REPORT MINER USER GUIDE

Version 1.2

Table of Contents

Overview

PFD documents exist in the trillions and support all types of personal and business activities. A large percentage of these documents were “born digital”; meaning that they were created from electronic files such as Microsoft Word or Excel and were converted to PDF. The other portion are simply scanned images that are stored inside a PDF “container”.

While the original premise of PDF format was to provide a common standard for storing and sharing final form documents, many knowledge workers and organizations have come to rely on the ability to easily and automatically take data stored within these documents and use it for different purposes such as locating specific data within a PDF; or exporting data in a table to a spreadsheet.

With all of the benefits of PDF, such as, providing compact, sharable, secure, and perfectly displayed documents, there are a lot of difficulties when attempting to use the data within a PDF document. Data stored within a PDF does not have the necessary metadata to identify individual words let alone complex data structures such as fields and tables.

Report Miner enables even novice users to quickly take data from PDF files and use it within other applications. Through a simple-to-use interface, knowledge workers can define data elements that are then used to automatically locate and extract this data to text files, XML, or spreadsheet formats.

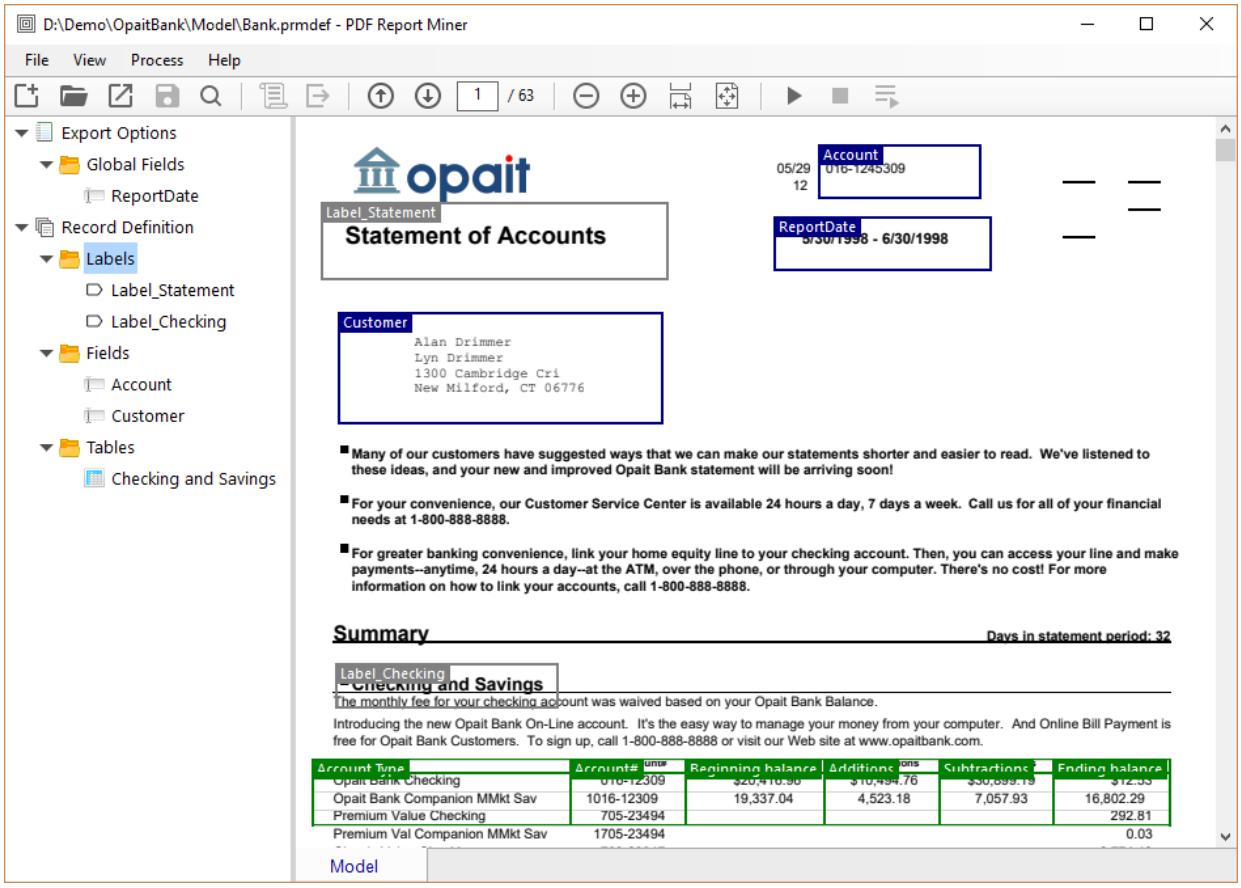
Although primarily defined for parsing and extracting data from PDF documents, the Report Miner is equally capable of processing scanned images, such as TIFF files, or text documents including print streams and EBCDIC files with vertical carriage controls. These extensions to the Report Miner makes it ideal for processing high volumes of images as well as the traditional Computer Output to Laser Disc (COLD) data files.

PDF Report Miner uses a custom PDF parsing and viewing engine with sophisticated text handling and automatic reading order detection, a feature that is lacking in most traditional PDF viewers or libraries. Reliable and quality text extraction is vital to successful parsing of more complicated data structures such as fields and tables.

PDF Report Miner is designed for flexibility and scalability. The available SDK for the .NET environment allows customization and precise control of the processing engine through plugins and a callback interface. Unlike many traditional PDF engines, the report miner can process very large PDF documents with many thousands of pages.

General Concepts

Report Miner is designed to allow anyone to quickly create simple rules to automatically parse PDF documents and extract data.



To perform this task, a PDF file is used to define and “draw” data elements. There are four primary data elements within Report Miner:

1. Global Fields - Global Fields allow the user to define data that can exist anywhere within a PDF document which can be used for processing such as splitting a PDF into multiple documents.

2. Labels - Labels are used as “anchor points” in order to locate other needed data. For instance, the label “Checking” can be used to locate the corresponding table data that includes transactional data.

3. Fields - Fields are the actual variable data elements that are extracted from the PDF file along with the field label (when defined).

4. Tables - Tables are data elements that are structured as rows and columns. Tables can have fixed or variable columns and rows and can extend across pages of a PDF document.

Creating a Model

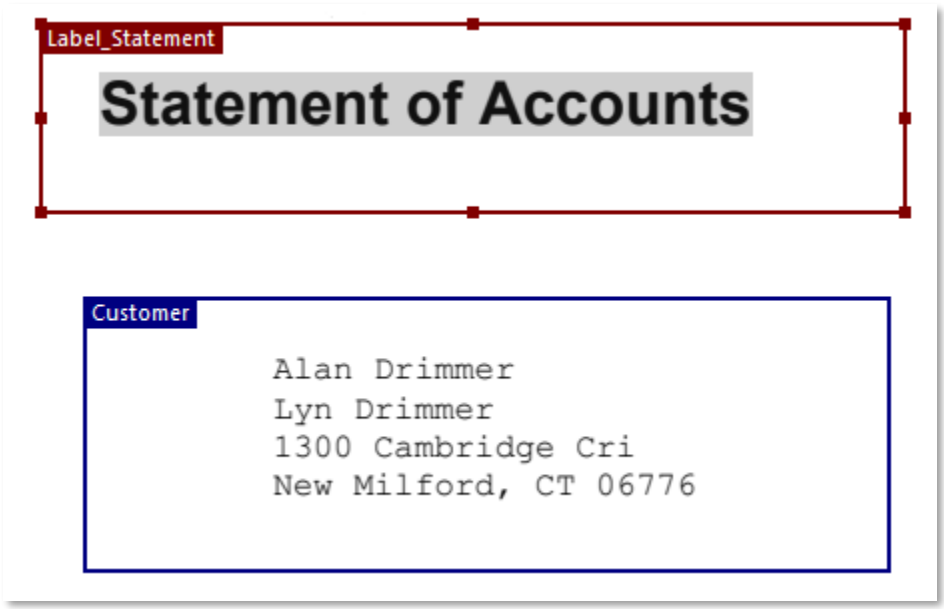
Overview

A model is stored in a project file and contains all of the processing and output rules for a particular target document. In order to create a model, an example PDF document must be imported into the Report Miner application. You may use the “New Model” option under the File menu, the corresponding button on the toolbar, or simply drag and drop the file from the Explorer. Once a model has been defined, it can be used to process many PDF files with similar structure. The processing engine can adopt even when the subsequent files are not exactly the same as the sample PDF file provided they roughly follow a similar structure. For instance, a single model can often be defined to locate and extract financial da

ta from multiple reports (e.g. quarterly 10-Q and annual 10-K) issued by a public company.

Defining a Label

Labels are used to locate data via a text search or through use of pattern matching. For instance, in the following figure, the label called “Statement” uses a text query for the words “Statement of Accounts” to locate the Customer field directly below it.

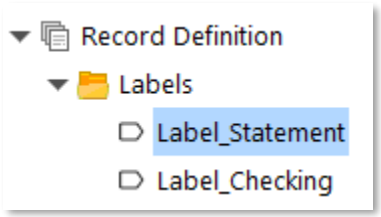


In this example, the label is acting as an anchor for the data field. The system will first locate the label and then find the field relative to its actual location.

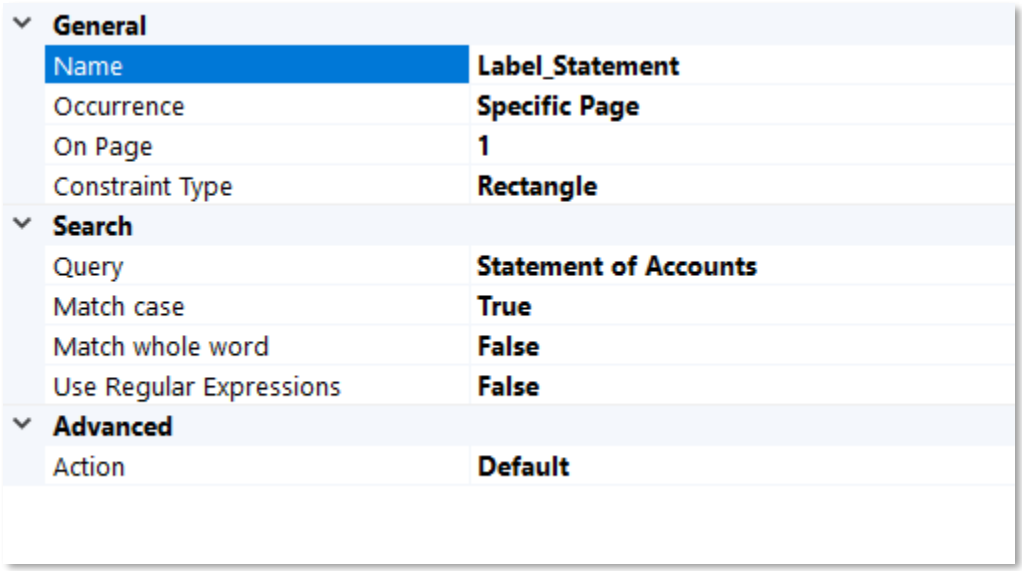
Labels can also be used as identifiers to trigger certain other actions. For instance, a label can be used to find pages from the report document that should be skipped (ignored) during processing.

Also, when the PDF document contains many sub-documents (e.g. an aggregated report of client statements), a label can be used to identify the boundaries of each sub-document. For example, if pages in a statement are numbered as “Page 1 of 7, Page 2 of 7, etc.”, a label can be defined to look for the string “Page 1 of” and trigger start of a new sub-document (record) when it is located on a page.

A label can be associated with more than one field.



In order to create a label, select the Labels node and then draw a boundary around the data from which you wish to create a label. Once created, the Label Property options will be displayed in the lower-left panel of the application.



Depending upon certain options, additional properties may or may not be displayed.

General Label Properties:

Name	The name of the label. Although not required, it is recommended to use unique and meaningful names to identify labels.
Occurrence	Determines where the label is expected. Options are listed below.
On Page	This field is only visible if the Occurrence is “Specific Page”. The value is the page number of the logical record, counting from 1.
Label	This field is only visible if the Occurrence is “Relative to Label”. The value is another label that identifies the page where this label can be detected.
Constraint Type	This property constrains the area for the label occurrence. The options are listed below.

Occurrence Options:

Any Page	The label can be detected on any page of the record.
Specific Page	The label can only occur on a specific record page. If selected, you must also specify on which page number, counting from 1, the label can be detected. For reports with multiple records, the page number is relative to the beginning of a specific record and not the original report file.
Relative to Label	The label can be detected with respect to another label. If selected, you must also specify the linked label. Both labels must be detected simultaneously on a page. This feature to define labels relative to other labels is useful to more precisely pin extracted data and avoid mis-hits. The linking can be nested to any number of levels. The system will detect and handle mutual or circular dependencies.

Constraint Type Options:

Rectangle	The label is constrained to reside entirely within the bounding box.
Horizontal	The Label is constrained between the left and right borders of the bounding box.
Vertical	The label is constrained between the top and bottom borders of the bounding box.
None	No constraints; the label can be detected anywhere on the page.

Search Label Properties:

Query	The search string used to locate a label
Match Case	If True, the query will only match exact case.
Match Whole Word	If True, the query will only match whole words.
Use Regular Expression	If True, the query will be interpreted as a regular expression pattern to match text.

Advanced Label Properties:

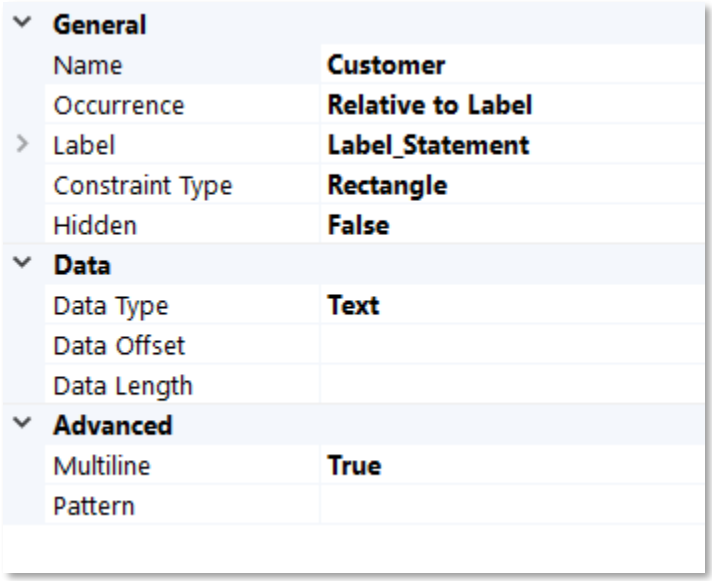
Action	Specifies the additional action to be performed when the label is detected. Options are below.

Action Options:

Default	The label is used as an anchor point with no further actions.
Skip Matching Text	The text within the bounding rectangle of the detected label is ignored for detecting fields and tables. This can be used to mask out certain portions of page during table detection. The masked-out section could include page numbers or page headers or footers if they interfere with detection logic.
Skip Entire Page	If the label is detected, the entire report page is skipped.

Defining a Field

Fields are the actual data values that you wish to locate and extract. Fields can have specific locations, defined relative to a label or be placed anywhere within a PDF document. To define a field, select the Fields node and then draw a box on the example report which contains the field value. Once a field is created, properties are displayed in the lower left pane.



General Field Properties:

Name	The name of the field. Although not required, it is recommended to use unique and meaningful names to identify fields.
Occurrence	Determines where the field is expected. The options are the same as for a label.
Constraint Type	This property constrains the area for the field occurrence. The options are the same as for a label.
Hidden	If this flag is true, the value of the field will be detected but not reported. Hidden field values can be used in macros to construct computed or composite values. See computed data types and macros later in this document.

Field Data Properties:

Data Type	Defines the data type of the field value in order to further restrict the type of the expected data. The list of supported data types is described below.
Data Offset	Allows you to define the beginning of the data you wish to extract within a located field value. For instance, if an account number always begins with “000” and this data is not needed you can set an offset to “3” which will extract the value after skipping the first 3 characters.
Data Length	Allows you to define the expected length of the field value. Field values longer than Data Length will be truncated. However, shorter values will not be padded.

Supported Data Types:

Text	All printable characters are allowed.
String	Allows letters, hyphens, periods and apostrophes.
Alphanumeric	Allows letters, digits, hyphens and periods.
Numeric	Allows digits, hyphens and periods.
Date	Represents a date or a combination of a date and a time.
Amount	Represents a decimal amount in a valid currency format.
Custom	A user defined type that has a regular expression mask that defines the data layout and permitted characters.
Computed	The value for a field that has this type is not detected but computed using literals, numbers and macros. The values of other fields can be combined to created a more complex computed value. See macros later in this document.

Advanced Field Properties:

Multiline	If true, the field value will span multiple text lines. If false, only the first line will be returned.
Pattern	Allows you to define a regular expression pattern on the detected field value.
Replace	This option is only shown if a regular expression pattern is specified. It defines a regular expression replace string. You may use captured groups to replace the field value.
Match Case	Only appears if a regular expression pattern is specified. If True, it forces the regular expression matching to be case sensitive.

Pattern Matching Example: Consider the “ReportDate” field in the example report. The value of the field is 5/30/1998 – 6/30/1998. However, we are only interested in the upper range of the date range, that is, 6/30/1998. To accurately extract this portion, we define the following Pattern:

.* - (.*)

In English, this means to match any number of character until we hit the hyphen (“- “). Then start capturing the patterns until the end of the field value. The first captured group in the regular expression matching is labeled as $1. Therefore, we only need to set the value of the Replace field to $1. Match Case is not relevant for this example.

Defining a Table

Tables can be located anywhere within a document or only on specific pages. Report Miner allows you to locate and extract this data regardless of where it is located and includes sophisticated algorithms to automatically detect location of tables and discover table structure including number of columns, rows, table headers and tables that span multiple pages.

To define a table, select the Tables node and then draw a box to include the current table header and rows. You may also select only a few rows and let Report Miner detect the extent of the table. Report Miner will also detect the table characteristics, including data types, when possible. Once a table is defined, the following properties are displayed in the lower left panel.

⌄	**General**	
	Name	**Checking and Savings**
	Occurrence	**Relative to Label**
›	Label	**Label_Checking**
⌄	**Structure**	
	Header Lines	**1**
	Table Columns	**(Collection)**
›	Primary Column	**Account Type**
	Number of Columns	**Fixed**
	Keep Empty Columns	**False**
⌄	**Advanced**	
	Multipage	**False**
	Multiline Rows	**False**
	Table Beginning	**Automatic**
	Table Ending	**Automatic**

General Table Properties:

Name	The name of the table. Although not required, it is recommended to use unique and meaningful names to identify tables.
Occurrence	Determines where the table is expected. The options are the same as for labels and fields.

Table Structure Properties:

Header Lines	This is the number of lines/rows from the top of the table used to define column headers. Headers that span multiple lines are concatenated for each column using a single space as a separator. Header lines are often automatically detected when defining a table. They can also be adjusted manually when automatic detection is ambiguous or incorrect.
Table Columns	Table columns are typically automatically identified by the Report Miner. Clicking on this property will allow you to view and edit the column properties. See column properties below.
Primary Column	Normally one column of the table is designated as the primary column. This column must have valid entry for each row. There are exceptions to this rule when cell values can span multiple rows and the primary column has a specific data type other than text. This is to allow for section headers and sub-headers. The Primary Column is also used in automatic detection of table ending.
Number of Columns	Allows you to define the expected number of columns. Options are “Fixed” and “Automatic”.
Keep Empty Columns	If this flag is set to True, then columns that don’t have any data are deleted from the detected table. If set to False, then empty columns are retained.

Advanced Table Properties:

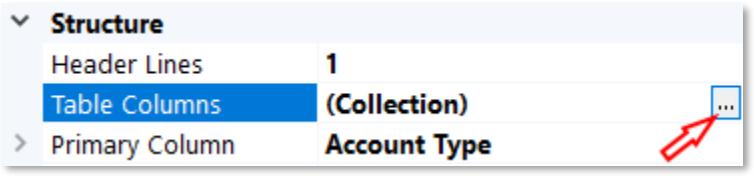
Multipage	If True, the table can potentially span multiple report pages. Segments of the table may or may not carry addition headers on subsequent pages. If False, the Report Miner will only look for the entire table on a single page.
Multiline Rows	If True, some table cells might have values that span multiple lines. Multi-line rows are more complex to detect and process. Set this value to False if all cell values are single lines.
Table Beginning	Determines the method used to identify the beginning (first row) of the table. Table detection options are described below.
Table Ending	Determines the method used to identify the end of the table. Table detection options are described below.

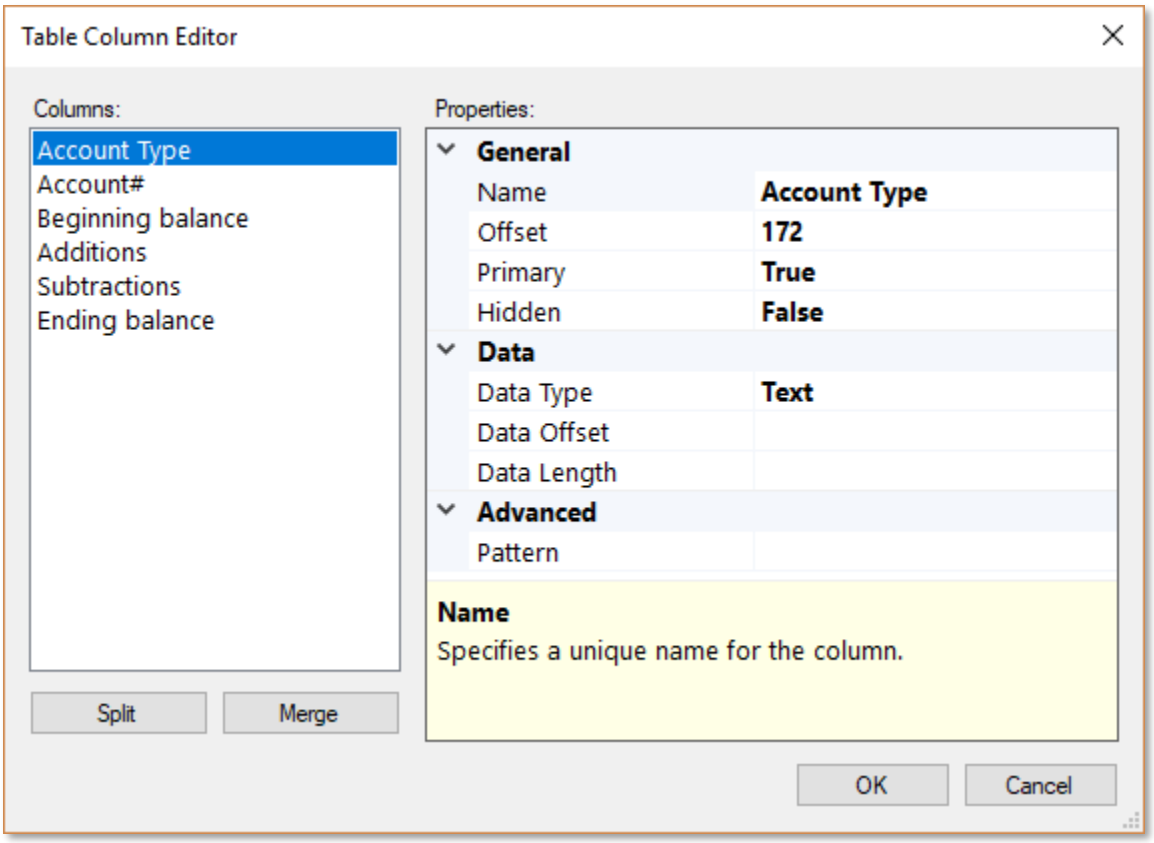
Table Detection Options:

Fixed	The beginning and/or ending of the table is fixed as in the samples report.
At Label	The table boundary is determined when a label is detected. For the beginning of the table the line with the label is included, while for the table end the line is excluded from the table.
After Label	The table boundary is determined as the row that follows the detection of a label. For the beginning of the table the line with the label is excluded, while for the table end the line is included in the table.
Automatic	The beginning and/or ending of the table is automatically detected.

Table Column Editor:

Click on the button that appears on the “Table Columns” property to display the Table Column Editor dialog.





In the Table Column Editor, selecting each column will allow you to specify properties or edit those that are auto-detected.

General Column Properties:

Name	The name or header of the column, unless automatically detected.
Offset	Specifies the initial offset from left edge of the table in PDF logical units.
Primary	If True, then the column must contain data for every cell. Exceptions apply if multiline rows are allowed.
Hidden	If True, then the column will be detected but not extracted.

Column Data Properties:

Data Type	Defines the data type of the column cell value in order to further restrict the type of the expected data. The list of supported data types is the same as for fields.
Data Offset	Allows you to define the beginning of the data you wish to extract within a detected cell value. Similar to a field value.
Data Length	Allows you to define the expected length of the field value. Field values longer than Data Length will be truncated. However, shorter values will not be padded.

Advanced Column Properties:

Pattern	Allows you to define a regular expression pattern on the detected cell value.
Replace	This option is only shown if a regular expression pattern is specified. It defines a regular expression replace string. You may use captured groups to replace the call value. Uses the same algorithm as for the field values.
Match Case	Only appears if a regular expression pattern is specified. If True, it forces the regular expression matching to be case sensitive.

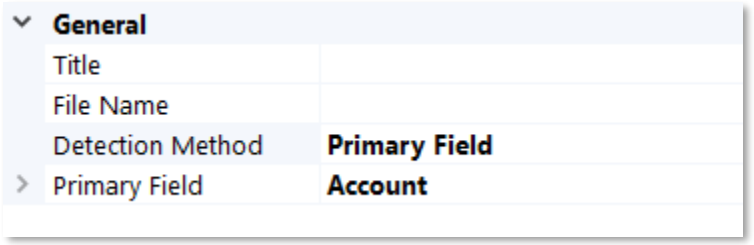
Split and Merge Buttons:

You may create new columns or merge adjacent columns into one using the Split and Merge buttons. The columns are normally auto-detected. However, if the table layout does not allow a clear separation of cell values into columns of data, or if you deliberately wish to combine columns, then these buttons would allow you to do so.

The column and header boundaries can also be adjusted using the mouse and after the Column properties dialog is closed.

Record Definition Options

Report Miner has the ability to take a single multi-page PDF document and split it into individual records. To set the required parameters, start by selecting the Record Definition node in the left panel which will display the below setting options.



The following are the available parameters:

Title – An optional title string constructed for each record. You may use a combination of detected fields and certain macros to construct a title for the record.

File Name - The file name for saving each individual PDF file. Macros are supported.

Detection Method - This property specifies the manner in which each new record is detected. There are four options:

1. **Single Record -** This option will treat the PDF document as a single record and not split or produce individual reports.

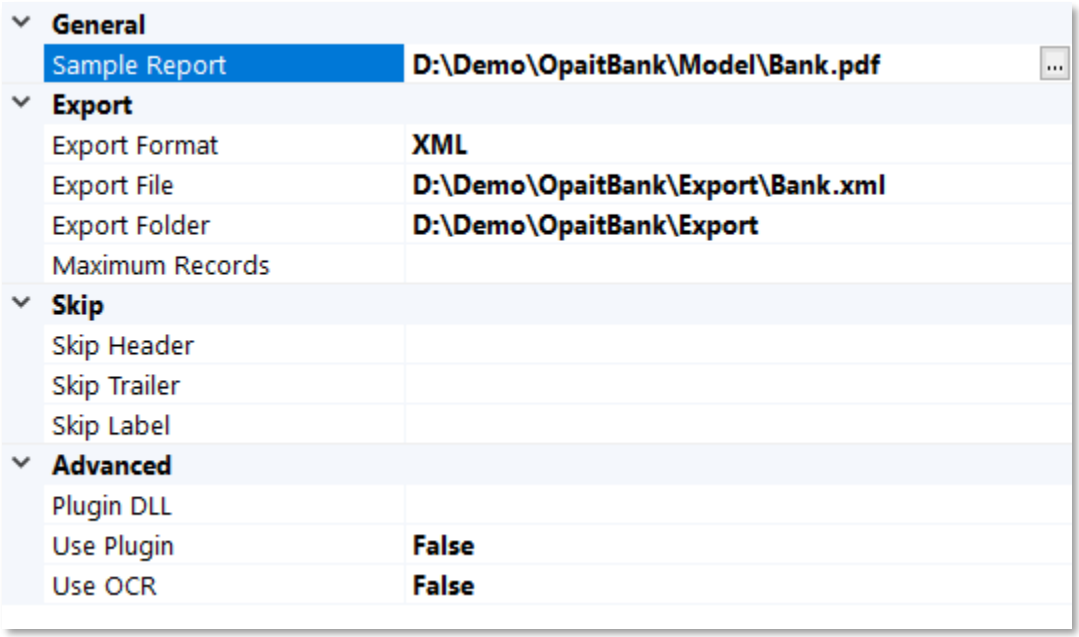
2. **Relative to Label -** This option allows you to designate a new record each time a designated label is located. When selecting this option, you are then asked to select the label. For example, the search string for the label could be “Page 1 of”.

3. **Primary Field -** This option allows you to designate a new record each time the detected value of a field designated as Primary is located. When selecting this option, you are then asked to select the primary field. For example, the primary could be tracking an account number and creating a new record when the account number changes.

4. **Page Count -** This option allows you to designate individual records based upon page count. When this option is selected, you are then required to provide a numeric value which will be used to create individual records when the provided page count is reached. Each record will have a fixed number of pages. The last record might end up with less number of pages.

Export Options

In order to test or run a model, you may define some basic export options. Export Options is accessed by selecting the “Export” node in the left pane. When selected, options are presented in the lower left-hand portion of the application.



General Properties:

Sample Report	Path to the sample report file used in model definition. The sample report file will be saved with the model definition. It will be automatically loaded when the model is opened for editing or testing.

Export Properties:

Export Format	The output of data extraction can be saved in XML, CSV, Pipe delimited or Text formats.
Export File	If specified, the extracted data will be written to this file in the selected Export Format.
Export Format	If specified, a sub-document will be constructed for each detected record in PDF format. These files will be saved to the Export Folder and linked to the corresponding extracted data in the Export File.
Maximum Records	You may limit the number of records extracted during test runs by specifying a positive Maximum Records parameter.

Skip Properties:

Skip Header	If your PDF documents have pages at the beginning that are not relevant, you can skip them by entering in a numeric value for the number of pages from the beginning to skip.
Skip Trailer	Similar to Skip Header, if your PDF documents have pages at the end that are not relevant, you can skip them by entering in a numeric value for the number of pages from the end, to skip.
Skip Label	You can also create labels which are used to determine which pages are non-essential and can be skipped. The value for this property is a selected label.

Skip Properties:

Plugin DLL	Report Miner API defines a simple callback interface for constructing custom plugins which can be loaded during model execution. The exported methods of the plugin will be called by the processing engine during the processing of the report. These methods allow more precise control and customization of the processing workflow. Please refer to the Report Miner API documentation for more details.
Use Plugin	This is a binary flag that can switch the loading of the plugin on and off for testing and model development. For instance, if you are extracting a specific cell from a table using the plugin, you will receive a single value if the plugin is running. By temporarily turning off the plugin, you will be able to examine the whole table to make sure that the plugin is extracting the right cell from the table.
Use OCR	PDF Report Miner seamlessly integrates with Tesseract 4.0 OCR engine to allow processing of image only elements that might be present inside the PDF report file. The Report Miner assumes that the Tesseract engine is installed under the default Program Files directory.

Viewing Output

In order to test output from your model, simply select the “Start Processing” button. The focus of the application will change to the “Output” tab that is selectable near the bottom of the application.

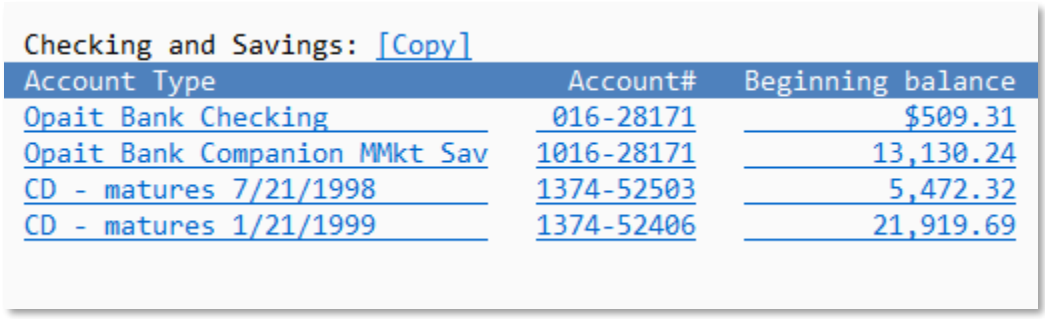
General Output

If a report is model is structured correctly, the following data is presented for all output.

- ❑ Processing time.
- ❑ Report Model Path
- ❑ Report File Path.
- ❑ Plugin DLL Path.
- ❑ Export File Path.
- ❑ Export Folder Path.

Model-Specific Output

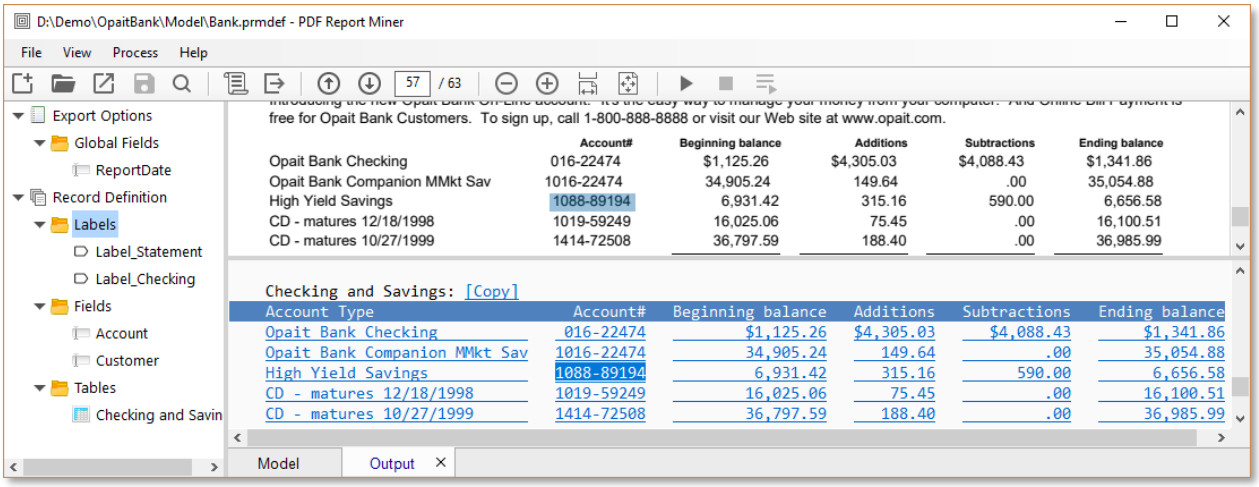
Your output will be displayed depending upon the actual data you have elected to locate and extract. Each field or table value is presented in blue and is hyperlinked to the location in the source report file that is processed.



Checking and Savings: [Copy]

Account Type	Account#	Beginning balance
Opait Bank Checking	016-28171	$509.31
Opait Bank Companion MMkt Sav	1016-28171	13,130.24
CD - matures 7/21/1998	1374-52503	5,472.32
CD - matures 1/21/1999	1374-52406	21,919.69

Clicking on any of the links will display the source file and highlight the data in a panel directly above the output display. You can use keyboard array keys to browse through the extracted data. The highlighting in the sample report will automatically be synchronized with the currently selected field or table cell.



Batch Processing

In a typical application of the PDF Report Miner, a model is first developed using a sample report. Data elements of interest, line single valued fields and tables, are defined on the sample report. The definition is saved in a data model project file. The data model is used to extract data from report files that have layouts that are similar to the sample report.

The batch processing feature of the PDF Report Miner allows you to test the model against any number of such reports as you develop the model. You may open any number of report files and test the model against them individually or in one batch operation.

1. Use the “Open Report” command or the corresponding button to show the open file dialog.

2. Choose one or more report files that you wish to process.

3. The report miner will create a new tab for each open report file.

4. To test a specific report file, first click on its tab which will display the report. Then use the Start Process command to run the model using the selected report file.

5. To test all report files, click on the Process All Reports command.

Macros

Using macros is an advanced method of further customizing the output from extraction. Macros can be used to dynamically route records, create custom titles and name files during runtime. The following macros are currently defined.

$Number	The ordinal number of the current record, counting from 1.
$Title	The title of the current record.
$Date	The current date in default format (yyyy-mm-dd)
$Date(format)	The current datetime formatted according to .NET 'format'.
$(name)	The value of a field with matching 'name'.
$(name[range])	A substring of the value of a matching field.
$$	An embedded single '$' and not start of a macro.

[range] is an optional parameter which can be used to further define extraction of field values:

[offset]	Extract text value started at given character offset.
[offset,length]	Defines both offset and length of a sub-string to extract.
[Full]	Include full value (instead of first line) of field value.

Notes:

1. A macro matches the first global or record field with the given name.
2. Use unique field names to avoid conflicts.
3. Field name comparison is case sensitive.
4. Invalid [range] parameters will not cause exceptions.

Example:

$(Account[2,6]) will skip the first 2 characters from the value of the field named Account and extract the next 6 characters. Therefore, if the value of the Account field is 123456789, the given macro will extract 345678.

Export Manager

Overview

PDF Export Manager is a companion product to PDF Report Minor. It provides a framework for managing multiple report models created in the Report Miner and automate the process of detection and processing of PDF files as they become available.

The Export Manager has a fully multi-threaded kernel acting as a high-performance document processing server. Export Manager works with a list of pre-defined report models and a set of watch folders. As PDF documents are placed into these folders, or arrive automatically via file transfer, the Export Manager pairs them up with the appropriate model and manages tasks like file locking, events handling and communicating with user or API processes.

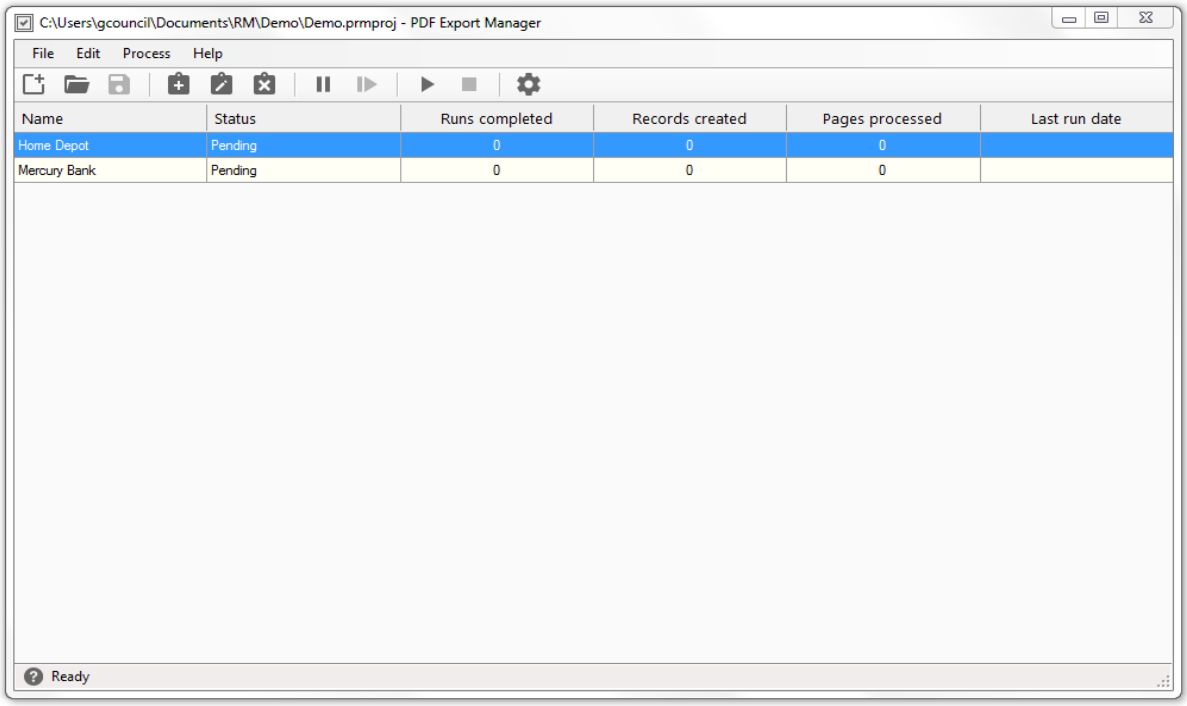
In time-critical or very high-volume environments, multiple instances of the Export Manager may be deployed to increase performance.

The Export Manager also works with the Report Miner API to handle more complex PDF documents or data extraction requirements, as are common in RPA (Robotic Process Automation) applications.

This guide assumes that you are somewhat familiar with the PDF Report Miner documentation.

General Concepts

PDF Export Manager operates as a background process that is configured using a simple-to-use application.



There are several functions which the application provides.

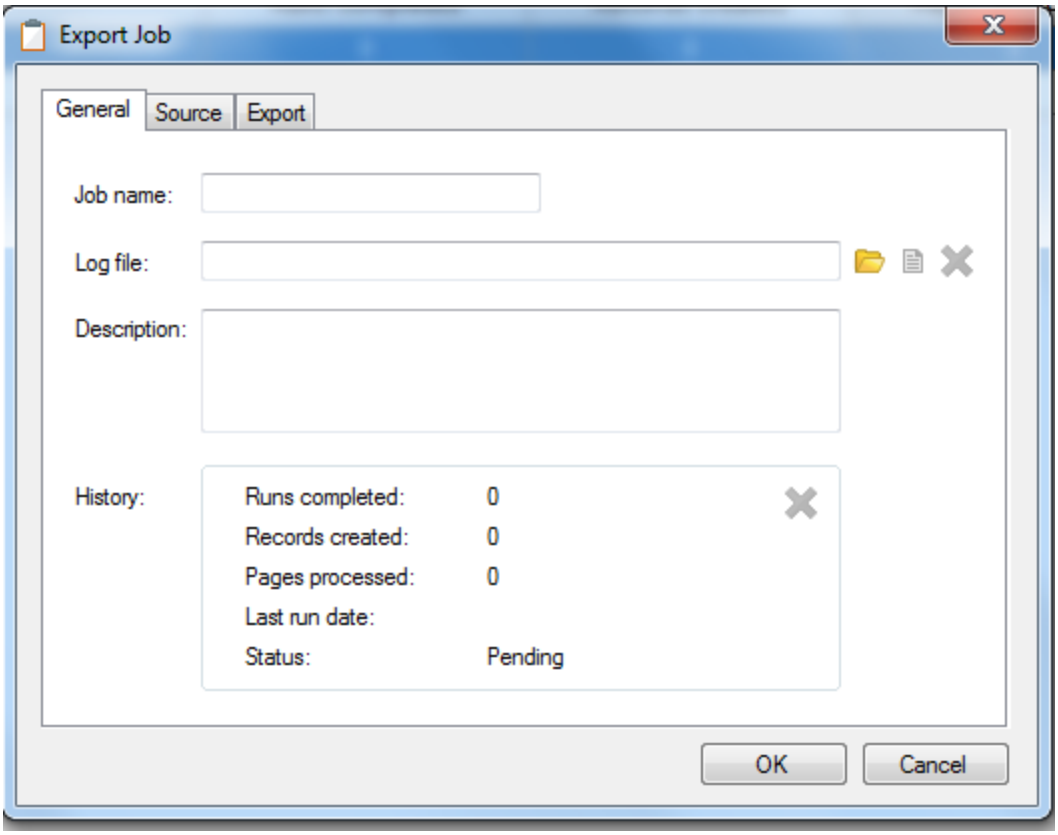
1. Create Jobs.
2. Edit Jobs
3. Run and Monitor Jobs

From a basic perspective, a job is a set of instructions that takes a report model, an input source, and an output path and processes documents via a monitored path. Jobs are part of “projects” which can contain many different jobs.

PDF Export Manager can actively manage many different jobs at once.

Creating a Job

To create a job, select the New Project button or option from the File menu. Then select “New Job” from the button bar. A dialog will open that allows you to establish the parameters of the job.



General Properties

The General properties tab allows you to define general information about your job.

Job Name - the name you give to the job. Each job name must be unique.

Log File - You may establish a file with the extension .log and then provide the path here. Export Manager will write the processing events to this file.

Description - You can provide the general description of the job here.

History - This is a read-only section that provides summary data of the processing history of the job.

Source Properties

The Source properties tab allows you to select the processing instructions and import settings.

Report Model Definition File - This property provides the location of the report model used to provide processing instructions to Export Manager.

Watched Folder Location - This property allows you to establish the path to a file share that is monitored by the Export Manager. When PDF documents are placed in this directory, Export Manager will automatically import and process them.

Watched File Name Pattern - This property allows you to specify the specific file name pattern that determines whether Export Manager will import the file or ignore it. You can wild cards to define the pattern.

Export Properties

Running Jobs

Once you complete defining a job, it will be displayed in the application. You can edit  or delete  jobs using the toolbar buttons. You cannot edit or delete jobs while they are running. You must first stop the job. There are several options when running jobs.

Run Job  allows you to start processing the job queue. All jobs, except those that are suspended, will be processed in turn.

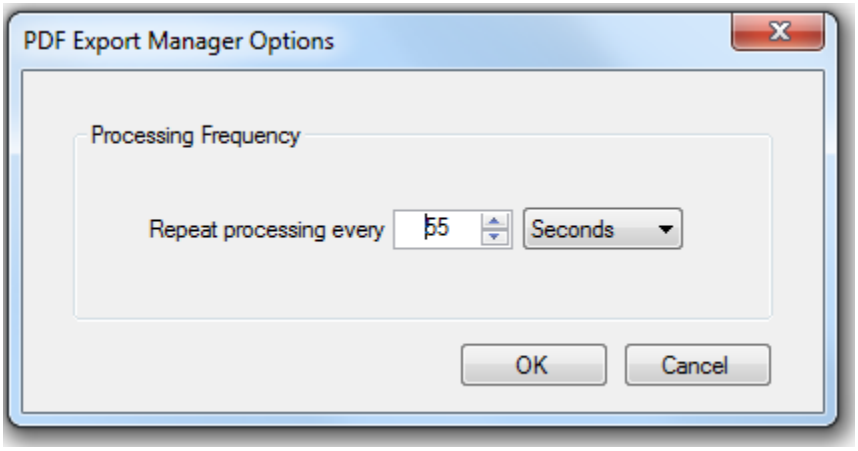
Stop Job  stops the processing of the job queue.

Suspend Job  allows you to suspend the selected jobs. Suspended jobs will not execute.

Resume Job  will cancel the suspended state of the selected jobs.

Options

The Options button  allows you to select the frequency at which Export Manager polls watch folders.



You can select intervals measured in seconds, minutes, and hours.

Export Results

The main screen of the Export Manager also acts as an output window for displaying progress and status messages. These messages are also stored in individual log files defined for each job. The output window has a fixed limit of 1,000 lines. Older messages will be deleted when this limit is reached. There is no limit for the number of messages stored in log file.

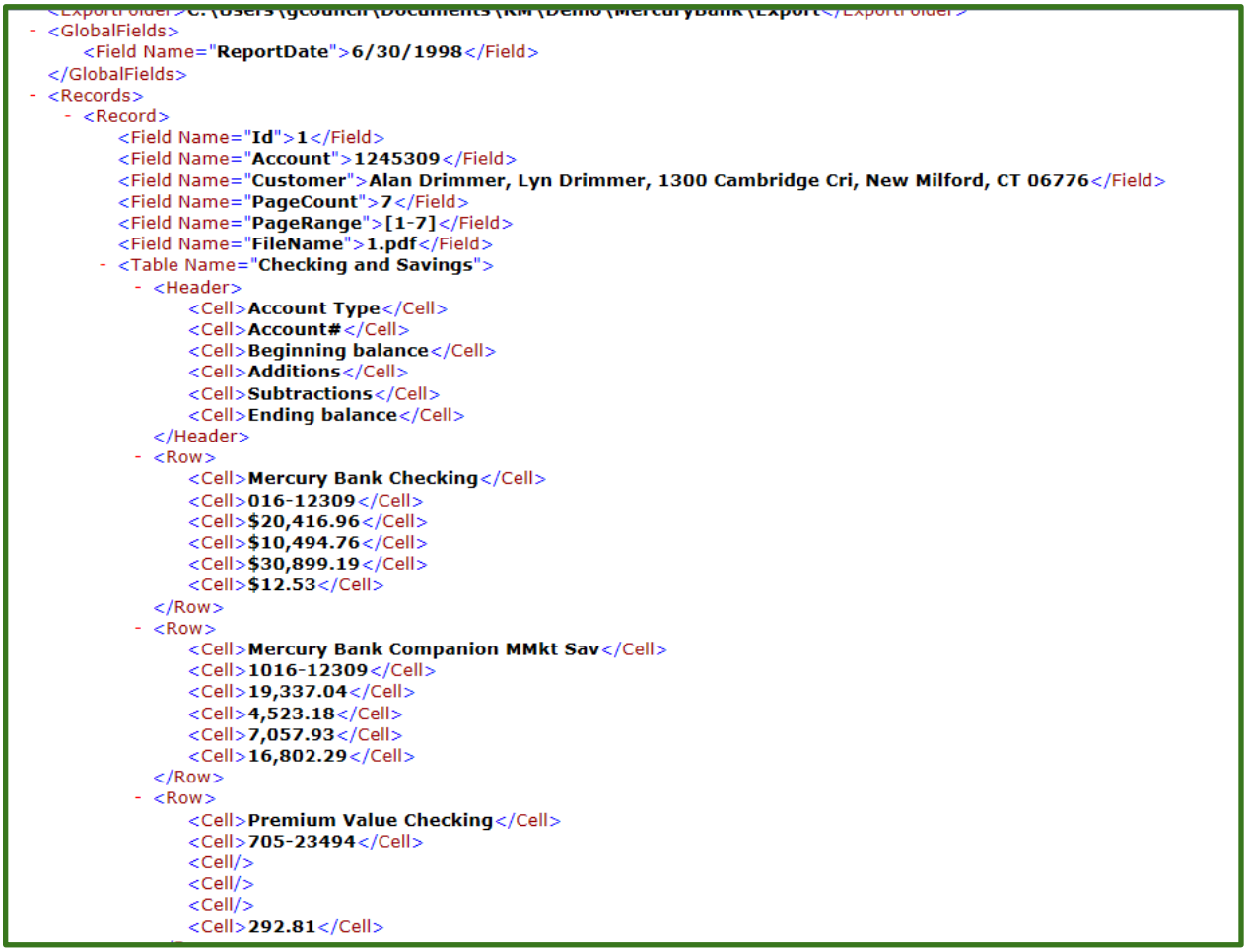
Name	Status	Runs completed	Records created	Pages processed	Last run
Mercury Bank	Pending	1	17	63	4/8/2018 9:4

```
Global Fields:
    ReportDate: 6/30/1998

Record 1: C:\Users\gcouncil\Documents\RM\Demo\MercuryBank\Export\1.pdf
    Pages:    7 [1-7]
    Account:  1245309
    Customer: Alan Drimmer, Lyn Drimmer, 1300 Cambridge Cri, New Milford, CT 06776

    Checking and Savings:
    Account Type                    Account#   Beginning balance    Additions   Subtractions   Ending balance
    ============================   ==========  =================   ==========  ============   ==============
    Mercury Bank Checking           016-12309         $20,416.96   $10,494.76     $30,899.19          $12.53
    Mercury Bank Companion MMkt Sav 1016-12309          19,337.04     4,523.18       7,057.93       16,802.29
    Premium Value Checking          705-23494                                                        292.81
    Premium Val Companion MMkt Sav  1705-23494                                                         0.03
    Classic Value Checking          709-89917                                                        2,774.12
    Classic Value Companion Sav     1709-89917                                                       3,625.36
    Classic Value Checking          855-52872                                                          722.35
    Classic Value Companion Sav     1855-52872                                                         443.91


Record 2: C:\Users\gcouncil\Documents\RM\Demo\MercuryBank\Export\2.pdf
    Pages:    4 [8-11]
    Account:  2245846
    Customer: Mitchell Diamond, 333 Ovington Blvd, Mitchell Field, NY 11771

    Checking and Savings:
    Account Type                    Account#   Beginning balance   Additions   Subtractions   Ending balance
    ============================   ==========  =================   =========   ============   ==============
    Mercury Bank Checking           016-22474          $1,125.26   $4,305.03     $4,088.43        $1,341.86
    Mercury Bank Companion MMkt Sav 1016-22474          34,905.24      149.64          .00        35,054.88
    High Yield Savings              1088-89194           6,931.42      315.16       590.00         6,656.58
    CD - matures 12/18/1998         1019-59249          16,025.06       75.45          .00        16,100.51
    CD - matures 10/27/1999         1414-72508          36,797.59      188.40          .00        36,985.99


Record 3: C:\Users\gcouncil\Documents\RM\Demo\MercuryBank\Export\3.pdf
    Pages:    3 [12-14]
    Account:  1285617
    Customer: Andy Olsen, 4108 Old Conejo St, Newberry, CA 90320
```

Once a job is done, all documents that are processed are moved to a sub-directory called “Done”. If the job fails, the documents are moved to a sub-directory called “Failed”.

XML File Output

Results from XML are delivered as records and tab/value pairs and table data is organized as table and cells as shown below:



```
<ExportFolder>C:\Users\gcouch\Documents\RM\Demo\MercuryBank\Export</ExportFolder>
- <GlobalFields>
    <Field Name="ReportDate">6/30/1998</Field>
  </GlobalFields>
- <Records>
  - <Record>
      <Field Name="Id">1</Field>
      <Field Name="Account">1245309</Field>
      <Field Name="Customer">Alan Drimmer, Lyn Drimmer, 1300 Cambridge Cri, New Milford, CT 06776</Field>
      <Field Name="PageCount">7</Field>
      <Field Name="PageRange">[1-7]</Field>
      <Field Name="FileName">1.pdf</Field>
    - <Table Name="Checking and Savings">
      - <Header>
          <Cell>Account Type</Cell>
          <Cell>Account#</Cell>
          <Cell>Beginning balance</Cell>
          <Cell>Additions</Cell>
          <Cell>Subtractions</Cell>
          <Cell>Ending balance</Cell>
        </Header>
      - <Row>
          <Cell>Mercury Bank Checking</Cell>
          <Cell>016-12309</Cell>
          <Cell>$20,416.96</Cell>
          <Cell>$10,494.76</Cell>
          <Cell>$30,899.19</Cell>
          <Cell>$12.53</Cell>
        </Row>
      - <Row>
          <Cell>Mercury Bank Companion MMkt Sav</Cell>
          <Cell>1016-12309</Cell>
          <Cell>19,337.04</Cell>
          <Cell>4,523.18</Cell>
          <Cell>7,057.93</Cell>
          <Cell>16,802.29</Cell>
        </Row>
      - <Row>
          <Cell>Premium Value Checking</Cell>
          <Cell>705-23494</Cell>
          <Cell/>
          <Cell/>
          <Cell/>
          <Cell>292.81</Cell>
```

CSV and Pipe File Output

For delimited files, data is delivered as rows and columns for tag/value data.

Id	Account	Customer	PageCoun	PageRang	FileName	Table1
1	1245309	Alan Drim	7	[1-7]	1.pdf	Checking and Savings.csv
2	2245846	Mitchell D	4	[8-11]	2.pdf	Checking and Savings.csv
3	1285617	Andy Olse	3	[12-14]	3.pdf	Checking and Savings.csv
4	8764534	Thomas D	4	[15-18]	4.pdf	Checking and Savings.csv
5	8763092	Lawrence	3	[19-21]	5.pdf	Checking and Savings.csv
6	2345982	Lana Stryk	4	[22-25]	6.pdf	Checking and Savings.csv
7	8730213	Steve Mar	3	[26-28]	7.pdf	Checking and Savings.csv
8	8835263	Chris Hen:	4	[29-32]	8.pdf	Checking and Savings.csv
9	8922345	Lou Guter	3	[33-35]	9.pdf	Checking and Savings.csv
10	2199802	Raju Mana	4	[36-39]	10.pdf	Checking and Savings.csv
11	4000234	Hugh One	3	[40-42]	11.pdf	Checking and Savings.csv
12	1011374	Mary and	4	[43-46]	12.pdf	Checking and Savings.csv
13	4490932	Gail Brave	3	[47-49]	13.pdf	Checking and Savings.csv
14	7784532	Afton Gill	4	[50-53]	14.pdf	Checking and Savings.csv
15	9836543	Michael E:	3	[54-56]	15.pdf	Checking and Savings.csv
16	6634214	Paul Spilk	4	[57-60]	16.pdf	Checking and Savings.csv
17	9033213	Ralph Maj	3	[61-63]	17.pdf	Checking and Savings.csv

If table data is present, a separate CSV file is created for each table that is processed with the table name provided as the file name.

Id	Account T	Account#	Beginning balance	Additions	Subtractions	Ending balance
1	Mercury B	016-12309	$20,416.96	$10,494.76	$30,899.19	$12.53
1	Mercury B	1016-1230	19,337.04	4,523.18	7,057.93	16,802.29
1	Premium	705-23494				292.81
1	Premium	1705-23494				0.03
1	Classic Va	709-89917				2,774.12
1	Classic Va	1709-89917				3,625.36
1	Classic Va	855-52872				722.35
1	Classic Va	1855-52872				443.91
2	Mercury B	016-22474	$1,125.26	$4,305.03	$4,088.43	########
2	Mercury B	1016-2247	34,905.24	149.64	0	35,054.88
2	High Yield	1088-8919	6,931.42	315.16	590	6,656.58
2	CD - matu	1019-5924	16,025.06	75.45	0	16,100.51
2	CD - matu	1414-7250	36,797.59	188.4	0	36,985.99
3	Mercury B	016-28171	$509.31	$2,018.30	$2,050.81	$476.80
3	Mercury B	1016-2817	13,130.24	44.85	120	13,055.09
3	CD - matu	1374-5250	5,472.32	28.02	0	5,500.34
3	CD - matu	1374-5240	21,919.69	112.23	0	22,031.92
4	Mercury B	016-22474	$1,125.26	$4,305.03	$4,088.43	########
4	Mercury B	1016-2247	34,905.24	149.64	0	35,054.88
4	High Yield	1088-8919	6,931.42	315.16	590	6,656.58
4	CD - matu	1019-5924	16,025.06	75.45	0	16,100.51

Text Output

Text output format for extracted data is similar to the way data is formatted in the output window and log files. The text output for all extracted data types is saved to a single file.