Borland®

Borland Connect 1.6 HF 1

Using Borland Connect

Borland Software Corporation
700 King Farm Blvd, Suite 400
Rockville, MD 20850

2015-12-07

Contents

Introduction

This guide describes how to install, configure, and run Borland Connect. Read this guide to learn how to:

- Install the Borland Connect software and get it running as a service.
- Install and configure a connector for a third-party system.
- License the Borland Connect software.
- Use Borland Connect to create a mapping between the Borland Hub and your third-party system. (Alternatively, you can configure your connector by manually modifying XML.)
- Code a custom connector.
- Set up, run, and manage synchronizations between the Borland Hub and your third-party system.

About Borland Connect

Borland Connect is a tool used to synchronize item and relationship data between the Borland Hub and other repositories or clients. Once data is synchronized into the Borland Hub, all of the change management capabilities of Borland Hub --including versioning, branching, tracing, labeling, and reporting-- are available on that data.

Connector Types

Borland Connect uses *connectors* to synchronize assets with third-party systems. Some connectors, like the connector for Rally, are installed with Borland Connect. (You can find them in the root directory where you installed Borland Connect.)

Connectors for other systems --like those for AccuWork and Subversion-- are available for download from the Borland community site (*http://community.microfocus.com/borland/*). New connectors are always being developed, so be sure to check the community often.

Custom Connectors

If you don't see a connector for your system on the Borland community site, you can use the Java API installed with Borland Connect to create your own. The Java API makes it possible for you to integrate any data repository with the Borland Hub.

Refer to *Using Borland Connect* and to the Borland community for more information on creating a custom connector.

Example Use Case

Rally and Borland StarTeam Agile

This section describes a common use case for using Borland Connect with Rally.

- Borland Connect will synchronize the Sprint, Story, and Task assets and relationships between Rally and the Borland Hub.
- When a Project Manager or Borland Administrator using StarTeam Agile Requirement InfoStreams pushes Stories to the Team, the Rally Backlog will immediately populate with the Stories while the

Project Manager or Borland Administrator has full visibility into the Team Backlog from the StarTeam user interface.

- As team members are updating Rally, the changes are visible to the Project Manager or Borland Administrator in the StarTeam user interface so they can visualize the team progress and understand the impact of any changes they wish to make to the requirements.
- New Sprints or Stories created in Rally that were not pushed by the Project Manager or Borland Administrator are also visible in the StarTeam user interface, allowing the Project Manager or Borland Administrator to understand the other work the development team is planning. StarTeam Labels can be created across all of the assets from Borland Connect so that the Project Manager or Borland Administrator is able to establish a complete snapshot for the Release milestones.

Installation

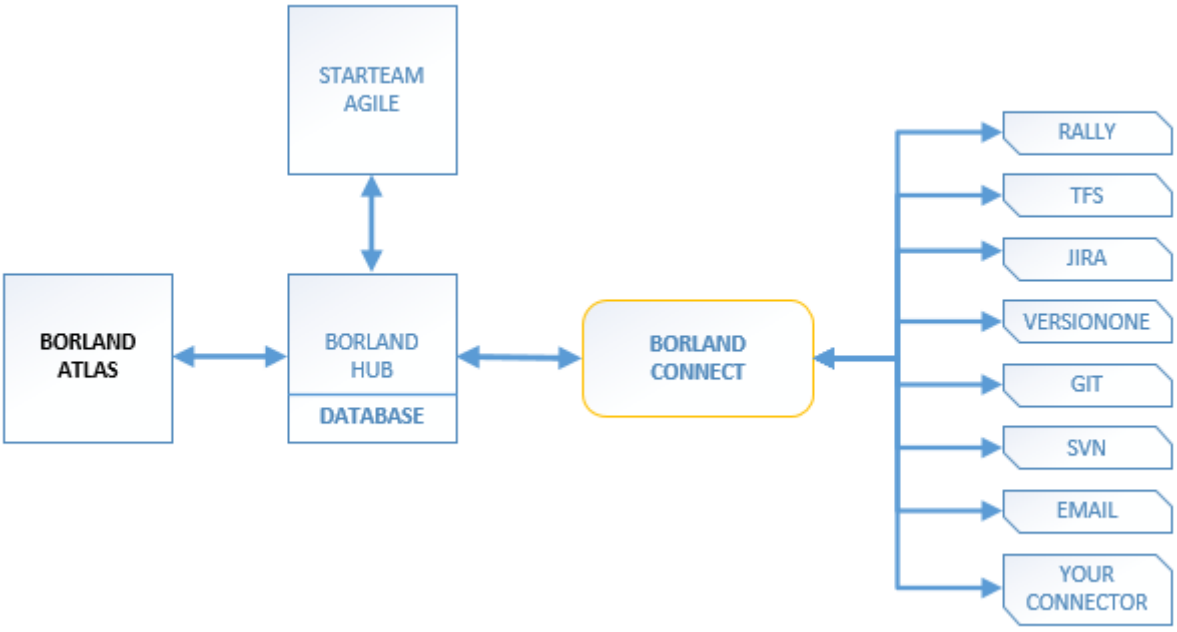
You are about to install the **Atlas Planning and Tracking Suite**. Before proceeding with installation, prepare the following items:

- A 64-bit quad core system with 16 GB RAM that meets the full set of system requirements.
- A database. If you do not have a database to use, a PostgreSQL database will be installed and configured by default. If you have one of the supported database types, you will need to continue to configure your database post installation.

A fully installed and configured **Atlas Planning And Tracking Suite** contains the following applications and components:

Atlas	A collaborative, flexible, agile requirements and delivery platform. It will be installed and run in a Tomcat server.
Borland Hub	A software change and configuration management server that stores the assets.
Database	Borland Hub connects to and supports many databases. If you do not have one installed, PostgreSQL will be installed.
Borland Connect	A web application that allows you to synchronize assets from third party tools into Borland Hub (and thereby, into Atlas). For example, you could connect Borland Hub to Rally via Borland Connect and have all stories from Rally appear in Atlas.
StarTeam Agile	A web application that will be installed and run in a Tomcat server. StarTeam Agile allows you to manage work using the agile methodology. If you use another tool to manage agile data, you can use Borland Connect to synchronize that data into Borland Atlas.
Tomcat Web Server	Two Tomcat web servers are installed. One runs Atlas, StarTeam Agile, and Borland Connect and the other runs the Search component.

The following depicts how the components are connected:



Installing the Atlas Planning and Tracking Suite

The following steps describe how to install the **Atlas Planning and Tracking Suite**. The suite includes Atlas, Borland Connect, StarTeam Agile, and Borland Hub. These applications will all be installed together.

Tip: If you want to install the Borland Hub on a separate server, you should plan to do that first. Go to that server, run the installer and select the **Borland Hub** option on the **Choose Install Set** screen. Complete that installation. Then, on the server for the web applications, choose the **Atlas Planning and Tracking Suite** option, during the installation, you will be asked to point to an existing Borland Hub, the one that you just set up.

Important: Unless you are upgrading from Atlas Planning and Tracking Suite 2.0, if you have any other version of Atlas, Borland Connect, StarTeam Agile, or the Borland Hub installed, you must uninstall them before proceeding. If you are a current Borland Connect customer and you want to maintain your mappings, you must retain a copy of `Connect.xml` before uninstalling. Afer installation, you can check your original `Connect.xml` file back into the `StarFlow Extensions/Connect` folder.

1. Run the installer.
2. Read the introduction and click **Next** on the **Introduction** screen.
3. Read the license agreement, select the option to accept the license agreement and click **Next**. The **Choose Install Set** screen appears. This is where you will be able to install either all of the applications or just install a new Borland Hub.
4. To install all of the web applications, choose the **Atlas Planning And Tracking Suite** option to install Atlas, Borland Connect, StarTeam Agile, and optionally, Borland Hub. Click **Next**.

   a) Select where you want the files installed on the **Choose Install Folder** screen. Do not click **Next** until you read the following:

   Important: The next part of the installation is where you choose the location. If you are upgrading, you need to make sure that you select the existing directory where the previous version was installed. If you used the default, you don't need to do anything in the next step because it is selected for you. However, if you chose a different location, you must browse to that folder and select it for the upgrade.

   If you are upgrading, you will receive an **Upgrade Required** message. Click **OK** and walk through the upgrade screens. Your upgrade will complete, and you are finished with the installation!

   If you are not upgrading, click **Next**. The **Borland Hub Connection** screen appears.

   To connect to an existing Borland Hub:

   1. Select **Connect to existing Borland Hub**.
   2. Click **Next**. The **Borland Hub Parameters** screen appears.
   3. Enter the Borland Hub connection information:

Borland Hub Host Name or IP Address	The IP address of an existing Borland Hub installation. **Important:** When specifying the address and port for Borland Hub in either the installation wizard or directly in the `ALMConfiguration.xml` file, you must use the actual IP address (or machine name) of the Borland Hub. Do not use "`localhost`" even when Borland Connect and the Borland Hub are running on the same machine.
Configuration Name	The configuration name of an existing Borland Hub installation.

Configuration GUID	The configuration GUID of an existing Borland Hub installation.
Configuration Port	The configuration port of an existing Borland Hub installation.
User Name	The user name of an existing Borland Hub installation.
Password	The user's password of an existing Borland Hub installation.

4. Click **Next**.

To install a new Borland Hub:

1. Select **Install new Borland Hub**.
2. Click **Next**. The **New Borland Hub** screen opens.
3. Enter a **Configuration Name** for the new Borland Hub.
4. If you already have a license key, select the **I have a license key** option.
5. Enter the license details.
6. Click **Next**.

5. To install the Borland Hub only, choose the **Borland Hub** option on the **Choose Install Set** screen.

Tip: After installation, if something goes wrong or a file is damaged or corrupted, you can attempt to repair the install by running this installer again

If you are upgrading, you need to:

1. Stop all Borland Hub server configurations using the Borland Hub **Admin Tool**.
2. Close the Borland Hub **Admin Tool**.
3. On the installer screen, click **Next**.
4. Click **Install**. The installer will install all upgrade files.

You are finished upgrading the Borland Hub!

If you are not upgrading, click **Next**.

1. Select the **Message Broker** option to install the Message Broker.
2. Select the **PostgreSQL** option to install a PostgreSQL database.
3. Click **Next**.
4. If you are installing a message broker, the Message Broker Options screen opens. Enter the port number on which the broker will listen.
5. If you are installing PostgreSQL, the **PostgreSQL Configuration** screen opens. Enter the password for the database superuser. Note that the username is `postgres`.
6. Click **Next**.

The **Choose Shortcut Folder** screen opens.

6. Select the shortcut folders and click **Next**. The Pre-Installation Summary screen opens.
7. Review the information and click **Install**.

Additional Configuration for Connecting to Existing Borland Hub

If you chose the option to **Connect to existing Borland Hub** in the Atlas installation wizard, the following configuration changes are required before using Borland Connect with Atlas:

1. Open the Borland Connect UI.
2. Navigate to the `HubDefault` data source.
3. Replace the default credentials with the correct account details for the Borland Connect user (this should be a server administrator account) in the **User Name** and **Password** fields.

4. In the **URL** field, replace the default StarTeam URL with the correct URL value.
5. Save the data source.
6. Click (**Settings**) on the main Borland Connect UI.
   a) Verify that the **Synchronization Frequency** field contains a valid integer.
   b) Verify that the **Maximum Number of Threads** field contains a valid integer.
   c) Click the **Error Notification** tab.
   d) In the **Email Settings** group, if there is a value in the **Frequency** field, verify that it contains a valid integer.
   e) Save the settings.
7. Go to the **Connections** page. For each connection, click (**Edit**). Look in the **Error Notification** group. If there is a value in the **Frequency** field, verify that it contains a valid integer.
8. Save the connection.
9. Restart the server.

Licensing

Borland Connect will run in evaluation mode for 60 days. At the end of this period, contact Micro Focus for a license file, `BCLicense.bin`. When you receive the license file, place it in the `data` directory referenced in either the `local.properties` file, or in the `BorlandConnectService.bat` file if you are running Borland Connect as a service.

Running the Borland Connect Server

You can run the Borland Connect server:

- Manually, by executing `start-connect.bat`
- Automatically, by configuring a Windows service to start the Borland Connect server each time your system starts

Both of these approaches make use of the configuration file `local.properties`, which contains parameters needed to start the Borland Connect server. This file is required if you start the server manually; it is optional if you start the server as a Windows service. Because of this, it is considered a best practice to configure Borland Connect options in the `local.properties` file, as described in the following section.

Before You Begin

Before you start the Borland Connect server, the `Connect.xml` file must be checked in to the StarFlow Extensions project located in the following folder:`StarFlow Extensions/Connect`. If this folder does not exist, you must create it and check in the `Connect.xml` file in order to start the Borland Connect server or UI.

Configure local.properties

To configure local properties:

1. In the root directory where you installed Borland Connect, open `local.properties.sample`.
2. Edit the following values:
   - url=

     Note: This is the URL to the file holding the XML configuration for an instance of Borland Connect.
   - user_directory=

     Note: The slashes in this path must be FORWARD (/). In addition, the service account under which you run the Borland Connect server must have rights to create files in the directory you specify here.
   - debug=

   Tip: Confirm that the values for user= and password= are valid, and consider changing the default password for the Administrator account.
3. Save `local.properties.sample` as `local.properties`.

Running the Borland Connect Server Manually

To run the Borland Connect server manually, run `start-connect.bat`.

This file is in the root directory where you installed Borland Connect. It is also available as a desktop icon called **Start Borland Connect** and through your Windows **Start** menu, depending upon options you chose during installation.

Note: If the `local.properties` file is inaccessible, the Borland Connect server will not start.

Configuring BorlandConnectService.bat

If you are running the Borland Connect server manually, you can skip this section. However, if you are running the Borland Connect server as a Windows service, you must first configure the `BorlandConnectService.bat` file as described here.

1. In the root directory where you installed Borland Connect, open `BorlandConnectService.bat`.
2. Edit the following lines by removing the REM statement and changing the values as appropriate for your environment:
   - set ST_USER=
   - set ST_PASSWORD=
   - set URL=

     Note: This is the URL to the file holding the XML configuration for an instance of Borland Connect.
   - set USER_DIRECTORY=

     Note: The slashes in this path must be BACKWARD (\). In addition, the service account under which you run the Borland Connect server must have rights to create files in the directory you specify here.
   - set DEBUG=

     Note: ST_USER, ST_PASSWORD, URL, and USER_DIRECTORY values are read from the `local.properties` file unless all of these parameters are specified here.
3. Update the SERVICE_LOG= parameter with the user data directory for Borland Connect.

Running the Borland Connect Server as a Windows Service

To run the Borland Connect server as a Windows service:

1. Configure `BorlandConnectService.bat` as described in the previous section.
2. Optionally, configure the `local.properties` as described earlier.
3. In the root directory where you installed Borland Connect, run the `encrypt.bat` to encrypt the password specified by the ST_PASSWORD= parameter in the `BorlandConnectService.bat` file.
4. In the root directory where you installed Borland Connect, run the `BorlandConnectService.bat` file. This creates the service named BorlandConnect. The service startup type is set to **Automatic** so that it starts each time the machine is rebooted.
5. To start the Borland Connect server as a service, open the **Windows Services** dialog and locate the BorlandConnect service.
6. Right-click the BorlandConnect service and choose **Start** from the context menu.

Note: If you want to remove the service, run `uninstall-serivce.bat`, located in the root directory where you installed Borland Connect.

Getting Started with Borland Connect

This topic describes how to start the Borland Connect UI, and it provides an overview of the Borland Connect dashboard and how it is used to configure data sources and connections, and how to run and manage synchronizations.

Note: There are some features available in Borland Connect that you can set only in the `Connect.xml` file. A sample of this file, named `Connect.xml.sample`, located in the installation directory, contains comments on how to implement these features.

Starting the Borland Connect UI

Before starting the Borland Connect UI, the Borland Connect server must be running. See *Running the Borland Connect Server* for more information.

1. Go to `http://<machine name>:8080/ConnectWeb/index.html`. The Borland Connect login page, **Borland Connect**, appears.
2. In the **Username** and **Password** fields, enter your Borland Hub user name and password, and then click **Login**. The Borland Connect UI appears.

Once you have logged in, take a moment to familiarize yourself with the Borland Connect dashboard, as described in *The Borland Connect Dashboard*.

Logging Out

Click the **Log out** button ().

Note: Logging out has no effect on the Borland Connect server.

The Borland Connect Dashboard

The main page of the Borland Connect UI is referred to as the *dashboard*. The dashboard consists of three main pages; each is associated with its own tab:

- The **Connections** page displays a list of your installed connections and a summary of their synchronization attributes -- the connection's data source, the direction in which assets are synchronized (the default is bi-directional), the last time a synchronization was run, the last time a synchronization succeeded, and the current status (which can refer to either the synchronization task itself, or to the configuration that defines it; more on this later).

  The **Connection Details** panel shows detailed information about the currently selected connection, including information about how projects, types, fields, and relationships are mapped between your third-party system and the Borland Hub.

  Tip: Borland Connect installs connections for several third-party systems, some of which you might not use in your organization, by default. Consider using the Delete action to remove unused connections.
- The **Data Sources** page displays a list of currently defined data sources. A **Data Source Details** panel shows detailed information for the currently selected data source; the information displayed on this panel varies based on the data source. You can also use this page to create and edit data sources.

- The **Borland Hub Settings** page lets you connect to the Borland Hub associated with your Borland Connect installation. Once connected, you use this page to identify the types of assets you plan to synchronize with your third-party system.

The Dashboard Toolbar

The Borland Connect dashboard toolbar contains icons that:

-  - Log you out of the Borland Connect UI. (Logging out has no effect on the Borland Connect server.)
-  - Display the **Settings** dialog box, from which you can configure general system settings such as resources and error notification.
-  - Get help for Borland Connect, either by displaying the *Using Borland Connect* guide, or accessing the Borland Connect community (*http://community.microfocus.com/borland/managetrack/borland_connect/*).

What to Do Next

Once you are familiar with the Borland Connect UI, consider doing one or more of the following:

- Configure general system settings, as described in *Setting Synchronization Resources* and *Email Settings*.
- Remove any unused connections from the dashboard by selecting them and choosing the **Delete** action.
- If you are using one of the connections installed by default, start by configuring its data source. See *Creating a Data Source* for more information.
- If you are using a connector from the Borland Connect community, install and configure it as described in *Installing a Connector*.

Starting and Stopping Connections

A connection needs to be fully defined before you can start it.

Tip: Borland Connect attempts to run a connection once it is defined. You can see a connection's status on the Borland Connect dashboard.

- To start a connection, click its **Start** button () on the dashboard. The button changes to the **Stop** button () and the status is updated.
- To stop a connection, click its **Stop** button () on the dashboard. The button changes to the **Start** button () and the status is updated.

Connection Status Values

The **Connections** page contains a grid listing each of your connections. One of the columns in the grid is **Status**. The following are the available **Status** values:

Disabled	The user-configurable value of `Enabled` in `Connect.xml` is `false`.
Failed	One or more sync-sets associated with this synchronization have failed and no subsequent iterations have succeeded. See the `Running` status for an example.
Idle	There are no projects to synchronize or the `projectMaps` are disabled.

Restarting	The application is re-reading the configuration file (`Connect.xml`) and is restarting all of the synchronizations. When the sync is complete, the status will change to `Running` or `Failed`.

Running	The status will be `Running` when the last available status for all sync-sets is `Success`.
	For example: A synchronization has three sync-sets (`ss1`, `ss2`, `ss3`):
	`Case 1 RUNNING:`
	`  Iteration 2: ss1=success, ss2=success, ss3=success`
	`  Iteration 3: ss1=ongoing`
	
	`Case 2 FAILED`
	`  Iteration 2: ss1=failed, ss2=success, ss3=success`
	`  Iteration 3: ss1=ongoing`
	
	`Case 3 RUNNING:`
	`  Iteration 2: ss1=failed, ss2=success, ss3=success`
	`  Iteration 3: ss1=success, ss2=ongoing`

Server Reinitializing	The application is reinitializing after changes to the configuration file. This status will display for all connections until the application has re-read the configuration file and created the associated sync-sets.
Unknown	The synchronization status is not known to the application. There is an error.

Setting Synchronization Resources

Use this procedure to specify values that Borland Connect uses when synchronizing assets of your third-party system with those in the Borland Hub.

1. Click (**Settings**). The **Settings** screen appears.
2. Click the **Synchronization** tab if it is not already active.
3. Complete the following fields for your environment:

Field	Description
Maximum Number of Threads	Specifies the number of threads over which you want to spread synchronizations when using multiple data sources or projects. The default is 4.
Synchronization Frequency	Specifies the time, in seconds, between synchronizations. The default is 20.

4. Click **Save** to save your changes.

Setting Up Error Notification

Borland Connect can generate email messages when system and synchronization errors occur. You specify default values for settings that are used by both types of errors on the **Error Notification** tab of the **Settings** dialog box: your SMTP server settings, who you want to receive error notification email, and how often. If you want, you can override email settings on an individual connection basis as described in *Creating a Connection*.

1. Click (**Settings**). The **Settings** screen appears.
2. Complete the **SMTP Settings** and **Email Settings** fields as described below.

3. When you are done, click **Save** to save your changes.

SMTP Settings Use these fields to specify information needed to identify and connect to your SMTP server.

Host	The email system's host name.
Port	The email system's port number.
Login Name	The email user.
Password	The password for the email user.
From	The email address from which notification emails will be sent.

Email Settings Use these fields to specify email recipients and how often you want Borland Connect to send error notification email.

Send Email To	Email addresses for the users you want to receive error notification email. Separate addresses with a semicolon (;).
Frequency	The frequency with which you want Borland Connect to send error notification email. The default is 60 minutes.

Installing a Connector

You might need to install a connector if one for your third-party system wasn't installed with Borland Connect. To install a connector:

1. Go to our community site *http://community.microfocus.com/borland/* and download the connector you need to the Borland Connect folder.
2. Open the `Readme.html` file for the connector you downloaded and follow all steps to configure it. (Each connector has its own configuration procedures.)
3. Start Borland Connect to load your connector.
4. Navigate to the `logs` directory (`c:\Users\davidf\AppData\Local\Borland\Borland Connect\logs\`, for example) and open the most recent `BC-Log*.log`. You will see a line that displays success or failure for loading your connector:

   ```
   14 Mon Mar 16 11:01:09 PDT 2015: Discovered com.git.connect.GitDataPool
   ```

   or

   ```
   14 Mon Mar 16 11:01:09 PDT 2015: Warning: Failed to load a connector.
   ```

   If the log contains a warning, the connector did not load properly. Refer to the connector's `Readme.html` and verify that all of the connector's required .JAR libraries have been installed.

Merging Connect.xml Content

The `Connect.xml` file that is used by Borland Connect is stored in the Borland Hub within the StarFlow Extensions project located in the following folder: `StarFlow Extensions/Connect`. There are several scenarios where you may want to combine content from other `Connect.xml` files into this file. For example, you may have different `Connect.xml` files from a Borland Connect version upgrade or from a new connector that you downloaded from the community. To combine these files:

1. Using the Borland StarTeam Cross-Platform Client, check out `Connect.xml` from the StarFlow Extensions project located in the `StarFlow Extensions/Connect` folder of your Borland Hub.
2. Open it for editing.
3. Open your new connector's `Connect.xml` or `Connect.xml.sample` - these files should be available in the root directory of your connector folder.
4. Copy the full `<DataSource>` element that you want from your new connector file and paste it into the appropriate location in the `Connect.xml` from the Borland Hub.
5. Copy the full `<Synchronization>` element that uses the `<DataSource>` element in the previous step and paste it into the appropriate location in the `Connect.xml` from the Borland Hub.
6. Save the `Connect.xml` file and check it back into the Borland Hub.

Example: Integrating an SCM Connector

The most common use case for an SCM connector is to manage your Release and Integration branches in Borland Hub while the development teams use a different SCM tool for their sandbox branches. This model allows each development team to work with their sandbox tooling of choice while still maintaining an enterprise-class single source for release and change management in Borland Hub. To set up the SCM Connector:

1. Choose the appropriate SCM connector and follow its `Readme` file to set it up. Some connectors are installed to your Borland Connect installation directory, and others are available from the Borland community (*http://community.microfocus.com/borland/*).
2. In the StarTeam Cross-Platform Client, navigate to the Release or Integration view and use the **Copy URL to Clipboard** option on the folder that you want to synchronize. Place the URL in the `<sourceRootPath>` of the `Connect.xml` project map, corresponding to the correct `Source Project` name.
3. Perform a complete file checkout from the Borland Hub Release view. These will be the files used to establish the initial branch in the other SCM tool.
4. Open the other SCM tool and create a new branch (Integration branch) adding the complete hierarchy of folders and files checked out from the `Borland Hub Release` view.
5. Copy the URL (or unique identifier) to the root folder of the newly created Integration branch in the other SCM tool. Place this URL in the `<targetRootPath>` of the `Connect.xml` project map, corresponding to the correct target project name.
6. Start Borland Connect with the SCM settings completed.
7. Create child branches from the Integration branch in the other SCM tool for the development team to work in. These can be shared branches, individual branches, feature branches or defect branches. The team is free to work in these branches in any way that matches your best practices. When code changes are completed in the sandbox and merged back into the Integration branch in the other SCM tool, Borland Connect will automatically synchronize the changes into the `Release view` of Borland Hub to be tracked and managed along with other contributing team changes. If any changes are made in the `Release view` directly in the Borland Hub UI or by other contributing teams, then Borland Connect will synchronize those changes back into the Integration view to be merged into the team sand boxes.

Creating a Data Source

This topic describes how to use the Borland Connect UI to:

- Create a data source.
- Enter the data source connection details.
- Test the connection to the data source.
- Add types from the data source that you want to use.

Tip: You need a data source in order to fully define a connection.

1. On the Borland Connect dashboard, click the **Data Sources** tab.
2. Above the **Data Sources** list, click **Create New Data Source**.
3. On the dialog box that opens, complete the **Name** and **Product** fields:

New Data Source Name	Enter a unique name for your data source.
Data Source Product	The products that are available in the list are the connectors that were added with the product installation or the ones that you downloaded from the Borland community and installed yourself.

4. Click **Add Data Source**. Your data source is added to the list of data sources. Select it to edit it. The data source pane is refreshed with additional fields that are specific to that product. Refer to the `Readme.html` file that was installed with your connector for more information on these fields.

   Tip: If you are editing a data source for a connector that was installed with Borland Connect, the **Type** table is populated with the default asset types specified in the `Connect.xml` associated with that data source. The values in this table remain disabled until you connect to the data source.

5. Modify the **TimeOffset** property, if necessary.

   The `TimeOffset` property is a Borland Hub property. When the offset isn't set in the `XML`, after the first update that occurs to the datapool, the measured offset is calculated, in terms of seconds. This is displayed in the log if `debug` is `HIGH`, or `debug` is `SYNC_RUN_DETAILS`. If the Borland Hub and other server are *in sync* this value should be less than `3 seconds`.

   - If the `TimeOffset` is specified in the configuration file then it is used to determine the DataPool's current time in relation to the time where Borland Hub is running.
   - `TimeOffset` is the value that is added to *pool time* in order to get local Time. If the *pool Time* is 6:30 PM, and the local time is 2:30 PM, offset is `-4 hours` or `-14400`.
   - If the `TimeOffset` is not specified, then it is dynamically calculated/adjusted every 5 minutes by getting the modified time after an update of an item. The first calculation sets the offSet, subsequent calculations adjust it by moving it 1/3 toward the newly calculated offset.

6. Click the **Connect** button. Borland Hub connects to the data source you have specified.
7. Click the **Add New Type** button (you might have to scroll to make the button visible), and then use the drop-down button in the new field to choose the types from your data source that you want to make available to the Borland Hub.

   Important: You must add types to your data sources before you can add them to a connection.

8. When you are done adding types, click the **Save** button to save the data source. The new data source appears in the **Available Data Sources** panel and can now be added to a connection as described in *Creating a Connection*.

Creating a Connection

In Borland Connect, a *connection* represents the relationship between two data sources, typically between the Borland Hub and the data source you have defined for your third-party system. You use the connection:

- To identify the data source whose assets you plan to synchronize with the Borland Hub.
- To specify the asset types, fields, and relationships you want to synchronize.
- To specify the projects you want Borland Connect to synchronize.

You perform these tasks using the **Create Connection Wizard**, as defined in the following procedure, or later, by editing an existing connection.

Note: The data source you want to use for a connection must exist before you can complete a connection. See *Creating a Data Source* for more information.

1. From the **Connections** tab on the dashboard, click the **Add Connection** button. The **Create Connection Wizard** appears.
2. On the **Data Source** tab, enter a name in the **Connection Name** field.
3. Optionally, use the fields in the **Error Notification** group box to override error notification email settings specified in **Settings**:
   - Enter email addresses (separated by a semicolon (;)) in the **Send Email To** field.
   - Change the value in the **Frequency** field.

   Tip: Email addresses and frequency values established on the **Settings** page are not displayed here.
4. Select the data source you want to use with this connection from the **Available Data Sources** panel and review its information in the **Selected Data Source Details** pane.

   Note: If the data source you need is not displayed in the **Available Data Sources** panel you will have to create it. See *Creating a Data Source* for more information.
5. Click the **Next** button if you want to define types and fields you want to synchronize now. The **Types and Fields** tab appears. See *Adding Types to Connections* for more information. Otherwise, click the **Save** button to save the connection.

   Note: If you save the connection at this point, note that it is incomplete.

Adding Types to a Connection

A *type* is a system asset such as a task, topic, change request, or story. When creating a connection, you must add at least one type to synchronize between your data source and the Borland Hub. For example, you may want to map a Rally *Hierarchical Requirement* to a Borland Hub *Story*.

Important: The types that are available to add to a connection are bound by the types that you added to your data source. If the type that you want is not listed, return to the **Data Sources** tab and add the type to the data source. See *Creating a Data Source* for more information.

1. If you are in the process of creating a connection, navigate to the **Types and Fields** tab. If you are editing an existing connection, on the dashboard, click the **Edit** button ( ) and navigate to the **Types and Fields** tab.
2. Select a direction for data synchronization from the **Default Sync Direction** list. All synchronizations will be performed in this direction unless you choose to override it. Values for synchronization direction are:

From Borland Hub	Data is only synchronized from the Borland Hub to your data source.
To Borland Hub	Data is only synchronized to Borland Hub from your data source.
Bi-Directional	Data is synchronized both from the Borland Hub and your data source, and vice versa. (This is the default.)

3. Click the **Add Type Mapping** button. The **Add Type Mapping** dialog box appears with the **Fields** tab selected by default.
4. The **<system name> Type** drop-down list contains all the types that you added to your data source. Select a type from the list.

   Tip: Borland Connect uses the data source product name for the <system name> displayed in the UI.
5. The **Borland Hub Type** drop-down list contains all the types available in the Borland Hub. Select the type you want to map to the third-party system type you just selected.

   Note: For connectors that support change set capabilities, such as Git or Subversion, the only item available from the **Borland Hub Type** drop-down list is `ChangeSet`. In addition, the only tab used to configure this type is the **Advanced** tab.
6. Click **Add Field Mapping**. You use the controls that appear to choose the fields for the type you are synchronizing that you want to map. For example, you might want to map the **Name**, **Description**, and **Status** fields of a **Story** type.
   a) Select a field to map for your third-party system.
   b) Select the synchronization direction in the **Direction** list for each field.
   c) The **HTML Conversion** list allows you to choose if your HTML-based fields should be converted to plain text or vice-versa. Select either **No Conversion**, **From Borland Hub**, or **To Borland Hub**.
   d) Select the **Borland Hub Field** to map to your field.
   e) If you have created user maps and you are currently mapping a user-based field, click the **Enable User Map** option.

   Tip: If the fields you are synchronizing use different values, you will want to create a value map. See *Adding Value Maps to Types* for more information.
7. Click the **OK** button to save your mappings and close the **Add Type Mapping** dialog box, or click the **Relationships** tab to specify the relationships you want to synchronize as described in *Adding Relationships to Types*.

Adding Relationships to Types

Most systems store relationship data as a standard property of an asset. So in order to synchronize relationships, Borland Hub needs to know the name of the property, type of data in the property, the relationship type you want to store the type as, and the direction in which you wish Borland Connect to synchronize this data.

1. If you are in the process of creating a connection, in the **Add Type Mapping** dialog box, click the **Relationships** tab. If you are editing an existing connection, click the **Relationships** button ( ) on the **Types and Fields** tab of the **Update Connection** window.
2. Click the **Add** button for each data source relationship you want to map. A new row appears in the list. The first column displays the **Type** name.
3. In the **FIeld Name** column, select the name of the property that contains the relationship value.
4. In the **Field Data Type** column, select the type of data stored in the property. For example, a Rally Work Product property contains `Type` or `Defect` type data.
5. In the **Type of Relationship** column, select one of the fixed list of relationship types found in the Borland Hub.

6. In the **Direction** column, select the mapping direction:

From Borland Hub	Data is only synchronized from the Borland Hub to your data source.
To Borland Hub	Data is only synchronized to Borland Hub from your data source.
Bi-Directional	Data is synchronized from the Borland Hub to your data source, and vice versa.

7. Click the **OK** button.
8. If you are in the process of creating a connection, optionally click the **<system name> Sync Criteria** tab to specify criteria you want to use to restrict synchronization to certain system assets. See *Adding Data Source Sync Criteria to Types* for more information.

   Tip: Borland Connect uses the data source product name for the <system name> displayed in the UI.

Specifying Synchronization Criteria for Third-Party Types

If you want, you can specify criteria that Borland Connect will use to restrict synchronization to a certain subset of types. You do this by selecting a type from your data source and creating a query to specify the values you want that type to match in order to be synchronized. You can use `AND` or `OR` operators to define as many query clauses as needed. For example, you could write a query to select only those `Rally Hierarchical Requirements` whose `type=Story` and whose `status=New`.

Tip: You can also specify synchronization criteria for Borland Hub types. See *Adding Borland Hub Sync Criteria to Types*.

1. If you are in the process of creating a connection, in the **Add Type Mapping** dialog box, click the **<system name> Sync Criteria** tab.

   Tip: Borland Connect uses the data source product name for the <system name> displayed in the UI.

   If you are editing an existing connection, click the **Sync Criteria** button ( ) on the **Types and Fields** tab of the **Update Connection** window.
2. Click the **Add** button. A new row appears in the list.
3. Select the data source property from the **Select Field** list. For example: `Blocked`.

   Tip: If your data source does not provide a list of property names, you can type it in this field.
4. In the **Value** column, enter the value that the field must contain in order to be synchronized. For example: `True`.
5. Optionally, click the **Add** button to specify synchronization criteria for another field. A field that allows you to specify `AND` or `OR` operators appears.
6. Choose the operator you want to use to construct your query.

   Note: You can use only one operator or the other when constructing your query. (You cannot mix `AND` and `OR` clauses, for example.)
7. If you want to specify synchronization criteria for Borland Hub types, click the **Borland Hub Sync Criteria** tab. See *Specifying Synchronization Criteria for Borland Hub Types*. Otherwise, click **OK** to save your changes.

Specifying Synchronization Criteria for Borland Hub Types

This optional page allows you to define what subset of data for the type should be synchronized. You will need to select a type from the Borland Hub and create a query to specify the values.

1. If you are in the process of creating a connection, in the **Add Type Mapping** dialog box of the **Create Connection Wizard**, click the **Borland Hub Sync Criteria** tab. If you are editing an existing connection, click the **Edit** button ( ) on the **Types and Fields** tab of the **Update Connection** window and then click the **Borland Hub Sync Criteria** tab.
2. Click the **Add** button. A new row appears in the list.
3. Select the Borland Hub property from the **Select FIeld** list. For example: `Blocked`.

   Tip: If your data source does not provide a list of property names, you can type it in this field.
4. In the **Value** column, enter the value that the field must contain in order to be synchronized. For example: `True`.
5. Optionally, click the **Add** button to specify synchronization criteria for another field. A field that allows you to specify `AND` or `OR` operators appears.
6. Choose the operator you want to use to construct your query.

   Note: You can use only one operator or the other when constructing your query. (You cannot mix `AND` and `OR` clauses, for example.)
7. Click **OK** to save your changes.

Modifying Advanced Values for Types

You use the **Advanced** tab of the **Type Mapping** dialog box to specify the names of the fields in your third-party system and in the Borland Hub that are used to store the unique ID of the types you are synchronizing -- if you are synchronizing stories and defects, you would enter the names of those fields here, for example.

1. If you are in the process of creating a connection, in the **Add Type Mapping** dialog box of the **Create Connection Wizard**, click the **Advanced** tab.

   If you are editing an existing connection, click the **Edit** button ( ) on the **Types and Fields** tab of the **Update Connection** window, and then click the **Advanced** tab.
2. Enter the names of the fields used to store the IDs of the types you are synchronizing. You must specify at least one of the type ID fields. The Borland Hub property defaults to `External ID`.
3. Click **OK**.

Adding Value Maps to Fields

Field values sometimes vary across systems. For example, in Rally, a value for `ScheduledState` may be `Complete`, whereas the corresponding field in the Borland Hub, `Status`, uses `Fixed`. You express these differences by defining value maps as part of your connection.

Note: You can add value maps only after the field map between your third-party system and the Borland Hub has been created.

1. From the dashboard, click the **Edit** button ( ) for the connection for which you will be defining one or more value maps. The **Update Connection** window appears.
2. In the **Update Connection** window, navigate to the **Types and Fields** tab.
3. In the **Enabled Types** pane, click the triangle to expand the row for the type that has a field for which you wish to create a value map. The field mappings you have defined for the selected type are displayed.
4. Click the **Edit Value Maps** button ( ). The **Value Maps** dialog box appears.
5. Click the **Add** button. A new row appears in the list.
6. Use the **Mapped Value** and **Borland Hub Field** fields to enter pairs of the values you want to map. For example, if you were creating a value map for status, you might enter `Complete` for the Rally value and `Fixed` for the Borland Hub.
7. Repeat steps 4 and 5 as needed to create additional value maps for the current field.
8. Click the **OK** button. The **Value Maps** dialog box closes.
9. Click the **Save** button on the **Update Connection** window to save your changes.

Adding Projects to a Connection

The last step of defining a connection is to map the projects from your data source to the projects in the Borland Hub.

1. If you are in the process of creating a connection, navigate to the **Projects** tab in the **Create Connection Wizard**. If you are editing an existing connection, click the **Edit** button ( ) on the dashboard for the connection whose projects you wish to specify, and then navigate to the **Projects** tab of the **Update Connection** window.
2. Click the **Add** button. A new row is added to the list.
3. Select the project you want to map from the **<system name> Project** list.

   Note: If you are using the email connector, you won't be selecting a project, you will be specifying an email address for Borland Connect to monitor. So, double click the list and type in the email address using the format: `emailAlias:password`.

   Tip: Borland Connect uses the data source product name for the <system name> displayed in the UI.
4. Select the corresponding Borland Hub project from the **Borland Hub Project** drop-down list.
5. Project synchronizations are enabled by default. If you wish to disable it, clear the check box.
6. Some data sources for SCM tools (like Git and Subversion) allow you to specify a deeper directory level synchronization. If it's available and you want to specify a directory deeper than the root directory for synchronization, click the directory button ( ) .

   The **Set Project Path** dialog box opens.

   For each data source, enter the path in the fields provided. Use the following formats:

Borland Hub	Full URL to project. To obtain this, open the Borland Hub, right click a project and select **Copy URL**.
Git	Start the path with a `/`. For example: `/ProjectName`.
Subversion	The repository is already specified on the **Connection Projects** page, so start the path with `/`. For example: `/ProjectName`.

   Click the **OK** button to close the **Set Project Path** dialog box.
7. Once you have added all projects you want to synchronize, click the **Save** button to save your changes.

User Maps

User maps define users that have different user names across multiple systems. For example, a user may be `User1` in Rally and `User 1` in the Borland Hub. User maps allow you to map these two users together so the systems know they are the same user. Use the following sections to learn how to create a user map and then to turn it on at a field level.

Creating User Maps

1. On the main Borland Connect UI, click the **User Maps** tab.
2. Click **Add User Map**. A row is added to the list.
3. Double click the `User Map` text in the **Name** field.
4. Rename the user map. This value is only used in Borland Connect and can be whatever you want it to be.
5. In the user map row, click  (**Add**). A new `user name` row is added as a child to the user map.
6. In the **Datasource** column, double click the new row to enable the list. Select the proper datasource for the user.
7. Double click the `User name` text in the **Name** field and type in the user name as it exists in that datasource.
8. Follow the same steps to add additional users/datasources to the user map.

Enabling User Maps

If you have created user maps, you still need to turn them on for the user fields.

1. Open a connection.
2. Navigate to the **Types and Fields** page.
3. Select a type and click  (**Edit**).
4. For each user-based field in the list, put a check in the **Enable User Map** column.

Developing a Custom Connector

This optional section provides an overview on how to code your own custom connector if it is not available from the product installation or on our community. Borland Connect provides a connector Java API that enables item data living in a repository to be synchronized with Borland Hub.

Once a connector is available for a repository, users can easily define the synchronization of items and their respective relationships through `Connect.xml`.

Each connector is responsible for executing operations to:

- Connect to the repository.
- Query the repository for items based on a given query/criteria.
- Fetch, insert, delete and update items within the repository.

Every connector is composed of the following two class implementations for executing the above operations:

DataPool.java	Contains methods for querying the repository for items to synchronize.
SyncItem.java	An in-memory representation of each item, including methods for getting and setting the item data.

Implementing DataPool.java

`DataPool` is responsible for establishing and managing the connection to a repository and fetching or creating the items in the repository when requested.

Each `DataPool` instance that is created by Borland Connect is a representation of the project to synchronize and a single item type and it's property data.

`DataPool` provides access to all of the configuration settings necessary to establish the `DataPool` connection with the repository:

- Credentials: `DataPool.getUserName()` and `DataPool.getPassword()`.
- Project name: `DataPool.getProjectName()`.
- Item Type name to create and fetch: `DataPool.getTypeName()`.
- Any attribute set in the configuration file for this DataPool instance: `DataPool.getConfigValue(String name)`. For example: repository URL.

Each `DataPool` instance must implement the following abstract methods in order to provide Borland Connect with the data/items from the repository to synchronize.

`SyncItem[] getAllItems()`	Returns the full set of Items from the repository matching the project and type context of the `DataPool` instance.
`prepareToSync(SyncItem[] items)`	Populates each of the items with the full set of values for the synchronized properties. The items passed in will be a subset of those returned from `getAllItems()`.
`boolean haveAnyItemsChanged(Date lastSyncTime)`	Returns `true` if there are any changed items in the project since the `lastSyncTime`.

boolean haveAnyItemsBeenDeleted(Date lastSyncTime)	Returns `true` if there are any deleted items in the project since the `lastSyncTime`.
String getDeletedBy(String id)	Returns the full name of the user that deleted the specified item ID.
Date getDeletedOn(String id)	Returns the date that the specified item ID was deleted from the project.
SyncItem createSyncItem(SyncItem parentSyncItem)	Creates a new item in the project with `parentSyncItem` (if specified) as the parent and returns the new item as a SyncItem.

Additionally, the base `DataPool` class provides the following methods for initialization and shutdown that can be overwritten to provide further custom implementation for the repository.

onSetup()	Called when the `DataPool` is initially created.
onStartSync(Date lastSyncTime)	Called at the beginning of each synchronization.
onEndSync()	Called at the end of each synchronization.
onClose()	Called when Borland Connect is shutting down the `DataPool`.
impersonate(String username)	Called prior to an update where the specified user should be persisted as the author.

User Interface Settings

Use the information in this section to modify the Borland Connect user interface with your custom connector.

Important: To use your custom connector with the Borland Connect user interface, a file named `com.borland.connect.DataPool` must reside in the `META-INF/services` folder of the jar. The contents contain the full class name of your datapool.

The following methods will call the base class and make changes to the list of properties:

- `DataPool.getProductName`
- `DataPool.getDefaultProps`
- `DataPool.getCustomProps`
- `DataPool.getDefaultPropsForType`

For example:

```
HashMap<String, String> r = super.getDefaultProps();
r.put(PoolConfig.USER, "MyDefaultUser");
return r;
```

Instead of modifying the hashmap directly, you can use these helper routines:

- `addProperty` - Use to add a property.
- `makeReadOnlyProperty` - Use to make a property read only.
- `makeHiddenProperty` - Use this to have the datapool set a value in a property but hide it from the users.
- `makePasswordProperty` - Use to make all characters entered in the password property into `x`.

A datapool can also implement `IMetaProvider` to provide repository specific data to drive the user interface. The methods in this interface are: `getTypes`, `getProjects`, `getProperties`, and `getEnums`.

See the Java doc for more information.

Implementing SyncItem.java

Each `SyncItem` instance is a representation of the in memory data for a single item in the repository and must implement the following abstract methods:

Object getValue(String property)	Returns the item's value for the specified property name.
boolean setValue(String property, Object value)	Sets the specified value on the item for the specified property name and returns true if the value was set in memory.
String getDisplayName()	Returns the String to display for this item.
SyncItem[] getHistory()	Returns the set of revisions representing this item's full history.
boolean update()	Update the item in the repository with the new values passed into `setValue` and return a boolean indicating if the update was successful.
remove()	Remove the item from the repository.
boolean lock()	Optionally lock the item in the repository while it's data is being synchronized.
unlock()	Unlock the item in the repository if it was locked during the synchronization.

Glossary

This glossary lists terms used throughout the *Using Borland Connect* guide.

Term	Definition
Connection	The configuration that permits synchronization of specific assets.
Connector	Java library that interacts with a third-party repository (Rally, for example) to perform synchronizations.
Connector type	The name of the third-party repository that a connector interacts with.
Custom Connector	A connector developed using the Java API installed with Borland Connect.
Data source	A specific instance of a connector (Rally running on a specific server, for example).
Synchronization	The process of synchronizing assets (typically bi-directionally) between a third-party repository (Rally, for example) and the Borland Hub.

Updates and SupportLine

Our Web site gives up-to-date details of contact numbers and addresses.

Contacting Micro Focus

Micro Focus is committed to providing world-class technical support and consulting services. Micro Focus provides worldwide support, delivering timely, reliable service to ensure every customer's business success.

All customers who are under a maintenance and support contract, as well as prospective customers who are evaluating products, are eligible for customer support. Our highly trained staff respond to your requests as quickly and professionally as possible.

Visit *http://supportline.microfocus.com/assistedservices.asp* to communicate directly with Micro Focus SupportLine to resolve your issues, or email supportline@microfocus.com.

Visit Micro Focus SupportLine at *http://supportline.microfocus.com* for up-to-date support news and access to other support information. First time users may be required to register to the site.

Information Needed by Micro Focus SupportLine

When contacting Micro Focus SupportLine, please include the following information if possible. The more information you can give, the better Micro Focus SupportLine can help you.

- The name and version number of all products that you think might be causing an issue.
- Your computer make and model.
- System information such as operating system name and version, processors, and memory details.
- Any detailed description of the issue, including steps to reproduce the issue.
- Exact wording of any error messages involved.
- Your serial number.

To find out these numbers, look in the subject line and body of your Electronic Product Delivery Notice email that you received from Micro Focus.

Creating a Dump File

If reporting a protection violation you might be asked to provide a dump (.dmp) file. To produce a dump file you use the Unexpected Error dialog box that is displayed when a protection violation occurs. Unless requested by Micro Focus SupportLine, leave the dump setting as `Normal` (recommended), click **Dump**, then specify a location and name for the dump file. Once the dump file has been written you can email it to Micro Focus SupportLine

You may also be asked to provide a log file created by the Consolidated Tracing Facility (CTF) - a tracing infrastructure that enables you to quickly and easily produce diagnostic information detailing the operation of a number of Micro Focus software components.

Creating Debug Files

If you encounter an error when compiling a program that requires you to contact Micro Focus SupportLine, your support representative might request that you provide additional debug files (as well as source and data files) to help us determine the cause of the problem. If so, they will advise you how to create them.

Index