enter the other data mapping values outlined below.

Note: *Once you have successfully configured the project, when creating a new project, you should choose the option to “Create Project from Existing Project” rather than “Use Default Template” so that all the project mappings get copied across to the new project.*

10.2.2. Configuring the User Mapping

(This section can be skipped if you enabled the 'AutoMap Users' option earlier).

To configure the mapping of users in the two systems, you need to go to Administration > Users > View Edit Users, which will bring up the list of users in the system. Then click on the “Edit” button for a particular user that will be editing defects in Axosoft:

Add / Edit User | System Administrator

Please review the information listed below and make any changes if necessary. Once you have finished, click [Update/Insert] to confirm them.

First Name*:

Middle Initial:

Last Name*:

Project Membership Passwords & Security **Data Mapping** TaraVault Membership

OnTimeDataSync ID:

You will notice that in the **Data Mapping** tab for the user is a list of all the configured data-synchronization plug-ins. In the text box next to the Axosoft Data-Sync plug-in you need to enter the **Login Name** for this username in Axosoft. This will allow the data-synchronization plug-in to know which user in SpiraTeam match which equivalent user in Axosoft. Click [Update] once you've entered the appropriate login name. You should now repeat for the other users who will be active in both systems.

10.2.3. Configuring the Release Mapping

When the data-synchronization service runs, when it comes across a release/iteration in SpiraTeam that it has not seen before, it will create a corresponding Release in Axosoft. Similarly if it comes across a new Release in Axosoft that it has not seen before, it will create a new Release in SpiraTeam. Therefore when using both systems together, it is recommended that you only enter new Releases in one system and let the data-synchronization service add them to the other system.

To see this mapping, inside SpiraTeam, navigate to Planning > Releases and click on the Release/Iteration in question. Make sure you have the 'Overview' tab visible and expand the "Details" section of the release/iteration:

Release: 3.0.0.1 - New Release [RL.000024]

Name:

Overview Incidents Reqs & Tasks Test Cases Test Runs Attachments History

▼ Details

Version #: Creator*:

Start/End Date: - Creation Date: 6/5/2013 6:07:56 PM

Resources: Active*:

Non-Working: person days Iteration*:

Plan Effort: 176.00 Available Effort: 176.00

OnTimeDataSync ID:

In addition to the standard fields and custom properties configured for Releases, you will see an additional text property called "**AxosoftDataSync ID**" that is used to store the mapped external identifier for the equivalent Version in Axosoft.

10.2.4. Configuring the Standard Field Mapping

Now that the projects, user and releases have been mapped correctly, we need to configure the standard incident fields. To do this, go to Administration > System > Data Synchronization and click on the "View Project Mappings" for the AxosoftDataSync plug-in entry:

Incident	Component
	Priority
	Severity
	Status
	Type

From this screen, you need to click on Priority, Severity and Status in turn to configure their values (Axosoft doesn't support different defect types):

a) Incident Status

Click on the "Status" hyperlink under Incident Standard Fields to bring up the Incident status mapping configuration screen:

Artifact Type: Incident			
Field Name: Status			
Status	Active	External Key	Primary
Assigned	Yes	<input type="text" value="Assigned"/>	Yes ▼
Closed	Yes	<input type="text" value="Closed"/>	Yes ▼
Duplicate	Yes	<input type="text"/>	Yes ▼
New	Yes	<input type="text" value="Open"/>	No ▼
Not Reproducible	Yes	<input type="text" value="On Hold"/>	Yes ▼
Open	Yes	<input type="text" value="Open"/>	Yes ▼
Reopen	Yes	<input type="text" value="In Progress"/>	Yes ▼
Resolved	Yes	<input type="text" value="Waiting"/>	Yes ▼

[▶ Update](#)

The table lists each of the incident statuses available in SpiraTeam and provides you with the ability to enter the matching Axosoft defect status names for each one. You can map multiple SpiraTeam fields to the same Axosoft fields (e.g. New and Open in SpiraTeam are both equivalent to Open in Axosoft), in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from Axosoft > SpiraTeam).

We recommend that you always point the New and Open statuses inside SpiraTeam to point to the "Open" status inside Axosoft and make Open in SpiraTeam the Primary status of the two. This is recommended so that as new incidents in SpiraTeam get synched over to Axosoft, they will get switched to the Open status in Axosoft which will then be synched back to "Open" in SpiraTeam. That way you'll be able to see at a glance which incidents have been synched with Axosoft and those that haven't.

Note: The Axosoft external key needs to exactly match the display name of the status inside Axosoft. If you change the name of a status in Axosoft, you'll need to update the value in the data-mapping configuration as well.

b) Incident Priority

Click on the "Priority" hyperlink under Incident Standard Fields to bring up the Incident Priority mapping configuration screen:

Artifact Type: Incident
Field Name: Priority

Priority	Active	External Key	Primary
1 - Critical	Yes	<input type="text" value="High"/>	No ▼
2 - High	Yes	<input type="text" value="High"/>	Yes ▼
3 - Medium	Yes	<input type="text" value="Medium"/>	Yes ▼
4 - Low	Yes	<input type="text" value="Low"/>	Yes ▼

Update

The table lists each of the incident priorities available in SpiraTeam and provides you with the ability to enter the matching Axosoft priority name for each one. You can map multiple SpiraTeam fields to the same Axosoft fields, in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from Axosoft > SpiraTeam).

Note: The Axosoft external key needs to exactly match the display name of the priority inside Axosoft. If you change the name of a priority in Axosoft, you'll need to update the value in the data-mapping configuration as well.

c) Incident Severity

Click on the "Severity" hyperlink under Incident Standard Fields to bring up the Incident severity mapping configuration screen:

Artifact Type: Incident
Field Name: Severity

Severity	Active	External Key	Primary
1 - Critical	Yes	<input type="text" value="Critical"/>	Yes ▼
2 - High	Yes	<input type="text" value="High Impact"/>	Yes ▼
3 - Medium	Yes	<input type="text" value="Medium Impact"/>	Yes ▼
4 - Low	Yes	<input type="text" value="Low Impact"/>	Yes ▼

Update

The table lists each of the incident severities available in SpiraTeam and provides you with the ability to enter the matching Axosoft severity name for each one. You can map multiple SpiraTeam fields to the same Axosoft fields, in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from Axosoft > SpiraTeam).

Note: The Axosoft external key needs to exactly match the display name of the severity inside Axosoft. If you change the name of a severity in Axosoft, you'll need to update the value in the data-mapping configuration as well.

10.2.5. Configuring the Custom Property Mapping

Now that the various SpiraTeam standard fields have been mapped correctly, we need to configure the custom property mappings. This is used for both custom properties in SpiraTeam that map to custom

fields in Axosoft and also for custom properties in SpiraTeam that are used to map to standard fields in Axosoft (currently only Replication Procedures) that don't exist in SpiraTeam.

From the View/Edit Project Data Mapping screen, you need to click on the name of the Incident Custom Property that you want to add data-mapping information for. We will consider the *three* different types of mapping that you might want to enter:

Artifact Field Mapping		
Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:		
Artifact Type	Standard Fields	Custom Properties
Requirement	Component Importance Status Type	
Test Case	Priority	
Incident	Component Priority Severity Status Type	Notes OS

a) Scalar Custom Properties

This refers to custom properties that have a simple user-entered value and don't need to have their specific options mapped between SpiraTeam and Axosoft. All of the custom property types except **List** and **Multi-List** fall into this category (e.g. Text, Date, User, Boolean, Decimal, Integer, etc.)

Click on the hyperlink of the text custom property under Incident Custom Properties to bring up the custom property mapping configuration screen. For text custom properties there will be no values listed in the lower half of the screen.

Edit Custom Property Data Mapping | OnTime Sample Project ([Change Project](#))

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the OnTimeDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Artifact Type:* Incident

Name:* Notes (Text)

External Key:* x

[▶ Update](#)

You need to lookup the display name of the custom field in Axosoft that matches this custom property in SpiraTeam. Once you have entered the id of the custom field, click [Update].

b) List Custom Properties

Click on the hyperlink of the list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen. For list custom properties there will be a textbox for both the custom field itself and a mapping table for each of the custom property values that need to be mapped:

Edit Custom Property Data Mapping | OnTime Sample Project ([Change Project](#))

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the OnTimeDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Artifact Type:* Incident
Name:* OS (MultiList)
External Key:*

Custom Value Mapping

Value	Active	External Key
Linux	Yes	<input type="text" value="Linux"/>
Mac OS X	Yes	<input type="text" value="Mac OS X"/>
Windows 7	Yes	<input type="text" value="Windows 7"/>
Windows 8	Yes	<input type="text" value="Windows 8"/>

First you need to lookup the display name of the custom field in Axosoft that matches this custom property in SpiraTeam. This should be entered in the 'External Key' field below the name of the custom property.

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the full name of the custom field value as specified in Axosoft.

Once you have updated the various mapping sections, you are now ready to use the service.

10.3. Using SpiraTeam with Axosoft

Now that the integration service has been configured and the service started, initially any incidents created in SpiraTeam for the specified projects will be imported into Axosoft and vice versa.

At this point we recommend opening the Windows Event Viewer and choosing the Application Log. In this log any error messages raised by the SpiraTeam Data Sync Service will be displayed. If you see any error messages at this point, we recommend immediately stopping the SpiraTeam service and checking the various mapping entries. If you cannot see any defects with the mapping information, we recommend sending a copy of the event log message(s) to Inflectra customer services (support@inflectra.com) who will help you troubleshoot the problem.

To use SpiraTeam with Axosoft on an ongoing basis, we recommend the following general processes be followed:

- When running tests in SpiraTeam, defects found should be logged through the 'Add Incident' option as normal.
- Developers can log new defects into either SpiraTeam or Axosoft. In either case they will get loaded into the other system.
- Once created in one of the systems and successfully replicated to the other system, the incident should not be modified again inside SpiraTeam

- At this point, the incident should not be acted upon inside SpiraTeam, and all data changes to the defect should be made inside Axosoft. To enforce this, you can modify the workflows set up in SpiraTeam so that the various fields are marked as inactive for all the incident statuses other than the “New” status. This will allow someone to submit an incident in SpiraTeam, but will prevent them making changes in conflict with Axosoft after that point.
- As the defect progresses through the Axosoft workflow, changes to the status, priority, severity, and resolution will be updated automatically in SpiraTeam. In essence, SpiraTeam acts as a read-only viewer of these incidents.
- You are now able to perform test coverage and incident reporting inside SpiraTeam using the test cases managed by SpiraTeam and the incidents managed on behalf of SpiraTeam inside Axosoft.

11. Using SpiraTeam with Redmine

This section outlines how to use SpiraTeam in conjunction with the open-source Redmine bug-tracking and project management system. The built-in integration service allows the quality assurance team to manage their requirements and test cases in SpiraTeam, execute test runs in SpiraTeam, and then have the new incidents generated during the run be automatically loaded into Redmine. Once the incidents are loaded into Redmine as issues, the development team can then manage the lifecycle of these issues in Redmine, and have the status changes in Redmine be reflected back in SpiraTeam.

In addition, any issues logged directly into Redmine will get imported into SpiraTeam so that they can be linked to test cases and requirements.

► **STOP! Please make sure you have first read the Instructions in Section 1 before proceeding!**

11.1. Configuring the Plug-In

The next step is to configure the plug-in within SpiraTeam so that the system knows how to access the Redmine server. To start the configuration, please open up SpiraTeam in a web browser, log in using a valid account that has System-Administration level privileges and click on the System > Data Synchronization administration option from the left-hand navigation:

Data Synchronization

SpiraTeam is able to communicate with a variety of external systems and applications using its background data synchronization service and library of available plug-ins.

This page allows you to view, add and modify the list of plug-ins, make changes to their configuration, and verify the status of the most recent synchronization. In addition, you can force a particular plug-in to perform a complete data-synchronization by resetting its date of last sync:

[Refresh](#) [+ Add](#)

Plug-In	Data Mapping	Last Sync Date	Status	Operations
BugzillaDataSync	View Project Mappings	6/21/2016 1:09:56 PM	Failure	Force Re-Sync Edit Delete View Errors
FogBugzDataSync	View Project Mappings	6/21/2016 1:09:56 PM	Failure	Force Re-Sync Edit Delete View Errors
JiraDataSync	View Project Mappings	6/21/2016 1:09:56 PM	Warning	Force Re-Sync Edit Delete View Errors
MsTfsDataSync	View Project Mappings	6/21/2016 1:09:56 PM	Warning	Force Re-Sync Edit Delete View Errors

This screen lists all the plug-ins already configured in the system. Depending on whether you chose the option to include sample data in your installation or not, you will see either an empty screen or a list of sample data-synchronization plug-ins.

If you already see an entry for **RedmineDataSync** you should click on its “Edit” link. If you don’t see such an entry in the list, please click on the [Add] button instead. In either case you will be taken to the following screen where you can enter or modify the Redmine Data-Synchronization plug-in:

Edit Plug-In | RedmineDataSync

[<< Back to Data Synchronization Home](#)

Please enter/edit the following information for the plug-in. The exact information that needs to be entered is specific to each plug-in, and you should refer to the documentation for the plug-in when entering/changing it:

Name*:	<input type="text" value="RedmineDataSync"/>
Description:	<input type="text" value="Synchronizes incidents in SpiraTeam with issues in Redmine"/>
Connection Info*:	<input type="text" value="http://demo.redmine.org"/>
Login*:	<input type="text" value="inflectrtest"/>
Password:	<input type="password" value="*****"/>
Time Offset*:	<input type="text" value="0"/>
	<input type="checkbox"/> Auto-Map Users
Custom 01:	<input type="text"/>
Custom 02:	<input type="text"/>
Custom 03:	<input type="text"/>
Custom 04:	<input type="text"/>
Custom 05:	<input type="text"/>
	<input type="button" value="Update"/>

You need to fill out the following fields for the Redmine Plug-in to operate correctly:

- **Name** – this needs to be set to **RedmineDataSync**. This needs to match the name of the plug-in DLL assembly that was copied into the C:\Program Files\SpiraTeam\Bin folder (minus the .dll file extension). If you renamed the RedmineDataSync.dll file for any reason, then you need to change the name here to match.
- **Description** – this should be set to a description of the plug-in. This is an optional field that is used for documentation purposes and is not actually used by the system.
- **Connection Info** – this should be the base URL of the Redmine installation. As an example, for the public demo installation of Redmine, it would be: <http://demo.redmine.org>
- **Login** – this should be set to a valid login to the Redmine installation – the login needs to have permissions to create and view bugs and versions within Redmine.
- **Password** – this should be set to the password of the login specified above.
- **Time Offset** – normally this should be set to zero, but if you find that issues being changed in Redmine are not being updated in SpiraTeam, try increasing the value as this will tell the data-synchronization plug-in to add on the time offset (in hours) when comparing date-time stamps. Also if your Redmine installation is running on a server set to a different time-zone, then you should add in the number of hours difference between the servers' time-zones here.
- **Auto-Map Users** – This changes the way that the plugin maps users in SpiraTeam to those in Redmine:
 - ▷ **Auto-Map = True**
With this setting, all users in SpiraTeam need to have the same username as those in Redmine. If this is the case then you do not need to perform the user-mapping task

outlined in section 11.2.2. This is a big time-saver if you can guarantee that all usernames are the same in both systems.

▷ **Auto-Map = False**

With this setting, users in SpiraTeam and Redmine are free to have different usernames because you specify the corresponding Redmine name for each user as outlined in section 11.2.2.

- ▶ **Custom 01** – This should be set to the word “**false**” if you want to have the plugin restrict synchronization to not create any new incidents in Spira.
- ▶ **Custom 02** – This should be set to the word “**false**” if you want to have the plugin restrict synchronization to not create any new issues in Redmine.
- ▶ **Custom 03 – 05** – these are not currently used by the Redmine data-sync plug-in and can be left blank.

11.2. Configuring the Data Mapping

Next, you need to configure the data mapping between SpiraTeam and Redmine. This allows the various projects, users, releases, incident types, statuses, priorities and custom property values used in the two applications to be related to each other. This is important, as without a correct mapping, there is no way for the integration service to know that an “Duplicate” incident in SpiraTeam is the same as a “Rejected” bug in Redmine (for example).

The following mapping information needs to be setup in SpiraTeam:

- ▶ The mapping of the project identifiers for the projects that need to be synchronized
- ▶ The mapping of users in the system
- ▶ The mapping of releases (equivalent to Redmine versions) in the system
- ▶ The mapping of the various standard fields in the system
- ▶ The mapping of the various custom properties in the system

Each of these is explained in turn below:

11.2.1. Configuring the Project Mapping

From the data synchronization administration page, you need to click on the “View Project Mappings” hyperlink next to the Redmine plug-in name. This will take you to the data-mapping home page for the currently selected project:

RedmineDataSync Project Data Mapping | Redmine Demo Project [\(Change Project\)](#)

[<< Back to Data Synchronization Home](#)

To configure this project for data mapping with the 'RedmineDataSync' plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name: Redmine Demo Project
External Key:*
Active:*

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Requirement	Importance Status	
Test Case	Priority	
Incident	Priority Severity Status Type	
Release		
Task	Priority Status	

If the project name does not match the name of the project you want to configure the data-mapping for, click on the “(Change Project)” hyperlink to change the current project.

To enable this project for data-synchronization with Redmine, you need to enter:

- **External Key** – This should be set to the name of the equivalent project in Redmine.
- **Active Flag** – Set this to ‘Yes’ so that SpiraTeam knows that you want to synchronize data for this project. Once the project has been completed, setting the value to “No” will stop data synchronization, reducing network utilization.

Click [Update] to confirm these settings. Once you have enabled the project for data-synchronization, you can now enter the other data mapping values outlined below.

Note: Once you have successfully configured the project, when creating a new project, you should choose the option to “Create Project from Existing Project” rather than “Use Default Template” so that all the project mappings get copied across to the new project.

11.2.2. Configuring the User Mapping

To configure the mapping of users in the two systems, you need to go to Administration > Users > View Edit Users, which will bring up the list of users in the system. Then click on the “Edit” button for a particular user that will be editing issues in Redmine:

Add/Edit User | Rory Jones

Please review the information listed below and make any changes if necessary. Once you have finished, click [Update/Insert] to confirm them.

First Name*:

Middle Initial:

Last Name*:

User Name*:

Department:

Email Address*:

Email Enabled*:

System Administrator*:

Active*:

Locked Out*:

RSS Token: Enable RSS Feeds

Project Membership | Passwords & Security | **Data Mapping**

RedmineDataSync ID:

You will notice that in the special Data Mapping tab for the user is a list of all the configured data-synchronization plug-ins. In the text box next to the Redmine Data-Sync plug-in you need to enter the **numeric ID** for this user in Redmine. This will allow the data-synchronization plug-in to know which user in Redmine matches this SpiraTeam user. Click [Update] once you've entered the appropriate ID value. You should now repeat for the other users who will be active in both systems.

11.2.3. Configuring the Release Mapping

Now that the projects and users have been mapped correctly, we need to configure the mapping between Releases/Iterations in SpiraTeam and Versions in Redmine. To do this, please navigate to Planning > Releases and click on the Release/Iteration in question. Make sure you have the 'Overview' tab visible and expand the "Details" section of the release/iteration:

Release: 1.0.0.0 - Release 1.0 [RL:000051]

Name:

Overview | Incidents * | Reqs & Tasks | Test Cases | Test Runs | Attachments | History *

▼ Details

Version #*: Creator*:

Start/End Date: - Creation Date: 8/27/2013 12:06:34 PM

Resources: Active*:

Non-Working: person days Iteration*:

Plan Effort: 192.00 Available Effort: 162.00

RedmineDataSync ID:

In addition to the standard fields and custom properties configured for Releases, you will see an additional text property called "**RedmineDataSync ID**" that is used to store the mapped external identifier for the equivalent Version in Redmine. You need to enter the numeric ID of the equivalent version in Redmine, enter it into this text-box and click [Save]. You should now repeat for all the other releases and iterations in the project.

In addition, any Versions that have already been created in Redmine will be automatically imported into SpiraTeam if they do not already exist in SpiraTeam and they have not already been mapped.

11.2.4. Configuring the Standard Field Mapping

Now that the projects, user and releases have been mapped correctly, we need to configure the standard incident fields. To do this, go to Administration > System > Data Synchronization and click on the “View Project Mappings” for the RedmineDataSync plug-in entry:

RedmineDataSync Project Data Mapping | Redmine Demo Project ([Change Project](#))

[<< Back to Data Synchronization Home](#)

To configure this project for data mapping with the 'RedmineDataSync' plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name: Redmine Demo Project

External Key:*

Active:* ▼

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Requirement	Importance Status	
Test Case	Priority	
Incident	Priority Severity Status Type	
Release		
Task	Priority Status	

From this screen, you need to click on Priority, Type and Status in turn to configure their values:

a) Incident Status

Click on the “Status” hyperlink under Incident Standard Fields to bring up the Incident status mapping configuration screen:

Edit Standard Field Data Mapping | Redmine Demo Project ([Change Project](#))

[<< Back to Project Mappings](#)

To configure the data mappings for this field, please fill in the external key values for the RedmineDataSync plug-in next to each SpiraTeam value:

Field Value	Active	External Key	Primary
Assigned	Yes	<input type="text" value="2"/>	Yes ▼
Closed	Yes	<input type="text" value="5"/>	Yes ▼
Duplicate	Yes	<input type="text" value="6"/>	No ▼
New	Yes	<input type="text" value="1"/>	Yes ▼
Not Reproducible	Yes	<input type="text" value="6"/>	Yes ▼
Open	Yes	<input type="text" value="2"/>	No ▼
Reopen	Yes	<input type="text" value="2"/>	No ▼
Resolved	Yes	<input type="text" value="3"/>	Yes ▼

[▶ Update](#)

The table lists each of the incident statuses available in SpiraTeam and provides you with the ability to enter the matching Redmine bug status ID for each one. You can map multiple SpiraTeam fields to the same Redmine fields (e.g. Open and Assigned in SpiraTeam are both equivalent to “In Progress” (ID=2) in Redmine), in which case only one of the two values can be listed as Primary = Yes as that’s the value that’s used on the reverse synchronization (from Redmine > SpiraTeam).

b) Incident Priority

Click on the “Priority” hyperlink under Incident Standard Fields to bring up the Incident Priority mapping configuration screen:

Edit Standard Field Data Mapping | Redmine Demo Project ([Change Project](#))

[<< Back to Project Mappings](#)

To configure the data mappings for this field, please fill in the external key values for the RedmineDataSync plug-in next to each SpiraTeam value:

Field Value	Active	External Key	Primary
1 - Critical	Yes	<input type="text" value="6"/>	Yes ▼
2 - High	Yes	<input type="text" value="5"/>	Yes ▼
3 - Medium	Yes	<input type="text" value="4"/>	Yes ▼
4 - Low	Yes	<input type="text" value="3"/>	Yes ▼

[▶ Update](#)

The table lists each of the incident priorities available in SpiraTeam and provides you with the ability to enter the matching Redmine priority ID for each one. You can map multiple SpiraTeam fields to the same Redmine fields, in which case only one of the two values can be listed as Primary = Yes as that’s the value that’s used on the reverse synchronization (from Redmine > SpiraTeam).

c) Incident Type

Incident types in SpiraTeam are equivalent to Trackers in Redmine. Click on the “Type” hyperlink under Incident Standard Fields to bring up the Incident type mapping configuration screen:

Edit Standard Field Data Mapping | Redmine Demo Project [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this field, please fill in the external key values for the RedmineDataSync plug-in next to each SpiraTeam value:

Field Value	Active	External Key	Primary
Bug	Yes	<input type="text" value="1"/>	Yes ▼
Change Request	Yes	<input type="text" value="2"/>	No ▼
Enhancement	Yes	<input type="text" value="2"/>	Yes ▼
Incident	Yes	<input type="text" value="1"/>	No ▼
Issue	Yes	<input type="text" value="3"/>	Yes ▼
Limitation	Yes	<input type="text" value="1"/>	No ▼
Risk	Yes	<input type="text" value="1"/>	No ▼
Training	Yes	<input type="text" value="3"/>	No ▼

[▶ Update](#)

The table lists each of the incident types available in SpiraTeam and provides you with the ability to enter the matching Redmine **Tracker ID** for each one. You can map multiple SpiraTeam fields to the same Redmine tracker values, in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from Redmine > SpiraTeam).

11.2.5. Configuring the Custom Property Mapping

Now that the various SpiraTeam standard fields have been mapped correctly, we need to configure the custom property mappings. This is used for custom properties in SpiraTeam that map to custom fields in Redmine. You will need to first make sure that the custom properties and associated custom lists have been created in both systems:

Edit Custom Properties | Redmine Demo Project [\(Change Project\)](#)

The following custom properties are currently defined for the selected artifact type. To view the custom properties for a different artifact type, click on the dropdown menu below.

You can add, edit or remove custom property definitions by clicking the appropriate link in the grid below:

Artifact Type: <input type="text" value="Incident"/>				
Field #	Property Name	Type	Legacy Name	Actions
1	Custom Property 1	Text	TEXT_01	> Edit Definition Remove
2	Custom Property 2	List	LIST_01	> Edit Definition Remove
3				> Add Definition
4				> Add Definition

From the View/Edit Project Data Mapping screen, you need to click on the name of the Incident Custom Property that you want to add data-mapping information for:

Project Name: Redmine Demo Project
 External Key:*
 Active:*

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Requirement	Importance Status	
Test Case	Priority	
Incident	Priority Severity Status Type	Custom Property 1 Custom Property 2
Release		
Task	Priority Status	

We will consider the two different types of mapping that you might want to enter.

a) Scalar Custom Properties

This refers to custom properties that have a simple user-entered value and don't need to have their specific options mapped between SpiraTeam and Redmine. All of the custom property types except **List** and **Multi-List** fall into this category (e.g. Text, Date, Boolean, Decimal, Integer, etc.)

Click on the hyperlink of the scalar custom property under Incident Custom Properties to bring up the custom property mapping configuration screen. For scalar custom properties there will be no values listed in the lower half of the screen.

Edit Custom Property Data Mapping | Redmine Demo Project [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the RedmineDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Name: Custom Property 1
 External Key:

Custom Value Mapping

Property Value	Active	External Key

You need to enter the ID of the custom field in Redmine that matches this custom property in SpiraTeam. Once you have entered the id of the custom field, click [Update].

b) List Custom Properties

This refers to custom properties that are either of type **List** or **Multi-List**. Click on the hyperlink of the list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen. For list custom properties there will be a textbox for both the custom field itself and a mapping table for each of the custom property values that need to be mapped:

Edit Custom Property Data Mapping | Redmine Demo Project ([Change Project](#))

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the RedmineDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Name: Custom Property 2

External Key:

Custom Value Mapping

Property Value	Active	External Key
Custom Value 1	Yes	<input type="text" value="Value 1"/>
Custom Value 2	Yes	<input type="text" value="Value 2"/>

First you need to find the ID of the custom field in Redmine that matches this custom property in SpiraTeam. This should be entered in the 'External Key' field below the name of the custom property.

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the full name (not the id this time) of the custom field value as specified in Redmine.

Once you have updated the various mapping sections, you are now ready to use the service.

11.3. Using SpiraTest with Redmine

Now that the integration service has been configured and the service started, initially any incidents created in SpiraTeam for the specified projects will be imported into Redmine. At this point we recommend opening the Windows Event Viewer and choosing the Application Log. In this log any error messages raised by the Data Synchronization service will be displayed. If you see any error messages at this point, we recommend immediately stopping the service and checking the various mapping entries. If you cannot see any issues with the mapping information, we recommend sending a copy of the event log message(s) to Inflectra customer services (support@inflectra.com) who will help you troubleshoot the problem.

To use SpiraTeam with Redmine on an ongoing basis, we recommend the following general processes be followed:

- ▶ When running tests in SpiraTeam, defects found should be logged through the Test Execution Wizard as normal.
- ▶ Developers can log new defects into either SpiraTeam or Redmine. In either case they will get loaded into the other system.
- ▶ Once created in one of the systems and successfully replicated to the other system, the incident should not be modified again inside SpiraTeam
 - ▷ All data changes to the issue should be made inside Redmine.
 - ▷ To enforce this, you can modify the workflows set up in SpiraTeam so that the various fields are marked as inactive for all the incident statuses other than the "New" status.
 - ▷ This will allow someone to submit an incident in SpiraTeam, but will prevent them making changes in conflict with Redmine after that point.
- ▶ As the issue progresses through the Redmine workflow, changes to the status, priority, tracker, and target version will be updated automatically in SpiraTeam, and any notes added will be

added to SpiraTeam as new comments. In essence, SpiraTeam acts as a read-only viewer of these incidents.

- You are now able to perform test coverage and incident reporting inside SpiraTeam using the test cases managed by SpiraTeam and the incidents managed on behalf of SpiraTeam inside Redmine.

12. Using Spira with VersionOne

This section outlines how to use SpiraTest, SpiraPlan or SpiraTeam (hereafter referred to as SpiraTeam) in conjunction with the Version One (V1) project management system from Collabnet (hereafter called V1).

The built-in integration service allows the quality assurance team to manage their requirements and test cases in SpiraTeam, execute test runs in SpiraTeam, and then have the new incidents generated during the run be automatically loaded into V1 as new defects. In addition, user stories in V1 will be automatically added into SpiraTeam as new requirements that you can write test plans for.

Once the incidents are loaded into V1 as defects, the development team can then manage the lifecycle of these defects in V1, and have the status changes in V1 be reflected back in SpiraTeam.

► **STOP! Please make sure you have first read the Instructions in Section 1 before proceeding!**

12.1. Configuring the Plug-In

This section outlines how to configure the integration service to connect to V1. It assumes that you already have a working installation of SpiraTeam v5.0 or later and an existing provisioned instance of V1. If you have an earlier version of SpiraTeam, you will need to upgrade to at least v5.0 before trying to integrate with V1.

The steps that need to be performed to configure integration with V1 are as follows:

- Setup the plug-in in SpiraTeam to point to the correct instance of V1
- Configure the data field mappings between SpiraTeam and V1
- Start synchronization and verify data transfer

12.1.1. Configuring the Plug-In

The next step is to configure the plug-in within SpiraTeam so that the system knows how to access the V1 instance. To start the configuration, please open up SpiraTeam in a web browser, log in using a valid account that has System-Administration level privileges and click on the System > Data Synchronization administration option from the left-hand navigation:

Plug-in	Data Mapping	Last Sync Date	Status	Operations
Version One	VersionOneDataSync	5/6/2018 8:15:21 AM	Success	Reset Sync Edit Delete View Errors

This screen lists all the plug-ins already configured in the system. Depending on whether you chose the option to include sample data in your installation or not, you will see either an empty screen or a list of sample data-synchronization plug-ins.

If you already see an entry for **VersionOneDataSync** you should click on its “Edit” link. If you don’t see such an entry in the list, please click on the [Add] button instead. In either case you will be taken to the following screen where you can enter or modify the V1 Data-Synchronization plug-in:

Version One Edit Plug-In

[← Back to Data Synchronization Home](#)

Please enter/edit the following information for the plug-in. The exact information that needs to be entered is specific to each plug-in, and you should refer to the documentation for the plug-in when entering/changing it:

Name:	<input type="text" value="VersionOneDataSync"/>	The physical name of the plugin (e.g. JiraDataSync) that matches the DLL being used
Caption:	<input type="text" value="Version One"/>	The display name of the plugin (e.g. Internal Jira) that can be used to distinguish between multiple instances of the same plugin type
Description:	<input type="text"/>	The long description of the plugin (e.g. 'Connects to the in-house bug-tracker')
Connection Info:	<input type="text" value="https://www1.v1host.com/techSolutions"/>	The URL or other connection string used to connect to the external bug-tracker
Login:	<input type="text" value="ben@v1.com"/>	
Password:	<input type="password" value="*****"/>	The login and password used to connect to Version One

Time Offset:	<input type="text" value="0"/> hours	The number of hours to look back for changes to synchronization. Normally should be set to zero (0) unless not being synchronized
Auto-Map Users	<input checked="" type="checkbox"/> Yes	
Custom 01:	<input type="text"/>	
Custom 02:	<input type="text"/>	
Custom 03:	<input type="text"/>	
Custom 04:	<input type="text"/>	
Custom 05:	<input type="text"/>	
	<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

You need to fill out the following fields for the V1 Plug-in to operate correctly:

- **Name** – this needs to be set to **VersionOneDataSync**. This needs to match the name of the plug-in DLL assembly that was copied into the C:\Program Files (x86)\SpiraTeam\Bin folder (minus the .dll file extension). If you renamed the OnTimeDataSync.dll file for any reason, then you need to change the name here to match.
- **Caption** – this is the display name of the V1 plugin and it can be meaningful name such as “Version One”, “V1”, or “V1 Instance 1”.

- **Description** – this should be set to a description of the plug-in. This is an optional field that is used for documentation purposes and is not actually used by the system.
- **Connection Info** – this should be the full URL to V1. This is typically something like: <https://www1.v1host.com/CompanyName>.
- **Login** – this should be set to the login that you use to access V1 through its web interface
- **Password** – this should be set to the password that you use to access V1 through its web interface
- **Time Offset** – normally this should be set to zero, but if you find that defects being changed in V1 are not being updated in SpiraTeam, try increasing the value as this will tell the data-synchronization plug-in to add on the time offset (in hours) when comparing date-time stamps.
- **Auto-Map Users** – This changes the way that the plugin maps users in SpiraTeam to those in V1:
 - ▷ **Auto-Map = True**
With this setting, all users in SpiraTeam need to have the same username as those in V1. If this is the case then you do not need to perform the user-mapping task outlined in section 12.2.2. This is a big time-saver if you can guarantee that all usernames are the same in both systems.
 - ▷ **Auto-Map = False**
With this setting, users in SpiraTeam and V1 are free to have different usernames because you specify the corresponding V1 login for each user as outlined in section 12.2.2.
- **Custom 01** – this should be set to “True” if you want the plugin to log warnings about missing user mappings
- **Custom 02-05** – these are not currently used by the V1 data-sync plug-in and can be left blank.

12.2. Configuring the Data Mapping

Next, you need to configure the data mapping between SpiraTeam and V1. This allows the various projects, users, releases, incident statuses, priorities, severities and custom property values used in the two applications to be related to each other. This is important, as without a correct mapping, there is no way for the integration service to know that an “Open” incident in SpiraTeam is the same as an “Open” defect in V1 (for example).

The following mapping information needs to be setup in SpiraTeam:

- The mapping of the project identifiers for the projects that need to be synchronized
- The mapping of users in the system
- The mapping of releases in the system
- The mapping of the various incident standard fields in the system
- The mapping of the various requirement standard fields in the system

Each of these is explained in turn below:

12.2.1. Configuring the Project Mapping

From the data synchronization administration page, you need to click on the “View Project Mappings” hyperlink next to the V1 plug-in name. This will take you to the data-mapping home page for the currently selected project:

Version One Project Data Mapping Sample V1 Project

[← Back to Data Synchronization Home](#)
[↻ Change Project](#)

To configure this project for data mapping with the 'Version One' plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name:*

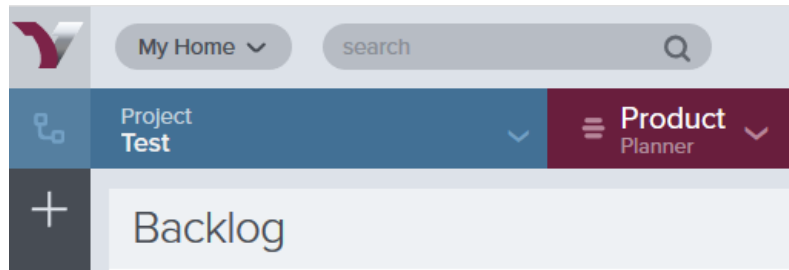
External Key:*

Active:* Yes

If the project name does not match the name of the project you want to configure the data-mapping for, click on the “(Change Project)” hyperlink to change the current project.

To enable this project for data-synchronization with V1, you need to enter:

- **External Key** – This should be set to the **name** of the project in V1:



If you have sub-projects, you can map to one of those using the syntax:
Project/SubProject

- **Active Flag** – Set this to 'Yes' so that SpiraTeam knows that you want to synchronize data for this project. Once the project has been completed, setting the value to “No” will stop data synchronization, reducing network utilization.

Click [Update] to confirm these settings. Once you have enabled the project for data-synchronization, you can now enter the other data mapping values outlined below.

Note: Once you have successfully configured the project, when creating a new project, you should choose the option to “Create Project from Existing Project” rather than “Use Default Template” so that all the project mappings get copied across to the new project.

12.2.2. Configuring the User Mapping

(This section can be skipped if you enabled the 'AutoMap Users' option earlier).

To configure the mapping of users in the two systems, you need to go to Administration > Users > View Edit Users, which will bring up the list of users in the system. Then click on the “Edit” button for a particular user that will be editing defects in V1:

Membership and Mapping

Project Membership Data Mapping

Version One ID:

Save Cancel

You will notice that in the **Data Mapping** tab for the user is a list of all the configured data-synchronization plug-ins. In the text box next to the V1 Data-Sync plug-in you need to enter the **Login Name** for this username in V1. This will allow the data-synchronization plug-in to know which user in SpiraTeam match which equivalent user in V1. Click [Update] once you've entered the appropriate login name. You should now repeat for the other users who will be active in both systems.

12.2.3. Configuring the Release/Iteration Mapping

When the data-synchronization service runs, when it comes across a new Release or Sprint/Timebox in V1 that it has not seen before, it will create a new Release or Iteration in SpiraTeam. Therefore, when using both systems together, it is recommended that you only enter new Releases in V1 and let the data-synchronization service add them to SpiraTeam.

To see this mapping, inside SpiraTeam, navigate to Planning > Releases and click on the Release/Iteration in question. Make sure you have the 'Overview' tab visible and expand the "Details" section of the release/iteration:

Release 1.0

1.0.0.0 [RL:23] Type: Major Release Status: Planned Operations

Overview Incidents Reqs & Tasks Test Cases Test Runs Attachments History

People **Properties**

Creator: System Administrator Version #: 1.0.0.0

Owner: -- None -- Version One ID: Timebox:10001

In addition to the standard fields and custom properties configured for Releases, you will see an additional text property called "**V1DataSync ID**" that is used to store the mapped external identifier for the equivalent Version in V1.

12.2.4. Configuring the Incident Field Mapping

Now that the projects, user and releases have been mapped correctly, we need to configure the standard incident fields. To do this, go to Administration > System > Data Synchronization and click on the "View Project Mappings" for the V1DataSync plug-in entry:



From this screen, you need to click on Status, Priority, Type in turn to configure their values (V1 doesn't support different defect severities):

a) Incident Status

Click on the "Status" hyperlink under Incident Standard Fields to bring up the Incident status mapping configuration screen:

Artifact Type: Incident
Field Name: Status

Status	Active	External Key Version One	Primary
Assigned	Yes	In Progress	<input checked="" type="checkbox"/> Yes
Closed	Yes	Done	<input checked="" type="checkbox"/> Yes
Duplicate	Yes		<input checked="" type="checkbox"/> Yes
New	Yes	Future	<input type="checkbox"/> No
Not Reproducible	Yes		<input checked="" type="checkbox"/> Yes
Open	Yes	Future	<input checked="" type="checkbox"/> Yes
Reopen	Yes		<input checked="" type="checkbox"/> Yes
Resolved	Yes	Accepted	<input checked="" type="checkbox"/> Yes

The table lists each of the incident statuses available in SpiraTeam and provides you with the ability to enter the matching V1 defect status names for each one. You can map multiple SpiraTeam fields to the same V1 fields (e.g. New and Open in SpiraTeam are both equivalent to Future in V1), in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from V1 > SpiraTeam).

We recommend that you always point the New and Open statuses inside SpiraTeam to point to the "Future" status inside V1 and make Open in SpiraTeam the Primary status of the two. This is recommended so that as new incidents in SpiraTeam get synched over to V1, they will get switched to the Future status in V1 which will then be synched back to "Open" in SpiraTeam. That way you'll be able to see at a glance which incidents have been synched with V1 and those that haven't.

Note: The V1 external key needs to exactly match the display name of the status inside V1. If you change the name of a status in V1, you'll need to update the value in the data-mapping configuration as well.

b) Incident Priority

Click on the "Priority" hyperlink under Incident Standard Fields to bring up the Incident Priority mapping configuration screen:

Artifact Type: Incident
Field Name: Priority

Priority	Active	External Key Version One	Primary
1 - Critical	Yes	<input type="text"/>	<input checked="" type="checkbox"/>
2 - High	Yes	High	<input checked="" type="checkbox"/>
3 - Medium	Yes	Medium	<input checked="" type="checkbox"/>
4 - Low	Yes	Low	<input checked="" type="checkbox"/>

The table lists each of the incident priorities available in SpiraTeam and provides you with the ability to enter the matching V1 priority name for each one. You can map multiple SpiraTeam fields to the same V1 fields, in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from V1 > SpiraTeam).

Note: The V1 external key needs to exactly match the display name of the priority inside V1. If you change the name of a priority in V1, you'll need to update the value in the data-mapping configuration as well.

c) Incident Type

Click on the "Type" hyperlink under Incident Standard Fields to bring up the Incident type mapping configuration screen:

Artifact Type: Incident
Field Name: Type

Type	Active	External Key Version One	Primary
Bug	Yes	Code	<input checked="" type="checkbox"/>
Change Request	Yes	<input type="text"/>	<input checked="" type="checkbox"/>
Enhancement	Yes	<input type="text"/>	<input checked="" type="checkbox"/>
Incident	Yes	<input type="text"/>	<input checked="" type="checkbox"/>
Issue	Yes	<input type="text"/>	<input checked="" type="checkbox"/>
Limitation	Yes	<input type="text"/>	<input checked="" type="checkbox"/>
Risk	Yes	<input type="text"/>	<input checked="" type="checkbox"/>
Training	Yes	Documentation	<input checked="" type="checkbox"/>

The table lists each of the incident types available in SpiraTeam and provides you with the ability to enter the matching V1 defect type name for each one. You can map multiple SpiraTeam fields to the same V1 fields, in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from V1 > SpiraTeam).

Note: The V1 external key needs to exactly match the display name of the defect type inside V1. If you change the name of a defect type in V1, you'll need to update the value in the data-mapping configuration as well.

12.2.5. Configuring the Requirement Field Mapping

Next, we need to configure the standard requirement fields. To do this, go to Administration > System > Data Synchronization and click on the "View Project Mappings" for the V1DataSync plug-in entry:

Artifact Type	Standard Fields
Requirement	Component Importance Status Type

From this screen, you need to click on Priority, Severity and Status in turn to configure their values (V1 doesn't support different defect types):

a) Requirement Status

Click on the "Status" hyperlink under Requirement Standard Fields to bring up the Requirement status mapping configuration screen:

Artifact Type: Requirement
Field Name: Status

Status	Active	External Key	Version One	Primary
Accepted	Yes	Accepted		<input checked="" type="checkbox"/>
Completed	Yes	Done		<input type="checkbox"/>
Developed	Yes	Done		<input checked="" type="checkbox"/>
In Progress	Yes	In Progress		<input checked="" type="checkbox"/>
Obsolete	Yes			<input checked="" type="checkbox"/>
Planned	Yes	Future		<input checked="" type="checkbox"/>
Rejected	Yes			<input checked="" type="checkbox"/>
Requested	Yes	Future		<input type="checkbox"/>
Tested	Yes	Done		<input type="checkbox"/>
Under Review	Yes	Future		<input type="checkbox"/>

The table lists each of the requirement statuses available in SpiraTeam and provides you with the ability to enter the matching V1 user story status names for each one. You can map multiple SpiraTeam fields to the same V1 fields (e.g. Requested and Planned in SpiraTeam are both equivalent to Future in V1), in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from V1 > SpiraTeam).

Note: The V1 external key needs to exactly match the display name of the status inside V1. If you change the name of a status in V1, you'll need to update the value in the data-mapping configuration as well.

b) Requirement Importance

Click on the "Importance" hyperlink under Requirement Standard Fields to bring up the Requirement Importance mapping configuration screen:

Artifact Type: Requirement
Field Name: Importance

Importance	Active	External Key	Version One	Primary
1 - Critical	Yes			<input checked="" type="checkbox"/>
2 - High	Yes	High		<input checked="" type="checkbox"/>
3 - Medium	Yes	Medium		<input checked="" type="checkbox"/>
4 - Low	Yes	Low		<input checked="" type="checkbox"/>

The table lists each of the requirement importances available in SpiraTeam and provides you with the ability to enter the matching V1 user story priority name for each one. You can map multiple SpiraTeam fields to the same V1 fields, in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from V1 > SpiraTeam).

Note: The V1 external key needs to exactly match the display name of the priority inside V1. If you change the name of a priority in V1, you'll need to update the value in the data-mapping configuration as well.

c) Requirement Type

Click on the "Type" hyperlink under Requirement Standard Fields to bring up the requirement type mapping configuration screen:

Artifact Type: Requirement
Field Name: Type

Type	Active	External Key	Version One	Primary
Design Element	Yes			<input checked="" type="checkbox"/>
Feature	Yes			<input checked="" type="checkbox"/>
Need	Yes			<input checked="" type="checkbox"/>
Package	Yes			<input checked="" type="checkbox"/>
Quality	Yes			<input checked="" type="checkbox"/>
Use Case	Yes			<input checked="" type="checkbox"/>
User Story	Yes	User Story		<input checked="" type="checkbox"/>

The table lists each of the requirement types available in SpiraTeam and provides you with the ability to enter the matching V1 user story type name for each one. You can map multiple SpiraTeam fields to the same V1 fields, in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from V1 > SpiraTeam).

Note: The V1 external key needs to exactly match the display name of the user story type inside V1. If you change the name of a user story type in V1, you'll need to update the value in the data-mapping configuration as well.

12.2.6. Mapping a Custom Property to the V1 Display ID

Version One has two unique IDs for all artifacts, a display ID (e.g. D-23232) and a physical ID (Defect:11291). Now the built in Data Sync ID in SpiraTeam will contain the physical ID of the V1 artifact.

However, if you also want to see the V1 display IDs in SpiraTeam, you can map a custom property to the special **V1DisplayId** external key. This can be done for requirements:

Edit Custom Property Data Mapping Sample V1 Project

[← Back to Project Mappings](#) [↻ Change Project](#)

To configure the data mappings for this custom property, please fill in the **Version One** external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Artifact Type:

Name: Text

External Key:

And for incidents:

Edit Custom Property Data Mapping Sample V1 Project

[← Back to Project Mappings](#) [↻ Change Project](#)

To configure the data mappings for this custom property, please fill in the **Version One** external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Artifact Type:

Name: Text

External Key:

Once you have updated the various mapping sections, you are now ready to use the service.

12.3. Using SpiraTeam with V1

Now that the integration service has been configured and the service started, initially any incidents created in SpiraTeam for the specified projects will be imported into V1 and vice versa. In addition, any existing user stories in V1 will get added to SpiraTeam as requirements.

At this point we recommend opening the Windows Event Viewer and choosing the Application Log. In this log any error messages raised by the SpiraTeam Data Sync Service will be displayed. If you see any error messages at this point, we recommend immediately stopping the SpiraTeam service and checking the various mapping entries. If you cannot see any defects with the mapping information, we recommend sending a copy of the event log message(s) to Inflectra customer services (support@inflectra.com) who will help you troubleshoot the problem.

To use SpiraTeam with V1 on an ongoing basis, we recommend the following general processes be followed:

- When running tests in SpiraTeam, defects found should be logged through the 'Add Incident' option as normal.
- Developers can log new defects into either SpiraTeam or V1. In either case they will get loaded into the other system.

- Once created in one of the systems and successfully replicated to the other system, the incident should not be modified again inside SpiraTeam
- User stories created in V1 will get added to SpiraTeam as requirements. You can now write test cases and associate them in SpiraTeam with these requirements.
- As the defect progresses through the V1 workflow, changes to the status, priority, severity, and resolution will be updated automatically in SpiraTeam. In essence, SpiraTeam acts as a read-only viewer of these incidents.
- You are now able to perform test coverage and incident reporting inside SpiraTeam using the requirements and test cases managed by SpiraTeam and the incidents managed on behalf of SpiraTeam inside V1.

13. Using Spira with OnTime 11

This section outlines how to use SpiraTest, SpiraPlan or SpiraTeam (hereafter referred to as SpiraTeam) in conjunction with the OnTime 11+ defect tracking system from AxoSoft. The built-in integration service allows the quality assurance team to manage their requirements and test cases in SpiraTeam, execute test runs in SpiraTest, and then have the new incidents generated during the run be automatically loaded into OnTime.

Once the incidents are loaded into OnTime as defects, the development team can then manage the lifecycle of these defects in OnTime, and have the status changes in OnTime be reflected back in SpiraTeam.

Note: This section refers to integrating with the older SOAP API that was available in AxoSoft OnTime 11 (2010). This API was removed from AxoSoft OnTime in 2012 and we recommend you use the AxoSoft OnTime 14 Plugin instead that is described in section 10 above if you are using OnTime 14 or later.

▶ Note: The OnTime11 Plug-In is Not Available on the Inflectra Cloud-Based DataSync Service.

13.1. Configuring the Integration Service

This section outlines how to configure the integration service to export incidents into OnTime and pick up subsequent status changes in OnTime and have them update SpiraTeam. It assumes that you already have a working installation of SpiraTest, SpiraPlan or SpiraTeam v2.3 or later and a working installation of OnTime 2010 or later. If you have an earlier version of SpiraTeam, you will need to upgrade to at least v2.3 before trying to integrate with OnTime.

The steps that need to be performed to configure integration with OnTime are as follows:

- Install and configure the OnTime SDK if you have not already done so
- Download the OnTime11 Data-Sync plug-in for SpiraTeam from our website
- Setup the plug-in in SpiraTeam to point to the correct instance of OnTime
- Configure the data field mappings between SpiraTeam and OnTime
- Start the service and verify data transfer

13.1.1. Install and Configure the OnTime SDK

The API for accessing OnTime remotely is a separate download from the main OnTime application, and should be downloaded and installed from AxoSoft's website onto your OnTime server. It typically adds a separate IIS virtual directory (e.g. <http://servername/OnTimeSdk>) to the existing OnTime virtual directory

(e.g. <http://servername/OnTime>). Please make sure you have both virtual directories listed in IIS before continuing.

Once you have installed the OnTime SDK, you need to navigate to the location that it was installed (typically `C:\inetpub\wwwroot\OnTimeSdk`) and open up the Web.Config file in Notepad and locate the “appSettings” part of the file:

```
<appSettings>
  <add key="ConnectionString" value="server=SERVER;database=OnTime;uid=USER;pwd=PASSWORD;"/>
  <add key="SecurityToken" value="{66ACD352-16C0-4485-8498-8C461BE7CE44}"/>
  <add key="WebServicesUser" value="1"/>
  <add key="EnableDataCache" value="False"/>
</appSettings>
```

You need to make sure that you fill out the ConnectionString that points to the Microsoft SQL Server database that OnTime is connecting to. Also for the SecurityToken field you need to generate a new GUID and add it to the file. This security token will be used by SpiraTeam when it connects to the OnTime API. Once you have made the necessary changes, save the file.

13.1.2. Download the OnTime Plug-In

Go to the Inflectra website and open up the page that lists the various downloads available for SpiraTeam (<http://www.inflectra.com/SpiraTeam/Downloads.aspx>). Listed on this page will be the OnTime11 Plug-In for SpiraTeam. Right-click on this link and save the Zip compressed folder to the hard-drive of the server where SpiraTeam is installed.

Open up the compressed folder and extract the `OnTimeDataSync.dll` file and place it in the `C:\Program Files\SpiraTeam\Bin` folder (it may be SpiraTest or SpiraPlan depending on which product you’re running). This folder should already contain the `DataSyncService.exe` and `DataSyncService.exe.config` files that are the primary files used for managing the data synchronization between SpiraTeam and other systems.

If you do not have an on-premise installation of SpiraTeam, but instead are using a hosted subscription provided by Inflectra (or a third party company) you will not have access to the DataSyncService background service. In such situations, you should use the Desktop DataSync application instead. This application is described in Appendix 1 and can be used instead of the server-based DataSyncService.

13.1.3. Configuring the Service

To configure the integration service, please open up the `DataSyncService.exe.config` file located in `C:\Program Files\SpiraTeam\Bin` with a text editor such as Notepad. Once open, it should look like:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <sectionGroup name="applicationSettings" type="System.Configuration.ApplicationSettingsGroup, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" >
      <section name="Inflectra.SpiraTest.DataSyncService.Properties.Settings"
type="System.Configuration.ClientSettingsSection, System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" requirePermission="false" />
    </sectionGroup>
  </configSections>
  <applicationSettings>
    <Inflectra.SpiraTest.DataSyncService.Properties.Settings>
      <setting name="PollingInterval" serializeAs="String">
        <value>600000</value>
      </setting>
      <setting name="WebServiceUrl" serializeAs="String">
        <value>http://localhost/SpiraTeam</value>
      </setting>
      <setting name="Login" serializeAs="String">
```

```

    <value>fredbloggs</value>
  </setting>
  <setting name="Password" serializeAs="String">
    <value>fredbloggs</value>
  </setting>
  <setting name="EventLogSource" serializeAs="String">
    <value>SpiraTeam Data Sync Service</value>
  </setting>
  <setting name="TraceLogging" serializeAs="String">
    <value>False</value>
  </setting>
</Inflectra.SpiraTest.DataSyncService.Properties.Settings>
</applicationSettings>
</configuration>

```

The sections that need to be verified and possibly changed are marked in yellow above. You need to check the following information:

- The polling interval allows you to specify how frequently the data-synchronization service will ask SpiraTeam and the external system for new data updates. The value is specified in milliseconds and we recommend a value no smaller than 5 minutes (i.e. 300,000ms). The larger the number, the longer it will take for data to be synchronized, but the lower the network and server overhead.
- The base URL to your instance SpiraTeam. It is typically of the form <http://<server name>/SpiraTeam>. Make sure that when you enter this URL on a browser on the server itself, the application login page appears.
- A valid login name and password to your instance of SpiraTeam. This user needs to be a member of the project(s) that will be synchronized with OnTime and needs to have at least **Incident create/modify/view** permissions and **Release create/modify/view** permissions in these projects.

Once you have made these changes, save the file and proceed to the next stage.

13.1.4. Configuring the Plug-In

The next step is to configure the plug-in within SpiraTeam so that the system knows how to access the OnTime server. To start the configuration, please open up SpiraTeam in a web browser, log in using a valid account that has System-Administration level privileges and click on the System > Data Synchronization administration option from the left-hand navigation:

Data Synchronization [\(Refresh\)](#)

SpiraTeam is able to communicate with a variety of external systems and applications using its background data synchronization service and library of available plug-ins.

This page allows you to view, add and modify the list of plug-ins, make changes to their configuration, and verify the status of the most recent synchronization. In addition, you can force a particular plug-in to perform a complete data-synchronization by resetting its date of last sync:

Plug-In	Data Mapping	Last Sync Date	Status	Operations
OnTimeDataSync	> View Project Mappings	3/22/2010 11:20:02 PM	Success	> Force Re-Sync Edit Delete

[➤ Add](#)

This screen lists all the plug-ins already configured in the system. Depending on whether you chose the option to include sample data in your installation or not, you will see either an empty screen or a list of sample data-synchronization plug-ins.

If you already see an entry for **OnTimeDataSync** you should click on its “Edit” link. If you don’t see such an entry in the list, please click on the [Add] button instead. In either case you will be taken to the following screen where you can enter or modify the OnTime Data-Synchronization plug-in:

Edit Plug-In | OnTimeDataSync
[<< Back to Data Synchronization Home](#)

Please enter/edit the following information for the plug-in. The exact information that needs to be entered is specific to each plug-in, and you should refer to the documentation for the plug-in when entering/changing it:

Name*:

Description:

Connection Info*:

Login*:

Password:

Time Offset*:

Auto-Map Users

Custom 01:

Custom 02:

Custom 03:

Custom 04:

Custom 05:

You need to fill out the following fields for the OnTime Plug-in to operate correctly:

- **Name** – this needs to be set to **OnTimeDataSync**. This needs to match the name of the plug-in DLL assembly that was copied into the C:\Program Files\SpiraTeam\Bin folder (minus the .dll file extension). If you renamed the OnTimeDataSync.dll file for any reason, then you need to change the name here to match.
- **Description** – this should be set to a description of the plug-in. This is an optional field that is used for documentation purposes and is not actually used by the system.
- **Connection Info** – this should be the full URL to the OnTime SDK. This is typically something like: <http://<OnTime server name>/OnTimeSdk>. You may need to check in the IIS Management Console of the OnTime server to verify the virtual directory name.
- **Login** – this should be set to the GUID that you specified in the Web.Config file in step 13.1.1 above.
- **Password** – this should be left blank as it's not used by the OnTime data-sync plug-in.
- **Time Offset** – normally this should be set to zero, but if you find that defects being changed in OnTime are not being updated in SpiraTeam, try increasing the value as this will tell the data-synchronization plug-in to add on the time offset (in hours) when comparing date-time stamps. Also if your OnTime installation is running on a server set to a different time-zone, then you should add in the number of hours difference between the servers' time-zones here.
- **Auto-Map Users** – this is not currently used by the OnTime data-sync plug-in and can be ignored.
- **Custom 01 – 05** – these are not currently used by the OnTime data-sync plug-in and can be left blank.

13.2. Configuring the Data Mapping

Next, you need to configure the data mapping between SpiraTeam and OnTime. This allows the various projects, users, releases, incident statuses, priorities, severities and custom property values used in the two applications to be related to each other. This is important, as without a correct mapping, there is no

way for the integration service to know that an “Open” incident in SpiraTeam is the same as an “Open” defect in OnTime (for example).

The following mapping information needs to be setup in SpiraTeam:

- The mapping of the project identifiers for the projects that need to be synchronized
- The mapping of users in the system
- The mapping of releases in the system
- The mapping of the various standard fields in the system
- The mapping of the various custom properties in the system

Each of these is explained in turn below:

13.2.1. Configuring the Project Mapping

From the data synchronization administration page, you need to click on the “View Project Mappings” hyperlink next to the OnTime plug-in name. This will take you to the data-mapping home page for the currently selected project:

OnTimeDataSync Project Data Mapping | Sample OnTime Project [\(Change Project\)](#)

[<< Back to Data Synchronization Home](#)

To configure this project for data mapping with the 'OnTimeDataSync' plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name: Sample OnTime Project

External Key:*

Active:* ▼

Artifact Field Mapping

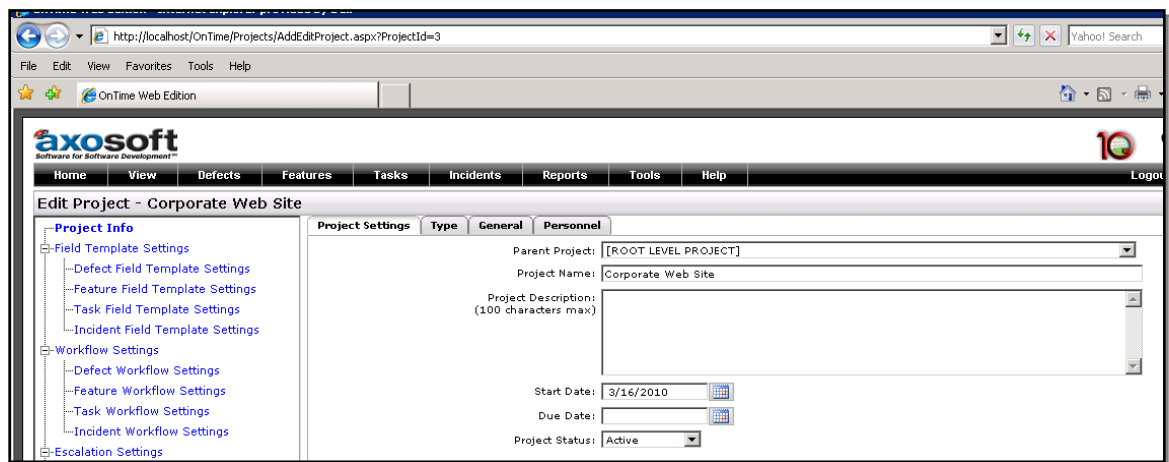
Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Incident	Priority Severity Status Type	ReplicationProcedures OnTime Operating System OnTime Environment
Release		
Task	Priority Status	

If the project name does not match the name of the project you want to configure the data-mapping for, click on the “(Change Project)” hyperlink to change the current project.

To enable this project for data-synchronization with OnTime, you need to enter:

- **External Key** – This should be set to the numeric ID of the project token in OnTime. You can find this in OnTime by selecting the project in the project explorer inside OnTime and then clicking the Edit icon. This brings up the project details screen:



The ID of the project is the value listed in the browser URL directly after the “ProjectId=” text. In the example above, the project ID would be 3.

- **Active Flag** – Set this to ‘Yes’ so that SpiraTeam knows that you want to synchronize data for this project. Once the project has been completed, setting the value to “No” will stop data synchronization, reducing network utilization.

Click [Update] to confirm these settings. Once you have enabled the project for data-synchronization, you can now enter the other data mapping values outlined below.

Note: Once you have successfully configured the project, when creating a new project, you should choose the option to “Create Project from Existing Project” rather than “Use Default Template” so that all the project mappings get copied across to the new project.

13.2.2. Configuring the User Mapping

To configure the mapping of users in the two systems, you need to go to Administration > Users > View Edit Users, which will bring up the list of users in the system. Then click on the “Edit” button for a particular user that will be editing defects in OnTime:

Add/Edit User | Fred Bloggs

Please review the information listed below and make any changes if necessary. Once you have finished, click [Update/Insert] to confirm them.

First Name*:

Middle Initial:

Last Name*:

User Name*:

Email Address*:

System Administrator*:

Active User*:

OnTimeDataSync ID:

You will notice that below the Active flag for the user is a list of all the configured data-synchronization plug-ins. In the text box next to the OnTime Data-Sync plug-in you need to enter the **Login ID** for this username in OnTime. This will allow the data-synchronization plug-in to know which user in SpiraTeam

match which equivalent user in OnTime. Click [Update] once you've entered the appropriate login name. You should now repeat for the other users who will be active in both systems.

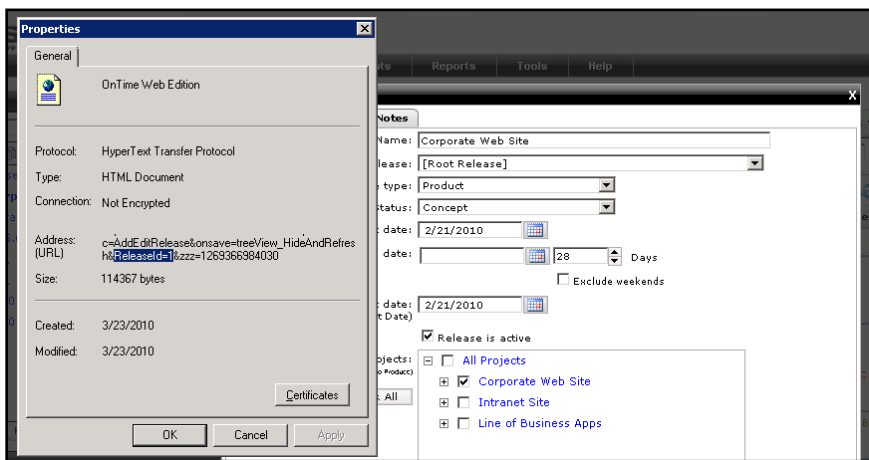
13.2.3. Configuring the Release Mapping

When the data-synchronization service runs, when it comes across a release/iteration in SpiraTeam that it has not seen before, it will create a corresponding Release in OnTime. Similarly if it comes across a new Release in OnTime that it has not seen before, it will create a new Release in SpiraTeam. Therefore when using both systems together, it is recommended that you only enter new Releases in one system and let the data-synchronization service add them to the other system.

However you may start out with the situation where you already have pre-existing Releases in both systems that you need to associate in the data-mapping. If you don't do this, you may find that duplicates get created when you first enable the data-synchronization service. Therefore for any Releases/Iterations that already exist in BOTH systems please navigate to Planning > Releases and click on the Release/Iteration in question. Make sure you have the 'Overview' tab visible and expand the "Details" section of the release/iteration:

In addition to the standard fields and custom properties configured for Releases, you will see an additional text property called "OnTimeDataSync ID" that is used to store the mapped external identifier for the equivalent Version in OnTime. You need to locate the ID of the equivalent version in OnTime, enter it into this text-box and click [Save]. You should now repeat for all the other pre-existing releases.

*Note: The OnTime ID can be found by looking at the URL inside OnTime when choosing to Edit the release in question. The URL will include the section: **ReleaseId=X** where X is the numeric ID of the version inside OnTime:*



13.2.4. Configuring the Standard Field Mapping

Now that the projects, user and releases have been mapped correctly, we need to configure the standard incident fields. To do this, go to Administration > System > Data Synchronization and click on the “View Project Mappings” for the OnTimeDataSync plug-in entry:

OnTimeDataSync Project Data Mapping | Sample OnTime Project [\(Change Project\)](#)

[<< Back to Data Synchronization Home](#)

To configure this project for data mapping with the 'OnTimeDataSync' plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name: Sample OnTime Project

External Key:*

Active:* ▼

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Incident	Priority Severity Status Type	ReplicationProcedures OnTime Operating System OnTime Environment
Release		
Task	Priority Status	

From this screen, you need to click on Priority, Severity and Status in turn to configure their values (OnTime doesn't support different defect types):

a) Incident Status

Click on the “Status” hyperlink under Incident Standard Fields to bring up the Incident status mapping configuration screen:

Field Value	Active	External Key	Primary
Assigned	Yes	<input type="text" value="Assigned"/>	<input type="button" value="Yes"/> ▼
Closed	Yes	<input type="text" value="Closed"/>	<input type="button" value="Yes"/> ▼
Duplicate	Yes	<input type="text" value="On Hold"/>	<input type="button" value="No"/> ▼
New	Yes	<input type="text" value="Open"/>	<input type="button" value="No"/> ▼
Not Reproducible	Yes	<input type="text" value="On Hold"/>	<input type="button" value="Yes"/> ▼
Open	Yes	<input type="text" value="Open"/>	<input type="button" value="Yes"/> ▼
Reopen	Yes	<input type="text" value="Open"/>	<input type="button" value="No"/> ▼
Resolved	Yes	<input type="text" value="Closed"/>	<input type="button" value="No"/> ▼

The table lists each of the incident statuses available in SpiraTeam and provides you with the ability to enter the matching OnTime defect status names for each one. You can map multiple SpiraTeam fields to the same OnTime fields (e.g. New and Open in SpiraTeam are both equivalent to Open in OnTime), in which case only one of the two values can be listed as Primary = Yes as that's the value that's used on the reverse synchronization (from OnTime > SpiraTeam).

We recommend that you always point the New and Open statuses inside SpiraTeam to point to the “Open” status inside OnTime and make Open in SpiraTeam the Primary status of the two. This is recommended so that as new incidents in SpiraTeam get synched over to OnTime, they will get switched to the Open status in OnTime which will then be synched back to “Open” in SpiraTeam. That way you’ll be able to see at a glance which incidents have been synched with OnTime and those that haven’t.

Note: The OnTime external key needs to exactly match the display name of the status inside OnTime. If you change the name of a status in OnTime, you’ll need to update the value in the data-mapping configuration as well.

b) Incident Priority

Click on the “Priority” hyperlink under Incident Standard Fields to bring up the Incident Priority mapping configuration screen:

Field Value	Active	External Key	Primary
1 - Critical	Yes	<input type="text" value="High"/>	No <input type="button" value="v"/>
2 - High	Yes	<input type="text" value="High"/>	Yes <input type="button" value="v"/>
3 - Medium	Yes	<input type="text" value="Medium"/>	Yes <input type="button" value="v"/>
4 - Low	Yes	<input type="text" value="Low"/>	Yes <input type="button" value="v"/>

The table lists each of the incident priorities available in SpiraTeam and provides you with the ability to enter the matching OnTime priority name for each one. You can map multiple SpiraTeam fields to the same OnTime fields, in which case only one of the two values can be listed as Primary = Yes as that’s the value that’s used on the reverse synchronization (from OnTime > SpiraTeam).

Note: The OnTime external key needs to exactly match the display name of the priority inside OnTime. If you change the name of a priority in OnTime, you’ll need to update the value in the data-mapping configuration as well.

c) Incident Severity

Click on the “Severity” hyperlink under Incident Standard Fields to bring up the Incident severity mapping configuration screen:

Field Value	Active	External Key	Primary
1 - Critical	Yes	<input type="text" value="Critical"/>	Yes <input type="button" value="v"/>
2 - High	Yes	<input type="text" value="High Impact"/>	Yes <input type="button" value="v"/>
3 - Medium	Yes	<input type="text" value="Medium Impact"/>	Yes <input type="button" value="v"/>
4 - Low	Yes	<input type="text" value="Low Impact"/>	Yes <input type="button" value="v"/>

The table lists each of the incident severities available in SpiraTeam and provides you with the ability to enter the matching OnTime severity name for each one. You can map multiple SpiraTeam fields to the same OnTime fields, in which case only one of the two values can be listed as Primary = Yes as that’s the value that’s used on the reverse synchronization (from OnTime > SpiraTeam).

Note: The OnTime external key needs to exactly match the display name of the severity inside OnTime. If you change the name of a severity in OnTime, you'll need to update the value in the data-mapping configuration as well.

13.2.5. Configuring the Custom Property Mapping

Now that the various SpiraTeam standard fields have been mapped correctly, we need to configure the custom property mappings. This is used for both custom properties in SpiraTeam that map to custom fields in OnTime and also for custom properties in SpiraTeam that are used to map to standard fields in OnTime (currently only Replication Procedures) that don't exist in SpiraTeam.

From the View/Edit Project Data Mapping screen, you need to click on the name of the Incident Custom Property that you want to add data-mapping information for. We will consider the *three* different types of mapping that you might want to enter:

OnTimeDataSync Project Data Mapping | Sample OnTime Project [\(Change Project\)](#)

[<< Back to Data Synchronization Home](#)

To configure this project for data mapping with the 'OnTimeDataSync' plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name: Sample OnTime Project

External Key:*

Active:* ▼

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact	Standard Fields	Custom Properties
Incident	Priority Severity Status Type	ReplicationProcedures OnTime Operating System OnTime Environment
Release		
Task	Priority Status	

a) Text Custom Properties

Click on the hyperlink of the text custom property under Incident Custom Properties to bring up the custom property mapping configuration screen. For text custom properties there will be no values listed in the lower half of the screen.

Edit Custom Property Data Mapping | Sample OnTime Project [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the OnTimeDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Name: Notes

External Key:

You need to lookup the display name of the custom field in OnTime that matches this custom property in SpiraTeam. Once you have entered the id of the custom field, click [Update].

b) List Custom Properties

Click on the hyperlink of the list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen. For list custom properties there will be a textbox for both the custom field itself and a mapping table for each of the custom property values that need to be mapped:

Edit Custom Property Data Mapping | Sample OnTime Project [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the OnTimeDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Name: OnTime Operating System

External Key:

Custom Value Mapping

Property Value	Active	External Key
Windows 2000	Yes	<input type="text" value="Windows 2000"/>
Windows 2003	Yes	<input type="text" value="Windows 2003"/>
Windows NT	Yes	<input type="text" value="Windows NT"/>
Windows XP	Yes	<input type="text" value="Windows XP"/>

First you need to lookup the display name of the custom field in OnTime that matches this custom property in SpiraTeam. This should be entered in the 'External Key' field below the name of the custom property.

Next for each of the Property Values in the table (in the lower half of the page) you need to enter the full name of the custom field value as specified in OnTime.

c) OnTime's Replication Procedures Field

If you want new defects in OnTime to be loaded with the "replication procedures" standard text field populated, then you will need to fill out this section. You first need to create an incident custom property in SpiraTeam of type 'TEXT' that will be used to store the environment description within SpiraTeam.

Then click on the hyperlink of this new list custom property under Incident Custom Properties to bring up the custom property mapping configuration screen:

Edit Custom Property Data Mapping | Sample OnTime Project [\(Change Project\)](#)

[<< Back to Project Mappings](#)

To configure the data mappings for this custom property, please fill in the OnTimeDataSync external key values for the custom property below. If this is a **custom list** property, you also need to fill in the external key for each of the custom property values as well:

Name: ReplicationProcedures

External Key:

All you need to do on this screen is enter the word “ReplicationProcedures” in the External Key textbox and the data-sync plug-in will know that this custom property is mapped to the built-in Replication Procedures field in OnTime. *Note that there is no space between the words Replication and Procedures!!*

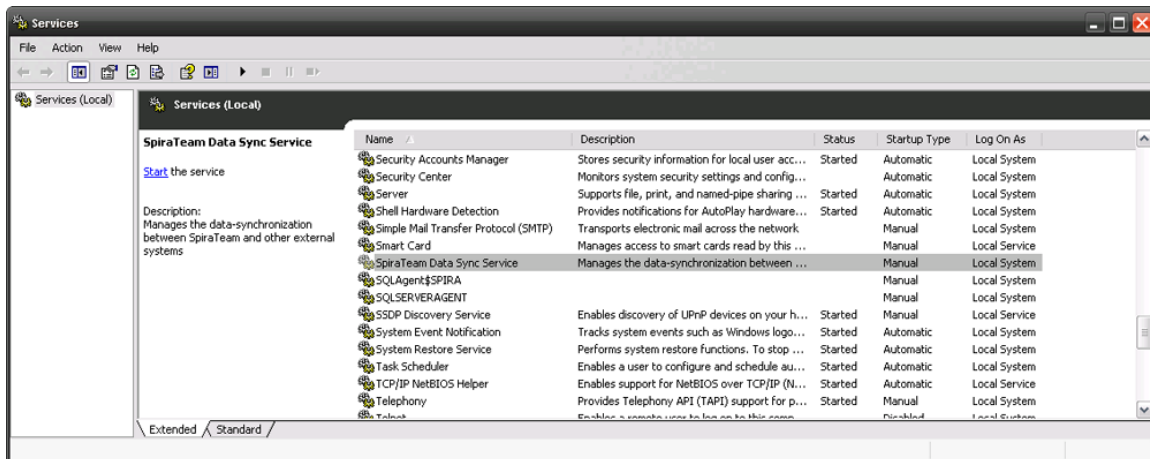
Once you have updated the various mapping sections, you are now ready to start the service.

13.3. Enabling the Data-Synchronization

13.3.1. Starting the Service

When SpiraTeam is installed, a Windows Service – SpiraTeam Data Sync Service – is installed along with the web application. However to avoid wasting system resources, this service is initially set to run manually. To ensure continued synchronization of SpiraTeam with OnTime, we recommend starting the service and setting its startup-type to Automatic.

To make these changes, open up the Windows Control Panel, click on the “Administrative Tools” link, and then choose the Services option. This will bring up the Windows Service control panel:



Click on the ‘SpiraTeam Data Sync Service’ entry and click on the link to start the service. Then right-click the service entry and choose the option to set the startup type to ‘Automatic’. This will ensure that synchronization continues between SpiraTeam and OnTime after a reboot of the server.

13.3.2. Using SpiraTeam with OnTime

Now that the integration service has been configured and the service started, initially any incidents created in SpiraTeam for the specified projects will be imported into OnTime.

At this point we recommend opening the Windows Event Viewer and choosing the Application Log. In this log any error messages raised by the SpiraTeam Data Sync Service will be displayed. If you see any error messages at this point, we recommend immediately stopping the SpiraTeam service and checking the various mapping entries. If you cannot see any defects with the mapping information, we recommend sending a copy of the event log message(s) to Inflectra customer services (support@inflectra.com) who will help you troubleshoot the problem.

To use SpiraTeam with OnTime on an ongoing basis, we recommend the following general processes be followed:

- When running tests in SpiraTeam, defects found should be logged through the ‘Add Incident’ option as normal.
- Once an incident has been created during the running of the test, it will now be populated across into OnTime as a defect. It will be populated with the information captured in SpiraTeam.

- At this point, the incident should not be acted upon inside SpiraTeam, and all data changes to the defect should be made inside OnTime. To enforce this, you can modify the workflows set up in SpiraTeam so that the various fields are marked as inactive for all the incident statuses other than the “New” status. This will allow someone to submit an incident in SpiraTeam, but will prevent them making changes in conflict with OnTime after that point.
- As the defect progresses through the OnTime workflow, changes to the status, priority, severity, and resolution will be updated automatically in SpiraTeam. In essence, SpiraTeam acts as a read-only viewer of these incidents.
- You are now able to perform test coverage and incident reporting inside SpiraTeam using the test cases managed by SpiraTeam and the incidents managed on behalf of SpiraTeam inside OnTime.

14. Using Spira with GitHub

This section outlines how to use SpiraTest, SpiraTeam or SpiraPlan (hereafter referred to as SpiraPlan) in conjunction with the GitHub integrated issue tracker.

► **STOP! Please make sure you have first read the Instructions in Section 1 before proceeding!**

The GitHub issue tracker is a simple and lightweight tool used to track problems with an associated git repository. The built-in integration service enables two-way syncing of new incidents, new comments, statuses, and releases (milestones).

14.1 Configuring the Integration Service

This section outlines how to set up the integration service between GitHub and SpiraPlan. It assumes that you already have a working installation of SpiraPlan and a GitHub repository with an issue tracker. To setup the service, you must be logged into SpiraPlan as a user with System-Administrator level privileges.

Inside SpiraPlan, go to the Administration page and navigate to the Integration > Data Synchronization webpage. Check that you don't already have a Plug-In called "GitHubDataSync", as shown below:

Plug-In

Sample GitHub Data Synchronization GitHubDataSync

If you already have a plug-in called **GitHub Data Sync**, please click on its "edit" button, otherwise please click the "Add" button to create a new plug-in:

GitHub Edit Plug-In

[← Back to Data Synchronization Home](#)

Please enter/edit the following information for the plug-in. The exact information that needs to be entered is specific to each plug-in, and you should refer to the documentation for the plug-in when entering/changing it:

Name:
The physical name of the plugin (e.g. JiraDataSync) that matches the DLL being used

Caption:
The display name of the plugin (e.g. Internal Jira) that can be used to distinguish between multiple instances of the same plugin type

Description:
The long description of the plugin (e.g. 'Connects to the in-house bug-tracker')

Connection Info:
The URL or other connection string used to connect to the external bug-tracker

Login:

Password:
The login and password used to connect to GitHub

You need to fill out the following fields for the GitHub Data Sync plugin to work properly:

- **Name** – This needs to be set to **GitHubDataSync**
- **Caption** – This is the display name of the plug-in, generally something generic like “GitHub” would work, but you should change it if you will be syncing with multiple GitHub projects.
- **Description** – The description of what you’re using the plug-in for. This field is entirely optional and is not used by the system in any way.
- **Connection Info** – The location of your GitHub account, removing the actual repository name. For example, if you have a repository such as <https://github.com/octocat/Hello-World>, you would simply enter “**octocat**” as the connection info. We will enter the repository name later when we setup the project mappings.
- **Login** – Your GitHub username
- **Password** – A GitHub personal access token with the “public_repo” permission. You can create a new one at <https://github.com/settings/tokens>
- **Time Offset** – This should be set to 0, but if you find that changes are not being synced, try increasing the value to tell the plugin to offset timestamps
- **Custom 01** – For on-premise GitHub Enterprise installations only, please enter the name of your server (e.g. <http://myserver>), if left blank, the data synchronization will assume you are using the cloud URL for GitHub (<https://www.github.com>)

The rest of the fields should be left blank. Once all those fields have been filled out, click the “Add” or “Save” button to save your changes.

14.2 Configuring Project Mappings

For this step, please ensure that you are in the SpiraPlan project you would like to sync with GitHub. For this example, the project is called “GitHub Data Sync.”

GitHub Project Data Mapping Library Information System

[← Back to Data Synchronization Home](#) [↻ Change Project](#)

To configure this project for data mapping with the 'GitHub' plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name:* Library Information System

External Key:*

Active:* Yes

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact Type	Standard Fields	Custom Properties
Requirement	Component Importance Status Type	URL Difficulty Classification Notes Review Date Ranking Decimal RFQ Response

Click on the “View Project Mappings” button for GitHub Data Sync. You need to fill out the following fields to sync correctly:

- **External Key** – The name of your GitHub repository. In the example above, where the URL in GitLab was <https://github.com/octocat/Hello-World>, you would simply enter “Hello-World” for this setting.
- **Active** – Set this to yes so that the Data Sync plug-in knows to synchronize with this project.

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact Type	Standard Fields	Custom Properties
Requirement	Component Importance Status Type	URL Difficulty Classification Notes Review Date Ranking Decimal RFQ Response
Test Case	Component Priority Status Type	URL Test Type Review Owner Browser
Incident	Component Priority Severity Status Type	Notes Operating System Web Browser Internal? Ranking Review Date Difficulty Reviewer Decimal

Now click the “Status” button within the “Incident” section to map the Incident statuses together. The purpose of this is so that the GitHub Data Sync plug-in knows what the equivalent status is in GitHub for an incident status in SpiraPlan.

Edit Standard Field Data Mapping GitHub Data Sync

[← Back to Project Mappings](#)
[↻ Change Project](#)

To configure the data mappings for this field, please fill in the external key values for the **Sample GitHub Data Synchronization** plug-in next to each SpiraPlan value.

Artifact Type: **Incident**
 Field Name: **Status**

Status	Active	External Key <small>Sample GitHub Data Synchronization</small>	Primary
Assigned	Yes	<input type="text" value="open"/>	<input checked="" type="checkbox"/> Yes
Closed	Yes	<input type="text" value="closed"/>	<input checked="" type="checkbox"/> Yes
Duplicate	Yes	<input type="text" value="closed"/>	<input type="checkbox"/> No
New	Yes	<input type="text" value="open"/>	<input type="checkbox"/> No
Not Reproducible	Yes	<input type="text" value="closed"/>	<input type="checkbox"/> No
Open	Yes	<input type="text" value="open"/>	<input type="checkbox"/> No
Reopen	Yes	<input type="text" value="open"/>	<input type="checkbox"/> No
Resolved	Yes	<input type="text" value="closed"/>	<input type="checkbox"/> No

Save

You must map every status in the system. Descriptions of the field are below:

- **External Key** – Either **open** or **closed**, which are the only two statuses in GitHub
- **Primary** – You must have exactly one primary key for **open** and one for **closed**. This is what status the plug-in should set the incident in SpiraPlan to when the status in GitHub changes.

Click “Save” and assuming everything was done correctly, the plug-in should work. Start your Data Sync service and verify that issues in GitHub appear inside SpiraPlan. Note that the Data Sync service is not running constantly, so it may take some time for changes to materialize.

Congratulations, you have just integrated your Spira instance with GitHub’s integrated issue tracker!

15. Using Spira with GitLab

This section outlines how to use SpiraTest, SpiraTeam or SpiraPlan (hereafter referred to as SpiraPlan) in conjunction with the GitLab integrated issue tracker.

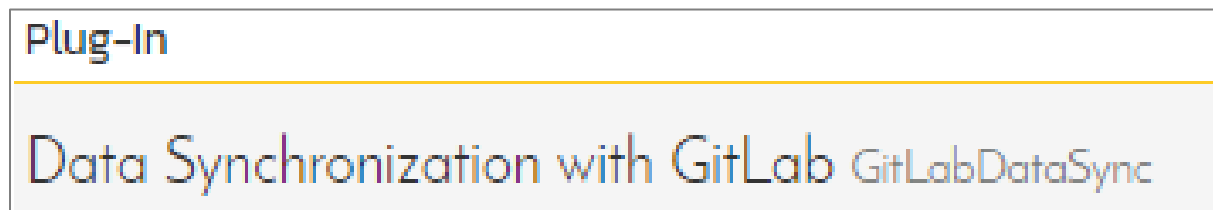
► **STOP! Please make sure you have first read the Instructions in Section 1 before proceeding!**

The GitLab issue tracker is a simple and lightweight tool used to track problems with an associated git repository. The built-in integration service enables two-way syncing of new incidents, new comments, statuses, and releases (milestones).

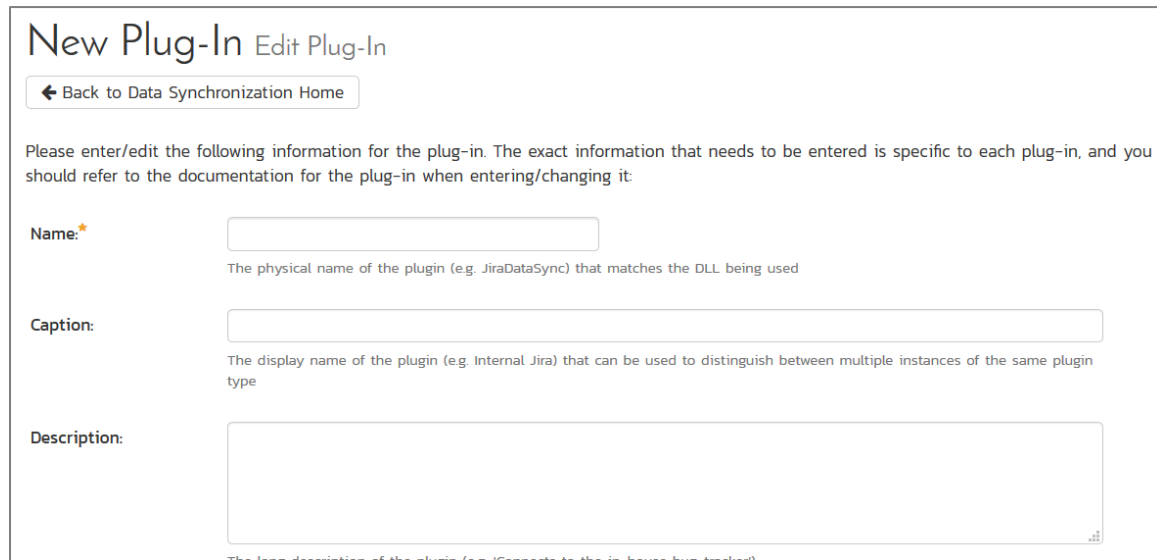
15.1 Configuring the Integration Service

This section outlines how to set up the integration service between GitLab and SpiraPlan. It assumes that you already have a working installation of SpiraPlan and a GitLab repository with an issue tracker. To setup the service, you must be logged into SpiraPlan as a user with System-Administrator level privileges.

Inside SpiraPlan, go to the Administration page and navigate to the Integration > Data Synchronization webpage. Check that you don't already have a Plug-In called "GitLabDataSync", as shown below:



If you already have a plug-in called **GitLabDataSync**, please click on its "edit" button, otherwise please click the "Add" button to create a new plug-in:

A screenshot of the "New Plug-In" configuration page. The page has a title "New Plug-In" and a link "Edit Plug-In". There is a "Back to Data Synchronization Home" button. Below that, there is a paragraph of instructions: "Please enter/edit the following information for the plug-in. The exact information that needs to be entered is specific to each plug-in, and you should refer to the documentation for the plug-in when entering/changing it:". There are three input fields: "Name:" with a small asterisk, "Caption:", and "Description:". Each field has a small text box below it with instructions. The "Name:" field has a small text box below it that says "The physical name of the plugin (e.g. JiraDataSync) that matches the DLL being used". The "Caption:" field has a small text box below it that says "The display name of the plugin (e.g. Internal Jira) that can be used to distinguish between multiple instances of the same plugin type". The "Description:" field has a small text box below it that says "The long description of the plugin (e.g. 'Connects to the in-house bug tracker')".

Now fill out this configuration page as follows:

Name: *	<input type="text" value="GitLabDataSync"/>	The physical name of the plugin (e.g. JiraDataSync) that matches the DLL being used
Caption:	<input type="text" value="Data Synchronization with GitLab"/>	The display name of the plugin (e.g. Internal Jira) that can be used to distinguish between multiple instances of the same plugin type
Description:	<input type="text" value="Synchronizes data with a sample GitLab repository"/>	The long description of the plugin (e.g. 'Connects to the in-house bug-tracker')
Connection Info: *	<input type="text" value="gitlab-examples"/>	The URL or other connection string used to connect to the external bug-tracker
Login: *	<input type="text" value="lennie"/>	
Password:	<input type="text" value="PERSONAL ACCESS TOKEN"/>	The login and password used to connect to GitLab Sync to local in-office instance
Time Offset: *	<input type="text" value="0"/> hours	The number of hours to look back for changes to synchronization. Normally should be set to zero (0) unless you find changes are not being synchronized
Auto-Map Users	<input type="button" value="No"/>	
Custom 01:	<input type="text"/>	

You need to fill out the following fields for the GitLab Data Sync plugin to work properly:

- **Name** – This needs to be set to **GitLabDataSync**
- **Caption** – This is the display name of the plug-in, generally something generic like “GitLab” would work, but you should change it if you will be syncing with multiple GitLab projects.
- **Description** – The description of what you’re using the plug-in for. This field is entirely optional and is not used by the system in any way.
- **Connection Info** – The location of your GitLab account, removing the actual repository name. For example, if you have a repository such as <https://gitlab.com/gitlab-examples/velociraptor>, you would simply enter “gitlab-examples” as the connection info. We will enter the repository name later when we setup the project mappings.
- **Login** – Your GitLab username
- **Password** – A GitLab personal access token with the “api” permission. You can create a new one at https://gitlab.com/profile/personal_access_tokens
- **Time Offset** – This should be set to 0, but if you find that changes are not being synced, try increasing the value to tell the plugin to offset timestamps
- **Custom 01** – For on-premise GitLab installations only, please enter the name of your server (e.g. <http://myserver>), if left blank, the data synchronization will assume you are using the cloud URL for GitLab (<https://www.gitlab.com>)

The rest of the fields should be left blank. Once all those fields have been filled out, click the “Add” or “Save” button to save your changes.

15.2 Configuring Project Mappings

For this step, please ensure that you are in the SpiraPlan project you would like to sync with GitLab. For this example, the project is called “GitLab Data Sync.”

Click on the “View Project Mappings” button for GitLab Data Sync. You need to fill out the following fields to sync correctly:

- **External Key** – The name of your GitLab repository. In the example above, where the URL in GitLab was <https://gitlab.com/gitlab-examples/velociraptor>, you would simply enter “velociraptor” for this setting.
- **Active** – Set this to yes so that the Data Sync plug-in knows to synchronize with this project.

GitLab Project Data Mapping Sample Application One

[← Back to Data Synchronization Home](#) [↻ Change Project](#)

To configure this project for data mapping with the 'GitLab' plug-in, you need to specify the key/id of the project in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Project Name:*

External Key:*

Active:* Yes

Artifact Field Mapping

Please click on each of the links below to configure the field-level data mapping for each of the types of artifact that you will be synchronizing in this project:

Artifact Type	Standard Fields	Custom Properties
Requirement	Component Importance Status Type	
Test Case	Component Priority Status Type	
Incident	Component Priority Severity Status Type	

15.2.1. Configuring the Incident Status Mapping

Now click the “Status” button within the “Incident” section to map the Incident statuses together. The purpose of this is so that the GitLab Data Sync plug-in knows what the equivalent status is in GitLab for an incident status in SpiraPlan.

Edit Standard Field Data Mapping Sample Application One

[← Back to Project Mappings](#) [↻ \(Change Project\)](#)

To configure the data mappings for this field, please fill in the external key values for the **GitLabDataSync** plug-in next to each SpiraTest value:

Artefact Type: Incident
Field Name: Status

Status	Active	External Key	GitLabDataSync	Primary
Assigned	Yes	<input type="text" value="opened"/>	<input type="checkbox"/>	No
Closed	Yes	<input type="text" value="closed"/>	<input checked="" type="checkbox"/>	Yes
Duplicate	Yes	<input type="text" value="closed"/>	<input type="checkbox"/>	No
New	Yes	<input type="text" value="opened"/>	<input checked="" type="checkbox"/>	Yes
Not Reproducible	Yes	<input type="text" value="closed"/>	<input type="checkbox"/>	No
Open	Yes	<input type="text" value="opened"/>	<input type="checkbox"/>	No
Reopen	No	<input type="text" value="opened"/>	<input type="checkbox"/>	No
Resolved	Yes	<input type="text" value="closed"/>	<input type="checkbox"/>	No

[Save](#)

You must map every status in the system. Descriptions of the field are below:

- **External Key** – Either **opened** or **closed**, which are the only two statuses in GitLab
- **Primary** – You must have exactly one primary key for **opened** and one for **closed**. This is what status the plug-in should set the incident in SpiraPlan to when the status in GitLab changes.

15.2.2. Configuring the User Mapping

To configure the mapping of users in the two systems, you need to go to Administration > Users > View Edit Users, which will bring up the list of users in the system. Then click on the “Edit” button for a particular user that will be editing issues in GitLab:

Membership and Mapping

[Project Membership](#) [Data Mapping](#)

Bugzilla ID:

FogBugz ID:

GitLab ID:

Jira ID:

Microsoft TFS ID:

[Save](#) [Cancel](#)

Click on the ‘Data Mapping’ tab to list all the configured data-synchronization plug-ins for this user. In the text box next to the GitLab Data-Sync plug-in you need to enter the login for this username in GitLab. This will allow the data-synchronization plug-in to know which user in SpiraTeam match which equivalent user in GitLab. Click [Save] once you’ve entered the appropriate login name. You should now repeat for the other users who will be active in both systems.

If you have set the “Auto-Map Users” option in the GitLab plugin, you can skip this section completely.

15.2.3. Configuring the Release Mapping

When the data-synchronization service runs, when it comes across a release/iteration in SpiraTeam that it has not seen before, it will create a corresponding “Milestone” in GitLab. Similarly, if it comes across a new Milestone in GitLab that it has not seen before, it will create a new Release in SpiraTeam. Therefore, when using both systems together, it is recommended that you only enter new Releases/Milestones in one system and let the data-synchronization service add them to the other system.

However, you may start out with the situation where you already have pre-existing Releases / Milestones in both systems that you need to associate in the data-mapping. If you don't do this, you may find that duplicates get created when you first enable the data-synchronization service. Therefore, for any Releases/Iterations that already exist in BOTH systems please navigate to Planning > Releases and click on the Release/Iteration in question. Make sure you have the ‘Overview’ tab visible and expand the “Details” section of the release/iteration:

Milestone 1

1.0.0.0.0000 [RL:20] Type: Iteration Status: In Progress

Overview Incidents * Test Cases Test Runs Attachments History

People		Properties	
Creator:	System Administrator	Version #:	1.0.0.0.0000
Owner:	-- None --	GitLab ID:	1

In addition to the standard fields and custom properties, you will see an additional text property called “**GitLab ID**” that is used to store the mapped external identifier for the equivalent Milestone in GitLab. You need to locate the ID of the equivalent version in GitLab, enter it into this text-box and click [Save]. You should now repeat for all the other pre-existing releases.

15.3. Using the Data Synchronization

Assuming everything was done correctly, the plug-in should start working. Start your Data Sync service and verify that issues in GitLab appear inside SpiraPlan. Note that the Data Sync service is not running constantly, so it may take some time for changes to materialize.

Congratulations, you have just integrated your Spira instance with GitLab's integrated issue tracker!

16. Using Spira with ServiceNow

This section outlines how to use SpiraTest, SpiraTeam or SpiraPlan (hereafter referred to as SpiraPlan) in conjunction with tables in ServiceNow.

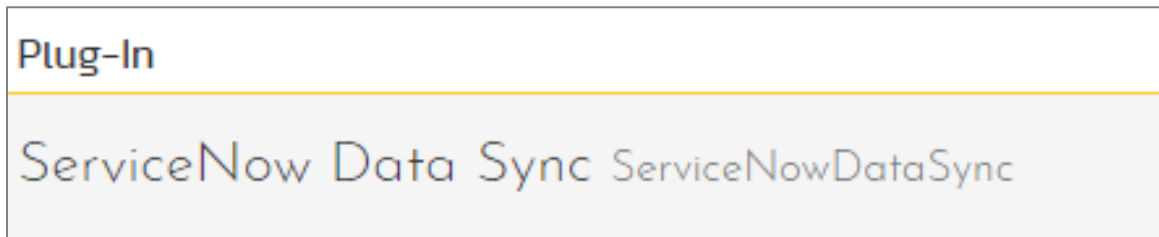
► **STOP! Please make sure you have first read the Instructions in Section 1 before proceeding!**

ServiceNow tables are a highly configurable system that can now be used in conjunction with SpiraPlan. This integration service enables two-way syncing of new incidents and requirements with any table in ServiceNow and syncing of updates from ServiceNow into SpiraPlan. Attachments will only be synced with creation.

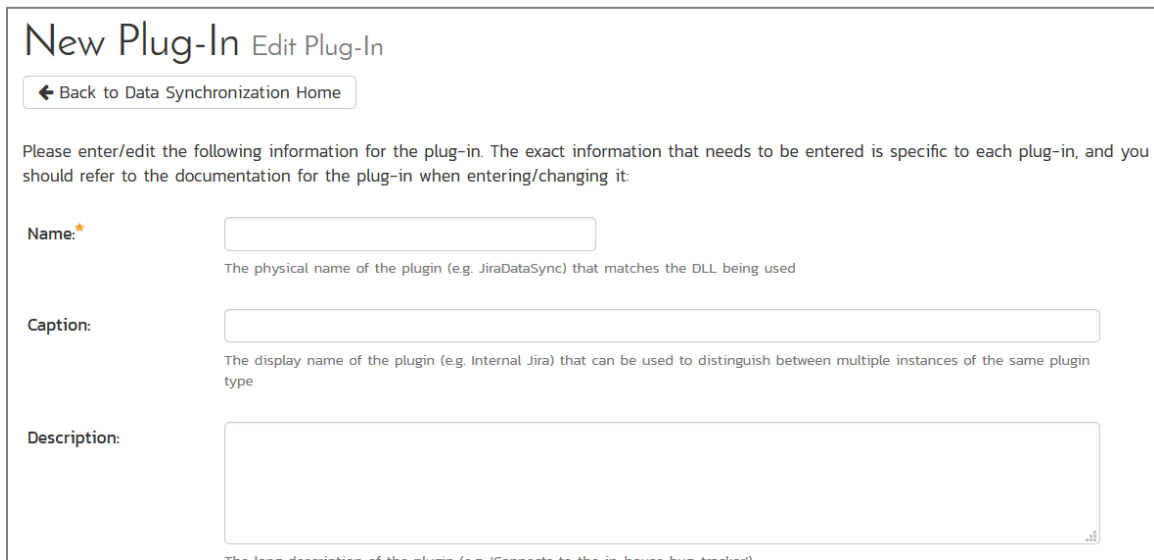
16.1 Configuring the Integration Service

This section outlines how to set up the integration service between ServiceNow and SpiraPlan. It assumes that you already have a working installation of SpiraPlan and appropriate ServiceNow tables. To setup the service, you must be logged into SpiraPlan as a user with System-Administrator level privileges.

Inside SpiraPlan, go to the Administration page and navigate to the Integration > Data Synchronization webpage. Check that you don't already have a Plug-In called "ServiceNowDataSync", as shown below:



If you already have a plug-in called **ServiceNowDataSync**, please click on its "edit" button, otherwise please click the "Add" button to create a new plug-in:

A screenshot of a web form titled "New Plug-In" with a sub-header "Edit Plug-In". At the top left is a button "← Back to Data Synchronization Home". Below is a paragraph of instructions: "Please enter/edit the following information for the plug-in. The exact information that needs to be entered is specific to each plug-in, and you should refer to the documentation for the plug-in when entering/changing it:". There are three input fields: 1. "Name:" with a red asterisk, a text box, and a sub-note: "The physical name of the plugin (e.g. JiraDataSync) that matches the DLL being used". 2. "Caption:" with a text box and a sub-note: "The display name of the plugin (e.g. Internal Jira) that can be used to distinguish between multiple instances of the same plugin type". 3. "Description:" with a large text area and a sub-note: "The long description of the plugin (e.g. 'Connects to the in-house bug-tracker')".

Now fill out this configuration page as follows:

Name:
The physical name of the plugin (e.g. JiraDataSync) that matches the DLL being used

Caption:
The display name of the plugin (e.g. Internal Jira) that can be used to distinguish between multiple instances of the same plugin type

Description:
The long description of the plugin (e.g. 'Connects to the in-house bug-tracker')

Connection Info:
The URL or other connection string used to connect to the external bug-tracker

Active: Yes

Login:

Password:
The login and password used to connect to New Plug-In

Time Offset: hours
The number of hours to look back for changes to synchronization. Normally should be set to zero (0) unless you find changes are not being synchronized

Auto-Map Users: Yes

You need to fill out the following fields for the ServiceNow Data Sync plugin to work properly:

- **Name** – This needs to be set to **ServiceNowDataSync**
- **Caption** – This is the display name of the plug-in, generally something generic like “ServiceNow” will work.
- **Description** – The description of what you’re using the plug-in for. This field is entirely optional and is not used by the system in any way.
- **Connection Info** – The location of your ServiceNow account. For example, if you’re on a dev instance, your connection info should be `https://devxxxx.service-now.com`
- **Login** – Your ServiceNow username
- **Password** – Your ServiceNow password
- **Time Offset** – Set this to how many hours *ahead* UTC is, so for EDT (UTC-4), you would put in *positive* 4.
- **Auto-Map Users** – Set to yes if you would like the plugin to map users one-to-one by checking first & last names. Set to no if you would like to map users manually.
- **Custom 01** – The name of the table you would like to sync with incidents in SpiraPlan. This can be found in the *name* field in the table definition within ServiceNow Studio.
- **Custom 02** – The name of the table column in ServiceNow which will decide which project an incident is created in. This can be found in ServiceNow Studio under the column name.
- **Custom 03** – The name of the table you would like to sync with requirements in SpiraPlan. This can be found in the *name* field in the table definition within ServiceNow Studio. This *cannot* be the same as the table in Custom 01.
- **Custom 04** - The name of the table column in ServiceNow which will decide which project a requirement is created in. This can be found in ServiceNow Studio under the column name.

Custom 01:	<input type="text" value="x_*****_spira_incident_table"/>
Custom 02:	<input type="text" value="spira_product"/>
Custom 03:	<input type="text" value="x_*****_spira_requirement_table"/>
Custom 04:	<input type="text" value="spira_product"/>
Custom 05:	<input type="text"/>
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

Custom 05 should be left blank. Once all those fields have been filled out, click the “Add” or “Save” button to save your changes.

16.2 Configuring Project Mappings

For this step, please ensure that you are in a SpiraPlan project you would like to sync with ServiceNow. For this example, the project is called “ServiceNowDataSync Test.”

Click on the “View Project Mappings” button for the ServiceNow Data Sync. You need to fill out the following fields to sync correctly:

- **External Key** – The value of the property in Custom 02/04 you would like to map with this project
- **Active** – Set this to yes so that the Data Sync plug-in knows to synchronize with this project.

ServiceNow Data Sync Product Data Mapping [ServiceNowDataSync Test](#)

[← Back to Data Synchronization Home](#)

To configure this product for data mapping with the ‘ServiceNow Data Sync’ plug-in, you need to specify the key/id of the product in the external system, and then click on the links below to actually configure the different fields that need to be mapped for the synchronization to operate:

Product Name: *	ServiceNowDataSync Test
External Key: *	<input type="text" value="ServiceNowDataSync Test"/>
Active: *	<input checked="" type="checkbox"/> Yes

The spira_product example above is a ‘choice’ type with the choices shown below: Type in the ‘Label’ field of the choice, *not* the ‘Value.’

Choices		New	Search	Label	Search
	Choices				
		≡ Label			≡ Value
<input type="checkbox"/>		<u>Barebones Product</u>			BP
<input type="checkbox"/>		<u>Library Information System</u>			LIS
<input type="checkbox"/>		<u>ServiceNowDataSync Test</u>			ST
+	<i>Insert a new row...</i>				

A brief note about field syncing in ServiceNow: The sheer configurability of ServiceNow meant some assumptions were made in the designing of this data sync. Specific column names are mapped to their counterparts in SpiraPlan based on the list below. If a field is not present in ServiceNow, it will simply not be synced.

SpiraPlan Field	ServiceNow Column Name
Name	name/short_description [both will work]
Description	description
Priority/Importance	priority
Author/Detected By	opened_by
Owner	assigned_to
Status	state

16.2.1. Configuring the Incident/Requirement Status Mappings

Now click the “Status” button within the “Incident” section to map the Incident statuses together. The purpose of this is so that the ServiceNow Data Sync plug-in knows what the equivalent status is in ServiceNow for an incident status in SpiraPlan. The process is identical for Requirement statuses, so repeat these steps with Requirement -> Status instead if you are also/only syncing requirements.

If you don't have a status equivalent in your table, you can ignore this section.

Edit Standard Field Data Mapping [ServiceNowDataSync Test](#)

[← Back to Product Mappings](#)

To configure the data mappings for this field, please fill in the external key values for the **ServiceNow Data Sync** plug-in next to each SpiraPlan value:

Artifact Type: **Incident**

Field Name: **Status**

Status	Active	External Key ServiceNow Data Sync	Primary
Assigned	Yes	<input type="text" value="Assigned"/>	<input checked="" type="checkbox"/>
Closed	Yes	<input type="text" value="Closed"/>	<input checked="" type="checkbox"/>
Duplicate	Yes	<input type="text" value="Duplicate"/>	<input checked="" type="checkbox"/>
New	Yes	<input type="text" value="New"/>	<input checked="" type="checkbox"/>
Not Reproducible	Yes	<input type="text" value="Not Reproducible"/>	<input checked="" type="checkbox"/>
Open	Yes	<input type="text" value="Open"/>	<input checked="" type="checkbox"/>
Reopen	Yes	<input type="text" value="Reopen"/>	<input checked="" type="checkbox"/>
Resolved	Yes	<input type="text" value="Resolved"/>	<input checked="" type="checkbox"/>

You must map every status in SpiraPlan to ServiceNow. Descriptions of the field are below:

- **External Key** – If state is a dropdown in ServiceNow, it's the 'Label' (*not* 'Value') of the choice, which is also what is shown in the ServiceNow UI. If state is a string in ServiceNow, just write in the value of the string to be mapped to the SpiraPlan status. Please take care to **match it exactly** (case, spaces, etc)
- **Primary** – You must have exactly one primary key for each ServiceNow status. This is what status the plug-in should set the incident in SpiraPlan to when the status in ServiceNow changes. This is only used if there are more options in SpiraPlan than ServiceNow.

Here are the corresponding statuses in ServiceNow

Choices	New	Search	Label	Value
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="text"/>	<input type="text" value="Assigned"/>	A
<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text" value="Closed"/>	C
<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text" value="Duplicate"/>	D
<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text" value="New"/>	N
<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text" value="Not Reproducible"/>	NR
<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text" value="Open"/>	O
<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text" value="Reopen"/>	R1
<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text" value="Resolved"/>	R2

16.2.2. Configuring the Priority/Importance Mapping

Now click the “Priority” button within the “Incident” section to map incident priorities. This will tell the ServiceNow Data Sync plug-in which priorities in ServiceNow map to those in SpiraPlan. The process is identical for Requirement importance, so repeat these steps with Requirement -> Importance instead if you are also/only syncing requirements.

If you don’t have a priority equivalent in your table, you can ignore this section.

Edit Standard Field Data Mapping [ServiceNowDataSync Test](#)

[← Back to Product Mappings](#)

To configure the data mappings for this field, please fill in the external key values for the ServiceNow Data Sync plug-in next to each SpiraPlan value:

Artifact Type: Incident
Field Name: Priority

Priority	Active	External Key ServiceNow Data Sync	Primary
1 - Critical	Yes	<input type="text" value="1 - Critical"/>	<input checked="" type="checkbox"/>
2 - High	Yes	<input type="text" value="2 - High"/>	<input checked="" type="checkbox"/>
3 - Medium	Yes	<input type="text" value="3 - Medium"/>	<input checked="" type="checkbox"/>
4 - Low	Yes	<input type="text" value="4 - Low"/>	<input checked="" type="checkbox"/>
5 - Planning	Yes	<input type="text" value="5 - Planning"/>	<input checked="" type="checkbox"/>

[Save](#)

You must map every priority/importance in SpiraPlan to ServiceNow. Descriptions of the field are below:

- **External Key** – If state is a dropdown in ServiceNow, it’s the ‘Label’ (*not* ‘Value’) of the choice, which is also what is shown in the ServiceNow UI. If state is a string in ServiceNow, just write in the value of the string to be mapped to the SpiraPlan status. Please take care to **match it exactly** (case, spaces, etc.)
- **Primary** – You must have exactly one primary key for each ServiceNow priority. This is what priority the plug-in should set the incident in SpiraPlan to when the priority in ServiceNow changes. This is only used if there are more options in SpiraPlan than ServiceNow.

Here are the corresponding priorities in ServiceNow:

☰ Choices New Search Label ▼ Search			
🔍 Choices			
⚙️	🔍	☰ Label	☰ Value
<input type="checkbox"/>	i	<u>1 - Critical</u>	1
<input type="checkbox"/>	i	<u>2 - High</u>	2
<input type="checkbox"/>	i	<u>3 - Medium</u>	3
<input type="checkbox"/>	i	<u>4 - Low</u>	4
<input type="checkbox"/>	i	<u>5 - Planning</u>	5

16.2.3. Cloning ServiceNow Fields

Due to some of the assumptions made in the creation of this integration, it is often necessary to create a hidden compatibility layer between the fields you use in your ServiceNow table and those recognized by the SpiraPlan data sync. This section will lay out how to create 'hidden' fields that will be kept in sync with those in SpiraPlan.

1. Create the field you would like to clone [for this example I will keep "Complete Notes" (visible by ServiceNow users) in sync with "Description" (for syncing with SpiraPlan)]
2. A brief discussion about the requirements for this to work
 - a. Both fields must be of the same type in ServiceNow
 - b. If you have a list or choice, the "Value" of each choice must map one-to-one between the hidden 'Spira' field and the visible ServiceNow field
3. These steps will show how to store description in complete notes when Spira creates an artifact in ServiceNow
 - a. If Studio is not open already, open System Applications > Studio, then select the application the tables are being synced with
 - b. In the top left, click "Create Application File" and select Server Development > Business Rule
 - c. Create a new Business Rule and assign it to the table you are working with [for this example I will use Spira Test Table]
 - d. Name it something helpful [like "description -> complete notes"]
 - e. Under *Where to run*, check the Insert box
 - f. Add a Filter Condition such that the field you're cloning with the Spira field is empty:

Filter Conditions

Add Filter Condition Add "OR" Clause

Complete Notes ▼ is empty ▼ AND OR X

- g. Under *Actions*, where it says "—choose field —" set it to the field you would like to be populated ["Complete Notes" in this example]
- h. To the right of the field name, set "To" to "Same as" and select the Spira field ["Description" in this example]

Set field values

Complete Notes ▼ Same as ▼ as Description ▼

-- choose field -- ▼ To -- value --

4. These steps will show you how to store complete notes in description when the user updates complete notes
 - a. In the top left, click "Create Application File" and select Server Development > Business Rule
 - b. Create a new Business Rule and assign it to the table you are working with [for this example I will use Spira Test Table]
 - c. Name it something helpful [like "[update] complete notes -> description"]
 - d. Under *Where to run*, check the Update box
 - e. Under *Actions*, where it says "—choose field —" set it to the Spira field you would like to be synced ["Description" in this example]
 - f. To the right of the field name, set "To" to "Same as" and select the ServiceNow field editable by the user ["Complete Notes" in this example]

Set field values

Description ▼ Same as ▼ as Complete Notes ▼

-- choose field -- ▼ To -- value --

5. These steps will show you how to store complete notes in description when the user creates a new artifact in ServiceNow
 - a. In the top left, click “Create Application File” and select Server Development > Business Rule
 - b. Create a new Business Rule and assign it to the table you are working with [for this example I will use Spira Test Table]
 - c. Name it something helpful [like “[creation] complete notes -> description”]
 - d. Under *Where to run*, check the Insert box
 - e. Add a Filter Condition such that the Spira field is empty:

Filter Conditions

Add Filter Condition Add "OR" Clause

Description ▼ is empty ▼ AND OR X

- f. Under *Actions*, where it says “—choose field—” set it to the Spira field you would like to be synced [“Description” in this example]
- g. To the right of the field name, set “To” to “Same as” and select the ServiceNow field editable by the user [“Complete Notes” in this example]

Set field values

Description ▼ Same as ▼ as Complete Notes ▼

-- choose field -- ▼ To -- value --

16.2.4. Configuring Custom Properties

This section assumes the custom properties in SpiraPlan and ServiceNow are of the same type (integer -> integer, SingleSelect -> SingleSelect, etc.) Custom property syncing **will not** work otherwise. This applies to both requirement and incident custom properties.

Click on a custom property mapping for a property you would like to sync. For the “External Key” right below the “Name” field put the column name (*not* the column label), so for the Operating System field in ServiceNow, you would put in “operating_system” No extra work is required for user (sys_user references in ServiceNow), text, integer, or date fields.

If your custom property is a single-select list (choice in ServiceNow), for each custom value, put in the “Label” (*not* Value) of the choice in ServiceNow. So your single-select should look similar:

* Type Choice 🔍 ⓘ

* Column label Difficulty

* Column name difficulty

Choices			
		≡ Label	≡ Value
<input type="checkbox"/>	<i>i</i>	<u>Difficult</u>	D
<input type="checkbox"/>	<i>i</i>	<u>Easy</u>	E
<input type="checkbox"/>	<i>i</i>	<u>Moderate</u>	M

Edit Custom Property Data Mapping [ServiceNowDataSync Test](#)

[← Back to Product Mappings](#)

To configure the data mappings for this custom property, please fill in the ServiceNow Data Sync external key values for the custom property below. If this is a custom list property, you also need to fill in the external key for each of the custom property values as well:

Artifact Type:

Name: List

External Key:

Custom Value Mapping

Value	Active	External Key ServiceNow Data Sync
Difficult	Yes	<input type="text" value="Difficult"/>
Easy	Yes	<input type="text" value="Easy"/>
Moderate	Yes	<input type="text" value="Moderate"/>

If you have a multi-select in SpiraPlan (List in ServiceNow), repeat the same steps as for a single-select, except instead putting "Label" in the external key, put "Value" instead. You should have something like this:

Edit Custom Property Data Mapping [ServiceNowDataSync_Test](#)

[← Back to Product Mappings](#)













To configure the data mappings for this custom property, please fill in the ServiceNow Data Sync external key values for the custom property below. If this is a custom list property, you also need to fill in the external key for each of the custom property values as well.

Artifact Type:* Incident
Name:* Operating System MultiList
External Key:*

Custom Value Mapping

Value	Active	External Key ServiceNow Data Sync
Android	Yes	<input type="text" value="Android Pie"/>
iOS	Yes	<input type="text" value="iOS 13"/>
Linux	Yes	<input type="text" value="Arch Linux"/>
Mac OS X	Yes	<input type="text" value="Mac OS Catalina"/>
Windows 10	Yes	<input type="text" value="Windows 10"/>
Windows 8	Yes	<input type="text" value="Windows 8"/>
Windows Server	Yes	<input type="text" value="Windows Server 2012"/>

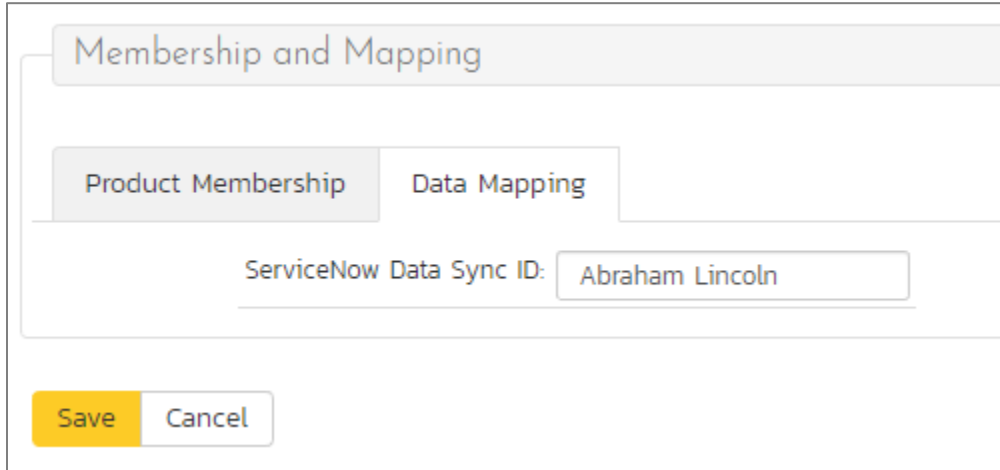
For these ServiceNow choices:

	Choices			 Label	 Value
<input type="checkbox"/>	 <u>Android</u>				Android Pie
<input type="checkbox"/>	 <u>iOS</u>				iOS 13
<input type="checkbox"/>	 <u>Linux</u>				Arch Linux
<input type="checkbox"/>	 <u>Mac OS</u>				Mac OS Catalina
<input type="checkbox"/>	 <u>Windows 10</u>				Windows 10
<input type="checkbox"/>	 <u>Windows 8</u>				Windows 8
<input type="checkbox"/>	 <u>Windows Server</u>				Windows Server 2012

16.2.5. Configuring the User Mapping

If you have set the “Auto-Map Users” option set to yes in the ServiceNow plugin, you can skip this section completely.

To configure the mapping of users in the two systems, you need to go to Administration > Users > View Edit Users, which will bring up the list of users in the system. Then click on the “Edit” button for a particular user that will be editing issues in ServiceNow:



The screenshot shows a web interface for user configuration. At the top is a header 'Membership and Mapping'. Below it are two tabs: 'Product Membership' and 'Data Mapping'. The 'Data Mapping' tab is selected. Underneath the tabs is a text input field labeled 'ServiceNow Data Sync ID:' containing the text 'Abraham Lincoln'. At the bottom of the form are two buttons: 'Save' (highlighted in yellow) and 'Cancel'.

Click on the ‘Data Mapping’ tab to list all the configured data-synchronization plug-ins for this user. In the text box labeled “ServiceNow Data Sync ID,” you need to enter the first and last name of the user in ServiceNow. This will allow the data-synchronization plug-in to know which user in SpiraPlan matches with an equivalent user in ServiceNow. Click [Save] once you’ve entered the appropriate login name. You should now repeat for the other users who will be active in both systems.

16.3. Using the Data Synchronization

Assuming everything was done correctly, the plug-in should start working. Start your Data Sync service and verify that issues in ServiceNow appear inside SpiraPlan. Note that the Data Sync service is not running constantly, so it may take some time for changes to materialize.

Congratulations, you have just integrated your SpiraPlan instance with ServiceNow!

Legal Notices

This publication is provided as is without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

This publication could include technical inaccuracies or typographical errors. Changes are periodically added to the information contained herein; these changes will be incorporated in new editions of the publication. Inflectra Corporation may make improvements and/or changes in the product(s) and/or program(s) and/or service(s) described in this publication at any time.

The sections in this guide that discuss internet web security are provided as suggestions and guidelines. Internet security is constantly evolving field, and our suggestions are no substitute for an up-to-date understanding of the vulnerabilities inherent in deploying internet or web applications, and Inflectra cannot be held liable for any losses due to breaches of security, compromise of data or other cyber-attacks that may result from following our recommendations.

SpiraTest®, SpiraPlan®, SpiraTeam® and Inflectra® are registered trademarks of Inflectra Corporation in the United States of America and other countries. Microsoft®, Windows®, Explorer® and Microsoft Project® are registered trademarks of Microsoft Corporation. All other trademarks and product names are property of their respective holders.

Please send comments and questions to:

Technical Publications

Inflectra Corporation

8121 Georgia Ave, Suite 504

Silver Spring, MD 20910-4957

U.S.A.

support@inflectra.com