

---

## Verify Tasks and Data Entry Panels

Chapter 9 describes the setup and operation of a Main job's Verify task. The chapter also examines:

- The design, development and modification of a Verify task's *Data Entry* panel;
- Techniques that help a Data Entry operator move easily, efficiently and productively through the panel's fields;
- Ways in which the Data Entry operator can move easily, efficiently and productively from one "problem" page or document to the next until the Verify task has completely processed the current batch.

Chapter 9 covers these topics:

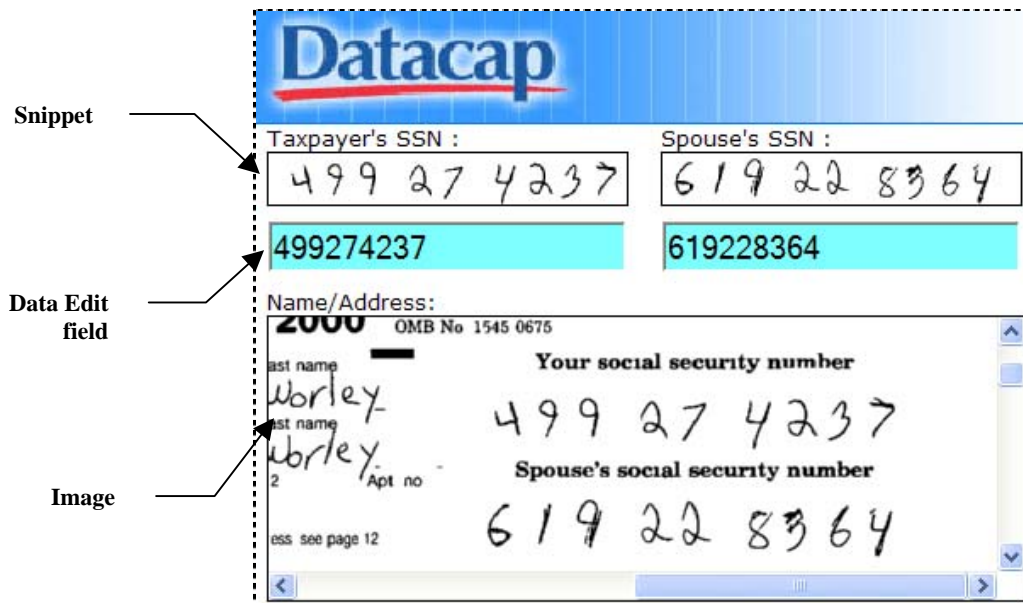
<b>I. Introduction .....</b>	<b>9-2</b>
<b>II. Components of a Verify Task.....</b>	<b>9-6</b>
<b>A. Configuration Components .....</b>	<b>9-7</b>
<b>B. Operating Components.....</b>	<b>9-8</b>
<b>III. Verify Task Setup .....</b>	<b>9-10</b>
<b>A. Prerequisites .....</b>	<b>9-10</b>
<b>B. Verify Task Setup – Task Project.....</b>	<b>9-11</b>
i. How to Construct a Data Entry Panel.....	9-14
a. Panel Controls and Properties.....	9-18
b. Tab Order Properties.....	9-22
<b>C. Verify Task Setup – Task Module.....</b>	<b>9-23</b>
<b>D. Verify Task Setup – Task Identity .....</b>	<b>9-25</b>
<b>E. Verify Task Setup – Task Criteria.....</b>	<b>9-27</b>
<b>F. Verify Task Setup – Task Settings .....</b>	<b>9-29</b>
<b>IV. Verify Task Operations .....</b>	<b>9-32</b>
<b>A. Data Entry Tools .....</b>	<b>9-36</b>
i. Menus and Toolbar.....	9-36
ii. Validate Return Button.....	9-40
<b>B. Hot Keys .....</b>	<b>9-40</b>
<b>V. How to Add Fields to a Data Entry Panel.....</b>	<b>9-44</b>
<b>A. How to Remove a Field from a Data Entry Panel .....</b>	<b>9-47</b>

## Introduction

A typical Verify task has a number of special features:

- The task is operator-intensive: it is launched and run by a qualified and authorized Data Entry operator.
- The operator uses a **Data Entry** panel to review and, if necessary, correct the *recognized* data of a *source* page.
- The operator can also use this **Data Entry** panel to assess the organization of the current batch into a series of documents – each with the correct type and number of pages.
- Optionally, a Verify task can validate *recognized* and *verified* data in specific fields of a *source* page.

Here is a very small portion of the 1040EZ application's **Data Entry** panel:



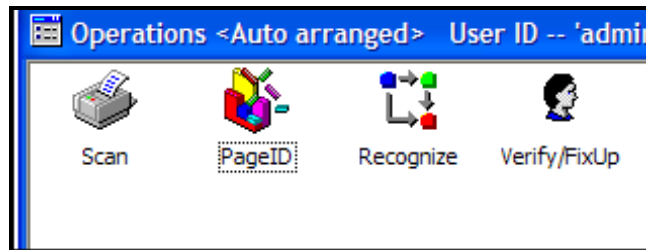
1040EZ Data Entry Panel

- ✓ 1040EZ is an indestructible training application! To access and experiment with this application's **Data Entry** panel, take the following steps:

Step	Action
1.	Be sure your Taskmaster Server is up and running.
2.	Select <b>Datacap Taskmaster</b> from the <b>Programs</b> options of your Windows Start button.

To Access the 1040EZ Data Entry Panel (continued)

Step	Action
3.	Open the <b>Applications</b> folder and the <b>1040EZ</b> sub-folder. Double-click on the <b>1040EZ Client</b> icon, and enter your administrative <b>User ID</b> , <b>Password</b> and <b>Station ID</b> in the <i>Login</i> dialog.
4.	When the <i>1040EZ Operations</i> window appears, double-click on the <b>Scan</b> icon. Without delay, the application's vScan task will create a new batch and place a set of Image files in the batch.



5. Press the No button when *Taskmaster* asks if you would like to continue scanning.
6. Double-click on the **PageID** icon. After this task runs, double-click on the **Recognize** icon to initiate recognition of data in the *source* pages of the batch...these are the pages with tax data.
7. When Recognition concludes, double-click on the **Verify/FixUp** icon to launch the application's Verify task and open the *1040EZ Data Entry* panel.

The Verify task's *Data Entry* panel presents an operator with data from the fields of a *source* page in the three formats highlighted in the example on the previous page:

- ◆ A **Snippet** displays the **image** of a field's value as it appears on the scanned page.
- ◆ An interactive **Data Edit** field displays the field's **recognized** value. If the recognizing task –a task such as Recognize - was unsure about its ability to recognize all characters correctly, the field's background color will change from blue to yellow.
- ◆ Images of part of the page, or of the entire page, are instantly available to the Data Entry operator for further clarification.

The operator can also check the organization of the batch into a series of documents and pages by reviewing the **Batch View** area at the bottom of the *Data Entry* panel (see the illustration on the next page.)

## Introduction

ID	TYPE	STATUS
20050005.002		
20050005.002.01	Document	0
TM000001	Page_1040ez	0
20050005.002.02	Document	1
TM000002	Page_1040ez	1
20050005.002.03	Document	1

**1040EZ Data Entry Panel - Batch View area**

Below is a very different example of a Verify task's *Data Entry* panel – different because it is a panel *immediately* after the *Autoform* utility has constructed it. The panel is in its *design* state: field pairs (snippets and Data Edit fields) have the names of their **Field** objects in the Document Hierarchy, and lack images and values.

Panel – design state

Page object - Document Hierarchy

**MQSW Data Entry panel  
Questionnaire 601 - Front Page**

- ✓ However, this early stage of panel development has significant advantages. As soon as you have fully defined your Document Hierarchy (Chapter 3) and have zoned the fields of a **Page** object on one or more fingerprints (Chapter 4), you can direct *Autoform* to set up a **Data Entry** panel for that page.

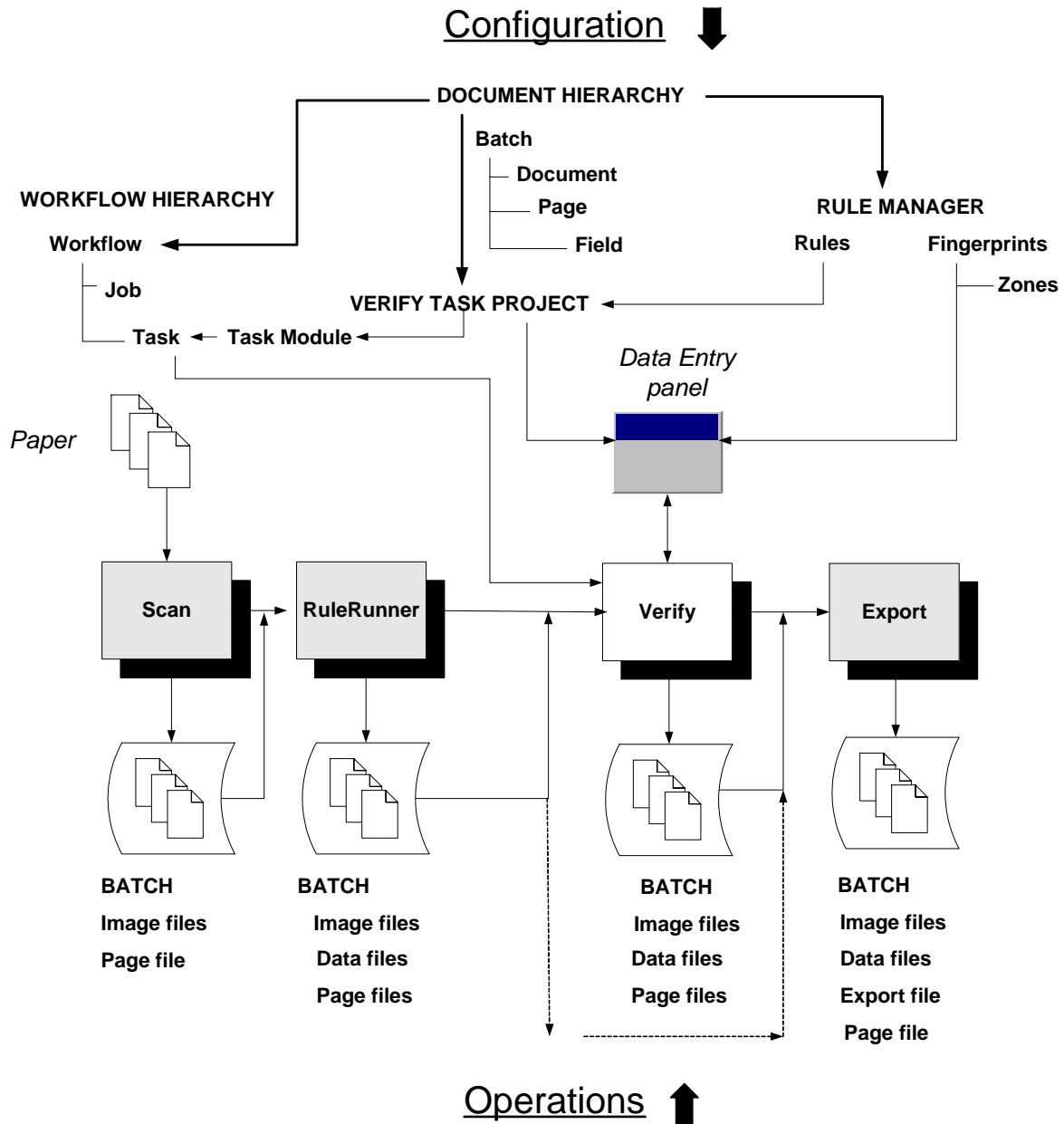
The utility will respond without delay, leaving you free to re-arrange the panel's controls to fit your needs and those of your operators. You are free, as well, to review and modify techniques an operator uses to handles Verify task operations as he or she processes the documents and pages in a batch.

Chapter 9 begins with a brief review of the Verify task's components (Page 6). It then explores:

- Fingerprints and zoning (Page 7)
- *Autoform* generation of **Data Entry** panels (Page 14)
- Controls of the **Data Entry** panel (Page 18)
- Verify Task Definition (Page 10)
  - Task Project (Page 11)
  - Task Module (Page 23)
  - Task Identity (Page 25)
  - Task Setup (Page 29)
- Verify Task Operations (Page 32)
- How to add and remove fields to existing panels (Page 44)

## Components of a Verify Task

This chart depicts the components of a Verify task's configuration at the top, its operating components at the bottom, and the task itself in the middle.



## Configuration Components

An application's **Document Hierarchy** is the core component of task configuration *and* operations - and must be fully assembled before you begin to define a Verify task.

- ✓ Chapter 3 provides complete explanations of the Document Hierarchy's structure, objects and properties – and of its contributions to all aspects of a *Taskmaster* application.

The illustration on the left below shows a portion of the Document Hierarchy of the pre-configured but indestructible *1040EZ* application. For a closer look:

- ◆ Open the application by following the steps on Page 2.
- ◆ Select **Workflow** from the **Settings** menu of the *Taskmaster Window*.
- ◆ In the *Workflow* tab of the *Task Master Administrator*, highlight the *1040EZ* Workflow ID on the left, and press the Setup button on the right.

Type \ Property	TYPE
1040EZ	Batch
Document	Document
DocumentSeparator	Page
Page_1040ez	Page
Anchor1	Field
Anchor2	Field
TaxpayerName	Field
SpouseName	Field
Address	Field
City	Field
State	Field
Zip	Field
TaxpayerSSN	Field
SpouseSSN	Field

**1040EZ Document Hierarchy**

Type \ Property	TYPE
MQSW	Batch
Other	Page
Questionnaire_601	Document
Front	Page
SurveyIDfr	Field
Anchor	Field
Q_ID	Field
Q1	Field
Q2	Field
Q3	Field
Q4	Field
Back	Page
SurveyIDbck	Field

**MQSW Document Hierarchy**

Above, on the right, is the Document Hierarchy of the fictional *MQSW* application, introduced on Page 4.

Parameters of the Document Hierarchy feed into – and are **required** by – nearly every other configuration component, including:

**Verify Task Project.** A Task Project provides setup forms and dialogs used during the Task Definition process – as well as the form that is the task's **Data Entry** panel. The Task Project also contains the software that powers the Verify task. Chapter 6 explains how Task Projects are assembled in the *Batch Pilot* workshop; Page 11 examines the Task Project of a Verify Task.

The definition of a Task Project begins with the assignment of a Document Hierarchy.

**Fingerprints and Zones.** The configuration of the task's *Data Entry* panel relies on **zoning** of the fields within a **fingerprint** that accurately represents a *source* page such as the *Front* page of the *MQSW* questionnaire illustrated on 4. The fingerprint itself corresponds to a **Page** object of the Document Hierarchy; each zoned field corresponds to a **Field** object. Chapter 4 explains all aspects of fingerprints and zones.

**Rules.** Rules that determine how a Verify task processes a batch and its contents are “bound” to specific objects of the Document Hierarchy...to the **Batch** object, and to **Document**, **Page** and **Field** objects. Chapter 5 examines the formulation of RuleSets, and rules; Page 27 explores the RuleSets of a sample Verify task, and shows you how they are added to the Task Project, and assigned to the task when the task is defined.

**Workflow Hierarchy.** A Verify task is a member of a Workflow Hierarchy that consists of a single **workflow** at the first level; one or more **jobs** at the second level, and the **tasks** assigned to each job at the third level. The application's Document Hierarchy is an initial and required property of the workflow: the *Document* Hierarchy and its parameters are inherited by all members of the *Workflow* Hierarchy. For more about Workflow Hierarchies, see Chapter 6; Page 23 looks at the addition of a Verify task to the *MQSW* workflow's Main job.

**Data Entry Panel.** A typical *Data Entry* panel is a form occupied principally by field-pair controls...snippets and Data Edit fields that correspond to the **Field** objects of the Document Hierarchy. Page 14 shows you how to generate the form and assign it to the task's Task Project. Page 18 describes steps you can take to modify the form's layout, and to add or remove controls.

The **Task Module** links the Verify task you define as part of the Workflow Hierarchy to the Task Project – and it is the only Configuration component that does not have a direct association with the Document Hierarchy. Page 23 describes the setup and assignment of a Task Module for the Verify task.

## Operating Components

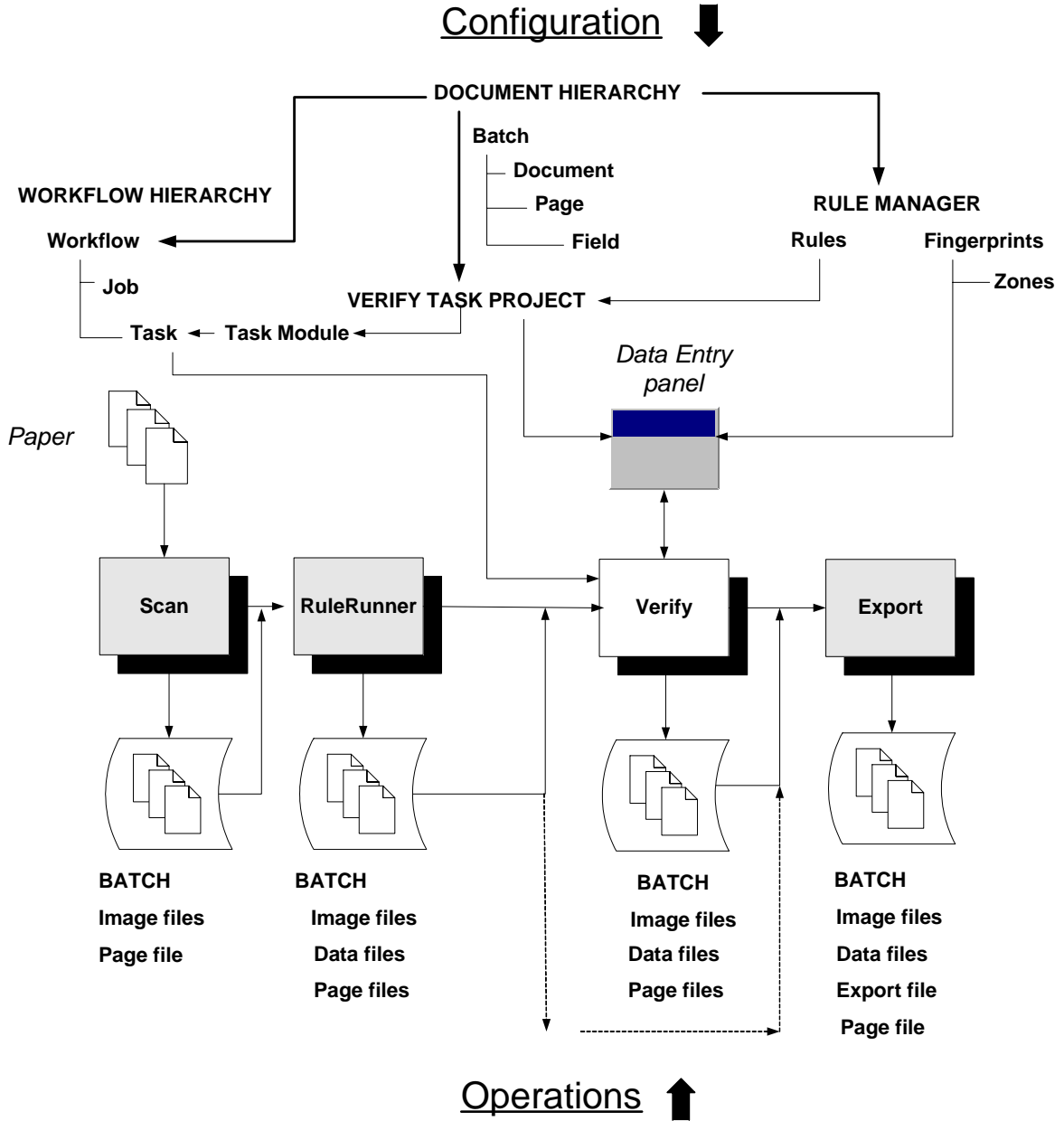
The chart's lower section highlights these components of a Verify task's operations:

**Job/Task Combination.** The Verify **task** belongs to a **job** – typically, a Main or Demo job. Together, they form a Job/Task Combination – *Main.Verify*, for example. Job/Task Combinations are *Taskmaster's* principal processing entities. When an operator launches a Verify task, she's actually running the applicable Job/Task Combination.

**RuleRunner Batch.** The RuleRunner task's critically important recognition procedures (Chapter 8) clearly identify the *source* pages in the batch; locate the **fields** on a *source* page; recognize their **data**; and add the *recognized* values to a **Data file** (.xml) that the task generates for the page. RuleRunner also generates a **Page file** (.xml) with details of the batch contents. These files, along with the Image files (.tif)



already in the batch, supply the Verify task with the data and images that occupy the field pairs of the *Data Entry* panel and, in some cases, with additional data that is not subject to verification.



**Task Processing.** Most (but not all!) Verify task operations take place within a *Data Entry* panel that rests inside the *Batch Pilot Window*. The panel displays data and images drawn from a *source* page; the window lists the contents of the batch as a series of documents and pages. The operator uses the panel’s tools to move from field to field within a page– and the window’s tools to move from page to page, and document to document - until verification of the batch is complete.

## Verify Task Setup

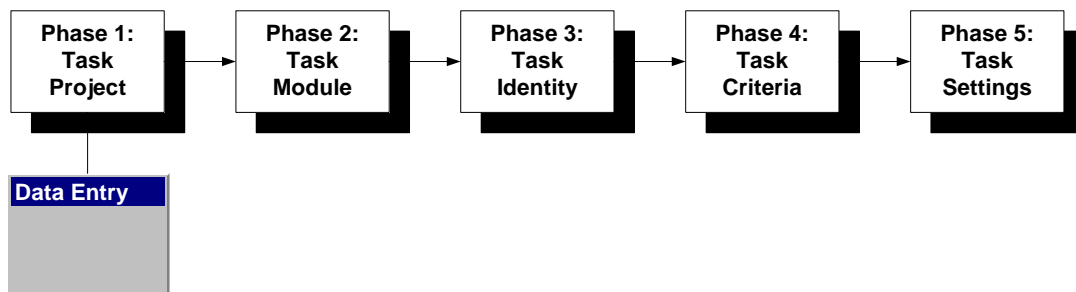
---

An operator's changes to the values in a page update the Data file for that page. Changes to the content or structure of the batch are reflected in the Page file the task generates for the batch.

- ✓ The section that begins on Page 32 introduces you to a Verify task's operations.

## Verify Task Setup

Setting up a Verify task follows standard procedures described in Chapter 6, with this important addition: in Phase 1, after you construct the Task Project, you'll assemble the task's *Data Entry* panel:

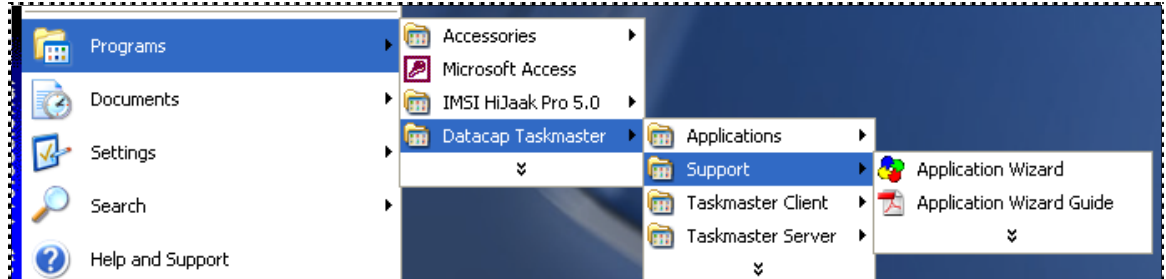


## Prerequisites

Successful setup of a Verify task can take place *only* if you have previously:

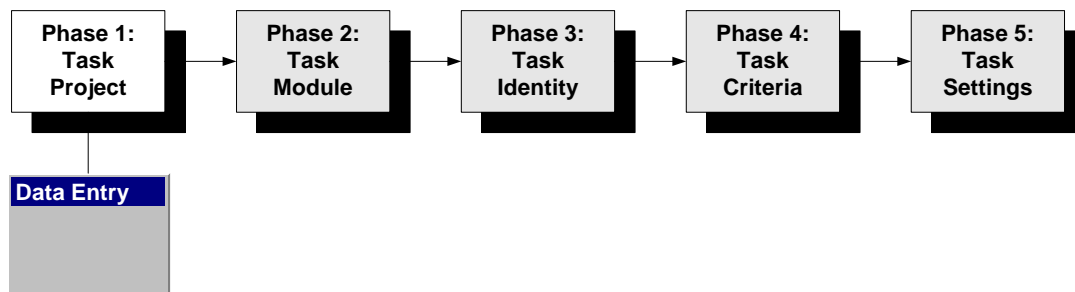
- ◆ Assembled a complete Document Hierarchy as a property of the workflow component of your application's Workflow Hierarchy (see Page 7 and Chapter 3).
  - ◆ Saved the Document Hierarchy as an XML file in your application's **Process** directory.
  - ◆ Constructed a fingerprint for the application's *source* page (Chapter 4).
  - ◆ Used *Rule Manager's* zoning procedures to link fields on the fingerprint to **Field** objects of the Document Hierarchy (Chapter 4).
  - ◆ Assigned special properties and Recognition criteria to *zoned* fields (Chapter 4).
  - ◆ Defined the rules that determine how the Verify task is process data that appears in the fields of the task's *Data Entry* panel.
- ✓ **Important!** This chapter does *not* re-examine the steps you take to meet these requirements. If appropriate, review Chapters 3, 4 and 5 - and use the *1040EZ* training application to experiment with Document Hierarchies and fingerprints - *before* you begin to define a Verify task.
  - ☛ When you use the Taskmaster Application Wizard to construct a new application, the wizard helps you define a Document Hierarchy and a sample fingerprint with zoned fields. The wizard then uses the *Autoform* utility to automatically generate the new

application's *Data Entry* panel. The *Application Wizard Guide* explains each step. To access the PDF version of this guide, select **Programs** from your computer's Start button options, and **Datacap Taskmaster**. Open the **Support** folder and double-click on **Application Wizard Guide**:



### Verify Task Setup – Task Project

Because the **Verify Task Project** is the point-of-convergence for the task's many elements and attributes, Phase 1 builds this component:



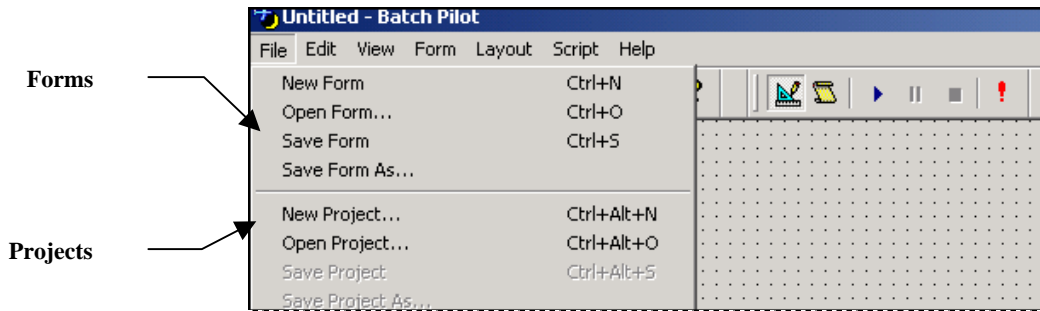
First, a Task Project is a file – **Verify.bpp**, for example – that you construct in the *Batch Pilot* workshop and place in your application's **Process** directory (Chapter 6). The file contains details of the Task Project's connections to

- ◆ The application's Document Hierarchy.
- ◆ The Task Definition's Workflow Hierarchy.
- ◆ The Task Definition's *Setup* parameters.
- ◆ The Task Definition's *Task Settings*.
- ◆ The Verify task's *Data Entry* panel.

The file also contains the code responsible for much of a Verify task's definition, and for its operations.

- ✓ To access *Batch Pilot*, select **Datacap Taskmaster** from the **Programs** options of your Windows Start button. Open the **Batch Pilot** folder and double-click on the **Batch Pilot** icon to open the *Batch Pilot Window*.

## Verify Task Setup

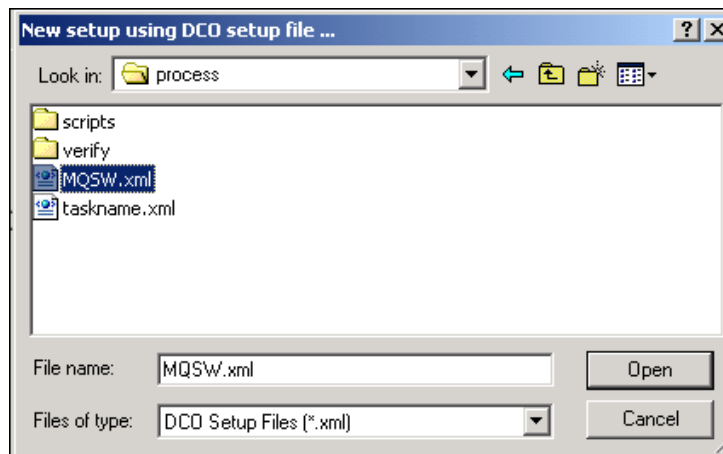


**Batch Pilot Window – File menu**

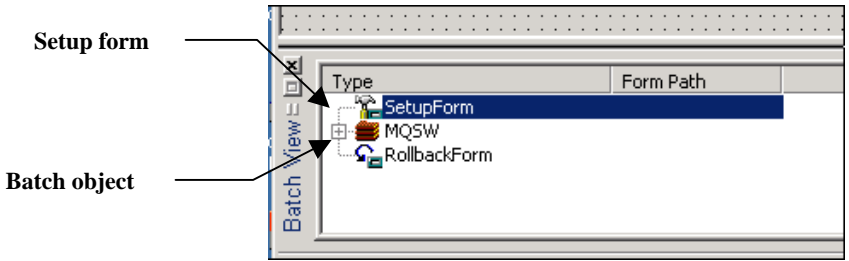
The opening segments of Phase 1 will use the **Project** and **Form** items of the *Batch Pilot Window's* **File** menu. The *Guide to Batch Pilot* – and *Batch Pilot Help* – explain all menus and toolbars.

### To Assemble a Verify Task Project

- | Step | Action  |
|------|---|
| 1.   | Be sure that the <b>Workflow</b> component of the <i>Taskmaster</i> Workflow Hierarchy that will contain the task includes a Document Hierarchy file ( <b>mqsw.xml</b> , for example).  |
| 2.   | Select <b>New Project</b> from the <i>Batch Pilot Window's</i> <b>File</b> menu. <i>Batch Pilot</i> will instantly ask you to enter the name and path of the Document Hierarchy file in the <i>Open File</i> dialog. Use the window's <b>Form</b> menu to be sure the project is <i>not</i> in <b>Design</b> mode.) |



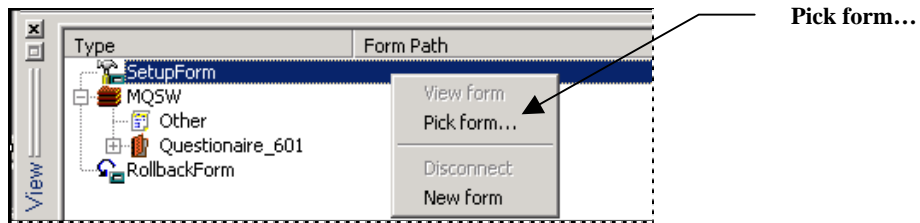
3. Select the application's Document Hierarchy file (.xml) from your application's **Process** directory, and click on the Open button to return to the *Batch Pilot Window*.
4. Confirm that the **Batch View** area at the bottom of the window displays a **Setup** form item in the **Type** column, as well as the **Batch** object of the Document Hierarchy you've specified (*MQSW* in the example below.)



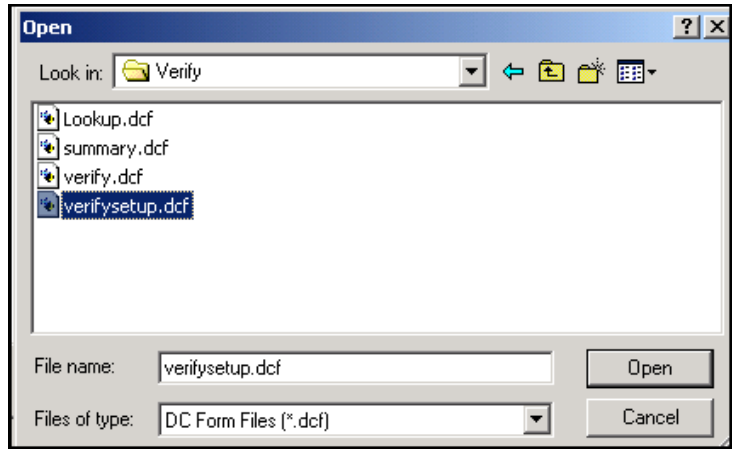
**To Assemble a Verify Task Project (continued)**

Step	Action
------	--------

- Highlight the *SetupForm* listing and right-click in the **FormPath** column.. Select the **Pick form...** option.

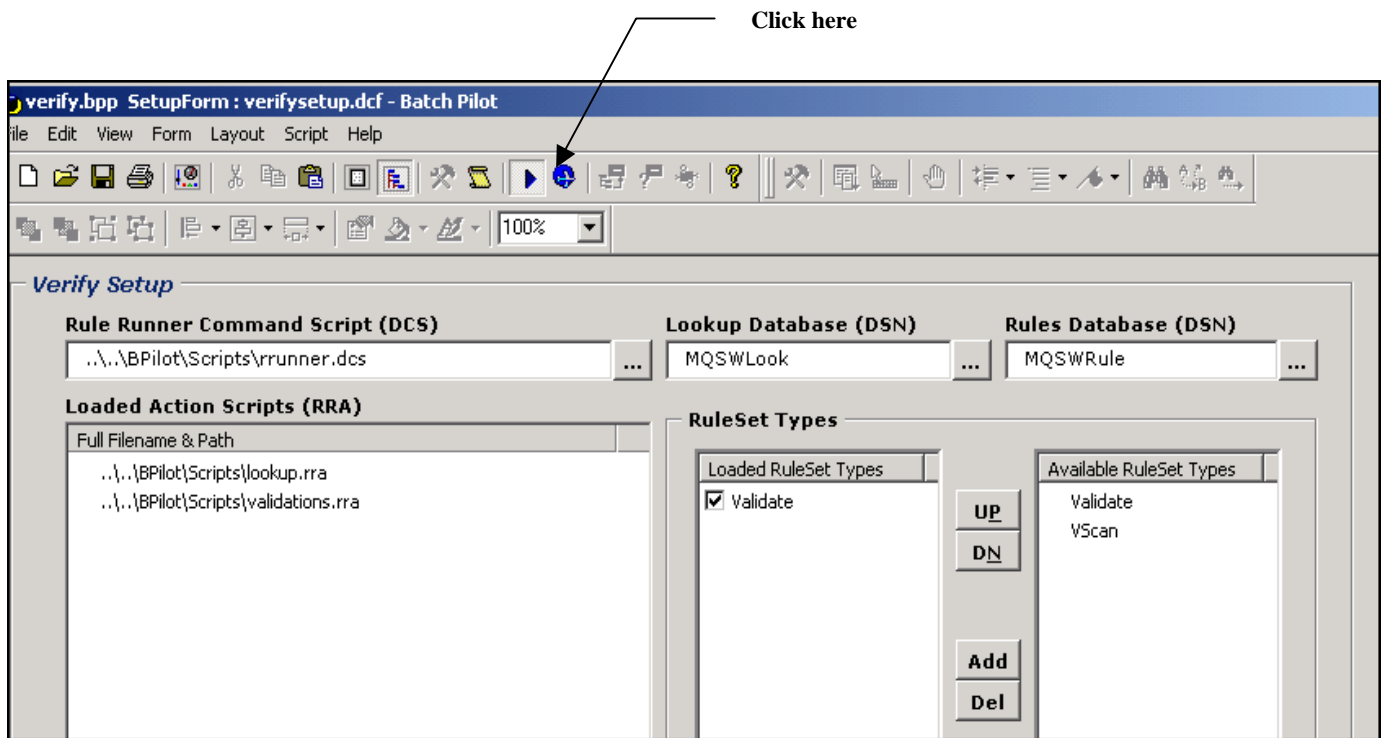


- Use the *Open File* dialog to navigate to the **Datacap** directory's **BPilot** folder.
- Select **iverifysetup.dcf** from the **Verify** folder and press the dialog's **Open** button.



- Select **Save** from the **File** menu to save this Task Project and make it instantly available to the Verify Task Definition (Page 25).
- Press the window **Run Setup** toolbar icon (▶) at the top of the window. *Batch Pilot* will show how the *Setup* dialog will look in Phase 4, and will display certain *default* settings for this Task Project's Verify task (see the illustration on the next page.)

## Verify Task Setup

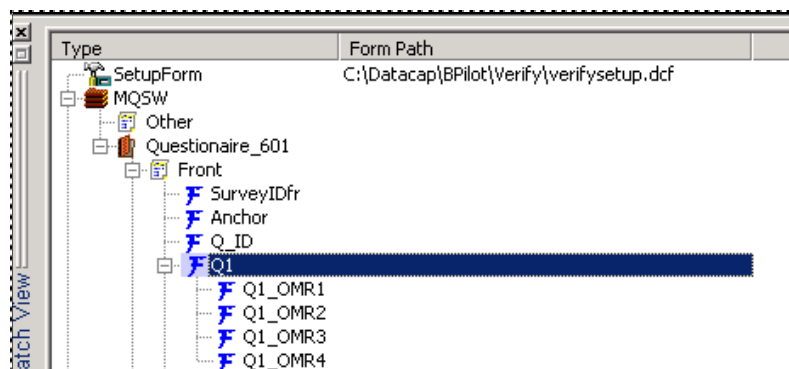


Batch Pilot Window – Verify Task Setup form

### How to Construct a Data Entry Panel

A Task Project *requires* a Document Hierarchy (Page 6).

After the Verify Task Project's structure is in place, select **Setup Tree** from the window's **View** menu. The Document Hierarchy will appear in the **Batch View** area in the bottom left-hand corner:

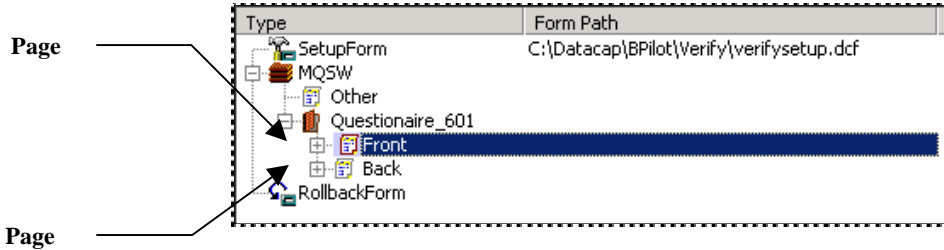


Batch Pilot Window - Batch View area

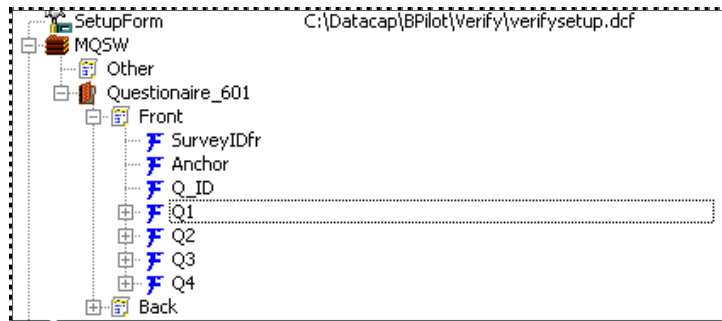
- ✓ The **Batch Pilot Window** makes it easy to move, size and re-size its members, the dialogs you place within its Data Area, and the controls you add to a dialog.

Take a close look at the levels of the *MQSW* hierarchy (as an example):

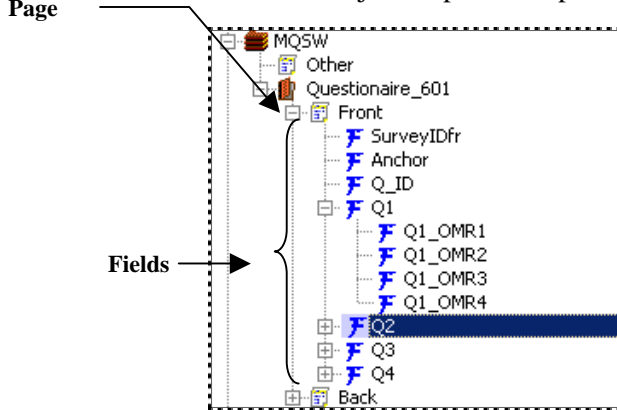
- At this early stage, no forms are bound to any objects of the Document Hierarchy.
- *Questionnaire\_601* is a **Document** object with two **Page** objects: *Front* and *Back*.



- The *Front* page has seven **Field** objects:



- Four **Field** objects represent **Optical Mark Recognition (OMR)** fields.



- *Rule Manager's Recognition Options Setup* dialog (Chapter 4) has established each OMR field – and has assigned values of a pre-defined **Dictionary** to its checkbox options (*Q1\_OMR1* – *Q1\_OMR4*.)

☛ **Very important!** The *Autoform* utility that builds the panel solicits information about the structure, contents and properties of a **Page** object *only*, as it prepares the **Data Entry** panel (see the next page).

In the example above, *Autoform* will put together a panel made up of seven **field pairs** – snippets and Data Edit fields - for the *Front* page. The Data Edit members of the OMR

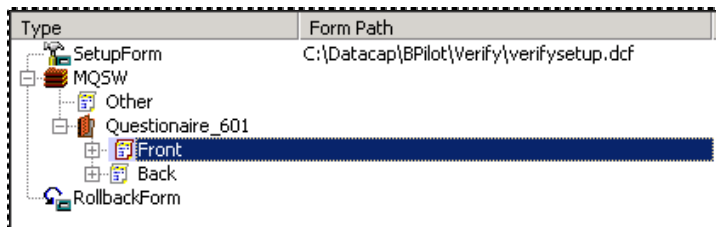
## Verify Task Setup

field pairs will be drop-down lists containing the four values of the dictionary: *Excellent*, *Good*, *Fair* and *Poor* – along with a *Blank* option.

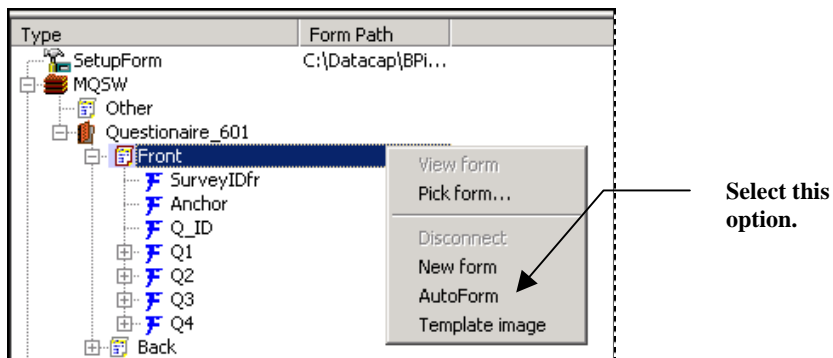
To put together a **Data Entry** panel, take these steps (using *MQSW* for examples):

Step	Action
------	--------

1. Open the Verify Task Project (.bpp) you previously defined (Page 11).
2. Select **Setup Tree** from the *Batch Pilot Window's* **View** menu.
3. Open the Task Project's Document Hierarchy.
4. Highlight the **Page** object with fields that are to be part of the **Data Entry** panel (*Front*, in the example below).

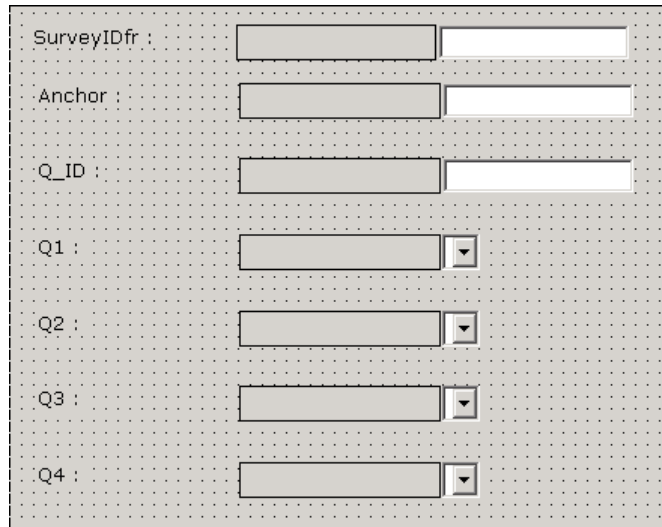


5. Right-click on the highlighted listing.



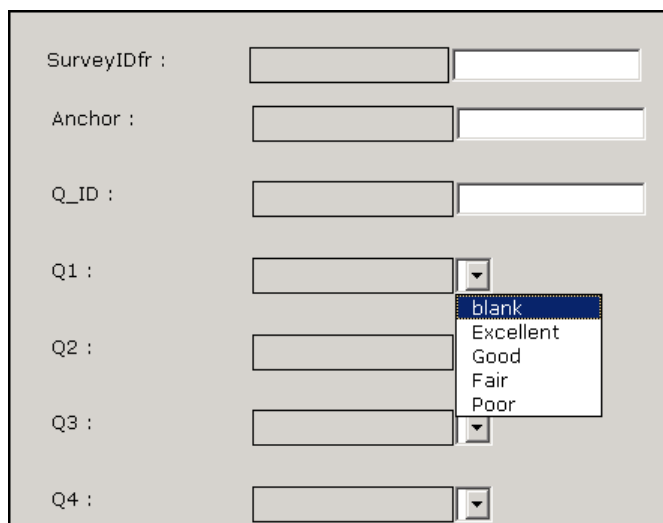
6. Select **Autoform** from the options.
7. Wait just a moment while Autoform constructs the panel and places it in the *Batch Pilot Window's* Data Area in its *Design* mode (illustrated on the next page.)
8. Un-toggle the **View** menu's **Design** option for a look at the new **Data Entry** panel in its *Operating* mode. (In this example, clicking on a drop-down list – *Q1*, for example – reveals the field's five options.)
9. The **Data Entry** panel is a new Datacap form (**dcf**). Use the **File** menu's Save item to save the form to the **Verify** folder of your application's **Process** directory.
10. Close the Verify Task Project.





The image shows a data entry panel in design mode. It features a dotted background. The panel contains six rows of controls. The first row is labeled 'SurveyIDfr' and has two text input fields. The second row is labeled 'Anchor' and also has two text input fields. The third row is labeled 'Q\_ID' and has two text input fields. The fourth, fifth, and sixth rows are labeled 'Q1', 'Q2', and 'Q3' respectively, and each has a text input field followed by a dropdown arrow. The seventh row is labeled 'Q4' and has a text input field followed by a dropdown arrow.

Front Page Data Entry Panel – *Design mode*



The image shows the same data entry panel in operating mode. The background is a solid light gray. The labels 'SurveyIDfr', 'Anchor', 'Q\_ID', 'Q1', 'Q2', 'Q3', and 'Q4' are on the left. The input fields are on the right. The 'Q1' dropdown menu is open, showing a list of options: 'blank', 'Excellent', 'Good', 'Fair', and 'Poor'. The 'blank' option is highlighted with a blue background.

Front Page Data Entry Panel – *Operating mode*

- ✓ The typical **Data Entry** panel needs more attention before an operator can use it effectively. In the example above:

The labels do not clearly identify fields on a questionnaire.

The snippets do not yet display images of values in their fields.

It is difficult to discern how effectively the Data Edit fields will display their data.

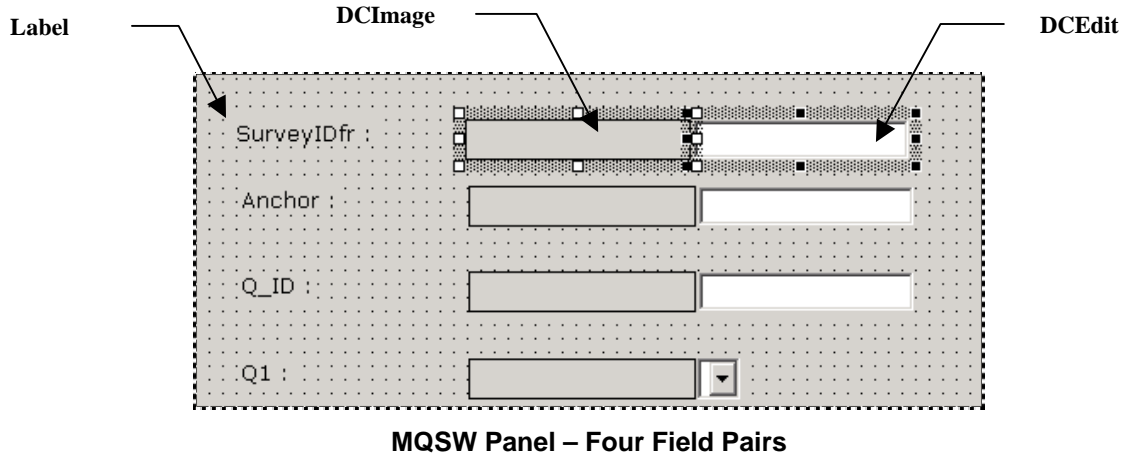
The panel lacks an obvious tool to display the full image of the current *Front* page – the **source** page that is currently being processed by the Verify task.

However, most of your efforts will involve reviewing key properties of the panel's controls to be sure they have been set up accurately.

## Verify Task Setup

To begin, access your new *Data Entry* panel in *Design* mode:

- ◆ In the *Batch Pilot* workshop, open the Verify Task Project (.bpp);
- ◆ Select **Setup Tree** from the *Batch Pilot Window's View* menu.
- ◆ Right-click on the **Page** object that is the basis for the panel.
- ◆ Select **View form** from the options.



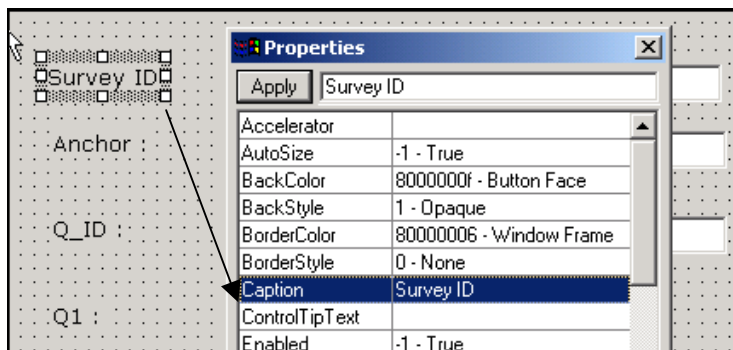
### Panel Controls and Properties

A *Data Entry* panel consists of controls.

The illustration above shows three controls for each of four **Field** objects of the Document Hierarchy. *Autoform* generates the controls; assigns their properties; and provides them with code.

Each trio consists of a **label** control, a **DCImage** control, and a **DCEdit** control. To view the properties of a control, right click on it and select **Properties** from the list of options.

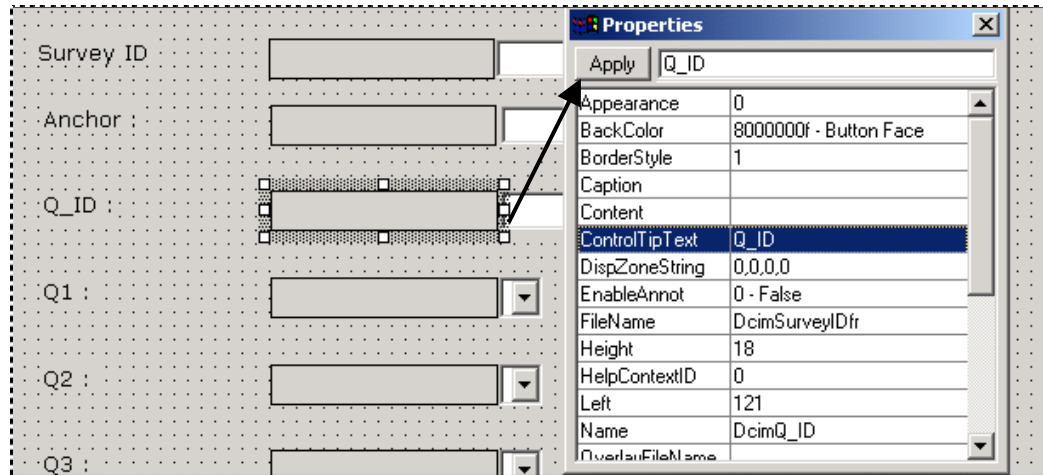
To enhance the panel's clarity, you may want to start by modifying one or more values that *Autoform* has assigned to the **Caption** properties of the **Label** controls.



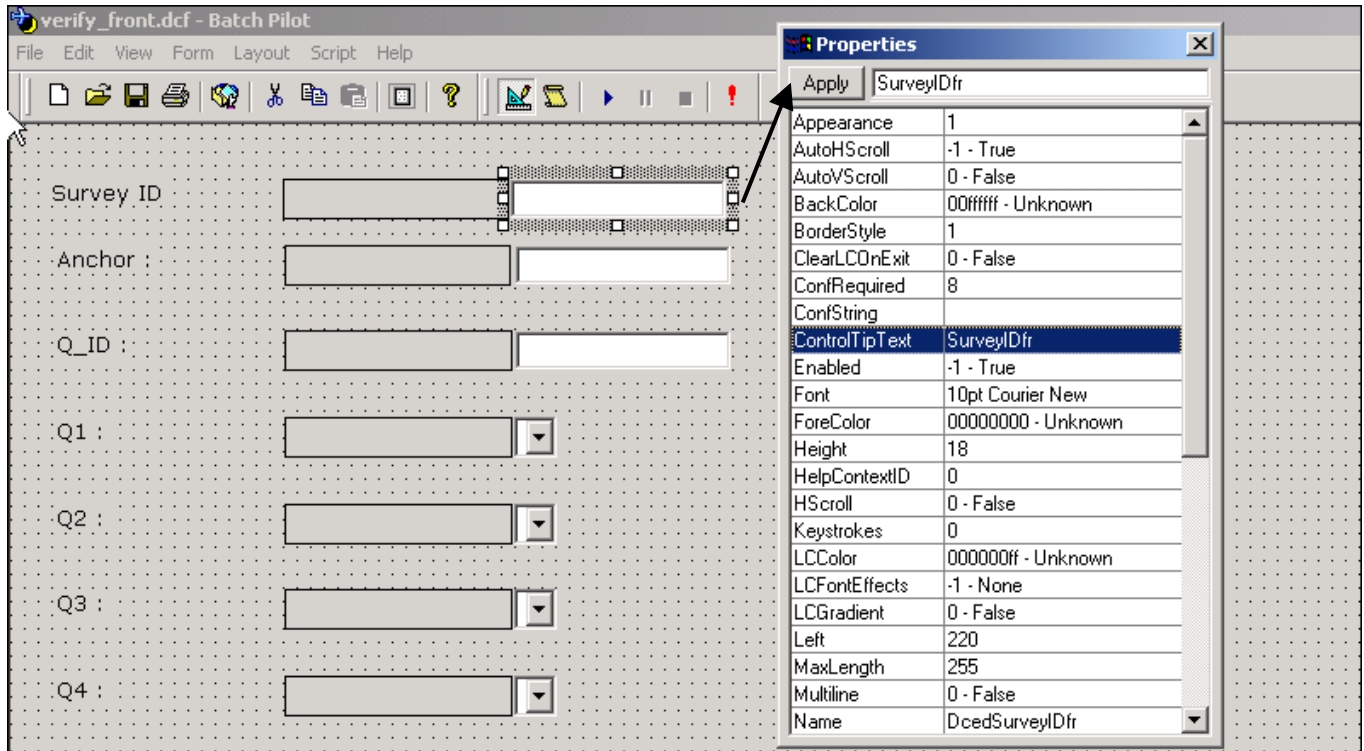
Label Control – *Caption* property

A snippet's image resides within a **DCImage** control. **Required** properties of this control include:

- **ControlTipText**: the **Field** object's ID (*Q\_ID*, in this case).
- **File Name**: a value assigned by *Autoform* to the overall image of the fingerprint that represents this page. **Alert!** This is the **File Name** value for *all DCImage* controls except the first, which is blank.
- **Name**: the **DCImage** designation (*Dcim*) combined with the **Field** object's ID (*Q\_ID*).

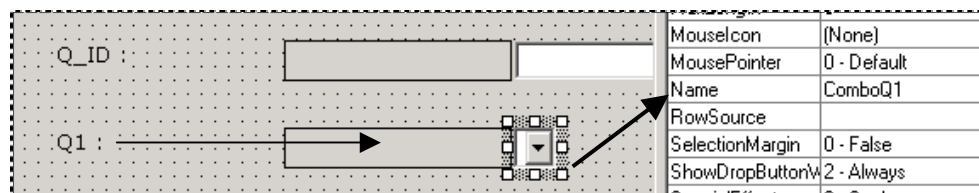


## Verify Task Setup



The interactive **DCedit** control contains a field's *recognized* value that is subject to change by an operator.

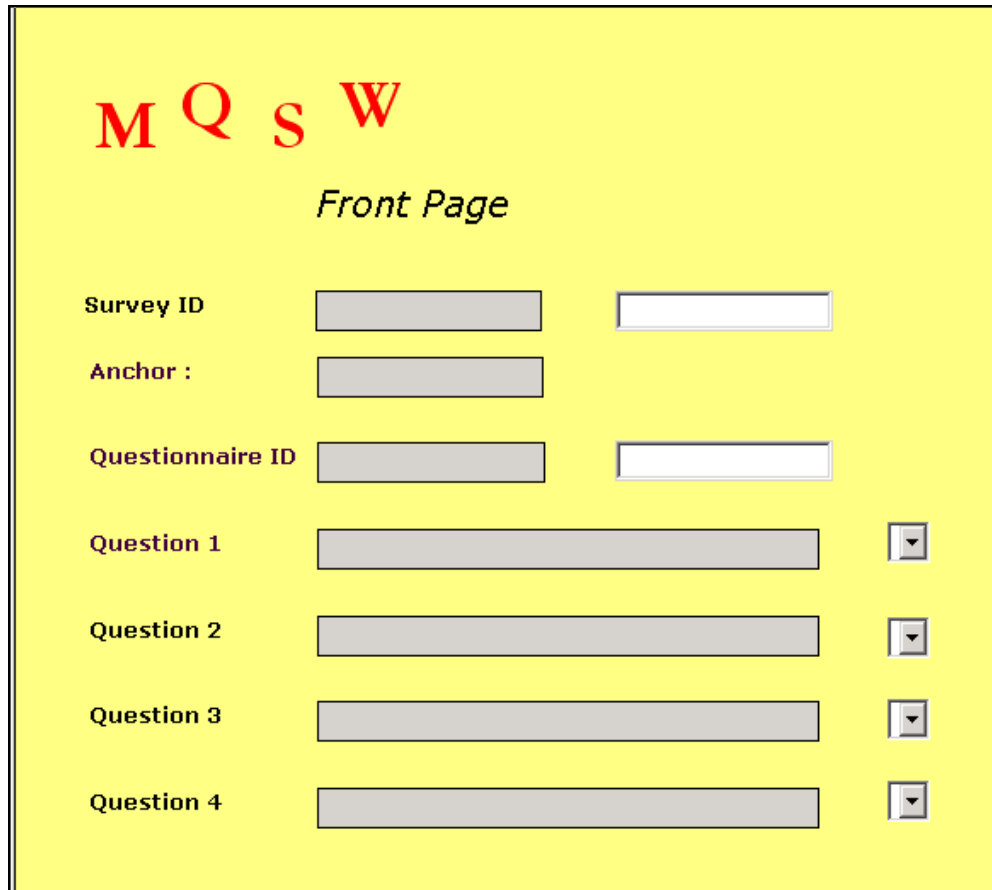
- **ControlTipText** indicates the **Field** object's ID (*SurveyIDfr*, in this case. Note the change in the control's label!).
- **Name** links the type of control to the **Field** object's ID – *DcedSurveyID*, in the example above but *ComboQ1* below:



- ✓ Each control, of course, has other properties that may deserve your attention. One example: the **ConfRequired** property of a **DCedit** control is a minimum Confidence Rating for values in the field. If the field's *recognized* data is below this level, the Verify task will change the field's background color to alert the operator to a problem.

And there may be other fields you'd like to add – fields that are *not* represented by **Field** objects of the Document Hierarchy...fields for an operator comments or notes, perhaps. Page 44 shows you how to add and remove fields from a *Data Entry* panel.

The panel for the *MQSW* application's *Front* page ended up looking like this:



**Data Entry Panel – MQSW Front Page**

When the Verify task actually runs, images will appear in the snippets on the left. *Recognized* data will appear in the **Survey ID** and **Anchor** Data Edit fields – and the operator will select an option from the drop-down lists for the **Question** fields.

The operator can also use tools of the *Batch Pilot Window* to expand and magnify a snippet's image or to review the full image of the current page.

### Tab Order Properties

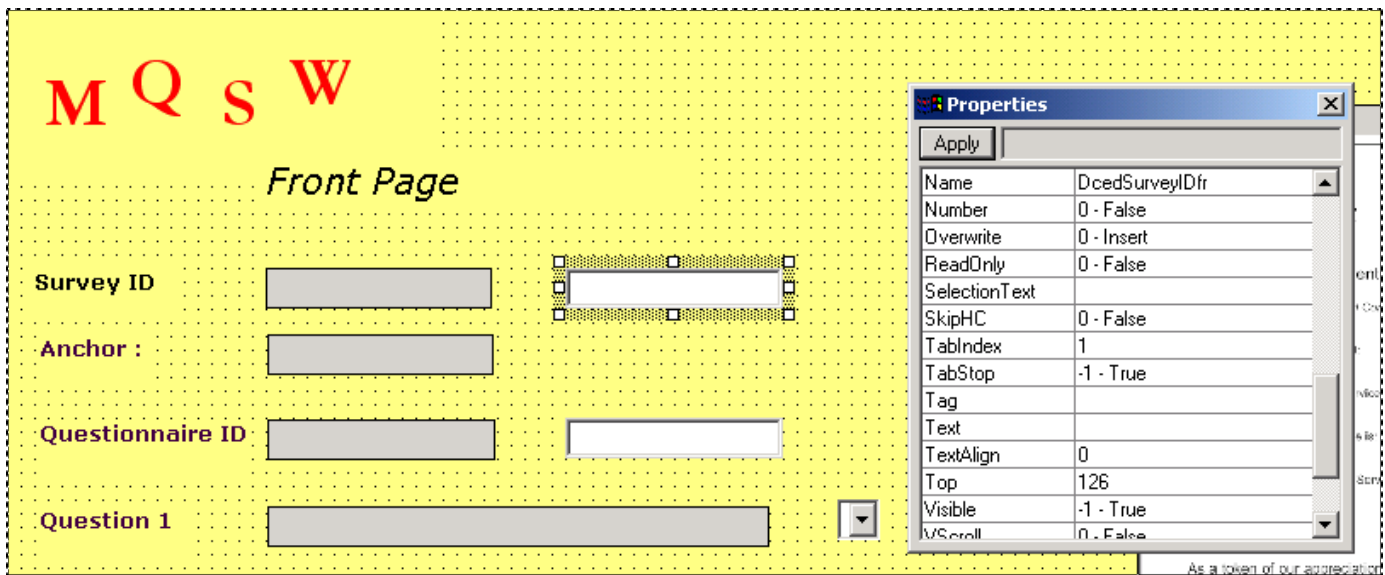
An operator can tab quickly from one Data Edit field to the next. Two properties (illustrated below) determine the tabbing order:

**TabStop** indicates whether or not a field can participate.

**TabIndex** specifies a field's relative placement in the tabbing sequence.

In this example, *-1 - True* is the **TabStop** is the value for each field including the first *Survey ID* field and the four *Question* fields.

- ✓ However, when the panel's designer removed the Data Edit field for the **Anchor** Field object – leaving only the image! – the **TabIndex** order was thrown off, and tests of the panel indicated that an operator could *not* tab from the *Survey ID* to *Questionnaire ID*.



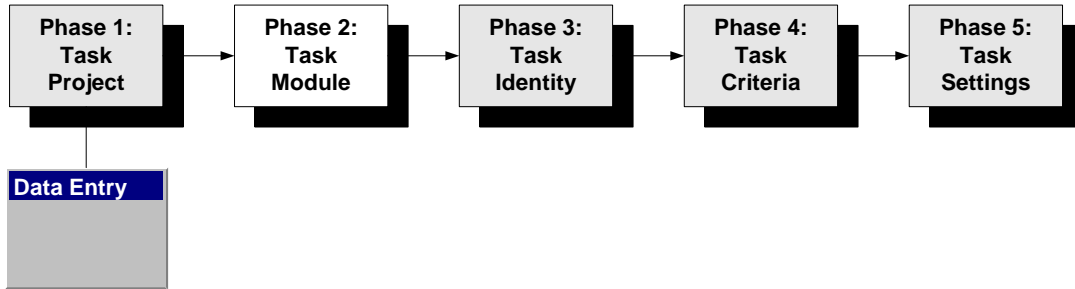
MQSW Front Page

To remedy the problem, the designer checked the **TabIndex** property of each field in the sequence to be sure that the field had a lower value than the field that was to follow. She then right-clicked on the panel itself and selected **Tab Order** to confirm the sequence of the **Dcredit** and **ComboBox** controls.

- ☛ At the conclusion of Phase 1 of Verify Task Setup, be sure that the Task Project (**verify.bpp**, for example) is part of your application's **Process** directory – and that the panel's Datacap form (**verify\_front.dcf**, in the preceding illustrations) is in the Process directory's **Verify** folder.
- ✓ Almost immediately after you add the Data Entry form to the **Verify** folder, Batch Pilot will add a second file, with an identical name but a **.dck** extensions. This file contains code for the “hot keys” an operator can use with the panel (Page 40).

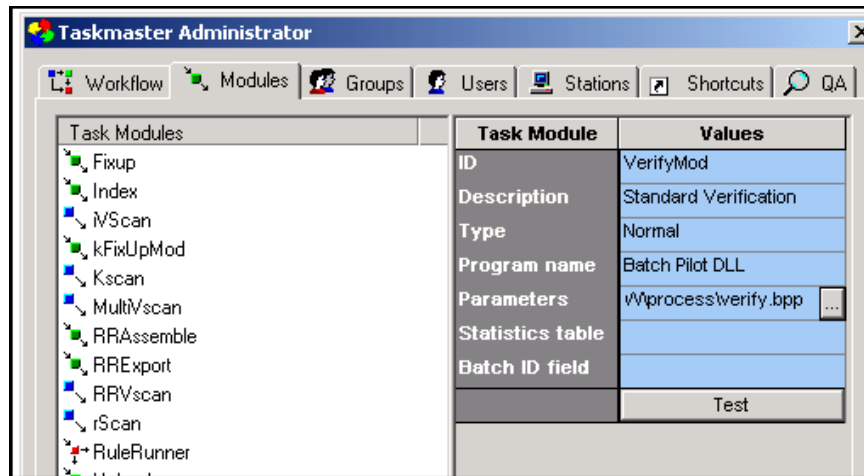
## Verify Task Setup – Task Module

A Task Module connects the Verify Task Definition to its Task Project (Page 11).



If you are setting up a new Verify task – and do not yet have a Task Module – you’ll take the steps below steps to define it. **Remember!** You cannot define a Task Module until a Task Project is firmly in place.

Step	Action
1.	Open the <i>Modules</i> tab of your application’s <i>Taskmaster Administrator</i> .
2.	Click on the Add button to clear the fields in the <b>Values</b> area on the right.
3.	Enter a unique Module <b>ID</b> and a brief but important <b>Description</b> of the module.

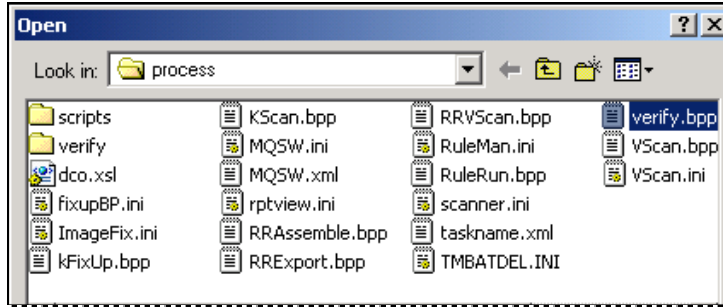


4. Select *Job Router* from the **Type** drop-down list if this Verify task can divert batches to a *child* job for review and repair if the task confronts unusual processing conditions (Chapter 10). Otherwise, select *Normal*.
5. Select *Batch Pilot DLL* from the **Program Name** drop-down list.
6. Click once in the **Parameters** field to display the field’s Browse button.

To Define a Verify Task Module (continued)

Step	Action
------	--------

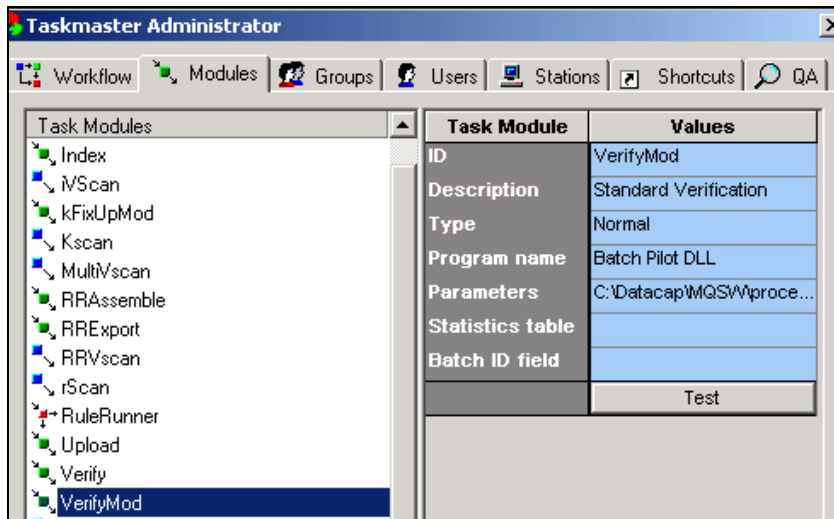
- Click on the Browse button to retrieve the *Open File* dialog. Select the Task Project file (.bpp) you assembled in Phase 1.



- Press the Open button to enter the file's name and path in the **Parameters** field.

Task Module	Values
ID	VerifyMod
Description	Standard Verification
Type	Normal
Program name	Batch Pilot DLL
Parameters	W\process\verify.bpp ...

- Press the Apply button at the bottom of the *Taskmaster Administrator*. Confirm that the new module's ID is now part of the **Task Modules** list on the left-hand side.

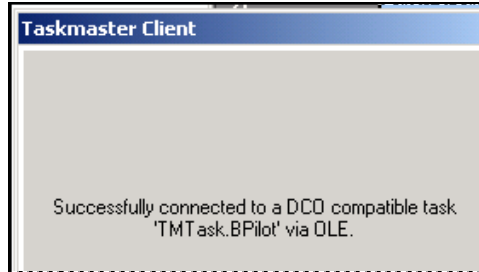




**To Define a Verify Task Module (continued)**

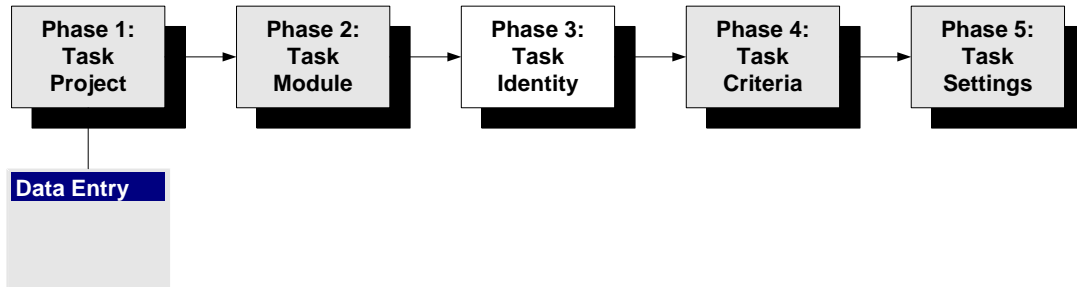
Step	Action
------	--------

10. Press the Test button. If the connection between the Task Module and Task Project is secure, you will receive this technical message:



**Verify Task Setup – Task Identity**

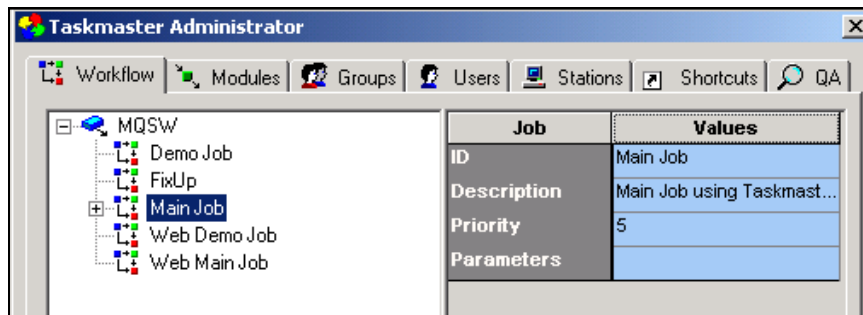
Phase 3 assembles the Verify task as a *Taskmaster* component, and assigns the Task Module (Phase 2) that will connect the Task Definition to its Task Project (Phase 1).



To provide a Verify task with its identity:

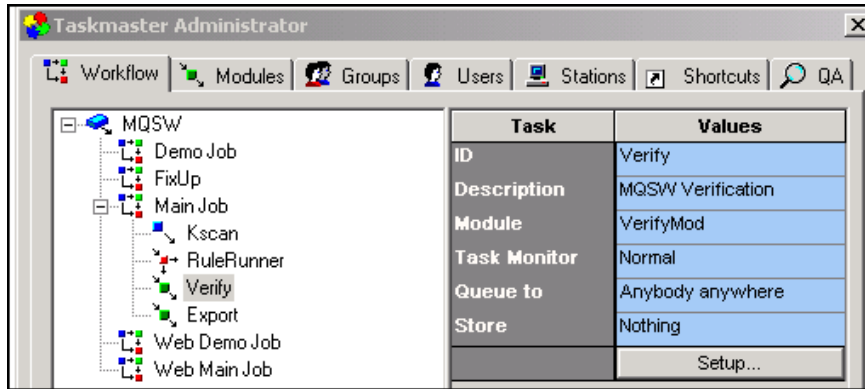
Step	Action
------	--------

1. Open the *Taskmaster Administrator's* Workflow tab.
2. Right-click on the **job** which will contain the Verify task (*Main Job*, in this example.)



### To Provide a Verify Task with an Identity (continued)

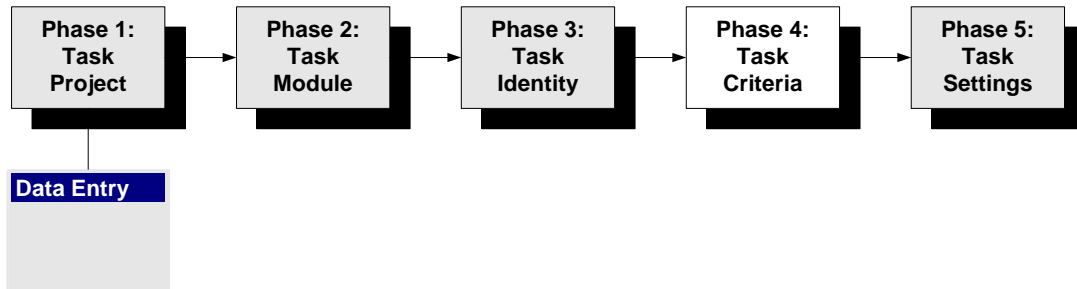
- | Step | Action |
|------|--------|
|------|--------|
3. Select **New** and **Task** from the options.
  4. Enter a *unique* Task ID in the open space below the Job ID. Be sure this value appears in the **ID** field of the **Values** area as well.



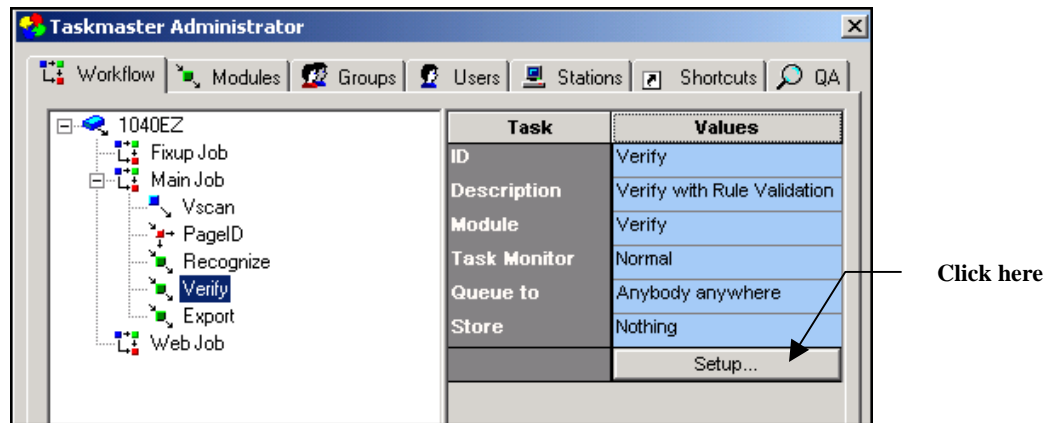
5. In the **Values** area, enter a brief but important **Description** of this Verify task.
  6. From the **Module** drop-down list, select the ID of the Task Module you decided on - or defined - in Phase 2 (Page 23).
  7. Do not modify the default value of the **Task Monitor** property.
  8. Select a value other than *Anybody Anywhere* from the **Queue to** drop-down list if access to the Verify task is limited to a station and/or operator involved with an earlier task.
  9. Select a value other than *Nothing* from the **Store** field if tasks after the Verify can be limited to the station and/or operator running the Verify task.
  11. Press the *Taskmaster Administrator's* Apply button to save the identifying properties of the Verify Task Definition.
- ✓ You'll find complete explanations of the Task Identity procedures in Chapter 6, or by pressing your F1 key when you are in the *Workflow* tab. This opens the set of *Taskmaster Help* topics that covers all aspects of the tab...including the elements of a Task Definition and of its **Queue to** and **Store** properties.

## Verify Task Setup – Task Criteria

A Verify task is a *RuleRunner* task because it applies rules of various types – rules that you define in the *Rules* panel of your application’s *Rule Manager Window* - well before you begin to setup the task.



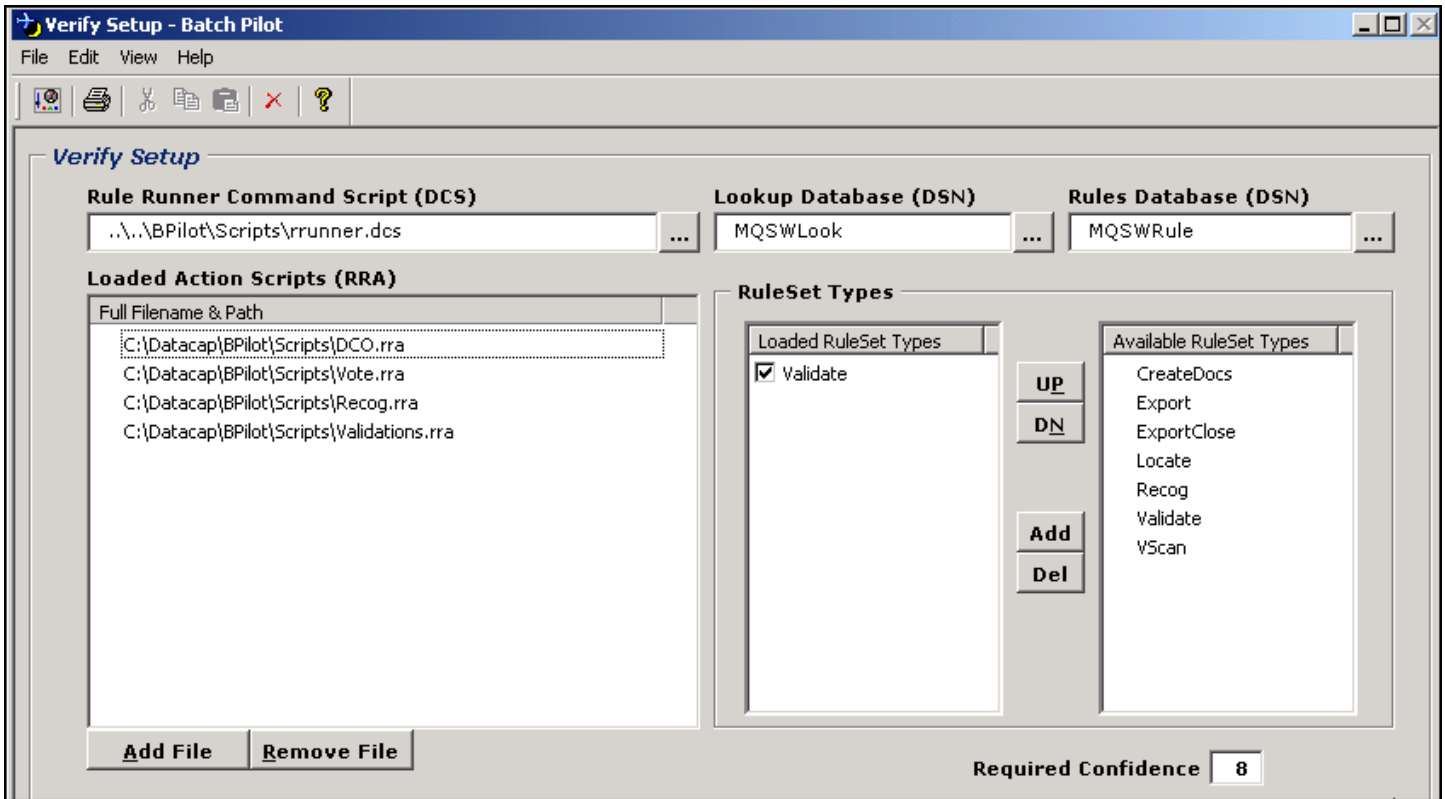
To assign pre-defined rules to the Verify task, highlight the Task ID in the *Taskmaster Administrator’s Workflow* tab – and press the Setup button. The *Task Setup* dialog will appear on your screen (see the illustration on the next page.)



1040EZ Taskmaster Administrator

Continued on the next page →

## Verify Task Setup



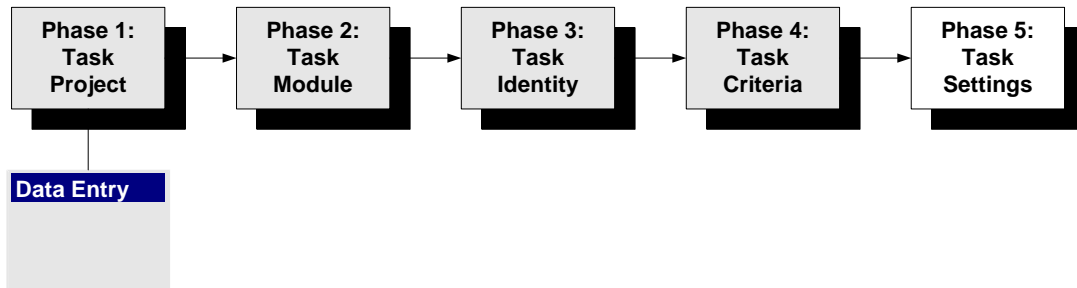
Verify Task Setup dialog

- ✓ **Very important!** These settings belong to the Verify Task Project – and are stored in the project's file (.bpb):
  - The **Available RuleSets Types** field on the right lists *all* RuleSet Types with rules that have been defined and applied to objects of the *MQSW* Document Hierarchy (in this example.)
  - The **Loaded Actions Scripts** field on the left identifies the Actions files that are members of the applicable RuleSet Type Actions Libraries (Chapter 4).
  - The **Loaded RuleSet Types** field in the middle lists the RuleSet Types with rules that *will be applied* by this Verify task. Here, the Verify task will call upon rules of one RuleSet Type only: **Validation**.

*Rule Manager's* scope, as well as its point-and-click mechanics, invite you to define rules of other sorts as well. Chapter 8 carefully examines each step you take to put together a task that applies the rules *you* define as it process a batch, it documents and pages, and the fields of a *source* page.

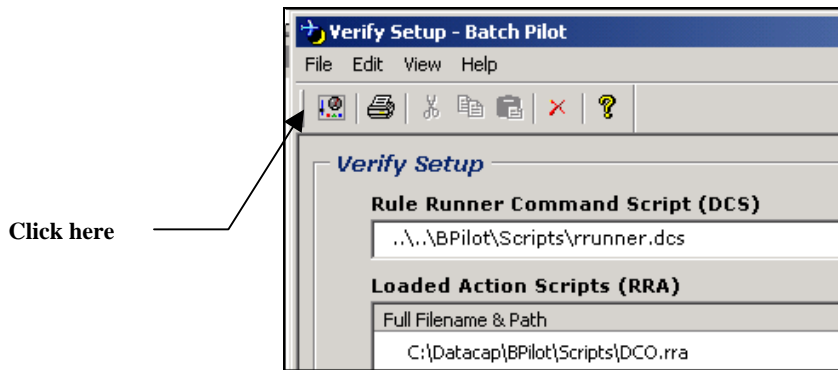
## Verify Task Setup – Task Settings

The concluding phase supplies the Task Definition with additional settings. For a Verify task, these settings are almost all *optional*.



The supplementary specifications are in the Task Project’s *Task Settings* dialog. To reach this dialog:

Step	Action
1.	Highlight the Task ID in the <b>Components</b> section of your <i>Taskmaster Administrator’s Workflow</i> tab.
2.	Press the Setup button to access the <i>Task Setup</i> dialog.
3.	Select <b>Task Settings</b> from the <i>Batch Pilot Window’s File</i> menu – or click on the <b>Task Settings</b> icon in the window’s toolbar. The <i>General</i> tab of the <i>Task Settings</i> dialog will appear on your screen (see the illustration on the next page.)



Verify Task Setup dialog

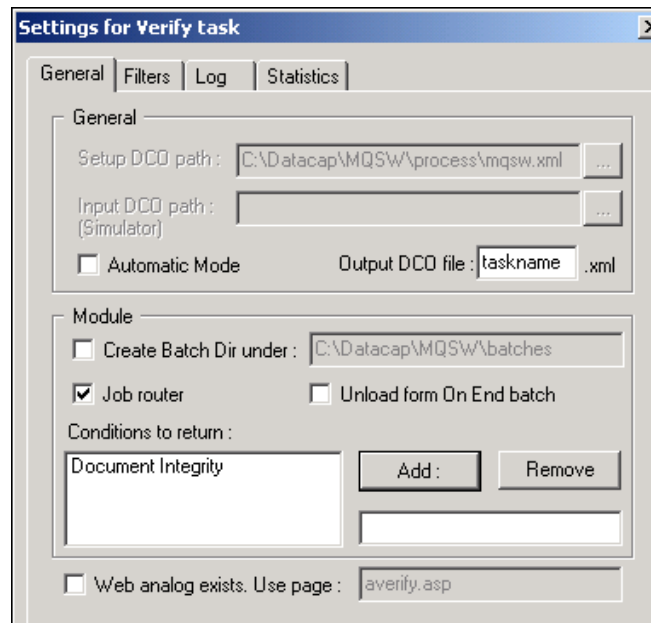
- ✓ Chapter 6 describes the four tabs of the *Task Settings* dialog. The next few pages focus on settings in the *General* and *Filters* tabs with special relevance to a Verify task.

## Verify Task Setup

---

If the Verify task can divert a **problem** batch from the *parent* job of which the task is a member to a *child* job for review and possible correction, you must add this detail to the *General* tab's **Module** area:

- Check the **Job Router** option.
  - Enter the name of the disruptive condition in the field just below the Add button.
  - Press the Add button to move the Condition ID to the **Conditions to Return** list.
- ✓ You also have to change the Task Module's **Type** value to *Job Router* (Page 23). Chapter 10 explores all aspects of problem detection and repair.

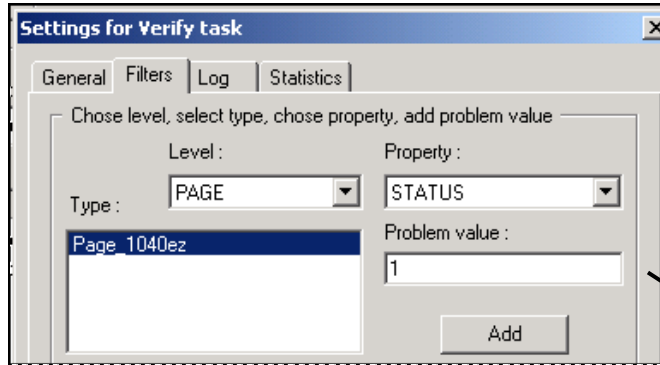


**Task Settings dialog – General tab**

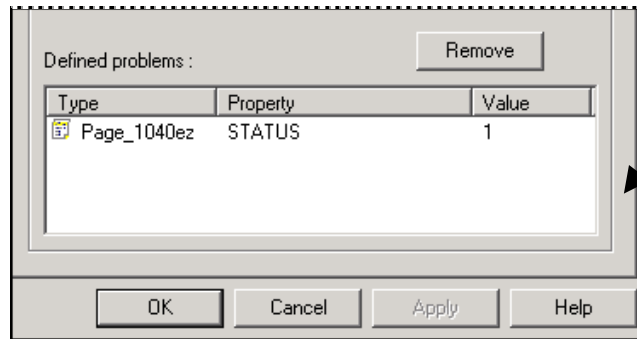
Settings in the *Filters* tab can contribute significantly to a Verify task's efficiency because they can limit the nature of the pages that are processed by the task.

The *1040EZ* training application uses the tab to define the simple but effective filter illustrated on the next page.

The filter acts on the *Page\_1040EZ Page* object of the Document Hierarchy. If the application's Recognize task has assigned a problem-free status ("1") to a *source* page, that page will **not** be subject to formal Verification procedures and will **not** appear in the *Data Entry* panel. All other *source* pages will be subject to formal Verification.

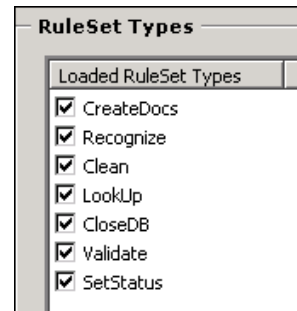
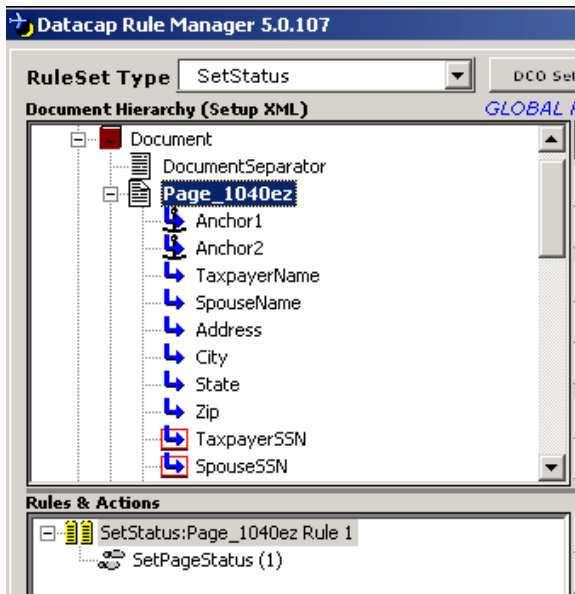


1040EZ Task Setting dialog - Filters tab (upper portion)



1040EZ Task Setting dialog - Filters tab (lower portion)

- ✓ **Very important!** A filter can only work if the application includes a rule that is bound to the object you've specified – the *Page\_1040ez* **Page** object in the example – and if the rule assigns a filtering value to the object's property. Below, a **SetStatus** rule uses a **SetStatus** action to assign "1" to the *Page\_1040ez* **Page** object.



RuleSet Types of the 1040EZ Recognize Task

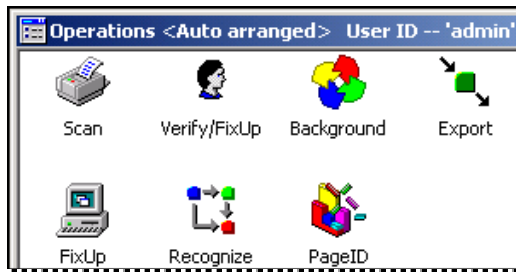
## Verify Task Operations

The illustration on the following page displays the *Data Entry* panel of the *1040EZ* application's Verify task in the middle of its operations.

- ☛ This panel is an excellent practice tool that is almost immediately available to you and your operators when you take these steps:

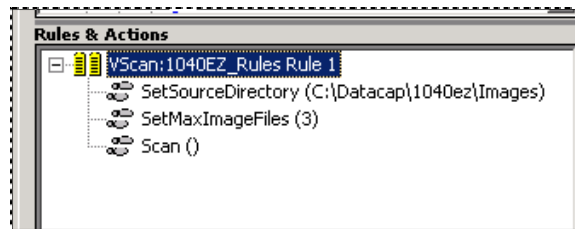
Step	Action
------	--------

1. Log onto the *pre-configured 1040EZ* application (Page 2).
2. Double-click on the **Scan** icon in the application's *Operations* window to launch the vScan task.



3. When Taskmaster asks if you would like to continue scanning, press the No button.
4. Double-click on the **PageID** icon to run the Page ID task. Follow up with the **Recognize** icon (and the Recognize task.)
5. Double-click on the **Verify/FixUp** icon to begin the Verification process and open the *Data Entry* panel.

- ✓ A **vScan** RuleSet determines how many pages will be “scanned” and added to a new batch. Although a number of test images are available in the application's **Images** directory, the RuleSet below directs the task to scan only three. (You can modify the parameter of the **SetMaxImageFiles** action so that the task processes more images – or less.



**vScan rule – Batch level**



**1040EZ Verify Task - Data Entry Panel**

If you run the 1040EZ Verify task and the *Data Entry* panel comes across a page such as Helen Osier’s Tax Return, the panel will present acceptable *and* unacceptable data in the following ways:

- A snippet will display the image of a field - or of a group of fields such as the taxpayer’s Name and Address.
  - A Data Edit field with a **blue background** contains *recognized* data that the RuleRunner task considered accurate and complete.
  - A Data Edit field with a **pink background** lacks *required* data – or contains data that is incorrect.
- ✓ Rules applied by the preceding Recognize task determine whether or not data is complete and accurate. These rules also:
- ◆ Re-format the amounts in the column on the right by removing decimal points and commas.
  - ◆ Use the *recognized* value in the **Taxpayer’s SSN** field to access a **Lookup database** containing taxpayer data. This allows the task to retrieve Ms. Osier’s

## Verify Task Operations

---

full name. However, because her record is incomplete, the panel is missing her address.

In this example, the Data Entry operator can tab from field to field, filling in missing or questionable data such as:

- ◆ Helen Osier's complete address.

Helen Leslie Osier		
1743 W. Main St.		
White Plains	NY	12733

- ◆ The Low Confidence “9” in the **Total Wages** field. (By default, Low Confidence characters are highlighted in bright yellow.)

Snippet:	Data:
93 257 57	9325757
390 00	39000

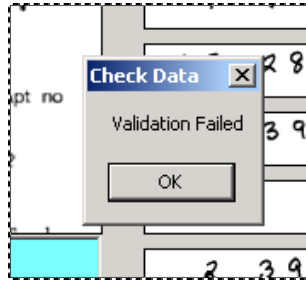
- ◆ The error in the Tax field (“06” should be “00”).

15 261 27	1526127	9. Total Payments
19 872 00	1987206	10. Tax

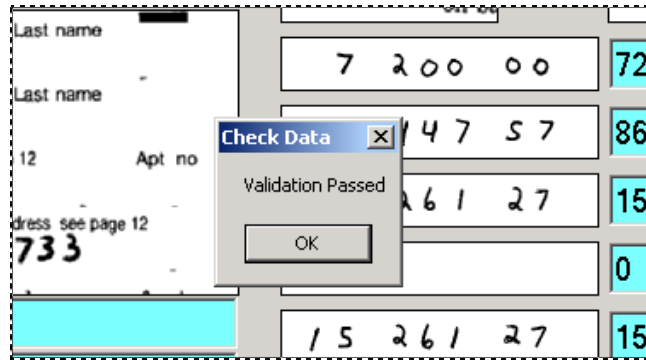
- ✓ At any point, the *1040EZ* operator can click on the Validate Return button at the bottom of the panel.

Validate Return (F4)

If problems remain, this warning below will appear. However, the background color of any problem fields that now have correct and complete data will change from pink to blue.

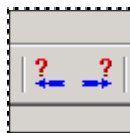


After the data in all fields is satisfactory, the task will alert the operator that the panel's *Validation Passed*.

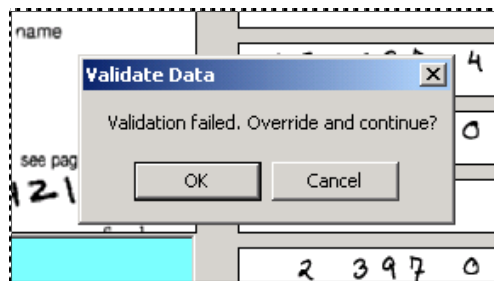


- ✓ **Remember!** A typical Verify task applies **Validation** rules to the panel's data. In the case of the 1040EZ task, these rules check to be sure that ZIP codes match City values. They also conduct simple calculations to confirm the mathematics in the amount fields.

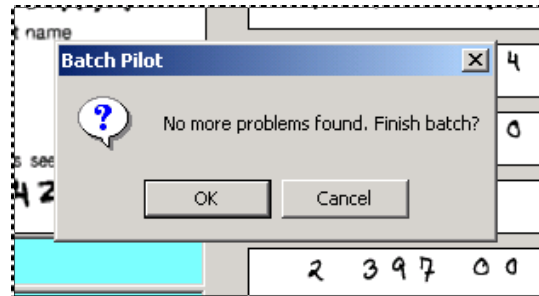
When the operator decides to move to the next problem page, he clicks on the toolbar's **Next Problem** icon.



**Alert!** If this page *still* has unsatisfactory data *and* if the operator has permission to override the task's validation procedures (Page 27), the dialog below will appear on his screen. Pressing the OK button retrieves the next problem page.



If there are no more problem pages, this message will appear:

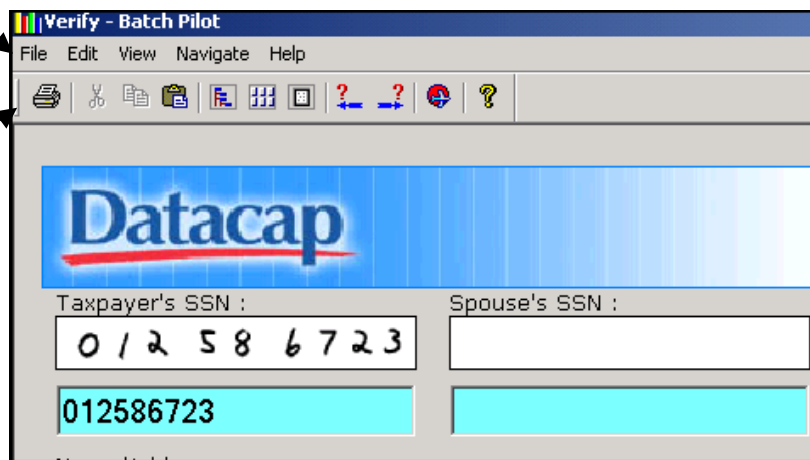


## Data Entry Tools

When a *Data Entry* panel occupies the *Batch Pilot Window*, the window provides the operator with a set of menus and tools designed specifically to enhance Verification.

Menus

Toolbar

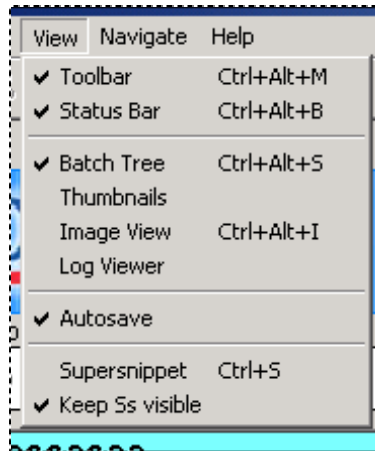


## Menus and Toolbar

The **File** menu's **Quit Task** item asks you if you want to put the batch on *Hold* – or terminate processing and assign a *Finished* status to the batch. The **Ctrl + Q** keyboard combination and the **Quit Task** toolbar icon duplicate this procedure.

The **Edit** menu does not contribute to Verification procedures.

The **View** menu has these important items:

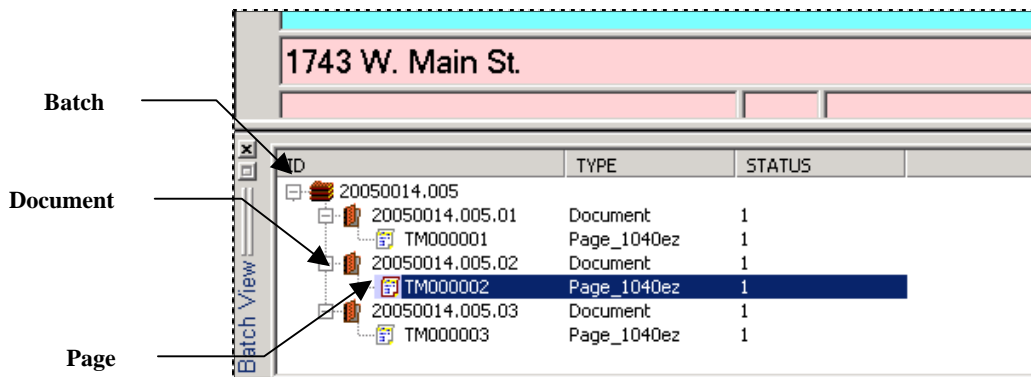


**View Menu**

**Toolbar:** toggles the *Batch Pilot Window's* Toolbar in and out of sight

**Status bar:** shows or hides the Status Bar at the bottom of the window. *Alert!* The Status Bar has valuable data (see the next page).

**Batch Tree:** opens or closes the Batch Contents hierarchy in the **Batch View** area at the bottom of the window. This structure can be a helpful tool for advanced Data Entry operators . The **Ctrl + Alt + S** keyboard combination and the **Toggle Batch Bar** toolbar icon duplicate this procedure.



**Batch Tree**

**Thumbnails:** Used by Scan tasks – not by Verify tasks.

**Image View:** opens or closes a full image of the current page. The **Ctrl + Alt + I** keyboard combination and the **Toggle Image** toolbar icon duplicate this procedure.

**Log Viewer:** opens or closes *Taskmaster's Log Viewer* dialog. The *Log Viewer* displays details of a Verify task's log *if* the *Log* tab of the *Task Settings* dialog direct the task to generate a Log file (Chapter 6).

**Autosave:** a option which, if selected, retains the current composition of the *Data Entry* panel and *Batch Pilot Window*.

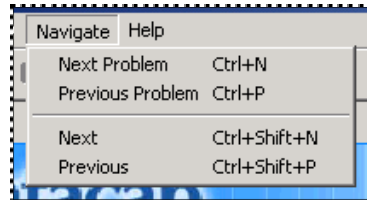
## Verify Task Operations

---

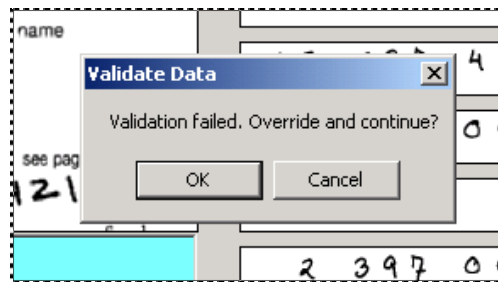
**Supersnippet:** Displays a larger view of the current field and its value. The **Ctrl + S** keyboard combination duplicates this procedure.

**Keep Ss Visible:** Automatically displays the Supersnippet for any field in which the operator places his or her cursor.

The *all-important* **Navigate** menu has these items:



**Next Problem:** validates data in the current *page* and moves to the next “problem” page. If any data in the current page is still incorrect or incomplete, the following warning appears.



### Validation Failed!

An *authorized* operator can override the validation of a page by pressing the OK button. *Be careful!* Overriding a validation error means that *all* values in the page are accepted as satisfactory – and the task moves on to the next page containing unsatisfactory data. Afterwards, the **Previous Problem** option cannot return the task to the original page.

If the operator presses the Cancel button instead, the Verify task moves to the next problem field but leaves the current field’s value and status unchanged.

The **Ctrl + N** keyboard combination and the **Next Problem** toolbar icon duplicate this procedure.

**Previous Problem:** validates the current page and issues the same warning (above) if problems remain. If an authorized operator overrides the validation alert, The Verify task treats all values as correct and returns to the previous page that still has problems.

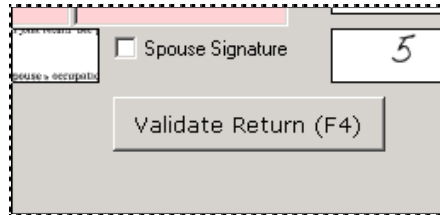
The **Ctrl + P** keyboard combination and the **Previous Problem** toolbar icon duplicate this procedure.

**Next:** retrieves field values and images of the next *source* page in the batch. The **Ctrl + Shift + N** keyboard combination duplicates this feature. *Alert!* This option accesses the next page, whether or not it has been validated.

**Previous:** retrieves field values and images of the next *source* page in the batch. The **Ctrl + Shift + N** keyboard combination duplicates this feature. *Alert!* This option accesses the previous page, whether or not it has been validated.

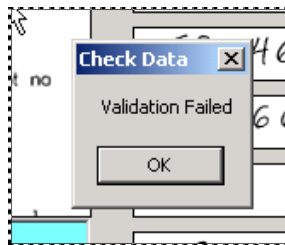
### Validate Return Button

The Validate Return button at the bottom of the *Data Entry* panel directs the Verify task to check all values in all fields of the current page.

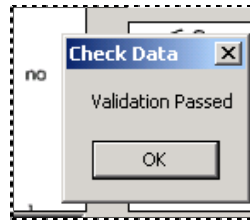


**Validate Return button**

If problems remain, this message appears and the background colors of the applicable fields change (from blue to pinks, for example.)



If the page is free of errors, this message appears, and the operator can use the **Next Problem** menu selection or toolbar icon to access the next page containing errors.



### Hot Keys

In addition to the keyboard combinations that accompany many *Data Entry* panel's menu items, the *Batch Pilot Window* itself has a set of standard "shortcut keys" –and you can define others.

Hot Key	Action	Index	Keycode
F1	<i>For Future Use</i>	1	112
F2	Mark/unmark page for review.	2	113
F3	Check integrity and raise condition if failed.	3	114
F4	For Future Use	4	115



Hot Keys (continued)

Hot Key	Action	Index	Keycode
F5	Check <i>Expected Docs</i> and raise condition if failed.	5	116
F6	Validate data and go to next document.	6	117
F7	Restore image from quarter view.	7	118
Shift+Alt+Q	Image quarter view (counter clockwise).	8	327761
Alt+Q	Image quarter view (clockwise).	9	65617
F10	Next Low Confidence character.	10	121
F11	Clear all controls.	11	122
Alt+V	Run validations.	12	65622
Alt+L	Go to next Low Confidence character / field.	13	65612
Shift+Alt+L	Go to next Low Confidence field.	40	327756
Ctrl+Alt+L	Go to next Low Confidence character / field. <i>Will not prompt you to go to the next problem page.</i>	-	-
Alt+Z	Clear current field.	14	65626
Alt+E	Image full view.	15	65605
Alt+H	Move focus to first field (field with lowest tab index)	16	65608
Alt+Number1 (Number Lock ON)	Navigate CCO down & left.	17	65633
Alt+Number2(Number Lock ON)	Navigate CCO down.	18	65634
Alt+Number3(Number Lock ON)	Navigate CCO down and right.	19	65635
Alt+Number4(Number Lock ON)	Navigate CCO left.	20	65636
Alt+Number5(Number Lock ON)	Navigate CCO. Set word to field value.	21	65637
Alt+Number6(Number Lock ON)	Navigate CCO right.	22	65638
Alt+Number7(Number Lock ON)	Navigate CCO up and left.	23	65639

### Hot Keys (continued)

Hot Key	Action	Index	Keycode
Alt+Number8(Number Lock ON)	Navigate CCO up.	24	65640
Alt+Number9(Number Lock ON)	Navigate CCO up and right.	25	65641
Ctrl+Alt+Number9(Number Lock ON)	Navigate CCO. Append word to field value.	26	196713
Alt+D	Delete the current document.	27	65604
Alt+R	Show/hide CCO lines.	28	65618
Alt+W	Show/hide CCO words.	29	65623
Ctrl+Alt+P	Print image.	30	196688
Shift+Alt+Number8 (Number Lock Off)	Move snippet contents up.	31	327718
Shift+Alt+Number2 (Number Lock Off)	Move snippet contents down.	32	327720
Shift+Alt+Number4 (Number Lock Off)	Move snippet contents left.	33	327717
Shift+Alt+Number6 (Number Lock Off)	Move snippet contents right.	34	327719
Shift+Alt+I	Zoom snippet contents in.	35	327753
Shift+Alt+O	Zoom snippet contents out.	36	327759
Shift+Alt+R	Restore default snippet display.	37	327762
Ctrl+Number5	For Future Use	38	131173
Ctrl+K	Delete remaining characters in the current field.	39	131147
Ctrl+Alt+R	Mark/Unmark for Rescan.	41	196690
Shift + Alt + S	For Future Use	42	327763
Shift + Alt + A	For Future Use	43	327745
Ctrl+N	Go to the next problem page in the batch.	-	-
Ctrl+P	Go to previous problem page in the batch.	-	-
Shift+Ctrl+N	Go to the next page in the batch.	-	-
Shift+Ctrl+P	Go to the previous page in the batch.	-	-

**Hot Keys (continued)**

<b>Hot Key</b>	<b>Action</b>	<b>Index</b>	<b>Keycode</b>
Ctrl+Q	Quit the current task.	-	-
Ctrl+S	Open/close a Supper Snippet	-	-
Ctrl+(KeyPad) Up	Scroll up in the image.	-	-
Ctrl+(KeyPad) Right	Scroll right in the image.	-	-
Ctrl+(KeyPad) Down	Scroll down in the image.	-	-
Ctrl+(KeyPad) Left	Scroll left in the image.	-	-
Ctrl+(KeyPad) Minus	Zoom out.	-	-
Ctrl+(KeyPad) Plus	Zoom in.	-	-

- ✓ This is a sampling of Hot Keys available to the **Data Entry** panel. Chapter 3 of the *Guide to Batch Pilot* shows you how to define additional Hot Keys.

# How to Add Fields to a Data Entry Panel

Adding a field **pair** – a snippet and Data Edit field - to a *Data Entry* panel, is a procedure that takes place in five stages. **Alert!** Be sure to work in a **Test** environment when you begin working with this procedure.

### Stage 1: Add the Field Object to the Document Hierarchy

If you are adding a brand new field, **be sure** to add its corresponding **Field** object to your application's Document Hierarchy first.

### Stage 2: Check the Field Object's Status

Check to be sure that the **Status** property of the new **Field** object is "0".

**Important!** If **Status** is not listed as a default property of **Field** objects in the *Document Hierarchy Setup* window, use a text reader such as *Notepad* to open the <app> .xml file located in the **Process** directory. Scroll down to the field's specifications and change the value of **Status** from "-1" to "0". This allows the field to be represented in the *Data Entry* panel.

### Stage 3: Add Field Controls to the Data Entry Form

The field will require two field controls in the *Data Entry* panel. During processing:

- The **Snippet** will display an image of the field and its value in the current *source* page;
- The **Data Edit field** will display the field's *recognized* value. This is an interactive field control: after comparing the value in the Data Edit field to the image in the Snippet, the operator can correct the value, if necessary.

You'll add the controls above to the **Datacap form (.dcf)** that contains the panel's structure and content. To access this form:

- ◆ Open the *Batch Pilot* workshop: from your Start button's **Program** listings, open the **Datacap Taskmaster** folder and the **Batch Pilot** sub-folder. Click once on the **Batch Pilot** option.
  - ◆ Use the **Open Form** item in the *Batch Pilot Window's* **File** menu to retrieve the **verify.dcf** form, which you'll find in your application's **Process** directory.
  - ◆ For extra caution, **save** the form **as** a different **.dcf** file
  - ◆ Move within the form to a "free" locale with plenty of space.
- ✓ To insert the controls, first
- ◆ Click on the **Toolbox** icon in the *Batch Pilot Window's* toolbar, or select **Toolbox** from the **Form** menu.

- ◆ *Be sure* the toolbox contains a **DcImage** icon *and* a **DcEdit** icon.

If either is missing:

- ◆ Right-click on the toolbox and select **Additional Controls**.
- ◆ In the *Additional Controls* dialog, check the **DcEdit** control option and/or the **Datacap Tiff Viewer Control** option (which produces the **DCImage** icon.)

Use the **DCImage** icon to place a rectangular Snippet control on the form, and the **DcEdit** icon to place a rectangular Data Edit field control just below the Snippet.

Save the form!

### Stage 3: Configure Properties and Events

Each control has a unique set of properties. To review and modify these properties, right click on a control you've added to the form to access the *Properties* dialog for that control.

- ✓ *Important!* The new Snippet control (using *5PaddTel* from the *Medical Claims* application as an example) requires the following specific values for three specific properties:

<b>ControlTipText</b>	5PaddTel
<b>FileName</b>	Dcim1aInsrID
<b>Name</b>	Dcim5PaddTel

The Data Edit field control needs these values:

<b>ControlTipText</b>	5PaddTel
<b>Name</b>	Dced5PaddTel

*Be careful!* Be sure to assign comparable values to the properties of your field pair –and do *not* deviate from the syntax in the example.

**Events** associated with the Data Edit field control also need a few lines of helpful code. If you double-click inside the control *after* you've supplied new values for its properties, the form's scripting mode moves into the forefront of the *Batch Pilot Window*:

Add this code to three events you can select from the drop-down list in the upper right-hand corner: **Enter**, **KeyPress** and **NoMoreLCChars**.

```
Sub Dced5PAddTel_Enter()  
    DcceditEnter(me)  
End Sub
```

```
Sub Dced5PAddTel_KeyPress()  
    Call stats_nKeyPress()  
End Sub  
  
Sub Dced5PAddTel_Enter()  
    Call NextLC1(True)  
End Sub
```

### Stage 5: Bind the Modified Form to the Verification Task Project

If you have been working with a “new” Data Entry form (.dcf), you have to **bind** it to the Verify task’s Task Project (.bpp). You can bind it to the existing project or, if you are being especially cautious, to a Task Project you have set up exclusively for development and testing.

If you are using a new form with a new name:

- ◆ Open *Batch Pilot* and the **Batch Pilot Window**.
- ◆ Select **Open Project** from the **File** menu; and open the applicable Verify Task Project.
- ◆ Select **Setup Tree** from the **View** menu.
- ◆ In the **Batch View** area at the bottom of the window, highlight the page just beneath the target document.
- ◆ Right-click and select **Pick Form**.
- ◆ When the *Open* dialog appears, select the appropriate **.dcf** file. This binds the form to the **Page** object.

Save the project!

## How to Remove a Field from a Data Entry Panel

Removing a field from a *Data Entry* panel involves fewer steps than adding a field - but just as much planning, and concern about the impact of such a procedure.

To remove a field:

- ◆ Open *Batch Pilot* and the *Batch Pilot Window* (Page 44).
- ◆ Select **Open Form** from the window's **File** menu to access the *Data Entry* panel's form (.dcf). (You'll probably find it in the application's **Process** directory.)
- ◆ Move within the form until you come to controls for the field pair you intend to remove.
- ◆ Right-click on the Snippet control and select **Delete**. Repeat this step for the Data Edit field.
- ◆ Typically, the field pair is accompanied by a Frame and perhaps a separate label. In all likelihood, you'll want to remove these controls as well.
- ◆ Change the **Status** of the **Field** object to "-1".
- ◆ Add a **SetDCOStatus** action – with "-1" as the parameter - to a **HideFields** RuleSet for this field.

Remember to save the **form!**