4

Taskmaster RRS Applications

Taskmaster 7 supports applications of two kinds: *Taskmaster* and *Taskmaster RRS*.

- A *Taskmaster* application uses Rulemanager to construct the application's Document Hierarchy; to set up its fingerprints; and to define the rules that will guide the application's *rulerunner* tasks as they process the contents of a batch. A *Taskmaster* application employs the Taskmaster Server Service to access and process the information between Taskmaster Clients and the application's files and databases.

- A *Taskmaster RRS* application takes advantage of the additional set up and testing facilities of Datacap Studio, and stronger connections to the *rulerunner* tasks in jobs of the *Taskmaster* workflow. In addition, a *Taskmaster RRS* application adds a Taskmaster Rulerunner Service to facilitate the transfer of setup information from Datacap Studio to the application's Taskmaster Clients.

Chapter 4 describes the structure and elements of a *Taskmaster RRS* application as it covers these topics:

4.1 Introduction to the Datacap RRS Application

The chart on the next page summarizes the structure and elements of a *Datacap RRS* application. The next few pages introduce these elements and an application's broad structure; follow-up sections examine and illustrated individual elements more closely.

✓ An application's development and operation employs two distinct but interlocked *Taskmaster 7.0* components:

Datacap Studio

You use Datacap Studio (represented at the top of the chart) to:

- Set up the application's Document Hierarchy. This is the application's fundamental structure: it consist of objects at four levels: **Batch, Document, Page** and **Field.**

- Assign values to the properties of specific objects.

- Define additional *setup* variables and, if applicable, assign values to these variables.

- Define *runtime* variables for specific objects. Values for these objects are assigned during task operations.

- Construct one or more fingerprints representing the Document Hierarchy's *source* page (a *source* page contains fields with data that is to be recognized, verified, validated and exported.). A fingerprint consists of an Image file (.tif) and a Recognition file (.cco).

- Zone specific fields on the fingerprint's image. These zones provide the Recognition file with information about the location of the zoned fields.

- Assign additional Recognition parameters to certain zoned fields.

- Define Ruleset/Rule combinations and bind them to objects of the Document Hierarchy. During the application's operations, *rulerunner* tasks will apply the Rulesets' rules as they process the Document Hierarchy's objects.
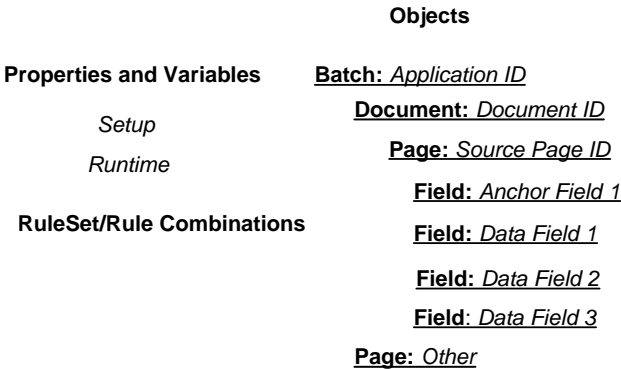
Taskmaster Client

The remainder of the chart highlights an application's other elements – those managed by Taskmaster Client.

Workflow Setup integrates the four previously defined components on the left to produce the workflow infrastructure on the right.
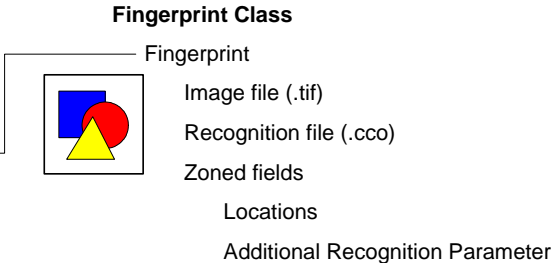
Security Setup defines and assigns security parameters that determine who has access to the application and its procedures.

Workflow operations carries out the tasks of a Main job – or the tasks of any other job.
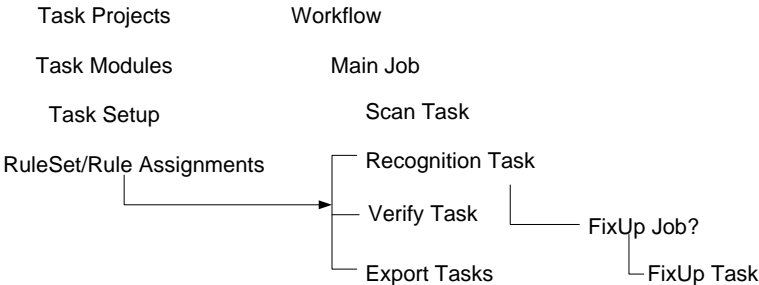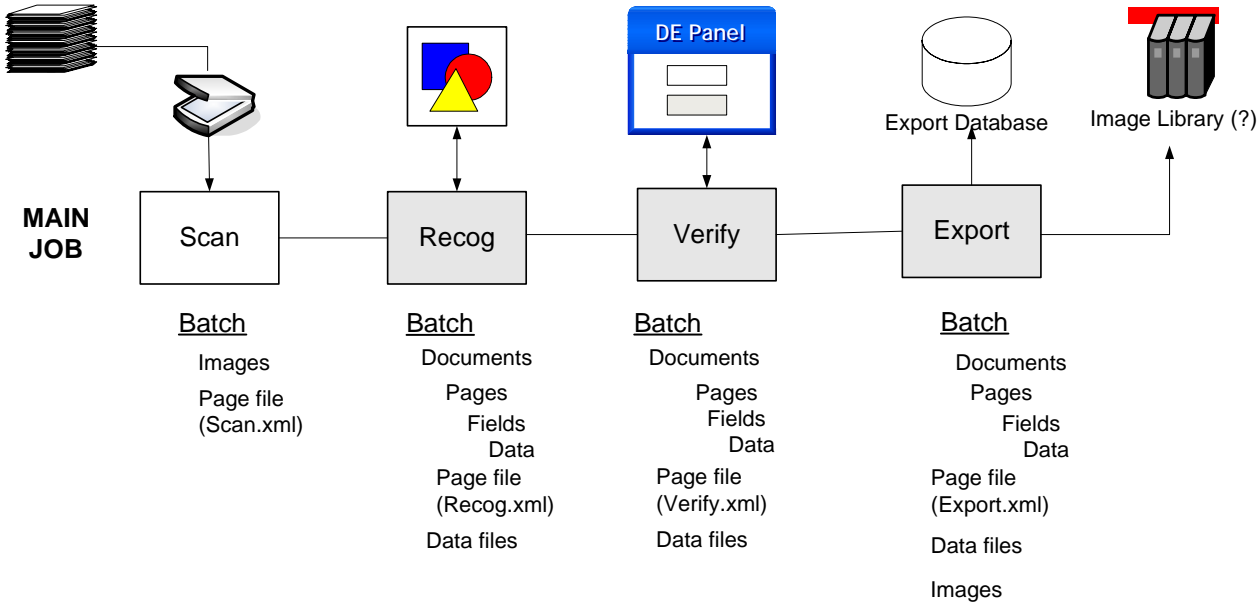
DOCUMENT HIERARCHY

Objects

Properties and Variables

Batch: *Application ID*

Setup

Document: *Document ID*

Runtime

Page: *Source Page ID*

RuleSet/Rule Combinations

Field: *Anchor Field 1*

Field: *Data Field 1*

Field: *Data Field 2*

Field: *Data Field 3*

Page: *Other*

SOURCE PAGE FINGERPRINTS

Fingerprint Class

Fingerprint

Image file (.tif)

Recognition file (.cco)

Zoned fields

Locations

Additional Recognition Parameter

WORKFLOW SETUP

Task Projects	Workflow
Task Modules	Main Job
Task Setup	Scan Task
RuleSet/Rule Assignments	Recognition Task
	Verify Task
	Export Tasks

FixUp Job?

FixUp Task

SECURITY SETUP

Users

User Groups

Workstations

Shortcut Icons - Operations

WORKFLOW OPERATIONS

Export Database   Image Library (?)

MAIN JOB

Scan	Recog	Verify	Export

Batch

Images

Page file (Scan.xml)

Batch

Documents

Pages

Fields

Data

Page file (Recog.xml)

Data files

Batch

Documents

Pages

Fields

Data

Page file (Verify.xml)

Data files

Batch

Documents

Pages

Fields

Data

Page file (Export.xml)

Data files

Images

4.1.1   How to Access an Application

The **Datacap Taskmaster Client Installation** procedures covered in the Taskmaster
Installation and Configuration Guide create icons on your desktop. Here are two:

1040EZ Client (in this example) accesses the *Please Login* dialog. Filling in the three
security fields and clicking on the OK button open the *Taskmaster Window's Operations*
tab.
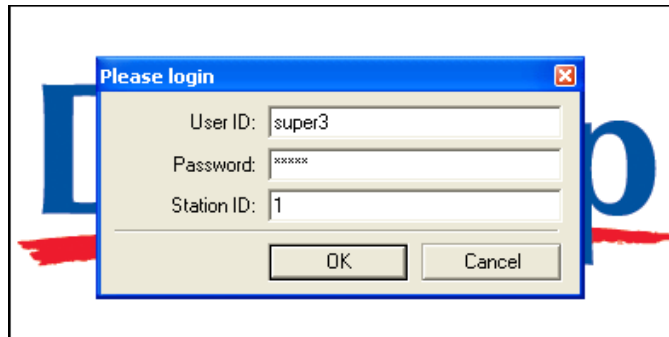
Please Login dialog
1040EZ Application (Taskmaster RRS)

Double-clicking on the **DStudio** icon launches the Select Application wizard.

You use the *Applications* screen to indicate which application you'll be working with,
and the follow-up *Taskmaster Login* screen to open DStudio's **Rulemanager Page**.
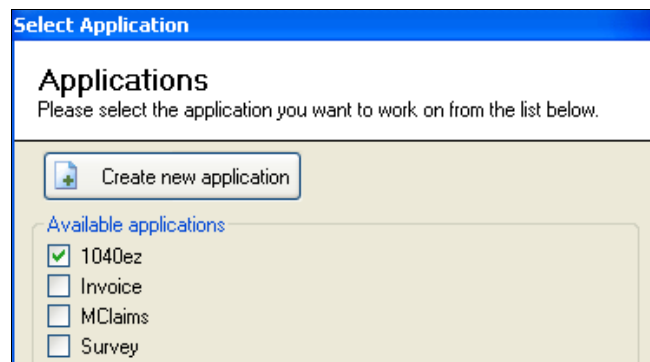
Select Application Wizard – *Application screen*

Select Application Wizard – *Taskmaster Login Screen*

Very important! You cannot sign on to Taskmaster Client and Datacap Studio at the same time, using the same security codes.

Just as important! The *1040EZ* training application is ***indestructible***. Do not hesitate to investigate its many features, and experiment with every procedure.

4.2 The Document Hierarchy

An application's Document Hierarchy is made up of **objects** at four **levels**. All objects have properties and variables. **Rulesets** and their rules have been "bound" to some objects.

For a brief examination of a Document Hierarchy look, follow the instructions in the previous section to open the *1040EZ* application's Datacap Studio – and its **Rulemanager Page**.

> Go immediately to the *Document Hierarchy* tab – and click on the **Lock** toggle on the left end of the tab's toolbar. *Warning!* Don't ignore this apparently minor but essential step: it allows you to examine the Document Hierarchy and its information, at your leisure.

1040EZ Document Hierarchy (partial)
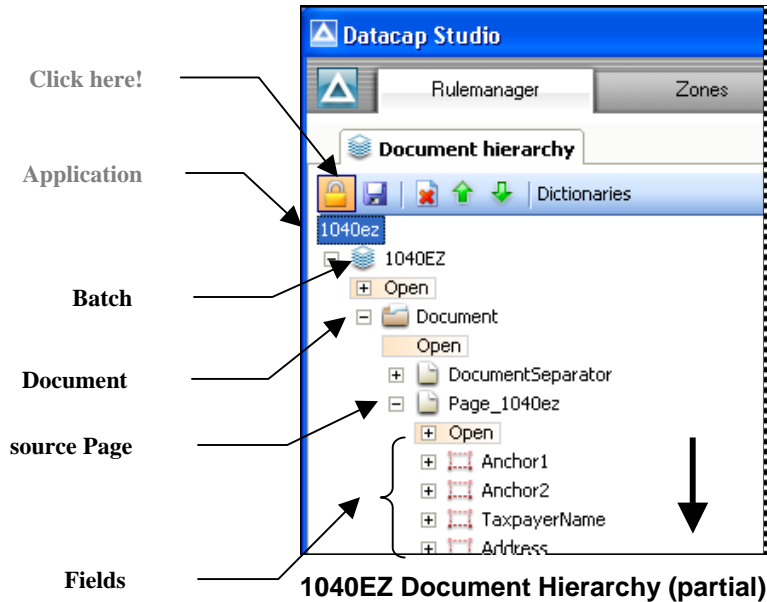
The illustration above displays a Document Hierarchy's basic parent/child structure.

✓ The **Application** level at the top identifies the application but is not really part of the hierarchy.

The **Batch** level can only have one object : *1040EZ* , in the example. This object is the parent to objects at the other levels. Rules assigned to the **Batch** object have an impact on an actual batch during processing, and on its contents.

Document objects help organize the batch into a series of documents, and the pages in each document.

Page objects are principal elements of the Document Hierarchy. The *source* **Page** object – *Page_1040ez*, in the example – is made up of the hierarchy's **Field** objects. An *Other* **Page** object (at the bottom of the *1040EZ* Document Hierarchy) is part of every hierarchy

because the object represents unidentified pages produced by the opening Scan task. Page objects are subject a number of Ruleset/Rule combinations.

Field objects are usually children of the source **Page** object, and often have multiple properties and variables, zone parameters. and special Recognition properties.
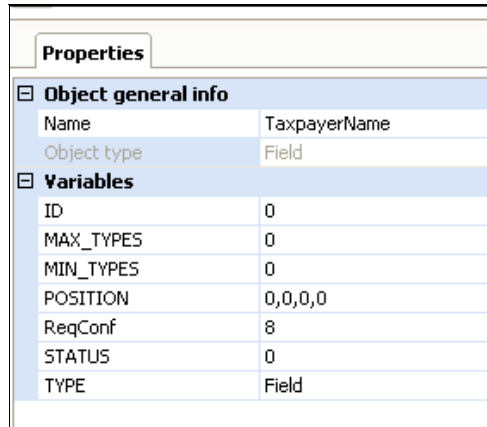
4.2.1   Anatomy of an Object

To appreciate the basis of a typical object:
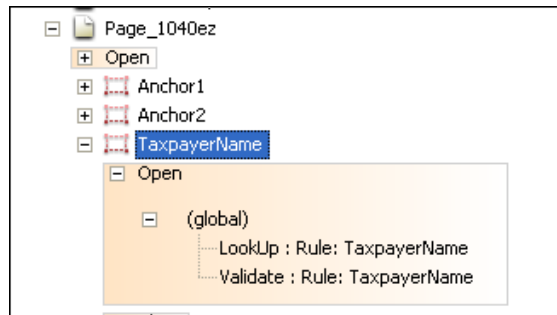
1.   Highlight the Object ID in the Document Hierarchy:
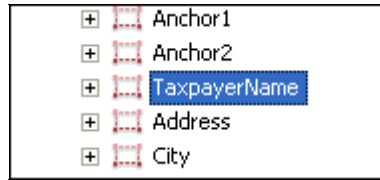
2.   Review the settings in the *Properties* dialog. (For more about these settings, see Chapter 5.)
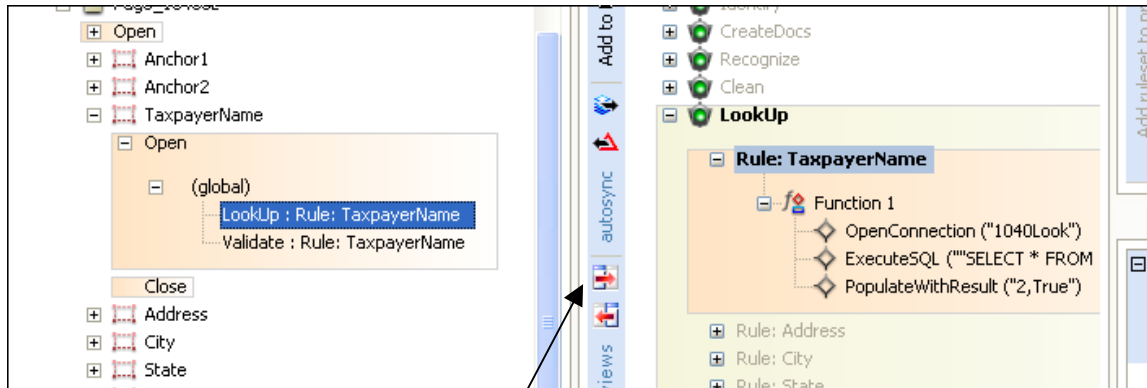

3.   Open the object's listing to reveal its global Ruleset/Rule combinations.

4. Highlight a Ruleset/Rule combination.

5. Click on the ➔ icon for a close look at its details.

Click here.

✓ The procedure above will give you a hands-on introduction you to a Document Hierarchy; its levels and their objects; and to their settings and rules. Chapter 5 is a complete review of the Document Hierarchy.
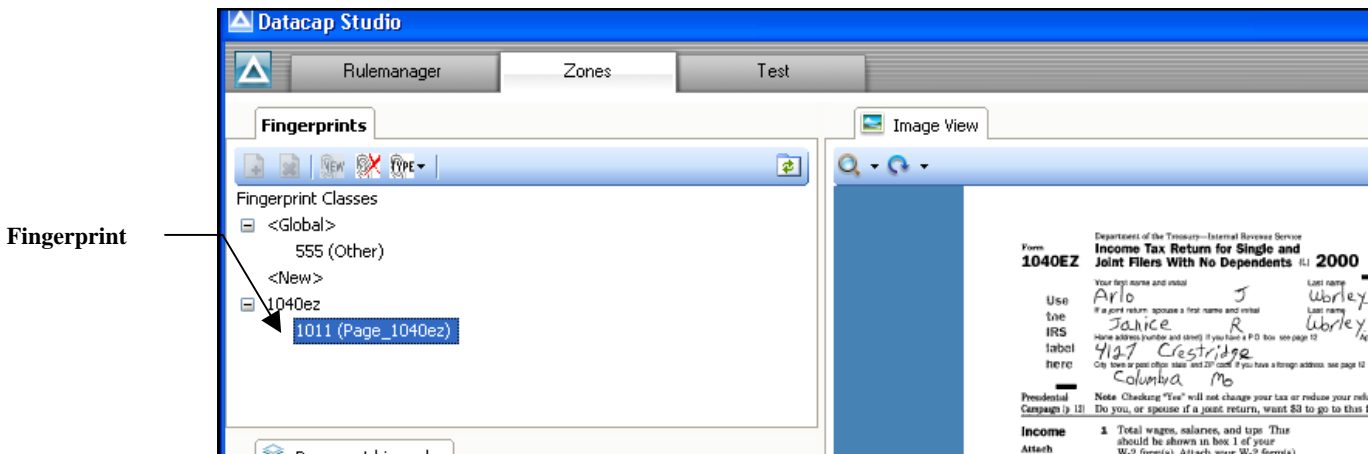
4.3 Fingerprints

A fingerprint is an important source of information for:

- the setup of the Document Hierarchy

- the definition of Ruleset/rule combinations in Datacap Studio

- the identification of *source* pages by a *rulerunner* task such as the *1040EZ* application's PageID task.

Datacap Studio's **Zones Page** sets up an application's fingerprint(s).

Because the *1040EZ* application works exclusively with one form and a *single* source page, it only needs just one fingerprint: *1011*:

Fingerprint Class: 1040EZ
Fingerprint: 1011(Page_1040ez)

In the *Fingerprints* tab, when you highlight the Fingerprint ID, the *Image View* tab displays the fingerprint's Image file – in this case, **1011.tif.**

A limited version of the *Document Hierarchy* tab is part of the **Zones Page** as well. If you highlight a **Field** object in this tab, the fingerprint's image will surround its zoned fields with red rectangles:
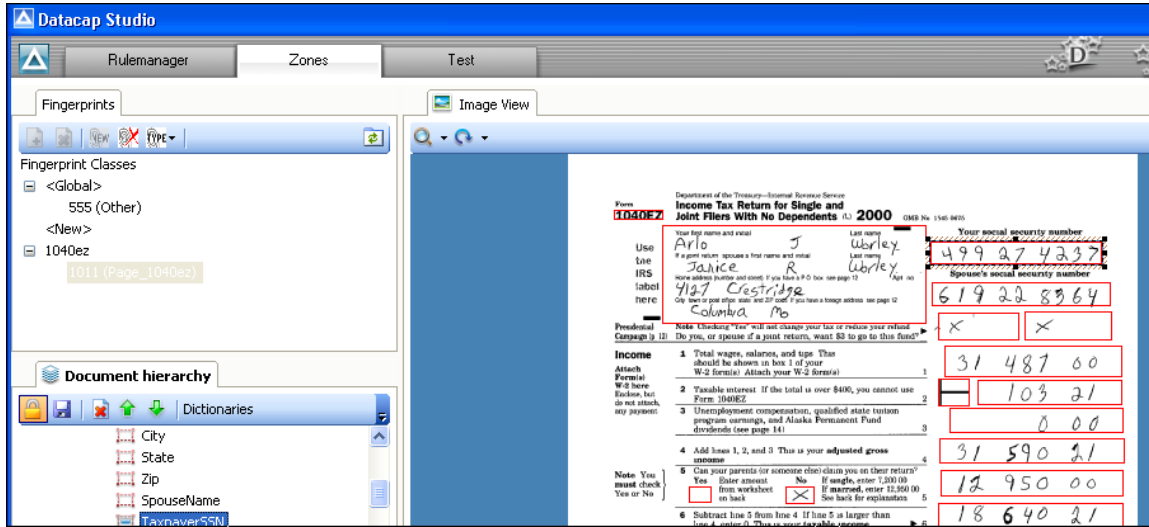
Zoning a field specifies it location within the fingerprint.

You can also assign Recognition properties to a zoned field.

Location and Recognition parameters are automatically assigned to variables of the field.

Location and Recognition parameters for all fields become part of the fingerprint's Recognition file (.cco).

Fingerprint 1011: Zoned Fields

The pre-configured *Invoices* application processes invoices and statements with varying formats and content. As a result, this application uses multiple fingerprints:
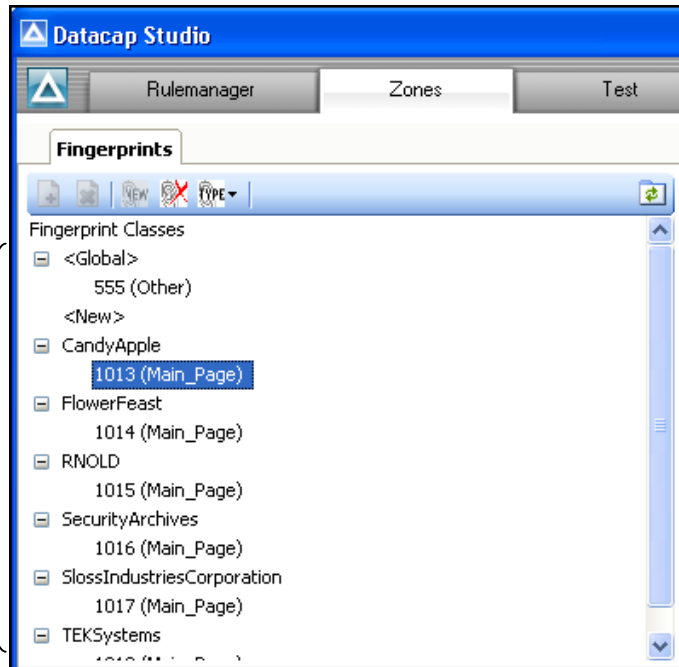
Fingerprints of the Invoices application

Invoices Application – *sample fingerprints*

One result is that a **Field** object on the *Invoices* Document Hierarchy might have a different value for a particular variable. In the example on the next page, the *Remittance_Zip* **Field** object has been separately zoned on six fingerprints (1013-1018), so the Document Hierarchy lists **POS** location coordinates for each field.
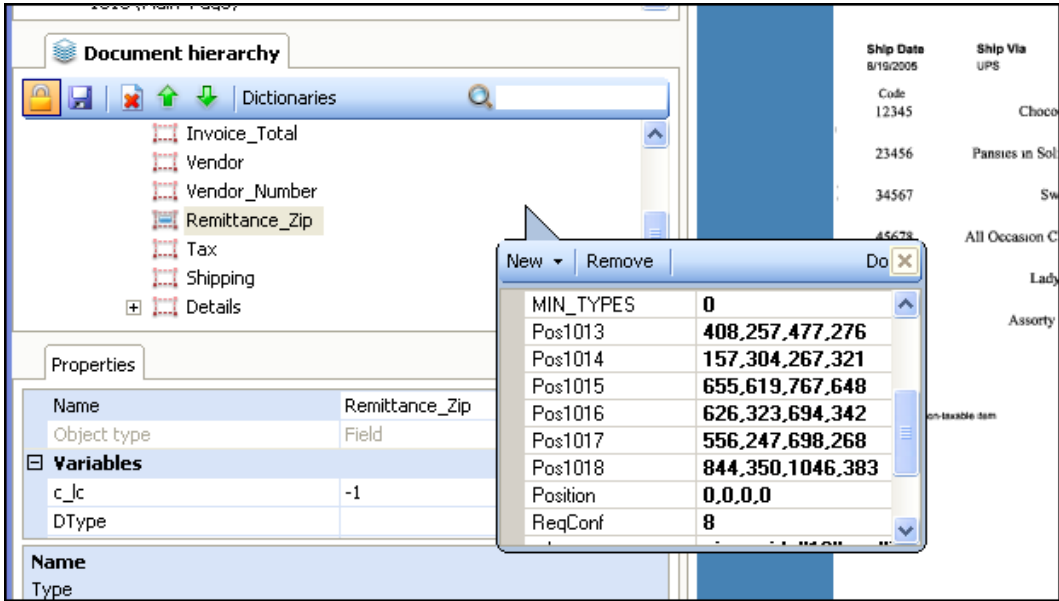
Remittance_Zip Field Object:
Fingerprint Position (POS) Values

✓ A fingerprint's Image and Recognition files are stored in the application's **fingerprints** directory. For example:
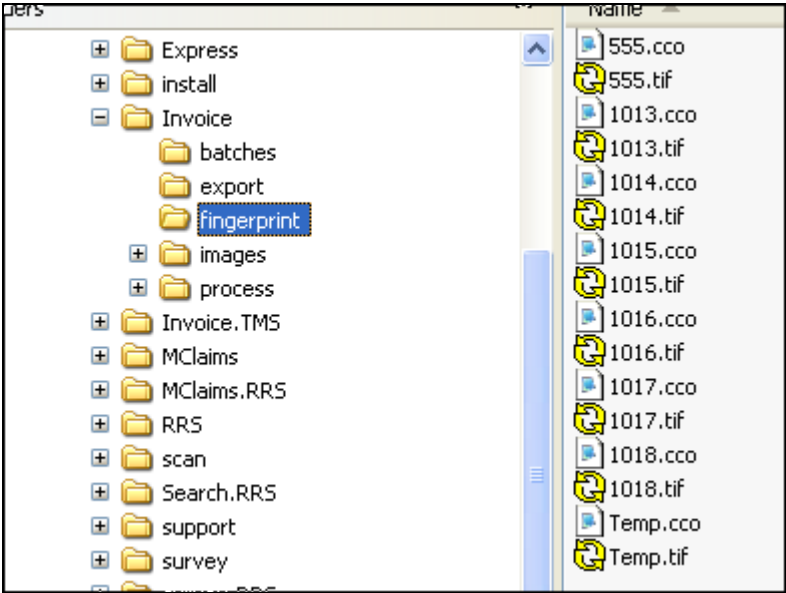
Invoices Application Fingerprints

4.4 Rulesets and Rules

A rule and its actions tell a task how to process an item represented by an object of the Document Hierarchy.

In this simple example from the *1040EZ* application,
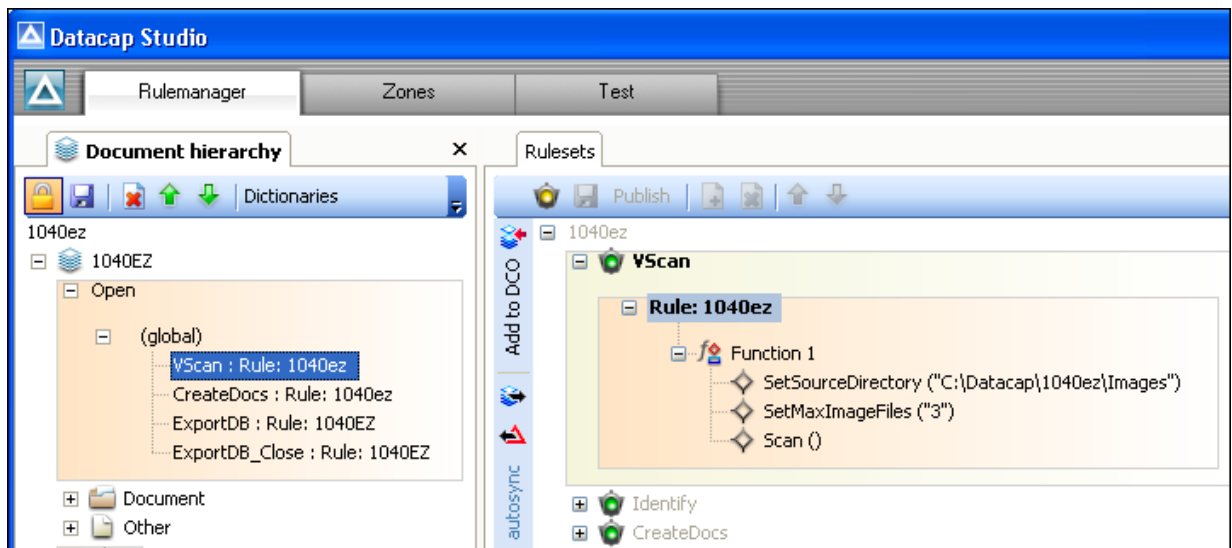
> The Ruleset is **VScan**

>> The rule is **Rule:1040ez**

>>> The actions of the rule's first and only function – **Function 1** - are:

```
SetSourceDirectory("C\Datacap\1040ez\Images")

 SetMaxImageFiles(3)

Scan()
```


1040EZ Datacap Studio: *Rulemanager Page*
VScan Rule

This Ruleset/Rule combination (**VScan:Rule: 1040EZ**) is bound to the Document Hierarchy's **Batch** object. When the Scan task of the *1040EZ* application's Main Job runs, the task will respond to the rule's actions – *if* the Ruleset has been linked to the Task Definition

This link is forged in two ways (illustrated on the next page):

1. Workflow, Job and Task Definitions are assembled in Taskmaster Client's *Taskmaster Administrator.*

2. Rulesets (and their rules) are assigned to *rulerunner* tasks in the **Task Profiles** tab of the **Rulemanager Pag**e.
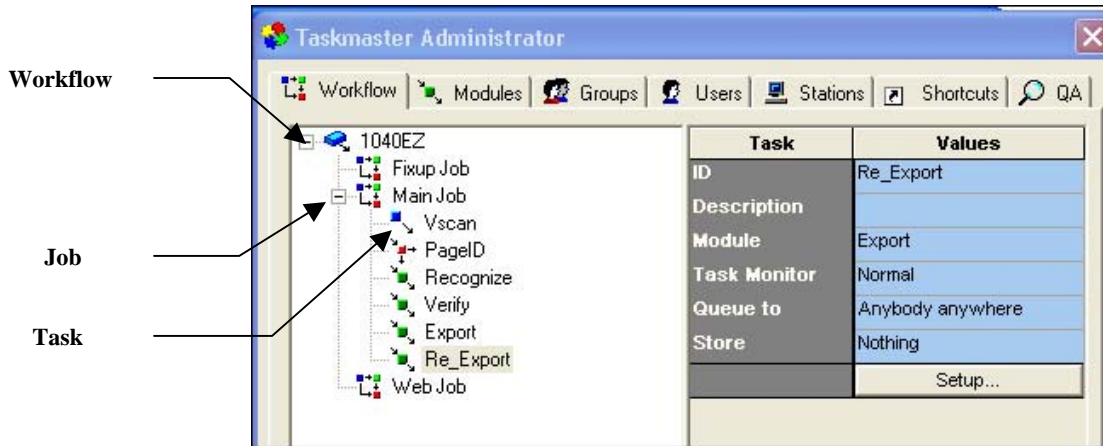
1040EZ Taskmaster Administrator – *Workflow tab*
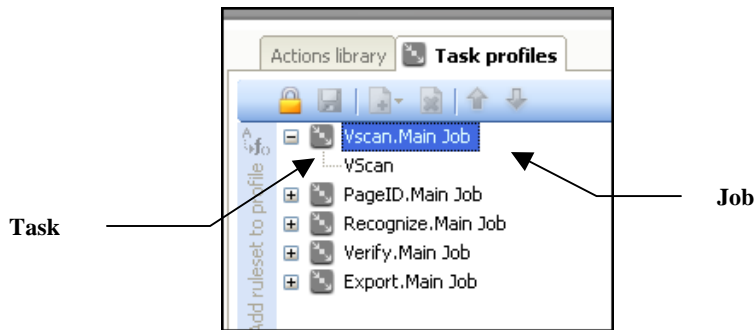
DStudio Rulemanager Page
Task Profiles tab

✓ Chapter 6 describes procedures for: defining Rulesets and rules; binding Ruleset/Rule combinations to objects of the Document Hierarchy; and linking Rulesets to Task Profiles. Chapter 8 examines techniques that test rules and their actions, and the connections between Ruleset/Rule combinations and the tasks that apply them.
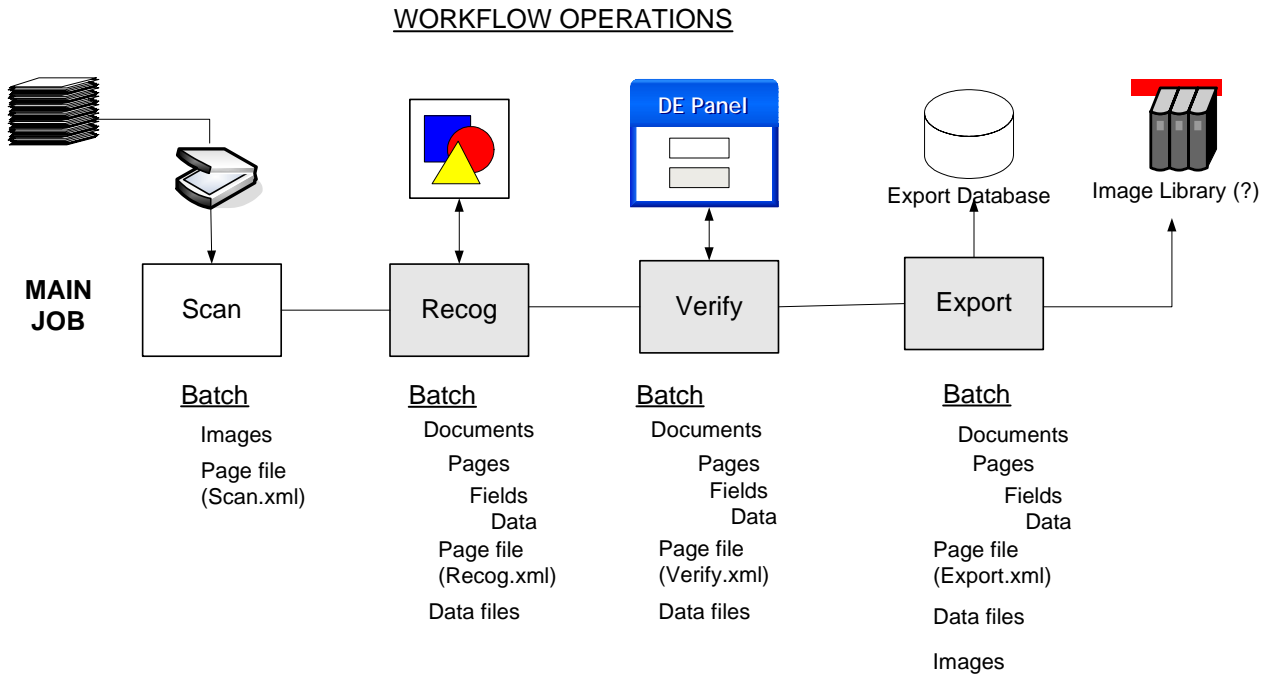
4.5 Workflow Operations

Job-Task shortcuts in Taskmaster Client's *Operations* tab initiate the batch processing procedures summarized graphically at the bottom of the chart on Page 3, and reproduced here:

WORKFLOW OPERATIONS

Scanning

A Main job's Scan task starts off by creating a new processing batch, then engages a physical scanner to scan paper and generate an Image file (.tif) for each page. Organizationally, the task associates every page with the *Other* **Page** object of the Document Hierarchy. The task concludes by preparing a Page file (**Scan.xml**) with details of the batch and its pages. *Important!* Usually, rules do not guide a Scan task that uses a scanner to process paper.

Recognition

In many ways, the Recognition task is the application's workhorse. Responding to numerous Ruleset/Rule combinations, the task:

- Assesses specifics in the **Scan.xml** Page file

- Matches certain pages in the batch to the application's fingerprint(s) – and sets aside the *source* pages.

- If required, identifies the Page Types of other pages.

- Locates the fields on every *source* page.

- Recognizes the data in selected fields.

- Prepares a Data file (.xml) for each *source* page, and adds the recognized values to this file.

- If appropriate, organizes the batch into a series of documents and pages, ensuring that each document has at least one *source* page.

- Generates a Page file (**recog.xml**) with specifics about the batch, its document and it pages. Within this file, the listing for a *source* page is a link to its Data file.

A Recognition task can take many other steps – all according to the Ruleset/Rule combinations bound to objects of the Document Hierarchy, and assigned to the task.

Verify

This task uses a pair of fields in a ***Data Entry Panel*** to present an operator with hard-to-read or missing *recognized* values on a *source* page.

> A Snippet displays the image of a field and its value.

> An interactive Data Edit field also contains the recognized value, but allows the operator to modify the value after reviewing the snippet's content.

As it processes a "problem" page, the task updates its Data file with any changes.

After finishing with a batch, a typical Verify validates the data in certain fields, and produces its own Page file (verify.xml).

There are few Ruleset/Rule combinations associated with most Verify tasks.

Export

Although an Export task may well respond to just one or two Ruleset/Rule combinations, one is likely to have a function with many Export actions. Each action retrieves data from a page's Data file, or from o other sources, and places the data in an Export file or Export database

Some Export tasks update the application's Image Library by adding the Image files in the batch.

4.6 Changes in Taskmaster Applications

This section reviews changes to *Taskmaster* applications involving the Rules database and actions as XML elements.

4.6.1 Rules Database

Important! A *Taskmaster 7.5* application does not use a Rules database to store information about Rulesets/Rules combinations and the actions that are part of a particular rule.

Instead, this information is stored in the Rules files (.rul) in the application's folder, in the **Libs** sub-folder of the **Datacap** directory's **RRS** folder.
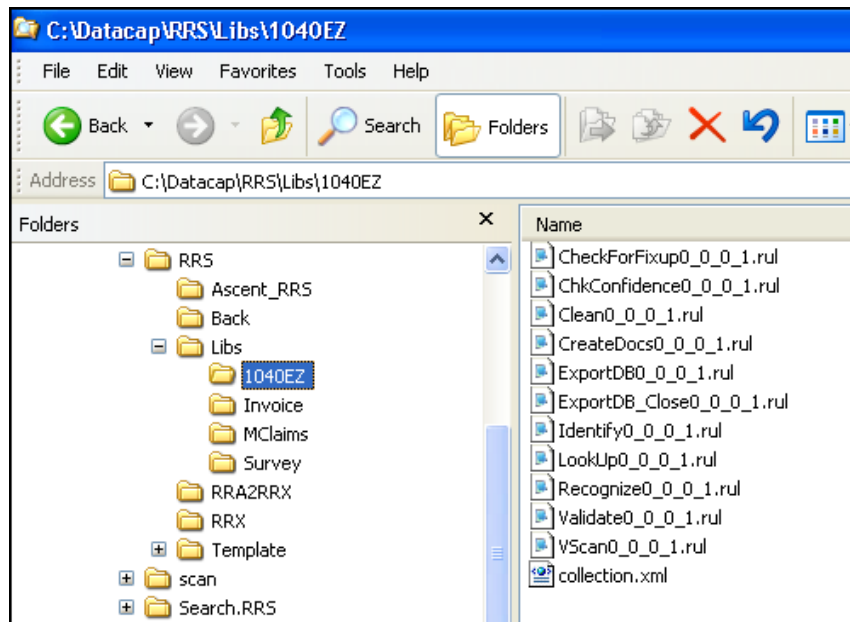
1040EZ Rules Files

Information in a Rules file specifies a Ruleset's rules and their actions, and the connections between Ruleset/Rule combinations and objects of the Document Hierarchy.

✓ Inside this folder, the application's **collection.xml** file summarizes the data in individual Rules files.

4.6.2 Actions (XML)

In a *Taskmaster RRS* application, an action is part of a Rulerunner Actions (.rrx) file located in the **RRX** sub-folder of the **Datacap** directory's **RRS** folder:

Datacap Rulerunner Actions files

Within a file, an action is an XML element. For example,

XML version

VScan actions

XML Declaration – *Vscan Actions*

On the next page, the *1040EZ* application's **VScan** Ruleset is a "well-formed" XML document.

Actions (XML)


```
VScan0_0_0_1.rul - Notepad
File  Edit  Format  View  Help
<ruleset name="VScan" id="1" v="0.0.0.1">
          <rule name="Rule: 1040ez" id="1">
                    <func name="Function 1">
                              <a name="SetSourceDirectory" ns="vscan">
                                        <p type="bInter"/>
                                        <p type="bDebug"/>
                                        <p type="strParam" v="C:\Datacap\1040ez\Images"/>
                              </a>
                              <a name="SetMaxImageFiles" ns="vscan">
                                        <p type="bInter"/>
                                        <p type="bDebug"/>
                                        <p type="strParam" v="3"/>
                              </a>
                              <a name="Scan" ns="vscan">
                                        <p name="bInter" type="bInter"/>
                                        <p name="bDebug" type="bDebug"/>
                              </a>
                    </func>
          </rule>
</ruleset>
```
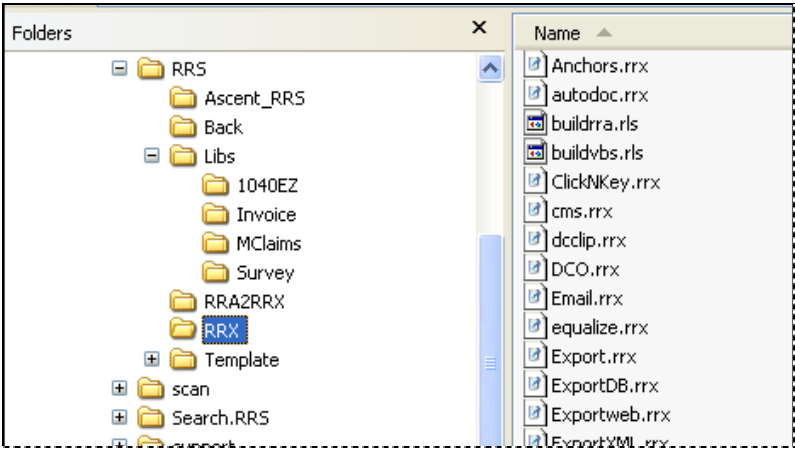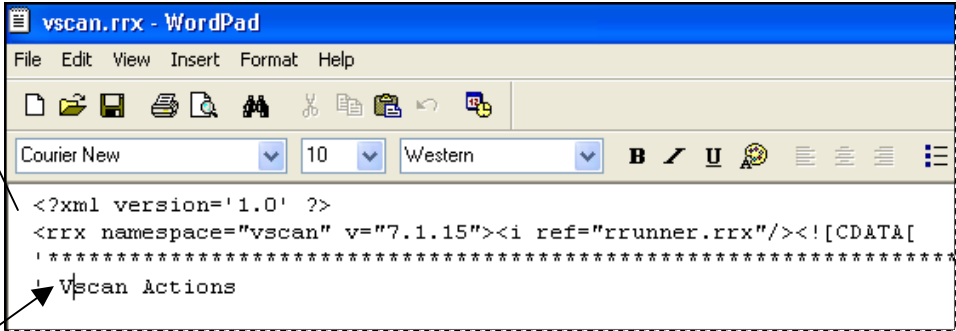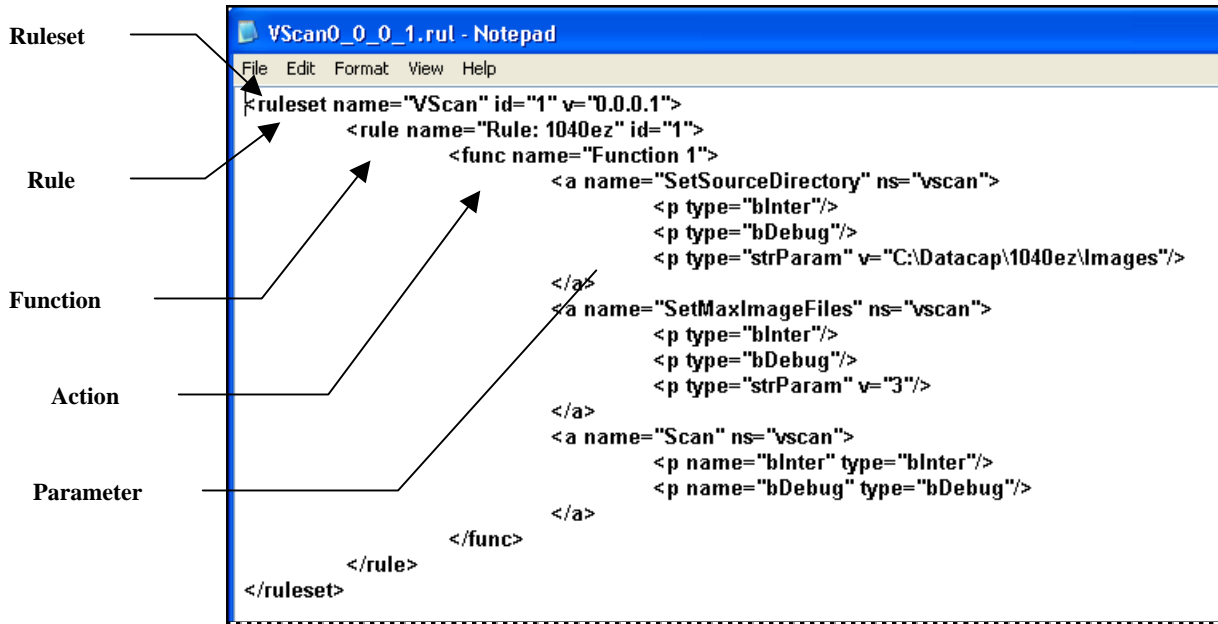
Labels (left side): Ruleset, Rule, Function, Action, Parameter

1040EZ Application:
VScan Ruleset