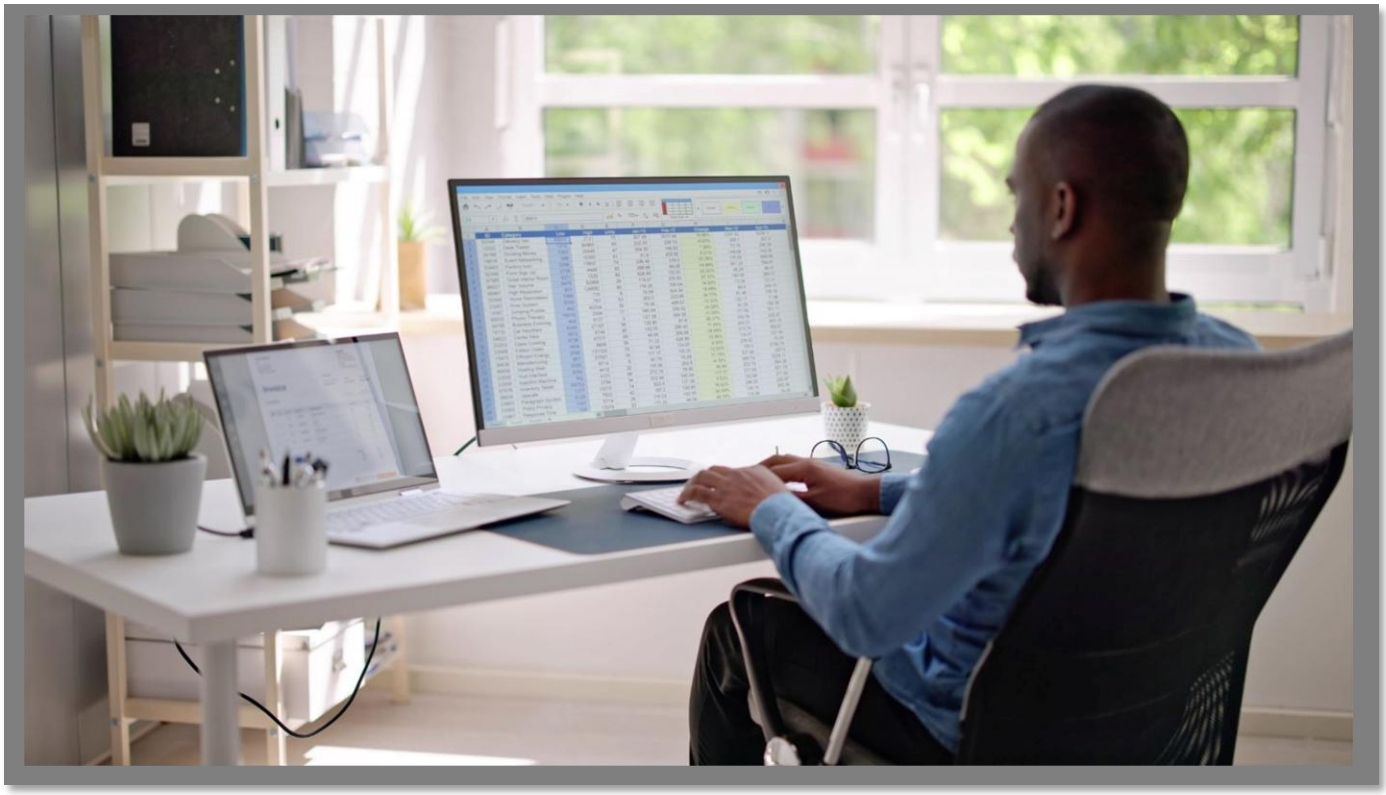
Arbutus Connectors

Jira

CONFIGURATION GUIDE



ARBUTUS

Powerful Analytics Simplified

Arbutus Connectors

Contents

Arbutus Connector – Jira

A. Introduction

The purpose of this Guide is to provide assistance with configuring the Arbutus Jira Connector using the ODBC Data Source Administrator. The configuration process can involve several technical steps that require a good understanding of IT systems and database management.

To make the most of this guide, it's advisable to have a good understanding of database connectivity, driver installation, and system settings. The ODBC Data Source Administrator, which is used as part of the configuration process, allows for the setup and management of data sources, enabling applications to access data from various database systems.

Due to the complexity and potential impact of these configurations, it is recommended that only those individuals with IT or database expertise undertake this task. In addition, it should also be understood that each client's network environment is different. A one-size-fits-all approach is rarely effective, as what works well in one environment may not be suitable in another.

B. About Jira

Jira is a popular project management and issue tracking tool developed by Atlassian. It is widely used by software development teams to plan, track, and manage their projects. Jira helps teams organize tasks, track progress, and collaborate effectively by providing features like customizable workflows, agile boards (such as Scrum and Kanban), and detailed reporting.

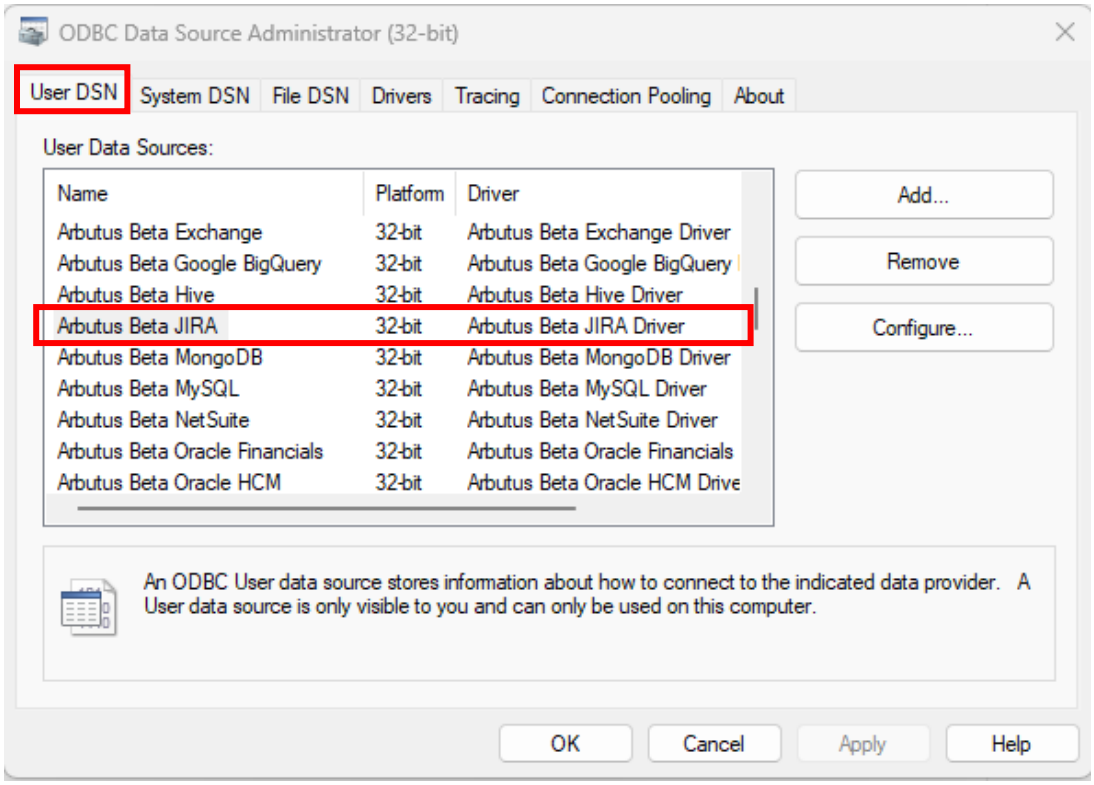
Jira stores its data in different ways depending on whether you are using Jira Cloud or Jira Server/Data Center. For example, with Jira Cloud, Amazon RS (Relational Database Service) is used for storing issue data, user information, and configuration settings, while attachments are sored in Amazon S3.

C. Determining if the Connector exists prior to configuring

Installation of the Arbutus Jira Connector is done at the time of installing the Arbutus software. For more information on this, please see the **Overview Guide Document**.

Once the Connector has been installed, the next step is to configure it.

Prior to configuring it, you can check to see if the Connector has been installed by opening the **32-bit ODBC Data Source Administrator**, pictured below, and clicking the **User DSN** tab. Included below is information on how you can access the **ODBC Data Source Administrator**.



- If the Arbutus Jira Connector appears in the list, it can be considered as installed.
- If it is not listed, it is likely that you did not select it during the installation or modification of the Arbutus software. In this case, it is recommended to reinstall the Arbutus software and choose the **Modify** option when prompted. For more details, please refer to the **Overview Guide Document**.

Below is the file path to access and run the **ODBC Data Source Administrator** application:

C:\Windows\SysWOW64\odbcad32.exe

Alternative, you can also try locating and opening the **ODBC Data Source Administrator** application by doing a search on your desktop application.

D. Configuring the Connector after it has been installed

Once you have verified that the Arbutus Connector has been installed, it is time to configure it.

This process is done using the **ODBC Data Source Administrator**. It can be described as "**editing the DSN configuration"**.

DSN, Drivers, and Data Sources
What is a DSN? DSN stands for Data Source Name, and is a unique name used to create a data connection to a database using open database connectivity (ODBC). A DSN is a data structure that contains the information required to connect to a database. It is essentially a string that identifies the source database, including the driver details, the database name, and often authentication credentials and other necessary connection parameters. DSNs facilitate a standardized method for applications to access databases without needing hard-coded connection details, enhancing flexibility and scalability in database management.

Arbutus Connectors

- ***Drivers*** are the components that process ODBC requests and return data to the application. If necessary, drivers modify an application's request into a form that is understood by the data source. The **Drivers** tab in the **ODBC Data Source Administrator** dialog box lists all drivers installed on your computer, including the name, version, company, file name, and file creation date of each driver.

- ***Data sources*** are the databases of files accessed by a driver and are identified by a data source name (DSN). You use the ODBC Data Source Administrator to add, configure, and delete data sources from your system.

All ODBC connections require that a DSN be configured to support the connection. When a client application wants to access an ODBC-compliant database, it references the database using the DSN.

The types of DSNs are:

- **User DSN** – User DSNs are local to a computer and can be used only by the current user. They are registered in the HKEY_Current_USER registry subtree.

- **System DSN** – System DSNs are local to a computer rather than dedicated to a user. The system or any user with privileges can use a data source set up with a system DSN. System DSNs are registered in the HKEY_LOCAL_MACHINE registry subtree.

- **File DSN** – File DSNs are file-based sources that can be shared among all users who have the same drivers installed and therefore have access to the database. These data sources need not be dedicated to a user nor be local to a computer. File data source names are identified by a file name with a .dsn extension.

User and system data sources are collectively known as *machine* data sources because they are local to a computer.

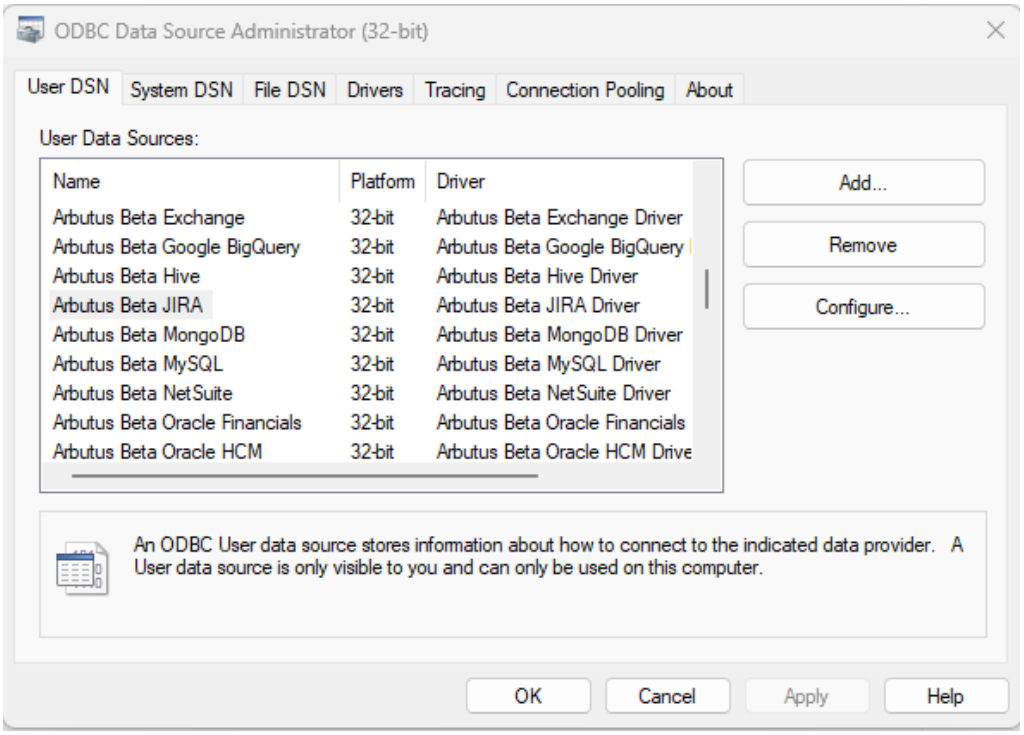
Each of these DSNs has a tab in the **ODBC Data Source Administrator** dialog.

The Arbutus ODBC Driver for Jira enables real-time access to Jira data, directly from any applications that support ODBC connectivity, the most widely supported interface for connecting applications with data.

Arbutus Connectors

Follow these steps to edit the DSN configuration and configure the Connector.

1. First open the **ODBC Data Source Administrator**.



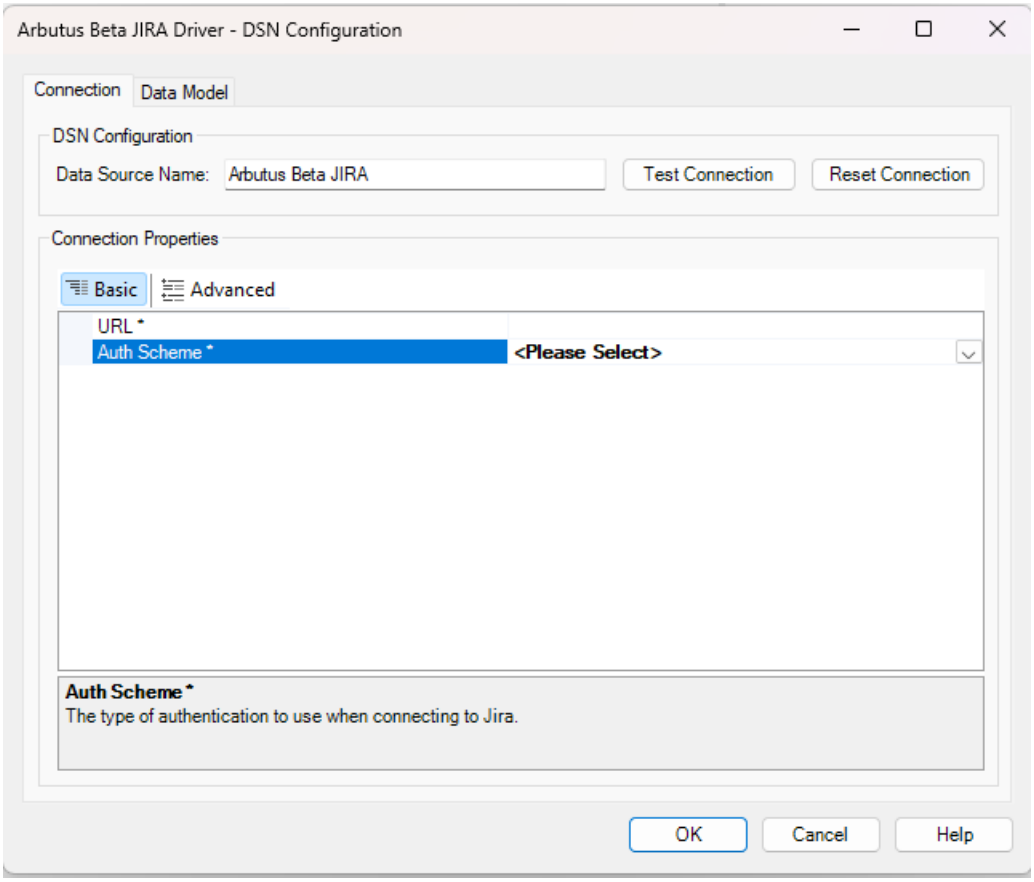
2. Click the **User DSN** tab.

   Selected data connectors are installed as **User DSN's** in Window's 32 Bit **ODBC Data Source Administrator**.

   Also, each of the data connector's names is prefaced with Arbutus, for example, **Arbutus Jira.**

3. Select the Arbutus Connector, in this case it is **Arbutus Jira**.
4. Click **Configure**.

This opens the **Arbutus Jira Driver – DSN Configuration** dialog.



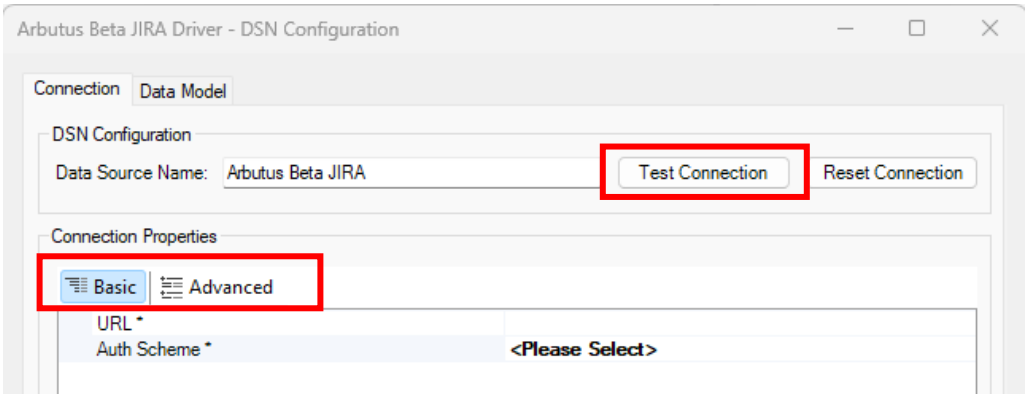
E. Editing the DSN properties – the Basic and Advanced tabs

With the DSN Configuration dialog open, the next step is to edit the DSN properties, where necessary, in the **Basic** and **Advanced** tabs. For example, editing the **Auth Scheme properties** (per screenshot below) to ensure correct authentication to the server is applied.

E1. Editing the DSN properties in the Basic tab

The properties listed in the **Basic** tab are typically the ones that are most commonly used, and as such are designed to be more user-friendly and straightforward, allowing you to quickly make changes without needing in-depth technical knowledge.

Once you have completed editing the properties in the **Basic** tab, you can go ahead and try testing the connection to the Jira system by clicking the **Test Connection** button, as highlighted in the screenshot below.



In the **Basic** tab, there is **one** main property to review:

1. **Auth Scheme** – click the dropdown to select from the list the appropriate scheme used for authentication. The options available for selection are as follows:
    - OAuth – select this if you want to use OAuth for authentication. Some of the key reasons to use this option include:
        - OAuth provides a secure way to authorize access to Jira without sharing your username and password. It uses tokens to grant access, which can be limited in scope and duration.
        - OAuth is ideal for integrating Jira with third-party applications, allowing them to access Jira data on behalf of users without exposing their credentials.
        - OAuth requires user consent for access, ensuring that users are aware of and approve the permissions granted to the application.
        - OAuth is a widely adopted standard for authorization, making it easier to implement and maintain across different systems and applications.

- PAT (Personal Access Token) – select this if you want to use a personal access token for authentication. Some of the key reasons to use this option include:
  - PATs provide a secure way to authenticate without using your username and password. If an external system is compromised, you can simply revoke the token instead of changing your password in all scripts and integrations.
  - PATs are ideal for integrating external applications with your Jira instance, allowing them to access Jira data securely.
  - PATs are created for individual users and carry the same permissions as the user who created them, ensuring that access is controlled and limited to what the user is allowed to do.
  - You can easily revoke a PAT if it is no longer needed or if you suspect it has been compromised, providing a straightforward way to manage access.

  Selecting **PAT** requires you to specify the following:
  - PAT – this is the Personal Access Token of the currently authenticated user.

- API Token – select this if you want to use an API token for authentication. Some of the key reasons to use this option include:
  - API tokens provide a secure way to authenticate without using your username and password. This reduces the risk of credential exposure.
  - API tokens are ideal for integrating external applications with your Jira instance, allowing them to access Jira data securely.
  - API tokens are created for individual users and carry the same permissions as the user who created them, ensuring controlled and limited access.
  - You can easily revoke an API token if it is no longer needed or if you suspect it has been compromised, providing a straightforward way to manage access.

Selecting **API Token** requires you to specify the following:

- User – this is the Jira user account used to authenticate. Together with **Password**, this field is used to authenticate against the Jira server.

- API Token – this is the API Token of the currently authenticated user.

- Basic – select this if you want to use basic authentication. Some of the key reasons to use this option include:
  - Basic authentication is straightforward to implement, making it suitable for simple scripts, manual API calls, or quick testing.
  - It allows you to directly use your Jira username and password (or an API token) to authenticate, which can be convenient for quick access.
  - Basic authentication is widely supported by various tools and libraries, making it easy to integrate with different systems.

  However, it is **important to note** that basic authentication is less secure compared to other methods like OAuth or API tokens, as it involves sending credentials with each request. For production environments, it's recommended to use more secure authentication methods

  Selecting **Basic** requires you to specify the following:

  - User – this is the Jira user account used to authenticate. Together with **Password**, this field is used to authenticate against the Jira server.

  - Password – this is the password used to authenticate the user. The **User** and **Password** are together used to authenticate with the server.

- Crowd – select this if you want to use Atlassian Crowd for authentication. Some of the key reasons to use this option include:
  - Crowd allows users to authenticate once and gain access to multiple Atlassian applications, including Jira, without needing to re-enter credentials.
  - Crowd helps manage user identities and access permissions centrally, simplifying the administration of user accounts and enhancing security.
  - Crowd can integrate with various directory services like Active Directory, LDAP, and others, providing a unified authentication mechanism.
  - By using Crowd, you can leverage its advanced security features, such as multi-factor authentication (MFA) and conditional access policies, to protect your Jira instance.

  Selecting **Crowd** requires you to specify the following:
  - User – this is the Jira user account used to authenticate. Together with **Password**, this field is used to authenticate against the Jira server.

  - Password – this is the password used to authenticate the user. The **User** and **Password** are together used to authenticate with the server.

  - SSO (Single Sign-On) Login URL – this is the identity provider's login URL.

  - SSO Exchange URL – this is the URL used for consuming the SAML response and exchanging it for service specific credentials.

    The ODBC Driver for Jira will use the URL specified here to consume a SAML response and exchange it for service specific credentials. The retrieved credentials are the final piece during the SSO connection that are used to communicate with Jira.

- SSO App Name – this is the App Name used with SSO for IdPs (Identity Providers) that require it.

  Along with **SSO App Password**, may be specified to identify and authenticate to your app configured in the SSO IdP. Currently only Crowd supports it.

- SSO App Password – this is the App Password used with SSO for IdPs that require it.

  Along with **SSO App Name**, may be specified to identify and authenticate to your app configured in the SSO IdP. Currently only Crowd supports it.

- OKTA – select this if you want to use Okta for authentication. Some of the key reasons to use this option include:

  - Okta provides SSO capabilities, allowing users to authenticate once and gain access to multiple applications, including Jira, without needing to re-enter credentials.
  - Okta helps manage user identities and access permissions centrally, simplifying the administration of user accounts and enhancing security.
  - By integrating with Okta, you can leverage its advanced security features, such as multi-factor authentication (MFA) and adaptive authentication, to protect your Jira instance.

  Note:
  Okta is a widely used cloud-based identity and access management (IAM) service that provides secure identity management and Single Sign-On (SSO) solutions. It helps organizations manage user authentication and authorization across various applications and services, ensuring secure access and streamlined user experience.

Selecting **OKTA** requires you to specify the following:

- User – this is the Jira user account used to authenticate. Together with **Password**, this field is used to authenticate against the Jira server.

- Password – this is the password used to authenticate the user. The **User** and **Password** are together used to authenticate with the server.

- SSO Login URL – this is the identity provider's login URL.

- SSO Properties - additional properties required to connect to the identity provider in a semicolon-separated list. This is used with the **SSO Login URL**.

- SSO Exchange URL - this is the URL used for consuming the SAML response and exchanging it for service specific credentials.

  The ODBC Driver for Jira will use the URL specified here to consume a SAML response and exchange it for service specific credentials. The retrieved credentials are the final piece during the SSO connection that are used to communicate with Jira.

- LDAP (Lightweight Directory Access Protocol) – select this if you want to use an LDAP directory for authentication and user management. Some of the key reasons to use this option include:

  - LDAP allows you to manage user identities and access permissions centrally, making it easier to maintain and update user information across multiple applications.

- If your organization uses a corporate directory like Active Directory or OpenLDAP, integrating Jira with LDAP ensures that user authentication and group memberships are consistent with your existing directory.
- By using LDAP, you can leverage the security features of your corporate directory, such as password policies and account lockout mechanisms.
- LDAP is designed to handle large numbers of users and groups, making it suitable for organizations of any size.

Selecting LDAP requires you to specify the following:

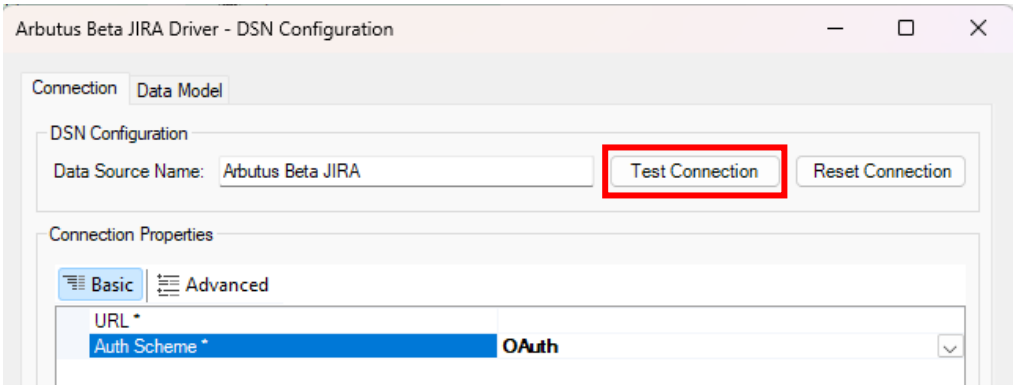
- User - this is the Jira user account used to authenticate. Together with **Password**, this field is used to authenticate against the Jira server.

- Password - this is the password used to authenticate the user. The **User** and **Password** are together used to authenticate with the server.

E2. Editing the DSN properties in the Advanced tab

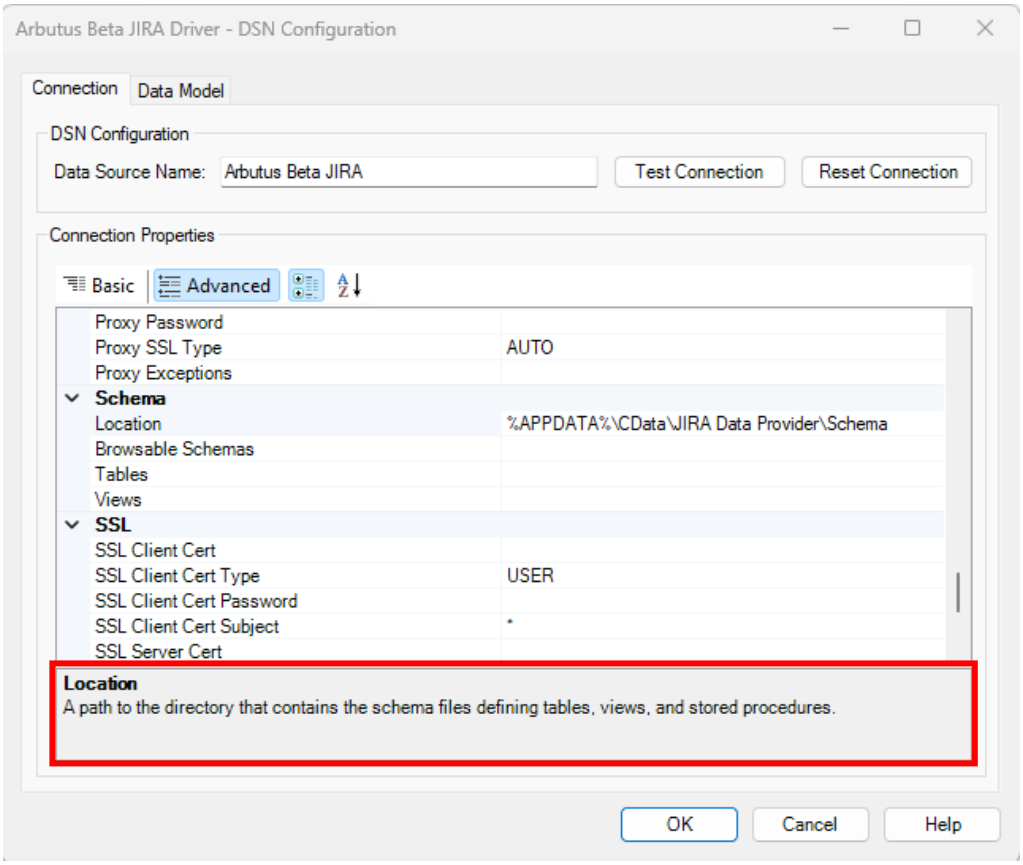
This tab includes more detailed and technical properties. It is intended for those users who need more control over the configuration and are comfortable with more complex options. The **Advanced** tab often includes properties that can fine-tune the behaviour of the system feature.

If you have already completed editing the properties in the **Basic** tab, as required, you do not necessarily need to also complete editing the properties in the **Advanced** tab. Instead, once you have completed editing the properties in the **Basic** tab, you may opt to proceed to testing the connection to the Jira system by clicking the **Test Connection** button.



There are a lot more properties included for editing in the **Advanced** tab.

However, it is useful to know that each property does provide a short description of it and as such serves as a guide in terms of what to edit and/or enter. This short description can be seen at the bottom of the **DSN Configuration** dialog box, as seen in the screenshot below.



If it is deemed necessary to complete some/all the properties in the **Advanced** tab, it is recommended that you refer to the description shown for any of the properties being edited and/or entered.

If required, more information on the properties listed in the **Advanced** tab can also be provided.

F. Other questions and/or request for assistance

There may be times when you need to consult with the technical support team at Arbutus Software. If so, please send an email request to support@ArbutusSoftware.com.

For more information, please refer to the CONTACT US page on our website.