Salesforce Connector

Contents

Chapter 1. Salesforce Connector

This guide describes how to install and use the IBM® Watson™ Explorer Engine Salesforce connector, which is available for Watson Explorer Engine.

The Salesforce connector enables Watson Explorer Engine applications to crawl Salesforce cloud-based CRM repositories and index the data that they contain.

To use the Salesforce connector you will create two search collections. The first collection crawls your Salesforce **Objects** data. The second collection crawls your Salesforce users. The Salesforce users collection is needed to add security to your Salesforce data collection. The procedures to configure both collections, as well as considerations you should make regarding security and API usage, are described in this documentation.

Additionally, this documentation contains a reference section, which describes the Salesforce connector's configurable options, and tips on how to debug the connector should problems arise. Release notes, which are comprised of a version-specific change log, and a known issues section, conclude the Salesforce connector documentation. For the initial version of the connector, the release notes section will be blank.

Note: The Salesforce connector is referred to as a bundled connector in the installation procedures of this documentation. Bundled connectors are repository specific connectors that are typically shipped with, but not installed in, Watson Explorer Engine foundational components. For a full list of other bundled connectors supported by Watson Explorer Engine, see Using Installed Seeds With Latest Connector Versions.

System Requirements

The Salesforce connector system requirements include the following:
- Watson Explorer Engine versions 10.0.0.1 or greater
- A Salesforce Edition supporting access through Salesforce Partner and Metadata APIs

Note: The Salesforce connector is available for all platforms that are supported by Watson Explorer Engine.

Chapter 2. Installing A New Or Updated Salesforce Connector

Watson Explorer Engine connectors are automatically installed as part of the product. However, updated and new connectors, may be delivered as archive packages available for download from IBM Fix Central outside of the Watson Explorer Engine release cycle.

New or updated connectors can be installed by extracting them into the top level directory of your Watson Explorer Engine installation:

- Linux default: `/opt/ibm/WEX/Engine/`
- Windows default: `\Program Files\IBM\WEX\Engine\`

The archive file contains the JAR files, other files, and repository nodes that using this connector requires.

Note: Before upgrading or installing a new Watson Explorer Engine connector, you must remove any previous version of that connector. Additionally, if an updated connector has a new name, a new XML node is created for your search collection and the previous XML node for your search collection is retained.

The next sections explain how to install a connector from a archive file using the standard connector installation procedure and, alternatively, how to manually install a specific connector.

Using the Standard Salesforce Connector Installation Procedure

About this task

To install a new or updated connector from an archive file (.zip file), do the following:

Procedure

1. Ensure that the existing connector is not being used by crawls of any associated resources.
2. Copy or move the archive file for the connector to the top level of your Watson Explorer Engine installation directory:

   Note: By default, the installation directory is `\Program Files\IBM\WEX\Engine` on Microsoft Windows systems, and `/opt/ibm/WEX/Engine` on Linux systems.
3. Extract the file. All files will be installed into the appropriate locations.

   Specifically, the following files will be installed:
   - `lib/java/plugins/CONNECTOR-VERSION.zip` - The connector plugin (where *CONNECTOR* is the name of the connector and *VERSION* is the specific version of the connector, such as 1.2.3).
   - `data/repository-supplements/function.vse-crawler-seed-CONNECTOR.xml` - The connector's seed component, where *CONNECTOR* is the name of the connector. When the repository is unpacked, Watson Explorer Engine will identify the new file in the `data/repository-supplements` directory and will install it into the product repository.
4. Log into your Watson Explorer Engine administration tool.

5. Navigate to **Management** > **Installation** > **Overview**. The installation screen displays.
6. In the **Repository** section of the **Installation** screen, click **unpack**. The message `Successfully unpacked repository files` displays when the new repository nodes have been successfully incorporated into the repository.

Results

After completing these steps, the new or updated connector will now be available as a seed when creating or modifying a site collection seed.

Manually Installing The Salesforce Connector

About this task

To manually install a new or updated connector, perform the following actions:

Procedure

1. Move the **crawl-seed** repository nodes for the new connector into the `data/repository-supplements` directory of your Watson Explorer Engine installation directory:

   Note: By default, the installation directory is `\Program Files\IBM\WEX\Engine` on Microsoft Windows systems, and `/opt/ibm/WEX/Engine` on Linux systems.
2. Move the archive file for the connector into the `lib/java/plugins` directory of your installation directory.
3. Move any JAR files for the connector into the `lib/java` directory in your installation directory. Alternately, they can also be installed into any directory that is located in the `Java CLASSPATH`.
4. To enable the new connector:
   a. Log into your Watson Explorer Engine administration tool.
   b. Navigate to **Management** > **Installation** > **Overview**. The installation screen displays.
   c. In the **Repository** section of the **Installation** screen, click **unpack**. The message `Successfully unpacked repository files` displays when the new repository node(s) have been successfully incorporated into the repository.

Results

After completing these steps, the new or updated connector will now be available as a seed when creating or modifying an existing site collection seed.

Chapter 3. Salesforce Connector Considerations

Salesforce data is comprised of **Objects**, which can be thought of as falling into three general groups: records, document folders, and social chatter. The Salesforce connector is able to crawl all of these Salesforce **Objects** but there are key differences among them, which you should consider before crawling your Salesforce data.

The primary difference among the Salesforce **Objects** is how the Salesforce connector handles security for them. The Salesforce connector provides security on all searchable Salesforce **Objects** and supports a user security model on the Salesforce **Objects** that comprise most of the Salesforce CRM data types, generally referred to as Salesforce records. However, Salesforce document folders and social chatter **Objects** use a different security model. In cases where the full security model is not supported, those **Objects** will only be viewable to the owners of those **Objects**. This ensures the highest level of security on those **Objects**.

Tip: By default, the Salesforce connector will crawl all searchable **Objects**. However, before crawling your Salesforce data, you may want to identify the **Object** types in your Salesforce repository and consider restricting the Salesforce connector to crawl only the **Objects** that are of most value to your users, while leaving certain **Objects** out. This will help conserve your Salesforce API calls (procedures on how to configure the Salesforce connector to limit the crawl to certain Objects are provided later in this documentation).

The next sections provide more detail about the key differences among these Salesforce Object types and how the Salesforce connector is designed to work with each of them. Because your Salesforce API usage may impact your ability to crawl and deliver data to your users, a section on how the Salesforce API can impact your ability to crawl your data is also provided.

Records Data

Salesforce records data is organized by **Object** type. The Salesforce connector starting point for retrieving Objects records data is the **isSearchable** attribute of the related Salesforce **Object**. This value must be set to `true` for **Objects** data to be crawled by the Salesforce connector.

Unless configured to retrieve a limited set of Salesforce **Objects**, which the connector can be configured to do in the connector seed settings (described later in this documentation), the Salesforce connector will retrieve all **Objects** data that has the attribute **isSearchable** set to `true`.

Important: The Salesforce connector checks that the **Objects** attribute value **isSearchable** is `true` to initially determine if those **Objects** are to be crawled. This initial starting point prevents the Salesforce connector from crawling the full array of all **Objects** data that may exist in your Salesforce application. Conversely, **Objects** in your Salesforce application that have the **isSearchable** attribute set to `false`, will not be included in the Salesforce connector crawl.

After the crawl, the default permission settings of the returned searchable results are determined by their **Organization Wide Default** settings. When the

Organization Wide Default for an **Object** type is `Public`, restrictions on who may view an **Object** are based on each user's **Profile** and **Permission Sets**.

When the **Organization Wide Default** for a Salesforce **Object** type is configured as `Private`, record-level sharing with **Users**, **Roles** and **Groups** is supported for certain **Object** types with certain limitations. Consult your Salesforce security model resources for more information.

Important: Some Salesforce **Objects** can be configured to use **Grant Access Using Hierarchies**. This feature is not supported by the Salesforce connector for **Custom Objects**. In such cases, the Salesforce connector will only use the **Objects** explicit, non-inherited, permissions to determine the rights associated with those **Objects**.

Important: Some Salesforce **Object** types inherit their permissions from another **Object** designated as the parent of that **Object**. In some cases, the Salesforce connector does not support record sharing settings for these child **Objects**, but instead restricts access to the record owner.

Note: Security based on APEX components is not supported by the Salesforce connector. Field-level security on records is supported by the connector, except when that security is dynamically based on the value of the field.

Document Folders

The Salesforce connector can crawl and retrieve Salesforce **Document Folders** data. The Salesforce connector stores rights for **Shared Folders**. **Personal Folders** are only accessible by the owner.

If your Salesforce data contains **Document Folders** data and you want to crawl that data and deliver it to your users, you will need to consider the following:

- Shared settings on **Document Folders** are supported by the Salesforce connector, except on hidden folders.
- **Document Folders** having a `namespace` prefix are not crawled by the Salesforce connector.

Social Chatter

The Salesforce connector crawls searchable social **Objects** such as **Feeds** and **Comments** on **Feeds**, **Cases**, and **Ideas**. To do so, the Salesforce connector uses a `vse_key` to link a **Comment** to its parent record. However, **Comment** data, which includes **FeedComment**, **CaseComment**, and **IdeaComment**, is only accessible by its owner.

API Usage

After accounting for the security and shared settings differences of the Salesforce **Objects** data that you want to crawl and deliver to your users, you will want to consider how your Salesforce API usage limitations will impact the Salesforce connector. Therefore, when preparing to crawl your data, consider the following:

- Your API usage limitations can unexpectedly be consumed if you are not careful as to understanding how the Salesforce connector will crawl your Salesforce data. For example: You might configure your Salesforce data collection to crawl only one **Object** type, with a limited number of records, in a large Salesforce organization, and be surprised to learn that you consumed a large number of

API calls. This is because the Salesforce connector will still need to crawl security information (as well as any Base64 binary fields) for this object type, which may greatly increase the API calls to your Salesforce repository. Therefore, when preparing to crawl your Salesforce data, be sure that the number of API calls allowed by Salesforce is large enough to crawl all the Salesforce **Objects** in your search, plus the security information configured in **OrganizationWideDefaults**, **Profiles**, **PermissionSets**, **Users**, **Groups**, **Roles**, **ShareObject** records, and binary fields.

- Binary data and certain types of security information are crawled at different rates, which affects the number of API calls used to crawl this data. This includes the Salesforce **Objects**: **OrganizationWideDefaults**, **Profiles**, **PermissionSets**, **Attachment**, **Documents**, **Document Folders**, **ContentVersion**, and metadata required by the crawl for all **Objects** and **Custom Objects**. This also includes binary field data which may exist on Salesforce **Objects** such as **FeedItem**, **Idea** and **Custom Object** types.
- For such items, there is a limitation of 1 binary field per API call, and a maximum of 10 records per API call otherwise. An additional daily allowance of API requests may be required if your Salesforce site contains many binary or security items.

  Note: For optional binary data fields, the Salesforce connector is designed to check your Salesforce **Objects** first to determine if binary data is present before making a separate API call to retrieve that data.

Continuous Update Mode

Continuous Update mode enables the Salesforce connector to periodically poll your Salesforce data collection for changes without re-crawling your entire Salesforce data collection. However, there are considerations you should make before enabling **Continuous Update** mode.

The first consideration involves how **Continuous Update** could impact your API usage. To configure **Continuous Update** mode, you enter an interval setting to tell the Salesforce connector how often it should check for changes to your Salesforce data collection. If this setting is configured to check too frequently in relation to how often your Salesforce data actually changes, you may make inefficient use of your API calls.

For example, you configure the Salesforce connector to check at 1 hour intervals for changes in your data collection, but changes of interest in your data collection typically occur about once per 24 hours. In such a scenario, the Salesforce connector would have checked for changes 24 times before retrieving any changes at all. Moreover, those same changes could have been retrieved more efficiently, and by using less API calls, by checking for changes only once per day.

Additionally, **Continuous Update** calculates security on each change in your Salesforce data collection. Therefore, if you have a complex hierarchy of **Roles**, **Profiles** and **Groups** defined in Salesforce, this can incur a high API cost, even if there are few changes to your Salesforce data collection. This is because the Salesforce connector will factor the entire hierarchy of permissions at the beginning of each **Continuous Update** cycle.

Tip: It is suggested that you if you enable **Continuous Update** mode that you closely monitor how it impacts your API calls.

The second consideration involves users permissions. **Continuous Update** mode only detects changes in the content of your Salesforce data collection. **Continuous Update** mode does not detect changes in cases where your content remains the same, but the record sharing or users permissions have changed. Therefore, changes in your users collection and record sharing are only updated after completing a full re-crawl of both collections.

Tip: Consider full re-crawls of your users collection when you are aware that big changes have been made to the **Roles**, **Permissions**, **Profiles**, and **Groups** hierarchy in your Salesforce CRM, which might result from an organizational change.

The configurable settings that enable **Continuous Update** mode are described in the Chapter 6, “Configuration Options Reference,” on page 17 section of this documentation. How to configure your Salesforce data collection to use Continuous Update mode is described in “Creating the Salesforce Data Collection” on page 9.

Chapter 4. Configuring the Salesforce Connector

This section describes how to configure the Watson Explorer Engine Salesforce connector to crawl your Salesforce CRM cloud-based application.

To use the Salesforce connector you will create two search collections. The first collection crawls your Salesforce data, which includes all searchable Salesforce records, document folders, and social chatter. The second collection crawls your Salesforce users. This collection is required to add security to the Salesforce objects in your data collection.

Note: The Watson Explorer Engine admin tool provides informational tool tips for each of the configurable Salesforce connector settings described in these procedures. You can access these tooltips by clicking on the question mark icon next to any setting. Alternatively, a complete listing of all Salesforce connector settings, and the tooltips that describe them, are provided in the Chapter 6, "Configuration Options Reference," on page 17 section of this documentation.

Creating the Salesforce Data Collection

This section describes how to create the Salesforce data collection.

About this task

To create your Salesforce data collection, log into your Watson Explorer Engine admin tool and do the following:

Procedure

1. Create a new search collection for your Salesforce data. This collection will be referred to as your Salesforce data collection. To create it, do the following:
   a. In the Watson Explorer Engine administration tool, click the **+** sign next to **Search collections**. The **New Search Collection** page displays.
   b. Enter a *name* for your Salesforce data collection in the **Name** field.
   c. In the **Copy defaults from** field, you can use the default setting, which is `default`, or optionally choose other default collection values from the list of available options.

      Note: This setting relates to what language your data contains. If your Salesforce data is English only, the `default` value should be fine. If your data contains other languages, click the tooltip for more information about this setting.
   d. In the **Description** text box, optionally choose to describe your collection or leave blank.
   e. Click **Add**. You have created a new search collection for your Salesforce data.
2. Add the Salesforce seed to your Salesforce data collection. The seed is the programmatic mechanism that enables the Salesforce connector to access your Salesforce data in the cloud. To add it, do the following:
   a. In the **Configuration** tab of your Salesforce data collection, click **Add a new seed**.

b. Select **Salesforce** from the scrollable list of available connectors, and click **add**. The **Crawling Configuration** page displays.

c. In the **Seeds** section, enter your Salesforce user name and password.

   Note: In some cases Salesforce may require you to append a security token to the password. Click the tooltip for more information on how to append this token if necessary. For more information on Salesforce security tokens, refer to your Salesforce documentation.

d. Configure optional connector settings in **Collection Details**, **Bounds**, **Continuous Update**, **Advanced**, and **Advanced - Logging** as appropriate for your crawl. Click the tooltips for more information on any of the optional connector settings or see the Chapter 6, "Configuration Options Reference," on page 17 in this documentation.

   Important: Though optional, the **Object Types** field, which is accessible in the **Collection Details** section of your **Seeds** settings, enables you to restrict your Salesforce data collection crawl to only those Salesforce **Objects** you would like to deliver to your users. If left blank, the Salesforce connector will crawl all searchable **Objects**. This can needlessly consume resources.

   Tip: If you are not sure which Salesforce **Objects** are searchable, it is recommended that you create a programmatic script, or use another utility to determine which Salesforce **Objects** are searchable before crawling all of your Salesforce data.

   Important: When enabled, **Run in continuous update mode**, will periodically poll Salesforce for changes. **Continuous Update** mode can impact your API call usage and there are security aspects of it to consider before enabling this feature. See "Continuous Update Mode" on page 7 for more information.

e. After configuring your connector seed settings, click **OK** or **Apply**. You have configured your Salesforce data collection seed.

3. Add the **Salesforce Converter** to your Salesforce data collection. The converter transforms your Salesforce CRM data into a format that can be indexed by Watson Explorer Engine. To add it, do the following:

   a. Click **Add a new converter**. A scrollable list of available converters displays.

   b. Select **Salesforce Converter** from the list of available converters.

   c. Click **Add**.

   d. Click **OK**. You have added the **Salesforce Converter**.

Results

You have created and configured your Salesforce data collection. At this point, you can test and crawl your Salesforce data collection by following the procedures described in "Testing and Crawling Your Search Collection" on page 12. However, you still need to create a Salesforce users collection to add security to your Salesforce data collection. The procedures to do so are described next.

Creating the Salesforce Users Collection

This section describes how to create the Salesforce users collection, which is needed to add security to your Salesforce data collection.

About this task

To create your Salesforce users collection, log into your Watson Explorer Engine admin tool and do the following:

Procedure

1. Create a new search collection for your Salesforce users collection. This collection will be referred to as your Salesforce users collection. To create it, do the following:
   a. In the Watson Explorer Engine administration tool, click the **+** sign next to **Search collections**.
   b. Enter a *name* for your Salesforce users collection in the **Name** field.
   c. In the **Copy defaults from** field, you can use the default setting, which is `default`, or optionally choose other default collection values from the list of available options. Click the tooltip for more information about this setting.
   d. Optionally choose to describe your collection or leave blank.
   e. Click **Add**.
2. Add the Salesforce Users seed to your data collection. The Salesforce Users seed is the programmatic mechanism that enables Watson Explorer Engine to crawl your Salesforce users and user authorization information such as **Roles**, **Groups**, and **Profiles**. To add it, do the following:
   a. In the **Configuration** tab of your Watson Explorer Engine search collection, click **Add a new seed**.
   b. Select **Salesforce Users** from the scrollable list of available connectors, and click **add**. The **Crawling Configuration** page displays.
   c. In the **Seeds** section, enter your Salesforce user name and password.

      Note: In some cases Salesforce may require you to append a security token to the password. Click the tooltip for more information on how to append this token if necessary. For more information on Salesforce security tokens, refer to your Salesforce documentation.
   d. Configure optional connector settings in **Advanced**, and **Advanced - Logging** as appropriate for your crawl. Click the tooltips for more information on any of the optional connector settings or see the Chapter 6, "Configuration Options Reference," on page 17 in this documentation.
   e. After configuring your connector settings, click **OK** or **Apply**.
3. Add the **Salesforce Converter** to your Salesforce users collection. The converter transforms your Salesforce CRM users data into a format that can be indexed by Watson Explorer Engine. To add it, do the following:
   a. Click **Add a new converter**. A scrollable list of available converters displays.
   b. Select **Salesforce Converter** from the list of available converters.
   c. Click **Add**.
   d. Click **OK**.
4. Add a **rights form component** to your Salesforce data collection. The rights form component is the programmatic mechanism that enables Watson Explorer Engine to add security to your Salesforce data collection. To add it, do the following:
   a. In Watson Explorer Engine admin tool, navigate to **Sources**.
   b. Click the **source** that is associated with your Salesforce data collection. The **Source** configuration screen displays.
   c. Click the **form** link or the **Form** tab. The **Form** screen displays.

d. Click **Add Form Component**. A scrollable list of form components displays.
e. Select **Salesforce Rights**.
f. In the **Salesforce Users Collection** field, enter the name you gave to your Salesforce users collection. Click the tooltip for more information.
g. Optional: Configure the **Salesforce User Name** and **Query Service URL**. Click the tooltips for more information about these fields.
h. Click **OK** or **Apply**.

Results

You have created and configured your Salesforce users collection. At this point, you can test and crawl your Salesforce users collection by following the procedures described in "Testing and Crawling Your Search Collection." The procedures to do so are described next.

Testing and Crawling Your Search Collection

About this task

Once you have created and configured a Salesforce data collection and a Salesforce users collection, you are ready to test and crawl your Salesforce CRM with the Salesforce connector and determine if your Salesforce connector is functioning correctly.

Note: The procedures described here assume you have Watson Explorer Engine Search Engine Query Service running. If not, navigate to **Management Services Search Engine** and click **start**.
To test and crawl your search collections, do the following for both of your collections:

Procedure

1. In the Watson Explorer Engine administration tool's **Configuration** tab, navigate to your site collection seed configuration display.
2. Click **Test It** to verify initial connectivity and that the Salesforce connector is successfully communicating with Salesforce.
3. Once connectivity has been successfully tested, authenticate with a user account, user name, and user password.
4. Perform a null search by simply clicking the search button in your Watson Explorer Engine or Watson Explorer Application Builder search display. If results are not displayed, something is likely not configured properly. A null search should return all results.
5. If results are successfully displayed, then test the search bar by searching for a document that is already known to be part of the index. For instance, an administrator may search for a document that they added to Salesforce previously and therefore know that this document should be returned as a search result by the Salesforce connector.
6. If a known document is successfully returned in the search results, then test security and ACLs (access control lists) by searching first for a document to which access should be granted based on the current account permissions, and then for a document that should not be accessible based on the current account permissions.

Important: During testing, you must confirm that users can only access documents whose security requirements are satisfied by a user's current security and access permissions. You should test security and ACL support by using several accounts with different Salesforce permissions.

7. At this point, it is suggested to check the overall document count in the search results. Perform a search query and note the number of results. Do the results make sense for your data?

Results

The Salesforce connector should successfully perform each of the tests in the previous list. If not, you may not have configured the Salesforce connector correctly, or an issue may exist in your Salesforce environment that is preventing successful connectivity and document retrieval. For more information see Chapter 7, “Salesforce Connector Debug Reference,” on page 19.

Chapter 5. Using Salesforce Connector Security with Application Builder

You can use the Salesforce connector security rights function with the Application Builder foundational component available in Watson Explorer.

Salesforce connector security rights are invoked by a configurable security script that is accessible in the Application Builder administration tool.

Salesforce Application Builder Security Overview

This section provides a general overview of the Salesforce connector security model that is available in the Application Builder foundational component.

To use the Application Builder foundational component with a Salesforce collection that includes group rights, you will need to configure a security script in the Application Builder administration tool. Application Builder invokes the security rights function in the Salesforce connector through use of the settings in this script.

Configuring the security script requires knowledge of several variable settings from your Salesforce installation. Descriptions for those settings are provided as comments in the Application Builder security script. These configurable settings enable the Application Builder security script to invoke the security rights function from the Salesforce connector by generally providing the authentication credentials needed to access your Salesforce application.

Adding Salesforce Application Builder Security

This section describes how to use Application Builder security with the Salesforce connector.

About this task

Note: This section assumes you have already configured a security collection for your Salesforce search collection. Additionally, it is assumed that you have successfully created an entity in the Application Builder administration tool and have established connectivity between that entity and your Salesforce search collection.

To configure Application Builder to use the security rights function of your Salesforce search collection, do the following:

Procedure

1. In the Application Builder administration tool, navigate to the entity that is associated with your Salesforce search collection.
2. Click **Configure entity**.
3. In the **Security options** section, set the **Configure access rights** toggle option to **on**.
4. In the **Connector rights file** dropdown menu, select the Salesforce connector from the list of available connectors.

5. In the **Rights code** text box, modify the provided code sample for your Salesforce installation. Comments in the script describe the settings that are required to be set.
6. In the **Specify field(s) required to access entity** box, enter the name of a field on the Salesforce document for which a user must have access to view the entire document. This is required for secure collaboration.
7. Click **Save entity**.

What to do next

Test your application with various user credentials to confirm that documents are being returned appropriately by Application Builder.

Chapter 6. Configuration Options Reference

This section describes all the configurable options for the Salesforce seed and Salesforce Users seed.

Note: The Salesforce Users seed contains a subset of the configurable Salesforce seed settings described below:

Seed Component

Salesforce Username - The Salesforce username to authenticate with. If you need to configure the proxy settings, use the parameters available in the **HTTP Specific** section under **Global Settings**.

Salesforce Password - The Salesforce password to authenticate with. To connect from an untrusted machine, a Salesforce security token must be appended to the password. A security token is an automatically generated key that you must add to the end of your password in order to log into Salesforce from an untrusted network. For example, if your password is *mypassword*, and your security token is *xxxx*, then you must enter *mypasswordxxxx* in the password field to properly authenticate with Salesforce and log in successfully.

Collection Details

Salesforce Organization Name - Enter the unique ID of your Salesforce Organization. This setting enables you to enter an optional value to differentiate URLs when more than one Salesforce seed is configured to be crawled.

Object Types - In this setting you can optional specify a comma separated list of Salesforce Object Types to crawl that will limit the scope of the crawl to the Object Types entered in this field. The default setting is to crawl all searchable Salesforce **Object Types**. For **Custom Object** names, append `__c` in order to match the Salesforce API convention for **Custom Object** names. Example: *`MyCustomObject`* would be entered as *`MyCustomObject__c`*.

Note: Do not specify **Comment Objects** such as `FeedComment`, `CaseComment`, `IdeaComment` without `FeedItem`, `Case`, `Idea` respectively. Do not specify **Tag Objects** without the parent specified. For example: Do not specify `AccountTag` without `Account`.

Bounds

Crawl All Versions - When enabled, the connector will crawl all document versions.

Batch Size - The number of records to fetch in a single batch (Salesforce query). Must be greater than `0`. In most cases, the default value of `500` should be sufficient.

Continuous Update

Run In Continuous Update Mode - Complete a **Full Crawl** before enabling **Continuous Update** mode. When enabled, the connector will periodically poll Salesforce for changes. When this option is enabled, the crawler is configured with an infinite maximum idle time.

Update Intervall - The amount of time that the connector waits before checking Salesforce for new updates (when running in **Continuous Update Mode**).

Advanced

Crawl Sandbox - When enabled, the connector will crawl the Salesforce sandbox.

Advanced - Logging

Enabled Connector Bootstrap Logging - When enabled, the connector will generate detailed startup trace logging. Enabling this option will have a negative impact on performance.

Connector Logging Configuration - Log4j configuration for the connector. Default configuration enables `ERROR` level logging. The token *`%LOGDIR%`* is replaced with the full path to the Watson Explorer Engine `temp` directory.

Chapter 7. Salesforce Connector Debug Reference

This section describes connector debugging techniques and methods that can help resolve common connector problems.

Validate user and password

Often simple connector issues can be attributed to basic account permission errors. Confirm that you are using an account with the appropriate permissions to crawl your repository and that you are using the right password for it.

Check the documentation

Be sure that you have correctly configured all installed Watson Explorer components and your connector, and that there are no missing steps, or incorrect configuration settings, which might be causing a problem in using the connector.

Tip: Rights functions for user collections are common connector pitfalls.

Eliminate resource-side errors

It is a good tactical step to "assume" the issue is with a Watson Explorer Engine connector but, at the same time, to make the administrator of the resource that you are crawling aware of any problems crawling that resource. The administrator may be aware of the issue and have a patch available. It never hurts to check.

Test multi-threaded versus single-threaded

To determine if a connector issue is related to multithreading, set the thread count to 1 and then test a new crawl. If an error is encountered, multithreading is not the source of the problem. Setting the thread count to 1 also has the benefit of making the log easier to read.

Enable bootstrap logging

If a connector is not starting at all, enable bootstrap logging to determine where the failure occurs when the connector is initiated. Bootstrap logging can enabled in the Watson Explorer Engine administration tool's seed configuration screen.

To activate bootstrap logging do the following:

1. From the seed configuration page of your site collection, go to **Configuration** > **Crawling**. The crawling configuration page displays.
2. Select the arrow and expand the **Advanced - Logging** collapsible menu.
3. Check the **enable connector bootstrap logging** box. Additionally, enter Log4j settings in the **Connector Logging Configuration** text box.
4. Click **OK/Apply**.

Enable connector logging

To troubleshoot a connector, you can enable a logging condition. To add a logging condition to the connector seed, do the following:

1. In the Watson Explorer Engine administration tool, select **Add A New Condition** from the seed in your site collection configuration screen.

   A pop-up window displays with a list of new conditions.
2. Scroll down and select **connector logging**.

Your goal is to capture a stack trace, which can help pinpoint what might be causing your connector problems.

Enable Log4J logging levels

Log4j enables you to activate different levels of logging without modifying the application binary thus avoiding a heavy performance cost. Logging behavior can be controlled by editing a Log4J configuration file.

Key logging levels that can be applied using the Log4j utility are the following:

- `OFF` - The `OFF` level has the highest possible rank and is intended to turn off logging.
- `FATAL` - The `FATAL` level designates very severe error events that will presumably lead the application to abort.
- `ERROR` - The `ERROR` level designates error events that might still allow the application to continue running.
- `WARN` - The `WARN` level designates potentially harmful situations.
- `INFO` - The `INFO` level designates informational messages that highlight the progress of the application at coarse-grained level.
- `DEBUG` - The `DEBUG` Level designates fine-grained informational events that are most useful to debug an application.
- `TRACE` - The `TRACE` Level designates finer-grained informational events than the DEBUG
- `ALL` - The `ALL` has the lowest possible rank and is intended to turn on all logging.

For more detailed information about Log4j and its configuration, see the online resources for Log4j.

Enable Oakland HTTP wire logging

Enabling logging for wire-level activity is useful for Watson Explorer Engine connectors that use HTTP connections. This is because the wire log records all data transmitted to and from your server(s) when executing HTTP requests. The wire log uses the `org.apache.http.wire` logging category, which should only be enabled to debug problems. Be aware that wire logging will produce a large amount of log data.

Check for missing JAR files

Be sure that you have all the JAR files needed. If the connector was installed correctly, the necessary JAR files should have been copied to the right location by default.

Open JMX port to profile resources

Java Management Extensions (JMX) supply tools for managing and monitoring applications, system objects, devices and service oriented networks.

To enable the JMX agent and configure its operation, you must set certain system properties when you start the Java virtual machine (JVM). For detailed instruction, consult help resources for using JMX and other JMX compliant tools.

Packet trace with Wireshark

If you are familiar with Wireshark and its advanced packet trace capabilities, it can be used instead of, or to augment, any packet tracing capabilities in the connector that you are using. Consult your Wireshark help resources for using the more powerful features of Wireshark tracing.

Profile resources

Use common performance testing methods to determine how fast the connector performs under a particular workload. Profiling the resources used under various work loads serve to pinpoint bugs relating to scalability, reliability, and resource usage.

Replicate in development environment

Replicate the production environment issue in your development environment and test for the same bug.

Reproduce without connector

Another simple test to determine if the connector is the source of the error, is to attempt to probe the remote resource without it. If you are unable to contact the remote resource without the connector, there may be a problem with your environment rather than with the connector. Common tools used to help in this regard include the following:

- Curl is a command line tool for sending and receiving files using URL syntax. Since Curl is used by many Watson Explorer Engine connectors, it is a great tool to help pinpoint the source of problems when crawling associated resource sites.
- Check that your problems are not browser specific. To do so, attempt to display search results in modern browsers such as Firefox, Internet Explorer, Chrome, and Safari. Test in the browser versions that are relevant to your users.
- Ping and Traceroute can be used to send packets of information to the remote data resource for the purpose of retrieving information, which can useful for testing your internet connection. Consult your operating system documentation on how to locate and execute the ping and traceroute utilities that are available in your environment.

Adjust crawler delay

In **Global Settings** > **Crawler Aggressiveness**, set the **Delay value** to `1`. This will increase requests on your server to help identify potential problems.

Note: We do not recommend setting the delay to `0`. Doing so can cause excessive resource usage on your crawling server, repository server, or both.

Validate web services

Check that all web services are performing correctly and that all the needed web services are activated in the server(s) where the data you are crawling is hosted. You can use a Web test to test Web services. Check online resources for writing specific web tests based on your environment.

Chapter 8. Release Notes

The Salesforce connector release notes are comprised of a change log of version-specific additions and fixes, and a known issues section.

Change Log

The this is the initial version of the Salesforce connector. There are no additions and fixes to report for the change log.

Known Issues

This is the initial version of the Salesforce connector. There are no known issues to report for this version of the Salesforce connector.