+CITIESDIGITAL

Financial Services – Using Workflow to File Client Documents

Introduction: One of the very first purposes that most organizations identify for Laserfiche Workflow is to have it auto-file their documents. Simply put, if your template and fields provide values that coincide with your folder structure, then Workflow could dynamically move documents to their proper folders. For a lot of organizations, this will be a simple and flawless setup, but for Financial Services firms who are trying to automatically file client documents, it can be a major challenge.

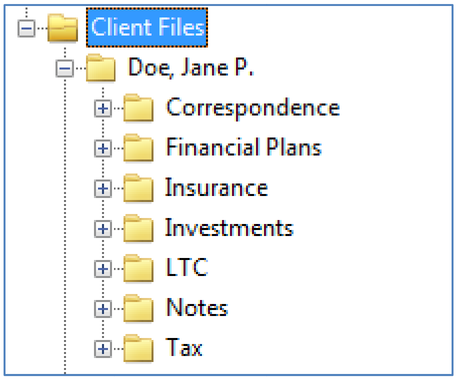
This guide will explain what the issues are and offer a few different solutions for firms who would like to automatically file their client documents.

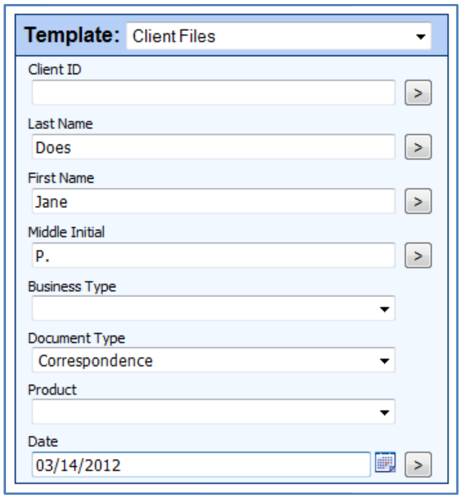
Pre-requisites:

- LF Client and Laserfiche Administration Console (version 8.3 or higher)
- Laserfiche Workflow (version 8.3 or higher)
- Access to the Laserfiche Workflow Designer

Overview:

Let's assume that we have the following folder structure and template for filing our client files. As you'll see below, we do have fields that represent the information that Workflow would need to dynamically route the document to its proper folder location. In this example, one of the first decisions that Workflow is asked to make, is which client folder to navigate to; as you can see the folder naming convention is Last Name, First Name Middle Initial.





Template: Client Files

Client ID:

Last Name: Does

First Name: Jane

Middle Initial: P.

Business Type:

Document Type: Correspondence

Product:

Date: 03/14/2012

Since Workflow would be asked to use the Last Name, First Name, and Middle Initial field to find the corresponding folder in Laserfiche, you would have to make sure that what is typed in the fields matches the exact name of the folder; herein lays the problem.

The problem

Most financial firms will have a client listing that is fairly extensive and in need of constant updating. As previously noted, it would be ideal to provide Workflow with information from drop down lists to ensure the accuracy of an auto file procedure, but with such a long and changing list of clients, it's just not realistic for most financial firms to maintain.

With the inability to use a drop down list for the client's name, there is room for error not only in someone mistyping a name, but in typing "Jim" instead of "James", or vice versa. Keep in mind that Workflow will dynamically route documents to folders, but will also create the folder if it doesn't exist. The slightest misstep in the client's name could end up creating a folder you didn't intend to exist, and a document missing until you realize the error that took place.

The Solutions

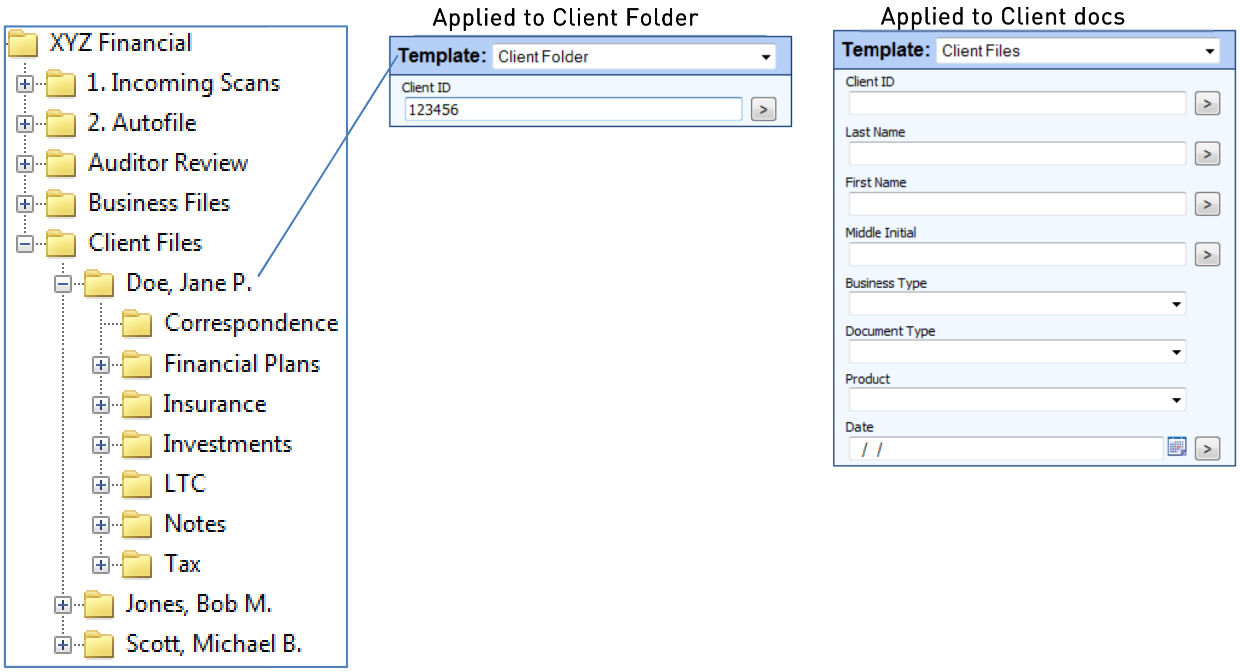
The following will outline a few different solutions that you can pursue to make auto-filing more accurate and reliable.

File by Unique Client ID

One of the most surefire ways to store a client file document to its correct folder is through the use of a unique Client ID. This method means that instead of Workflow just dynamically following/building a folder path based on template fields, you'll search through the repository to find a folder that has a matching ID. To use this method there are a few prerequisites:

1. The template you use for your Client Files must include a field for the unique client ID.
2. The client ID field must be populated for the document prior to the Workflow being initiated.
3. Your root folder for each client must also contain the populated client ID.
4. You must have Workflow 8.3 or higher since it contains the "Search Repository" activity.

To demonstrate what the overall solution could look like, I'll be using the following folder structure and templates …



Applied to Client Folder

Template: Client Folder

Client ID: 123456

Applied to Client docs

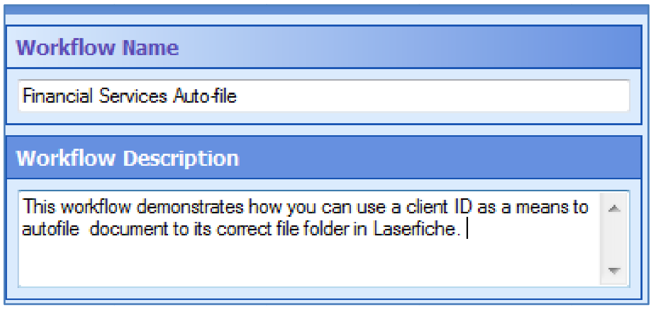
Template: Client Files

Essentially I'll scan my documents into the "1. Incoming Scans" folder, apply my template information, and then move the document to the "2. Autofile" folder, where the workflow will be initiated and file my document. If the Workflow doesn't find a client folder with the matching client ID, or finds more than one match, I'll simply have the document moved to a review folder where a user will need to review why it couldn't automatically be filed.

Configuring Workflow to File by Client ID

The following will detail how to configure Laserfiche Workflow to file documents based upon a unique client ID. It assumes that your root client folders have a template assigned to them that contains a unique client ID, which Workflow will use to match up the client ID contained within the document template.

Step 1: Create a new workflow called "Financial Services Auto-file".



Workflow Name

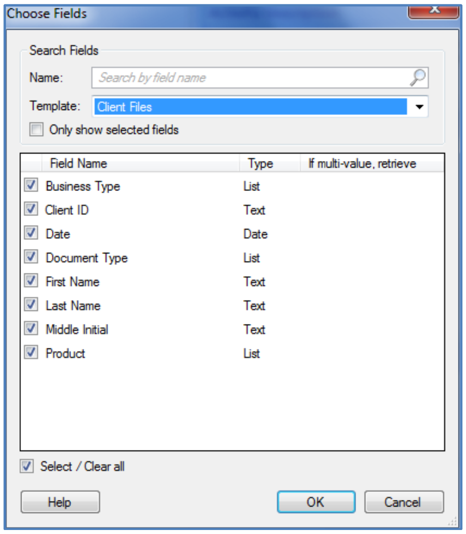
Financial Services Auto-file

Workflow Description

This workflow demonstrates how you can use a client ID as a means to autofile document to its correct file folder in Laserfiche.

Step 2: You will now configure the new workflow and take the following steps…

a. Drag the **Retrieve Field Values** activity into the GUI. You may rename the activity and description as you see fit, but make sure to configure it to retrieve the values of the template that you apply to your client files. In this case, my template is named "Client Files".



Choose Fields

Search Fields

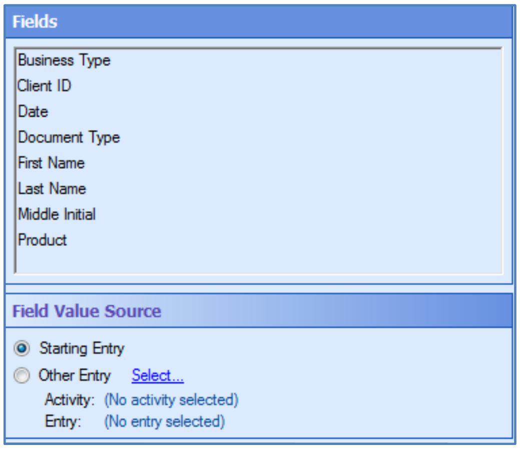
Name: Search by field name

Template: Client Files

□ Only show selected fields

	Field Name	Type	If multi-value, retrieve
☑	Business Type	List	
☑	Client ID	Text	
☑	Date	Date	
☑	Document Type	List	
☑	First Name	Text	
☑	Last Name	Text	
☑	Middle Initial	Text	
☑	Product	List	

☑ Select / Clear all



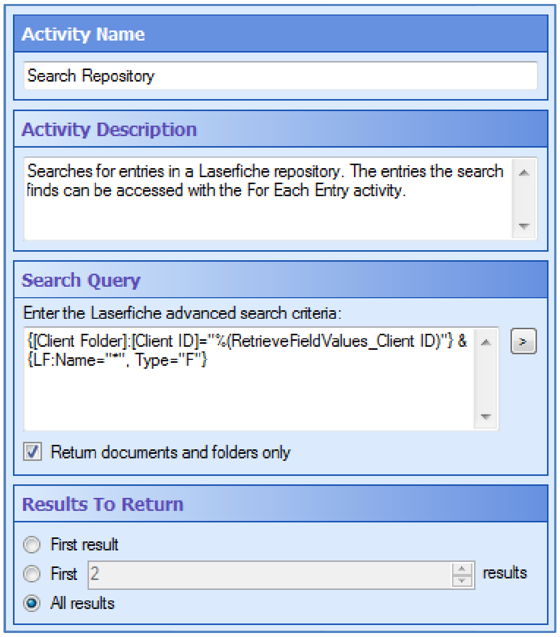
Fields

Business Type
Client ID
Date
Document Type
First Name
Last Name
Middle Initial
Product

Field Value Source

● Starting Entry
○ Other Entry Select...
Activity: (No activity selected)
Entry: (No entry selected)

b. Drag the **Search Repository** activity into the GUI (beneath the Retrieve Field Values). You may rename the activity and description as you see fit. You will also need to populate the "Search Query" with the specific search syntax that Workflow will be using to find the corresponding client folder that contains the matching client ID.



Activity Name

Search Repository

Activity Description

Searches for entries in a Laserfiche repository. The entries the search finds can be accessed with the For Each Entry activity.

Search Query

Enter the Laserfiche advanced search criteria:

```
{[Client Folder]:[Client ID]="%(RetrieveFieldValues_Client ID)"} & {LF:Name="*", Type="F"}
```

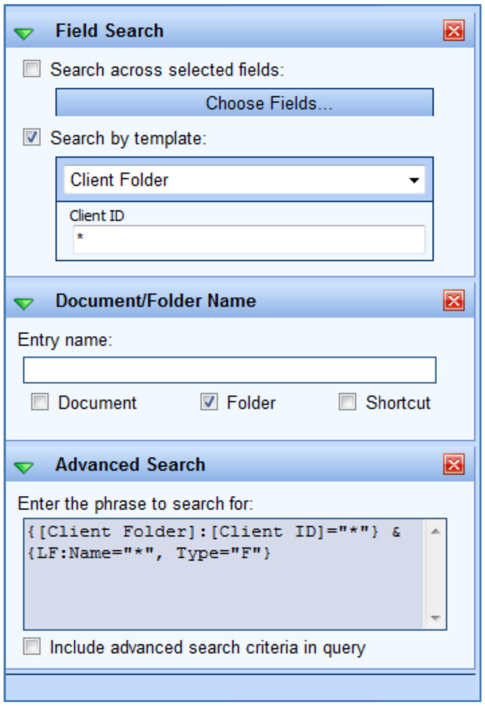
☑ Return documents and folders only

Results To Return

○ First result

○ First 2 results

● All results



Field Search

□ Search across selected fields:

Choose Fields...

☑ Search by template:

Client Folder

Client ID: *

Document/Folder Name

Entry name:

□ Document ☑ Folder □ Shortcut

Advanced Search

Enter the phrase to search for:

```
{[Client Folder]:[Client ID]="*"} & {LF:Name="*", Type="F"}
```

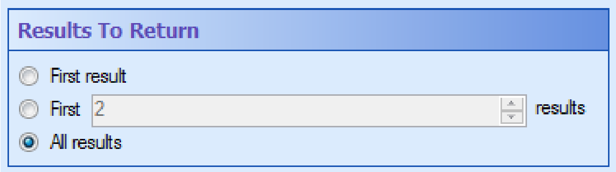
□ Include advanced search criteria in query

How do I generate the Search Query?

- ➢ Simply configure the search, in the Laserfiche Client, that represents the search Workflow will be performing. Also include the "Advanced Search", and then initiate the search.
- ➢ Once the search starts, you'll see that the Advanced Search will now be populated with search syntax. Copy the syntax and place it in the "Search Repository" WF activity.
- ➢ Finally, replace the asterisk in the [Client ID]="*"} portion of your search with the retrieved field value of the Client ID. Shown below is what the final search looked like for the example used in this guide.

  {[Client Folder]:[Client ID]="%(RetrieveFieldValues_Client ID)"} & {LF:Name="*", Type="F"}

c. Drag the **Routing Decision** activity into the GUI (beneath the Search Repository activity). In the previous Search Repository activity, you may have noticed that the last piece of the configuration provides an option for how many results you want returned (shown below).

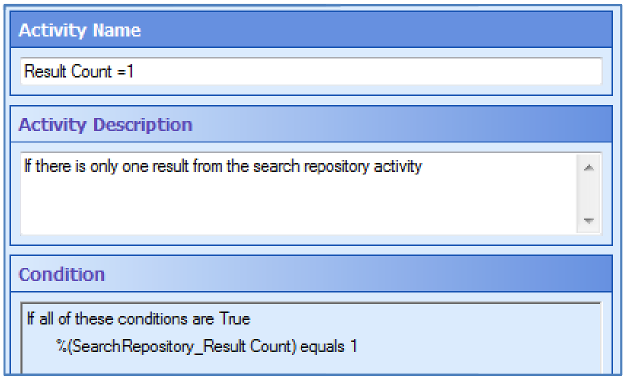


Results To Return

○ First result
○ First 2 results
● All results

As you probably also noticed (and configured), this option should be set to "All results" when using the solution outlined. While this probably seems counterintuitive, in that there should not be more than one result, we'll want to program Workflow in case there is less or more than one. Use the following configurations for the two branches of the routing decision…

Branch 1



Activity Name

Result Count =1

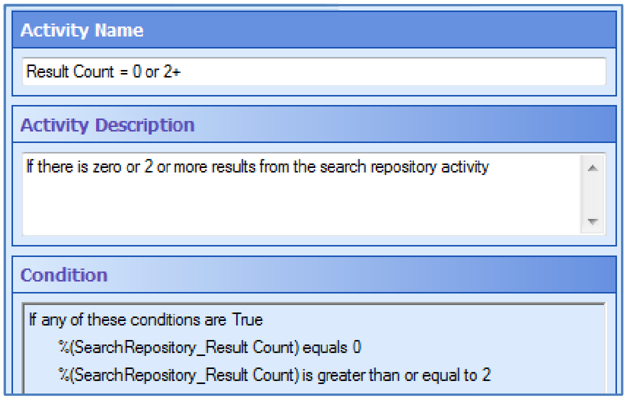
Activity Description

If there is only one result from the search repository activity

Condition

If all of these conditions are True
    %(SearchRepository_Result Count) equals 1

Branch 2



Activity Name

Result Count = 0 or 2+

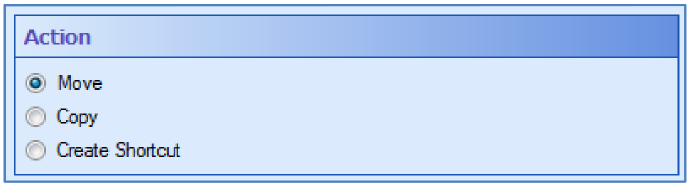
Activity Description

If there is zero or 2 or more results from the search repository activity

Condition

If any of these conditions are True
    %(SearchRepository_Result Count) equals 0
    %(SearchRepository_Result Count) is greater than or equal to 2

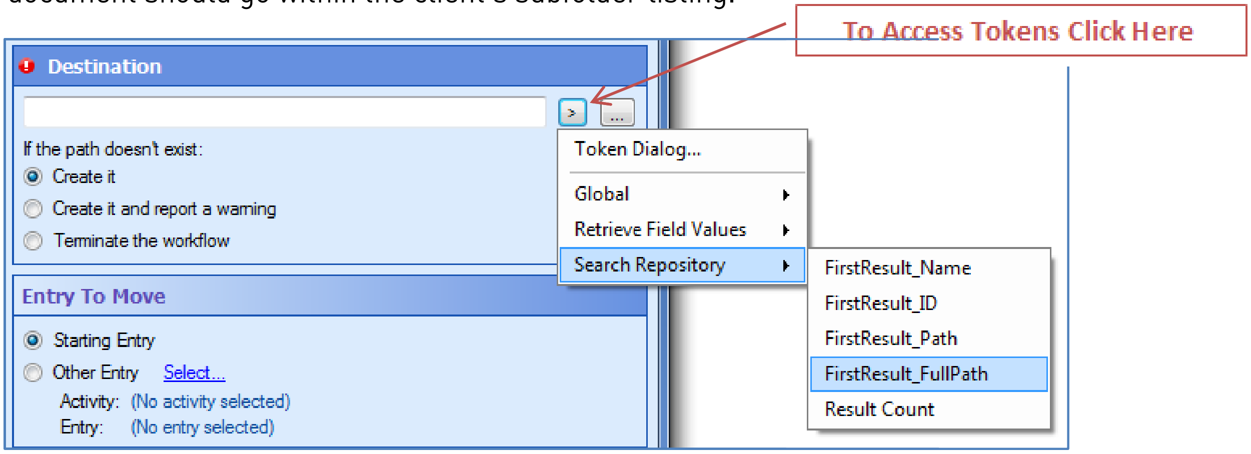
d. Drag the **Move Entry** activity beneath Branch 1 of the Routing Decision. You may rename the activity and description as you see fit, but make sure to use the following configuration for the activity (shown below).



Action

● Move
○ Copy
○ Create Shortcut

To configure the Destination, you'll need to access the tokens (shown below) for both the output result of the Search Activity, as well as the tokens representing where the document should go within the client's subfolder listing.



To Access Tokens Click Here

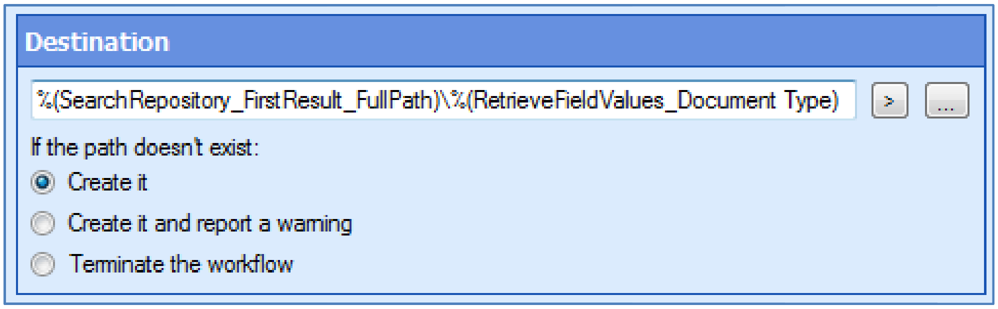
Destination

If the path doesn't exist:
● Create it
○ Create it and report a warning
○ Terminate the workflow

Entry To Move

● Starting Entry
○ Other Entry Select...
Activity: (No activity selected)
Entry: (No entry selected)

Since my example files the documents to a client's subfolder listing based upon whatever the document type is, the destination for my move entry is the following.



Destination

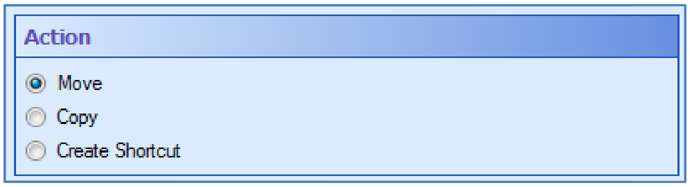
%(SearchRepository_FirstResult_FullPath)\%(RetrieveFieldValues_Document Type)

If the path doesn't exist:
● Create it
○ Create it and report a warning
○ Terminate the workflow

%(SearchRepository_FirstResult_FullPath) represents the client's root folder.

%(RetrieveFieldValues_Document Type) represents the subfolder where the document will be stored in the client's folder.

e. Drag a new **Move Entry** activity beneath Branch 2 of the routing decision. You may rename the activity and description as you see fit, but make sure to use the following configuration for the activity (shown below).

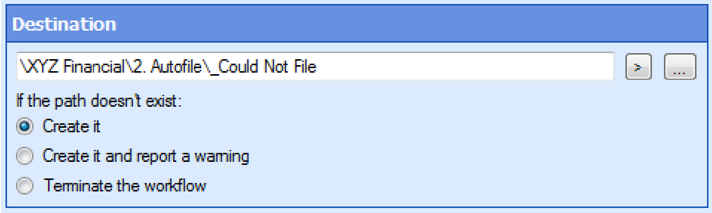


Action

● Move
○ Copy
○ Create Shortcut

Remember, Branch 2 of our routing decision is for any document whose search result yields no entries found, or more than one. In either case, Workflow wouldn't know where to store the document in the client folder structure, so you'll want to set a static destination where documents go so that someone can review and determine why they were not filed properly.

In my example, I've decided to create a subfolder within the 2. Autofile folder called "_Could Not File". With this in place I can easily have someone review the contents of the folder and determine why Workflow could not file the document. It's a great quality control measure and helps to ensure that filing is as accurate as possible.

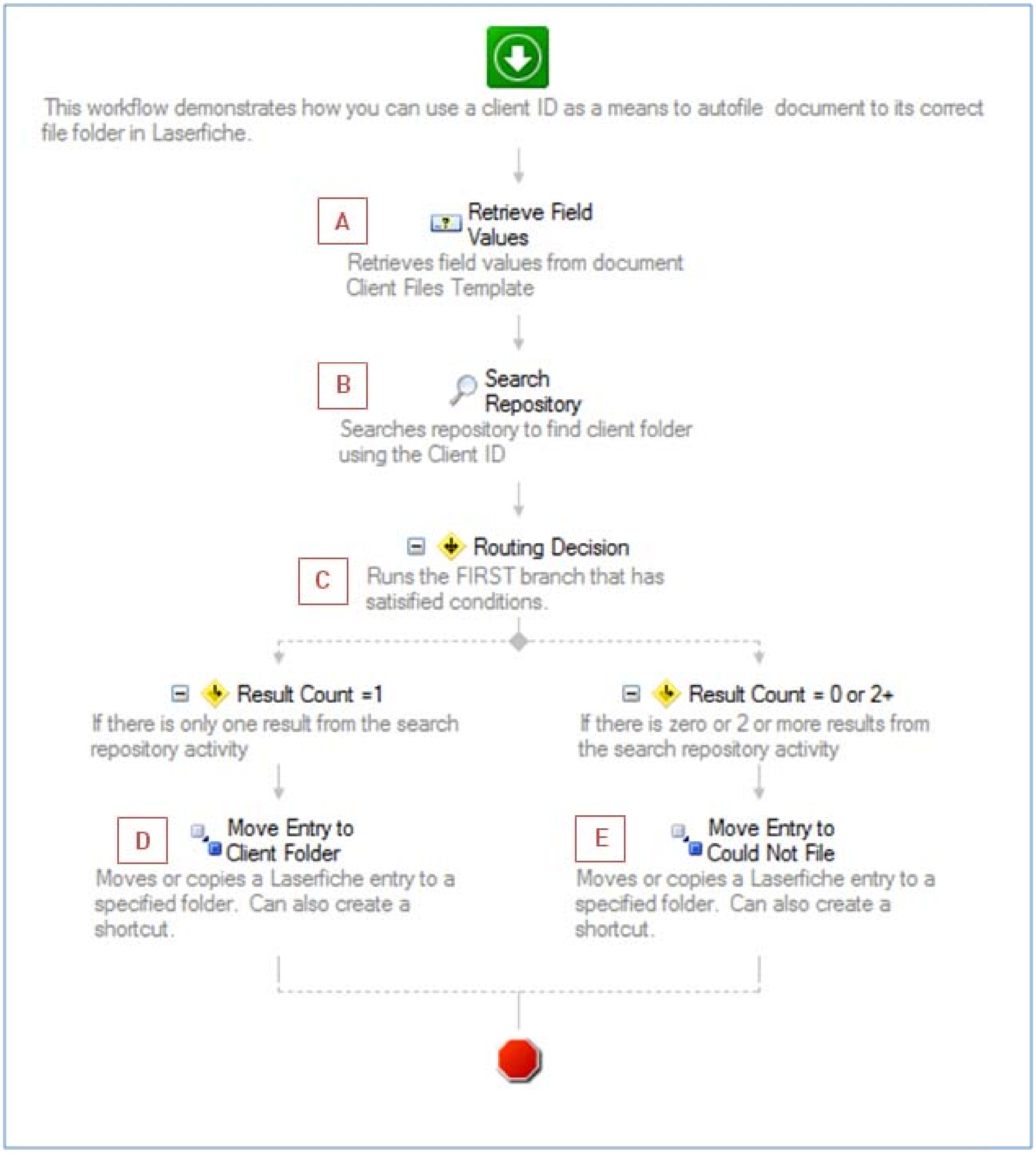


Destination

\XYZ Financial\2. Autofile_Could Not File

If the path doesn't exist:
● Create it
○ Create it and report a warning
○ Terminate the workflow

f. Review the Workflow that you have built; it should look similar to what is shown below.



This workflow demonstrates how you can use a client ID as a means to autofile document to its correct file folder in Laserfiche.

A Retrieve Field Values
Retrieves field values from document Client Files Template

B Search Repository
Searches repository to find client folder using the Client ID

C Routing Decision
Runs the FIRST branch that has satisified conditions.

Result Count =1
If there is only one result from the search repository activity

D Move Entry to Client Folder
Moves or copies a Laserfiche entry to a specified folder. Can also create a shortcut.

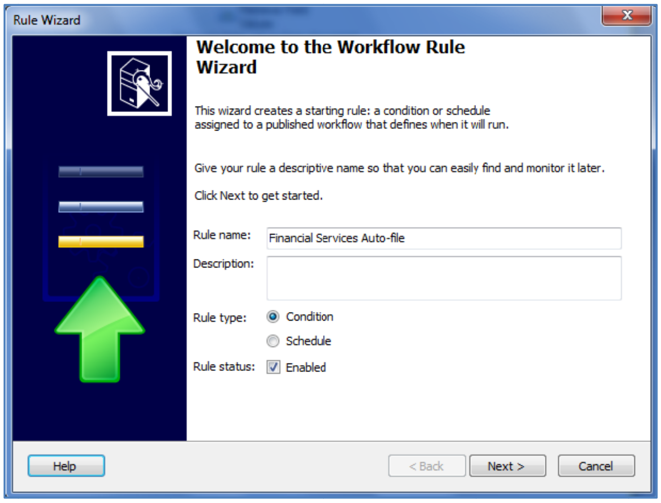
Result Count = 0 or 2+
If there is zero or 2 or more results from the search repository activity

E Move Entry to Could Not File
Moves or copies a Laserfiche entry to a specified folder. Can also create a shortcut.

Please note that the Workflow configuration steps A-G are mapped above for your reference.

g. Publish the Workflow

h. Create the starting rules for the workflow utilizing the Rule Wizard as shown below.



Rule Wizard

Welcome to the Workflow Rule Wizard

This wizard creates a starting rule: a condition or schedule assigned to a published workflow that defines when it will run.

Give your rule a descriptive name so that you can easily find and monitor it later.

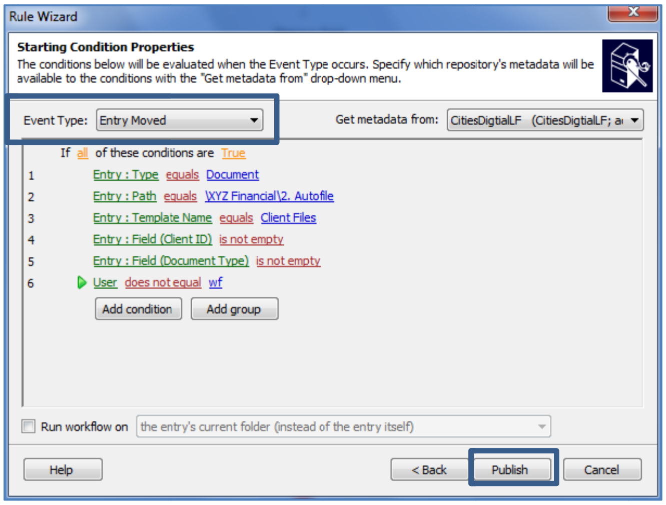
Click Next to get started.

Rule name: Financial Services Auto-file

Description:

Rule type: ● Condition ○ Schedule

Rule status: ☑ Enabled



Rule Wizard

Starting Condition Properties

The conditions below will be evaluated when the Event Type occurs. Specify which repository's metadata will be available to the conditions with the "Get metadata from" drop-down menu.

Event Type: Entry Moved

Get metadata from: CitiesDigtialLF (CitiesDigtialLF; a

If all of these conditions are True

1 Entry : Type equals Document

2 Entry : Path equals \XYZ Financial\2. Autofile

3 Entry : Template Name equals Client Files

4 Entry : Field (Client ID) is not empty

5 Entry : Field (Document Type) is not empty

6 User does not equal wf

☐ Run workflow on the entry's current folder (instead of the entry itself)

Please Note: The starting conditions noted are the bare minimum that Cities Digital would recommend utilizing, and you will need to make sure that the conditions are specific to your Laserfiche configuration. Also, make sure that the "User does not equal…" is set to the appropriate workflow user name that is used in your specific connection profile.

Once you have published the workflow, you're ready to test the configuration.

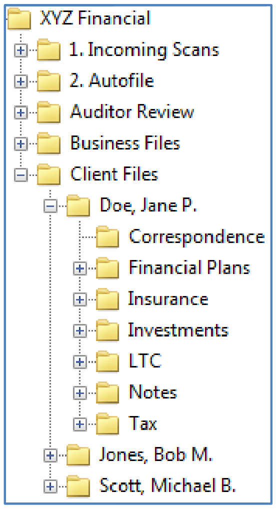
File by Dynamic Folder Path w/ Validation

If you'll strictly be using template information to dynamically build the folder path for where a document should be stored, you'll want to have some sort of validation to ensure that the field information used by Workflow is in fact accurate. Again, this ties back to the problem of having to use a free form field for things like Client Last Name, First Name or Account number. If you can't ensure that critical pieces of information are input in the exact same way every time, and someone types "Jim" instead of "James", you'll have rogue folders created and lost documents.

To use this method there are a few prerequisites:

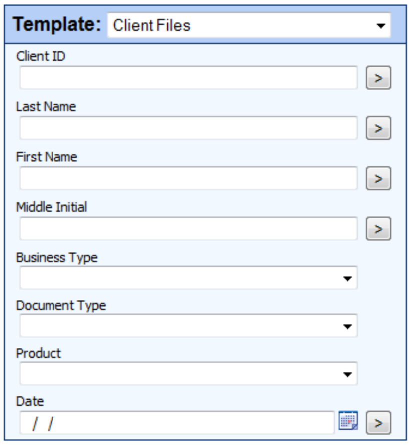
1. The template you use for your Client Files must include field information that could dynamically generate the name of your client folder.
2. Documents must be fully indexed prior to initiating the Workflow.
3. You must have Workflow 8.3 or higher due to feature changes in the routing decision tokens.

To demonstrate what the overall solution could look like, I'll be using the following folder structures and template…



- XYZ Financial
  - 1. Incoming Scans
  - 2. Autofile
  - Auditor Review
  - Business Files
  - Client Files
    - Doe, Jane P.
      - Correspondence
      - Financial Plans
      - Insurance
      - Investments
      - LTC
      - Notes
      - Tax
    - Jones, Bob M.
    - Scott, Michael B.

Applied to Client docs



Template: Client Files

Client ID

Last Name

First Name

Middle Initial

Business Type

Document Type

Product

Date / /

Essentially I'll scan my documents into the "1. Incoming Scans" folder, apply my template information, as well as the "New Client" tag if applicable. Once I'm finished indexing, I will move the document to the "2. Autofile" folder and Workflow will initiate and file the document if the folder path is validated. If it is not a valid path, and the "New Client" tag is not set, it will move the document to a review folder.

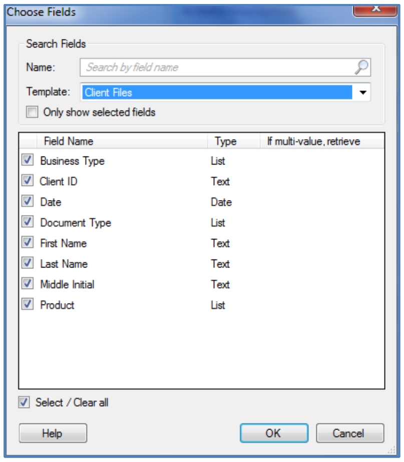
Configuring Workflow to File by Dynamic Folder Path w/ Validation

Step 1: Create a tag called "New Client" in the Administration Console. Users will apply this tag to a document / file any time that they know there isn't already a Laserfiche client file folder structure in place.

Step 2: Create a new workflow called "Financial Services – Autofile + Validation" (or whatever you prefer).

Step 3: You will now configure the new workflow and take the following steps…

a. Drag the **Retrieve Field Values** activity into the GUI. You may rename the activity and description as you see fit, but make sure to configure it to retrieve the values of the template that you apply to your client files. In this case, my template is named "Client Files".



Choose Fields

Search Fields

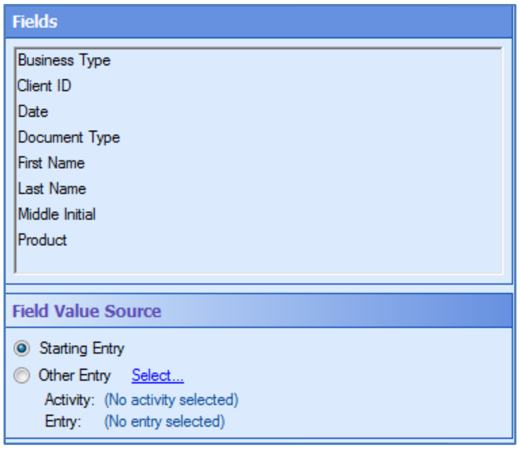
Name: Search by field name

Template: Client Files

□ Only show selected fields

	Field Name	Type	If multi-value, retrieve
☑	Business Type	List	
☑	Client ID	Text	
☑	Date	Date	
☑	Document Type	List	
☑	First Name	Text	
☑	Last Name	Text	
☑	Middle Initial	Text	
☑	Product	List	

☑ Select / Clear all



Fields

Business Type
Client ID
Date
Document Type
First Name
Last Name
Middle Initial
Product

Field Value Source

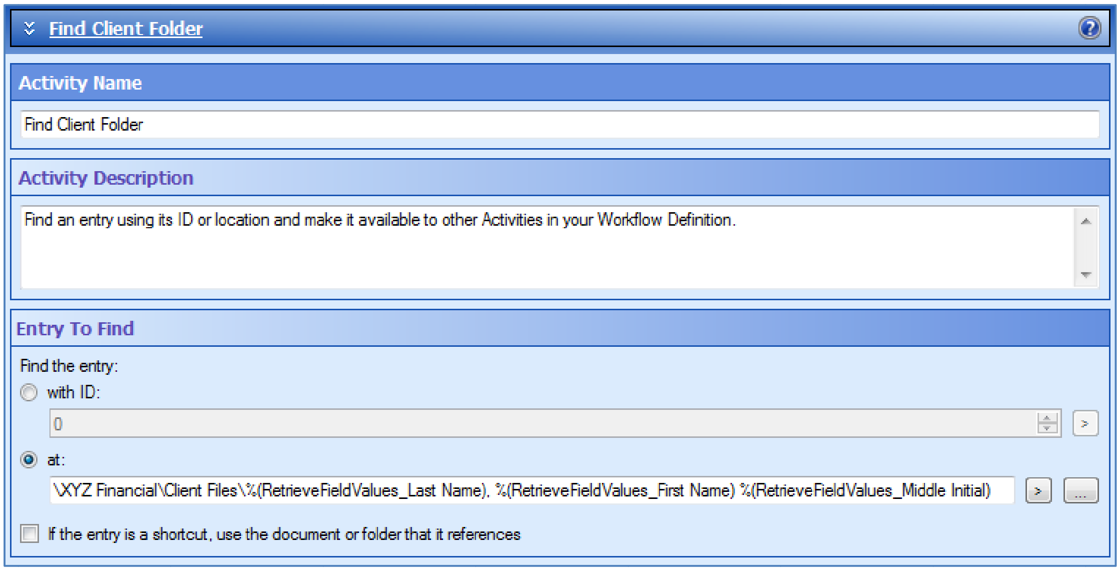
● Starting Entry

○ Other Entry Select...

Activity: (No activity selected)

Entry: (No entry selected)

b. Drag the **Find Entry** activity beneath the Retrieve Filed Values activity. Essentially this activity will use the retrieved field values of the previous activity to determine if the root client folder already exists. Because my naming convention for the client root folder is **Last Name, First Name Middle Initial**, I will configure the activity as follows:



Find Client Folder

Activity Name

Find Client Folder

Activity Description

Find an entry using its ID or location and make it available to other Activities in your Workflow Definition.

Entry To Find

Find the entry:

○ with ID:

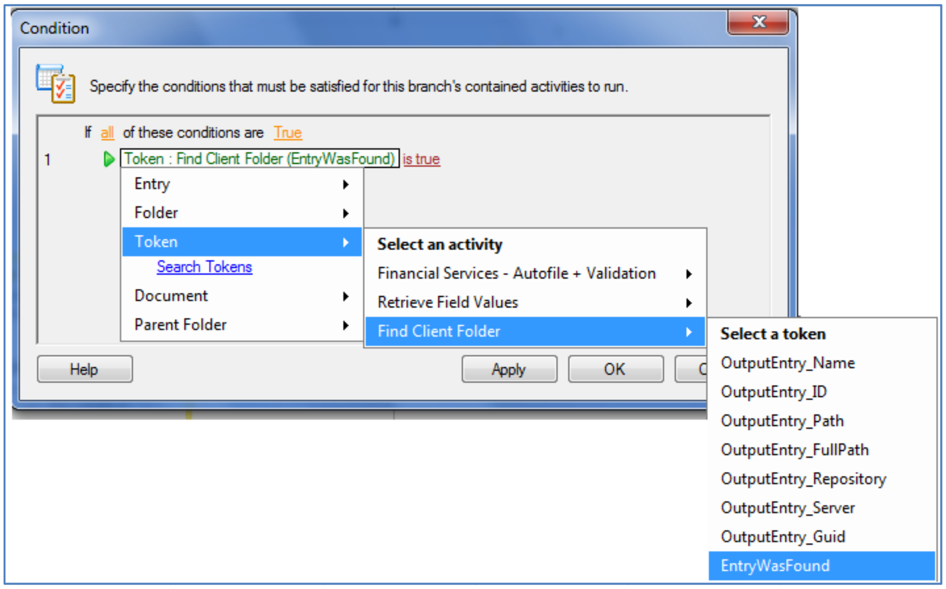
0

● at:

\XYZ Financial\Client Files\%(RetrieveFieldValues_Last Name), %(RetrieveFieldValues_First Name) %(RetrieveFieldValues_Middle Initial)

□ If the entry is a shortcut, use the document or folder that it references

c. Drag the **Routing Decision** activity beneath the Find Entry activity. Name the activity "Valid Client File Folder Path?" and provide a customized description. This routing decision will be used in combination the previous Find Entry to make a logical statement about whether the folder path that was searched for was found. The condition will be based upon the Token for the Fine Client Folder activity (as shown below)

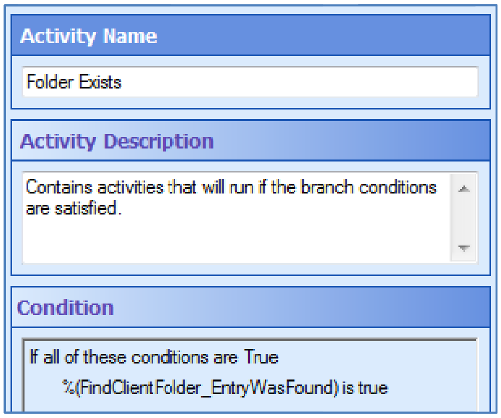


Specify the conditions that must be satisfied for this branch's contained activities to run.

If all of these conditions are True

1 Token : Find Client Folder (EntryWasFound) is true

Branch 1 – Folder Exists



Activity Name

Folder Exists

Activity Description

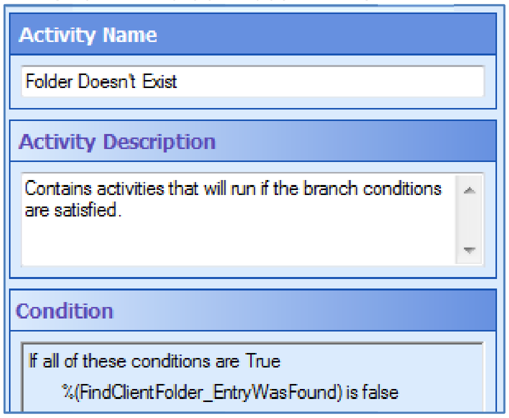
Contains activities that will run if the branch conditions are satisfied.

Condition

If all of these conditions are True

%(FindClientFolder_EntryWasFound) is true

Branch 2 – Folder Doesn't Exist



Activity Name

Folder Doesn't Exist

Activity Description

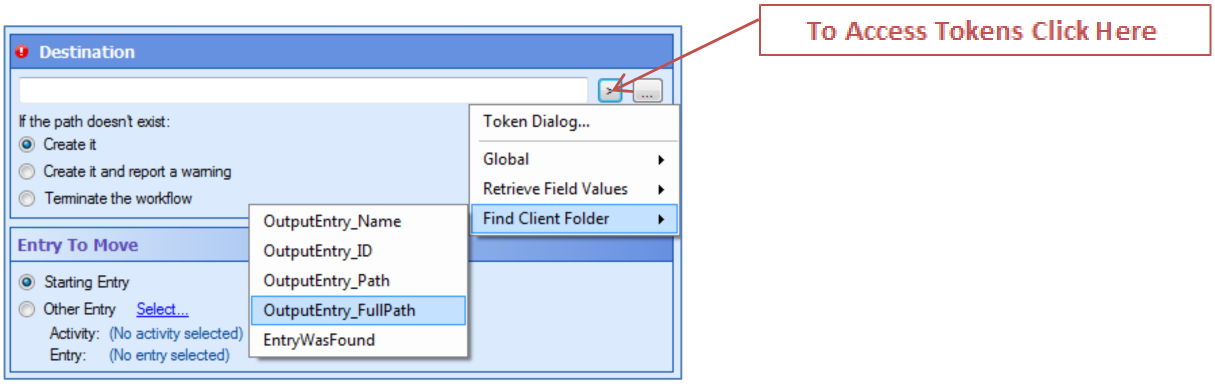
Contains activities that will run if the branch conditions are satisfied.

Condition

If all of these conditions are True

%(FindClientFolder_EntryWasFound) is false

d. Drag the **Move Entry** activity beneath Branch 1 of the previously configured routing decision. Branch 1 should represent that the client's root file folder exists, which means workflow should move the document to its final storage location. In my example, documents are stored to the client's root folder and then to a subfolder… the subfolder names are the same as the values found in my drop down list for document type, so my configuration would use the following tokens for the destination.



To Access Tokens Click Here

Destination

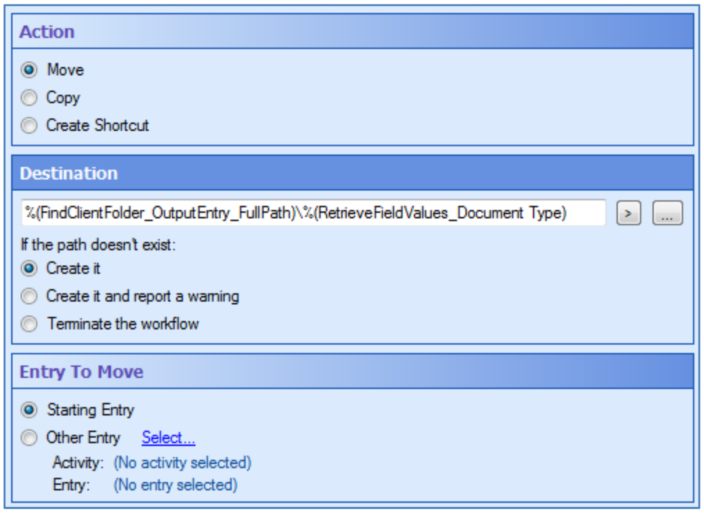
If the path doesn't exist:

● Create it
○ Create it and report a warning
○ Terminate the workflow

Entry To Move

● Starting Entry
○ Other Entry Select...
Activity: (No activity selected)
Entry: (No entry selected)

Notice that the first token I use is the "OutputEntry_FullPath" from the Fine Client Folder activity; since it was valid, I can just use that token to get me to the client folder, and then select the document type token from the Retrieve Field Values activity. In the end, my full configuration for the move entry will be…



Action

● Move
○ Copy
○ Create Shortcut

Destination

%(FindClientFolder_OutputEntry_FullPath)\%(RetrieveFieldValues_Document Type)

If the path doesn't exist:

● Create it
○ Create it and report a warning
○ Terminate the workflow

Entry To Move

● Starting Entry
○ Other Entry Select...
Activity: (No activity selected)
Entry: (No entry selected)

Now that we have configured Branch 1 of the routing decision, we'll turn our attention to Branch 2 of the Workflow, and what to do when the folder path is not valid.

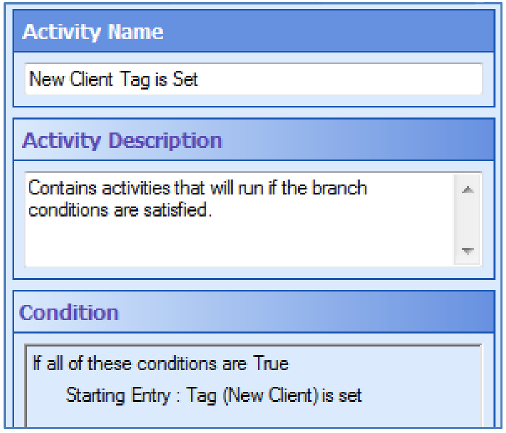
e. Drag a new **Routing Decision** activity beneath Branch 2 (Folder Doesn't Exist) of the previous routing decision activity. Remember, Branch 2 of the previous decision is used to essentially say that the client's folder doesn't exist. The reason that the folder doesn't exist could be one of two options…

Option 1: The folder path doesn't exist because the document being filed is for a brand new client / account.

Option 2: The folder path doesn't exist because the template field information for the client was typed incorrectly.

The end user should know whether Option 1 is in fact the case for what they're filing, in which case they will apply the "New Client" tag to the document. Therefore this routing decision activity will indicate whether the "New Client" tag is set or not.

Branch 1 – New Client Tag is Set



Activity Name

New Client Tag is Set

Activity Description

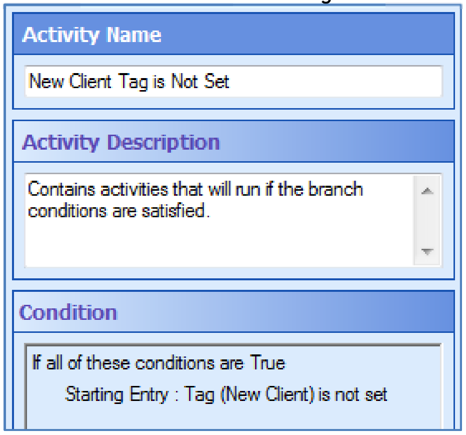
Contains activities that will run if the branch conditions are satisfied.

Condition

If all of these conditions are True

Starting Entry : Tag (New Client) is set

Branch 2 – New Client Tag is Not Set



Activity Name

New Client Tag is Not Set

Activity Description

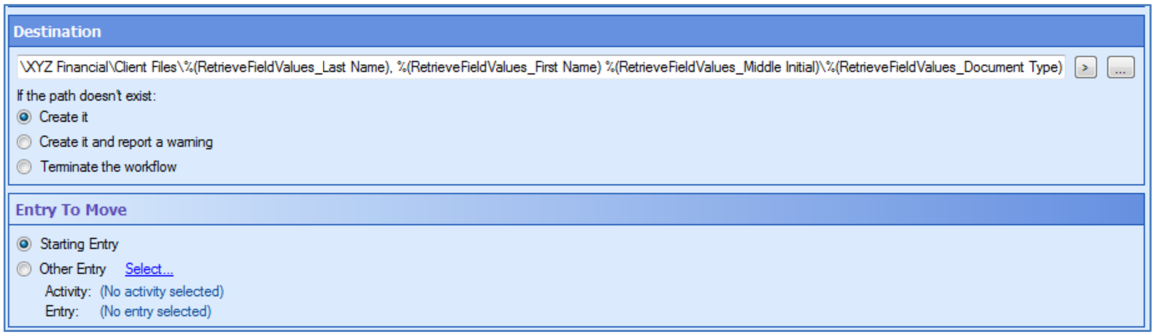
Contains activities that will run if the branch conditions are satisfied.

Condition

If all of these conditions are True

Starting Entry : Tag (New Client) is not set

f. Drag a new **Move Entry** activity beneath Branch 1 (New Client Tag is Set) and configure it to store the document to its final storage location. In my example, I'll be using the retrieved field values from my very first activity to help build the path that doesn't currently exist.



Destination

\XYZ Financial\Client Files\%(RetrieveFieldValues_Last Name), %(RetrieveFieldValues_First Name) %(RetrieveFieldValues_Middle Initial)\%(RetrieveFieldValues_Document Type)

If the path doesn't exist:

- ● Create it
- ○ Create it and report a warning
- ○ Terminate the workflow

Entry To Move

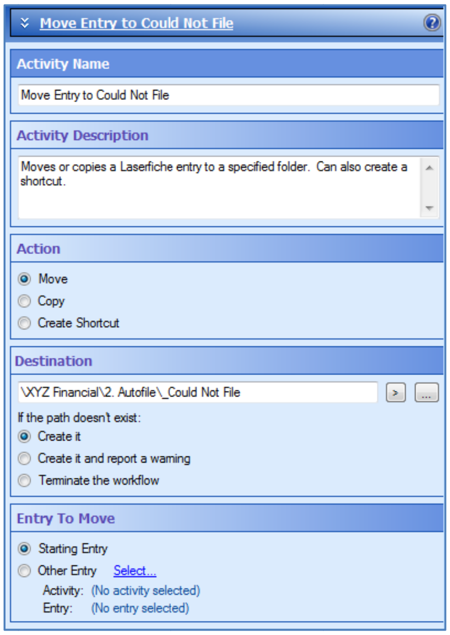
- ● Starting Entry
- ○ Other Entry Select...

Activity: (No activity selected)

Entry: (No entry selected)

g. Drag a new **Move Entry** activity beneath Branch 2 (New Client Tag is Not Set) and configure it to store the document to a review folder so that a user may review and determine why Workflow could not file the document.

In my example, I determined that Workflow should move the document to a folder called "_Could Not File", contained within my "2. Autofile folder". From here a user could review the document, make corrections, and then resubmit to initiate the Workflow again.



Move Entry to Could Not File

Activity Name

Move Entry to Could Not File

Activity Description

Moves or copies a Laserfiche entry to a specified folder. Can also create a shortcut.

Action

- ● Move
- ○ Copy
- ○ Create Shortcut

Destination

\XYZ Financial\2. Autofile_Could Not File

If the path doesn't exist:

- ● Create it
- ○ Create it and report a warning
- ○ Terminate the workflow

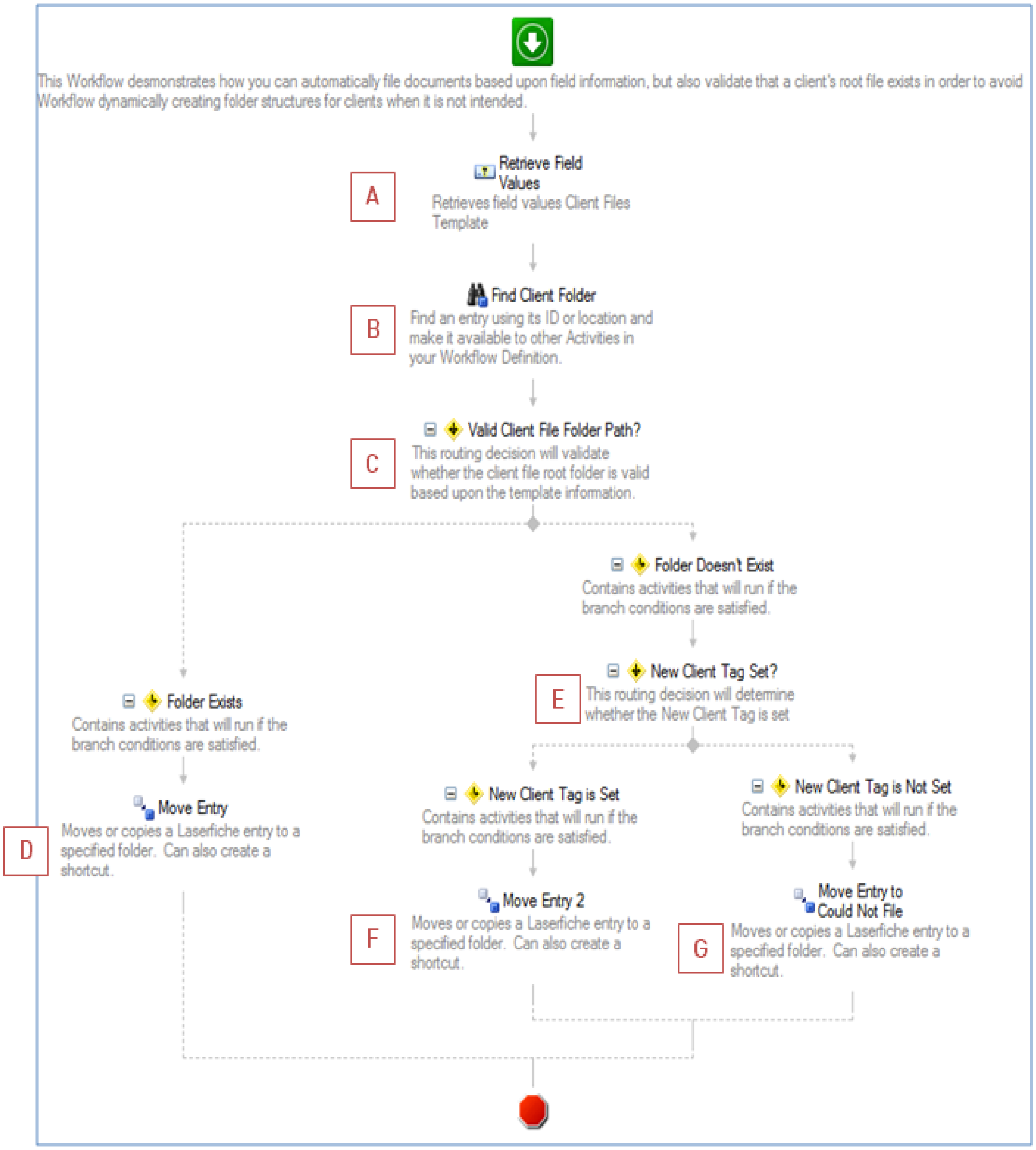
Entry To Move

- ● Starting Entry
- ○ Other Entry Select...

Activity: (No activity selected)

Entry: (No entry selected)

h. Review the Workflow that you have built; it should look similar to what is shown below.



This Workflow desmonstrates how you can automatically file documents based upon field information, but also validate that a client's root file exists in order to avoid Workflow dynamically creating folder structures for clients when it is not intended.

Retrieve Field Values
Retrieves field values Client Files Template

Find Client Folder
Find an entry using its ID or location and make it available to other Activities in your Workflow Definition.

Valid Client File Folder Path?
This routing decision will validate whether the client file root folder is valid based upon the template information.

Folder Doesn't Exist
Contains activities that will run if the branch conditions are satisfied.

New Client Tag Set?
This routing decision will determine whether the New Client Tag is set

Folder Exists
Contains activities that will run if the branch conditions are satisfied.

Move Entry
Moves or copies a Laserfiche entry to a specified folder. Can also create a shortcut.

New Client Tag is Set
Contains activities that will run if the branch conditions are satisfied.

New Client Tag is Not Set
Contains activities that will run if the branch conditions are satisfied.

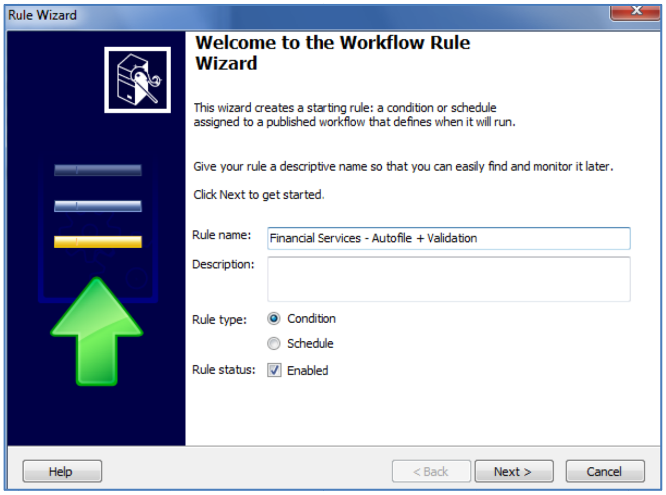
Move Entry 2
Moves or copies a Laserfiche entry to a specified folder. Can also create a shortcut.

Move Entry to Could Not File
Moves or copies a Laserfiche entry to a specified folder. Can also create a shortcut.

Please note that the Workflow configuration steps A-G are mapped above for your reference.

i. Publish the Workflow

j. Create the starting rules for the workflow utilizing the Rule Wizard as shown below.



Rule Wizard

Welcome to the Workflow Rule Wizard

This wizard creates a starting rule: a condition or schedule assigned to a published workflow that defines when it will run.

Give your rule a descriptive name so that you can easily find and monitor it later.

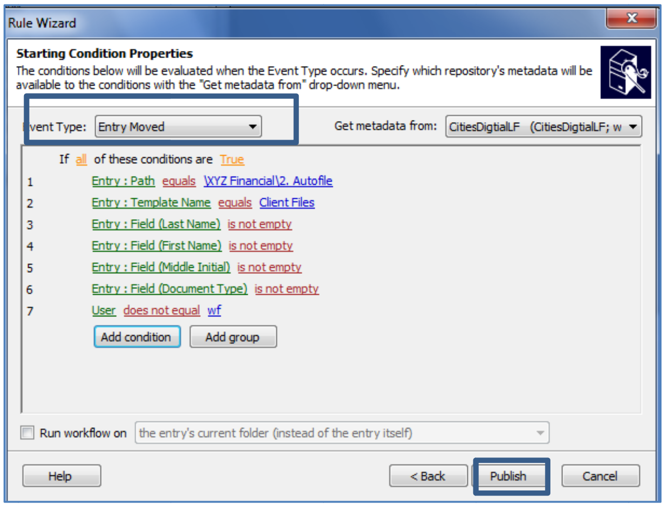
Click Next to get started.

Rule name: Financial Services - Autofile + Validation

Description:

Rule type: ● Condition ○ Schedule

Rule status: ☑ Enabled



Rule Wizard

Starting Condition Properties

The conditions below will be evaluated when the Event Type occurs. Specify which repository's metadata will be available to the conditions with the "Get metadata from" drop-down menu.

Event Type: Entry Moved

Get metadata from: CitiesDigtialLF (CitiesDigtialLF; w

If all of these conditions are True

1 Entry : Path equals \XYZ Financial\2. Autofile

2 Entry : Template Name equals Client Files

3 Entry : Field (Last Name) is not empty

4 Entry : Field (First Name) is not empty

5 Entry : Field (Middle Initial) is not empty

6 Entry : Field (Document Type) is not empty

7 User does not equal wf

☐ Run workflow on the entry's current folder (instead of the entry itself)

Please Note: The starting conditions noted are the bare minimum that Cities Digital would recommend utilizing, and you will need to make sure that the conditions are specific to your Laserfiche configuration. Also, make sure that the "User does not equal…" is set to the appropriate workflow user name that is used in your specific connection profile.

Once you have published the workflow, you're ready to test the configuration.