

ManifoldCF- End-user Documentation

Table of contents

1 Overview.....	4
1.1 Defining Output Connections.....	5
1.2 Defining Transformation Connections.....	8
1.3 Defining Authority Groups.....	11
1.4 Defining Repository Connections.....	12
1.5 Defining Notification Connections.....	16
1.6 Defining User Mapping Connections.....	19
1.7 Defining Authority Connections.....	22
1.8 Creating Jobs.....	25
1.9 Executing Jobs.....	31
1.10 Status Reports.....	32
1.11 History Reports.....	35
1.12 A Note About Credentials.....	38
2 Output Connection Types.....	38
2.1 Amazon Cloud Search Output Connection.....	38
2.2 CMIS Output Connection.....	39
2.3 ElasticSearch Output Connection.....	41
2.4 File System Output Connection.....	43
2.5 HDFS Output Connection.....	44
2.6 MetaCarta GTS Output Connection.....	44
2.7 Null Output Connection.....	45
2.8 OpenSearchServer Output Connection.....	45
2.9 Solr Output Connection.....	47

3 Transformation Connection Types.....	50
3.1 Allowed Documents.....	50
3.2 Metadata Adjuster.....	51
3.3 Null Transformer.....	53
3.4 Tika Content Extractor.....	53
4 User Mapping Connection Types.....	54
4.1 Regular Expression User Mapping Connection.....	54
5 Authority Connection Types.....	55
5.1 Active Directory Authority Connection.....	55
5.2 Alfresco Webscript Authority Connection.....	56
5.3 CMIS Authority Connection.....	58
5.4 EMC Documentum Authority Connection.....	58
5.5 Generic Authority.....	60
5.6 Generic Database Authority Connection.....	61
5.7 LDAP Authority Connection.....	63
5.8 OpenText LiveLink Authority Connection.....	65
5.9 Autonomy Meridio Authority Connection.....	66
5.10 Microsoft SharePoint ActiveDirectory Authority Connection....	69
5.11 Microsoft SharePoint Native Authority Connection.....	71
6 Repository Connection Types.....	73
6.1 Alfresco Repository Connection.....	73
6.2 Alfresco Webscript Repository Connection.....	75
6.3 CMIS Repository Connection.....	77
6.4 EMC Documentum Repository Connection.....	79
6.5 Dropbox Repository Connection.....	82
6.6 Individual Email Repository Connection.....	85
6.7 IBM FileNet P8 Repository Connection.....	87
6.8 Generic WGET-Compatible File System Repository Connection...	88
6.9 Generic Connector.....	90

6.10 Generic Database Repository Connection.....	92
6.11 Google Drive Repository Connection.....	99
6.12 HDFS Repository Connection (WGET compatible).....	105
6.13 Jira Repository Connection.....	107
6.14 OpenText LiveLink Repository Connection.....	108
6.15 Autonomy Meridio Repository Connection.....	112
6.16 Generic RSS Repository Connection.....	115
6.17 Microsoft SharePoint Repository Connection.....	121
6.18 Generic Web Repository Connection.....	133
6.19 Windows Share/DFS Repository Connection.....	145
6.20 Wiki Repository Connection.....	150
7 Notification Connection Types.....	150
7.1 Slack Notifications.....	150
7.2 Rocket.Chat Notifications.....	152

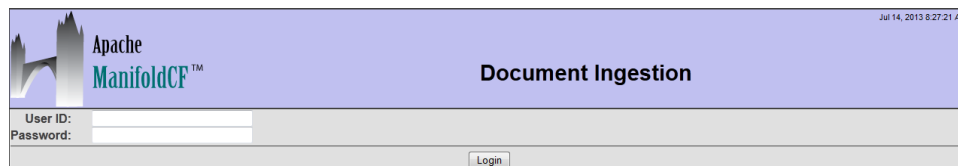
1 Overview

This manual is intended for an end-user of ManifoldCF. It is assumed that the Framework has been properly installed, either by you or by a system integrator, with all required services running and desired connection types properly registered. If you think you need to know how to do that yourself, please visit the "Developer Resources" page.

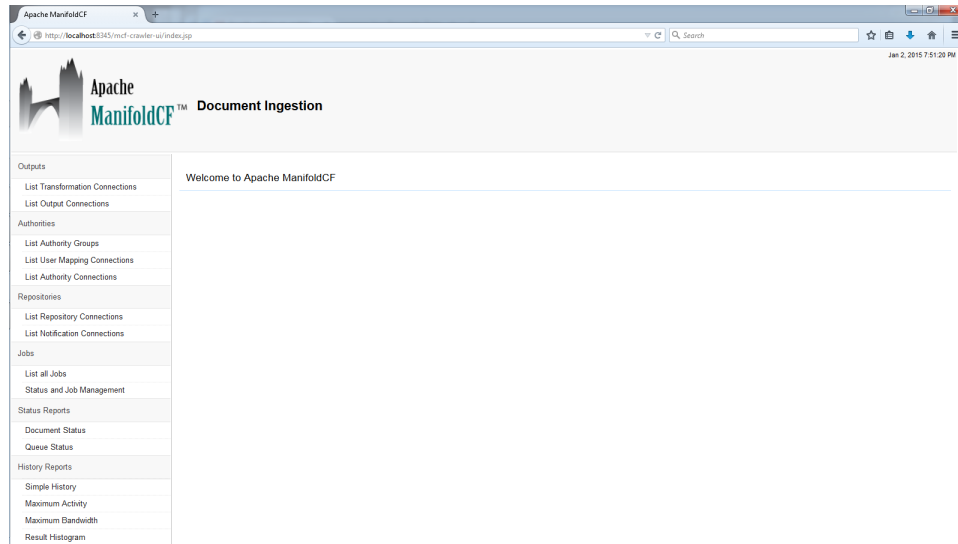
Most of this manual describes how to use the ManifoldCF user interface. On a standard ManifoldCF deployment, you would reach that interface by giving your browser a URL something like this: `http://my-server-name:8345/mcf-crawler-ui`. This will, of course, differ from system to system. Please contact your system administrator to find out what URL is appropriate for your environment.

The ManifoldCF UI has been tested with Firefox and various incarnations of Internet Explorer. If you use another browser, there is a small chance that the UI will not work properly. Please let your system integrator know if you find any browser incompatibility problems.

When you enter the Framework user interface the first time, you will first be asked to log in:



Enter the login user name and password for your system. By default, the user name is "admin" and the password is "admin", although your system administrator can (and should) change this. Then, click the "Login" button. If you entered the correct credentials, you should see a screen that looks something like this:



On the left, there are menu options you can select. The main pane on the right shows a welcome message, but depending on what you select on the left, the contents of the main pane will change. Before you try to accomplish anything, please take a moment to read the descriptions below of the menu selections, and thus get an idea of how the Framework works as a whole.

1.1 Defining Output Connections

The Framework UI's left-side menu contains a link for listing output connections. An output connection is a connection to a system or place where documents fetched from various repositories can be written to. This is often a search engine.

All jobs must specify an output connection. You can create an output connection by clicking the "List Output Connections" link in the left-side navigation menu. When you do this, the following screen will appear:

Outputs		List of Output Connections			
List Output Connections					
Authorities					
List Authority Connections					

On a freshly created system, there may well be no existing output connections listed. If there are already output connections, they will be listed on this screen, along with links that allow you to view, edit, or delete them. To create a new output connection, click the "Add new

output connection" link at the bottom. The following screen will then appear:

 A screenshot of the 'Edit an Output Connection' dialog box. On the left is a sidebar with 'Outputs' selected, containing links for 'List Output Connections' and 'Authorities'. The main panel has tabs for 'Name' and 'Type', with 'Name' currently active. It contains input fields for 'Name:' and 'Description:', and a 'Cancel' button at the bottom right.

The tabs across the top each present a different view of your output connection. Each tab allows you to edit a different characteristic of that connection. The exact set of tabs you see depends on the connection type you choose for the connection.

Start by giving your connection a name and a description. Remember that all output connection names must be unique, and cannot be changed after the connection is defined. The name must be no more than 32 characters long. The description can be up to 255 characters long. When you are done, click on the "Type" tab. The Type tab for the connection will then appear:

 A screenshot of the 'Edit an Output Connection' dialog box, now on the 'Type' tab. The 'Connection type:' dropdown menu is open, showing a list of options: 'ElasticSearch', 'ElasticSearch', 'MetaCarta GTS', 'Null', 'OpenSearchServer', and 'Solr'. 'Continue' and 'Cancel' buttons are visible at the bottom right.

The list of output connection types in the pulldown box, and what they are each called, is determined by your system integrator. The configuration tabs for each different kind of output connection type are described in separate sections below.

After you choose an output connection type, click the "Continue" button at the bottom of the pane. You will then see all the tabs appropriate for that kind of connection appear, and a "Save" button will also appear at the bottom of the pane. You must click the "Save" button when you are done in order to create your connection. If you click "Cancel" instead, the new connection will not be created. (The same thing will happen if you click on any of the navigation links in the left-hand pane.)

Every output connection has a "Throttling" tab. The tab looks like this:

 A screenshot of the 'Edit output connection' dialog box, now on the 'Throttling' tab. The sidebar is the same. The main panel has tabs for 'Name', 'Type', and 'Throttling', with 'Throttling' currently active. It contains a label 'Max connections (per JVM):' followed by a text input field containing the number '10'. 'Save' and 'Cancel' buttons are at the bottom right.

On this tab, you can specify only one thing: how many open connections are allowed at any given time to the system the output connection talks with. This restriction helps prevent that system from being overloaded, or in some cases exceeding its license limitations. Conversely, making

this number larger allows for greater overall throughput. The default value is 10, which may not be optimal for all types of output connections. Please refer to the section of the manual describing your output connection type for more precise recommendations.

Please refer to the section of the manual describing your chosen output connection type for a description of the tabs appropriate for that connection type.

After you save your connection, a summary screen will be displayed that describes your connection's configuration. This looks something like this (although the details will differ somewhat based on what connection type you chose):

View Output Connection Status	
Name:	null
Description:	
Connection type:	Null
Max connections:	10
Connection status: Connection working	
Refresh Edit Delete Re-index all associated documents Remove all associated records	

The summary screen contains a line where the connection's status is displayed. If you did everything correctly, the message "Connection working" will be displayed as a status. If there was a problem, you will see a connection-type-specific diagnostic message instead. If this happens, you will need to correct the problem, by either fixing your infrastructure, or by editing the connection configuration appropriately, before the output connection will work correctly.

Also note that there are five buttons along the bottom of the display: "Refresh", "Edit", "Delete", "Re-index all associated documents", and "Remove all associated records". We'll go into the purpose for each of these buttons in turn.

The "Refresh" button simply reloads the view page for the output connection, and updates the connection status. Use this button when you have made changes to the external system your output connection is connected to that might affect whether the connection will succeed or not.

The "Edit" button allows you to go back and edit the connection parameters. Use this button if you want to change the connection's characteristics or specifications in any way.

The "Delete" button allows you to delete the connection. Use this button if you no longer want the connection to remain in the available list of

output connections. Note that ManifoldCF will not allow you to delete a connection that is being referenced by a job.

The "Re-index all associated documents" button will nullify the recorded versions of all documents currently indexed via this connection. This is not a button you would use often. Click it when you have changed the configuration of whatever system the output connection is describing, and therefore all documents will eventually need to be reindexed.

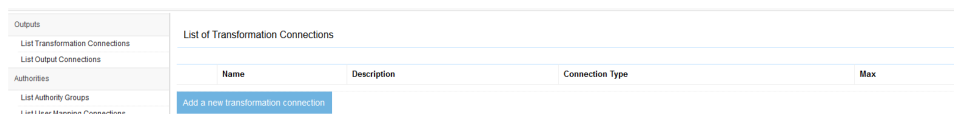
The "Remove all associated documents" button will remove from ManifoldCF all knowledge that any indexing has taken place at all to this connection. This is also not a button you would use often. Click it when you have removed the entire index that the output connection describes from the target repository.

1.2 Defining Transformation Connections

The Framework UI's left-side menu contains a link for listing transformation connections. A transformation connection is a connection to an engine where documents fetched from various repositories can be manipulated. This typically involves metadata extraction or mapping.

A job does not need to specify any transformation connections. In many cases, the final destination search engine has an included data conversion pipeline. But in the case where such data extraction and conversion is not available, ManifoldCF provides a way of taking care of it internally.

You can create a transformation connection by clicking the "List Transformation Connections" link in the left-side navigation menu. When you do this, the following screen will appear:



List of Transformation Connections			
Name	Description	Connection Type	Max
Add a new transformation connection			

On a freshly created system, there may well be no existing transformation connections listed. If there are already transformation connections, they will be listed on this screen, along with links that allow you to view, edit, or delete them. To create a new transformation connection, click the "Add new transformation connection" link at the bottom. The following screen will then appear:

The tabs across the top each present a different view of your transformation connection. Each tab allows you to edit a different characteristic of that connection. The exact set of tabs you see depends on the connection type you choose for the connection.

Start by giving your connection a name and a description. Remember that all transformation connection names must be unique, and cannot be changed after the connection is defined. The name must be no more than 32 characters long. The description can be up to 255 characters long. When you are done, click on the "Type" tab. The Type tab for the connection will then appear:

The list of transformation connection types in the pulldown box, and what they are each called, is determined by your system integrator. The configuration tabs for each different kind of transformation connection type are described in separate sections below.

After you choose a transformation connection type, click the "Continue" button at the bottom of the pane. You will then see all the tabs appropriate for that kind of connection appear, and a "Save" button will also appear at the bottom of the pane. You must click the "Save" button when you are done in order to create your connection. If you click "Cancel" instead, the new connection will not be created. (The same thing will happen if you click on any of the navigation links in the left-hand pane.)

Every transformation connection has a "Throttling" tab. The tab looks like this:

On this tab, you can specify only one thing: how many open connections are allowed at any given time to the system the transformation connection talks with. This restriction helps prevent that system from

being overloaded, or in some cases exceeding its license limitations. Conversely, making this number larger allows for greater overall throughput. The default value is 10, which may not be optimal for all types of output connections. Please refer to the section of the manual describing your transformation connection type for more precise recommendations.

Please refer to the section of the manual describing your chosen transformation connection type for a description of the tabs appropriate for that connection type.

After you save your connection, a summary screen will be displayed that describes your connection's configuration. This looks something like this (although the details will differ somewhat based on what connection type you chose):

View Transformation Connection Status	
Name:	NullTransformation
Description:	Null Transformation
Connection type:	Null
Max connections:	10
Connection status:	Connection working
Refresh Edit Delete	

The summary screen contains a line where the connection's status is displayed. If you did everything correctly, the message "Connection working" will be displayed as a status. If there was a problem, you will see a connection-type-specific diagnostic message instead. If this happens, you will need to correct the problem, by either fixing your infrastructure, or by editing the connection configuration appropriately, before the transformation connection will work correctly.

Also note that there are three buttons along the bottom of the display: "Refresh", "Edit", and "Delete". We'll go into the purpose for each of these buttons in turn.

The "Refresh" button simply reloads the view page for the transformation connection, and updates the connection status. Use this button when you have made changes to the external system your transformation connection is connected to that might affect whether the connection will succeed or not.

The "Edit" button allows you to go back and edit the connection parameters. Use this button if you want to change the connection's characteristics or specifications in any way.

The "Delete" button allows you to delete the connection. Use this button if you no longer want the connection to remain in the available list of

transformation connections. Note that ManifoldCF will not allow you to delete a connection that is being referenced by a job.

1.3 Defining Authority Groups

The Framework UI's left-side menu contains a link for listing authority groups. An authority group is a collection of authorities that all cooperate to furnish security for each document from repositories that you select. For example, a SharePoint 2010 repository with the Claims Based authorization feature enabled may contain documents that are authorized by SharePoint itself, by Active Directory, and by others. Documents from such a SharePoint repository would therefore refer to a authority group which would have a SharePoint native authority, a SharePoint Active Directory authority, and other SharePoint claims based authorities as members. But most of the time, an authority group will consist of a single authority that is appropriate for the repository the authority group is meant to secure.

Since you need to select an authority group when you define an authority connection, you should define your authority groups before setting up your authority connections. If you don't have any authority groups defined, you cannot create authority connections at all. But if you select the wrong authority group when setting up your authority connection, you can go back later and change your selection.

It is also a good idea to define your authority groups before creating any repository connections, since each repository connection will also need to refer back to an authority group in order to secure documents. While it is possible to change the relationship between a repository connection and its authority group after-the-fact, in practice such changes may cause many documents to be reindexed the next time an associated job is run.

You can create an authority group by clicking the "List Authority Groups" link in the left-side navigation menu. When you do this, the following screen will appear:

Outputs	List of Authority Groups	
List Output Connections		
Authorities	Name	Description
	Add a new authority group	

If there are already authority groups, they will be listed on this screen, along with links that allow you to view, edit, or delete them. To create a

new authority group, click the "Add a new authority group" link at the bottom. The following screen will then appear:

The tabs across the top each present a different view of your authority group. For authority groups, there is only ever one tab, the "Name" tab.

Give your authority group a name and a description. Remember that all authority group names must be unique, and cannot be changed after the authority group is defined. The name must be no more than 32 characters long. The description can be up to 255 characters long. When you are done, click on the "Save" button. You must click the "Save" button when you are done in order to create or update your authority group. If you click "Cancel" instead, the new authority group will not be created. (The same thing will happen if you click on any of the navigation links in the left-hand pane.)

After you save your authority group, a summary screen will be displayed that describes the group, and you can proceed on to create any authority connections that belong to the authority group, or repository connections that refer to the authority group.

1.4 Defining Repository Connections

The Framework UI's left-hand menu contains a link for listing repository connections. A repository connection is a connection to the repository system that contains the documents that you are interested in indexing.

All jobs require you to specify a repository connection, because that is where they get their documents from. It is therefore necessary to create a repository connection before indexing any documents.

A repository connection also may have an associated authority group. This specified authority group determines the security environment in which documents from the repository connection are attached. While it is possible to change the specified authority group for a repository connection after a crawl has been done, in practice this will require all documents associated with that repository connection be reindexed in order to be searchable by anyone. Therefore, we recommend that you set up your desired authority group before defining your repository connection.

You can create a repository connection by clicking the "List Repository Connections" link in the left-side navigation menu. When you do this, the following screen will appear:

Outputs	List of Repository Connections					
	List Output Connections					
		Name	Description	Connection Type	Authority	Max
	View Edit Delete	RSS		RSS	None(globalAuthority)	100
Authorities	List Authority Connections					
	Add new connection					

On a freshly created system, there may well be no existing repository connections listed. If there are already repository connections, they will be listed on this screen, along with links that allow you to view, edit, or delete them. To create a new repository connection, click the "Add a new connection" link at the bottom. The following screen will then appear:

The tabs across the top each present a different view of your repository connection. Each tab allows you to edit a different characteristic of that connection. The exact set of tabs you see depends on the connection type you choose for the connection.

Start by giving your connection a name and a description. Remember that all repository connection names must be unique, and cannot be changed after the connection is defined. The name must be no more than 32 characters long. The description can be up to 255 characters long. When you are done, click on the "Type" tab. The Type tab for the connection will then appear:

The list of repository connection types in the pulldown box, and what they are each called, is determined by your system integrator. The

configuration tabs for each different kind of repository connection type are described in this document in separate sections below.

You may also at this point select the authority group to use to secure all documents fetched from this repository with. You do not need to define your authority group's authority connections before doing this step, but you will not be able to search for your documents after indexing them until you do.

After you choose the desired repository connection type and an authority group (if desired), click the "Continue" button at the bottom of the pane. You will then see all the tabs appropriate for that kind of connection appear, and a "Save" button will also appear at the bottom of the pane. You must click the "Save" button when you are done in order to create or update your connection. If you click "Cancel" instead, the new connection will not be created. (The same thing will happen if you click on any of the navigation links in the left-hand pane.)

Every repository connection has a "Throttling" tab. The tab looks like this:

Bin regular expression	Description	Max avg fetches/min
	No throttling specified	

On this tab, you can specify two things. The first is how many open connections are allowed at any given time to the system the repository connection talks with. This restriction helps prevent that system from being overloaded, or in some cases exceeding its license limitations. Conversely, making this number larger allows for smaller average search latency. The default value is 10, which may not be optimal for all types of repository connections. Please refer to the section of the manual describing your authority connection type for more precise recommendations. The second specifies how rapidly, on average, the crawler will fetch documents via this connection.

Each connection type has its own notion of "throttling bin". A throttling bin is the name of a resource whose access needs to be throttled. For example, the Web connection type uses a document's server name as the throttling bin associated with the document, since (presumably) it will be access to each individual server that will need to be throttled independently.

On the repository connection "Throttling" tab, you can specify an unrestricted number of throttling descriptions. Each throttling description consists of a regular expression that describes a family of throttling bins, plus a helpful description, plus an average number of fetches per minute for each of the throttling bins that matches the regular expression. If a given throttling bin matches more than one throttling description, the most conservative fetch rate is chosen.

The simplest regular expression you can use is the empty regular expression. This will match all of the connection's throttle bins, and thus will allow you to specify a default throttling policy for the connection. Set the desired average fetch rate, and click the "Add" button. The throttling tab will then appear something like this:

If no throttle descriptions are added, no fetch-rate throttling will be performed.

Please refer to the section of the manual describing your chosen repository connection type for a description of the tabs appropriate for that connection type.

After you save your connection, a summary screen will be displayed that describes your connection's configuration. This looks something like this (although the details will differ somewhat based on what connection type you chose):

The summary screen contains a line where the connection's status is displayed. If you did everything correctly, the message "Connection working" will be displayed as a status. If there was a problem, you will see a connection-type-specific diagnostic message instead. If this happens, you will need to correct the problem, by either fixing your

infrastructure, or by editing the connection configuration appropriately, before the repository connection will work correctly.

Also note that there are four buttons along the bottom of the display: "Refresh", "Edit", "Delete", and "Clear all related history". We'll go into the purpose for each of these buttons in turn.

The "Refresh" button simply reloads the view page for the repository connection, and updates the connection status. Use this button when you have made changes to the external system your repository connection is connected to that might affect whether the connection will succeed or not.

The "Edit" button allows you to go back and edit the connection parameters. Use this button if you want to change the connection's characteristics or specifications in any way.

The "Delete" button allows you to delete the connection. Use this button if you no longer want the connection to remain in the available list of repository connections. Note that ManifoldCF will not allow you to delete a connection that is being referenced by a job.

The "Clear all related history" button will remove all history data associated with the current repository connection. This is not a button you would use often. History data is used to construct reports, such as the "Simple History" report. It is valuable as a diagnostic aid to understand what the crawler has been doing. There is an automated way of configuring ManifoldCF to remove history that is older than a specified interval before the current time. But if you want to remove all the history right away, this button will do that.

1.5 Defining Notification Connections

The Framework UI's left-side menu contains a link for listing notification connections. A notification connection is a connection to an engine that generates notification messages, such as email or text messages, specifically to note the end or unexpected termination of a job.

Jobs may specify one or more notification connections. You can create a notification connection by clicking the "List Notification Connections" link in the left-side navigation menu. When you do this, the following screen will appear:

On a freshly created system, there may well be no existing notification connections listed. If there are already notification connections, they will be listed on this screen, along with links that allow you to view, edit, or delete them. To create a new notification connection, click the "Add new notification connection" link at the bottom. The following screen will then appear:

The tabs across the top each present a different view of your notification connection. Each tab allows you to edit a different characteristic of that connection. The exact set of tabs you see depends on the connection type you choose for the connection.

Start by giving your connection a name and a description. Remember that all notification connection names must be unique, and cannot be changed after the connection is defined. The name must be no more than 32 characters long. The description can be up to 255 characters long. When you are done, click on the "Type" tab. The Type tab for the connection will then appear:

The list of notification connection types in the pulldown box, and what they are each called, is determined by your system integrator. The configuration tabs for each different kind of notification connection type are described in separate sections below.

After you choose a notification connection type, click the "Continue" button at the bottom of the pane. You will then see all the tabs appropriate for that kind of connection appear, and a "Save" button will also appear at the bottom of the pane. You must click the "Save" button when you are done in order to create your connection. If you click "Cancel" instead, the new connection will not be created. (The same

thing will happen if you click on any of the navigation links in the left-hand pane.)

Every notification connection has a "Throttling" tab. The tab looks like this:

On this tab, you can specify only one thing: how many open connections are allowed at any given time to the system the notification connection talks with. This restriction helps prevent that system from being overloaded, or in some cases exceeding its license limitations. Conversely, making this number larger allows for greater overall throughput. The default value is 10, which may not be optimal for all types of notification connections. Please refer to the section of the manual describing your notification connection type for more precise recommendations.

Please refer to the section of the manual describing your chosen notification connection type for a description of the tabs appropriate for that connection type.

After you save your connection, a summary screen will be displayed that describes your connection's configuration. This looks something like this (although the details will differ somewhat based on what connection type you chose):

The summary screen contains a line where the connection's status is displayed. If you did everything correctly, the message "Connection working" will be displayed as a status. If there was a problem, you will see a connection-type-specific diagnostic message instead. If this happens, you will need to correct the problem, by either fixing your infrastructure, or by editing the connection configuration appropriately, before the output connection will work correctly.

Also note that there are three buttons along the bottom of the display: "Refresh", "Edit", and "Delete". We'll go into the purpose for each of these buttons in turn.

The "Refresh" button simply reloads the view page for the notification connection, and updates the connection status. Use this button when you have made changes to the external system your output connection is connected to that might affect whether the connection will succeed or not.

The "Edit" button allows you to go back and edit the connection parameters. Use this button if you want to change the connection's characteristics or specifications in any way.

The "Delete" button allows you to delete the connection. Use this button if you no longer want the connection to remain in the available list of notification connections. Note that ManifoldCF will not allow you to delete a connection that is being referenced by a job.

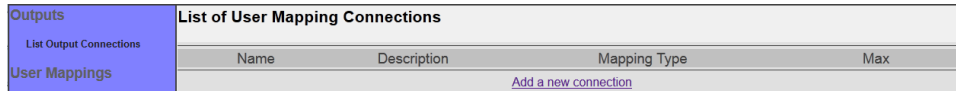
1.6 Defining User Mapping Connections

The Framework UI's left-side menu contains a link for listing user mapping connections. A user mapping connection is a connection to a system that understands how to map a user name into a different user name. For example, if you want to enforce document security using LiveLink, but you have only an Active Directory user name, you will need to map the Active Directory user name to a corresponding LiveLink one, before finding access tokens for it using the LiveLink Authority.

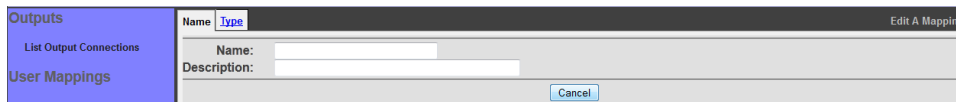
Not all user mapping connections need to access other systems in order to be useful. ManifoldCF, for instance, comes with a regular expression user mapper that manipulates a user name string using regular expressions alone. Also, user mapping is not needed for many, if not most, authorities. You will not need any user mapping connections if the authorities that you intend to create can all operate using the same user name, and that user name is in the form that will be made available to ManifoldCF's authority servlet at search time.

You should define your mapping connections before setting up your authority connections. An authority connections may specify a mapping connection that precedes it. For the same reason, it's also convenient to define your mapping connections in the order that you want to process the user name. If you don't manage to do this right the first time, though, there is no reason you cannot go back and fix things up.

You can create a mapping connection by clicking the "List User Mapping Connections" link in the left-side navigation menu. When you do this, the following screen will appear:



On a freshly created system, there may well be no existing mapping connections listed. If there are already mapping connections, they will be listed on this screen, along with links that allow you to view, edit, or delete them. To create a new mapping connection, click the "Add a new connection" link at the bottom. The following screen will then appear:



The tabs across the top each present a different view of your mapping connection. Each tab allows you to edit a different characteristic of that connection. The exact set of tabs you see depends on the connection type you choose for the connection.

Start by giving your connection a name and a description. Remember that all mapping connection names must be unique, and cannot be changed after the connection is defined. The name must be no more than 32 characters long. The description can be up to 255 characters long. When you are done, click on the "Type" tab. The Type tab for the connection will then appear:



The list of mapping connection types in the pulldown box, and what they are each called, is determined by your system integrator. The configuration tabs for each different kind of mapping connection type included with ManifoldCF are described in separate sections below.

After you choose a mapping connection type, click the "Continue" button at the bottom of the pane. You will then see all the tabs appropriate for that kind of connection appear, and a "Save" button will also appear at the bottom of the pane. You must click the "Save" button when you are done in order to create your connection. If you click "Cancel" instead, the new connection will not be created. (The same thing will happen if you click on any of the navigation links in the left-hand pane.)

Every mapping connection has a "Prerequisites" tab. This tab allows you to specify which mapping connection needs to be run before this one (if any). The tab looks like this:

Note: It is very important that you do not specify prerequisites in such a way as to create a loop. To make this easier, ManifoldCF will not display any user mapping connections in the pulldown which, if selected, would lead to a loop.

Every mapping connection has a "Throttling" tab. The tab looks like this:

On this tab, you can specify only one thing: how many open connections are allowed at any given time to the system the mapping connection talks with. This restriction helps prevent that system from being overloaded, or in some cases exceeding its license limitations. Conversely, making this number larger allows for smaller average search latency. The default value is 10, which may not be optimal for all types of mapping connections. Please refer to the section of the manual describing your mapping connection type for more precise recommendations.

Please refer to the section of the manual describing your chosen mapping connection type for a description of the tabs appropriate for that connection type.

After you save your connection, a summary screen will be displayed that describes your connection's configuration. This looks something like this (although the details will differ somewhat based on what connection type you chose):

The summary screen contains a line where the connection's status is displayed. If you did everything correctly, the message "Connection

working" will be displayed as a status. If there was a problem, you will see a connection-type-specific diagnostic message instead. If this happens, you will need to correct the problem, by either fixing your infrastructure, or by editing the connection configuration appropriately, before the mapping connection will work correctly.

Also note that there are three buttons along the bottom of the display: "Refresh", "Edit", and "Delete". We'll go into the purpose for each of these buttons in turn.

The "Refresh" button simply reloads the view page for the mapping connection, and updates the connection status. Use this button when you have made changes to the external system your mapping connection is connected to that might affect whether the connection will succeed or not.

The "Edit" button allows you to go back and edit the connection parameters. Use this button if you want to change the connection's characteristics or specifications in any way.

The "Delete" button allows you to delete the connection. Use this button if you no longer want the connection to remain in the available list of mapping connections. Note that ManifoldCF will not allow you to delete a connection that is being referenced by another mapping connection or an authority connection.

1.7 Defining Authority Connections

The Framework UI's left-side menu contains a link for listing authority connections. An authority connection is a connection to a system that defines a particular security environment. For example, if you want to index some documents that are protected by Active Directory, you would need to configure an Active Directory authority connection.

Bear in mind that only specific authority connection types are compatible with a given repository connection type. Read the details of your desired repository type in this document in order to understand how it is designed to be used. You may not need an authority if you do not mind that portions of all the documents you want to index are visible to everyone. For web, RSS, and Wiki crawling, this might be the situation. Most other repositories have some native security mechanism, however.

You can create an authority connection by clicking the "List Authority Connections" link in the left-side navigation menu. When you do this, the following screen will appear:

Outputs List Output Connections Authorities List Authority Connections	List of Authority Connections			
		View Edit Delete	Name	Description
			AD	Active Directory
	Add a new connection			

On a freshly created system, there may well be no existing authority connections listed. If there are already authority connections, they will be listed on this screen, along with links that allow you to view, edit, or delete them. To create a new authority connection, click the "Add a new connection" link at the bottom. The following screen will then appear:

Outputs List Output Connections Authorities List Authority Connections	Name	Type	Edit an Authority
	Name:		
	Description:		
	Cancel		

The tabs across the top each present a different view of your authority connection. Each tab allows you to edit a different characteristic of that connection. The exact set of tabs you see depends on the connection type you choose for the connection.

Start by giving your connection a name and a description. Remember that all authority connection names must be unique, and cannot be changed after the connection is defined. The name must be no more than 32 characters long. The description can be up to 255 characters long. When you are done, click on the "Type" tab. The Type tab for the connection will then appear:

Outputs List Output Connections Authorities List Authority Groups	Name	Type	Edit an Authority
	Connection type: Active Directory		
	Authority group: --Select a group--		
	Authorization domain: Default domain (None) Continue Cancel		

The list of authority connection types in the pulldown box, and what they are each called, is determined by your system integrator. The configuration tabs for each different kind of authority connection type are described in this document in separate sections below.

On this tab, you must also select the authority group that the authority connection you are creating belongs to. Select the appropriate authority group from the pulldown.

You also have the option of selecting a non-default authorization domain. An authorization domain describes which of possibly several user identities the authority connection is associated with. For example, a single user may have an Active Directory identity, a LiveLink identity, and a FaceBook identity. Your authority connection will be appropriate

to only one of those identities. The list of specific authorization domains available is determined by your system integrator.

After you choose an authority connection type, the authority group, and optionally the authorization domain, click the "Continue" button at the bottom of the pane. You will then see all the tabs appropriate for that kind of connection appear, and a "Save" button will also appear at the bottom of the pane. You must click the "Save" button when you are done in order to create your connection. If you click "Cancel" instead, the new connection will not be created. (The same thing will happen if you click on any of the navigation links in the left-hand pane.)

Every authority connection has a "Prerequisites" tab. This tab allows you to specify which mapping connection needs to be run before this one (if any). The tab looks like this:

Every authority connection also has a "Throttling" tab. The tab looks like this:

On this tab, you can specify only one thing: how many open connections are allowed at any given time to the system the authority connection talks with. This restriction helps prevent that system from being overloaded, or in some cases exceeding its license limitations. Conversely, making this number larger allows for smaller average search latency. The default value is 10, which may not be optimal for all types of authority connections. Please refer to the section of the manual describing your authority connection type for more precise recommendations.

Please refer to the section of the manual describing your chosen authority connection type for a description of the tabs appropriate for that connection type.

After you save your connection, a summary screen will be displayed that describes your connection's configuration. This looks something like this (although the details will differ somewhat based on what connection type you chose):

Outputs	View Authority Connection Status	
List Output Connections		
Authorities		
List Authority Groups	Name: MyAuthorityConnection	Description: My authority connection
List User Mapping Connections	Authority type: Null	Max connections: 10
List Authority Connections	Authority group: MyAuthorityGroup	Authorization domain:
Repositories	Prerequisite user mapping: No prerequisites	
List Repository Connections	Connection status: Connection working	
Jobs	Refresh Edit Delete	
List all Jobs		

The summary screen contains a line where the connection's status is displayed. If you did everything correctly, the message "Connection working" will be displayed as a status. If there was a problem, you will see a connection-type-specific diagnostic message instead. If this happens, you will need to correct the problem, by either fixing your infrastructure, or by editing the connection configuration appropriately, before the authority connection will work correctly.

Also note that there are three buttons along the bottom of the display: "Refresh", "Edit", and "Delete". We'll go into the purpose for each of these buttons in turn.

The "Refresh" button simply reloads the view page for the authority connection, and updates the connection status. Use this button when you have made changes to the external system your authority connection is connected to that might affect whether the connection will succeed or not.

The "Edit" button allows you to go back and edit the connection parameters. Use this button if you want to change the connection's characteristics or specifications in any way.

The "Delete" button allows you to delete the connection. Use this button if you no longer want the connection to remain in the available list of authority connections.

1.8 Creating Jobs

A "job" in ManifoldCF is a description of a set of documents. The Framework's job is to fetch this set of documents come from a specific repository connection, transform them using zero or more transformation connections, and send them to a specific output connection. The repository connection that is associated with the job will determine exactly how this set of documents is described, and to some degree how they are indexed. The output connection associated with the job can

also affect how each document is indexed, as will any transformation connections that are specified.

Every job is expected to be run more than once. Each time a job is run, it is responsible not only for sending new or changed documents to the output connection, but also for notifying the output connection of any documents that are no longer part of the set. Note that there are two ways for a document to no longer be part of the included set of documents: Either the document may have been deleted from the repository, or the document may no longer be included in the allowed set of documents. The Framework handles each case properly.

Deleting a job causes the output connection to be notified of deletion for all documents belonging to that job. This makes sense because the job represents the set of documents, which would otherwise be orphaned when the job was removed. (Some users make the assumption that a ManifoldCF job represents nothing more than a task, which is an incorrect assumption.)

Note that the Framework allows jobs that describe overlapping sets of documents to be defined. Documents that exist in more than one job are treated in the following special ways:

- When a job is deleted, the output connections are notified of deletion of documents belonging to that job only if they don't belong to another job
- The version of the document sent to an output connection depends on which job was run last

The subtle logic of overlapping documents means that you probably want to avoid this situation entirely, if it is at all feasible.

A typical non-continuous run of a job has the following stages of execution:

1. Adding the job's new, changed, or deleted starting points to the queue ("seeding")
2. Fetching documents, discovering new documents, and detecting deletions
3. Removing no-longer-included documents from the queue

Jobs can also be run "continuously", which means that the job never completes, unless it is aborted. A continuous run has different stages of execution:

1. Adding the job's new, changed, or deleted starting points to the queue ("seeding")
2. Fetching documents, discovering new documents, and detecting deletions, while reseeding periodically

Note that continuous jobs cannot remove no-longer-included documents from the queue. They can only remove documents that have been deleted from the repository.

A job can independently be configured to start when explicitly started by a user, or to run on a user-specified schedule. If a job is set up to run on a schedule, it can be made to start only at the beginning of a schedule window, or to start again within any remaining schedule window when the previous job run completes.

There is no restriction in ManifoldCF as to how many jobs many running at any given time.

You create a job by first clicking on the "List All Jobs" link on the left-side menu. The following screen will appear:

Name	Output Connection	Repository Connection	Schedule Type
CNN	Solr	RSS	Specified time

You may view, edit, or delete any existing jobs by clicking on the appropriate link. You may also create a new job that is a copy of an existing job. But to create a brand-new job, click the "Add a new job" link at the bottom. You will then see the following page:

Give your job a name. Note that job names do not have to be unique, although it is probably less confusing to have a different name for each one. Then, click the "Connection" tab:

Now, you should select the repository connection name. Bear in mind that whatever you select cannot be changed after the job is saved the first time.

Add an output, or more than one, by selecting the output in the pulldown, selecting the prerequisite pipeline stage, and clicking the "Add output" button. Note that once the job is saved the first time, you cannot delete an output. But you can rearrange your document processing pipeline in most other ways whenever you want to, including adding or removing transformation connections.

If you do not have any transformation connections defined, you will not be given the option of inserting a transformation connection into the pipeline. But if you have transformation connections defined, and you want to include them in the document pipeline, you can select them from the transformation connection pulldown, type a description into the description box, and then click one of the "Insert before" buttons to insert it into the document pipeline.

If you do not have any notification connections defined, you will not be given the option of adding one or more notifications to the end of the job. But if you have notification connections defined, and you want to include them, you can select them from the notification connection pulldown, type a description into the description box, and then click the appropriate "Add" button to add it into the notification list.

You also have the opportunity to modify the job's priority and start method at this time. The priority controls how important this job's documents are, relative to documents from any other job. The higher the number, the more important it is considered for that job's documents to be fetched first. The start method is as previously described; you get a choice of manual start, starting on the beginning of a scheduling window, or starting whenever possible within a scheduling window.

Make your selections, and click "Continue". The rest of the job's tabs will now appear, and a "Save" button will also appear at the bottom of the pane. You must click the "Save" button when you are done in order to create or update your job. If you click "Cancel" instead, the new job will not be created. (The same thing will happen if you click on any of the navigation links in the left-hand pane.)

All jobs have a "Scheduling" tab. The scheduling tab allows you to set up schedule-related configuration information:

The screenshot shows the 'Scheduling' tab in the ManifoldCF configuration interface. It includes fields for 'Schedule type', 'Recrawl interval', 'Maximum recrawl interval', 'Expiration interval', and 'Reseed interval'. Below these is a section for 'No schedule specified' and a 'Scheduled time' section with dropdowns for day of week, hour, minute, month, and day of month. The 'Maximum run time' is set to 0 minutes. The 'Job invocation' is set to 'Complete'.

On this tab, you can specify the following parameters:

- Whether the job runs continuously, or scans every document once
- How long a document should remain alive before it is 'expired', and removed from the index
- The minimum interval before a document is re-checked, to see if it has changed
- The maximum interval before a document is re-checked, to see if it has changed
- How long to wait before reseeding initial documents

The last four parameters only make sense if a job is a continuously running one, as the UI indicates.

The other thing you can do on this time is to define an appropriate set of scheduling records. Each scheduling record defines some related set of intervals during which the job can run. The intervals are determined by the starting time (which is defined by the day of week, month, day, hour, and minute pulldowns), and the maximum run time in minutes, which determines when the interval ends. It is, of course, possible to select multiple values for each of the pulldowns, in which case you be describing a starting time that had to match at least one of the selected values for each of the specified fields.

Once you have selected the schedule values you want, click the "Add Scheduled Time" button:

	Name	Connection	Scheduling	Forced Metadata	URLs	Canonicalization	URL Mappings	Exclusions	Time Values	Security	Metadata	Dechromed Content	Edit job test
Connections													
Groups													
Mapping Connections													
Repository Connections													
Jobs													
History Connections													
Jobs Management													
Jobs Status													
Jobs Activity													
Jobs Bandwidth													
Jobs Histogram													
Jobs Miscellaneous													
Jobs Help													
Jobs Log Out													

Schedule type:

Recrawl interval (if continuous): minutes (blank=infinity)

Maximum recrawl interval (if continuous): minutes (blank=infinity)

Expiration interval (if continuous): minutes (blank=infinity)

Reseed interval (if continuous): minutes (blank=infinity)

Scheduled time:

Maximum run time: minutes Job invocation:

Scheduled time:

Maximum run time: minutes Job invocation:

The example shows a schedule where crawls are run on Saturday and Sunday nights at 2 AM, and run for no more than 4 hours.

The rest of the job tabs depend on the types of the connections you selected. Please refer to the section of the manual describing the appropriate connection types corresponding to your chosen repository and output connections for a description of the job tabs that will appear for those connections.

After you save your job, a summary screen will be displayed that describes your job's specification. This looks something like this (although the details will differ somewhat based on what connections you chose):

Outputs	View a Job												
Last Transformation Connections													
List Output Connections													
Authorities													
List Authority Groups													
List User Mapping Connections													
List Authority Connections													
Repositories													
List Repository Connections													
Jobs													
List all Jobs													
Status and Job Management													
Status Reports													
Document Status													
Queue Status													
History Reports													
Simple History													
Maximum Activity													
Maximum Bandwidth													
Result Histogram													
Miscellaneous													
Help													
Log Out													

Name:

Stage:

Pipeline:

The "Edit" button allows you to go back and edit the job specification. Use this button if you want to change the job's details in any way.

The "Delete" button allows you to delete the job. Use this button if you no longer want the job to exist. Note that when you delete a job in ManifoldCF, all documents that were indexed using that job are removed from the index.

The "Copy" button allows you to edit a copy of the current job. Use this button if you want to create a new job that is based largely on the current job's specification. This can be helpful if you have many similar jobs to create.

The "Reset seeding" button will cause ManifoldCF to forget the seeding history of the job. Seeding is the process of discovering documents that have been added or modified. Clicking this button insures that ManifoldCF will examine all documents in the repository on the next crawl. This is not something that is done frequently; ManifoldCF is pretty good at managing this information itself, and will automatically do the same thing whenever a job specification is changed. Use this option if you've updated your connector software in a way that requires all documents to be re-examined.

1.9 Executing Jobs

You can follow what is going on, and control the execution of your jobs, by clicking on the "Status and Job Management" link on the left-side navigation menu. When you do, you might see something like this:

Outputs		Status of Jobs						
Authorities	List Output Connections							
		Name	Status	Start Time	End Time	Documents	Active	Processed
	Start Start minimal	CNN	Not yet run	Not started	Never run	0	0	0
	List Authority Connections	Refresh						

From here, you can click the "Refresh" link at the bottom of the main pane to see an updated status display, or you can directly control the job using the links in the leftmost status column. Allowed actions you may see at one point or another include:

- Start (start the job)
- Start minimal (start the job, but do only the minimal work possible)
- Abort (abort the job)
- Pause (pause the job)
- Resume (resume the job)
- Restart (equivalent to aborting the job, and starting it all over again)

- Restart minimal (equivalent to aborting the job, and starting it all over again, doing only the minimal work possible)

The columns "Documents", "Active", and "Processed" have very specific means as far as documents in the job's queue are concerned. The "Documents" column counts all the documents that belong to the job. The "Active" column counts all of the documents for that job that are queued up for processing. The "Processed" column counts all documents that are on the queue for the job that have been processed at least once in the past.

Using the "minimal" variant of the listed actions will perform the minimum possible amount of work, given the model that the connection type for the job uses. In some cases, this will mean that additions and modifications are indexed, but deletions are not detected. A complete job run is usually necessary to fully synchronize the target index with the repository contents.

1.10 Status Reports

Every job in ManifoldCF describes a set of documents. A reference to each document in the set is kept in a job-specific queue. It is sometimes valuable for diagnostic reasons to examine this queue for information. The Framework UI has several canned reports which do just that.

Each status report allows you to select what documents you are interested in from a job's queue based on the following information:

- The job
- The document identifier
- The document's status and state
- When the document is scheduled to be processed next

1.10.1 Document Status

A document status report simply lists all matching documents from within the queue, along with their state, status, and planned future activity. You might use this report if you were trying to figure out (for example) whether a specific document had been processed yet during a job run.

Click on the "Document Status" link on the left-hand menu. You will see a screen that looks something like this:

Select the desired connection. You may also select the desired document state and status, as well as specify a regular expression for the document identifier, if you want. Then, click the "Continue" button:

Select the job whose documents you want to see, and click "Continue" once again. The results will display:

Identifier	Job	State	Status	Scheduled	Scheduled Action	Retry Count	Retry Limit
http://rss.cnn.com/rss/edition_topstories.rss	CNN	Processed	Inactive		Process		
http://rss.cnn.com/rss/edition_topstories/302M-0EP2Cn/index.html	CNN	Processed	Inactive		Process		
http://rss.cnn.com/rss/edition_topstories/302D-0u-x1R/index.html	CNN	Processed	Inactive		Process		
http://rss.cnn.com/rss/edition_topstories/302E-33MMN0/index.html	CNN	Processed	Inactive		Process		
http://rss.cnn.com/rss/edition_topstories/302F-0m-0E/index.html	CNN	Processed	Inactive		Process		
http://rss.cnn.com/rss/edition_topstories/302G-0m-0E/index.html	CNN	Processed	Inactive		Process		
http://rss.cnn.com/rss/edition_topstories/302H-0m-0E/index.html	CNN	Processed	Inactive		Process		
http://rss.cnn.com/rss/edition_topstories/302I-0m-0E/index.html	CNN	Processed	Inactive		Process		
http://rss.cnn.com/rss/edition_topstories/302J-0m-0E/index.html	CNN	Processed	Inactive		Process		
http://rss.cnn.com/rss/edition_topstories/302K-0m-0E/index.html	CNN	Processed	Inactive		Process		

You may alter the criteria, and click "Go" again, if you so choose. Or, you can alter the number of result rows displayed at a time, and click "Go" to redisplay. Finally, you can page up and down through the results using the "Prev" and "Next" links.

1.10.2 Queue Status

A queue status report is an aggregate report that counts the number of occurrences of documents in specified classes. The classes are specified as a grouping within a regular expression, which is matched against all specified document identifiers. The results that are displayed are counts of documents. There will be a column for each combination of document state and status.

For example, a class specification of "(")" will produce exactly one result row, and will provide a count of documents that are in each state/status combination. A class description of "(.*)", on the other hand, will create one row for each document identifier, and will put a "1" in the column representing state and status of that document, with a "0" in all other column positions.

Click the "Queue Status" link on the left-hand menu. You will see a screen that looks like this:

Select the desired connection. You may also select the desired document state and status, as well as specify a regular expression for the document identifier, if you want. You will probably want to change the document identifier class from its default value of "(.*)". Then, click the "Continue" button:

Select the job whose documents you want to see, and click "Continue" once again. The results will display:

Identifier Class	Inactive	Processing	Expiring	Deleting	About to Process	About to Expire	Waiting for Processing	Waiting for Expiration	Waiting Forever
	11	0	0	0	0	0	0	0	0

Previous Next Rows: 0-END Rows per page: 20

You may alter the criteria, and click "Go" again, if you so choose. Or, you can alter the number of result rows displayed at a time, and click "Go" to redisplay. Finally, you can page up and down through the results using the "Prev" and "Next" links.

1.11 History Reports

For every repository connection, ManifoldCF keeps a history of what has taken place involving that connection. This history includes both events that the framework itself logs, as well as events that a repository connection or output connection will log. These individual events are categorized by "activity type". Some of the kinds of activity types that exist are:

- Job start
- Job end
- Job abort
- Various connection-type-specific read or access operations
- Various connection-type-specific output or indexing operations

This history can be enormously helpful in understand how your system is behaving, and whether or not it is working properly. For this reason, the Framework UI has the ability to generate several canned reports which query this history data and display the results.

All history reports allow you to specify what history records you are interested in including. These records are selected using the following criteria:

- The repository connection name
- The activity type(s) desired
- The start time desired
- The end time desired
- The identifier(s) involved, specified as a regular expression

- The result(s) produced, specified as a regular expression

The actual reports available are designed to be useful for diagnosing both access issues, and performance issues. See below for a summary of the types available.

1.11.1 Simple History Reports

As the name suggests, a simple history report does not attempt to aggregate any data, but instead just lists matching records from the repository connection's history. These records are initially presented in most-recent-first order, and include columns for the start and end time of the event, the kind of activity represented by the event, the identifier involved, the number of bytes involved, and the results of the event. Once displayed, you may choose to display more or less data, or reorder the display by column, or page through the data.

To get started, click on the "Simple History" link on the left-hand menu. You will see a screen that looks like this:

Now, select the desired repository connection from the pulldown in the upper left hand corner. If you like, you can also change the specified date/time range, or specify an identifier regular expression or result code regular expression. By default, the date/time range selects all events within the last hour, while the identifier regular expression and result code regular expression matches all identifiers and result codes.

Next, click the "Continue" button. A list of pertinent activities should then appear in a pulldown in the upper right:

1.11.4 Result Histogram Reports

A result histogram report is an aggregate report is used to count the occurrences of each kind of matching result for all matching events. MHL

1.12 A Note About Credentials

If any of your selected connection types require credentials, you may find it necessary to approach your system administrator to obtain an appropriate set. System administrators are often reluctant to provide accounts and credentials that have any more power than is utterly necessary, and sometimes not even that. Great care has been taken in the development of all connection types to be sure they require no more privilege than is utterly necessary. If a security-related warning appears when you view a connection's status, you must inform the system administrator that the credentials are inadequate to allow the connection to accomplish its task, and work with him/her to correct the problem.

2 Output Connection Types

2.1 Amazon Cloud Search Output Connection

The Amazon Cloud Search output connection type send documents to a specific path within a specified Amazon Cloud Search instance. The connection type furthermore "batches" documents to reduce cost as much as is reasonable. As a result, some specified documents may be sent at the end of a job run, rather than at the time they would typically be indexed.

The connection configuration information for the Amazon Cloud Search Output Connection type includes one additional tab: the "Server" tab. This tab looks like this:

You must supply the "Server host" field in order for the connection to work.

The Amazon Cloud Search Output Connection type does not contribute any tabs to a job definition.

The Amazon Cloud Search Output Connection type can only accept text content that is encoded in a UTF-8-compatible manner. It is highly recommended to use the Tika Content Extractor in the pipeline prior to the Amazon Cloud Search Output Connection type in order to convert documents to an indexable form.

In order to successfully index ManifoldCF documents in Amazon Cloud Search, you will need to describe a Cloud Search schema for receiving them. The fields that the Amazon Cloud Search output connection type sends are those that it gets specifically from the document as it comes through the ManifoldCF pipeline, with the addition of two hard-wired fields: "f_bodytext", containing the document body content, and "document_uri", containing the document's URI. You may also need to use the Metadata Adjuster transformation connection type to make sure that document metadata sent to Amazon Cloud Search agree with the schema you have defined there. Please refer to this document for details of how to set up an Amazon Cloud Search schema.

2.2 CMIS Output Connection

The CMIS Output Connection type allows you to migrate content to any CMIS-compliant repository.

By default each CMIS Connection manages a single CMIS repository, this means that if you have multiple CMIS repositories exposed by a single endpoint, you need to create a specific connection for each CMIS repository.

CMIS repository documents are typically secured by using the CMIS Authority Connection type. This authority type, however, does not have access to user groups, since there is no such functionality in the CMIS specification at this time. As a result, most people only use the CMIS connection type in an unsecured manner.

A CMIS Output connection has the following configuration parameters on the output connection editing screen:

The screenshot shows the 'Edit an Output Connection' window in the Apache ManifoldCF interface. The 'Server' tab is selected, displaying configuration fields for an output connection. The fields include: Binding (AtomPub), Username (admin), Password (masked with dots), Protocol (http), Server (localhost), Port (9091), Path (/chemistry-spifcms-server:inmemo), CMIS Query - Target folder (SELECT * FROM cmis:folder WHERE cmis:name='Apache ManifoldCF'), Create Timestamp Tree (Disabled), and Repository ID (optional). The 'Save' and 'Cancel' buttons are at the bottom.

Select the correct CMIS binding protocol (AtomPub or Web Services) and enter the correct username, password and the endpoint to reference the CMIS document server services.

The endpoint consists of the HTTP protocol, hostname, port and the context path of the CMIS service exposed by the CMIS server:

`http://HOSTNAME:PORT/CMIS_CONTEXT_PATH`

The CMIS Query must be provided to select your own drop zone in the target folder that should be an existent CMIS folder.

By default the crawler will replicate the same source path structure for each content in that target folder.

Considering to have your contents in your source repository inside the following path:

`/MySourceRepo/Invoices`

And supposing to have configured your CMIS Output Connection with the default value of the CMIS Query for your target folder:

```
SELECT * FROM cmis:folder WHERE cmis:name='Apache ManifoldCF'
```

All the migrated contents will be dropped in the following target CMIS folder:

`/path/to/your/Apache ManifoldCF/MySourceRepo/Invoices`

Optionally you can provide the repository ID to select one of the exposed CMIS repository, if this parameter is null the CMIS Connector will consider the first CMIS repository exposed by the CMIS server.

Note that, in a CMIS system, a specific binding protocol has its own context path, this means that the endpoints are different:

for example the endpoint of the AtomPub binding exposed by the actual version of the InMemory Server provided by the OpenCMIS framework is the following:

`http://localhost:8080/chemistry-opencmis-server-inmemory-war-0.5.0-SNAPSHOT/atom`

The Web Services binding is exposed using a different endpoint:

`http://localhost:8080/chemistry-opencmis-server-inmemory-war-0.5.0-SNAPSHOT/services/RepositoryService`

2.3 ElasticSearch Output Connection

The ElasticSearch Output Connection type allows ManifoldCF to submit documents to an ElasticSearch instance, via the XML over HTTP API. The connector has been designed to be as easy to use as possible.

After creating an ElasticSearch output connection, you have to populate the parameters tab. Fill in the fields according your ElasticSearch configuration. Each ElasticSearch output connector instance works with one index. To work with multiple indexes, just create one output connector for each index.

Name	Type	Throttling	Parameters
List Output Connections List Authority Connections Server Location (URL): Index name: Index type: http://localhost:9200/ test attachment Save Cancel			

The parameters are:

- Server location: An URL that references your ElasticSearch instance. The default value (`http://localhost:9200`) is valid if your ElasticSearch instance runs on the same server than the ManifoldCF instance.
- Index name: The connector will populate the index defined here.

Once you created a new job, having selected the ElasticSearch output connector, you will have the ElasticSearch tab. This tab let you:

- Fix the maximum size of a document before deciding to index it. The value is in bytes. The default value is 16MB.
- The allowed mime types. Warning it does not work with all repository connectors.

- The allowed file extensions. Warning it does not work with all repository connectors.

In the history report you will be able to monitor all the activities. The connector supports three activities: Document ingestion (Indexation), document deletion and index optimization. The targeted index is automatically optimized when the job is ending.

Outputs List Output Connections Authorities List Authority Connections Repositories List Repository Connections Jobs List all Jobs Status and Job Management Status Reports Document Status Queue Status History Reports	Simple History Report									
	Connection: -- Not specified --		Activities:		Deletion (ElasticSearch) Indexation (ElasticSearch) Optimize (ElasticSearch)					
	Start Time		1 pm		14		January		17th	
	End time:		2 pm		15		February		18th	
	Entity match:		3 pm		16		March		19th	
	Result code match:		-- Not specified --		-- Not specified --		-- Not specified --		-- Not specified --	
	Go		0		1		on		on	
	Start Time		Activity		Identifier		Result Code		Bytes	
	03-19-2012 15:56:05.039		Optimize (ElasticSearch)		http://localhost:9200/test/_optimize		OK		0	
	03-19-2012 15:55:55.154		job stop		133216844989/Alfresco ingestion against an ElasticSearch ser...		OK		0	

You may also refer to ElasticSearch's user documentation. Especially important is the need to configure the ElasticSearch index mapping *before* you try to index anything. If you have not configured the ElasticSearch mapping properly, then the documents you send to ElasticSearch via ManifoldCF will not be parsed, and once you send a document to the index, you cannot fix this in ElasticSearch without discarding your index. Specifically, you will want a mapping that enables the attachment plug-in, for example something like this:

```
{
  "attachment" :
  {
    "properties" :
    {
      "file" :
```

```
{
  "type" : "attachment",
  "fields" :
  {
    "title" : { "store" : "yes" },
    "keywords" : { "store" : "yes" },
    "author" : { "store" : "yes" },
    "content_type" : { "store" : "yes" },
    "name" : { "store" : "yes" },
    "date" : { "store" : "yes" },
    "file" : { "term_vector":"with_positions_offsets", "store":"yes" }
  }
}
```

Obviously, you would want your mapping to have details consistent with your particular indexing task. You can change the mapping or inspect it using the *curl* tool, which you can download from <http://curl.haxx.se>. For example, to inspect the mapping for a version of Elasticsearch running locally on port 9200:

```
curl -XGET http://localhost:9200/index/_mapping
```

2.4 File System Output Connection

The File System output connection type allows ManifoldCF to store documents in a local filesystem, using the conventions established by the Unix utility called *wget*. Documents stored by this connection type will not include any metadata or security information, but instead consist solely of a binary file.

The connection configuration information for the File System output connection type includes no additional tabs. There is an additional job tab, however, called "Output Path". The tab looks like this:

Fill in the path you want the connection type to use to write the documents to. Then, click the "Save" button.

2.5 HDFS Output Connection

The HDFS output connection type allows ManifoldCF to store documents in HDFS, using the conventions established by the Unix utility called *wget*. Documents stored by this connection type will not include any metadata or security information, but instead consist solely of a binary file.

The connection configuration information for the HDFS output connection type includes one additional tab: the "Server" tab. This tab looks like this:

Fill in the name node URI and the user name. Both are required. For the HDFS output connection type, there is an additional job tab called "Output Path". The tab looks like this:

Fill in the path you want the connection type to use to write the documents to. Then, click the "Save" button.

2.6 MetaCarta GTS Output Connection

The MetaCarta GTS output connection type is designed to allow ManifoldCF to submit documents to an appropriate MetaCarta GTS search appliance, via the appliance's HTTP Ingestion API.

The connection type implicitly understands that GTS can only handle text, HTML, XML, RTF, PDF, and Microsoft Office documents. All other document types will be considered to be unindexable. This helps prevent jobs based on a GTS-type output connection from fetching data that is large, but of no particular relevance.

When you configure a job to use a GTS-type output connection, two additional tabs will be presented to the user: "Collections" and "Document Templates". These tabs allow per-job specification of these GTS-specific features.

More here later

2.7 Null Output Connection

The null output connection type is meant primarily to function as an aid for people writing repository connection types. It is not expected to be useful in practice.

The null output connection type simply logs indexing and deletion requests, and does nothing else. It does not have any special configuration tabs, nor does it contribute tabs to jobs defined that use it.

2.8 OpenSearchServer Output Connection

The OpenSearchServer Output Connection allow ManifoldCF to submit documents to an OpenSearchServer instance, via the XML over HTTP API. The connector has been designed to be as easy to use as possible.

After creating an OpenSearchServer output connection, you have to populate the parameters tab. Fill in the fields according your OpenSearchServer configuration. Each OpenSearchServer output connector instance works with one index. To work with multiple indexes, just create one output connector for each index.

Name	Type	Throttling	Parameters
Outputs List Output Connections Authorities List Authority Connections Repositories Server Location (URL): http://localhost:8080/ Index name: index User name: admin API Key: 54a51ee4f27cbbcb7a771352b980567f Save Cancel			

The parameters are:

- **Server location:** An URL that references your OpenSearchServer instance. The default value (`http://localhost:8080/`) is valid if your OpenSearchServer instance runs on the same server than the ManifoldCF instance.
- **Index name:** The connector will populate the index defined here.
- **User name and API Key:** The credentials required to connect to the OpenSearchServer instance. It can be left empty if no user has been created. The next figure shows where to find the user's informations in the OpenSearchServer user interface.

OpenSearchServer (community edition) Connected as: (logout) Visit the OSS web site Documentation

Indices Runtime Privileges

Create a new user

User name: Choose a name for the user that will be created

Password: Choose a password for the user that will be created

Confirm password: Retype the password

Admin user: ☒ Check the box if the user is admin

Can monitor: ☐ Check the box if the user is able to monitor

API Key: Key to be used with API

Add privileges on following index:

Index	Role
-------	------

Once you created a new job, having selected the OpenSearchServer output connector, you will have the OpenSearchServer tab. This tab let you:

- Fix the maximum size of a document before deciding to index it. The value is in bytes. The default value is 16MB.
- The allowed mime types. Warning it does not work with all repository connectors.
- The allowed file extensions. Warning it does not work with all repository connectors.

Outputs Authorities Repositories Jobs Status Reports History Reports

Name Connection Scheduling Hop Filters OpenSearchServer Paths Edit a Job

Max file size (bytes):

Allowed MIME Types :

- application/msword
- application/vnd.ms-excel
- application/vnd.openxmlformats-officedocument.wordprocessingml.document
- application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
- text/html
- application/pdf
- application/vnd.ms-powerpoint
- application/vnd.openxmlformats-officedocument.presentationml.presentation

Allowed file extensions:

- doc
- docx
- xlsx
- pptx
- html
- pdf
- odt
- ods
- rtf

In the history report you will be able to monitor all the activities. The connector supports three activities: Document ingestion (Indexation), document deletion and index optimization. The targeted index is automatically optimized when the job is ending.

Outputs		Simple History Report									
List Output Connections											
Authorities	Connection: -- Not specified -- RiskSystem										
	Activities: Deletion (OpenSearchServer) Indexation (OpenSearchServer) Optimize (OpenSearchServer)										
List Authority Connections	Start time: 9 am 55 July 7th 2009 10 am 56 August 6th 2010 12 am 57 September 8th 2011										
Repositories	End time: -- Not specified -- -- Not specified -- -- Not specified -- -- Not specified -- -- Not specified -- 12 am 1 January 1st 2005 1 am 1 February 2nd 2006										
List Repository Connections											
Jobs	Entity match: Result code match: Go										
List all Jobs											
Status and Job Management											
Status Reports											
Document Status Queue Status											
History Reports											
	<u>Start Time</u>	<u>Activity</u>	<u>Identifier</u>	<u>Result Code</u>	<u>Bytes</u>	<u>Time</u>	<u>Result Description</u>				
	09-09-2011 12:57:07.463	document	file:/Users/kweller/Desktop/OSS_demo/opensearch-server/debug-module	OK	0	1					
	09-09-2011 12:57:10.140	Indexation (OpenSearchServer)	file:/Users/kweller/Desktop/OSS_demo/opensearch-server/start.bat	OK	1227	93					
	09-09-2011 12:57:10.088	document	file:/Users/kweller/Desktop/OSS_demo/opensearch-server/start.bat	OK	1227	188					
	09-09-2011 12:57:16.521	Indexation (OpenSearchServer)	file:/Users/kweller/Desktop/obrowseview_demo.jar	OK	251	289					
	09-09-2011 12:57:16.377	Indexation (OpenSearchServer)	file:/Users/kweller/Desktop/OSS_demo/test.jar	OK	193	61					
	09-09-2011 12:57:16.387	document	file:/Users/kweller/Desktop/OSS_demo/test.jar	OK	193	61					
	09-09-2011 12:57:16.866	Indexation (OpenSearchServer)	file:/Users/kweller/Desktop/opensearch-server-job-parameters.PNG	OK	128773	169					
	09-09-2011 12:57:15.947	document	file:/Users/kweller/Desktop/OSS_demo/opensearch-server-job-parameters.PNG	OK	0	1					
	09-09-2011 12:57:15.842	document	file:/Users/kweller/Desktop/opensearchserver-job-parameters.PNG	OK	128773	275					
	09-09-2011 12:57:15.834	Indexation (OpenSearchServer)	file:/Users/kweller/Desktop/trngui test code review	OK	508071	675					
	09-09-2011 12:57:15.691	Indexation (OpenSearchServer)	file:/Users/kweller/Desktop/Demo%20to%20OSS%20to%20Enterprise%20demo capture 2.ppt	OK	150697	321					
	09-09-2011 12:57:05.687	document	file:/Users/kweller/Desktop/Demo OSS - Enterprise/cvrennet.cap	OK	150697	431					

You may also refer to the OpenSearchServer's user documentation.

2.9 Solr Output Connection

The Solr output connection type is designed to allow ManifoldCF to submit documents to either an appropriate Apache Solr instance, via the Solr HTTP API, or alternatively to a Solr Cloud cluster. The configuration parameters are initially set to appropriate default values for a stand-alone Solr instance.

When you create a Solr output connection, multiple configuration tabs appear. The first tab is the "Solr type" tab. Here you select whether you want your connection to communicate to a standalone Solr instance, or to a Solr Cloud cluster:

Select which kind of Solr installation you want to communicate with. Based on your selection, you can proceed to either the "Server" tab (if a standalone instance) or to the "ZooKeeper" tab (if a SolrCloud cluster). The "Server" tab allows you to configure the HTTP parameters appropriate for communicating with a standalone Solr instance:

If your Solr setup is a standalone instance, fill in the fields according to your Solr configuration. The Solr connection type supports only basic authentication at this time; if you have this enabled, supply the credentials as requested on the bottom part of the form.

The "Zookeeper" tab allows your to configure the connection type to communicate with a Solr Cloud cluster:

Here, add each ZooKeeper instance in the SolrCloud cluster to the list of ZooKeeper instances. The connection comes preconfigured with "localhost" as being a ZooKeeper instance. You may delete this if it is not the case.

The next tab is the "Schema" tab, which allows you to specify the names of various Solr fields into which the Solr connection type will place built-in document metadata:

The most important of these is the document identifier field, which **MUST** be present for the connection type to function. This field will be used to uniquely identify the document within Solr, and will contain the document's URL. The Solr connection type will treat this field as being a unique key for locating the indexed document for further modification or deletion. The other Solr fields are optional, and largely self-explanatory.

The next tab is the "Arguments" tab, which allows you to specify arbitrary arguments to be sent to Solr:

Fill in the argument name and value, and click the "Add" button. Bear in mind that if you add an argument with the same name as an existing one, it will replace the existing one with the new specified value. You can delete existing arguments by clicking the "Delete" button next to the argument you want to delete.

Use this tab to specify any and all desired Solr update request parameters. You can, for instance, add `update.chain=myChain` to select a specific document processing pipeline/chain to use for processing documents. See the Solr documentation for more valid arguments.

The next tab is the "Documents" tab, which allows you to do document filtering based on size and mime types. By specifying a maximum document length in bytes, you can filter out documents which exceed that size (e.g. 10485760 which is equivalent to 10 MB). If you only want to add documents with specific mime types, you can enter them into the "included mime types" field (e.g. "text/html" for filtering out all documents but HTML). The "excluded mime types" field is for excluding documents with specific mime types (e.g. "image/jpeg" for filtering out JPEG images). The tab looks like:

The fifth tab is the "Commits" tab, which allows you to control the commit strategies. As well as committing documents at the end of every job, an option which is enabled by default, you may also commit each document within a certain time in milliseconds (e.g. "10000" for committing within 10 seconds). The commit within strategy will leave the responsibility to Solr instead of ManifoldCF. The tab looks like:

Outputs	Name	Type	Throttling	Server	Schema	Arguments	Documents	Commits	Edit output connection 'test'
List Output Connections	Commit at end of every job: ☒								
Authorities	Commit each document within (ms):								Save Cancel

When you are done, don't forget to click the "Save" button to save your changes! When you do, a connection summary and status screen will be presented, which may look something like this:

Outputs	View Output Connection Status	
List Output Connections	Name: Solr	Description:
Authorities	Connection type: Solr	Max connections: 10
List Authority Connections	Server update handler=/update/extract Password=***** Server status handler=/admin/ping Realm= Server remove handler=/update	
Repositories	Parameters: Server port=8983 User ID= Solr core name= Server web application=solr Server protocol=http Server name=localhost Solr id field name=id	
List Repository Connections	Arguments: No arguments	
Jobs	Connection status: Transient error: IO exception checking. Connection refused	
List all Jobs	Refresh Edit Delete	
Status and Job Management	Re-ingest all associated documents	
Status Reports		
Document Status		
Queue Status		
History Reports		
Simple History		

Note that in this example, the Solr connection is not responding, which is leading to an error status message instead of "Connection working".

3 Transformation Connection Types

3.1 Allowed Documents

The Allowed Documents transformation filter is used to limit the documents that will be fetched and passed down the pipeline for indexing. The filter allows documents to be restricted by mime type, by extension, and by length.

It is important to note that these various methods of filtering rely on the upstream repository connection type to implement. Some repository connection types do not implement all of the available methods of filtering. For example, filtering by URL (and hence file extension) makes little sense in the context of a repository connection type whose URLs do not include a full file name.

As with all document transformers, more than one Allowed Documents transformation filter can be used in a single pipeline. This may be useful if other document transformers (such as the Tika Content Extractor, below) change the characteristics of the document being processed.

The Allowed Documents transformation connection type does not require anything other than standard configuration information.

The Allowed Documents transformation connection type contributes a single tab to a job definition. This is the "Allowed Contents" tab, which looks like this:

Fill in the maximum desired document length, the set of extensions that are allowed, and the set of mime types that are allowed. All extensions and mime types are case insensitive. For extensions, the special value "." matches a missing or empty extension.

3.2 Metadata Adjuster

The Metadata Adjuster transformation filter reorganizes and concatenates metadata based on rules that you provide. This can be very helpful in many contexts. For example, you might use the Metadata Adjuster to label all documents from a particular job with a particular tag in an index. Or, you might need to map metadata from (say) SharePoint's schema to your final output connection type's schema. The Metadata Adjuster permits you to handle both of the scenarios.

As with all document transformers, more than one Metadata Adjuster transformation filter can be used in a single pipeline. This may be useful if other document transformers (such as the Tika Content Extractor, below) change the metadata of the document being processed.

The Metadata Adjuster transformation connection type does not require anything other than standard configuration information.

The Metadata Adjuster transformation connection type contributes one tab to a job definition. This is the "Metadata expressions" tab. The "Metadata expressions" tab looks like this:

On the left, you must supply the target metadata name. You may then choose among two possibilities: either you choose to request that this metadata field be removed from the target document, or you can choose to specify a rule that would provide a value for that target metadata item. You can provide more than one rule for the same metadata item, by simply adding additional lines to the table with the same target metadata name. But if you specify a rule to remove the metadata field from the document, that selection overrides all others.

Metadata rules are just strings, which may have field specifications within them. For example, the rule value "hello" will generate a single-valued metadata field with the value "hello". The rule "hello $\${there}$ ", on the other hand, will generate a value for each value in the incoming field named "there". If the incoming field had the values "a", "b", and "c", then the rule would generate a three-valued field with values "hello a", "hello b", and "hello c". If more than one incoming field is specified, then a combinatoric combination of the field values will be produced.

You can also use regular expressions in the substitution string, for example: " $\${there}[0-9]*$ ", which will extract the first sequence of sequential numbers it finds in the value of the field "there", or " $\${there}|string(.*)|1$ ", which will include everything following "string" in the field value. (The third argument specifies the regular expression group number, with an optional suffix of "l" or "u" meaning upper-case or lower-case.)

Enter a parameter name, and either select to remove the value or provide an expression. If you chose to supply an expression, enter the expression in the box. Then, click the "Add" button.

When your expressions have been edited, you can uncheck the "Keep all metadata" checkbox in order to prevent unspecified metadata fields from being passed through. Uncheck the "Remove empty metadata values" checkbox if you want to index empty metadata values.

3.3 Null Transformer

The null transformer does nothing other than record activity through the transformer. It is thus useful primarily as a coding model, and a diagnostic aid. It requires no non-standard configuration information, and provides no tabs for a job that includes it.

3.4 Tika Content Extractor

The Tika Content Extractor transformation filter converts a binary document into a UTF-8 text stream, plus metadata. This transformation filter is used primarily when incoming binary content is a possibility, or content that is not binary but has a non-standard encoding such as Shift-JIS. The Tika Content Extractor extracts metadata from the incoming stream as well. This metadata can be mapped within the Tika Content Extractor to metadata field names appropriate for further use downstream in the pipeline.

As with all document transformers, more than one Tika Content Extractor transformation filter can be used in a single pipeline. In the case of the Tika Content Extractor, this does not seem to be of much utility.

The Tika Content Extractor transformation connection type does not require anything other than standard configuration information.

The Tika Content Extractor transformation connection type contributes three tabs to a job definition. These are the "Field mapping" tab, the "Exceptions" tab, and the "Boilerplate" tab. The "Field mapping" tab looks like this:

The screenshot shows the 'Field mapping' tab of the Tika Content Extractor configuration. On the left is a sidebar with navigation links: Outputs, List Transformation Connections, List Output Connections, Authorities, List Authority Groups, List User Mapping Connections, List Authority Connections, Repositories, and List Repository Connections. The main area has tabs for Name, Connection, Scheduling, Hop Filters, Forced Metadata, Repository Paths, Allowed contents, Move metadata, Add metadata, and Field mapping (which is active). The 'Field mapping' tab contains a table with two columns: 'Metadata field name' and 'Final field name'. Above the table, it says 'No field mapping specified' with an 'Add' button. Below the table, there is a checkbox 'Keep all metadata' which is checked. At the bottom are 'Save' and 'Cancel' buttons.

Enter a Tika-generated metadata field name, and a final field name, and click the "Add" button to add the mapping to the list. Uncheck the "Keep all metadata" checkbox if you want unspecified Tika metadata to be excluded from the final document.

The "Exceptions" tab looks like this:

Uncheck the checkbox to allow indexing of document metadata even when Tika fails to extract content from the document.

The "Boilerplate" tab looks like this:

Select the HTML boilerplate removal option you want. These are implementations provided by the "Boilerpipe" project; they are lightly documented, so you will need to experiment with your particular application to find the one most appropriate for your application.

4 User Mapping Connection Types

4.1 Regular Expression User Mapping Connection

The Regular Expression user mapping connection type is very helpful for rote user name conversions of all sorts. For example, it can easily be configured to map the standard "user@domain" form of an Active Directory user name to (say) a LiveLink equivalent, e.g. "domain\user". Since many repositories establish such rote conversions, the Regular Expression user mapping connection type is often all that you will ever need.

A Regular Expression user mapping connection type has one special tab in the user mapping connection editing screen: "User Mapping". This tab looks like this:

The mapping consists of a match expression, which is a regular expression where parentheses "(" and ")" mark sections you are interested in, and a replace string. The sections marked with parentheses are called "groups" in regular expression parlance. The replace string consists of constant text plus substitutions of the groups from the match, perhaps modified. For example, "\$ (1)" refers to the first

group within the match, while "\$ (1l)" refers to the first match group mapped to lower case. Similarly, "\$ (1u)" refers to the same characters, but mapped to upper case.

For example, a match expression of `^(.*)\@([A-Z|a-z|0-9|_|-]*)\. (.*)$` with a replace string of `$(2)\$(1l)` would convert an Active Directory username of `MyUserName@subdomain.domain.com` into the user name `subdomain\myusername`.

When you are done, click the "Save" button. When you do, a connection summary and status screen will be presented, which may look something like this:

Outputs List Output Connections User Mappings List User Mapping Connections Authorities List Authority Connections Repositories List Repository Connections	View Mapping Connection Status		
	Name:	ll_regexp	Description: Livelink regexp mapping
	Mapping type:	Regular expression mapper	Max connections: 10
	Prerequisite user mapping:	No prerequisites	
	User name regular expression:	^(.*)\@([A-Z a-z 0-9 _ -]*)\. (.*)\$	
	User expression:	\$(2)\\$(1l)	
	Connection status:	Connection working	
		Refresh Edit Delete	

5 Authority Connection Types

5.1 Active Directory Authority Connection

An active directory authority connection is essential for enforcing security for documents from Windows shares, Microsoft SharePoint (in ActiveDirectory mode), and IBM FileNet repositories. This connection type needs to be provided with information about how to log into an appropriate Windows domain controller, with a user that has sufficient privileges to be able to look up any user's ID and group relationships.

An Active Directory authority connection type has two special tabs in the authority connection editing screen: "Domain Controller", and "Cache". The "Domain Controller" tab looks like this:

Outputs List Output Connections Authorities List Authority Connections	Name Type Throttling Domain Controller Cache Edit an Authority							
	Domain Controllers:		Domain controller name	Domain suffix	Administrative user name	Administrative password	Authentication	Login name AD attribute
	Add to End						DIGEST-MD	sAMAccountName
	Save Cancel							

As you can see, the Active Directory authority allows you to configure multiple connections to different, but presumably related, domain controllers. The choice of which domain controller will be accessed is determined by traversing the list of configured domain controllers from

top to bottom, and finding the first one that matches the domain suffix field specified. Note that a blank value for the domain suffix will match all users.

To add a domain controller to the end of the list, fill in the requested values. Note that the "Administrative user name" field usually requires no domain suffix, but depending on the details of how the domain controller is configured, may sometimes only accept the "name@domain" format. When you have completed your entry, click the "Add to end" button to add the domain controller rule to the end of the list. Later, when other domain controllers are present in the list, you can click a different button at an appropriate spot to insert the domain controller record into the list where you want it to go.

The Active Directory authority connection type also has a "Cache" tab, for managing the caching of individual user responses:

Here you can control how many individual users will be cached, and for how long.

When you are done, click the "Save" button. When you do, a connection summary and status screen will be presented, which may look something like this:

View Authority Connection Status						
Name:		AD				
Authority type:		Active Directory				
Description:		Max connections: 10				
Domain Controllers:						
Domain controller name	Domain suffix	Administrative user name	Administrative password	Authentication	Login name AD attribute	
localhost		foo	*****	DIGEST-MD5 GSSAPI	sAMAccountName	
Cache lifetime:		1 minutes				
Cache LRU size:		1000				
Connection status:		Threw exception: 'Couldn't communicate with domain controller 'localhost': localhost:389'				
Refresh Edit Delete						

Note that in this example, the Active Directory connection is not responding, which is leading to an error status message instead of "Connection working".

5.2 Alfresco Webscript Authority Connection

The Alfresco Webscript authority connection type helps secure documents indexed using the Alfresco Webscript repository connection type.

Independently of how the permissions schema is finally configured within the Alfresco instance, the Alfresco Webscript Authority service can retrieve the ACLs tokens associated to the users at request time. The connector is based on a single, secured service that directly enquires the Alfresco instance for the users permissions at all levels. The permissions tokens returned will be consistent with the Alfresco permissions model, therefore this Authority Connector makes sense to work only with the Alfresco Webscript Repository Connector and not any other connector

IMPORTANT: in order to put available the required services within Alfresco, it is necessary **FIRST** to install and deploy within the Alfresco instance the following Alfresco Webscript

. Please follow the instructions in the README file.

The Alfresco Webscript Authority Connection has a single configuration tab in the authority connection editing screen called "Server" where one needs to configure the Alfresco's services endpoint:

The screenshot shows a web-based configuration interface for an authority connection. At the top, there are tabs: 'Name', 'Type', 'Prerequisites', 'Throttling', and 'Server'. The 'Server' tab is selected. The form contains the following fields:

- Protocol: A dropdown menu showing 'http'.
- Host name: A text input field containing 'localhost'.
- Port: A text input field containing '8080'.
- Context: A text input field containing '/alfresco/service'.
- User name: A text input field containing 'admin'.
- Password: A text input field with masked characters (dots).

At the bottom of the form, there are two buttons: 'Save' and 'Cancel'.

As you can see, the Alfresco endpoint settings are quite simple and almost self-explicative:

- Protocol: HTTP or HTTPS depending on your instance
- HostName: IP or domain where the Alfresco instance is hosted in your network
- Port: port where the Alfresco instance has been deployed
- Context: URL path where the webscript services has been deployed (/alfresco/service by default) after installed the WebScript
- User: user ID used for performing the authored request to Alfresco. The user **MUST** have enough privileges for consulting any other user permissions, therefore admin user used to be a good idea
- Password: password of the above user in Alfresco

5.3 CMIS Authority Connection

A CMIS authority connection is required for enforcing security for documents retrieved from CMIS repositories.

The CMIS specification includes the concept of authorities only depending on a specific document, this authority connector is only based on a regular expression comparator.

A CMIS authority connection has the following special tab that you will need to configure: the "Repository" tab. The "Repository" tab looks like this:

Name	Type	Throttling	Repository	User Mapping
Endpoint: http://localhost:8080/cmis/				
Repository ID: uuid				
Save Cancel				

The repository configuration will be only used to track an ID for a specific CMIS repository. No calls will be performed against the CMIS repository.

When you are done, click the "Save" button. You will then see a summary and status for the authority connection:

View Authority Connection Status	
Name:	CMIS Authority
Description:	
Authority type:	CMIS
Max connections:	10
Parameters:	usermapping=(.*)=5(1) repositoryId=uuid endpoint=http://localhost:8080/cmis/
Connection status:	Connection working
Refresh Edit Delete	

5.4 EMC Documentum Authority Connection

A Documentum authority connection is required for enforcing security for documents retrieved from Documentum repositories.

This connection type needs to be provided with information about what Content Server to connect to, and the credentials that should be used to retrieve a user's ACLs from that machine. In addition, you can also specify whether or not you wish to include auto-generated ACLs in every user's list. Auto-generated ACLs are created within Documentum for every folder object. Because there are often a very large number of folders, including these ACLs can bloat the number of ManifoldCF access tokens returned for a user to tens of thousands, which can negatively impact performance. Even more notably, few Documentum installations make any real use of these ACLs in any way. Since Documentum's ACLs are purely additive (that is, there are no mechanisms for 'deny' semantics), the impact of a missing ACLs is only to block a user from

seeing something they otherwise could see. It is thus safe, and often desirable, to simply ignore the existence of these auto-generated ACLs. A Documentum authority connection has three special tabs you will need to configure: the "Docbase" tab, the "User Mapping" tab, and the "System ACLs" tab.

The "Docbase" tab looks like this:

Enter the desired Content Server docbase name, and enter the appropriate credentials. You may leave the "Domain" field blank if the Content Server you specify does not have Active Directory support enabled.

The purpose of the User Mapping tab is to control whether user lookups in Documentum are case sensitive or not. The "User Mapping" tab looks like this:

Here you can specify whether the mapping between incoming user names and Content Server user names is case sensitive or case insensitive. No other mappings are currently permitted. Typically, Documentum instances operate in conjunction with Active Directory, such that Documentum user names are either the same as the Active Directory user names, or are the Active Directory user names mapped to all lower case characters. You may need to consult with your Documentum system administrator to decide what the correct setting should be for this option.

The "System ACLs" tab looks like this:

Here, you can choose to ignore all auto-generated ACLs associated with a user. We recommend that you try ignoring such ACLs, and only choose the default if you have reason to believe that your Documentum content is protected in a significant way by the use of auto-generated ACLs. Your

may need to consult with your Documentum system administrator to decide what the proper setting should be for this option.

When you are done, click the "Save" button. When you do, a connection summary and status screen will be presented:

Outputs List Output Connections Authorities List Authority Connections Repositories List Repository Connections Jobs List all Jobs Status and Job Management	View Authority Connection Status	
	Name:	Documentum
	Authority type:	Documentum
	Description:	Max connections: 10
	Parameters:	usesystemacl=true docbasepassword=***** docbaseusername=username domain=somedomain.com usernamecaseinsensitive=false docbase=mydocbase
	Connection status:	Connection failed: Transient remote exception creating session: Connection refused to host: 127.0.0.1; nested exception is: java.net.ConnectException: Connection refused: connect
	Refresh Edit Delete	

Pay careful attention to the status, and be prepared to correct any problems that are displayed.

5.5 Generic Authority

Generic authority is intended to be used with Generic Connector and provide authentication tokens based on generic API. The idea is that you can use it and implement only the API which is designed to be fine grained and as simple as it is possible to handle all tasks.

API should be implemented as xml web page (entry point) returning results based on provided GET params. It may be a simple server script or part of the bigger application. API can be secured with HTTP basic authentication.

There are 2 actions:

- check
- auth

Action is passed as "action" GET param to the endpoint.

[endpoint]?action=check

Should return HTTP status code 200 providing information that endpoint is working properly. Any content returned will be ignored, only the status code matters.

[endpoint]?action=auth&username=UserName@Domain

Parameters:

- username - name of the user we want to resolve and fetch tokens.

Result should be valid XML of form:

```
<auth exists="true|false">
  <token>token_1</token>;
  <token>token_2</token>;
  ...
</auth>
```

`exists` attribute is required and it carries information whether user is valid or not.

5.6 Generic Database Authority Connection

The generic database connection type allows you to generate access tokens from a database table, served by one of the following databases:

- Postgresql (via a Postgresql JDBC driver)
- SQL Server (via the JTDS JDBC driver)
- Oracle (via the Oracle JDBC driver)
- Sybase (via the JTDS JDBC driver)
- MySQL (via the MySQL JDBC driver)

This connection type cannot be configured to work with other databases than the ones listed above without software changes. Depending on your particular installation, some of the above options may not be available.

A generic database authority connection has four special tabs on the repository connection editing screen: the "Database Type" tab, the "Server" tab, the "Credentials" tab, and the "Queries" tab. The "Database Type" tab looks like this:

Select the kind of database you want to connect to, from the pulldown.

Also, select the JDBC access method you want from the access method pulldown. The access method is provided because the JDBC specification has been recently clarified, and not all JDBC drivers work the same way as far as resultset column name discovery is concerned. The "by name" option currently works with all JDBC drivers in the list except for the MySQL driver. The "by label" works for the current MySQL driver, and may work for some of the others as well. If the queries you supply for your generic database jobs do not work correctly, and you see an error message about not being able to find required columns in the result,

you can change your selection on this pulldown and it may correct the problem.

The "Server" tab looks like this:

Here you have a choice. Either you can choose to specify the database host and port, and the database name or instance name, or you can provide a raw JDBC connection string that is appropriate for the database type you have chosen. This latter option is provided because many JDBC drivers, such as Oracle's, now can connect to an entire cluster of Oracle servers if you specify the appropriate connection description string.

If you choose the second option, just consult your JDBC driver's documentation and supply your string. If there is anything entered in the raw connection string field at all, it will take precedence over the database host and database name fields.

If you choose the first option, the server name and port must be provided in the "Database host and port" field. For example, for Oracle, the standard Oracle installation uses port 1521, so you would enter something like, "my-oracle-server:1521" for this field. PostgreSQL uses port 5432 by default, so "my-postgresql-server:5432" would be required. SQL Server's standard port is 1433, so use "my-sql-server:1433".

The service name or instance name field describes which instance and database to connect to. For Oracle or PostgreSQL, provide just the database name. For SQL Server, use "my-instance-name/my-database-name". For SQL Server using the default instance, use just the database name.

The "Credentials" tab is straightforward:

Enter the database user credentials.

The "Queries" tab looks like this:

Here you supply two queries. The first query looks up the user name to find a user id. The second query looks up access tokens corresponding to the user id. Details of what you supply for these queries will depend on your database schema.

After you click the "Save" button, you will see a connection summary screen, which might look something like this:

Note that in this example, the generic database authority connection is not properly authenticated, which is leading to an error status message instead of "Connection working".

5.7 LDAP Authority Connection

An LDAP authority connection can be used to provide document security in situations where there is no native document security model in place. Examples include Samba shares, Wiki pages, RSS feeds, etc.

The LDAP authority works by providing user or group names from an LDAP server as access tokens. These access tokens can be used by any repository connection type that provides for access tokens entered on a per-job basis, or by the JCIFs connection type, which has explicit user/group name support built in, meant for Samba shares.

This connection type needs to be provided with information about how to log into an appropriate LDAP server, as well as search expressions needed to look up user and group records. An active directory authority

connection type has a single special tab in the authority connection editing screen: the "LDAP" tab:

Fill in the requested values. Note that the "Server base" field contains the LDAP domain specification you want to search. For example, if you have an LDAP domain for "people.myorg.com", the server based might be "dc=com,dc=myorg,dc=people".

When you are done, click the "Save" button. When you do, a connection summary and status screen will be presented, which may look something like this:

Note that in this example, the LDAP connection is not responding, which is leading to an error status message instead of "Connection working".

Example configuration for ActiveDirectory server to fetch user groups:

- Server: [xxx.yyy.zzz.ttt]
- Port: 389
- Server base: [DC=domain,DC=name]
- Bind as user: [user@domain.name]
- Bind with password: [password for that user]
- User search base: CN=Users
- User search filter: sAMAccountName={0}
- User name attribute: sAMAccountName
- Group search base: CN=Users
- Group search filter: (member:1.2.840.113556.1.4.1941:={0})
- Group name attribute: sAMAccountName
- Member attribute is DN: yes (tick the checkbox)

member:1.2.840.113556.1.4.1941: gives you recursive check for nested groups

5.8 OpenText LiveLink Authority Connection

A LiveLink authority connection is needed to enforce security for documents retrieved from LiveLink repositories.

In order to function, this connection type needs to be provided with information about the name of the LiveLink server, and credentials appropriate for retrieving a user's ACLs from that machine. Since LiveLink operates with its own list of users, you may also want to specify a rule-based mapping between an Active Directory user and the corresponding LiveLink user. The authority type allows you to specify such a mapping using regular expressions.

A LiveLink authority connection has two special tabs you will need to configure: the "Server" tab and the "Cache" tab.

The "Server" tab looks like this:

Select the manner you want the connection to use to communicate with LiveLink. Your options are:

- Internal (native LiveLink protocol)
- HTTP (communication with LiveLink through the IIS web server)
- HTTPS (communication with LiveLink through IIS using SSL)

Also, you need to enter the name of the desired LiveLink server, the LiveLink port, and the LiveLink server credentials. If you have selected communication using HTTP or HTTPS, you must provide a relative CGI path to your LiveLink. You may also need to provide web server credentials. Basic authentication and older forms of NTLM are supported. In order to use NTLM, specify a non-blank server domain name in the "Server HTTP domain" field, plus a non-qualified user name and password. If basic authentication is desired, leave the "Server HTTP

domain" field blank, and provide basic auth credentials in the "Server HTTP NTLM user name" and "Server HTTP NTLM password" fields. For no web server authentication, leave these fields all blank.

For communication using HTTPS, you will also need to upload your authority certificate(s) on the "Server" tab, to tell the connection which certificates to trust. Upload your certificate using the browse button, and then click the "Add" button to add it to the trust store.

The "Cache" tab allows you to configure how the authority connection keeps an individual user's information around:

Here you set the time a user's information is kept around (the "Cache lifetime" field), and how many simultaneous users have their information cached (the "Cache LRU size" field).

When you are done, click the "Save" button. You will then see a summary and status for the authority connection:

We suggest that you examine the status carefully and correct any reported errors before proceeding. Note that in this example, the LiveLink server would not accept connections, which is leading to an error status message instead of "Connection working".

5.9 Autonomy Meridio Authority Connection

A Meridio authority connection is required for enforcing security for documents retrieved from Meridio repositories.

This connection type needs to be provided with information about what Document Server to connect to, what Records Server to connect to, and what User Service Server to connect to. Also needed are the Meridio

credentials that should be used to retrieve a user's ACLs from those machines.

Note that the User Service is part of the Meridio Authority, and must be installed somewhere in the Meridio system in order for the Meridio Authority to function correctly. If you do not know whether this has yet been done, or on what server, please ask your system administrator.

A Meridio authority connection has the following special tabs you will need to configure: the "Document Server" tab, the "Records Server" tab, the "User Service Server" tab, and the "Credentials" tab. The "Document Server" tab looks like this:

The screenshot shows the 'Document Server' tab of the Meridio configuration interface. The interface has a sidebar on the left with sections: 'Outputs' (containing 'List Output Connections'), 'Authorities' (containing 'List Authority Connections'), and 'Repositories' (containing 'List Repository Connections'). The main panel has tabs: 'Name', 'Type', 'Throttling', 'Document Server' (active), 'Records Server', 'User Service Server', and 'Credentials'. The 'Document Server' tab contains the following fields: 'Document webservice server protocol:' with a dropdown menu showing 'http'; 'Document webservice server name:' with an empty text box; 'Document webservice server port:' with an empty text box; 'Document webservice location:' with the value 'DMWS/MeridioDMWS.asmx'; 'Document webservice server proxy host:' with an empty text box; and 'Document webservice server proxy port:' with an empty text box. At the bottom right of the main panel are 'Save' and 'Cancel' buttons. An 'Edit an Authority' link is visible in the top right corner of the main panel.

Select the correct protocol, and enter the correct server name, port, and location to reference the Meridio document server services. If a proxy is involved, enter the proxy host and port. Authenticated proxies are not supported by this connection type at this time.

Note that, in the Meridio system, while it is possible that different services run on different servers, this is not typically the case. The connection type, on the other hand, makes no assumptions, and permits the most general configuration.

The "Records Server" tab looks like this:

The screenshot shows the 'Records Server' tab of the Meridio configuration interface. The interface is similar to the previous one, with the same sidebar and tabs. The 'Records Server' tab is active, showing the following fields: 'Record webservice server protocol:' with a dropdown menu showing 'http'; 'Record webservice server name:' with an empty text box; 'Record webservice server port:' with an empty text box; 'Record webservice location:' with the value 'RMWS/MeridioRMWS.asmx'; 'Record webservice server proxy host:' with an empty text box; and 'Record webservice server proxy port:' with an empty text box. At the bottom right of the main panel are 'Save' and 'Cancel' buttons. An 'Edit an Authority' link is visible in the top right corner of the main panel.

Select the correct protocol, and enter the correct server name, port, and location to reference the Meridio records server services. If a proxy is involved, enter the proxy host and port. Authenticated proxies are not supported by this connection type at this time.

Note that, in the Meridio system, while it is possible that different services run on different servers, this is not typically the case. The connection type, on the other hand, makes no assumptions, and permits the most general configuration.

The "User Service Server" tab looks like this:

You will require knowledge of where the special Meridio Authority extensions have been installed in order to fill out this tab.

Select the correct protocol, and enter the correct server name, port, and location to reference the Meridio user service server services. If a proxy is involved, enter the proxy host and port. Authenticated proxies are not supported by this connection type at this time.

Note that, in the Meridio system, while it is possible that different services run on different servers, this is not typically the case. The connection type, on the other hand, makes no assumptions, and permits the most general configuration.

The "Credentials" tab looks like this:

Enter the Meridio server credentials needed to access the Meridio system.

When you are done, click the "Save" button. You will then see a screen looking something like this:

In this example, logon has not succeeded because the server on which the Meridio Authority is running is unknown to the Windows domain under which Meridio is running. This results in an error message, instead of the "Connection working" message that you would see if the authority was working properly.

Since Meridio uses Windows IIS for authentication, there are many ways in which the configuration of either IIS or the Windows domain under which Meridio runs can affect the correct functioning of the Meridio Authority. It is beyond the scope of this manual to describe the kinds of analysis and debugging techniques that might be required to diagnose connection and authentication problems. If you have trouble, you will almost certainly need to involve your Meridio IT personnel. Debugging tools may include (but are not limited to):

- Windows security event logs
- ManifoldCF logs (see below)
- Packet captures (using a tool such as WireShark)

If you need specific ManifoldCF logging information, contact your system integrator.

5.10 Microsoft SharePoint ActiveDirectory Authority Connection

A Microsoft SharePoint ActiveDirectory authority connection is meant to furnish access tokens from Active Directory for a SharePoint instance that is configured to use Claims Based authorization. It cannot be used in any other situation.

The SharePoint ActiveDirectory authority is meant to work in conjunction with a SharePoint Native authority connection, and provides authorization information from one or more Active Directory domain controllers. Thus, it is only needed if Active Directory groups are used to furnish access to documents for users in the SharePoint system.

Documents must be indexed using a Microsoft SharePoint repository connection where the "Authority type" is specified to be "Native". If the "Authority type" is specified to be "Active Directory", then instead you should configure an Active Directory authority connection, described above.

This connection type needs to be provided with information about how to log into an appropriate Windows domain controller, with a user that has sufficient privileges to be able to look up any user's ID and group relationships.

A SharePoint Active Directory authority connection type has two special tabs in the authority connection editing screen: "Domain Controller", and "Cache". The "Domain Controller" tab looks like this:

As you can see, the SharePoint Active Directory authority allows you to configure multiple connections to different, but presumably related, domain controllers. The choice of which domain controller will be accessed is determined by traversing the list of configured domain controllers from top to bottom, and finding the first one that matches the domain suffix field specified. Note that a blank value for the domain suffix will match all users.

To add a domain controller to the end of the list, fill in the requested values. Note that the "Administrative user name" field usually requires no domain suffix, but depending on the details of how the domain controller is configured, may sometimes only accept the "name@domain" format. When you have completed your entry, click the "Add to end" button to add the domain controller rule to the end of the list. Later, when other domain controllers are present in the list, you can click a different button at an appropriate spot to insert the domain controller record into the list where you want it to go.

The SharePoint Active Directory authority connection type also has a "Cache" tab, for managing the caching of individual user responses:

Here you can control how many individual users will be cached, and for how long.

When you are done, click the "Save" button. When you do, a connection summary and status screen will be presented, which may look something like this:

Outputs	View Authority Connection Status						
List Output Connections							
Authorities	Name:	SharePoint Active Directory			Description:	SharePoint Active Directory authority	
List Authority Groups	Authority type:	SharePointActiveDirectory			Max connections:	10	
List User Mapping Connections	Authority group:	SharePoint			Authorization domain:		
List Authority Connections	Prerequisite user mapping: No prerequisites						
Repositories	Domain Controllers:	Domain controller name	Domain suffix	Administrative user name	Administrative password	Authentication	Login name AD attribute
List Repository Connections		mydomaincontroller	mydomain.com	to	-----	DIGEST-4/DIGEST-SHA1	sAMAccountName
Jobs	Cache lifetime:	1 minutes					
List all Jobs	Cache LRU size:	1000					
Status and Job Management	Connection status:	Threw exception: 'Couldn't communicate with domain controller 'mydomaincontroller'. mydomaincontroller:389'					
		Refresh Edit Delete					

Note that in this example, the SharePoint Active Directory connection is not responding, which is leading to an error status message instead of "Connection working".

5.11 Microsoft SharePoint Native Authority Connection

A Microsoft SharePoint Native authority connection is meant to furnish access tokens from the same SharePoint instance that the documents are coming from. You should use this authority type whenever you are trying to secure documents using a SharePoint repository connection that is configured to the use "Native" authority type.

If your SharePoint instance is configured to use the Claims Based authorization model, you may combine a SharePoint Native authority connection with other SharePoint authority types, such as the SharePoint ActiveDirectory authority type, to furnish complete authorization support. However, if Claims Based authorization is not configured, the SharePoint Native authority connection is the only authority type you should need to use.

A SharePoint authority connection has two special tabs on the authority connection editing screen: the "Server" tab, and the "Cache" tab. The "Server" tab looks like this:

Outputs	Name	Type	Prerequisites	Throttling	Server	Cache	Edit an Authority
List Output Connections							
Authorities	Server SharePoint version: SharePoint Services 2.0 (2003)						
List Authority Groups	Claims based authorization: Disabled						
List User Mapping Connections	Server protocol: http						
List Authority Connections	Server name: localhost						
Repositories	Server port:						
List Repository Connections	Site path:						
Jobs	User name:						
List all Jobs	Password:						
Status and Job Management	SSL certificate list: No certificates present						
Status Reports	Add Certificate: Browse... No file selected.						
Document Status	Save Cancel						
Queue Status							
History Reports							

Select your SharePoint server version from the pulldown. Check with your SharePoint system administrator if you are not sure what to select.

Select whether your SharePoint server is configured for Claims Based authorization or not. Check with your SharePoint system administrator if you are not sure what to select.

SharePoint uses a web URL model for addressing sites, subsites, libraries, and files. The best way to figure out how to set up a SharePoint connection type is therefore to start with your web browser, and visit the topmost root of the site you wish to crawl. Then, record the URL you see in your browser.

Select the server protocol, and enter the server name and port, based on what you recorded from the URL for your SharePoint site. For the "Site path" field, type in the portion of the root site URL that includes everything after the server and port, except for the final ".aspx" file. For example, if the SharePoint URL is "http://myserver:81/sites/somewhere/index.aspx", the site path would be "/sites/somewhere".

The SharePoint credentials are, of course, what you used to log into your root site. The SharePoint connection type always requires the user name to be in the form "domain\user".

If your SharePoint server is using SSL, you will need to supply enough certificates for the connection's trust store so that the SharePoint server's SSL server certificate can be validated. This typically consists of either the server certificate, or the certificate from the authority that signed the server certificate. Browse to the local file containing the certificate, and click the "Add" button.

The "Cache" tab looks like this:

The screenshot shows the 'Cache' configuration tab. On the left, there's a sidebar with 'Outputs' and 'Authorities' sections. The main panel has tabs for 'Name', 'Type', 'Prerequisites', 'Throttling', 'Server', and 'Cache'. The 'Cache' tab is selected, displaying 'Cache lifetime: 1 minutes' and 'Cache LRU size: 1000'. At the bottom, there are 'Save' and 'Cancel' buttons. A link 'Edit authority 'Native SharePoint authority instance'' is visible in the top right of the main panel.

Fill in the desired caching parameters.

After you click the "Save" button, you will see a connection summary screen, which might look something like this:

Outputs	View Authority Connection Status	
List Output Connections		
Authorities	Name: SharePoint Native	Description: Native SharePoint authority instance
List Authority Groups	Authority type: SharePointNative	Max connections: 10
List User Mapping Connections	Authority group: SharePoint	Authorization domain:
List Authority Connections	Prerequisite user mapping: No prerequisites	
Repositories	Server SharePoint version: SharePoint Services 2.0 (2003)	
List Repository Connections	Server protocol: http	
Jobs	Server name: localhost	
List all Jobs	Server port: 8080	
Status and Job Management	Site path: /alfresco/api	
Status Reports	User name: domain\username	
Document Status	Password: *****	
Queue Status	SSL certificate list: No certificates present	
History Reports	Authorization model: Unknown	
Simple History	Cache lifetime: 1 minutes	
	Cache LRU size: 1000	
	Connection status: The site at http://localhost did not exist	
	Refresh Edit Delete	

Note that in this example, the SharePoint connection is not actually referencing a SharePoint instance, which is leading to an error status message instead of "Connection working".

6 Repository Connection Types

6.1 Alfresco Repository Connection

The Alfresco Repository Connection type allows you to index content from an Alfresco repository.

This connector is compatible with any Alfresco version (2.x, 3.x and 4.x).

This connection type has no support for any kind of document security, except for hand-entered access tokens provided on a per-job basis.

An Alfresco connection has the following configuration parameters on the repository connection editing screen:

Outputs	Name	Type	Throttling	Server
List Output Connections	Edit connection 'An Alfresco repository'			
Authorities	Tenant domain: Username: admin Password: ***** Protocol: http v Server: localhost Port: 8080 Path: /alfresco/api Socket Timeout: 120000			
List Authority Connections	Save Cancel			
Repositories				
List Repository Connections				
Jobs				

Enter the correct username, password and the endpoint to reference the Alfresco document server services.

You also can set a specific Socket Timeout to correctly tune the connector for your own Alfresco repository settings.

If you have a Multi-Tenancy environment configured in the repository, be sure to also set the tenant domain parameter.

The endpoint consists of the HTTP protocol, hostname, port and the context path of the Alfresco Web Services API exposed by the Alfresco server.

By default, if you don't have changed the context path of the Alfresco webapp, you should have an endpoint address similar to the following:

`http://HOSTNAME:PORT/alfresco/api`

After you click the "Save" button, you will see a connection summary screen, which might look something like this:

View Repository Connection Status	
Outputs	
List Output Connections	
Authorities	
List Authority Connections	
Repositories	
List Repository Connections	
Jobs	
List all Jobs	
Status and Job Management	
Status Reports	
Document Status	
Queue Status	
History Reports	
	Name: Alfresco 4.2.c Description: An Alfresco repository Connection type: Alfresco Authority: None (global authority) Max connections: 10 Throttling: Bin regular expression No throttles Description Max avg fetches/min Parameters: tenantDomain= username=admin password=***** protocol=http server=localhost port=8080 path=/alfresco-amp/api socketTimeout=120000 Connection status: Connection working Refresh Edit Delete

When you configure a job to use the Alfresco repository connection an additional tab is presented. This is the "Lucene Query" tab:

Name Connection Scheduling Hop Filters Lucene Query	
Outputs	
List Output Connections	
Authorities	
	Lucene Query: PATH:"/app:company_home/*" AND TYPE:"cm:content" Cancel

The Lucene Query tab allows you to specify the query based on the Lucene Query Language to get all the result documents that need to be ingested.

Please note that if the Lucene Query is null, the connector will ingest all the contents in the Alfresco repository under the Company Home.

Please also note that the Alfresco Connector during the ingestion process, for each result, if it will find a folder node (that must have any child association defined for the node type), it will ingest all the children of the folder node; otherwise it will directly ingest the document (that must have any d:content as one of the properties of the node).

When you are done, and you click the "Save" button, you will see a summary page looking something like this:

Outputs	View a Job			
List Output Connections				
Authorities	Name: Alfresco ingestion against a Null Output connection			
List Authority Connections	Output connection: Null Output	Repository connection:	Alfresco 2.1.0	
Repositories	Priority: 5	Start method:	Don't automatically start	
List Repository Connections	Schedule type: Scan every document once	Minimum recrawl interval:	Not applicable	
	Expiration interval: Not applicable	Reseed interval:	Not applicable	
Jobs	No scheduled run times			
List all Jobs	Maximum hop count for link type 'child': Unlimited			
Status and Job Management	Hop count mode: Delete unreachable documents			
Status Reports				
Document Status				
Queue Status				
	Lucene Query: PATH:"/app.company_home/" AND TYPE:"cm:folder"			
	Edit Delete Copy			

6.2 Alfresco Webscript Repository Connection

The Alfresco Webscript Repository connection type allows you to index content from an Alfresco repository. It also supports document security, in conjunction with the Alfresco Webscript Authority connection Type.

The current connector relies on a set of services that have been developed on top of Alfresco for easing content and metadata retrieving from an Alfresco instance. These services will be available within an Alfresco instance after installing the following artifact: Alfresco Indexer. Please, make sure that you, FIRST, install and deploy within the desired Alfresco instance the AMP generated after building the Alfresco Indexer project before trying to index content from ManifoldCF.

By default each Alfresco Webscript Connection manages a single Alfresco repository, this means that if you have multiple Alfresco repositories that you want to manage, you need to create a specific connection for each one.

The Alfresco Webscript Repository connector has a single configuration tab in the repository connection editing screen called "Server" where one needs to configure the Alfresco's services endpoint:

Name	Type	Throttling	Server
Edit connection 'Alfresco'			
Protocol: http			
Host name: localhost			
Port: 8080			
Context: /alfresco/service			
Store protocol: workspace			
Store ID: SpacesStore			
User name: admin			
Password: *****			
Save Cancel			

As you can see, the Alfresco endpoint settings are quite simple and almost self-explicative:

- Protocol: HTTP or HTTPS depending on your instance
- HostName: IP or domain where the Alfresco instance is hosted in your network
- Port: port where the Alfresco instance has been deployed
- Context: URL path where the webscript services has been deployed (/alfresco/service by default) after installed the WebScript
- Store Protocol: Alfresco's store protocol to be used (workspace/archive)
- StoreID: store's name
- User: user ID used for performing the authored requests to Alfresco. The user **MUST** have enough privileges for all the request, therefore admin user used to be a good idea
- Password: password of the above user in Alfresco

After you click the "Save" button, you will see a connection summary screen, which might look something like this:

View Repository Connection Status				
Name:	Alfresco		Description:	Alfresco
Connection type:	Alfresco Webscript		Max connections:	10
Authority group:	Alfresco			
Throttling:	☐ Bin regular expression No throttles	Description	Max avg fetches/min	
Protocol:	http			
Host name:	localhost			
Port:	8080			
Context:	/alfresco/service			
Store protocol:	workspace			
Store ID:	SpacesStore			
User name:	admin			
Password:	*****			

When you configure a job to use the Alfresco Webscript repository connection an additional tab is presented. This is the "Filtering Configuration" tab:

Name	Connection	Scheduling	1. Filtering Configuration	2. Output Path	Edit a Job
Enable document processing?			☒		
Alfresco site filtering configuration:			No filtering configuration specified Site name Add		
Alfresco mime type filtering configuration:			No filtering configuration specified Mime type name Add		
Alfresco aspect filtering configuration:			No filtering configuration specified Aspect name Add		
Alfresco metadata filtering configuration:			No filtering configuration specified Metadata source Metadata value Add		
Save			Cancel		

The Filtering Configuration tab allows you to fully customize the crawling job on Alfresco by defining which kind of content do you want to index from your Alfresco repository. The configuration tab consist on a list of filters that can be combined for indexing only certain documents or certain parts of the repository:

- Site Filtering: List of sites to be crawled. Only documents belonging to these sites will be indexed
- MimeType Filtering: Allow to filter documents by mimetype in Alfresco
- Aspect Filtering: index only those documents associated to the configured aspects
- Metadata Filtering: index only those documents that at least have the specified value for one of the configured metadata field

6.3 CMIS Repository Connection

The CMIS Repository Connection type allows you to index content from any CMIS-compliant repository.

By default each CMIS Connection manages a single CMIS repository, this means that if you have multiple CMIS repositories exposed by a single endpoint, you need to create a specific connection for each CMIS repository.

CMIS repository documents are typically secured by using the CMIS Authority Connection type. This authority type, however, does not have access to user groups, since there is no such functionality in the CMIS

specification at this time. As a result, most people only use the CMIS connection type in an unsecured manner.

A CMIS connection has the following configuration parameters on the repository connection editing screen:

Select the correct CMIS binding protocol (AtomPub or Web Services) and enter the correct username, password and the endpoint to reference the CMIS document server services.

The endpoint consists of the HTTP protocol, hostname, port and the context path of the CMIS service exposed by the CMIS server:

`http://HOSTNAME:PORT/CMIS_CONTEXT_PATH`

Optionally you can provide the repository ID to select one of the exposed CMIS repository, if this parameter is null the CMIS Connector will consider the first CMIS repository exposed by the CMIS server.

Note that, in a CMIS system, a specific binding protocol has its own context path, this means that the endpoints are different:

for example the endpoint of the AtomPub binding exposed by the actual version of the InMemory Server provided by the OpenCMIS framework is the following:

`http://localhost:8080/chemistry-opencmis-server-inmemory-war-0.5.0-SNAPSHOT/atom`

The Web Services binding is exposed using a different endpoint:

`http://localhost:8080/chemistry-opencmis-server-inmemory-war-0.5.0-SNAPSHOT/services/RepositoryService`

After you click the "Save" button, you will see a connection summary screen, which might look something like this:

View Repository Connection Status	
Outputs	
List Output Connections	
Authorities	
List Authority Connections	
Repositories	
List Repository Connections	
Jobs	
List all Jobs	
Status and Job Management	
Status Reports	
Document Status	
Queue Status	
Document Reports	

Name:	CMIS Ingestion	Description:	Ingestion from a CMIS repository
Connection type:	CMIS	Max connections:	10
Authority:	None (global authority)		
Throttling:	Bin regular expression No throttles	Description	Max avg fetches/min
Parameters:	username=dummyuser password=***** binding=atom protocol=http server=localhost port=8080 path=/chemistry-opencmis-server-inmemory-war/atom		
Connection status:	Connection working		
Refresh Edit Delete			

When you configure a job to use the CMIS repository connection an additional tab is presented. This is the "CMIS Query" tab:

Name		Connection	Scheduling	Hop Filters	CMIS Query	Edit job 'CMIS Ingestion Job'
Outputs						
List Output Connections						
Authorities						
List Authority Connections						
Repositories						
List Repository Connections						
Jobs						
List all Jobs						
Status and Job Management						
Status Reports						
Document Status						
Queue Status						
Document Reports						

CMIS Query:	SELECT * FROM cmis:folder WHERE cmis:name='testdata'	Save	Cancel
-------------	--	------	--------

The CMIS Query tab allows you to specify the query based on the CMIS Query Language to get all the result documents that need to be ingested. Note that the CMIS Connector during the ingestion process, for each result, if it will find a folder node (that must have cmis:folder as the baseType), it will ingest all the children of the folder node; otherwise it will directly ingest the document (that must have cmis:document as the baseType).

When you are done, and you click the "Save" button, you will see a summary page looking something like this:

View a Job	
Outputs	
List Output Connections	
Authorities	
List Authority Connections	
Repositories	
List Repository Connections	
Jobs	
List all Jobs	
Status and Job Management	
Status Reports	
Document Status	
Queue Status	
Document Reports	

Name:	CMIS Ingestion Job
Output connection:	Null Output
Repository connection:	CMIS Ingestion
Priority:	5
Start method:	Don't automatically start
Schedule type:	Scan every document once
Expiration interval:	Not applicable
Minimum recrawl interval:	Not applicable
Reseed interval:	Not applicable
No scheduled run times	
Maximum hop count for link type 'child':	Unlimited
Hop count mode:	Delete unreachable documents
CMIS Query:	SELECT * FROM cmis:folder WHERE cmis:name='testdata'
Edit Delete Copy	

6.4 EMC Documentum Repository Connection

The EMC Documentum connection type allows you index content from a Documentum Content Server instance. A single connection allows you to reach all documents contained on a single Content Server instance. Multiple connections are therefore required to reach documents from multiple Content Server instances.

For each Content Server instance, the Documentum connection type allows you to index any Documentum content that is of type

dm_document, or is derived from dm_document. Compound documents are handled as well, but only by mean of the component documents that make them up. No other Documentum construct can be indexed at this time.

Documents described by Documentum connections are typically secured by a Documentum authority. If you have not yet created a Documentum authority, but would like your documents to be secured, please follow the direction in the section titled "EMC Documentum Authority Connection".

A Documentum connection has the following special tabs: "Docbase", and "Webtop". The "Docbase" tab allows you to select a Content Server to connect to, and also to provide appropriate credentials. The "Webtop" tab describes the location of a Webtop server that will be used to display the documents from that Content Server, after they have been indexed.

The "Docbase" tab looks like this:

Enter the Content Server Docbase instance name, and provide your credentials. You may leave the "Domain" field blank, if the Content Server instance does not have AD integration enabled.

The "Webtop" tab looks like this:

Enter the components of the base URL of the Webtop instance you want to use for serving the documents. Remember that this information will only be used to construct a URL to the document to allow user inspection; it will not be used for any crawling activities.

When you are done, click the "Save" button. When you do, a connection summary and status screen will be presented:

View Repository Connection Status			
Name:	Documentum		
Description:			
Connection type:	Documentum		
Authority:	None (global authority)		
Max connections:	10		
Throttling:	Bin regular expression	Description	Max avg fetches/min
	No throttles		
Parameters:	webtopbaseurl=http://localhost/webtop/ docbasepassword=***** docbaseusername=myusername domain=mydomain.com docbasename=mydocbase		
Connection status:	Connection temporarily failed: Connection refused to host: 127.0.0.1, nested exception is: java.net.ConnectException: Connection refused: connect		
	Refresh Edit Delete		

Pay careful attention to the status, and be prepared to correct any problems that are displayed.

A job created to use a Documentum connection has the following additional tabs associated with it: "Paths", "Document Types", "Content Types", "Security", and "Path Metadata".

The "Paths" tab allows you to construct the paths within Documentum that you want to scan for content. If no paths are selected, all content will be considered eligible.

The "Document Types" tab allows you to select what document types you want to index. Only document types that are derived from `dm_document`, which are flagged by the system administrator as being "indexable", will be presented for your selection. On this tab also, for each document type you index, you may choose included specific metadata for documents of that type, or you can check the "All metadata" checkbox to include all metadata associated with documents of that type.

The "Content Types" tab allows you to select which Documentum mime-types are to be included in the document set. Check the types you want to include, and uncheck the types you want to exclude.

The "Security" tab allows you to disable or enable Documentum security for the documents described by this job. You can turn off native Documentum security by clicking the "Disable" radio button. If you do this, you may also enter your own access tokens, which will be applied to all documents described by the job. The form of the access tokens you enter will depend on the governing authority connection type. Click the "Add" button to add each access token.

The "Path Metadata" tab allows you to send each document's path information as metadata to the index. To enable this feature, enter the name of the metadata attribute you want this information to be sent into the "Path attribute name" field. Then, add the rules you want to the list of rules. Each rule has a match expression, which is a regular expression where parentheses "(" and ")" mark sections you are interested in. These sections are called "groups" in regular expression parlance. The replace string consists of constant text plus substitutions of the groups from the match, perhaps modified. For example, "\$1" refers to the first group within the match, while "\$1l" refers to the first match group mapped to lower case. Similarly, "\$1u" refers to the same characters, but mapped to upper case.

For example, suppose you had a rule which had `"*/(.*)/(.*)/*"` as a match expression, and `"$(1) $(2)"` as the replace string. If presented with the path `Project/Folder_1/Folder_2/Filename`, it would output the string `Folder_1 Folder_2`.

If more than one rule is present, the rules are all executed in sequence. That is, the output of the first rule is modified by the second rule, etc.

6.5 Dropbox Repository Connection

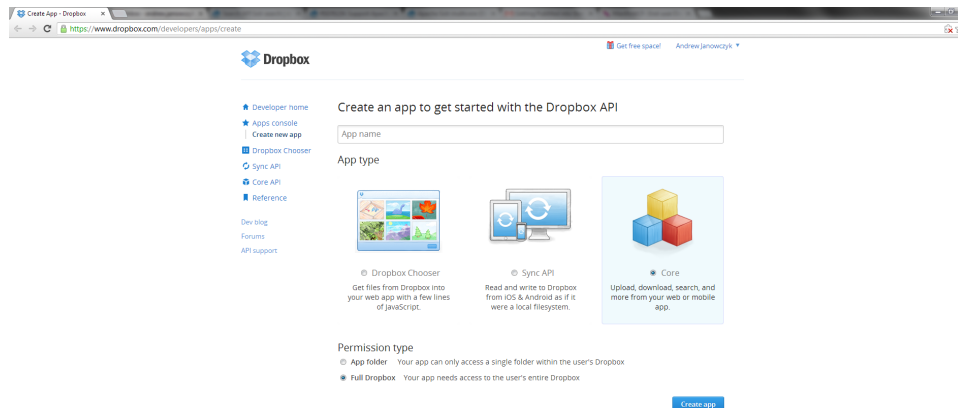
The Dropbox Repository Connection type allows you to index content from Dropbox.

Each Dropbox Connection manages access to a single dropbox repository. This means that if you have multiple dropbox repositories (i.e. different users), you need to create a specific connection for each dropbox repository and provide the associated authentication information.

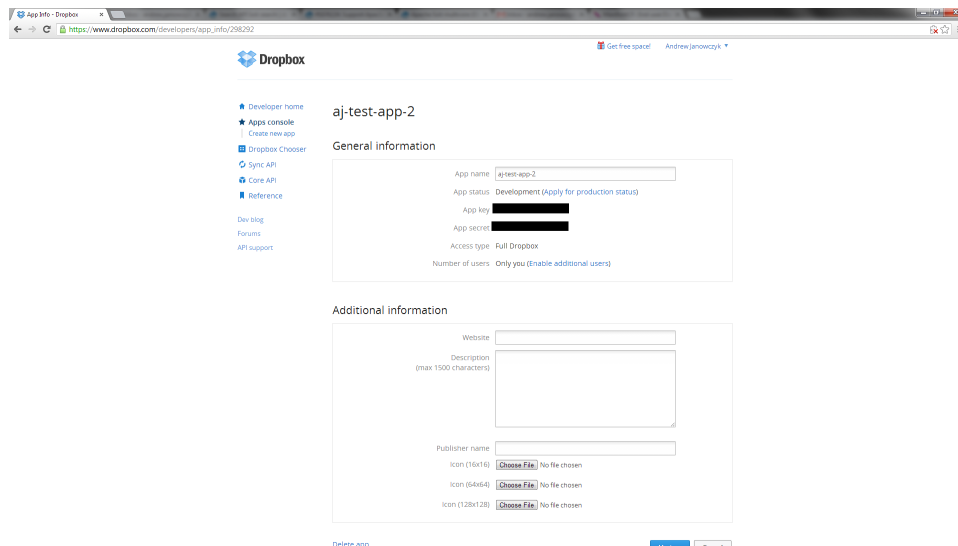
This connection type has no support for any kind of document security, except for hand-entered access tokens provided on a per-job basis.

A Dropbox connection has the following configuration parameters on the repository connection editing screen:

As we can see there are 4 pieces of information which are needed to create a successful connection. The application key and secret are given by dropbox when you register your application for a development license. This is typically done through the application developer Dropbox website.



For our purposes, we need to select "Core" as the application type as we'll be using REST services to communicate with dropbox. Also we select "full access". This merits a small discussion. Typically an application which wants to store and retrieve information does so from an application specific folder. In this case, we assume that the user wants to have access to their files as is, and not copy them into a manifoldcf specific folder. As a result, we have selected full access instead of "App folder".



Afterwards we can see the app key and app secret which are the two pieces of information requested by the connector.

Now each user must confirm their acceptance of allowing your application to access their dropbox. This is done through a run-of-the-

mill OAUTH approach. After providing your application key and secret, the user is directed to a dropbox website asking them if they wish to grant permission to your application. After they accept the request, dropbox provides a client key and secret. These are the last two pieces of information needed for the dropbox connector. This process is covered in depth at the dropbox website which shows examples of how to generate the two needed client tokens.

After you click the "Save" button, you will see a connection summary screen, which might look something like this:

View Repository Connection Status			
Name:	dropbox-connector		
Description:			
Connection type:	DropBox		
Authority:	None (global authority)		
Max connections:	10		
Throttling:	No throttles		
Parameters:	app_key= [redacted] app_secret= [redacted] key= [redacted] secret= [redacted]		
Connection status:	Connection working		
Refresh Edit Delete			

When you configure a job to use the Dropbox repository connection an additional tab is presented. This is the "Dropbox Folder to Index" tab:

Name Connection Scheduling Hop Filters Forced Metadata Solr Field Mapping Dropbox Folder to Index	
Dropbox Folder to Index:	/
Save Cancel	

The Dropbox Folder to Index tab allows you to specify the directory which the dropbox connector will index. Dropbox uses unix style paths, with "/" indicating the root path (and thus the entire dropbox). For example if you want to just index the Photos directory, you would specify "/Photos".

Note that the Dropbox Connector during the ingestion process, for each result, when it find a folder node, it will ingest all the children of the folder node; otherwise it will directly ingest the document.

When you are done, and you click the "Save" button, you will see a summary page looking something like this:

View a Job	
Outputs	Name: dropbox-solr
Authorities	Output connection: solr
Repositories	Repository connection: dropbox
Jobs	Priority: 5
Jobs	Start method: Don't automatically start
Jobs	Schedule type: Scan every document once
Jobs	Expiration interval: Not applicable
Jobs	Minimum recrawl interval: Not applicable
Jobs	Reseed interval: Not applicable
Jobs	No scheduled run times
Jobs	No forced metadata
Jobs	Maximum hop count for link type 'child': Unlimited
Jobs	Hop count mode: Delete unreachable documents
Jobs	Field mappings: Metadata field name: Solr field name
Jobs	No field mapping specified
Jobs	Dropbox Folder to Index: /Photos
Jobs	Index:
Jobs	Edit Delete Copy

6.6 Individual Email Repository Connection

The Individual Email connection type allows you to index content from a single email account, using IMAP, IMAP-SSL, POP3, or POP3-SSL email protocols. Multiple connections are required to support multiple email accounts.

This connection type provides no support at this time for securing email documents.

An Email connection has the following special tabs: "Server" and "URL". The "Server" tab allows you to describe a server and email account, while the "URL" tab allows you describe a URL template that will be used to form the URL for individual emails that are crawled.

The "Server" tab looks like this:

Name		Type	Throttling	Server	URL
Edit connection 'My email'					
Outputs	Protocol:	IMAP			
Authorities	Host name:				
Authorities	Port:				
Authorities	User name:				
Authorities	Password:				
Repositories	Configuration properties:				
Repositories	Add				
Repositories	no server properties specified				
Jobs	Save				
Jobs	Cancel				

Select an email protocol, and type in the name of the email host. Also type in the user name and password. If the port differs from the default for the selected protocol, you may enter a port as well.

The "URL" tab looks like this:

Enter a URL template to be used for generating the URL for each individual email. Use the special substitution tokens "`$(FOLDERNAME)`" and "`$(MESSAGEID)`" to include the individual message's folder name and message ID in the URL, in url-encoded form.

After you click the "Save" button, you will see a connection summary screen, which might look something like this:

Note that in this example, the email connection cannot reach the server, which is leading to an error status message instead of "Connection working".

When you configure a job to use a repository connection of the email type, two additional tabs are presented. These are, in order, "Metadata", and "Filter".

The "Metadata" tab looks something like this:

Select any of the checkboxes to include that metadata in the crawl.

The "Filter" tab looks something like this:

Select one or more folders to include in the pulldown. Then, if you wish to limit the documents included further by applying search criteria, you may select a field, type in a search value, and click the "Add" button. Note that all the fields must match for the email to be included in the crawl.

6.7 IBM FileNet P8 Repository Connection

The IBM FileNet P8 connection type allows you to index content from a FileNet P8 server instance. A connection allows you to reach all files kept on that server. Multiple connections are required to support multiple servers.

This connection type secures documents using the Active Directory authority. If you have not yet created an Active Directory authority, but would like your documents to be secured, please follow the direction in the section titled "Active Directory Authority Connection".

A FileNet connection has the following special tabs: "Server", "Object Store", "Document URL", and "Credentials". The "Server" tab allows you to connect to a specific FileNet P8 Server, while the "Object store" tab allows you to specify the desired FileNet object store. The "Document URL" tab allows you to set up the parameters of each indexed document's URL, while the "Credentials" tab allows you to specify the credentials to use to access the FileNet object store.

The "Server" tab looks like this:

Select the appropriate protocol, and provide the server name, port, and service location.

The "Object Store" tab looks like this:

Type in the name of the FileNet domain you want to connect to, and the name of the FileNet object store within that domain.

The "Document URL" tab looks like this:

This tab allows you to provide the basic URL that will be how each indexed document will be loaded for the user. Select the protocol. Type in the host name. Type in the port. And, type in the location.

The "Credentials" tab looks like this:

Type in the FileNet user ID and password to allow the FileNet connection type access to the FileNet repository.

When you are done filling in the connection information, click the "Save" button. You should see something like this:

More here later

6.8 Generic WGET-Compatible File System Repository Connection

The generic file system repository connection type was developed in part as an example, demonstration, and testing tool, which reads

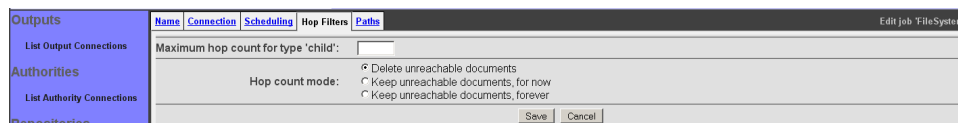
simple files in directory paths, and partly as ManifoldCF support for the Unix utility called *wget*. In the latter mode, the File System Repository Connector will parse file names that were created by *wget*, or by the *wget*-compatible File System Output Connector, and turn these back into full URL's to external web content.

This connection type has no support for any kind of document security, except for hand-entered access tokens provided on a per-job basis.

The File System repository connection type provides no configuration tabs beyond the standard ones. However, please consider setting a "Maximum connections per JVM" value on the "Throttling" tab to at least one per worker thread, or 30, for best performance.

Jobs created using a file-system-type repository connection have two tabs in addition to the standard repertoire: the "Hop Filters" tab, and the "Repository Paths" tab.

The "Hop Filters" tab allows you to restrict the document set by the number of child hops from the path root. While this is not terribly interesting in the case of a file system, the same basic functionality is also used in the Web connection type, where it is a more important feature. The file system connection type gives you a way to see how this feature works, in a more predictable environment:

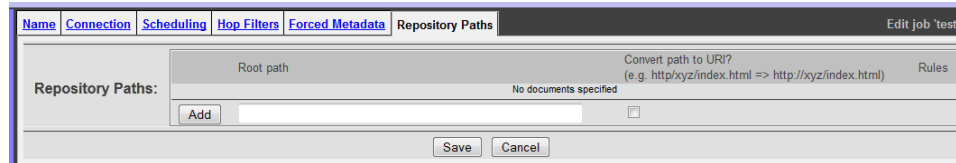


In the case of the File System connection type, there is only one variety of relationship between documents, which is called a "child" relationship. If you want to restrict the document set by how far away a document is from the path root, enter the maximum allowed number of hops in the text box. Leaving the box blank indicates that no such filtering will take place.

On this same tab, you can tell the Framework what to do should there be changes in the distance from the root to a document. The choice "Delete unreachable documents" requires the Framework to recalculate the distance to every potentially affected document whenever a change takes place. This may require expensive bookkeeping, however, so you also have the option of ignoring such changes. There are two varieties of this latter option - you can ignore the changes for now, with the option of turning back on the aggressive bookkeeping at a later time, or

you can decide not to ever allow changes to propagate, in which case the Framework will discard the necessary bookkeeping information permanently.

The "Repository Paths" tab looks like this:



This tab allows you to type in a set of paths which function as the roots of the crawl. For each desired path, type in the path, select whether the root should behave as an WGET repository or not, and click the "Add" button to add it to the list. The form of the path you type in obviously needs to be meaningful for the operating system the Framework is running on.

Each root path has a set of rules which determines whether a document is included or not in the set for the job. Once you have added the root path to the list, you may then add rules to it. Each rule has a match expression, an indication of whether the rule is intended to match files or directories, and an action (include or exclude). Rules are evaluated from top to bottom, and the first rule that matches the file name is the one that is chosen. To add a rule, select the desired pulldowns, type in a match file specification (e.g. "*.txt"), and click the "Add" button.

6.9 Generic Connector

Generic connector allows you to index any source that follows provided API specification. The idea is that you can use it and implement only the API which is designed to be fine grained and as simple as it is possible to handle document indexing.

API should be implemented as xml web page (entry point) returning results based on provided GET params. It may be a simple server script or part of the bigger application. API can be secured with HTTP basic authentication.

There are 4 actions:

- check
- seed
- items

- item

Action is passed as "action" GET param to the endpoint.

[endpoint]?action=check

Should return HTTP status code 200 providing information that endpoint is working properly. Any content returned will be ignored, only the status code matters.

[endpoint]?action=seed&startDate=YYYY-MM-DDTHH:mm:ssZ&endDate=YYYY-MM-DDTHH:mm:ssZ

Parameters:

- startDate - the start of time frame which should be applied to returned seeds. If this is a first run - this parameter will not be provided meaning that all documents should be returned.
- endDate - the end of time frame. Always provided.

startDate and endDate parameters are encoded as YYYY-MM-DD'T'HH:mm:ss'Z'. Result should be valid XML of form:

```
<seeds>
  <seed id="document_id_1" />
  <seed id="document_id_2" />
  ...
</seeds>
```

Attributes id are required.

[endpoint]?action=items&id[]=document_id_1&id=document_id_2

Parameters:

- id[] - array of document IDs that should be returned

Result should be valid XML of form:

```
<items>
  <item id="document_id_1">
    <url>[http://document_uri]</url>
    <version>[document_version]</version>
    <created>2013-11-11T21:00:00Z</created>
    <updated>2013-11-11T21:00:00Z</updated>
    <filename>filename.ext</filename>
    <mimetype>mime/type</mimetype>
    <metadata>
      <meta name="meta_name_1">meta_value_1</meta>
      <meta name="meta_name_2">meta_value_2</meta>
    ...
  </item>
  ...
</items>
```

```

</metadata>
<auth>
<token>auth_token_1</token>
<token>auth_token_2</token>
...
</auth>
<related>
<id>other_document_id_1</id>
<id>other_document_id_2</id>
...
</related>
<content>Document content</content>
</item>
...
</items>

```

id, url, version are required, the rest is optional.

If auth tag is provided - document will be treated as non-public with defined access tokens, if it is omitted - document will be public.

If content tag is omitted - connector will ask for document content as action=item separate API call.

You may provide related document ids when document repository is a graph or tree. Provided documents will also be indexed. In case you want to use relations - seeding do not have to return all documents, only starting points. Rest of documents will be fetched using relations.

[entrypoint]?action=item&id=document_id

Parameters:

- id - requested document ID

Result should be the document content. It does not have to be XML - you may return binary data (PDF, DOC, etc) which represent the document.

You may provide custom parameters by defining them in Job specification. All defined parameters will be sent as additional GET parameters with every API call

You may override provided auth tokens and define forced tokens in Job specification. If you change security model to "forced" and do not provide any tokens - all documents will be public.

6.10 Generic Database Repository Connection

The generic database connection type allows you to index content from a database table, served by one of the following databases:

- Postgresql (via a Postgresql JDBC driver)
- SQL Server (via the JTDS JDBC driver)
- Oracle (via the Oracle JDBC driver)
- Sybase (via the JTDS JDBC driver)
- MySQL (via the MySQL JDBC driver)

This connection type cannot be configured to work with other databases than the ones listed above without software changes. Depending on your particular installation, some of the above options may not be available.

The generic database connection type currently has no per-document notion of security. It is possible to set document security for all documents specified by a given job. Since this form of security requires you to know what the actual access tokens are, you must have detailed knowledge of the authority connection you intend to use, and what sorts of access tokens it produces.

A generic database connection has three special tabs on the repository connection editing screen: the "Database Type" tab, the "Server" tab, and the "Credentials" tab. The "Database Type" tab looks like this:

Select the kind of database you want to connect to, from the pulldown.

Also, select the JDBC access method you want from the access method pulldown. The access method is provided because the JDBC specification has been recently clarified, and not all JDBC drivers work the same way as far as resultset column name discovery is concerned. The "by name" option currently works with all JDBC drivers in the list except for the MySQL driver. The "by label" works for the current MySQL driver, and may work for some of the others as well. If the queries you supply for your generic database jobs do not work correctly, and you see an error message about not being able to find required columns in the result, you can change your selection on this pulldown and it may correct the problem.

The "Server" tab looks like this:

Here you have a choice. Either you can choose to specify the database host and port, and the database name or instance name, or you can provide a raw JDBC connection string that is appropriate for the database type you have chosen. This latter option is provided because many JDBC drivers, such as Oracle's, now can connect to an entire cluster of Oracle servers if you specify the appropriate connection description string.

If you choose the second option, just consult your JDBC driver's documentation and supply your string. If there is anything entered in the raw connection string field at all, it will take precedence over the database host and database name fields.

If you choose the first option, the server name and port must be provided in the "Database host and port" field. For example, for Oracle, the standard Oracle installation uses port 1521, so you would enter something like, "my-oracle-server:1521" for this field. Postgresql uses port 5432 by default, so "my-postgresql-server:5432" would be required. SQL Server's standard port is 1433, so use "my-sql-server:1433".

The service name or instance name field describes which instance and database to connect to. For Oracle or Postgresql, provide just the database name. For SQL Server, use "my-instance-name/my-database-name". For SQL Server using the default instance, use just the database name.

The "Credentials" tab is straightforward:

Enter the database user credentials.

After you click the "Save" button, you will see a connection summary screen, which might look something like this:

View Repository Connection Status			
Name:	JDBC	Description:	JDBC connection
Connection type:	JDBC	Max connections:	10
Authority group:	None (global authority)		
Throttling:	Bin regular expression	Description	Max avg fetches/min
Parameters:	Database name=database Host=localhost:5432 Raw driver string= Password=***** JDBC column access method=name JDBC Provider=postgresql User name=somebody		
Connection status:	Threw exception: 'Error getting connection: FATAL: password authentication failed for user "somebody"'		
Refresh Edit Delete Clear All Related History			

Note that in this example, the generic database connection is not properly authenticated, which is leading to an error status message instead of "Connection working".

When you configure a job to use a repository connection of the generic database type, several additional tabs are presented. These are, in order, "Queries", and "Security".

The "Queries" tab looks something like this:

The screenshot shows the 'Queries' tab in the ManifoldCF configuration interface. The sidebar on the left lists various configuration categories. The main area contains four query input fields, each with a label and a SQL query. The queries are as follows:

- Seeding query:** (return ids that need to be checked)
`SELECT idfield AS $(IDCOLUMN) FROM documenttable WHERE modifydatefield > $(STARTTIME) AND modifydatefield <= $(ENDTIME)`
- Version check query:** (return ids and versions for a set of documents; leave blank if no versioning capability)
`SELECT idfield AS $(IDCOLUMN), versionfield AS $(VERSIONCOLUMN) FROM documenttable WHERE idfield IN $(IDLIST)`
- Access token query:** (return ids and access tokens for a set of documents; leave blank if no security capability)
`SELECT docidfield AS $(IDCOLUMN), acfield AS $(TOKENCOLUMN) FROM acstable WHERE docidfield IN $(IDLIST)`
- Data query:** (return ids, urls, and data for a set of documents)
`SELECT idfield AS $(IDCOLUMN), urlfield AS $(URLCOLUMN), datafield AS $(DATACOLUMN) FROM documenttable WHERE idfield IN $(IDLIST)`

At the bottom of the main area are 'Save' and 'Cancel' buttons.

You must supply at least two queries. (All other queries are optional.) The purpose of these queries is to obtain the data needed for the database to be properly crawled. But in order for you to write these queries, you must make some decisions first. Basically, you need to figure out how best to map the constructs within your database to the requirements of the Framework. The following are the sorts of queries you can provide:

- Obtain a list of document identifiers corresponding to changes and additions that occurred within a specified time window (see below)
- Given a set of document identifiers, find the corresponding version strings (see below)
- Given a set of document identifiers, find the corresponding list of access tokens for each document identifier (see below)
- Given a set of document identifiers, find information about the document, consisting of the document's data, access URL, and metadata
- Given a set of document identifiers, find multivalued metadata from the document (multiple such queries)

The Framework uses a unique document identifier to describe every document within the confines of a defined repository connection. This

document identifier is used as a primary key to locate information about the document. When you set up a generic-database-type job, the database you are connecting to must have a similar concept. If you pick the wrong thing for a document identifier, at the very least you could find that the crawler runs very slowly.

Obtaining the list of document identifiers that represents the changes that occurred over the given time frame must return at least all such changes. It is acceptable (although not ideal) for the returned list to be bigger than that.

If you want your database connection to function in an incremental manner, you must also come up with the format of a "version string". This string is used by the Framework to determine if a document has changed. It must change whenever anything that might affect the document's indexing changes. (It is not a problem if it changes for other reasons, as long as it fulfills that principle criteria.)

The queries you provide get substituted before they are used by the connection. The example queries, which are present when the queries tab is first opened for a new job, show many of these substitutions in roughly the manner in which they are intended to be used. For example, "\$ (IDCOLUMN)" will substitute a column name expected by the connection to contain the document identifier into the query. The list of substitution strings are as follows:

String name	Meaning/use
IDCOLUMN	The name of an expected resultset column containing a document identifier
VERSIONCOLUMN	The name of an expected resultset column containing a version string
TOKENCOLUMN	The name of an expected resultset column containing an access token
URLCOLUMN	The name of an expected resultset column containing a URL
DATACOLUMN	The name of an expected resultset column containing document data
STARTTIME	A query string value containing a start time in milliseconds since epoch

ENDTIME	A query string value containing an end time in milliseconds since epoch
IDLIST	A query string value containing a parenthesized list of document identifier values

It is often necessary to construct composite values using SQL query operators in order to create the version strings, URLs, and data which the JDBC connection type needs. Each database has its own caveats in this regard. Consult your database manual to be sure you are constructing your queries in a manner consistent with best practices for that database. For example, for MySQL databases, NULL column values will prevent straightforward concatenation of results. You might expect the following to work:

```
SELECT id AS $(IDCOLUMN), CONCAT("http://my.base.url/
show.html?record=", id) AS $(URLCOLUMN), CONCAT(name, " ",
description, " ", what_ever) AS $(DATACOLUMN) FROM accounts
WHERE id IN $(IDLIST)
```

But, any NULL column values will disrupt the concatenation in MySQL, so you must instead write your query like this:

```
SELECT id AS $(IDCOLUMN), CONCAT("http://my.base.url/
show.html?record=", id) AS $(URLCOLUMN), CONCAT(name, "
", IFNULL(description, ""), " ", IFNULL(what_ever, "")) AS
$(DATACOLUMN) FROM accounts WHERE id IN $(IDLIST)
```

Also, use caution when constructing queries that include time-based components. "\$(STARTTIME)" and "\$(ENDTIME)" provide times in milliseconds since epoch. If the modified date field is not in this unit, the seeding query may not select the desired document identifiers. You should convert "\$(STARTTIME)" and "\$(ENDTIME)" to the appropriate timestamp unit for your system within your query. The following table gives several sample query fragments that can be used to convert the helper strings "\$(STARTTIME)" and "\$(ENDTIME)" into other date and time types. The first column names the SQL database type that the following query phrase corresponds to, the second column names the output data type for the query phrase, and the third gives the query phrase itself using "\$(STARTTIME)" as an example time in milliseconds since epoch. These query phrases are intended as guidelines for creating an appropriate query phrase in each language. Each query phrase is

designed to work with the most current version of the database software available at the time of publishing for this document. If your modified date field is not of the type given in the second column, the query phrase may not provide an appropriate output for date comparisons.

Database Type	Date Type	Sample Query Phrase
Oracle	date	TO_DATE ('1970/01/01:00:00:00', 'yyyy/mm/ dd:hh:mi:ss') + ROUND (\$ (STARTTIME)/86400000)
Oracle	timestamp	TO_TIMESTAMP('1970-01-01 00:00:00') + interval '\$(STARTTIME)/1000' second
Postgres SQL	timestamp	date '1970-01-01' + interval '\$(STARTTIME) milliseconds'
MS SQL Server (\$>\$6.5)	datetime	DATEADD(ms, \$(STARTTIME), '19700101')
Sybase (10+)	datetime	DATEADD(ms, \$(STARTTIME), '19700101')

When you create a job based on a general database connection, the job's queries are initially populated with examples. These examples should give you a good idea of what columns your queries should return - in most cases, the only columns you need to return are the ones that appear in the example queries. However, for the file data query, you may also return columns that are not specified in the example. When you do this, the extra return column values will be passed to the index as metadata for the document. The metadata name used will be the corresponding resultlist column name of the resultset.

For example, the following file data query (written for PostgreSQL) will return documents with the metadata fields "metadata_a" and "metadata_b", in addition to the required primary document body and URL:

```
SELECT id AS $(IDCOLUMN), characterdata AS $(DATACOLUMN),  
'http://mydynamicserver.com?id=' || id AS $(URLCOLUMN),
```

```
publisher AS metadata_a, distributor AS metadata_b FROM
mytable WHERE id IN ${IDLIST}
```

The "Security" tab simply allows you to add specific access tokens to all documents indexed with a general database job. In order for you to know what tokens to add, you must decide with what authority connection these documents will be secured, and understand the form of the access tokens used by that authority connection type. This is what the "Security" tab looks like:

Here, you can turn off security, if you want no access tokens to be transmitted with each document. Or, you can leave security "Enabled", and either create a list of access tokens by hand, using the widget on this tab, or leave these blank and provide an access token query on the "Queries" tab. To add access tokens by hand, enter a desired access token, and click the "Add" button. You may enter multiple access tokens.

6.11 Google Drive Repository Connection

The Google Drive Repository Connection type allows you to index content from Google Drives.

Each Google Drive Connection manages access to a single drive repository. This means that if you have multiple Google Drives (i.e. different users), you need to create a specific connection for each drive repository and provide the associated authentication information.

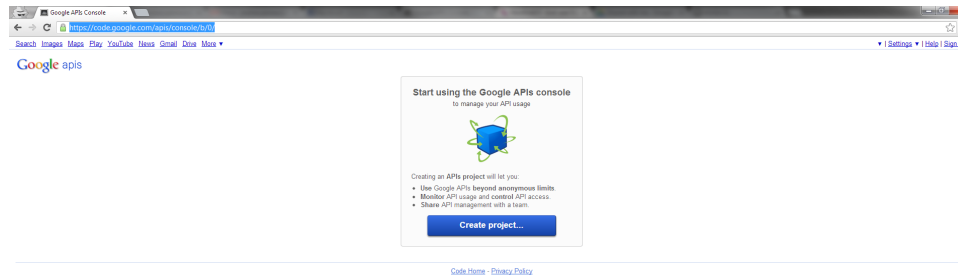
This connection type has no support for any kind of document security, except for hand-entered access tokens provided on a per-job basis.

This connection type can index only those documents whose binary content can be obtained through the pertinent Google APIs. At the moment, native Google documents such as Google Spreadsheet do not appear to be supported.

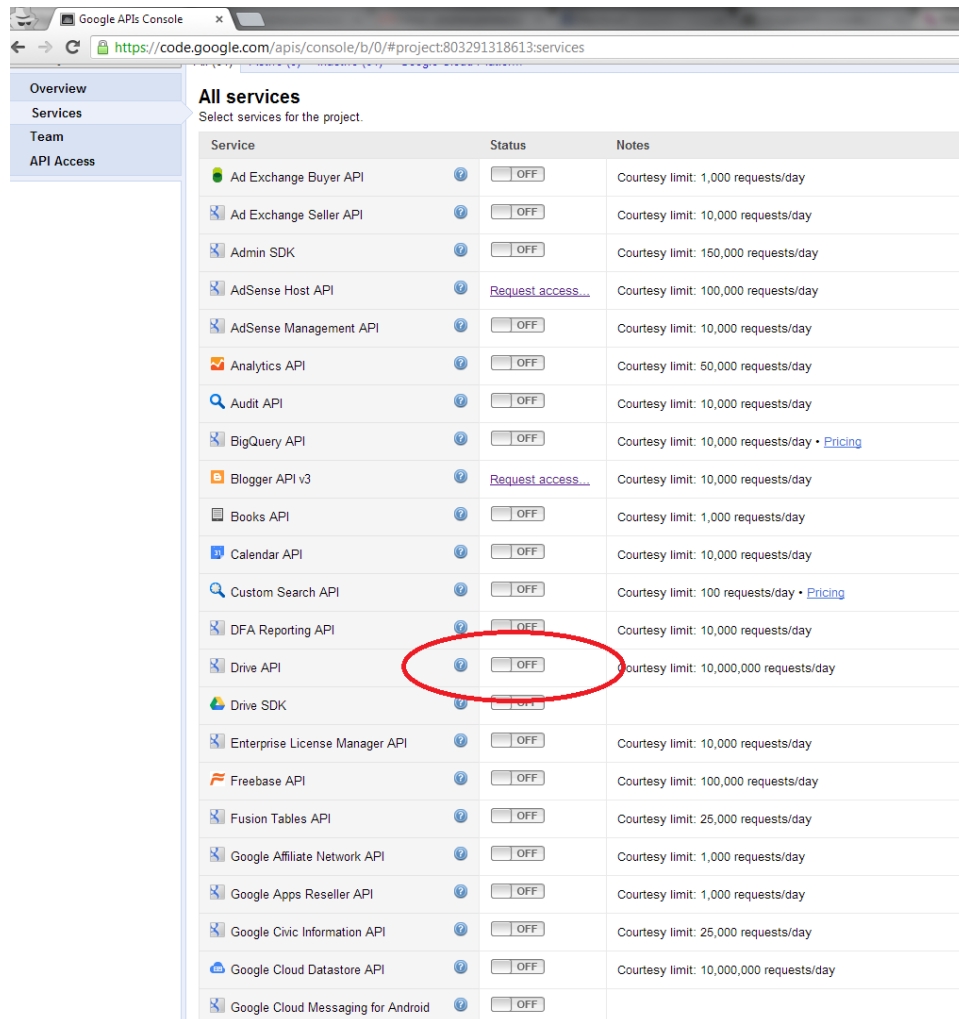
A Google Drive connection has the following configuration parameters on the repository connection editing screen:

Outputs	Name	Type	Throttling	Server
List Output Connections Authorities List Authority Connections				RefreshToken: Client ID: Client Secret ID: Save Cancel

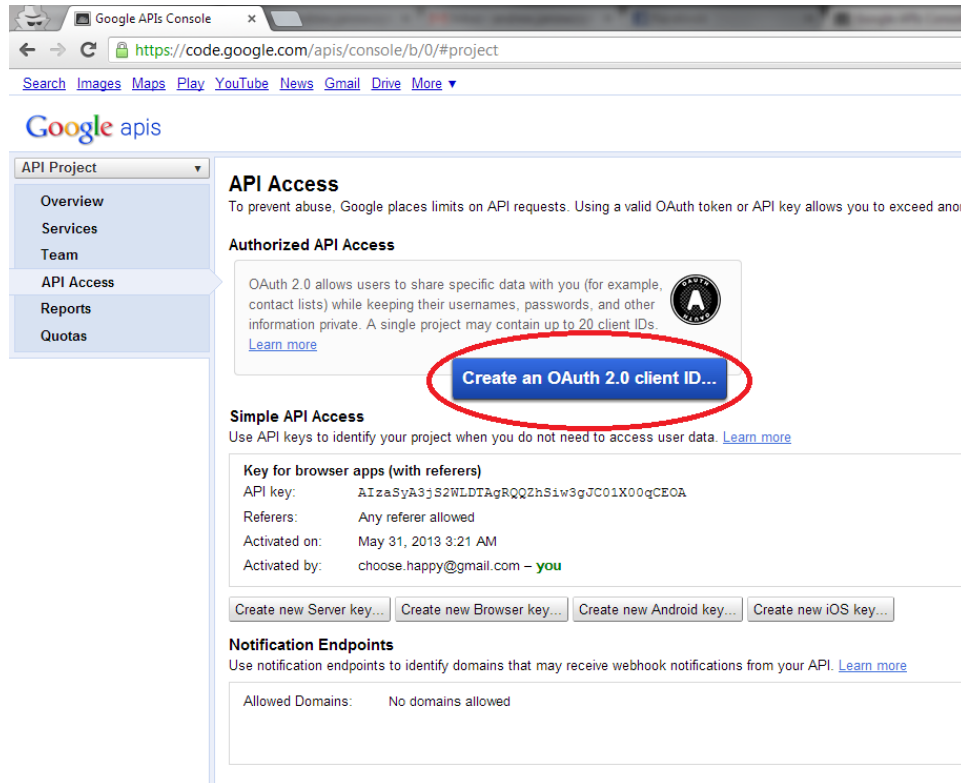
As we can see there are 3 pieces of information which are needed to create a successful connection. The Client ID and Client Secret given by Google Drive when you register your application for a development license. This is typically done through the Google APIs Console.



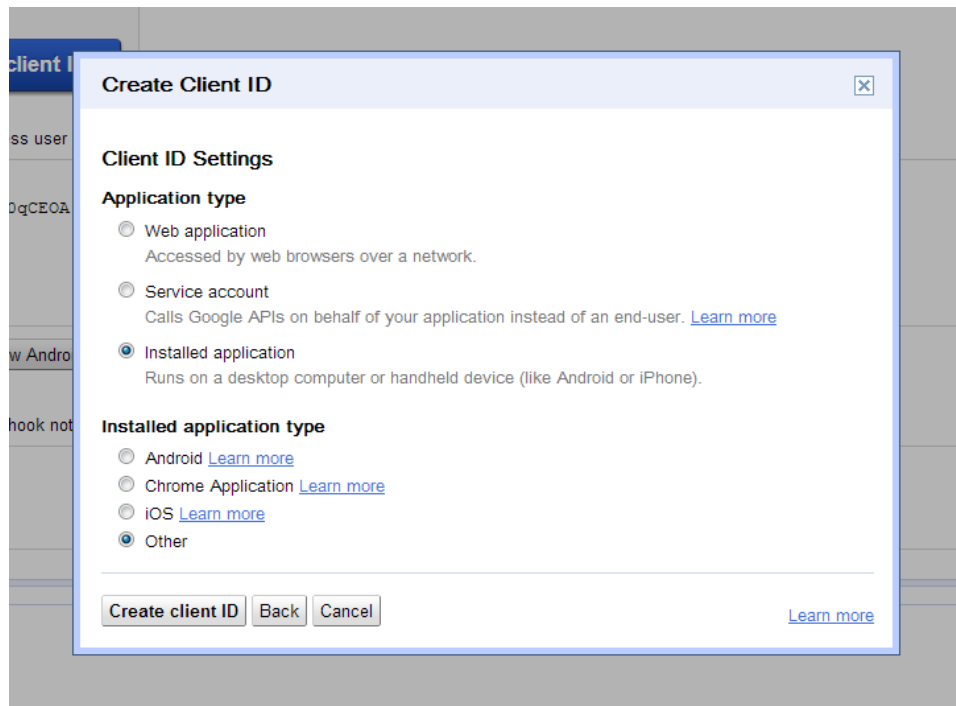
Once having created a project, we must enable the Google Drive API:



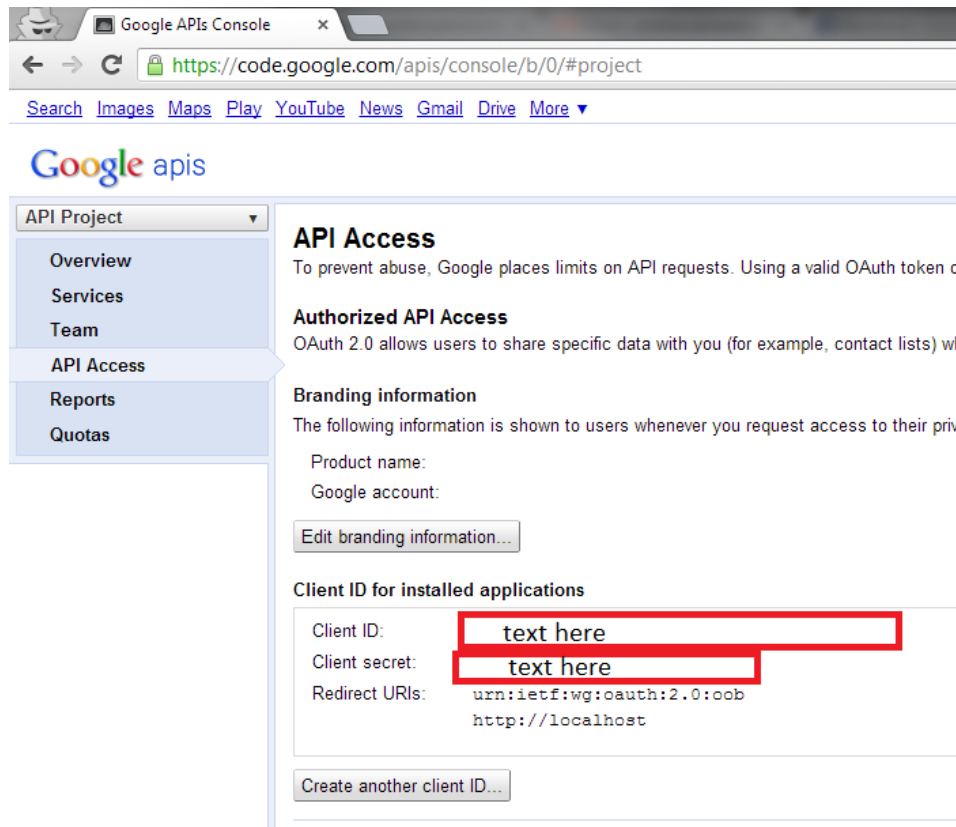
Then going to the API Access link on the right side, we need to select create an OAuth 2.0 client ID:



After filling in the necessary information, we need to select what type of application we'd like. For our purposes we need to select installed application:



Afterwards the connector requests our Client ID and Client secrets (where the red boxes are):



Now each user must confirm their acceptance of allowing your application to access their google drive. This is done through a run-of-the-mill OAUTH approach, but needs to be done beforehand. Once the steps are completed, a long-life refresh token is presented, which is then used by the connector. For completeness, we present the needed steps below since they require some manual work.

1. Browse to here: `https://accounts.google.com/o/oauth2/auth?scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive.readonly&state=%2Fprofile&redirect_uri=https%3A%2F%2Flocalhost&response_type=code&client_id=<CLIENT_ID>&approval_prompt=`
2. This returns a link (after acceptance), where a code is embedded in the URL: `https://localhost/?state=/profile&code=<CODE>`
3. Use a tool like *curl* (<http://curl.haxx.se>) to perform a POST to "`https://accounts.google.com/o/oauth2/token`", using the body: `grant_type=authorization_code&redirect_uri=https%3A%2F%2Flocalhost&client_secret=<CLIENT_SECRET>&client_id=<CLIENT_ID>&code=`

- The response is then a json response which contains the `refresh_token`.

After you click the "Save" button, you will see a connection summary screen, which might look something like this:

The screenshot shows a web interface with a sidebar on the left containing links: Outputs, List Output Connections, Authorities, List Authority Connections, Repositories, List Repository Connections, and Jobs. The main content area is titled 'View Repository Connection Status' and displays details for a 'Google Drive' connection. The details include: Name: Google Drive, Description: (empty), Connection type: googledrive, Authority: None (global authority), Max connections: 10, Throttling: No throttles, and Client ID, Client Secret, and RefreshToken fields. The connection status is 'Connection working'. There are 'Refresh', 'Edit', and 'Delete' links at the bottom right.

When you configure a job to use the Google Drive repository connection an additional tab is presented. This is the "Google Drive Seed Query" tab:

The screenshot shows a web interface with a sidebar on the left containing links: Outputs, List Output Connections, and Authorities. The main content area has tabs: Name, Connection, Scheduling, Hop Filters, Forced Metadata, Google Drive Seed Query, and Edit job 'gdrsema'. The 'Google Drive Seed Query' tab is active, showing a text input field with the value 'mimeTypes=application/vnd.google-apps.folder' and a 'Save' button.

This tab allows you to specify the query which will be used to seed documents for the indexing process. The query language is specified on the Drive Search Parameters site. Directories which meet the seed query are fully crawled as the query on applies to seeds. The default query indexes the entire drive. Lastly, native Google documents such as spreadsheets and word documents are exported to PDF and then ingested.

6.12 HDFS Repository Connection (WGET compatible)

The HDFS repository connection operates much like the File System Repository Connection, except it reads data from the Hadoop File System rather than a local disk. It, too, is capable of understanding directories written in the manner of the Unix utility called *wget*. In the latter mode, the HDFS Repository Connector will parse file names that were created by *wget*, or by the wget-compatible HDFS Output Connector, and turn these back into full URL's pointing to external web content.

This connection type has no support for any kind of document security, except for hand-entered access tokens provided on a per-job basis.

The HDFS repository connection type has an additional configuration tab above and beyond the standard ones, called "Server". This is what it looks like:

Enter the HDFS name node URI, and the user name, and click the "Save" button.

Jobs created using an HDFS repository connection type have two tabs in addition to the standard repertoire: the "Hop Filters" tab, and the "Repository Paths" tab.

The "Hop Filters" tab allows you to restrict the document set by the number of child hops from the path root. This is what it looks like:

In the case of the HDFS connection type, there is only one variety of relationship between documents, which is called a "child" relationship. If you want to restrict the document set by how far away a document is from the path root, enter the maximum allowed number of hops in the text box. Leaving the box blank indicates that no such filtering will take place.

On this same tab, you can tell the Framework what to do should there be changes in the distance from the root to a document. The choice "Delete unreachable documents" requires the Framework to recalculate the distance to every potentially affected document whenever a change takes place. This may require expensive bookkeeping, however, so you also have the option of ignoring such changes. There are two varieties of this latter option - you can ignore the changes for now, with the option of turning back on the aggressive bookkeeping at a later time, or you can decide not to ever allow changes to propagate, in which case the Framework will discard the necessary bookkeeping information permanently.

The "Repository Paths" tab looks like this:

This tab allows you to type in a set of paths which function as the roots of the crawl. For each desired path, type in the path, select whether the root should behave as an WGET repository or not, and click the "Add" button to add it to the list.

Each root path has a set of rules which determines whether a document is included or not in the set for the job. Once you have added the root path to the list, you may then add rules to it. Each rule has a match expression, an indication of whether the rule is intended to match files or directories, and an action (include or exclude). Rules are evaluated from top to bottom, and the first rule that matches the file name is the one that is chosen. To add a rule, select the desired pulldowns, type in a match file specification (e.g. "*.txt"), and click the "Add" button.

6.13 Jira Repository Connection

The Jira Repository Connection type allows you to index tickets from Atlassian's Jira.

This repository connection type is meant to secure documents in conjunction with the Jira Authority Connection type. Please read the associated documentation to configure document security.

A Jira connection has the following configuration parameters on the repository connection editing screen:

Name	Type	Throttling	Server
JIRA protocol: http			
JIRA host:			
JIRA port:			
JIRA REST API path: /rest/api/2/			
Client ID (Optional):			
Client Secret (Optional):			

Save Cancel

As we can see there are 3 pieces of information which are needed to create a successful connection. The Client ID and Client Secret are the username and password of a Jira account. The JiraUrl is the base endpoint of the particular Jira Instance, for example: <https://searchbox.atlassian.net> is the Jira version for searchbox which is hosted at Atlassian.

After you click the "Save" button, you will see a connection summary screen, which might look something like this:

View Repository Connection Status	
Name:	jira
Description:	
Connection type:	Jira
Authority:	None (global authority)
Max connections:	10
Throttling:	☐ Bin regular expression ☒ No throttles
Description	Max avg fetches/min
JIRA protocol:	http
JIRA host:	localhost
JIRA port:	1234
JIRA REST API path:	/rest/api/2/
Client ID (Optional):	
Client Secret (Optional):	*****
Connection status:	Connection temporarily failed: IO exception: Connection to http://localhost:1234 refused
Refresh Edit Delete	

When you configure a job to use the Jira repository connection an additional tab is presented. This is the "Seed Query" tab:

Seed Query	
Jira Query:	ORDER BY createDate Asc
Save Cancel	

This tab allows you to specify the query which will be used to find documents for the indexing process. The query language is specified on the Jira Advanced Searching site. Directories which meet the seed query are fully crawled as the query only applies to seeds.

6.14 OpenText LiveLink Repository Connection

The LiveLink connection type allows you to index content from LiveLink repositories. LiveLink has a rich variety of different document types and metadata, which include basic documents, as well as compound documents, folders, workspaces, and projects. A LiveLink connection is able to discover documents contained within all of these constructs.

Documents described by LiveLink connections are typically secured by a LiveLink authority. If you have not yet created a LiveLink authority, but would like your documents to be secured, please follow the direction in the section titled "OpenText LiveLink Authority Connection".

A LiveLink connection has the following special tabs: "Server", "Document Access", and "Document View". The "Server" tab allows you to select a LiveLink server to connect to, and also to provide appropriate credentials. The "Document Access" tab describes the location of the LiveLink web interface, relative to the server, that will be used to fetch document content from LiveLink. The "Document View" tab affects how URLs to the fetched documents are constructed, for viewing results of searches.

The "Server" tab looks like this:

Select the manner you want the connection to use to communicate with LiveLink. Your options are:

- Internal (native LiveLink protocol)
- HTTP (communication with LiveLink through the IIS web server)
- HTTPS (communication with LiveLink through IIS using SSL)

Also, you need to enter the name of the desired LiveLink server, the LiveLink port, and the LiveLink server credentials. If you have selected communication using HTTP or HTTPS, you must provide a relative CGI path to your LiveLink. You may also need to provide web server credentials. Basic authentication and older forms of NTLM are supported. In order to use NTLM, specify a non-blank server domain name in the "Server HTTP domain" field, plus a non-qualified user name and password. If basic authentication is desired, leave the "Server HTTP domain" field blank, and provide basic auth credentials in the "Server HTTP NTLM user name" and "Server HTTP NTLM password" fields. For no web server authentication, leave these fields all blank.

For communication using HTTPS, you will also need to upload your authority certificate(s) on the "Server" tab, to tell the connection which certificates to trust. Upload your certificate using the browse button, and then click the "Add" button to add it to the trust store.

The "Document Access" tab looks like this:

The server name is presumed to be the same as is on the "Server" tab. Select the desired protocol for document retrieval. If your LiveLink server is using a non-standard HTTP port for the specified protocol for document retrieval, enter the port number. If your LiveLink server is using NTLM authentication to access documents, enter an Active Directory user name, password, and domain. If your LiveLink is using HTTPS, browse locally for the appropriate certificate authority certificate, and click "Add" to upload that certificate to the connection's trust store. (You may also use the server's certificate, but that is less resilient because the server's certificate may be changed periodically.)

The "Document View" tab looks like this:

If you want each document's view URL to be the same as its access URL, you can leave this tab unchanged. If you want to direct users to a different CGI path when they view search results, you can specify that here.

When you are done, click the "Save" button. You will see a summary screen that looks something like this:

Make note of and correct any reported connection errors. In this example, the connection has been correctly set up, so the connection status is "Connection working".

A job created to use a LiveLink connection has the following additional tabs associated with it: "Paths", "Filters", "Security", and "Metadata".

The "Paths" tab allows you to manage a list of LiveLink paths that act as starting points for indexing content:

Build each path by selecting from the available dropdown, and clicking the "+" button. When your path is complete, click the "Add" button to add the path to the list of starting points.

The "Filters" tab controls the criteria the LiveLink job will use to include or exclude content. The filters are basically a list of rules. Each rule has a document match field, and a matching action ("Include" or "Exclude"). When a LiveLink connection encounters a document, it evaluates the rules from top to bottom. If the rule matches, then it will be included or excluded from the job's document set depending on what you have selected for the matching action. A rule's match field specifies a character match, where "*" will match any number of characters, and "?" will match any single character.

Enter the match field value, select the match action, and click the "Add" button to add to the list of filters.

The "Security" tab allows you to disable (or enable) LiveLink security for the documents associated with this job:

If you disable security, you can add your own access tokens to all jobs in the document set as they are indexed. The format of the access tokens you would enter depends on the governing authority associated with the job's repository connection. Enter a token and click the "Add" button to add it to the list.

The "Metadata" tab allows you to select what specific metadata values from LiveLink you want to pass to the index:

If you want to pass all available LiveLink metadata to the index, then click the "All metadata" radio button. Otherwise, you need to build LiveLink metadata paths and add them to the metadata list. Select the next metadata path segment, and click the appropriate "+" button to add it to the path. You may add folder information, or a metadata category, at any point.

Once you have drilled down to a metadata category, you can select the metadata attributes to include, or check the "All attributes in this category" checkbox. When you are done, click the "Add" button to add the metadata attributes that you want to include in the index.

You can also use the "Metadata" tab to have the connection send path data along with each document, as a piece of document metadata. To enable this feature, enter the name of the metadata attribute you want this information to be sent into the "Path attribute name" field. Then, add the rules you want to the list of rules. Each rule has a match expression, which is a regular expression where parentheses "(" and ")" mark sections you are interested in. These sections are called "groups" in regular expression parlance. The replace string consists of constant text plus substitutions of the groups from the match, perhaps modified. For example, "\$ (1)" refers to the first group within the match, while "\$ (1l)" refers to the first match group mapped to lower case. Similarly, "\$ (1u)" refers to the same characters, but mapped to upper case.

For example, suppose you had a rule which had ".*/(.*)/(.*)/*" as a match expression, and "\$ (1) \$ (2)" as the replace string. If presented with the path Project/Folder_1/Folder_2/Filename, it would output the string Folder_1 Folder_2.

If more than one rule is present, the rules are all executed in sequence. That is, the output of the first rule is modified by the second rule, etc.

6.15 Autonomy Meridio Repository Connection

An Autonomy Meridio connection allows you to index documents from a set of Meridio servers. Meridio's architecture allows you to separate

services on multiple machines - e.g. the document services can run on one machine, and the records services can run on another. A Meridio connection type correspondingly is configured to describe each Meridio service independently.

Documents described by Meridio connections are typically secured by a Meridio authority. If you have not yet created a Meridio authority, but would like your documents to be secured, please follow the direction in the section titled "Autonomy Meridio Authority Connection".

A Meridio connection has the following special tabs on the repository connection editing screen: the "Document Server" tab, the "Records Server" tab, the "Web Client" tab, and the "Credentials" tab. The "Document Server" tab looks like this:

The screenshot shows the 'Document Server' tab of a configuration window. On the left is a sidebar with 'Outputs' selected. The main area has tabs for 'Name', 'Type', 'Throttling', 'Document Server' (active), 'Records Server', 'Credentials', and 'Web Client'. The 'Document Server' tab contains the following fields:

- Document webservice server protocol: http (dropdown)
- Document webservice server name: (text box)
- Document webservice server port: (text box)
- Document webservice location: /DMWS/MeridioDMWS.asmx (text box)
- Document webservice server proxy host: (text box)
- Document webservice server proxy port: (text box)

 At the bottom right are 'Save' and 'Cancel' buttons.

Select the correct protocol, and enter the correct server name, port, and location to reference the Meridio document server services. If a proxy is involved, enter the proxy host and port. Authenticated proxies are not supported by this connection type at this time.

Note that, in the Meridio system, while it is possible that different services run on different servers, this is not typically the case. The connection type, on the other hand, makes no assumptions, and permits the most general configuration.

The "Records Server" tab looks like this:

The screenshot shows the 'Records Server' tab of the same configuration window. The fields are:

- Record webservice server protocol: http (dropdown)
- Record webservice server name: (text box)
- Record webservice server port: (text box)
- Record webservice location: /RMWS/MeridioRMWS.asmx (text box)
- Record webservice server proxy host: (text box)
- Record webservice server proxy port: (text box)

 At the bottom right are 'Save' and 'Cancel' buttons.

Select the correct protocol, and enter the correct server name, port, and location to reference the Meridio records server services. If a proxy is involved, enter the proxy host and port. Authenticated proxies are not supported by this connection type at this time.

Note that, in the Meridio system, while it is possible that different services run on different servers, this is not typically the case. The

connection type, on the other hand, makes no assumptions, and permits the most general configuration.

The "Web Client" tab looks like this:

The purpose of the Meridio Connection web client tab is to allow the connection to build a useful URL for each document it indexes. Select the correct protocol, and enter the correct server name, port, and location to reference the Meridio web client service. No proxy information is required, as no documents will be fetched from this service.

The "Credentials" tab looks like this:

Enter the Meridio server credentials needed to access the Meridio system.

When you are done, click the "Save" button to save the connection. You will see a summary screen, looking something like this:

Note that in this example, the Meridio connection is not actually correctly configured, which is leading to an error status message instead of "Connection working".

Since Meridio uses Windows IIS for authentication, there are many ways in which the configuration of either IIS or the Windows domain under which Meridio runs can affect the correct functioning of the Meridio connection. It is beyond the scope of this manual to describe the kinds of

analysis and debugging techniques that might be required to diagnose connection and authentication problems. If you have trouble, you will almost certainly need to involve your Meridio IT personnel. Debugging tools may include (but are not limited to):

- Windows security event logs
- ManifoldCF logs (see below)
- Packet captures (using a tool such as WireShark)

If you need specific ManifoldCF logging information, contact your system integrator.

Jobs based on Meridio connections have the following special tabs: "Search Paths", "Content Types", "Categories", "Data Types", "Security", and "Metadata".

More here later

6.16 Generic RSS Repository Connection

The RSS connection type is specifically designed to crawl RSS feeds. While the Web connection type can also extract links from RSS feeds, the RSS connection type differs in the following ways:

- Links are only extracted from feeds
- Feeds themselves are not indexed
- There is fine-grained control over how often feeds are refetched, and they are treated distinctly from documents in this regard
- The RSS connection type knows how to carry certain data down from the feeds to individual documents, as metadata

Many users of the RSS connection type set up their jobs to run continuously, configuring their jobs to never refetch documents, but rather to expire them after some 30 days. This model works reasonably well for news, which is what RSS is often used for.

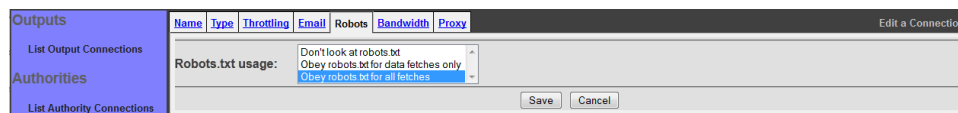
This connection type has no support for any kind of document security, except for hand-entered access tokens provided on a per-job basis.

An RSS connection has the following special tabs: "Email", "Robots", "Bandwidth", and "Proxy". The "Email" tab looks like this:

Enter an email address. This email address will be included in all requests made by the RSS connection, so that webmasters can report any difficulties that their sites experience as the result of improper throttling, etc.

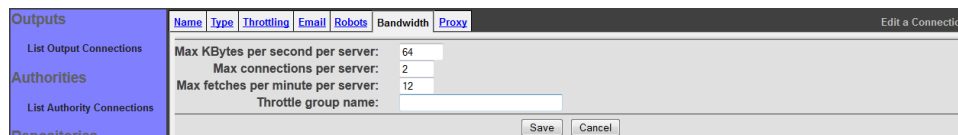
This field is mandatory. While an RSS connection makes no effort to validate the correctness of the email field, you will probably want to remain a good web citizen and provide a valid email address. Remember that it is very easy for a webmaster to block access to a crawler that does not seem to be behaving in a polite manner.

The "Robots" tab looks like this:



Select how the connection will interpret robots.txt. Remember that you have an interest in crawling people's sites as politely as is possible.

The "Bandwidth" tab looks like this:



This tab allows you to control the maximum rate at which the connection fetches data, on a per-server basis, as well as the maximum fetches per minute, also per-server. Finally, the maximum number of socket connections made per server at any one time is also controllable by this tab.

The screen shot displays parameters that are considered reasonably polite. The default values for this table are all blank, meaning that, by default, there is no throttling whatsoever! Please do not make the mistake of crawling other people's sites without adequate politeness parameters in place.

The "Throttle group" parameter allows you to treat multiple RSS-type connections together, for the purposes of throttling. All RSS-type connections that have the same throttle group name will use the same pool for throttling purposes.

The "Bandwidth" tab is related to the throttles that you can set on the "Throttling" tab in the following ways:

- The "Bandwidth" tab sets the maximum values, while the "Throttling" tab sets the average values.
- The "Bandwidth" tab does not affect how documents are scheduled in the queue; it simply blocks documents until it is safe to go ahead, which will use up a crawler thread for the entire period that both the wait and the fetch take place. The "Throttling" tab affects how often documents are scheduled, so it does not waste threads.

Because of the above, we suggest that you configure your RSS connection using both the "Bandwidth" and the "Throttling" tabs. Select maximum values on the "Bandwidth" tab, and corresponding average values estimates on the "Throttling" tab. Remember that a document identifier for an RSS connection is the document's URL, and the bin name for that URL is the server name. Also, please note that the "Maximum number of connections per JVM" field's default value of 10 is unlikely to be correct for connections of the RSS type; you should have at least one available connection per worker thread, for best performance. Since the default number of worker threads is 30, you should set this parameter to at least a value of 30 for normal operation.

The "Proxy" tab allows you to specify a proxy that you want to crawl through. The RSS connection type supports proxies that are secured with all forms of the NTLM authentication method. This is quite typical of large organizations. The tab looks like this:

The screenshot shows the ManifoldCF web interface. On the left is a sidebar with navigation links: 'Outputs' (with sub-links 'List Output Connections', 'Authorities', 'List Authority Connections', and 'Repositories'), 'Outputs', 'Authorities', 'List Authority Connections', and 'Repositories'. The main content area has tabs for 'Name', 'Type', 'Throttling', 'Email', 'Robots', 'Bandwidth', and 'Proxy'. The 'Proxy' tab is selected. It contains fields for 'Proxy host:', 'Proxy port:', 'Proxy authentication domain:', 'Proxy authentication user name:', and 'Proxy authentication password:'. There are 'Save' and 'Cancel' buttons at the bottom right of the form. A link 'Edit a Connection' is visible in the top right corner of the main area.

Enter the proxy server you will be proxying through in the "Proxy host" field. Enter the proxy port in the "Proxy port" field. If your server is authenticated, enter the domain, username, and password in the corresponding fields. Leave all fields blank if you want to use no proxy whatsoever.

When you save your RSS connection, you should see a status screen that looks something like this:

View Repository Connection Status	
Outputs	
List Output Connections	
Authorities	
List Authority Connections	
Repositories	
List Repository Connections	
Jobs	
List all Jobs	
Status and Job Management	
Status Reports	
Document Status	
Queue Status	
History Reports	
Simple History	

Name:	RSS	Description:	
Connection type:	RSS	Max connections:	10
Authority:	None (global authority)		
Throttling:	Bin regular expression	Description	Max avg fetches/min
	No throttles		
Parameters:	Proxy port= Proxy authentication password=***** Max server connections=2 Proxy host= KB per second=64 Robots usage=all Proxy authentication user name= Max fetches per minute=12 Email address=kwright@metacarta.com Proxy authentication domain= Throttle group=		
Connection status:	Connection working		
	Refresh Edit Delete		

Jobs created using connections of the RSS type have the following additional tabs: "URLs", "Canonicalization", "URL mappings", "Exclusions", "Time Values", "Security", "Metadata", and "Dechromed Content". The URLs tab is where you describe the feeds that are part of the job. It looks like this:

View Repository Connection Status	
Outputs	
List Output Connections	
Authorities	
List Authority Connections	
Repositories	
List Repository Connections	
Jobs	
List all Jobs	
Status and Job Management	
Status Reports	
Document Status	
Queue Status	
History Reports	
Simple History	

Name	Connection	Scheduling	Forced Metadata	URLs	Canonicalization	URL Mappings	Exclusions	Time Values	Security	Metadata	Dechromed Content
Save Cancel											

Enter the list of feed URLs you want to crawl, separated by newlines. You may also have comments by starting lines with ("#") characters.

The "Canonicalization" tab controls how the job handles url canonicalization. Canonicalization refers to the fact that many different URLs may all refer to the same actual resource. For example, arguments in URLs can often be reordered, so that a=1&b=2 is in fact the same as b=2&a=1. Other canonical operations include removal of session cookies, which some dynamic web sites include in the URL.

The "Canonicalization" tab looks like this:

View Repository Connection Status	
Outputs	
List Output Connections	
Authorities	
List Authority Connections	

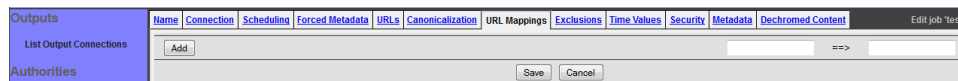
Name	Connection	Scheduling	Forced Metadata	URLs	Canonicalization	URL Mappings	Exclusions	Time Values	Security	Metadata	Dechromed Content
URL regular expression: Description: Reorder? ☐ Remove JSP sessions? ☐ Remove ASP sessions? ☐ Remove PHP sessions? ☐ Remove BV sess											
No canonicalization specified - all URLs will be reordered and have all sessions removed											
Add Save Cancel											

The tab displays a list of canonicalization rules. Each rule consists of a regular expression (which is matched against a document's URL), and some switch selections. The switch selections allow you to specify whether arguments are reordered, or whether certain specific kinds of session cookies are removed. Specific kinds of session cookies that are recognized and can be removed are: JSP (Java applications servers), ASP (.NET), PHP, and Broadvision (BV).

If a URL matches more than one rule, the first matching rule is the one selected.

To add a rule, enter an appropriate regular expression, and make your checkbox selections, then click the "Add" button.

The "Mappings" tab permits you to change the URL under which documents that are fetched will get indexed. This is sometimes useful in an intranet setting because the crawling server might have open access to content, while the users may have restricted access through a somewhat different URL. The tab looks like this:



The "Mappings" tab uses the same regular expression/replacement string paradigm as is used by many connection types running under the Framework. The mappings consist of a list of rules. Each rule has a match expression, which is a regular expression where parentheses "(" and ")" mark sections you are interested in. These sections are called "groups" in regular expression parlance. The replace string consists of constant text plus substitutions of the groups from the match, perhaps modified. For example, "\$ (1)" refers to the first group within the match, while "\$ (1l)" refers to the first match group mapped to lower case. Similarly, "\$ (1u)" refers to the same characters, but mapped to upper case.

For example, suppose you had a rule which had "http://(.*)/(.*)/" as a match expression, and "http://\$(2)/" as the replace string. If presented with the path http://Server/Folder_1/Filename, it would output the string http://Folder_1/Filename.

If more than one rule is present, the rules are all executed in sequence. That is, the output of the first rule is modified by the second rule, etc.

To add a rule, fill in the match expression and output string, and click the "Add" button.

The "Exclusions" tab looks like this:

Here you can enter a set of regular expressions, one per line, which describe which document URLs to exclude from the job. This can be very helpful if you are crawling RSS feeds that include a variety of content where you only want to index a subset of the content.

The "Time Values" tab looks like this:

Fill in the desired time values. A description of each value is below.

Value	Description
Feed connect timeout	How long to wait, in seconds, before giving up, when trying to connect to a server
Default feed refetch time	If a feed specifies no refetch time, this is the time to use instead (in minutes)
Minimum feed refetch time	Never refetch feeds faster than this specified time, regardless of what the feed says (in minutes)
Bad feed refetch time	How long to wait before trying to refetch a feed that contains parsing errors (in minutes, empty is infinity)

The "Security" tab allows you to assign access tokens to the documents indexed with this job. In order to use it, you must first decide what authority connection to use to secure these documents, and what the access tokens from that authority connection look like. The tab itself looks like this:

To add an access token, fill in the text box with the access token value, and click the "Add" button. If there are no access tokens, security will be considered to be "off" for the job.

The "Metadata" tab allows you to specify arbitrary metadata to be indexed along with every document from this job. Documents from connections of the RSS type already receive some metadata having to do with the feed that referenced them. Specifically:

Name	Meaning
PubDate	This contains the document origination time, in milliseconds since Jan 1, 1970. The date is either obtained from the feed, or if it is absent, the date of fetch is included instead.
Source	This is the name of the feed that referred to the document.
Title	This is the title of the document within the feed.
Category	This is the category of the document within the feed.

The "Dechromed Content" tab allows you to index the description of the content from the feed, instead of the document's contents. This is helpful when the description of the documents in the feeds you are crawling is sufficient for indexing purposes, and the actual documents are full of navigation clutter or "chrome". The tab looks like this:

Select the mode you want the connection to operate in.

6.17 Microsoft SharePoint Repository Connection

The Microsoft SharePoint connection type allows you to index documents from a Microsoft SharePoint site. Bear in mind that a single SharePoint

installation actually represents a set of sites. Some sites in SharePoint are directly related to others (e.g. they are subsites), while some sites operate relatively independently of one another.

The SharePoint connection type is designed so that one SharePoint repository connection can access all SharePoint sites from a specific root site through its explicit subsites. It is the case that it is desirable in some very large SharePoint installations to access all SharePoint sites using a single connection. But the ManifoldCF SharePoint connection type does not support that model as of yet. If this functionality is important for you, contact your system integrator.

Documents described by SharePoint connections can be secured in either one of two ways. Either you can choose to secure documents using Active Directory SIDs (in which case, you must use the Active Directory authority type), or you may choose to use native SharePoint groups and users for authorization. The latter must be used in the following cases:

- You have native SharePoint groups or users created which do not correspond to Active Directory SIDs
- Your SharePoint 2010 is configured to use Claims Based authorization mode
- You have ActiveDirectory groups that have more than roughly 1000 members

In general, native SharePoint authorization is the preferred model, except in legacy situations. If you choose to use native SharePoint authorization, you will need to define one or more authorities of type "SharePoint/XXX" associated with the same authority group as your SharePoint connection. Please read the sections of this manual that describe how to configure SharePoint/Native and SharePoint/AD authorities. Bear in mind that SharePoint when configured to run in Claim Space mode (available starting in SharePoint 2010) uses a federated authorization model, so you should expect to create more than one authority when working with a SharePoint Claims Based installation. If your SharePoint is not using Claims Based authorization, then a single authority of type "SharePoint/Native" is sufficient.

If you wish to use the legacy support for the Active Directory authority, then read the section titled "Active Directory Authority Connection" instead.

A SharePoint connection has two special tabs on the repository connection editing screen: the "Server" tab, and the "Authority type" tab. The "Server" tab looks like this:

Select your SharePoint server version from the pulldown. If you do not select the correct server version, your documents may either be indexed with insufficient security protection, or you may not be able to index any documents. Check with your SharePoint system administrator if you are not sure what to select.

SharePoint uses a web URL model for addressing sites, subsites, libraries, and files. The best way to figure out how to set up a SharePoint connection type is therefore to start with your web browser, and visit the topmost root of the site you wish to crawl. Then, record the URL you see in your browser.

Select the server protocol, and enter the server name and port, based on what you recorded from the URL for your SharePoint site. For the "Site path" field, type in the portion of the root site URL that includes everything after the server and port, except for the final "aspx" file. For example, if the SharePoint URL is "http://myserver:81/sites/somewhere/index.aspx", the site path would be "/sites/somewhere".

The SharePoint credentials are, of course, what you used to log into your root site. The SharePoint connection type always requires the user name to be in the form "domain\user".

If your SharePoint server is using SSL, you will need to supply enough certificates for the connection's trust store so that the SharePoint server's SSL server certificate can be validated. This typically consists of either the server certificate, or the certificate from the authority that signed the server certificate. Browse to the local file containing the certificate, and click the "Add" button.

The SharePoint connection "Authority type" tab allows you to select the authorization model used by the connection. It looks like this:

The screenshot shows a configuration window for a connection. At the top, there are tabs: Name, Type, Throttling, Server, and Authority type. The 'Authority type' tab is active, showing 'Authority type: ☐ SharePoint ☒ Active Directory'. Below this, there are 'Save' and 'Cancel' buttons. On the left side, there are links for 'Outputs', 'List Output Connections', 'Authorities', and 'List Authority Groups'.

Select the authority model you wish to use.

After you click the "Save" button, you will see a connection summary screen, which might look something like this:

The screenshot shows the 'View Repository Connection Status' page. The left sidebar contains links for 'Outputs', 'List Output Connections', 'Authorities', 'List Authority Groups', 'List User Mapping Connections', 'List Authority Connections', 'Repositories', 'List Repository Connections', 'Jobs', 'List all Jobs', 'Status and Job Management', 'Status Reports', 'Document Status', 'Queue Status', and 'History Reports'. The main content area shows details for a connection named 'SharePoint'. The details include:

- Name: SharePoint
- Description: My SharePoint connection
- Connection type: SharePoint
- Authority group: None (global authority)
- Max connections: 10
- Throttling: No throttles
- Server SharePoint version: SharePoint Services 2.0 (2003)
- Server protocol: http
- Server name: localhost
- Server port: /sites/somewhere
- Site path: domain\username
- User name: *****
- Password: *****
- SSL certificate list: No certificates present
- Authority type: SharePoint
- Connection status: The site at http://localhost/sites/somewhere did not exist

 At the bottom, there are buttons for 'Refresh', 'Edit', 'Delete', and 'Clear All Related History'.

Note that in this example, the SharePoint connection is not actually referencing a SharePoint instance, which is leading to an error status message instead of "Connection working".

Since SharePoint uses Windows IIS for authentication, there are many ways in which the configuration of either IIS or the Windows domain under which SharePoint runs can affect the correct functioning of the SharePoint connection. It is beyond the scope of this manual to describe the kinds of analysis and debugging techniques that might be required to diagnose connection and authentication problems. If you have trouble, you will almost certainly need to involve your SharePoint IT personnel. Debugging tools may include (but are not limited to):

- Windows security event logs
- ManifoldCF logs (see below)
- Packet captures (using a tool such as WireShark)

If you need specific ManifoldCF logging information, contact your system integrator.

When you configure a job to use a repository connection of the generic database type, several additional tabs are presented. These are, in order, "Paths", "Security", and "Metadata".

The "Paths" tab allows you to build a list of rules describing the SharePoint content that you want to include in your job. When the SharePoint connection type encounters a subsite, library, list, or file, it looks through this list of rules to determine whether to include the subsite, library, list, or file. The first matching rule will determine what will be done.

Each rule consists of a path, a rule type, and an action. The actions are "Include" and "Exclude". The rule type tells the connection what kind of SharePoint entity it is allowed to exactly match. For example, a "File" rule will only exactly match SharePoint paths that represent files - it cannot exactly match sites or libraries. The path itself is just a sequence of characters, where the "*" character has the special meaning of being able to match any number of any kind of characters, and the "?" character matches exactly one character of any kind.

The rule matcher extends strict, exact matching by introducing a concept of implicit inclusion rules. If your rule action is "Include", and you specify (say) a "File" rule, the matcher presumes implicit inclusion rules for the corresponding site and library. So, if you create an "Include File" rule that matches (for example) `/MySite/MyLibrary/MyFile`, there is an implied "Site Include" rule for `/MySite`, and an implied "Library Include" rule for `/MySite/MyLibrary`. Similarly, if you create a "Library Include" rule, there is an implied "Site Include" rule that corresponds to it. Note that these shortcuts only applies to "Include" rules - there are no corresponding implied "Exclude" rules.

The "Paths" tab allows you to build these rules one at a time, and add them either to the bottom of the list, or insert them into the list of rules at any point. Either way, you construct the rule you want to append or insert by first constructing the path, from left to right, using your choice of text and context-dependent pull-downs with existing server path information listed. This is what the tab may look like for you. Bear in mind that if you are using a connection that does not display the status, "Connection working", you may not see the selections you should in these pull-downs:

To build a rule, first build the rule's matching path. Make an appropriate selection or enter desired text, then click either the "Add Site", "Add Library", "Add List", or "Add Text" button, depending on your choice. Repeat this process until the path is what you want it to be. At this point, if the SharePoint connection does not know what kind of entity your path describes, you will need to select the SharePoint entity type that you want the rule to match also. Select whether this is an include or exclude rule. Then, click the "Add New Rule" button, to add your newly-constructed rule at the end of the list.

The "Security" tab allows you to specify whether SharePoint's security model should be applied to this set of documents, or not. You also have the option of applying some specified set of access tokens to the documents described by the job. The tab looks like this:

Select whether SharePoint security is on or off using the radio buttons provided. If security is off, you may add access tokens in the text box and click the "Add" button. The access tokens must be in the proper form expected by the authority that governs your SharePoint connection for this feature to be useful.

The "Metadata" tab allows you to specify what metadata will be included for each document. The tab is similar to the "Paths" tab, which you may want to review above:

The main difference is that instead of rules that include or exclude individual sites, libraries, lists, or documents, the rules describe inclusion

and exclusion of document or list item metadata. Since metadata is associated with files and list items, all of the metadata rules are applied only to file paths and list item paths, and there are no such things as "site" or "library" metadata path rules.

If an exclusion rule matches a file's path, it means that no metadata from that file will be included at all. There is no way to individually exclude a single field using an exclusion rule.

To build a rule, first build the rule's matching path. Make an appropriate selection or enter desired text, then click either the "Add Site", "Add Library", "Add List", or "Add Text" button, depending on your choice. Repeat this process until the path is what you want it to be. Select whether this is an include or exclude rule. Either check the box for "Include all metadata", or select the metadata you want to include from the pulldown. (The choices of metadata fields you are presented with are determined by which SharePoint library is selected. If your rule path does not uniquely specify a library, you cannot select individual fields to include. You can only select "All metadata".) Then, click the "Add New Rule" button, to put your newly-constructed rule at the end of the list.

You can also use the "Metadata" tab to have the connection send path data along with each document, as a piece of document metadata. To enable this feature, enter the name of the metadata attribute you want this information to be sent into the "Attribute name" field. Then, add the rules you want to the list of rules. Each rule has a match expression, which is a regular expression where parentheses "(" and ")" mark sections you are interested in. These sections are called "groups" in regular expression parlance. The replace string consists of constant text plus substitutions of the groups from the match, perhaps modified. For example, "\$ (1)" refers to the first group within the match, while "\$ (1l)" refers to the first match group mapped to lower case. Similarly, "\$ (1u)" refers to the same characters, but mapped to upper case.

For example, suppose you had a rule which had ".*/(.*)/(.*)/*" as a match expression, and "\$ (1) \$ (2)" as the replace string. If presented with the path Project/Folder_1/Folder_2/Filename, it would output the string Folder_1 Folder_2.

If more than one rule is present, the rules are all executed in sequence. That is, the output of the first rule is modified by the second rule, etc.

Example: How to index a SharePoint 2010 Document Library

Let's say we want to index a Document Library named Documents. The following URL displays contents of the library : <http://iknow/Documents/Forms/AllItems.aspx>

☐	Type	Name	Modified	Approval Status
		Bordro_Ozluk	02.05.2012 19:23	Approved
		Egitim_Gelisim	02.05.2012 19:23	Approved
		Global_de_Yasam	02.05.2012 19:23	Approved
		Idari_Isler	02.05.2012 19:24	Approved
		Ise_Alim_Kariyer	02.05.2012 19:24	Approved
		ik_docs	28.03.2012 10:47	Approved
		Organizasyon_Yonetimi	02.05.2012 19:24	Approved
		Ucret_Yan_Haklar	02.05.2012 19:24	Approved
		Vekaletname	04.04.2012 10:07	Draft

Note that there exists eight folders and one file in our library. Some folders have sub-folders. And leaf folders contain files. The following single Path Rule is sufficient to index all files in Documents library.

	Path match	Type	Action
Insert New Rule			
Delete	/Documents/*	file	include
Add New Rule			

If we click Library Tools > Library > Modify View, we will see complete list of all available metadata.

View Name:

Web address of this view:
 http://iknow/Documents/Forms/.aspx 

This view appears by default when visitors follow a link to this document library. If you want to delete this view, first make another view the default.

Display	Column Name	Position from Left
☒	Type (icon linked to document)	1
☒	Name (linked to document with edit menu)	2
☒	Modified	3
☒	Approval Status	4
☒	Created	5
☒	Title	6
☒	ID	7
☐	Approver Comments	8
☐	Check In Comment	9
☐	Checked Out To	10

ManifoldCF's UI also displays all available Document Libraries and their associated metadata too. Using this pulldown, you can select which fields you want to index.

New rule: ☐ Include all metadata

To create the metadata rule below, click Metadata tab in Job settings. Select Documents from --Select library-- and "Add Library" button. As

soon as you have done this, all available metadata will be listed. Enter * in the textbox which is right of the Add Text button. And click Add Text button. When you have done this Path Match becomes /Documents/*. After this you can multi select list of metadata. This action will populate Fields with CheckoutUser, Created, etc. Click Add New Rule button. This action will add this new rule to your Metadata rules.

Delete	/Documents/*	include	false	CheckoutUser, Created
--------	--------------	---------	-------	-----------------------

Finally click the "Save" button at the bottom of the page. You will see a page looking something like this:

Path rules:	Path match	Rule type		Action
	/Documents/*	file		include
Metadata:	Path match	Action	All metadata?	Fields
	/Documents/*	include	false	CheckoutUser, Created

Some Final Notes

- If you don't add * to Patch match rule, your selected fields won't be used. In other words Path match rule /Documents won't match the document /Documents/ik_docs/diger/sorular.docx
- We can include all metadata using the checkbox. (without selecting from the pulldown list)
- If we were to index only docx files, our Patch match rule would be / Documents/*.docx

Example: How to index SharePoint 2010 Lists

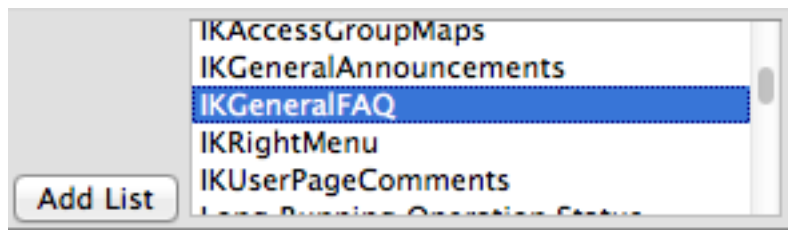
Lists are a key part of the architecture of Windows SharePoint Services. A document library is another form of a list, and while it has many similar properties to a standard list, it also includes additional functions to enable document uploads, retrieval, and other functions to support document management and collaboration. [1]

An item added to a document library (and other libraries) must be a file. You can't have a library without a file. A list on the other hand doesn't have a file, it is just a piece of data, just like SQL Table.

Let's say we want to index a List named IKGeneralFAQ. The following URL displays contents of the list : <http://iknow/Lists/IKGeneralFAQ/AllItems.aspx>

ID	IKFAQPageID	IKFAQPage	Title	IKFAQAnswer
2539	225	Pages/Acil-Durum-Yonetimi.aspx	Acil durumlarda yapılması gerekeler nelerdir?	Bu bilgilerin bulunduğu Güvenlik el kitabına QDMS üzerinden ulaşabilirsiniz.
2540	225	Pages/Acil-Durum-Yonetimi.aspx	Acil durum tatbikatları ne zaman yapılır? Çalışanların tatbikatlardan haberi olur mu?	Tatbikatların tümü haberli olarak yapılmakta, tatbikat öncesinde tüm lokasyon geneline bilgilendirme maili atılmaktadır.Yılda bir kez her lokasyonda acil durum tatbikatları yapılmaktadır.
2541	225	Pages/Acil-Durum-Yonetimi.aspx	Acil durumlarda kimleri arayacağız ?	http://benimyerim/specialsections/hr/Pages/Idarisler.aspx linkinden bilgi alabilirsiniz.

Note that the Lists do not have files. It looks like an Excel spreadsheet. In ManifoldCF Job Settings, Path tab displays all available Lists. It lets you to select the name of the list you want to index.



After we select IKGeneralFAQ, hit Add List button and Save button, we have the following Path Rule:

Path rules:	Path match	Rule type	Action
	/IKGeneralFAQ	list	include

The above single Path Rule is sufficient to index content of IKGeneralFAQ List. Note that unlike the document libraries, we don't need * here.

If we click List Tools > List > Modify View we will see complete list of all available metadata.

Display	Column Name	Position from Left
☒	ID	1
☒	IKFAQPageID	2
☒	IKFAQPage	3
☒	Title (linked to item with edit menu)	4
☒	IKFAQAnswer	5
☒	Title	6

ManifoldCF's Metadata UI also displays all available Lists and their associated metadata too. Using this pulldown, you can select which fields you want to index.

Add New Rule

New rule: /IKGeneralFAQ/*

Include
Exclude

☐ Include all metadata

ID
IKFAQAnswer
IKFAQCategory
IKFAQPage
IKFAQPageID

To create the metadata rule below, click Metadata tab in Job settings. Select IKGeneralFAQ from --Select list-- and "Add List" button. As soon as you have done this, all available metadata will be listed. After this you can multi select list of metadata. This action will populate Fields with ID, IKFAQAnswer, IKFAQPage, IKFAQPageID, etc. Click Add New Rule button. This action will add this new rule to your Metadata rules.

Metadata:	Path match	Action	All metadata?	Fields
	/IKGeneralFAQ/*	include	false	ID, IKFAQAnswer, IKFAQPage, IKFAQPageID, Title

Finally click the "Save" button at the bottom of the page. You will see a page looking something like this:

Path rules:	Path match			Rule type	Action
	/IKGeneralFAQ			list	include
Metadata:	Path match	Action	All metadata?	Fields	
	/IKGeneralFAQ/*	include	false	ID, IKFAQAnswer, IKFAQPage, IKFAQPageID, Title	

Some Final Notes

- Note that, when specifying Metadata rules, UI automatically adds * to Path match rule for Lists. This is not the case with Document Libraries.

- We can include all metadata using the checkbox. (without selecting from the pulldown list)

6.18 Generic Web Repository Connection

The Web connection type is effectively a reasonably full-featured web crawler. It is capable of handling most kinds of authentication (basic, all forms of NTLM, and session-based), and can extract links from the following kinds of documents:

- Text
- HTML
- Generic XML
- RSS feeds

The Web connection type differs from the RSS connection type in the following respects:

- Feeds are indexed, if the output connection accepts them
- Links are extracted from all documents, not just feeds
- Feeds are treated just like any other kind of document - you cannot control how often they refetch independently
- There is support for limiting crawls based on hop count
- There is support for controlling exactly what URLs are considered part of the set, and which are excluded

In other words, the Web connection type is neither as easy to configure, nor as well-targeted in its separation of links and data, as the RSS connection type. For that reason, we strongly encourage you to consider using the RSS connection type for all applications where it might reasonably apply.

Many users of the Web connection type set up their jobs to run continuously, configuring their jobs to occasionally refetch documents, or to not refetch documents ever, and expire them after some period of time.

This connection type has no support for any kind of document security, except for hand-entered access tokens provided on a per-job basis.

A Web connection has the following special tabs: "Email", "Robots", "Bandwidth", "Access Credentials", and "Certificates". The "Email" tab looks like this:

Enter an email address. This email address will be included in all requests made by the Web connection, so that webmasters can report any difficulties that their sites experience as the result of improper throttling, etc.

This field is mandatory. While a Web connection makes no effort to validate the correctness of the email field, you will probably want to remain a good web citizen and provide a valid email address. Remember that it is very easy for a webmaster to block access to a crawler that does not seem to be behaving in a polite manner.

The "Robots" tab looks like this:

Select how the connection will interpret robots.txt and <meta name="robots ...> tags on HTML pages. Remember that you have an interest in crawling people's sites as politely as is possible.

The "Bandwidth" tab allows you to specify a list of bandwidth rules. Each rule has a regular expression matched against a URL's throttle bin. Throttle bins, in connections of the Web type, are simply the server name part of the URL. Each rule allows you to select a maximum bandwidth, number of connections, and fetch rate. You can have as many rules as you like; if a URL matches more than one rule, then the most conservative value will be used.

This is what the "Bandwidth" tab looks like:

The screen shot shows the tab configured with a setting that is reasonably polite. The default value for this tab is blank, meaning that, by default, there is no throttling whatsoever! Please do not make the

mistake of crawling other people's sites without adequate politeness parameters in place.

To add a rule, fill in the regular expression and the appropriate rule limit values, and click the "Add" button.

The "Bandwidth" tab is related to the throttles that you can set on the "Throttling" tab in the following ways:

- The "Bandwidth" tab sets the maximum values, while the "Throttling" tab sets the average values.
- The "Bandwidth" tab does not affect how documents are scheduled in the queue; it simply blocks documents until it is safe to go ahead, which will use up a crawler thread for the entire period that both the wait and the fetch take place. The "Throttling" tab affects how often documents are scheduled, so it does not waste threads.

Because of the above, we suggest that you configure your Web connection using both the "Bandwidth" and the "Throttling" tabs. Select maximum values on the "Bandwidth" tab, and corresponding average values estimates on the "Throttling" tab. Remember that a document identifier for a Web connection is the document's URL, and the bin name for that URL is the server name. Also, please note that the "Maximum number of connections per JVM" field's default value of 10 is unlikely to be correct for connections of the Web type; you should have at least one available connection per worker thread, for best performance. Since the default number of worker threads is 30, you should set this parameter to at least a value of 30 for normal operation.

The Web connection's "Access Credentials" tab describes how pages get authenticated. There is support on this tab for both page-based authentication (e.g. basic auth or all forms of NTLM), as well as session-based authentication (which involves the fetch of many pages to establish a logged-in session). The initial appearance of the "Access Credentials" tab shows both kinds of authentication:

Comparing Page and Session Based Authentication:

Authentication Detail	Page Based Authentication	Session Based Authentication
HTTP Return Codes	4xx range, usually 401	Usually 3xx range, often 301 or 302
How it's recognized as a login request	4xx range codes always indicate a challenged response	Recognized by patterns in the URL or content. Manifold must be told what to look for. 3xx range HTTP codes are also used for normal content redirects so there's no built-in way for Manifold to tell the difference, that's why it needs regex-based rules.
How Login form is Rendered in normal Web Browser	Standard Browser popup dialog. IE, Firefox, Safari, etc. all have their own specific style.	Server sends custom HTML or Javascript. Might use red text, might not. Might show a login form, or maybe a "click here to login" link. Can be a regular page, or Javascript popup, there's no specific standard.
Login Expiration	Usually doesn't expire, depends on server's policy. If it does expire at all, usually based calendar dates and not related to this specific login.	Often set to several minutes or hours from the the last login in current browser session. A long spider run might need to re-login several times.
HTTP Header Fields	From server: WWW-Authenticate: Basic or NTLM with Realm From client: Authorization: Basic or NTLM	From server: Location: and Set-Cookie: From client: Cookie: Cookie values frequently change.

Each kind of authentication has its own list of rules.

Specifying a page authentication rule requires simply knowing what URLs are protected, and what the proper authentication method and credentials are for those URLs. Enter a regular expression describing the protected URLs, and select the proper authentication method. Fill in the credentials. Click the "Add" button.

Specifying a correct session authentication rule usually requires some research. A single session-authentication rule usually corresponds to a single session-protected site. For that site, you will need to be able to describe the following for session authentication to function:

- The URLs of pages that are protected by this particular site session security
- How to detect when a page fetch is part of the login sequence
- How to fill in the appropriate forms within the login sequence with appropriate login information

A Web connection labels pages that are part of the login sequence "login pages", and pages that are protected site content "content pages". A Web connection will not attempt to index login pages. They are special pages that have but one purpose: establishing an authenticated session.

Remember, the goals of the setup you have to go through are as follows:

- Identify what site, or part of the site, has protected content
- Identify which http/https fetches are not content, but are in fact part of a "login sequence", which a normal person has to go through to get the appropriate cookies

If all this is not complicated enough, your research also has to cover two very different cases: when you are first entering the site anew, and second when you try to fetch a content page and you are no longer logged in, because your session has expired. In both cases, the session authentication rule must be able to properly log in and fetch content, because you cannot control when a page will be fetched or refetched by the Framework.

One key piece of data you will supply is a regular expression that basically describes the set of URLs for which the content is protected, and for which the right cookies have to be in place for you to get at the "real" content. Once you've specified this, then for each protection zone (described by its URL regexp), you need to specify how ManifoldCF should identify whether a given fetch should be considered part of the login sequence or not. It's not enough to just identify the URL of login pages, since (for instance) if your session has expired you may well have a redirection get fetched instead of the content you want. So you specify each class of login page as one of three types, using not only the URL to identify the class (this is where you get the second regexp), but also something about what is on the page: whether it is a redirection to a URL

(yes, again described by a URL regexp), whether it has a form with a specified name (described by a regexp), or whether it has a specific link on it (once again, described by a regexp).

As you can see, you declare a page to be a login page by identifying it both by its URL, and by what the crawler finds on the page when it fetches it. For example, some session-protected sites may redirect you to a login screen when your session expires. So, instead of fetching content, you would be fetching a redirection to a specific page. You do not want either the redirection, or the login screen, to be considered content pages. The correct way to handle such a setup would be to declare one kind of login page to consist of a redirection to the login screen URL, and another kind of login page to consist of the login screen URL with the appropriate form. Furthermore, you would want to supply the correct login data for the form, and allow the form to be submitted, and so the login form's target may also need to be declared as a login page.

The kinds of content that a Web connection can recognize as a login page are the following:

- A redirection to a specific URL, as described by a regular expression
- A page that has a form of a particular name on it, as described by a regular expression
- A page that has a link on it to a specific target, as described by a regular expression
- A page that has specific content on it, as described by a regular expression

Note that in three of the cases above that there is an implicit flow through the login sequence that you describe by specifying the pages in the login sequence. For example, if upon session timeout you expect to see a redirection to a link, or family of links (remember, it's a regexp, so you can describe that easily), then as part of identifying the redirection as belonging to the login sequence, the web connector also now has a new link to fetch - the redirection link - which is what it does next. The same applies to forms. If the form name that was specified is found, then the web connector submits that form using values for the form elements that you specify, and using the submission type described in the actual form tag (GET, POST, or multi-part). Any other elements of the form are left in whatever state that the HTML specified; no Javascript is ever evaluated. Thus, if you think a form element's value is being set by Javascript, you have to figure out what it is being set to and enter this

value by hand as part of the specification for the "form" type of login page. Typically this amounts to a user name and password.

In the fourth login sequence case, where specific page content is matched to determine that a page belongs in the login sequence, there is no implicit flow to a subsequent page. In this case you must supply an *override URL*, which describes which page to go to to continue the login sequence. In fact, you are allowed to provide an override URL for all four cases above, but this is only recommended when the web connector would not automatically find the right subsequent page URL on its own.

To add a session authentication rule, fill in a regular expression describing the site pages that are being protected, and click the "Add" button:

Note that you can now add login page descriptions to the newly-created rule. To add a login page description, enter a URL regular expression, a type of login page, a target link or form name regular expression, and click the "Add" button.

When you add a login page of the "form" type, you can then add form fill-in information to the login page, as seen below:

Supply a regular expression for the name of the form element you want to set, and also provide a value. If you want the value to not be visible in clear text, fill in the "password" column instead of the "value" column. You can usually figure out the name of the form and its elements by

viewing the source of the HTML page in a browser. When you are done, click the "Add" button.

Form data that is not specified will be posted with the default value determined by the HTML of the page. The Web connection type is unable, at this time, to execute Javascript, and therefore you may need to fill out some form values that are filled in by Javascript in order to get the form to post in a useful way. If you have a form that relies heavily on Javascript to post properly, you may need considerable effort and web programming skills to figure out how to get these forms to post properly with a Web connection. Luckily, such obfuscated login screens are still rare.

A series of login pages form a "login page sequence" for the site. For each login page, the Web connection decides what page to fetch next by what you specified for the login page criteria. So, for a redirection to a specific URL, the next page to be fetched will be that redirected URL. For a form, the next page fetched will be the action page indicated by the specified form. For a link to a target, the next page fetched will be the target URL. When the login page sequence ends, the next page fetched after that will be the original content page that the Web connection was trying to fetch when the login sequence started.

Debugging session authentication problems is best done by looking at a Simple History report for your Web connection. A Web connection records several types of events which, between them, can give a very strong picture of what is happening. These event types are as follows:

Event type	Meaning
Fetch	This event records the fetch of a URL. The HTTP response is recorded as the response code. In addition, there are several negative code values which the connect generates when the HTTP operation cannot be done or does not complete.
Begin login	This event occurs when the connection detects the transition to a login page sequence. When a login sequence is entered, no other pages from that protected site will be fetched until the login sequence is completed.

End login	This event occurs when the connection detects the transition from a login page sequence back to normal content fetching. When this occurs, simultaneous fetching for pages from the site are re-enabled.
-----------	--

The "Certificates" tab is used in conjunction with SSL, and permits you to define independent trust certificate stores for URLs matching specified regular expressions. You can also allow the connection to trust all certificates it sees, if you so choose. The "Certificates" tab looks like this:

Type in a URL regular expression, and either check the "Trust everything" box, or browse for the appropriate certificate authority certificate that you wish to trust. (It will also work to simply trust a server's certificate, but that certificate may change from time to time, as it expires.) Click "Add" to add the certificate rule to the list.

When you are done, and you click the "Save" button, you will see a summary page looking something like this:

When you create a job that uses a repository connection of the Web type, the tabs "Hop Filters", "Seeds", "Canonicalization", "Inclusions", "Exclusions", "Security", and "Metadata" will all appear. These tabs allow you to configure the job appropriately for a web crawl.

The "Hop Filters" tab allows you to specify the maximum number of hops from a seed document that a document can be before it is no longer considered to be part of the job. For connections of the Web type, there

are two different kinds of hops you can count as well: "link" hops, and "redirection" hops. Each of these represents an independent count and cutoff value. A blank value means no cutoff value at all.

For example, if you specified a maximum "link" hop count of 5, and left the "redirect" hop count blank, then any document that requires more than five links to reach from a seed will be considered out-of-set. If you specified both a maximum "link" hop count of 5, and a maximum "redirect" hop count 2, then any document that requires more than five links to reach from a seed, and more than two redirections, will be considered out-of-set.

The "Hop Filters" tab looks like this:

On this same tab, you can tell the Framework what to do should there be changes in the distance from the root to a document. The choice "Delete unreachable documents" requires the Framework to recalculate the distance to every potentially affected document whenever a change takes place. This may require expensive bookkeeping, however, so you also have the option of ignoring such changes. There are two varieties of this latter option - you can ignore the changes for now, with the option of turning back on the aggressive bookkeeping at a later time, or you can decide not to ever allow changes to propagate, in which case the Framework will discard the necessary bookkeeping information permanently. This last option is the most efficient.

The "Seeds" tab is where you enter the starting points for your crawl. It looks like this:

Enter a list of seeds, separated by newline characters. Blank lines, or lines that begin with a '#' character, will be ignored.

The "Canonicalization" tab controls how a web job converts URLs into a standard form. It looks like this:

The tab displays a list of canonicalization rules. Each rule consists of a regular expression (which is matched against a document's URL), and some switch selections. The switch selections allow you to specify whether arguments are reordered, or whether certain specific kinds of session cookies are removed. Specific kinds of session cookies that are recognized and can be removed are: JSP (Java applications servers), ASP (.NET), PHP, and Broadvision (BV).

If a URL matches more than one rule, the first matching rule is the one selected.

To add a rule, enter an appropriate regular expression, and make your checkbox selections, then click the "Add" button.

The "Inclusions" tab lets you specify, by means of a set of regular expressions, exactly what URLs will be included as part of the document set for a web job. The tab looks like this:

You will need to provide a series of zero or more regular expressions, separated by newlines. The regular expressions are considered to match if they are found anywhere within the URL. They do not need to match the entire URL.

Remember that, by default, a web job includes all documents in the world that are linked to your seeds in any way that the web connection type can determine.

If you wish to restrict which documents are actually processed within your overall set of included documents, you may want to supply some regular expressions on the "Exclusions" tab, which looks like this:



Once again you will need to provide a series of zero or more regular expressions, separated by newlines. The regular expressions are considered to match if they are found anywhere within the URL. They do not need to match the entire URL.

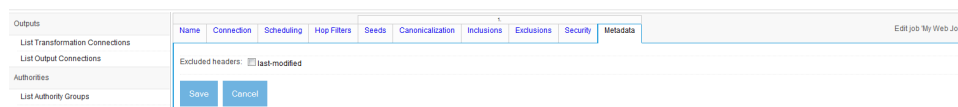
It is typical to use the "Exclusions" tab to remove documents from consideration which are suspected to contain content that both has no extractable links, and is not useful to the index you are trying to build, e.g. movie files.

The "Security" tab allows you to specify the access tokens that the documents in the web job get indexed with, and looks like this:



You will need to know the format of the access tokens for the governing authority before you can add security to your documents in this way. Enter the access token you desire and click the "Add" button.

The "Metadata" tab allows you to exclude specific optional HTTP header metadata along with all documents belonging to a web job. (A standard set of "fixed" HTTP headers are always included.) It looks like this:



Select the desired HTTP header metadata you wish to exclude.

6.19 Windows Share/DFS Repository Connection

The Windows Share connection type allows you to access content stored on Windows shares, even from non-Windows systems. Also supported are Samba and various third-party Network Attached Storage servers.

DFS nodes and referrals are fully supported, provided the referral machine names can be looked up properly via DNS on the server where the Framework is running. For each document, a Windows Share connection creates an index identifier that can be either a "file:" IRI's, or a mapped "http:" URI's, depending on how it is configured. This allows for a great deal of flexibility in deployment environments, but also may require some work to properly set up. In particular, if you intend to use file IRI's as your identifiers, you should check with your system integrator to be sure these are being handled properly by the search component of your system. When you use a browser such as Internet Explorer to view a document from a Windows file system called `\\servername\sharename\dir1\filename.txt`, the browser converts that to an IRI that looks something like this: `file:///servername/sharename/dir1/filename.txt`. While this seems simple, major complexities arise when the underlying file name has special characters in it, such as spaces, "#" symbols, or worse still, non-ASCII characters. Unfortunately, every version of Internet Explorer handles these situations somewhat differently, so there is not any fully correct way for the Windows Share connection type to convert file names to IRI's. Instead, the connection always uses a standard canonical form, and expects the search results display system component to know how to properly form the right IRI for the browser or client being used.

If you are interested in enforcing security for documents crawled with a Windows Share repository connection, you will need to first configure an authority connection of the Active Directory type to control access to these documents. The Share/DFS connector type can also be used with the LDAP authority connection type.

A Windows Share connection has a single special tab on the repository connection editing screen: the "Server" tab:

You must enter the name of the server to form the connection with in the "Server" field. This can either be an actual machine name, or a domain name (if you intend to connect to a Windows domain-based DFS root). If you supply an actual machine name, it is usually the right thing to do to provide the server name in an unqualified form, and provide a fully-qualified domain name in the "Domain name" field. The user name also should usually be unqualified, e.g. "Administrator" rather than "Administrator@mydomain.com". Sometimes it may work to leave the "Domain name" field blank, and instead supply a fully-qualified machine name in the "Server" field. It never works to supply both a domain name and a fully-qualified server name.

The "Use SIDs" checkbox allows you to control whether the connection uses SIDs as access tokens (which is appropriate for Windows servers and NAS servers that are secured by Active Directory), or user/group names (which is appropriate for Samba servers, and other CIFS servers that use LDAP for security, in conjunction with the LDAP Authority connection type). Check the box to use SIDs.

Please note that you should probably set the "Maximum number of connections per JVM" field, on the "Throttling" tab, to a number smaller than the default value of 10, because Windows is not especially good at handling multithreaded file requests. A number less than 5 is likely to perform as well with less chance of causing server-side problems.

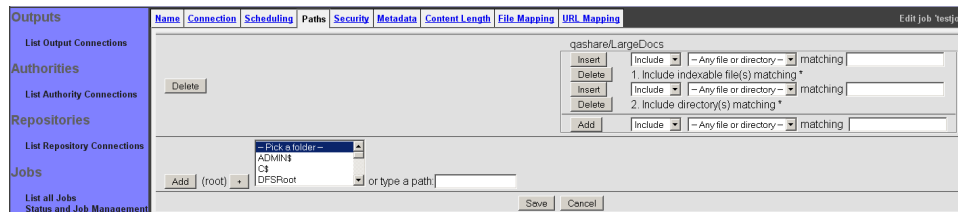
After you click the "Save" button, you will see a connection summary screen, which might look something like this:

Outputs	View Repository Connection Status			
List Output Connections	Name: ShareTest		Description:	
Authorities	Connection type:	Windows Share Connector	Max connections: 10	
List Authority Connections	Authority:	AD		
Repositories	Throttling:	Bin regular expression No throttles	Description	Max avg fetches/min
List Repository Connections	Parameters:	Password=***** Domain/Realm=mydomain.com User Name=Administrator Server=myserver/		
Jobs	Connection status:	Could not connect: smb://myserver/		
List all Jobs Status and Job Management	Refresh Edit Delete			

Note that in this example, the Windows Share connection is not responding, which is leading to an error status message instead of "Connection working".

When you configure a job to use a repository connection of the Windows Share type, several additional tabs are presented. These are, in order, "Paths", "Security", "Metadata", "Content Length", "File Mapping", and "URL Mapping".

The "Paths" tab looks like this:



This tab allows you to construct starting-point paths by drilling down, and then add the constructed paths to a list, or remove existing paths from the list. Without any starting paths, your job includes zero documents.

Make sure your connection has a status of "Connection working" before you open this tab, or you will see an error message, and you will not be able to build any paths.

For each included path, a list of rules is displayed which determines what folders and documents get included with the job. These rules will be evaluated from top to bottom, in order. Whichever rule first matches a given path is the one that will be used for that path.

Each rule describes the path matching criteria. This consists of the file specification (e.g. "*.txt"), whether the path is a file or folder name, and whether a file is considered indexable or not by the output connection. The rule also describes the action to take should the rule be matched: include or exclude. The file specification character "*" is a wildcard which matches zero or more characters, while the character "?" matches exactly one character. All other characters must match exactly.

Remember that your specification must match all characters included in the file's path. That includes all path separator characters ("/"). The path you must match always begins with an initial path separator. Thus, if you want to exclude the file "foo.txt" at the root level, your exclude rule must match "/foo.txt".

To add a rule for a starting path, select the desired values of all the pulldowns, type in the desired file criteria, and click the "Add" button. You may also insert a new rule above any existing rule, by using one of the "Insert" buttons.

The "Security" tab looks like this:

The "Security" tab lets you control three things: File security, share security, and (if security is off) the security tokens attached to all documents indexed by the job.

File security is the security Windows applies to individual files. This kind of security is supported by practically all Windows-compatible NAS-type servers, so you may use this feature without cause for concern.

Share security is the security Windows applies to Windows shares. This is an older kind of security that is no longer prevalent in most enterprise organizations. Many modern NAS systems and Samba also do not support this security model. If you enable this kind of security in your job while crawling against a system that does not support it, your job will not run correctly; the first document access will cause an error, and the job will abort.

If you turn off file security, you have the option of adding index access tokens of your own to all documents crawled by the job. These tokens must, of course, be in a form appropriate for the governing authority connection. Type the token into the box and click the "Add" button. It is unusual to use this feature other than for demonstrations.

The "Metadata" tab looks like this:

This tab allows you to ingest a document's path, as modified by a set of regular expression rules, as a piece of document metadata. Enter the metadata name you want in the "Path attribute name" field. Then, add the rules you want to the list of rules. Each rule has a match expression, which is a regular expression where parentheses "(" and ")" mark sections you are interested in. These sections are called "groups" in regular expression parlance. The replace string consists of constant text plus substitutions of the groups from the match, perhaps modified. For example, "\$ (1)" refers to the first group within the match, while "\$ (1l)" refers to the first match group mapped to lower case. Similarly, "\$ (1u)" refers to the same characters, but mapped to upper case.

For example, suppose you had a rule which had `".*/(.*)/(.*)/.*"` as a match expression, and `"$(1) $(2)"` as the replace string. If presented with the path `Project/Folder_1/Folder_2/Filename`, it would output the string `Folder_1 Folder_2`.

If more than one rule is present, the rules are all executed in sequence. That is, the output of the first rule is modified by the second rule, etc.

The "Content Length" tab looks like this:

This tab allows you to set a maximum content length cutoff value, to avoid having the job try to index exceptionally large documents. Enter the desired maximum value. A blank value indicates an unlimited cutoff length.

The "File Mapping" tab looks like this:

The mappings specified here are similar in all respects to the path attribute mapping setup described above. The mappings are applied to change the actual file path discovered by the crawler into a different file path. This can sometimes be useful if there is some kind of conversion process between raw documents and parallel data files that contain extracted data.

The "URL Mapping" tab looks like this:

The mappings specified here are similar in all respects to the path attribute mapping setup described above. If no mappings are present, the file path is converted to a canonical file IRI. If mappings are present, the conversion is presumed to produce a valid URL, which can be used to access the document via some variety of Windows Share http server.

Accessing some servers may result in "Couldn't connect to server: Logon failure: unknown user name or bad password"

Connection Status, because the default version of NTLM used for authentication is incompatible. If this is the case, the Windows

Share repository connector can be configured to use NTLMv1, rather than the NTLMv2 default. This is done by setting the property "org.apache.manifoldcf.crawler.connectors.jcifs.usentlmv1" to "true" in properties.xml file.

6.20 Wiki Repository Connection

The Wiki repository connection type allows you to index content from the main space of a Wiki or MediaWiki site. The connection type uses the Wiki API in order to fetch content. Only publicly visible documents will be indexed, and there is thus typically no need of an authority for Wiki content.

This connection type has no support for any kind of document security, except for hand-entered access tokens provided on a per-job basis.

A Wiki connection has only one special tab on the repository connection editing screen: the "Server" tab. The "Server" tab looks like this:

The screenshot shows a web-based configuration interface for a Wiki repository connection. On the left, there is a sidebar with links for 'Outputs', 'List Output Connections', 'Authorities', and 'List Authority Connections'. The main area has tabs for 'Name', 'Type', 'Throttling', and 'Server'. The 'Server' tab is active, displaying a form with the following fields: 'Protocol' (a dropdown menu showing 'http'), 'Server name' (a text input field), 'Port' (a text input field), and 'Path name' (a text input field containing '/w'). At the bottom right of the form are 'Save' and 'Cancel' buttons. An 'Edit a Connection' link is visible in the top right corner of the main area.

The protocol must be selected in the "Protocol" field. At the moment only the "http" protocol is supported. The server name must be provided in the "Server name" field. The server port must be provided in the "Port" field. Finally, the path part of the Wiki URL must be provided in the "Path name" field and must start with a "/" character.

When you configure a job to use a repository connection of the Wiki type, no additional tabs are currently presented.

7 Notification Connection Types

7.1 Slack Notifications

The Slack notification connection allows you to send job notifications to a Slack channel. The connection type uses the Slack Incoming WebHook API in order to deliver messages to Slack.

A Slack notification connection has only one special tab on the notification connection editing screen: the "Slack WebHook" tab. The "Slack WebHook" tab looks like this:

Name	Type	Throttling	Slack WebHook	Edit a notification connection
WebHook URL: https://hooks.slack.com/services/				
Proxy Host:				
Proxy Port:				
Proxy NTLM Username:				
Proxy NTLM Password:				
Proxy NTLM Authentication Domain:				
Save Cancel				

When you configure a job to use a notification connection of the Slack notification type, an additional tab "Message" is presented.

The "Messages" tab looks like this:

Name	Connection	Scheduling	Hop Filters	Repository Paths	Message
Job finished Channel: #jobs Message: The job finished successfully					
Job stopped due to error abort Channel: Message:					
Job stopped due to manual abort Channel: Message:					

This tab allows you to set the notification messages for the different job statuses. The "Channel" specifies the name of the Slack channel, where the message will be sent to. If no channel is defined, the message is sent to the default channel of the Slack Incoming WebHook.

The "Message" supports Markdown formatting. Refer to the Slack Custom Integrations documentation for more information.

7.2 Rocket.Chat Notifications

The Rocket.Chat notification connection allows you to send job notifications to a Rocket.Chat channel. The connection type uses the Rocket.Chat REST API to post messages.

A Rocket.Chat notification connection has only one special tab on the notification connection editing screen: the "Rocket.Chat REST API" tab. The tab looks like this:

Form titled "Edit a notification connection" with tabs: Name, Type, Throttling, and Rocket.Chat REST API (selected).

Fields:

- Server URL:
- User:
- Password:
- Proxy Host:
- Proxy Port:
- Proxy NTLM Username:
- Proxy NTLM Password:
- Proxy NTLM Authentication Domain:

Buttons: Save, Cancel

Enter the server URL of your Rocket.Chat instance and the user credentials here. The user field takes either the username or the email address of your Rocket.Chat user. Fill out the proxy fields if you need to connect to your Rocket.Chat through an http proxy.

When you configure a job to use a notification connection of the Rocket.Chat notification type, an additional tab "Rocket.Chat Notifications" is presented.

The tab looks like this:

Edit job - MyJob

Name
Connection
Scheduling
Hop Filters
Repository Paths
Rocket.Chat Notifications

Global Rocket.Chat Settings

Default Channel:

Alias:

Emoji:

Avatar:

Job finished

Channel:

Message:

The job finished successfully.

This tab allows you to set the notification messages for the different job statuses. The tab begins with a section for global settings, followed by multiple sections for each notification type. The default channel defines the channel, to which the messages are posted. You can customize the appearance of the posted message with the alias, emoji and avatar fields.

The "Channel" field in the second section allows you to override the default channel for the corresponding notification type. If no channel is defined, the message is sent to the specified default channel.

The "Message" field holds the actual message that will be posted to Rocket.Chat. The field supports Markdown for formatting the message. Refer to the Rocket.Chat documentation for detailed information.