2

Overview of the Taskmaster Application

Chapter 2 provides a broad description of the *Taskmaster* application's structure and components. Subsequent chapters thoroughly examine each component.

Chapter 2 covers these topics:

Introduction

The chart on the following page depicts the architecture of a fictional *Taskmaster* application (*MQSW*) as three configuration hierarchies that result in the *MQSW* workflow.

- Objects at the four levels of the **Document Hierarchy** formally delineate all aspects of the application's focus, including the make-up of the **form** to be processed. Page 3 examines the Document Hierarchy's structure, and properties of objects at **Batch**, **Document**, **Page** and **Field** levels.
- Members of a **RuleSet Hierarchy** define processing rules, and apply the rules to individual objects of the Document Hierarchy. (For explanations of the RuleSet Hierarchies, see Page 7.)
- The components of an application's **Workflow Hierarchies** – **workflow**, **jobs** and **tasks** – incorporate specifications of both the Document Hierarchy and of the RuleSet Hierarchies. Page 11 describes the configuration of the *MQSW* Workflow Hierarchy.

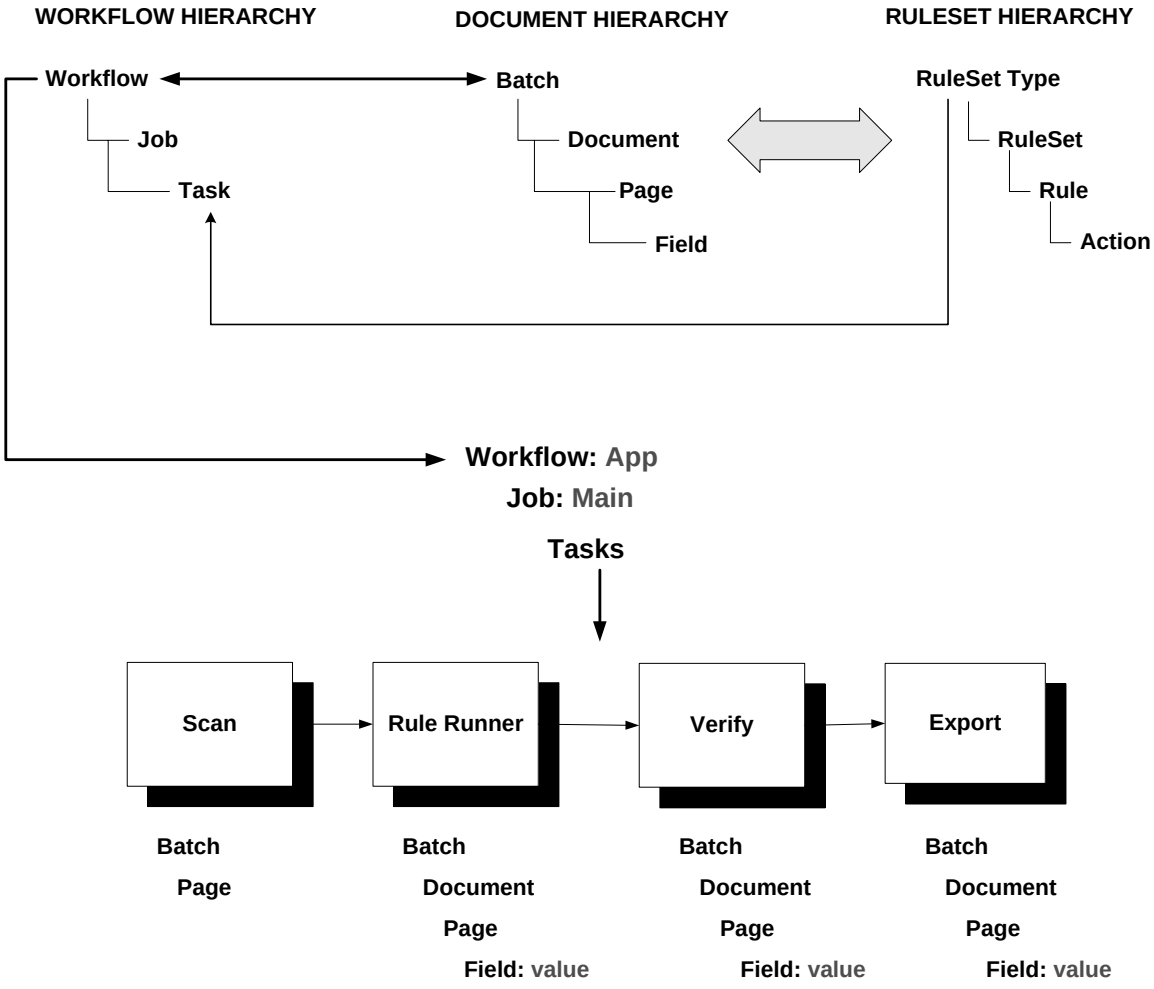
The lower portion of the chart shows the tasks of the application's Main job – the job that takes care of a form – an MQSW Questionnaire, in this case - from the moment it enters the Scanning queue right up until the data on the *Questionnaire 6* **document's** *Front* and *Back* **pages** have been verified, validated and exported.

Each *MQSW* task relies in one way or another on the structure and content of the Document Hierarchy. Certain tasks, however, are not ***rules-driven*** and therefore do not respond to rules and actions of a RuleSet Hierarchy. Most Scan tasks, for example, operate in response to settings that govern the physical scanner's performance as it responds to the task's processing cues.

Note, too, the one-to-one relationships between the levels of the Document Hierarchy (at the top) and the **Data** categories listed below four of the tasks. This correspondence stipulates how and when the *MQSW* application retrieves, evaluates and exports **data** about batches, documents, pages – and the **fields** on every ***source*** page…every *Front* or *Back* page, for example. For more about the capture of an application's **data**, see Page 14.

✓ Page 22 introduces you to the Taskmaster Application Wizard and to the steps you can take to set up your own *Taskmaster* application.

Taskmaster Application



Document Hierarchy

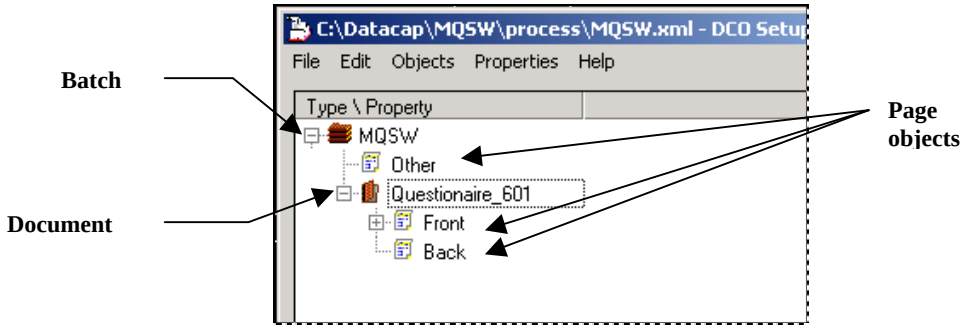
The Document Hierarchy is the backbone of a *Taskmaster* configuration:

- Its levels – **Batch**, **Document**, **Page** and **Field** - form a compact, reliable structure that is shared by all elements within the application.
- **Objects** at a particular level define a unique configuration and are the processing units for that level.
- An object's ***setup*** properties provide it with an identity, and define certain aspects of the object's role.
- An object's ***runtime*** properties and variables accumulate the data that is processed during task operations.

The Document Hierarchy is also a **"setup DCO"** file: **mqsw.xml** is in the **Process** directory of the **MQSW** application **folder**. This folder is a member of the configuration's **Datacap** directory (Page 19).

After installation of your **Datacap Taskmaster** configuration is complete and you have set up your own *Taskmaster* application, you can review the application's Document Hierarchy by taking the steps on Page 18.

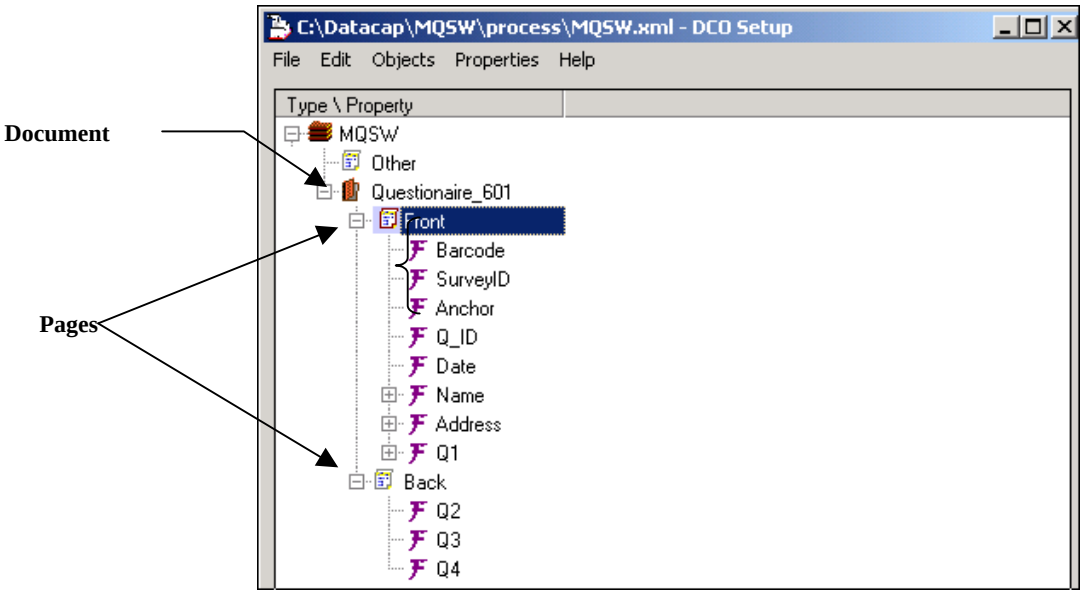
These steps will open the *MQSW* application's ***Document Hierarchy Setup*** window (in this example).



Document Hierarchy Setup Window – *MQSW*

In the example above, the window displays the hierarchy's **Batch** object (*MQSW*), one **Document** object (*Questionnaire_601*) and three **Page** objects: *Other* is a child of the **Batch** object, while *Front* and *Back* belong to the **Document** object.

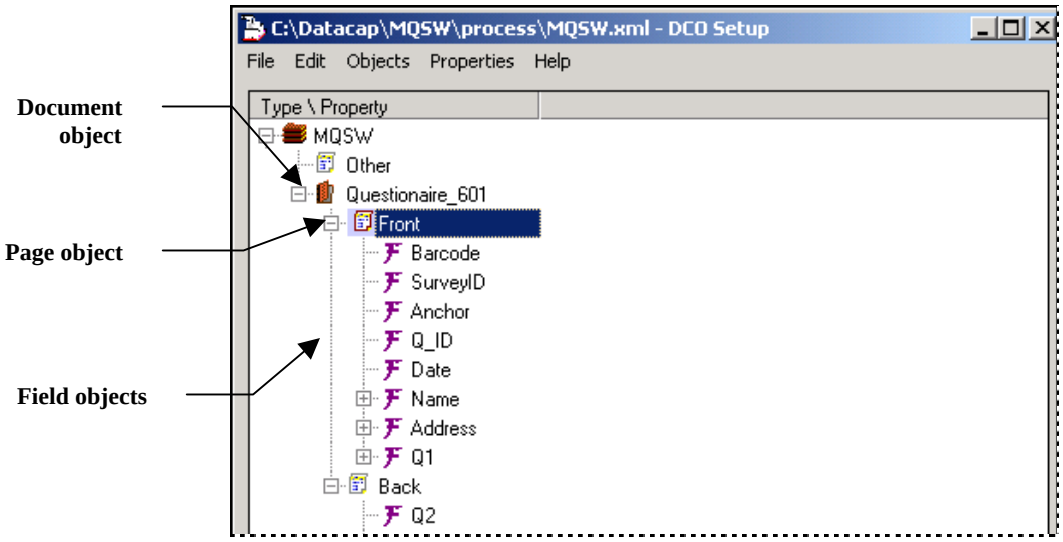
Clicking on the **+** icon next to a **Document** object lists its **Page** objects. These are the types of pages that the document can – or must – contain.



Document Hierarchy Setup Window - *MQSW*

In this example, the *Front* **Page** object represents one of the application's ***source*** pages – a page with *user-entered* data that is to be located, recognized, verified, validated and exported. (The *Back* page is the other ***source*** page.)

Field objects of the Document Hierarchy represent fields on a *Front* or *Back* page, and are children of the of the applicable **Page** object. To review the list of **Field** objects, click on the + icon next to the parent **Page** object. (A partial list is illustrated below.)

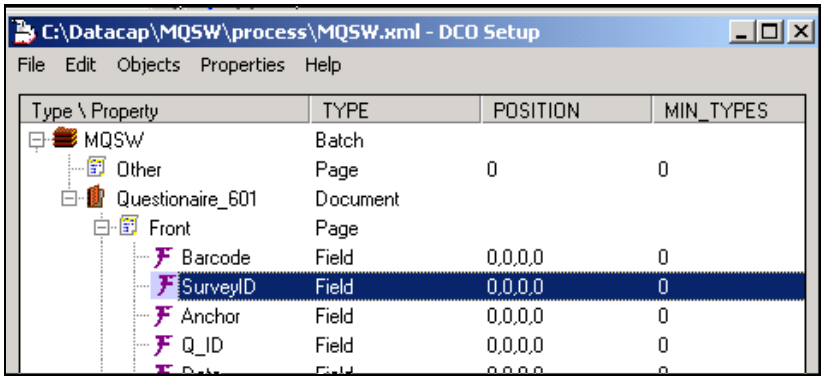


Document Hierarchy Setup Window - *MQSW*

Properties

A Document Hierarchy's objects have **properties**. Some are ***setup*** properties that govern the way an object will be processed; others are ***runtime*** properties with values assigned during processing. (***Runtime*** properties are often referred to as variables.)

You can review an object's key ***setup*** properties by highlighting the applicable Object ID and selecting **View All** from the **Properties** menu. (An object may have other properties, which are listed in the Document Hierarchy file (**MQSW.xml**, in this example.)



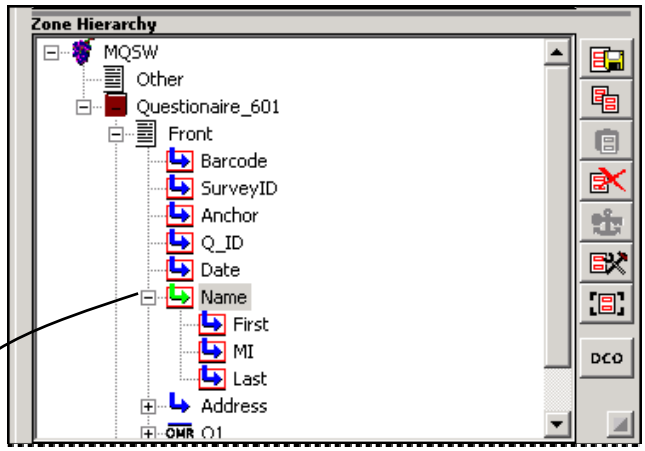
Document Hierarchy Setup Window – *MQSW*
Field Object Properties: SurveyID

The chart on Page 3 emphasizes the central position of the Document Hierarchy – and its links to other elements of a *Taskmaster* application's architecture. The paragraphs below review the Document Hierarchy's roles within an application, using *MQSW* as an example.

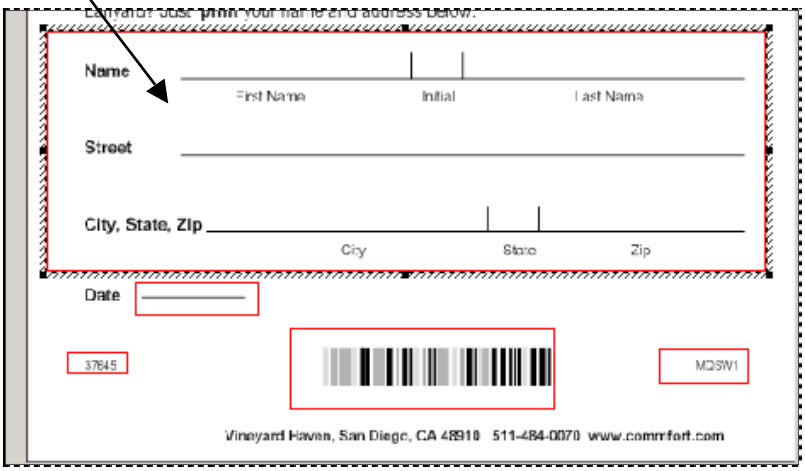
Rule Manager and the Document Hierarchy

Rule Manager procedures define the *MQSW* Document Hierarchy; objects at each level of the hierarchy appear prominently in both panels of *MQSW* ***Rule Manager Window***.

In the **Zone Hierarchy** display of the *Fingerprints* panel (illustrated on the next page), selection of a **Field** object initiates "zoning" procedures that link this object to a field of the *Front* or *Back* page. (Chapter 4 explains zoning procedures.)



MQSW Rule Manager Window – *Fingerprints & Zones Panel*
Zone Hierarchy Display



MQSW Fingerprint

RuleSet Hierarchies

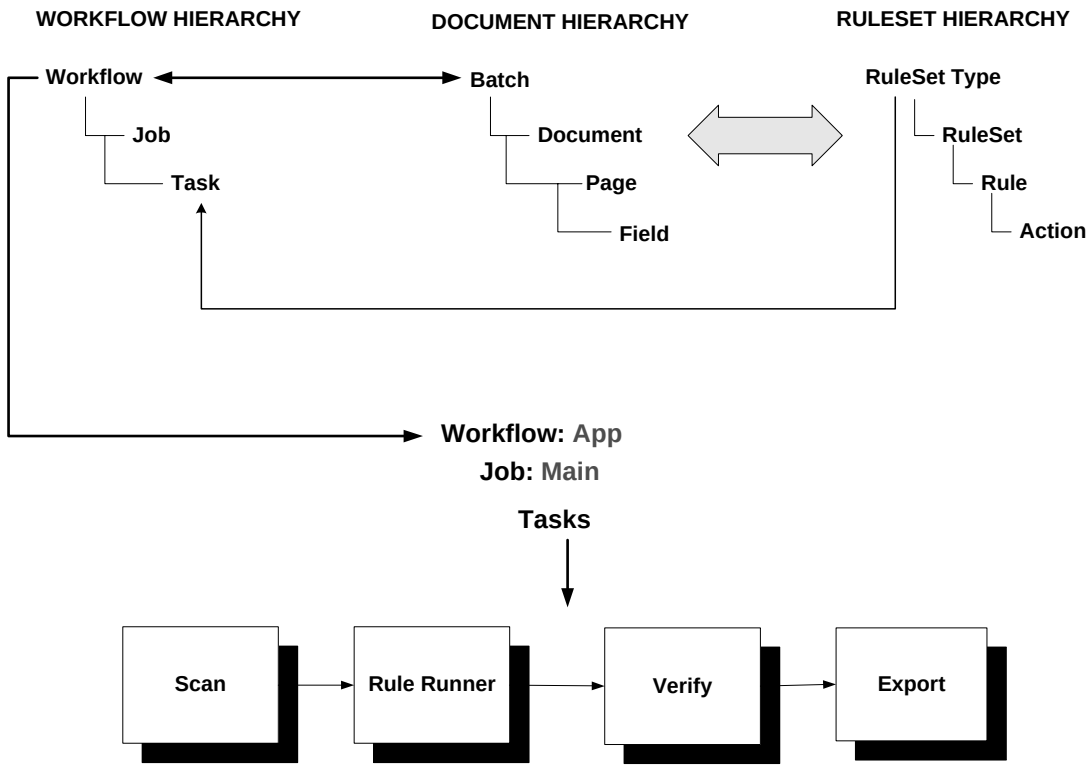
Most tasks operate in response to:

- **Rules** that belong to…
- **RuleSets** of a particular **RuleSet Type** that are bound to…
- **Objects** of the Document Hierarchy.

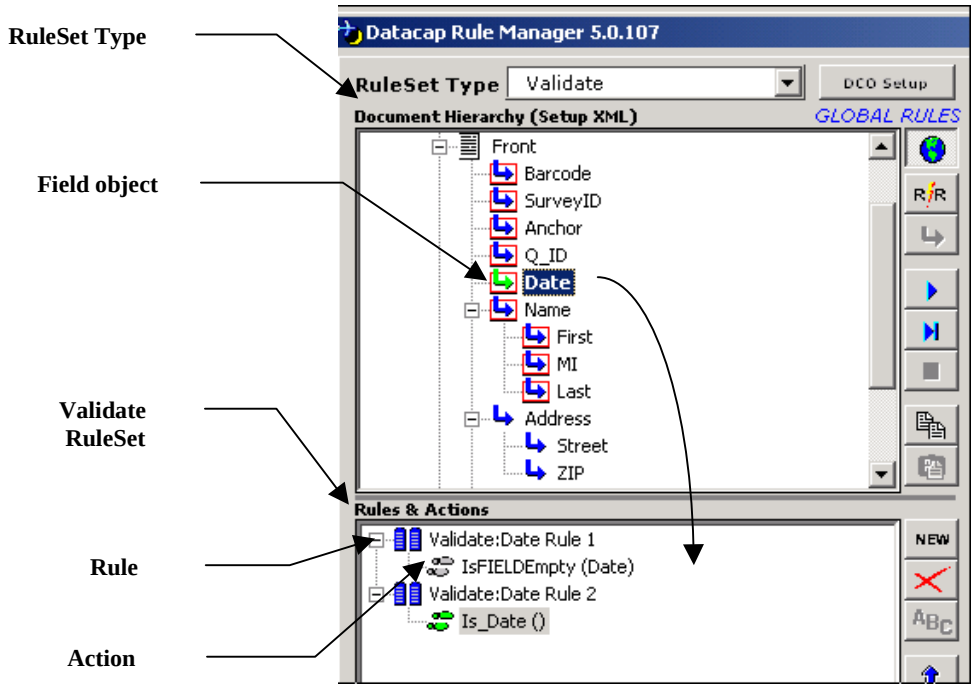
A rule, in turn, consists of one or more **actions**.

After a RuleSet has been constructed and linked to an object of the Document Hierarchy, it is assigned to one or more Task Definitions – and will be applied during workflow operations (Page 11).

Taskmaster Application



Look again at the formation of a simple RuleSet and its connection to the *Date* **Field** object of the *MQSW* Document Hierarchy. (For a closer look, you would go to the *Rules* panel of the *MQSW* ***Rule Manager Window***.)



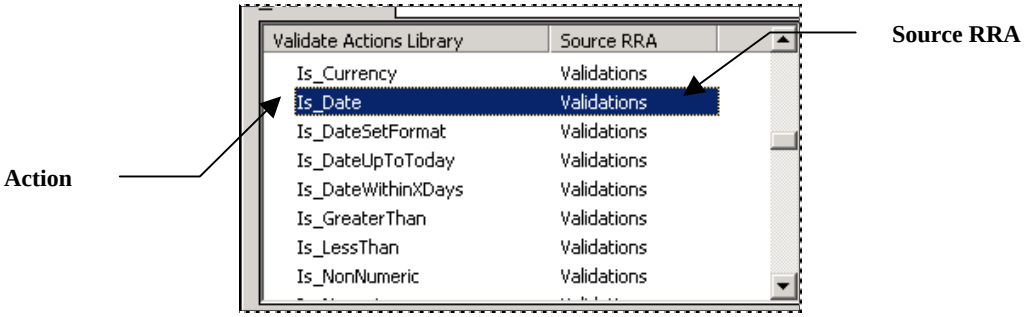
MQSW Rule Manager Window – *Rules panel*

At the top – in this example – *Validate* is the name of a **RuleSet Type**: rules of this type contain **actions** that check on an object's ***runtime*** value.

The *Date* **Field** object is the target of a **Validate** RuleSet that verifies the existence of a value in the **Date** field of a processed **Front** page, and the format of the value.

The **Rules & Actions** area lists the two rules that comprise the RuleSet – and their actions: `IsFIELDEmpty(Date)` and `Is_Date()`.

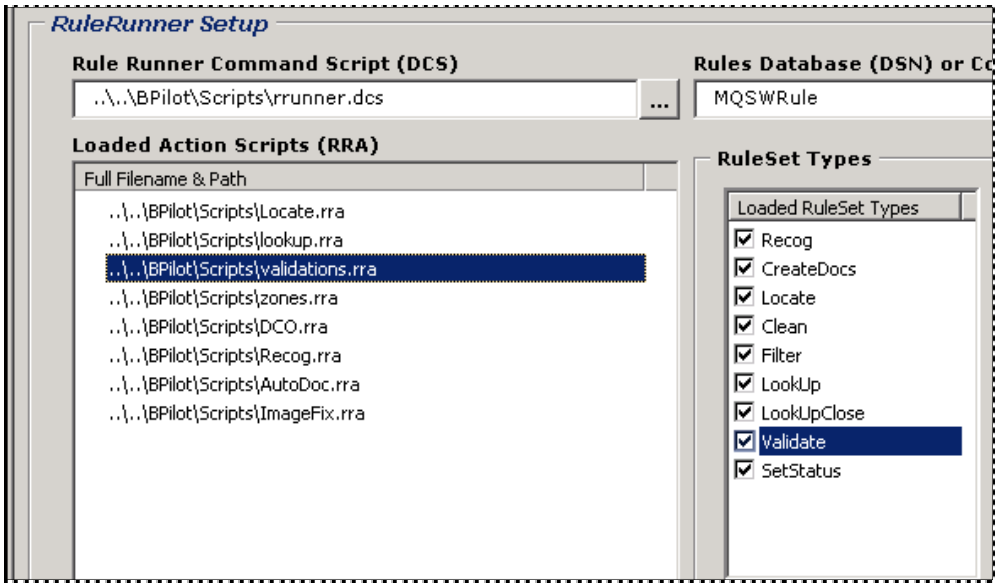
These actions were selected from a list of all the actions that can be included in rules of the **Validate** RuleSet Type:



MQSW Rule Manager Window – *Rules panel*

In this example, the script for the **Is_Date** action is part of the **validate.rra** file - its **R**ule**R**unner **A**ctions (.rra) file.

The Task Definition of the RuleRunner **task** identifies the **Validate** RuleSet Type as one of nine types with rules that the task will call upon as it processes the fields of a *Front* page, including the ***runtime*** values of the *Date* **Field** object:



Setup Parameters – *RuleRunner Task*

This illustration shows that the RuleRunner task has access to any RuleSet of the **Validate** RuleSet Type…and of any other RuleSet Type listed in the **Loaded RuleSet Types** field on the right.

The **Loaded RuleSet Types** field lists only those RuleSet Types with rules that the task will actually run – and the order in which the task will run them.

Therefore, the RuleRunner task will first apply **Recog** RuleSets to the appropriate objects of the *MQSW* Document Hierarchy, then the **CreateDocs** RuleSets. The task will finish by applying the **Validate** and **SetStatus** RuleSets.

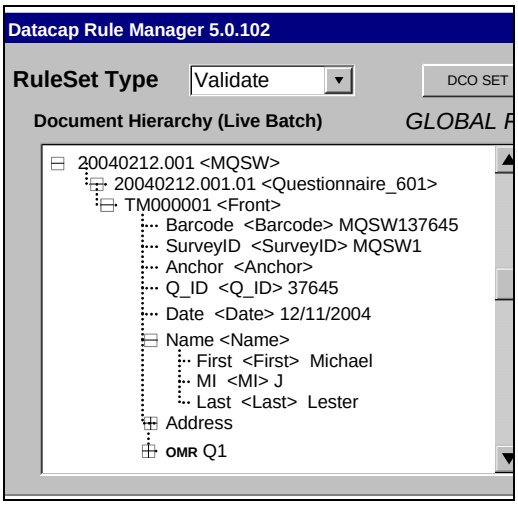
✓ Chapter 6 thoroughly explains how tasks that apply rules are set up.

The **Document Hierarchy** area in the *Rules* panel of the ***Document Hierarchy Window*** lists objects at each level, in two modes – ***setup*** and ***runtime***.

In ***setup*** mode, the panel displays **RuleSets** that have been defined and **bound** to individual objects. The example on Page 8 shows how validating RuleSet (**Validate**) consists of two **rules** that are bound to the *Date* **Field** object.

This area can also display objects of the Document Hierarchy with their ***runtime*** values…values assigned to objects when a **Live Batch** is processed (illustrated below)

✓ ***Very important!*** Chapter 3 examines all aspects of a Document Hierarchy –and shows you how to set up and modify the Document Hierarchy for your application. Chapter 4 describes the use of fingerprints; Chapter 5 thoroughly examines rules and their actions.



MQSW Rule Manager Window – *Rules Panel*
Document Hierarchy Display (Runtime)

✓ The *Rule Manager Reference* has complete explanations of *Rule Manager* and the ***Rule Manager Window***.

Workflow Hierarchies

The *Workflow* tab of an application's ***Taskmaster Administrator*** manages the application's **Workflow Hierarchies**.

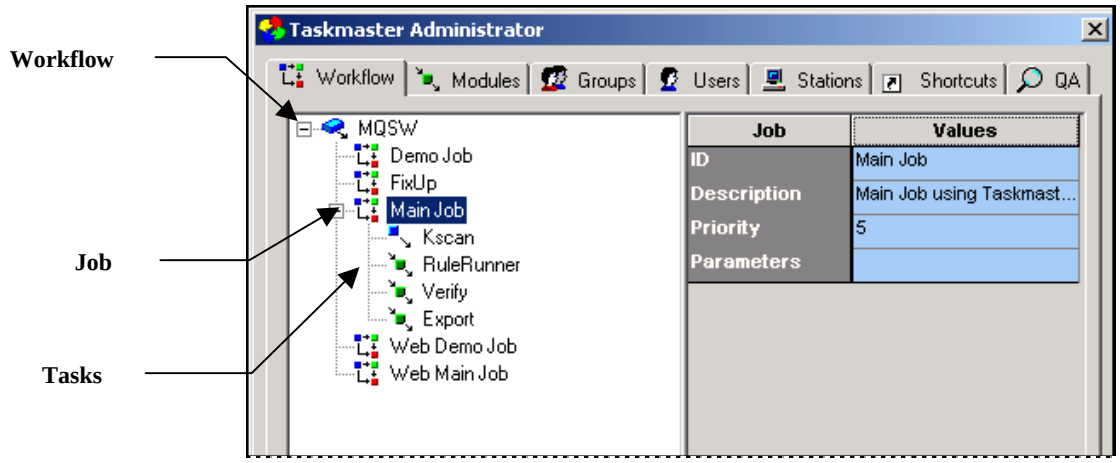
A Workflow Hierarchy has this structure:

Workflow

Job

Task

In the illustration below, *MQSW* identifies a **workflow**, and *Main Job* is one of five **jobs** that are part of the *MQSW* workflow. The *Main Job*, in turn, relies on four **tasks** to meet its processing objectives.



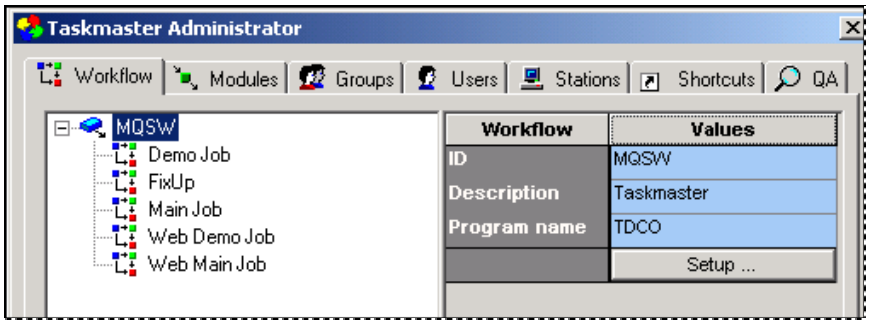
MQSW Taskmaster Administrator – *Workflow tab*

Workflows

A **workflow** component – *MQSW*, in this case - occupies the top level. If you highlight the Workflow ID, most of its properties and their values show up on the right-hand side (illustrated on the next page.)

The "TDCO" **Program Name** specifies that this hierarchy and its members refer to a Document Hierarchy for ***setup*** and ***runtime*** parameters.

Clicking on the Setup button accesses the application's Document Hierarchy – **MQSW.xml** (Page 3). This is a workflow's ***critical*** property.



Taskmaster Administrator – *Workflow tab*

Jobs

A typical *Taskmaster* workflow has multiple jobs. Using the *MQSW* workflow as an example:

Main Job is the application's workhorse. Its four tasks scan paper questionnaires; create images and add them to a processing batch; locate, recognize, verify and validate the data on each *Front* or *Back* ***source*** page; and export the data to a file or database.

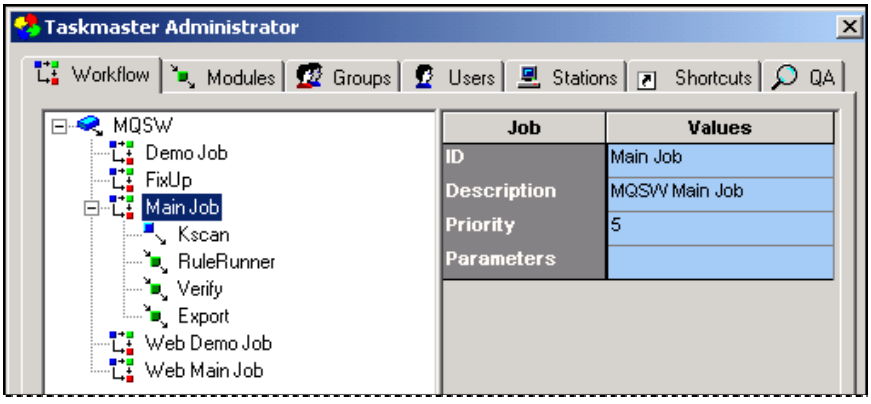
Demo scans images rather than paper, and is a powerful training tool.

FixUp corrects problems identified by the Main Job's RuleRunner or Verify task.

Web Job uses the Internet to scan and verify paper forms from remote locations.

Web Demo Job uses the Internet to scan and verify images from remote locations.

When you highlight a Job ID on the tab's left side, values of its properties occupy the right side.

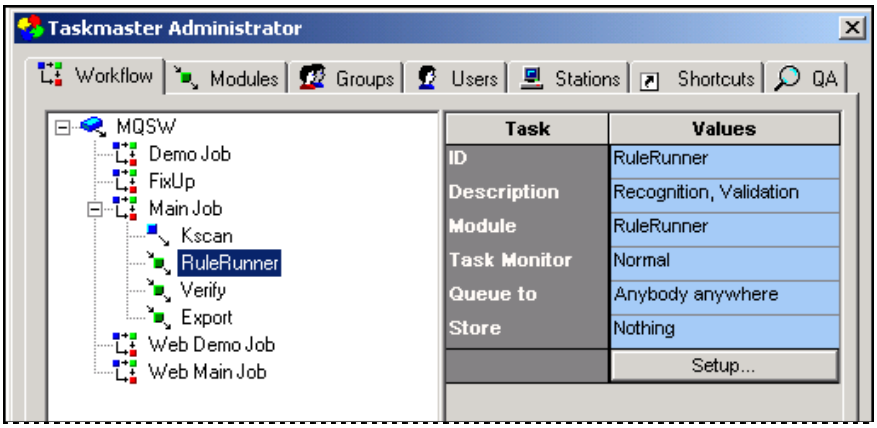


Properties of the MQSW Main Job

Tasks

A task is a set of highly focused procedures that help meet a job's overall objectives. Together, a task and the job to which it belongs form a **Job/Task Combination**: a Job/Task Combination is the ***principal operating entity*** of a *Taskmaster* application.

For example, a *RuleRunner* task is part of the *Main Job*; the resulting Job/Task Combination is referred to as `Main Job.RuleRunner`. When you highlight the task's ID on the left-hand side of the *Workflow* tab, the task's properties and their values appear on the right. Clicking on the Setup button accesses the task's ***Setup*** dialog (Page 9).



Properties of the RuleRunner Task

☛ The following chapters cover task setup and operations:

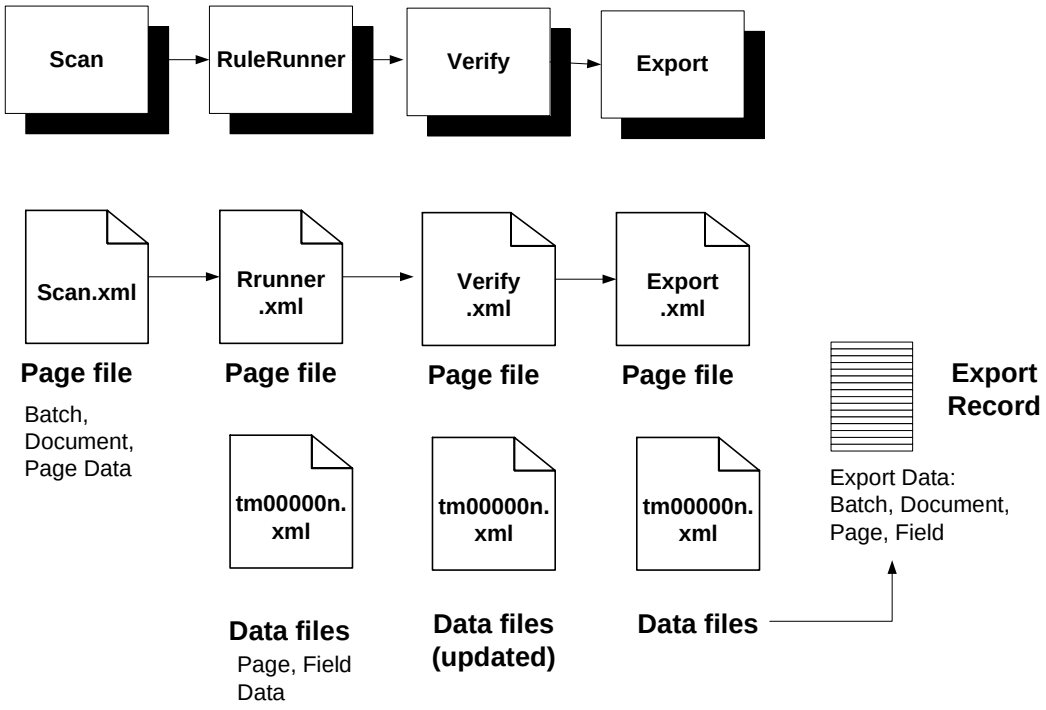
- Chapter 6: task components, settings and setup.
- Chapter 7: Scan tasks, including VScan tasks
- Chapter 8: *RuleRunner* tasks – tasks that operate exclusively in response to pre-defined rules and their actions.
- Chapter 9: Verify tasks and ***Data Entry*** panels.
- Chapter 10: FixUp tasks.

Application Data

The chart on Page 3 shows how the tasks involved in *Taskmaster* operations locate, process and store data linked to the four levels of the Document Hierarchy: **Batch**, **Document**, **Page** and **Field.** It does not, however, explain where the data comes from!

The charts below, and on the next few pages, provide a more detailed guide to the ***sources*** of application data; the procedural stages that recognize, review and validate data; and the ways in which data is used to assemble the closing Export Record.

A summary view of data gathering emphasizes the importance of two types of files in every batch: the **Page file** for every task, and the **Data file** for every ***source*** page.



Each task produces its own **Page file** (**<task>.xml**). The Page file lists the contents of the batch – its documents and pages – in a format that is appropriate for the task. The current task's Page file provides essential processing information to the next task in line.

In the *MQSW* example, the RuleRunner task also generates a **Data file** (**tm00000*n*.xml**) for each ***source*** page in the batch – each *Front* or *Back* page. Ultimately, the Data file contains recognized, verified and validated values for the fields in the page.

The Export task extracts data from its Page file (**export.xml**) and from the Data files in the batch when it puts together **Export Records** for individual ***source*** pages.

Page Files

A Page file accumulates information from objects at three levels of the Document Hierarchy (using the *MQSW* Document Hierarchy for examples):

- **Batch:** the **Batch** object's **Type** specification (*MQSW*); the unique identity of the current batch (*20050254.002*, for example); and any other data assigned to batch-level variables of the Document Hierarchy, and solicited by rules applied to the Document Hierarchy's **Batch** object.
- **Document:** the **Document** object's **Type** specification (*Questionaire_601*); the unique identity of the each document in the batch (*20050254.002.**01***, for example); the current document's processing **Status**; and any additional "Document Data" assigned to document-level variables of the Document Hierarchy and solicited by rules applied to the Document Hierarchy's **Document** object.
- **Page:** the **Page** object's **Type** specification (*Front, Back, Other*); the unique identity of the each page in the batch (TM000001, TM000002, etc.); the name of the Image file (.tif) representing a page; a hyperlink to a **Data file** if the page is a ***source*** page; and any additional data assigned to page-level variables of the Document Hierarchy and solicited by rules applied to this **Page** object of the Document Hierarchy.
- **Field:** the recognized, verified and, in some cases, validated values of fields in a ***source*** page's Data file (.xml). Although the Scan task creates the Data file, the RuleRunner task identifies individual fields in the ***source*** page, formats the file according to the structure of the Document Hierarchy, and assigns runtime values to individual pages. Later, a task such as Verifiy can modify the file's data.

The table below is a portion of a Page file generated by the RuleRunner task, after the task has processed a batch with one document and two ***source*** pages. (The steps that *MQSW* would take to locate and process data in a batch with 200 pages would be the same.)

RuleRunner.xml is a Page file with these "tags" (and sample values):

Tag	Sample Value	Description
B	**20040258.002**	Batch ID
TYPE:	**MQSW**	The value assigned to the **Batch** object of the Document Hierarchy.
ED:	**1**	The ***estimated*** document count entered by the task's operator in the ***StartBatch Panel***.
AD:	**1**	The RuleRunner task's ***actual*** document count.
EP:	**2**	The ***estimated*** page count entered by the task's operator in the ***StartBatch Panel***.

Page File Contents (continued)

Tag	Sample Value	Description
AP:	2	The RuleRunner task's ***actual*** page count.
STATUS:	0	The processing status of the batch. "0" indicates that the RuleRunner task has not discovered problems with makeup or the batch. "1" specifies a "problem" batch.
<variables>		Data assigned to additional batch-level variables that you specify.
D	**20040258.002.01**	The Document ID assigned to the ***single*** document containing all pages of the batch (in this example.)
TYPE:	Questionaire_601	The value of a **Document** object's **TYPE** property that the RuleRunner task has assigned to the current document.
STATUS:	0	The processing status of the batch. "0" indicates a problem-free document; "1" specifies a "problem" document.
P	**TM000001**	The Page ID of this document's first page.
TYPE:	Front	The TYPE property of the ***source*** **Page** object of the *MQSW* Document Hierarchy.
STATUS:	48	The processing status assigned to the page: "48" indicates *RecogDoneOK* .
IMAGEFILE:	tm000001.tif	The name of the Image file (.tif) associated with this page.
DATAFILE:	tm000001.xml	The name of the Image file (.tif) associated with this ***source*** page.
Confidence:	0.4953352	RuleRunner's assessment of its ability to recognize the page according to its Page Type – and, because this is ***source*** page, recognize the values in individual fields. "1" = Low Confidence; "9" = High Confidence.
Image_Offset:	0,0	Measurements indicating how far the page's image has been offset for clearer recognition.
TemplateID:	1010	The ID of the *MQSW* fingerprint that RuleRunner has matched to this ***source*** page. A **fingerprint** is a fully-configured representation of a *Front* or *Back* page that locates fields on the page, and applies rules to the page and its fields.

Page File Contents (continued)

Tag	Sample Value	Description
P	**TM000002**	The Page ID of this document's second page.
TYPE:	**Back**	The **TYPE** property assigned to this **Page** object of the MQSW Document Hierarchy.
STATUS:	**48**	The processing status assigned to the page: "74" indicates *NoData*.
IMAGEFILE:	**tm000002.tif**	The name of the Image file (.tif) associated with this page.
DATAFILE:	**tm000002.xml**	The Data file assigned to this page.
Confidence:	**0.738591**	RuleRunner's assessment of its ability to recognize the page according to its Page Type.
TemplateID:	**1011**	The Fingerprint ID assigned to pages that are ***not*** matched to an actual *MQSW* fingerprint.

Data Files

A Data file accumulates recognized and, ultimately, verified and validated values for the fields of a single *Front* ***source*** page.

In the example, **tm000001.xml** is a Data file with tags and values similar to those in the table below. ***Alert!*** This table lists only the first few **Field** objects that would appear in an actual *MQSW* Data file.

Tag	Sample Value	Description
P	**TM000001**	Batch ID
F	**Barcode**	The identity of the first **Field** object in the *MQSW* Document Hierarchy – and a discernable field at the top of the *Front* page..
Text Value -	**MSSW137645**	The field's *recognized* value.
Char conf -	**9999966999**	Confidence ratings of the field's recognized characters.
TYPE:	**Barcode**	The **TYPE** property assigned to this **Field** object of the MQSW Document Hierarchy.
Position:	**0,0,0,0**	The RuleRunner task's determination of the field's location. The "0,0,0,0" coordinates indicate that the task did not attempt to locate this field on the current page.
STATUS:	0	The processing status of the field.

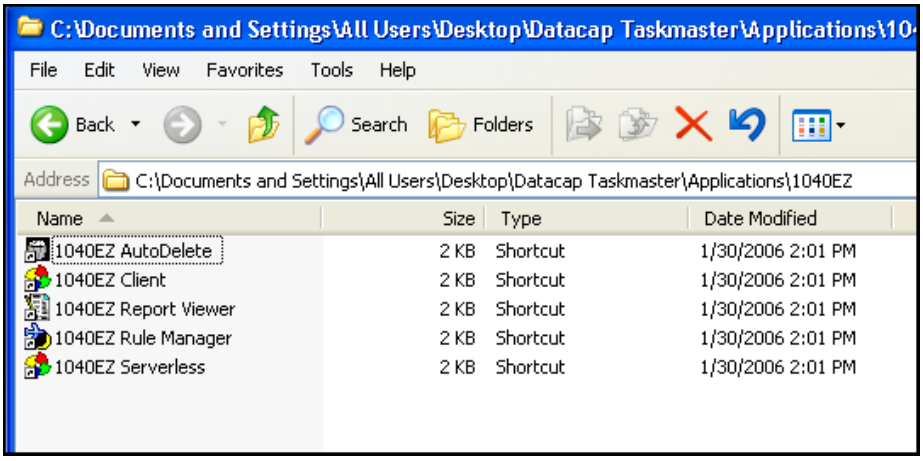
Pre-Configured Applications

This section uses the *1040EZ* training application for examples as it shows you how to locate and review an application's components.

✓ *1040EZ* is a compact, fully-configured application that you can access and explore without difficulty. Although the *Taskmaster* application ***you*** assemble will have a different focus, its underlying structure and elements will closely resemble those of *1040EZ.*

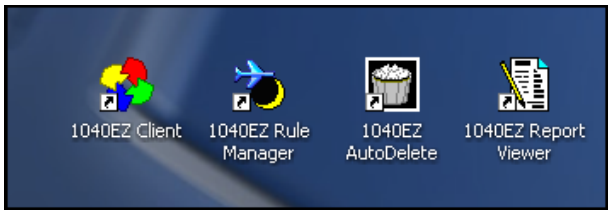
Application Shortcuts

Datacap Taskmaster Installation places a **Datacap Taskmaster** folder on your computer's desktop. The **Applications** sub-folder has a folder for each pre-configured application, including *1040EZ.* And the sub-folder has these shortcuts:

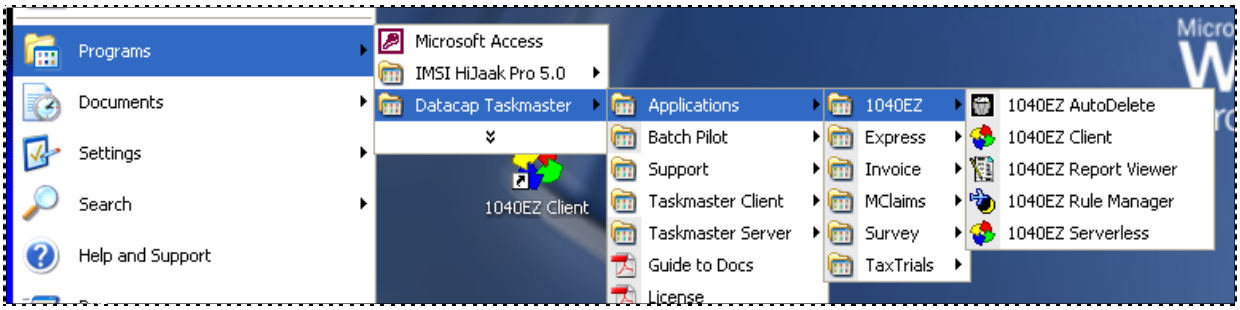


Shortcuts of the 1040EZ Application

For simplicity, we recommend that you "send" the four shortcuts illustrated below to your desktop:



You can also reach these shortcuts by selecting **Datacap Taskmaster** from the Windows Start button's list of **Programs**, and **1040EZ** from the **Applications** (illustrated on the next page.)

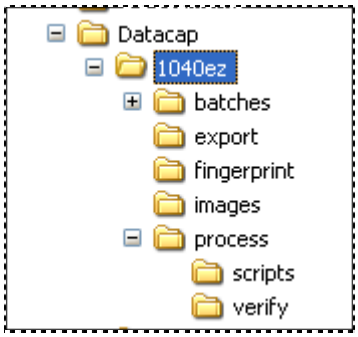


✓ To access a *Taskmaster* application:

- Double-click on its "client" icon…**1040EZ Clien**t, in this case.
- Enter your UserID and Password in the ***Login*** dialog.
- Press the dialog's OK button.

Application Folders and Files

Each application – including a new application you set up – has a folder that belongs to your configuration's overall **Datacap** directory. An application folder (**1040EZ**, for example) has with five sub-folders. The **Process** folder, in turn, has two sub-folders.



In this group…

batches will have a sub-folder for each of the application's processing batches.

export may have an Export file for each batch (some applications export data to a database instead.)

images stores images of ***source*** pages and of other pages such as *Attachments* or *Document Separator* pages. Images of the ***source*** pages are the basis for the definition of an application's fingerprints. In addition, a VScan task typically processes the Images files in this directory (Chapter 7).

Process holds the application's databases (.mdb), Document Hierarchy (.xml), Task Projects (.bpp), and Settings files (.ini). This folder has two sub-folders:

- **scripts** is the site for any special-purpose scripts developed for the application.

- **verify** will hold the ***Data Entry*** panel used by operators of the application's Verification tasks, and a file with the panel's keyboard shortcuts.

The **dco.xsl** file is a "style sheet" that determines the structure and appearance of the Document Hierarchy.

Files of the Application's Process Directory

The *1040EZ* applixation's **Process** directory holds:

The Document Hierarchy file (**1040ez.xml**).

- The Document Hierarchy defines an application's structure at four levels: **Batch**, **Document**, **Page** and **Field**.

Batch Pilot's Task Project files (.bpp).

- These files contain software and settings to power and guide tasks.

Settings files (.ini).

- Some contain parameters for tasks such as vScan (**VScan.ini**); specifications to govern a physical scanner and the scanning process (**scanner.ini**); criteria for *Report Viewer's* report generation (**rptview.ini**); and specifications that determine *Rule Manager's* focus and scope (**RuleMan.ini**).

Internet Settings files (.icp) for the tasks of an application that runs in the *Taskmaster Web* environment.

The Access databases of a *Taskmaster* application.

These include (using the *1040EZ* application for examples):

- **1040Adm.mdb:** stores definitions of all aspects of the application's Workflow Hierarchies, as well as all its security criteria.
- **1040Eng.mdb**: contains processing details of every batch - its documents pages, fields and values.
- **1040Export.mdb:** stores the exported data for each batch (Page **Error! Bookmark not defined.**).
- **1040Look.mdb**: the look-up database that is used to validate data in selected fields.
- **1040Rule.mdb:** assembles and maintains information about the application's fingerprints and their zones, and about RuleSets, rules and actions.
- **rptview.mdb:** contains data used to format and generate reports.

Each task generates a Page file after it processes a batch. The **taskname.xml** file provides a task with the structure of its Page file.

Components of the Pre-Configured 1040EZ Application

In addition to folders and files located firmly within the **Datacap** directory, the *1040EZ* application consists of:

- The *1040EZ* Document Hierarchy with objects at four levels: **Batch**, **Document**, **Page** and **Field**.
- A Workflow Hierarchy consisting a workflow, its jobs, and the tasks assigned to each job.
- Task Projects for all tasks that need them.
- Task Modules that link Task Definitions to their Task Projects.
- A Task Definition for each task.
- Job-Task shortcut icons all set to launch individual Job/ Task Combinations.
- Application Security credentials that determine who can carry out administrative and operating procedures, from which stations.
- Administrative, processing and rules databases.
- A *Rule Manager* dedicated to the application.
- RuleSet Types with their Actions libraries.
- Fingerprints of the application's ***source*** page(s).
- The Verify task's ***Data Entry*** panel.

New Applications

This section briefly introduces you to the Taskmaster Application Wizard - and very briefly describes the wizard's alternative approaches to setting up a *Taskmaster Rulemanager* application.

✓ ***Important!*** The Application Wizard is accompanied by a text – the *Application Wizard Guide* - that fully describes the wizard's purpose, screens and results.

The Taskmaster Application Wizard

Screens of the Taskmaster Application Wizard guide you through the steps you take to define and install a completely new *Taskmaster* application, or reproduce an existing application.

The illustration on the next page highlights the two alternative approaches identified as options on the wizard's *Mode* screen:

Approach 1: Create a New Application launches a sequence of screens that

- Assemble the basic structure of a new application;
- Help you as you define the application's Document Hierarchy;
- Provide the application with a fully-configured set of jobs and tasks that include *Taskmaster Web* capabilities;
- Determine the controls and layout of the application's ***Data Entry*** panel;
- Supply the application with Security components and parameters;
- Allow you to run tasks of the application's Demo job without delay.

Approach 2: Copy an Existing Application moves almost instantly to

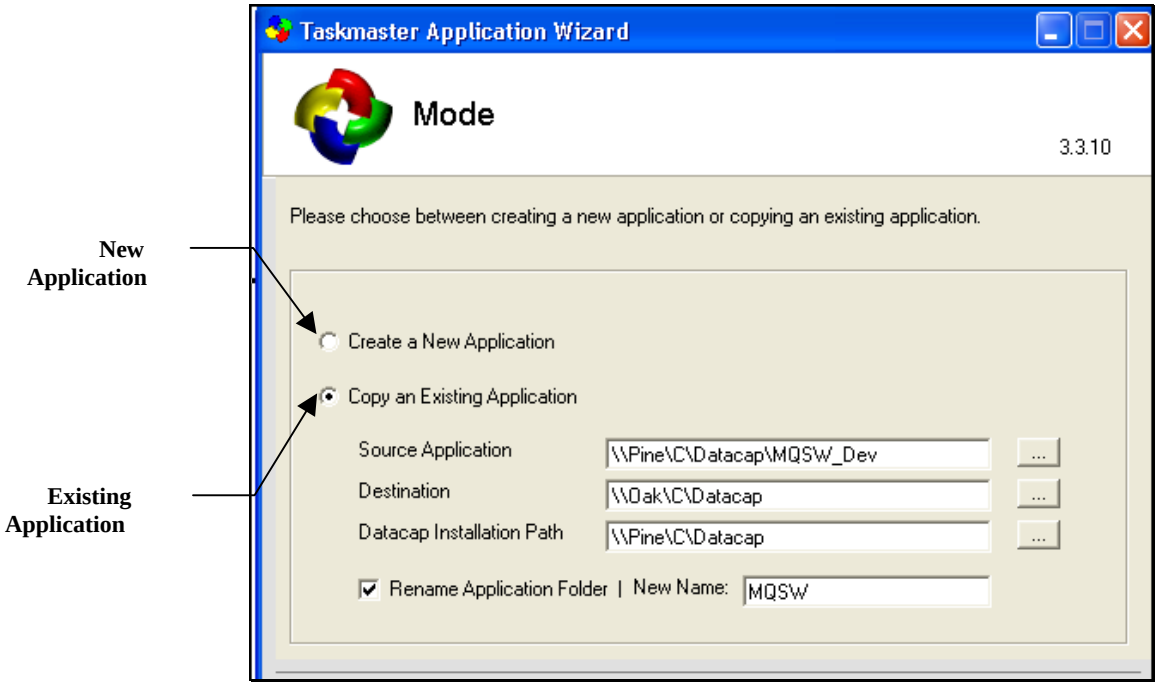
- Duplicate the *Taskmaster* application you specify;
- Place the application in any location you designate;

- "Point" the application to an existing **Datacap** directory on a File Server you select;
- Give the application a new name;
- Update paths in settings and parameters throughout application.

✓ Because the Taskmaster Application Wizard's procedures differ according to the option you select, separate sections of the *Application Wizard Guide* cover each approach.

Both procedures result in a new application with the following unique components:

- A Document Hierarchy that defines the application's processing and recognition structure in terms of its batches, documents, pages and fields.
- Fingerprints of the application's ***source*** pages…pages with fields containing data that is to be recognized, verified, validated and exported.
- Rules that determine how individual tasks process documents and their pages, and fields and their data.
- A Verification task's ***Data Entry*** panel.
- Workflow Hierarchies consisting of a workflow and its jobs, and the tasks assigned to each job.
- Default Security parameters that include User ID's, Station ID's, etc.
- A *Report Viewer* that generates reports on task operations



Taskmaster Application Wizard – *Mode Screen*