10

FixUp Tasks

This chapter reviews the steps you take to set up *Taskmaster's* FixUp tasks…tasks that take care of problems with images, data, document content and batch organization.

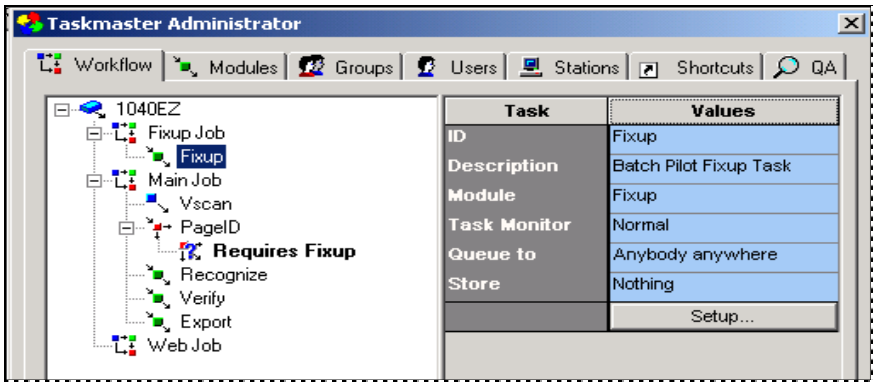
The chapter's contents include:

Introduction

Often, within a *Taskmaster* workflow, a task in a **Main job** can divert **problem batches** to a **FixUp job** for further attention and repair. In this kind of configuration:

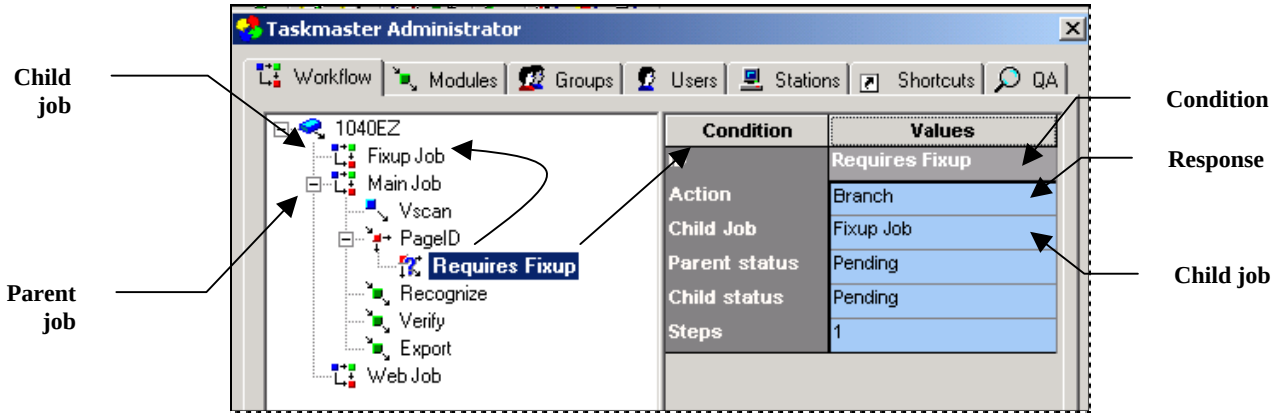
- The Main job is the ***parent*** job.
- The FixUp job is the ***child*** job.
- The operator of the FixUp **job's** FixUp **task** is responsible for assessing and correcting problems with the batch, its documents, its pages – ***and*** its data.

Below, the Workflow Hierarchy of the pre-configured *1040EZ* application has opened to reveal connections between the PageID task of the 1040EZ workflow's *Main* job, and the FixUp task of the *FixUp* job.



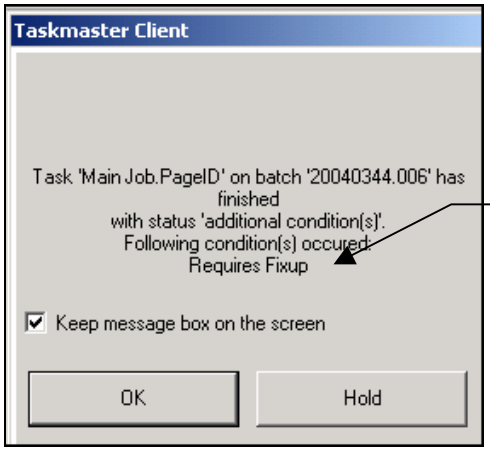
1040EZ Taskmaster Administrator – *Workflow tab*
Requires Fixup Condition

✓ A processing condition – in this case, *Requires FixUp* – forms the connection between the ***parent*** Main job and the ***child*** Fixup job. When you highlight the condition in the **Components** list on the left, properties of the condition appear on the right:



Condition Properties – *Requires Fixup*

When the PageID task encounters this pre-defined processing condition, the task does not conclude its work with the batch and assign a *Pending* status for the ***next*** task. Instead, it issues this message, and assigns a *Pending* status to the first task of the FixUp job:



This chapter examines the settings and operation of two types of FixUp tasks:

- **kFixUp** tasks (Page 23) use Kofax drivers as they work with batches formed by kScan tasks (Chapter 7).
- **iFixUp** tasks (Page 46) use ISIS drivers as they remedy problems involving batches formed by iScan tasks (Chapter 7).

First, however, take a close look at alternative ways to define the **processing conditions** that alert a task to branch a batch from a Main job's task to a FixUp job and its task(s).

✓ You can use the pre-configured *1040EZ* training application to explore the basics of FixUp tasks and their operations, and to set up tasks of your own. To access this application:

- Be sure your Taskmaster Server Service is up and running;
- Select **Datacap Taskmaster** from the **Programs** options of your Windows Start button;
- Open the **Applications** folder and the **1040EZ** sub-folder;
- Double-click on the **1040EZ** icon.
- Use your administrative security codes to logon to *Taskmaster* and the *1040EZ* application.

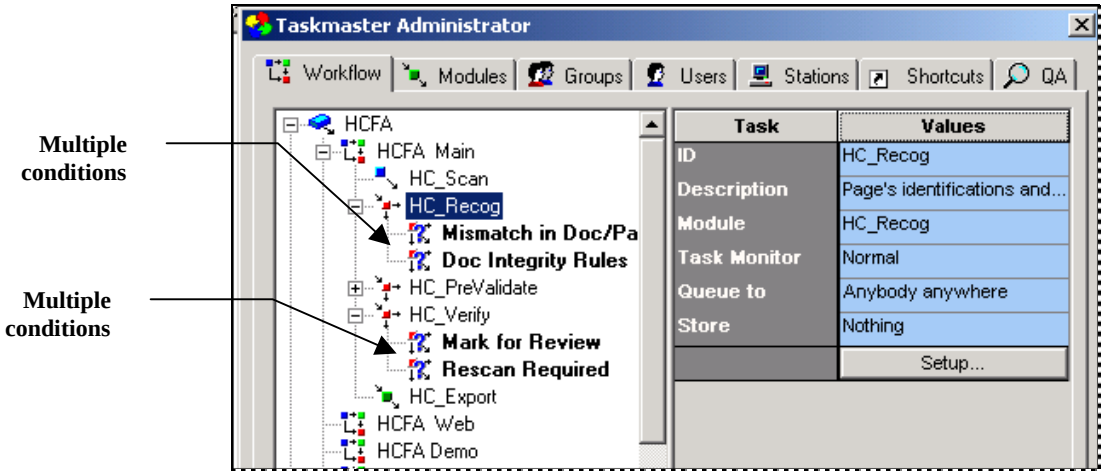
In addition, the *HCFA* and *UB92* applications of the pre-configured *Taskmaster for Medical Claims* system provide examples of FixUp configuration and procedures. (To access *Medical Claims*, follow the steps above – but select **MClaims** and **MClaims Client** from the **Applications** folder.)

The illustrations and examples in Chapter 10 are drawn from both the *1040EZ* and *Medical Claims* applications…and from the fictional *MQSW* application.

Processing Conditions

If the *1040EZ* application's PageID **task** (on the previous page) encounters a processing condition that has been labeled *Requires Fixup,* the task will place the batch in a *Pending* status and forward it to the FixUp **job** to await the attention of a FixUp **operator.**

In the *HCFA* application of the *Taskmaster for Medical Claims* system, ***two*** conditions have been defined and assigned to the HC_Recog task of the HCFA Main job. Two additional conditions have been defined and assigned to the HC_Verify task.



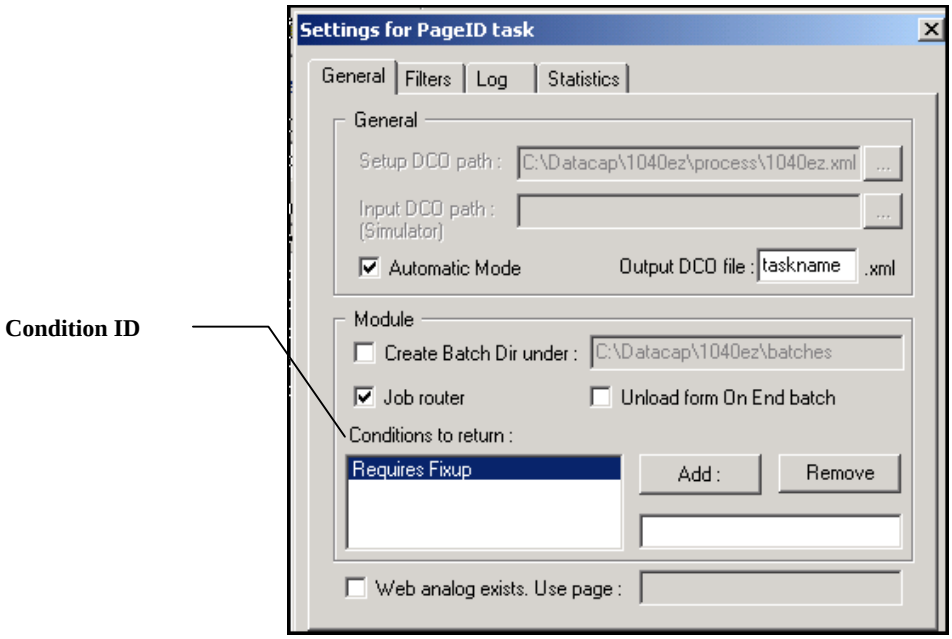
Medical Claims Taskmaster Administrator

✓ ***Very important!*** A condition is **task-specific**…it is set up exclusively for the task which displays its name. In the example above, ***Mismatch in Doc/Page Count*** belongs to the HC_Recog task and no other; ***Mark for Review*** is part of the HC_Verify task.

How to Define a Condition ID

Return to the *Workflow* tab of the *1040EZ* ***Taskmaster Administrator***. Select the Main job's PageID task listing, and click on the Setup button. When the ***Task Setup*** dialog appears, select **Task Settings** from the **File** menu to open the ***Task Settings*** dialog for the PageID task (Chapter 6 describes the features of the ***Task Settings*** dialog.)

A FixUp **rule** that governs the performance of a task that encounters a condition such as *Requires FixUp* employs a **Task_RaiseConditions** action that identifies the condition (Page 11). First, however, you have to define the condition.



Task Settings dialog – *General tab*
PageID Task

To add the *String* value of a Condition ID, you'll take these steps:

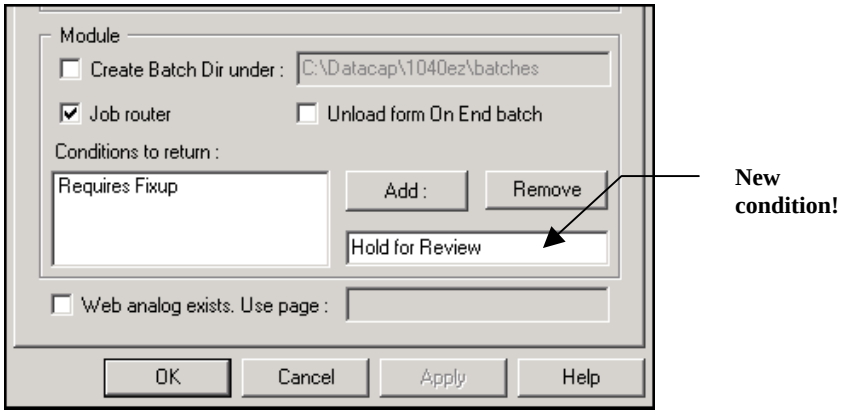
Step	Action

1. Select the Task ID of the ***parent*** job's task in the *Workflow* tab of the ***Taskmaster Administrator.***

2. Press the Setup button.

3. In the ***Task Setup*** window, select **Task Settings** from the **File** menu.

4. In the **Module** area of the ***Task Settings*** dialog, enter the Condition ID in the field below the Add button.



5. Click on the Add button. Be sure the Condition ID moves into the **Conditions to Return** list.

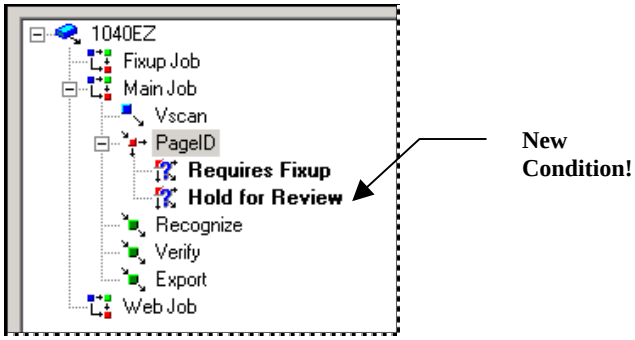
To Add the String Value of a Condition ID (continued)

Step	Action

6. Press the OK button and close the ***Task Setup*** window.

7. Press the ***Taskmaster Administrator's*** Apply and Done buttons.

8. Highlight the Task ID to be sure the new condition is listed.

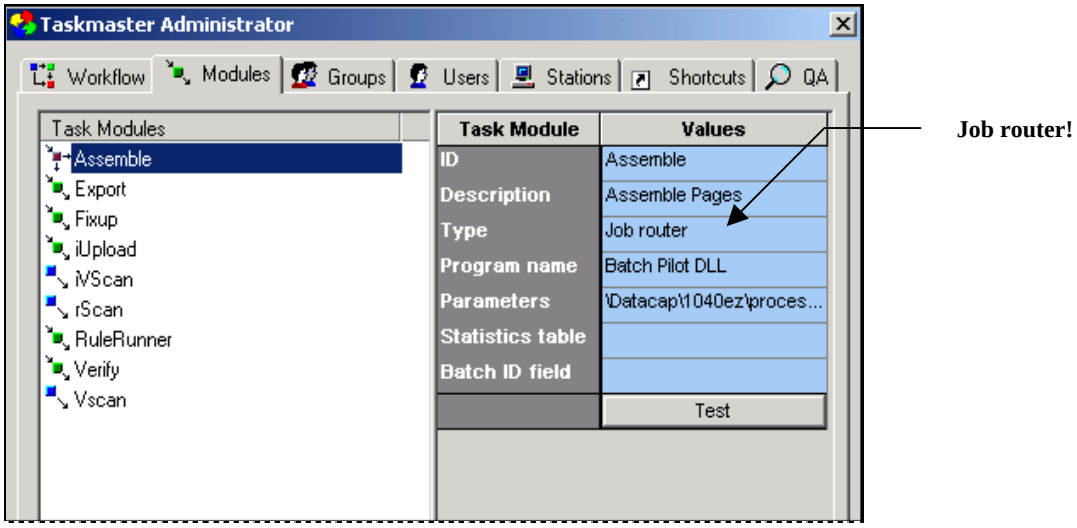


✓ ***Important!*** In this example, *Hold for Review* is the task's ***second*** condition. When you use an action such as **Task_RaiseCondition** (Page 11), this condition has a ***zero-based*** index value of "1".

Job Router Task Types

The illustration of **Module** settings in Step #4 above includes a **Job Router** check box. To allow this task to divert a batch from its ***parent j***ob to a ***child*** job, you ***must*** select this option.

You must be sure, as well, that the task's **Task Module** has a comparable **Type** property. The *1040EZ* application's PageID task uses an *Assemble* module with these properties:

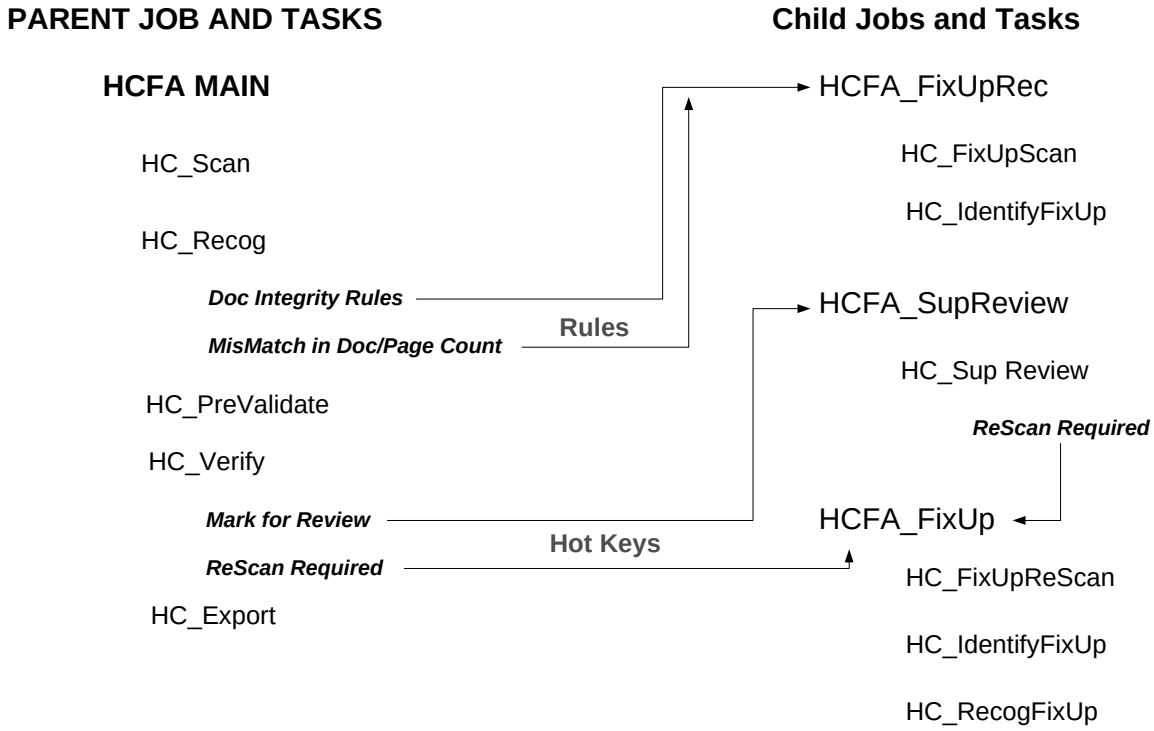


Taskmaster Administrator – *Modules tab*

- ✓ A task cannot branch a batch from a Main ***parent*** job to a FixUp ***child*** job (in this case) unless both the Module Definition ***and*** the Task Settings assign **Job Router** capabilities.
- ☛ Don't hesitate to use the *1040EZ* application - in a **Test** environment! - to develop, sharpen and maintain your condition-defining skills.

Rules, Actions and the Document Hierarchy

The chart below shows how and why the HCFA Main ***parent*** job refers problem batches to various FixUp ***child*** jobs and their tasks:



The chart points out an important distinction between the conditions governing the response of the HC_Recog task – and those of the HC_Verify tasks and the HC_Sup Review task.

- Rules and actions determine how the HC_Recog task responds to problems with the internal Document Integrity of a batch, or to incorrect counts of documents and pages within the batch.
- A Data Entry operator's use of pre-defined keyboard combinations (Hot Keys) automatically dispatches a troublesome batch to the HCFA _SupReview ***child*** job or to the HCFA_FixUpRescan ***child*** job for review and possible repair.

Additional considerations are less obvious but just as important:

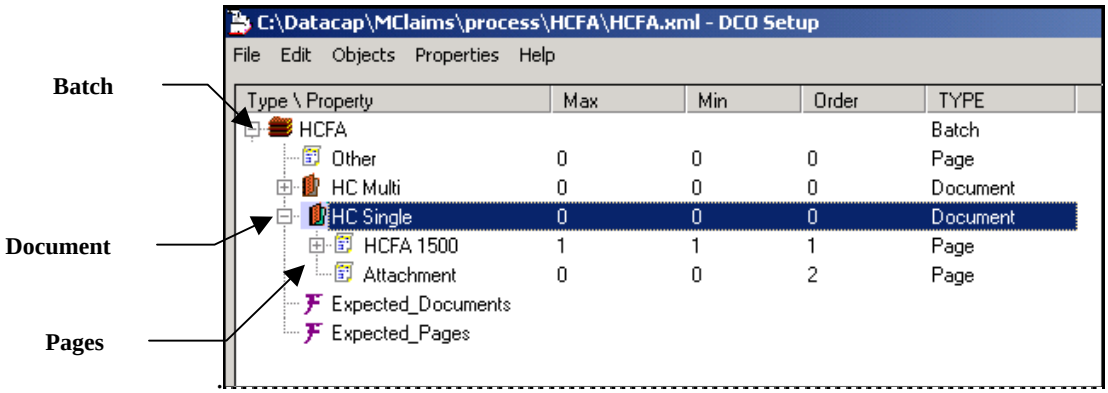
- Because the HCFA_FixUpRec **job's** HC_FixUpScan **task** handles batch organization problems, the task does not need to re-scan paper. Therefore, it operates in *FixUp* mode rather than *Rescan* mode.
- In contrast, the HC_FixUpReScan **task** of the HCFA_FixUp **job** operates in *Rescan* mode. This allows the task to resolve problems with batch organization ***and*** page quality.

Page 19 discusses the two modes, and Page 15 shows you how Hot Keys send a problem *Medical Claims* batch to the HCFA_FixUp ***child*** job. The next few paragraphs examine the way properties assigned to specific objects of an application's Document Hierarchy determine the required makeup of a batch – and how "FixUp" rules applied to those objects check the content and organization of each batch during processing, and forward problem batches to HCFA_FixUpRec ***child*** job.

Objects and Properties of a Document Hierarchy

A Document Hierarchy is an application's configuration ***and*** processing structure. Chapter 2 describes the Document Hierarchy's essential role; Chapter 3 shows you how to provide the hierarchy with objects at four levels: **Batch**, **Document**, **Page** and **Field** – and how to assign values to the objects' properties.

Below is a compact view of the *HCFA* application's Document Hierarchy. (For your own close-up look, click on your Windows start button and select these options: **Programs/Datacap Taskmaster/Applications/MClaims/ HCFA Rule Manager**. When the Rule Manager Window opens, press the DCO button in the **Zone Hierarchy** area of the *Fingerprints & Zones* panel.)



HCFA Document Hierarchy

The *HCFA* Document Hierarchy has one **Batch** object –*HCFA* – with two **Document** objects: *HCFA Multi* and *HC Single*.

The highlighted *HC Single* document (in this example) has two **Page** objects. *HCFA 1500* represents the document's ***source*** page… the page with user-entered data that will be recognized, validated, verified and exported. *Attachment* represents images of a claim's supplementary pages

The objects' **Min**, **Max** and **Order** properties contain important values:

- At the **Document** level, "0" indicates that a batch (one level up) does not have to have a multi-page document (*HC Multi*) or a single-page document (*HC Single*). In fact, when the HC_Scan task first forms the batch, it consists entirely of *Other* pages assigned to a catch-all document.

- If the batch does include an *HC Single* document (in this example), properties of the *HCFA 1500* **Page** object specify that the document ***must*** have one – but only one - *HCFA 1500* page (*Max = 1, Min = 1*) and that this page be the document's first (*Order = 1*).
- An *HC Single* document can have an unlimited number of *Attachment* pages ((*Max = 0, Min = 0*) but these pages ***must*** come after the document's ***source*** page…its *HCFA 1500* page.

✓ These are simple but essential **Document Integrity** requirements and any batch that fails to meet these standards will be subject to FixUp procedures ***if*** there are rules to back up the demands.

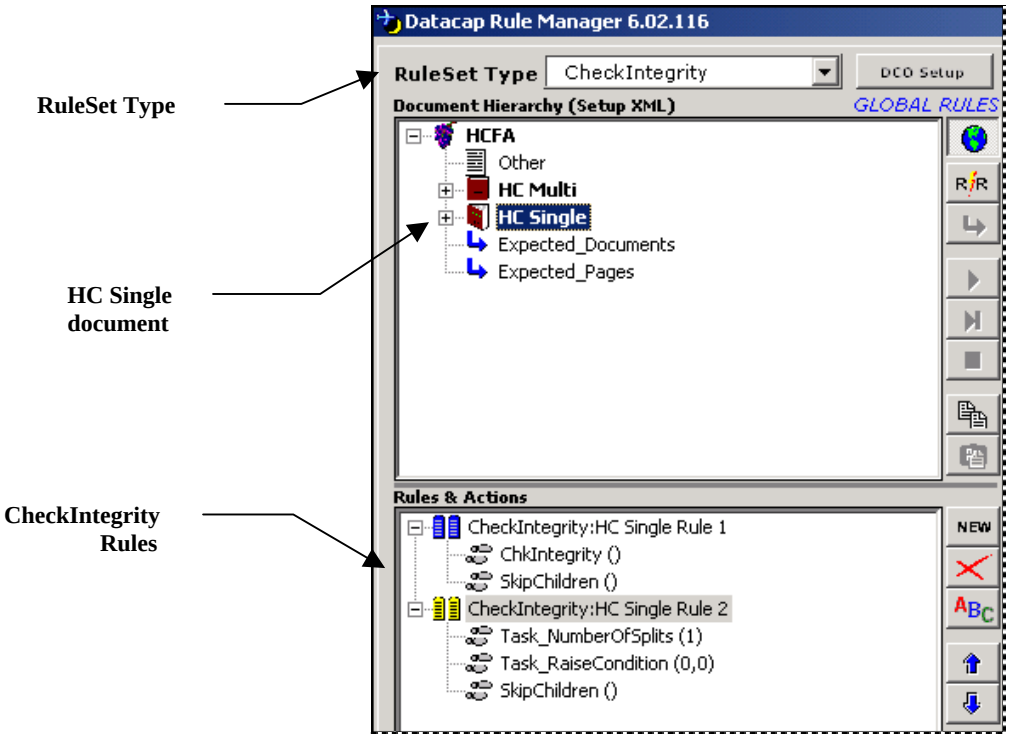
☛ ***Alert!*** Note, too, the *Expected_Documents* and *Expected_Pages* **Field** objects at the bottom of the hierarchy. These fields are children of the **Batch** object rather than of a **Page** object. When a Scan task runs, the operator uses a ***Start Batch*** dialog to enter ***runtime*** values to indicate the *expected* number of documents in the batch, and the *expected* number of pages in the batch (For more about this process, see Chapter 7.)

Later, if Page and Document Counts rules are in place, the HC_Recog task will check to see if *actual* counts of pages and documents equals *expected* counts. If there is a discrepancy, the task will forward the batch to the HCFA FixUpRec ***child*** job for review and repair (see the chart on Page 9).

Document Integrity Rules

The *Rules* panel of the *HCFA* application's ***Rule Manager Window*** contains the rules that define the ***Doc Integrity Rules*** condition and the ***MisMatch in Page Counts*** condition – and link them to the HC_Recog task.

The illustration on the next page depicts the structure of the **CheckIntegrity** RuleSet that evaluates the integrity of each *HC Single* document.



HCFA Rue Manager Window – *Rules panel*

In this example:

- **CheckIntegrity: HC Single Rule 1** uses its **ChkIntegrity** action to audit the integrity of an *HC Single* document. Does it have one (but only one) *HCFA-1500* page? Is this the first page in the batch? Are all other pages identified as *Attachments*, and do they come after the *HCFA-1500* page?
- If "Yes" is the answer to each question, the **SkipChildren** action intervenes to prevent **CheckIntegrity** rules from being applied to "children" of the document (in this case, pages). The RuleSet closes without going to **CheckIntegrity: HC Single Rule 2.**
- If "No" is the answer to any of the questions, Rule 1 closes and Rule 2 takes over. Here:

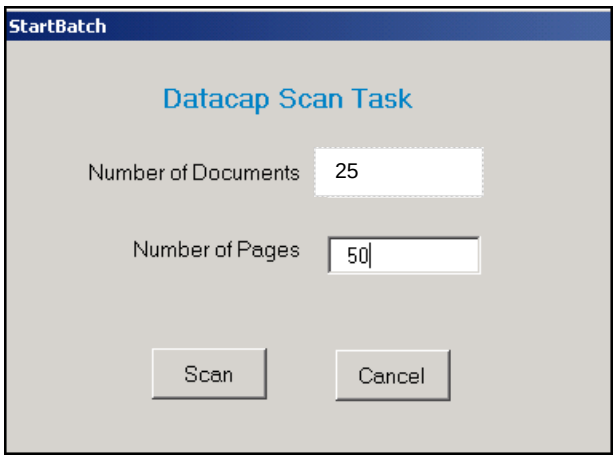
  `Task_NumberOfSplits(1)` splits the batch from the ***parent*** job.

  `Task_RaiseCondition(0,0)` responds to its first parameter ("0") by assigning the entire batch to the ***child*** FixUp job designated in the *Workflow* tab of the ***Taskmaster Administrator.*** The action's second parameter (also "0") associates the first processing condition – ***Doc Integrity Rules*** – with the diversion of the current batch to the ***child*** FixUp job.

  `SkipChildren(0)` avoids the application of additional Check Integrity rules at the Field level.

Document and Page Count Rules

Operations of a kScan or iScan task begin when the task presents the task's operator with a ***StartBatch*** panel - and the operator fills in his estimates of the **Number of Documents** in the scanner's tray, and the **Number of Pages**:



kScan StartBatch Panel

These amounts are recorded in the task's Page file (Chapter 6) and picked up by succeeding tasks. In the case of the *Medical Claims* applications, they become the values of the Document Hierarchy's *Expected_Document a*nd *Expected_Pages* fields (see the illustration on Page 10).

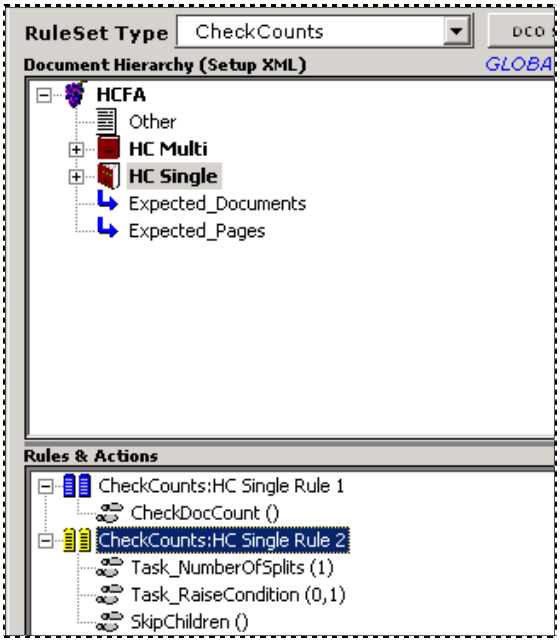
Typically, a Recognition task such as the *HCFA* application's HC_Recog task will compare *expected* counts to *actual* counts. If the task is responsible for applying Document and Page Count rules, it can branch a batch with discrepancies to a ***child*** FixUp job.

In the example on the next page, a **CheckCounts** rule is "bound" to the *HC Single* **Document** object. (For a complete explanation of rules and Rule Definitions, see Chapter 5.)

This **CheckCounts** RuleSet has two rules:

- In the first, the **CheckDocCount** action compares *expected* document to *actual* documents. If the numbers agree, the RuleSet closes.
- If the numbers do not agree, the task…
    - temporarily splits the batch from the ***parent*** job;
    - associates it with the task's second condition (***MisMatch in Doc/Page Count***, in this case);
    - delivers the batch to the first FixUp ***child*** job;

- directs the task to forego **CheckCounts** rules applied to pages within the document.

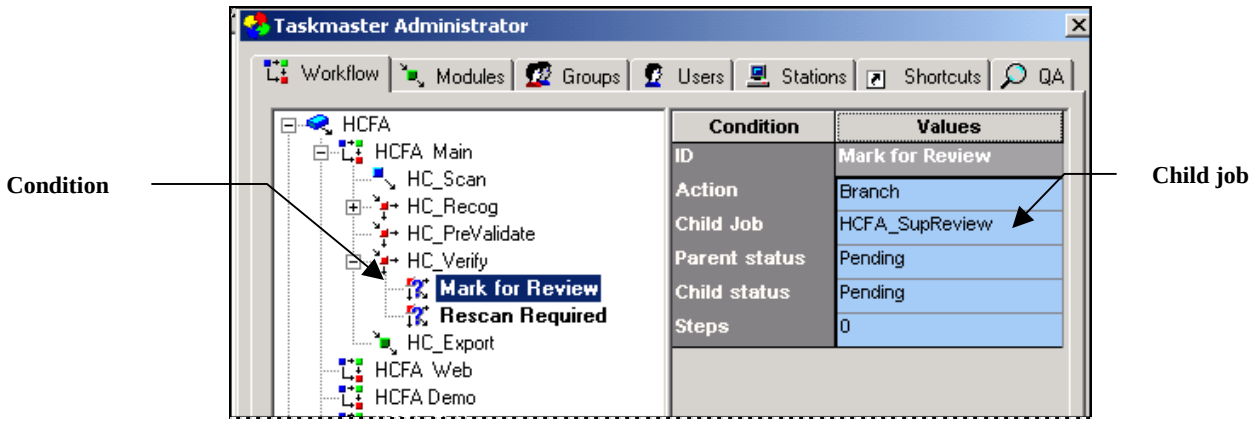


HCFA Rule Manager Window – *Rules panel*

Important! Although the default *HCFA* application does not count pages, *Rule Manager's* **CheckCounts** RuleSet has a **CheckPageCount** action that you can bind to a **Page** object.

How FixUp Hot Keys Work

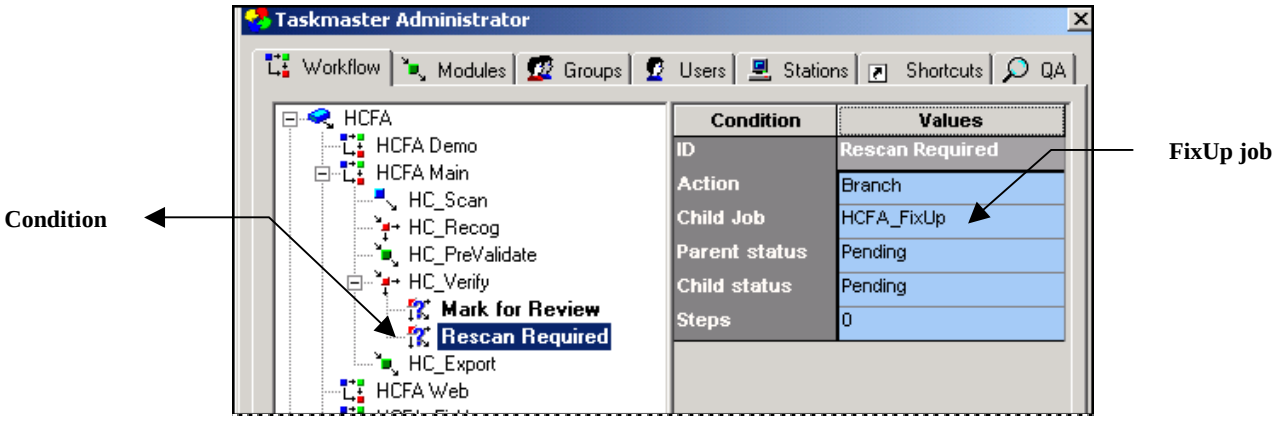
In the chart on Page 9, the HC_Verify task of the HCFA Main job employs two conditions: ***Mark for Review*** and ***Rescan Required.***



HCFA Taskmaster Administrator – *Workflow tab*

Important! These conditions respond to mechanical stimuli: if the task's operator decides to mark a page for further review (and send the batch to the HCFA_SupReview ***child*** job), he or she can use a "hot key" combination while in the ***Data Entry*** panel. (Chapter 9 lists a Verify task's hot keys.)

Similarly, if one or more pages need to be re-scanned, the operator uses a different hot key combination to forward the batch to the HCFA_Fixup ***child*** job.

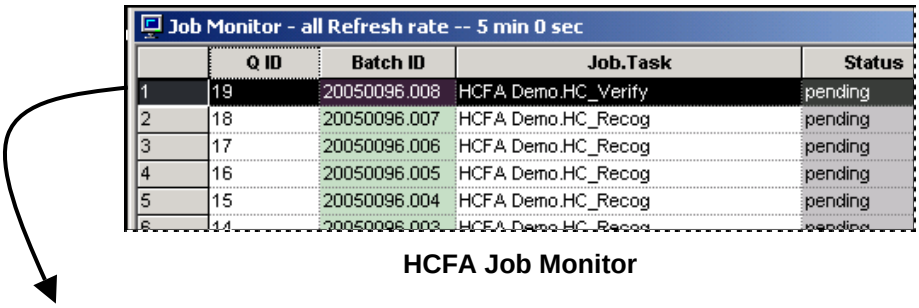


HCFA Taskmaster Administrator – *Workflow tab*

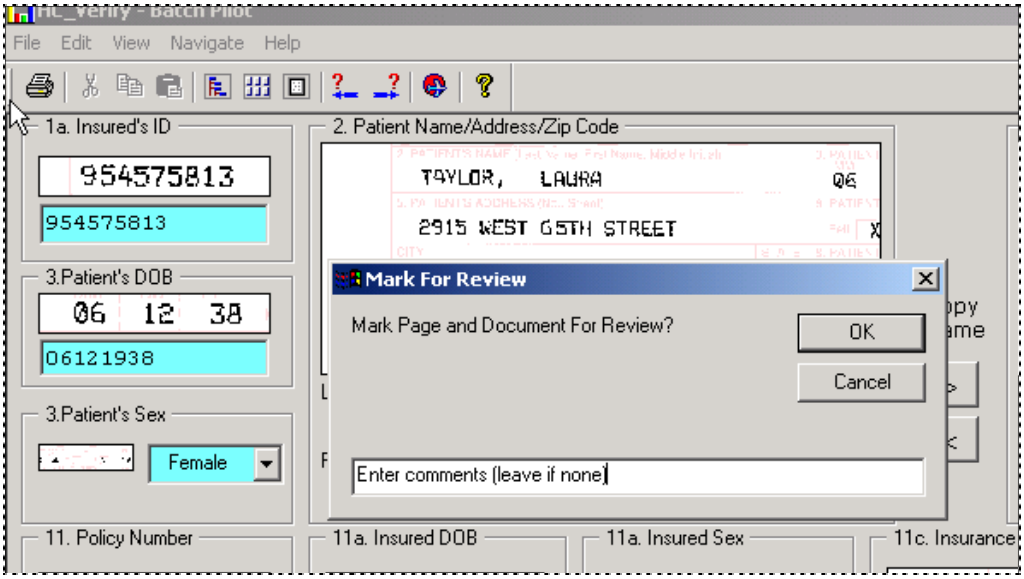
The next page shows a portion of the *HCFA* application's ***Data Entry*** panel. In this example, two pre-defined Hot Keys give the Data Entry operator a chance to branch the current batch to a ***child*** job for review and possible repair:

- **F2** marks the current page for further review. ***After*** the operator has finished verifying the batch, the batch will be diverted to the HCFA_SupReview ***child*** job.
- **Ctrl+Alt+R** marks the current page for re-scan – and branches the batch to the HCFA_FixUp ***child*** job, in accordance with the properties of the HC_Verify task shown on the previous page.

Using the mechanics of the F2 key as an example, the procedure begins when the operator launches a batch waiting for Verification.



HCFA Job Monitor

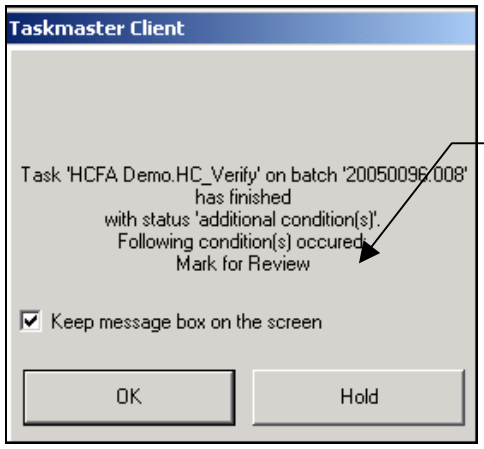


HCFA Data Entry Panel

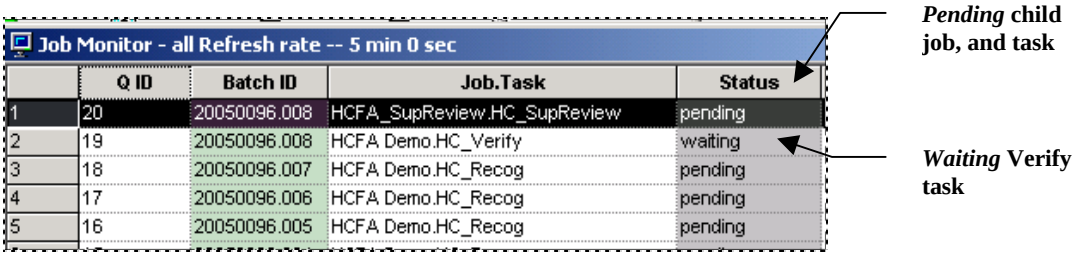
If the operator feels that a page in the batch warrants a Supervisor's attention, he presses his keyboard's F2 button: the Verify task displays the ***Mark for Review*** dialog.

After inserting an *optional* comment and closing the dialog, the operator completes his work with the batch – almost.

The *Taskmaster Message* pad now alerts the operator that the Verify task has encountered a *Mark for Review* condition. As a result, the batch will be branched to the HC_SupReview ***task*** of the HCFA_SupReview ***job***.

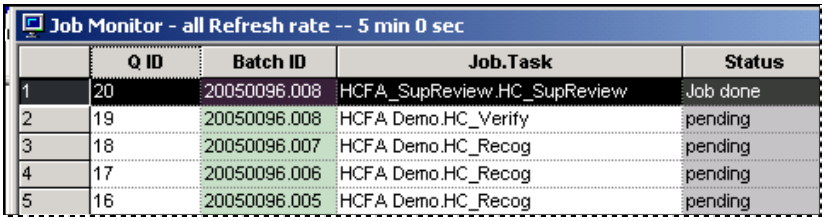


Taskmaster Message Pad



HCFA Job Monitor

Now, it is the Supervisor's responsibility to run the HC_SupReview task and take care of any problems. When she's finished, the *HCFA **Job Monitor*** will list a HCFA_SupReview job with a *Job done* status, and an HC_Verify task with a *pending* status – now waiting for the final efforts of the Data Entry operator.



HCFA Job Monitor

✓ A Data Entry operator can take advantage of numerous, pre-defined Hot Keys to navigate through a panel and a batch, and Hot Keys such as F2 and Ctrl+Alt+R to forward a batch to a child job. (See Chpater 9 for a list of all pre-defined Hot Keys.)

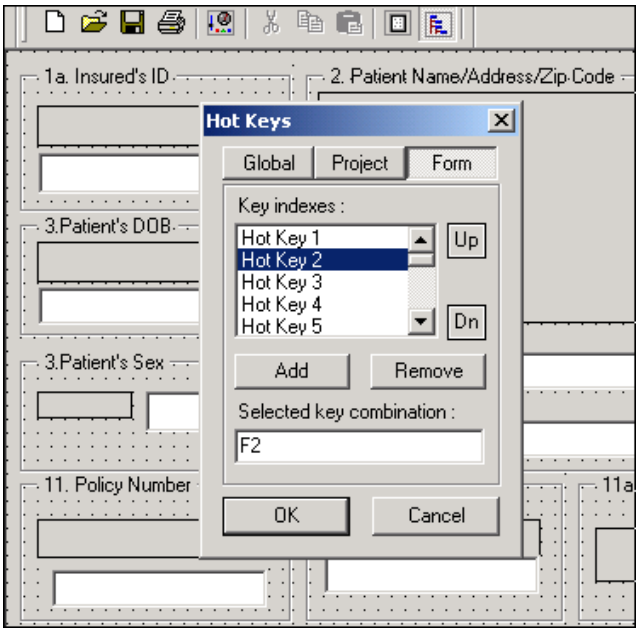
However, although these Hot Keys exist, the link between a particular Hot Key and a specific condition that is a property of a specific task does not.

Linking a Hot Key to a Task's Condition

A Hot Key is a property of a Data Entry form (.dcf) – and the form is a component of a Verify Task Project (bpp). Both the project and its form are set up in the *Batch Pilot* workshop (Chapter 6).

The *HCFA* application's Verify task operates according to parameters and code provided by the **HC_BPVerify.bpp** Task Project. The project's ***runtime*** form (**hcfa.dcf**) is the foundation for the *HCFA* application's ***Data Entry*** panel – and is a component of the project.

Hot Keys are properties of the form. In this illustration, *Hot Key 2* is the F2 key.



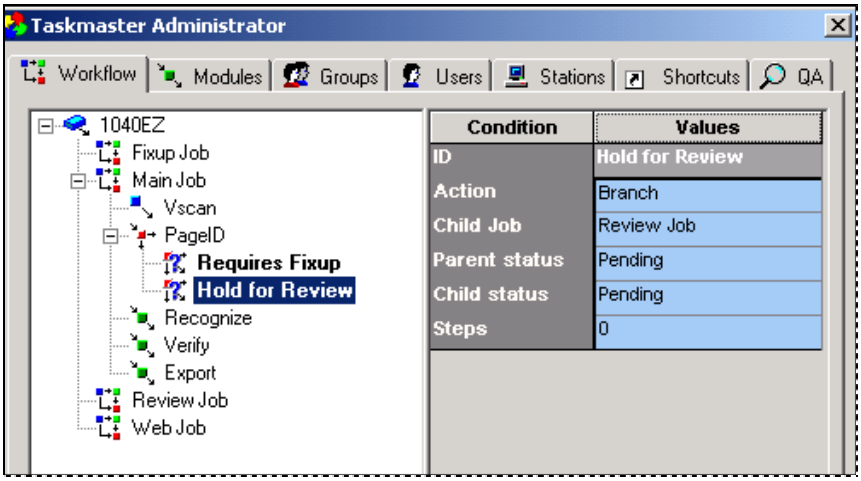
The ***behavior*** of the F2 key as a property of the form, the project and –ultimately – the Verify task is a ***scripted*** element of the project itself.

Technically, a ***Pilot_OnHotKey(KeyIndex)*** subroutine introduces the F2 Hot Key, and calls the **MarkForReview(Pilot.ActiveObject)** function. This function, in turn, determines exactly what will happen when an operator clicks on the F2 key.

✓ ***Very important!*** A Hot Key that diverts a batch to a ***child*** job in response to a processing condition by a task in the ***parent*** job cannot operate correctly without a script. For assistance in preparing such a script, be sure to contact your Datacap Implementation Specialist.

Child Job Definitions

A task in the ***parent*** Main job cannot branch a batch to a FixUp ***child*** job until the ***child*** job has been fully defined.



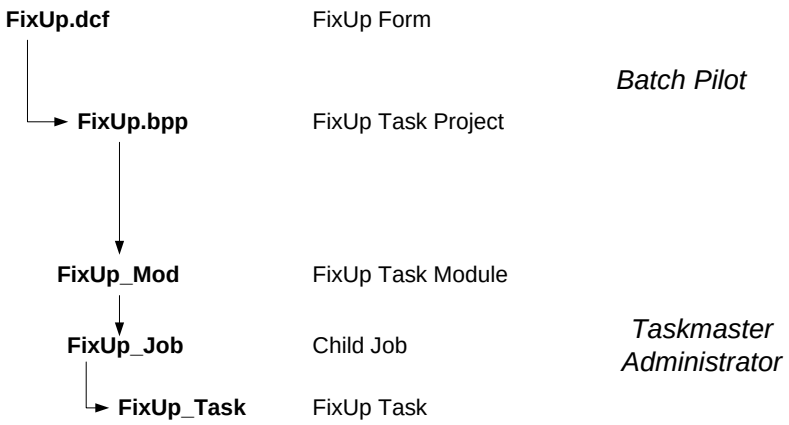
Chapter 6 fully describes the straightforward procedure of adding a ***parent*** job (such as a Main or Demo job) to a workflow.

However, a ***child*** job is different:

- The job and its tasks become available ***only*** in response to an action of a task in the ***parent*** job.
- Upon concluding its work with a batch, the ***child*** job must return the batch to the ***parent*** job.
- This ability to form a parent/child relationship is an attribute of the ***child*** job and, therefore, a property of its Job Definition.
- The definition of a ***child*** job is complete ***only*** when it includes the designation of an opening task that functions in a FixUp role, or in another, clearly delineated secondary role.

These considerations mean that defining the ***child*** job is a process outlined in the chart on the next page:

- ***Very important!*** The definition of a ***child*** job begins with the definition of the opening task's Task Project in the *Batch Pilot* environment.
- The second stage creates a Task Module for the iFixUp or kFixUp task.
- The third stage defines the ***child*** job.
- The concluding stage defines the ***child*** job's opening task and links it to the job.



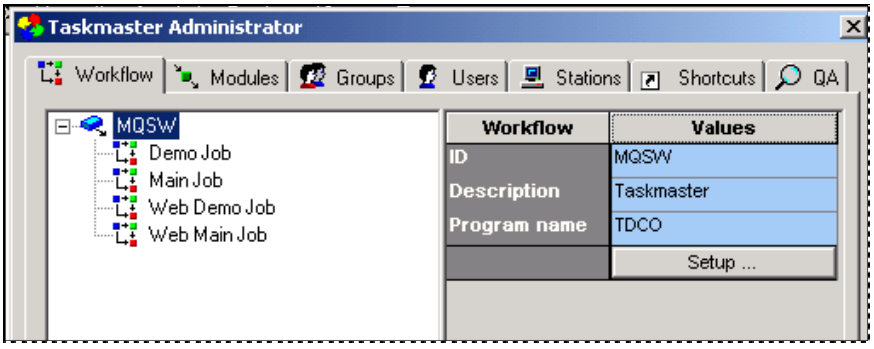
To define a ***child*** job, take these steps:

Step	Action

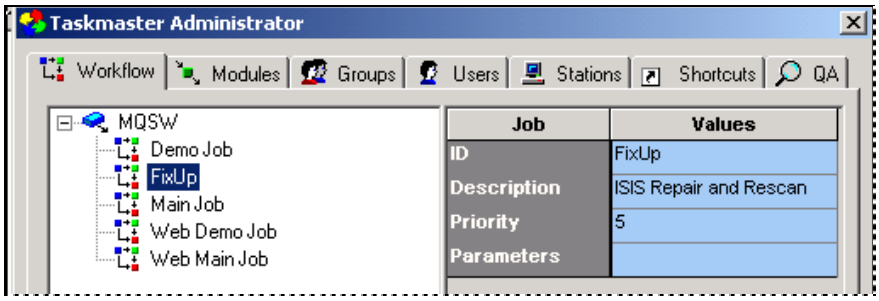
1. Define the **FixUp Task Project** (.bpp) by following the steps on Page 25 (kFixUp) or Page 49 (iFixUp). Keep in mind that your application's Document Hierarchy (.xml) is a ***required*** property of the new Task Project – and that the Task Project alos requires a FixUp form (.dcf).

2. In the *Modules* tab of the ***Task Master Administrator***, define a Task Module to link the FixUp Task Definition to the Task Project - Page 29 (kFixUp) and Page 53 (iFixUp).

3. In the *Workflow* tab, highlight the application's Worlflow component in the **Components** area on the left and click on the Add button at the bottom of the tab.



4. Enter the ***child*** job's Job ID (illustrated on the next page.)



To Define a Child Job (continued)

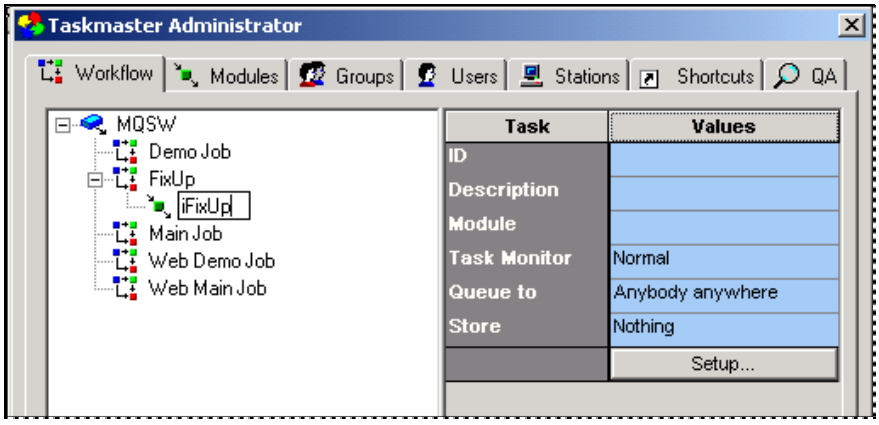
Step	Action

5. Add a brief but ***important*** **Description** in the **Values** column of the **Properties** area on the right.

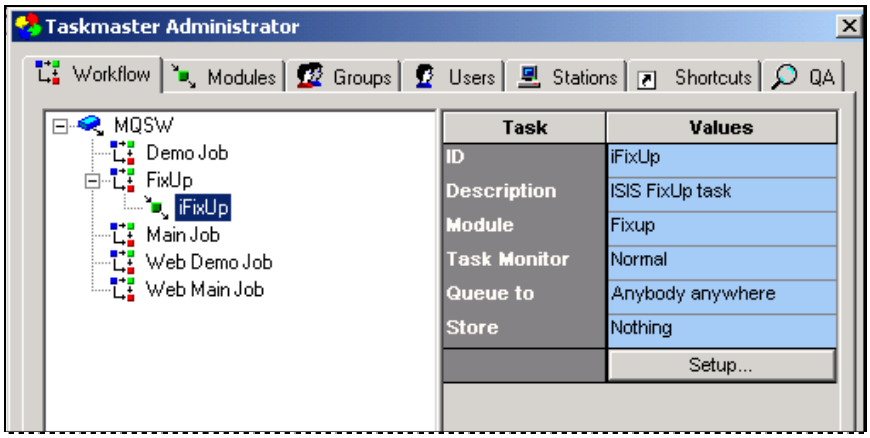
6. Press the Apply and Done buttons at the bottom of the tab. Be sure the new Job ID continues to be listed in the **Components** area.

7. Highlight the ***child*** job's Job ID and click on the Add button.

8. Enter the Task ID you're assigning to the job's opening task.



9. Add a brief but ***important*** **Description** of the task – an iFixUp task, in this examples.



To Define a Child Job (continued)

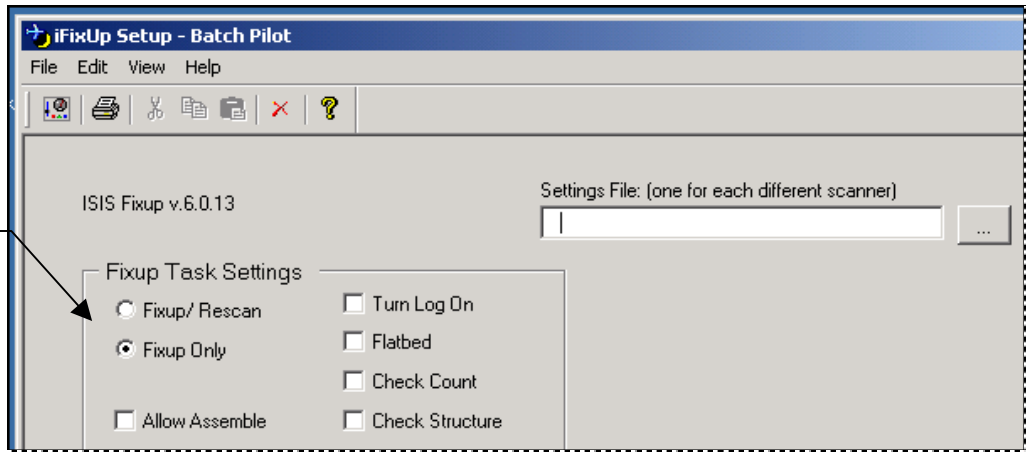
Step	Action

10. Select the Task Module you defined in Step #2 from the **Module** drop-down list.

11. Press the Apply and Done buttons at the bottom of the tab. Be sure the new Task ID continues to be listed in the **Components** area.

12. Press the Setup button to assign settings to the kFixUp task (Page 33) or iFixUp task (Page 57).

Processing modes



✓ ***Alert!*** It's not too early to decide on the FixUp task's processing **mode**. Will the task's scope be limited to batch organization problems (**FixUp Only** in the example above) or can it physically re-scan paper as well (the **FixUp/Rescan** option above). This determination has an immediate impact on the steps you take to configure a kFixUp task (Page 23) or an iFixUp task (Page 46).

kFixUp Tasks – Setup and Operation

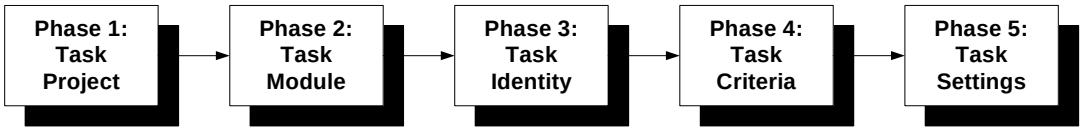
A **kFixUp** task has a ***setup*** component and a distinctly different ***runtime*** component. You work with the ***setup*** component when you define the task or modify its Task Definition. A qualified FixUp operator uses the ***runtime*** component when he or she reviews, repairs and sometimes re-processes problem batches.

✓ ***Important!*** A kFixUp task is closely related to a workflow's kScan task (Chapter 7). Many features are identical; for example, if you assign **Fixup/Rescan** capabilities to the kFixUp task, the task may employ the same source device and probably the same scanner as the kScan task. Other ***setup*** properties differ, however, and are described in the following sections.

kFixUp Tasks - Setup

Like most tasks, a kFixUp **Task Definition** needs a Task Identity, a Task Project, a Task Module and basic setup specifications. (For a complete review of Task Definition components, see Chapter 6.)

The configuration of a new kFixUp task is a process with five phases. ***Alert!*** Chapter 6 describes details of this process. This section examines those elements of task configuration that are associated with the setup of kFixUp tasks.



Phase 1 (on the next page) adds a kFixUp **Task Project** (.bpp) to your application's **Process** directory.

During Phase 2 (Page 29), you'll define the **Task Module** that will connect your kFixUp task to the Task Project you established in Phase 1.

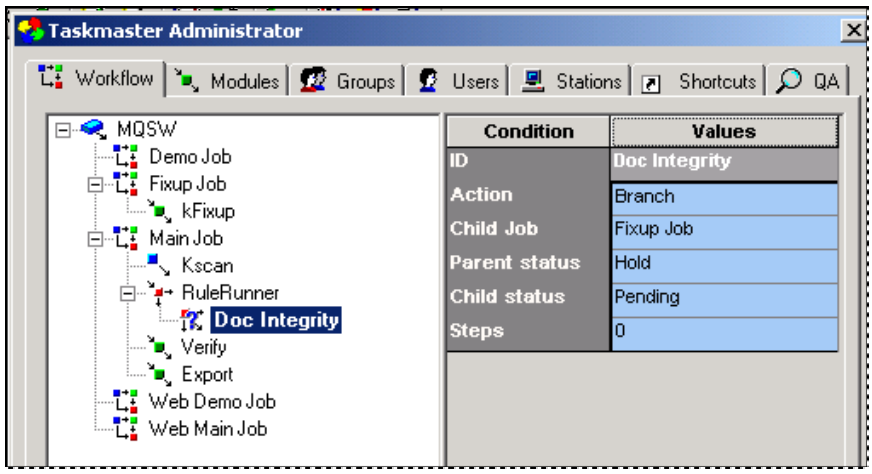
Phase 3 provides the kFixUp Task Definition with a formal identity and assigns key properties to the task itself (Page 31).

Phase 4 uses the ***kFixUp Setup*** dialog to assign a KOFAX-certified scanner to the task – if the task can rescan problem pages - and operating specifications to the task and scanner, if applicable (Page 33).

Phase 5 reviews and modifies values in the tabs of the ***Task Settings*** dialog (Page 38).

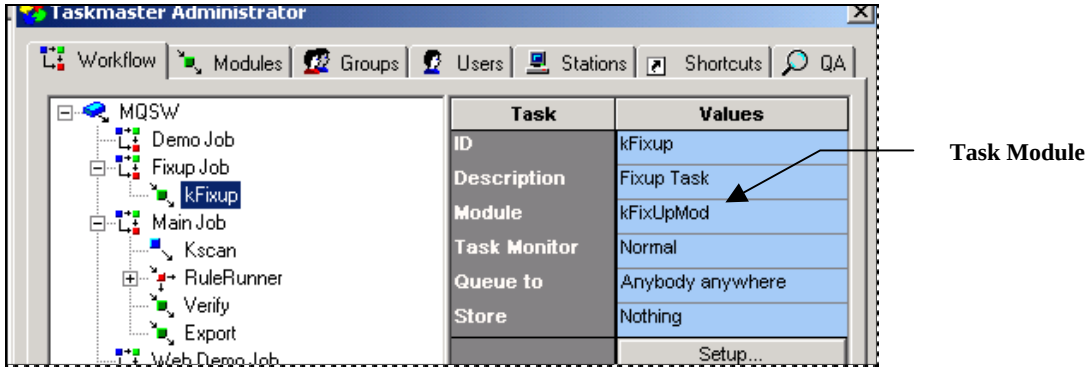
The ***setup*** specifications for a kFixUp task will be part of its Task Project (.bpp). You can add them to the Task Project after you have a Task Module to connect the Task Definition to the Task Project, and have the established a Task Identity that designates the Task Module as a key property (Page 29).

The following illustrations show how the RuleRunner task of the fictional *MQSW* application's Main job will branch to the FixUp job if the task encounters a "Doc Integrity" condition.

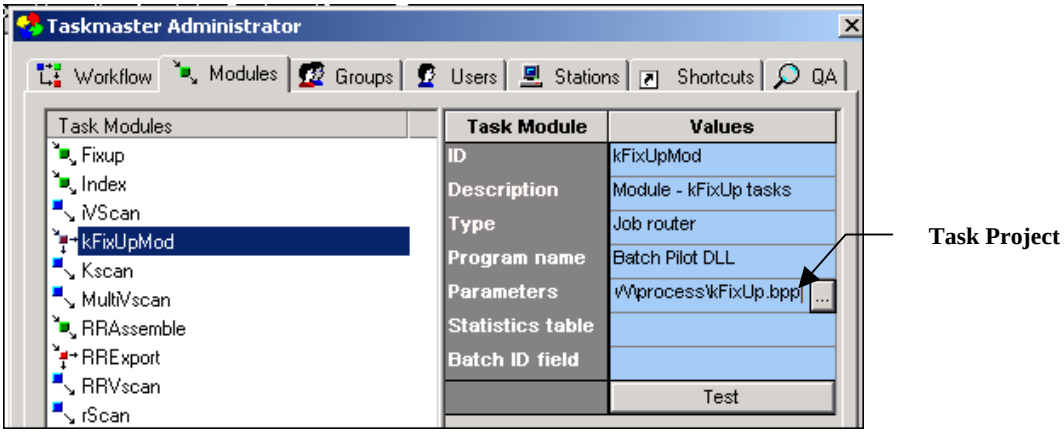


MQSW Taskmaster Administrator – *Workflow tab*

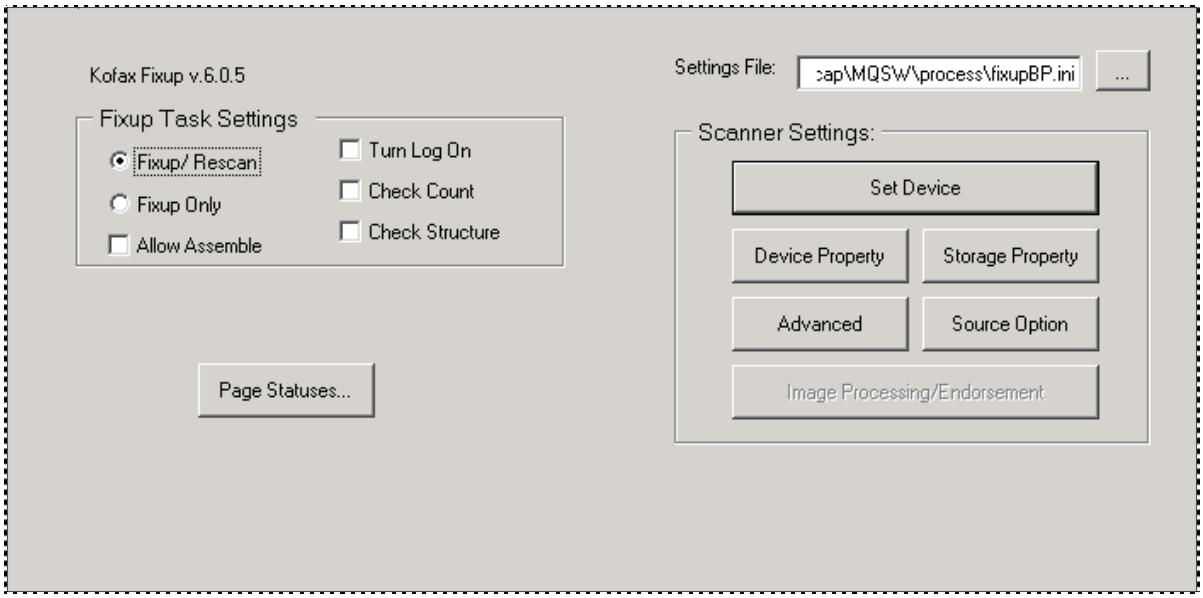
The FixUp job's *kFixUp* task employs the *kFixUpMod* Task Module:



The Task Module, in turn, links the kFixUp Task Definition to the *kFixUp* Task Project (.bpp) that you have ***previously*** assembled.



Now, if you highlight, the *kFixUp* Task ID in the **Components** area of the ***Taskmaster Administrator's*** *Workflow* tab, and click on the ***Setup*** button in the **Properties** area, the Task Project's ***Setup*** dialog will appear on your screen. (The section that begins on Page 33 describes the features and options of the kFixUp task's ***Setup*** dialog.

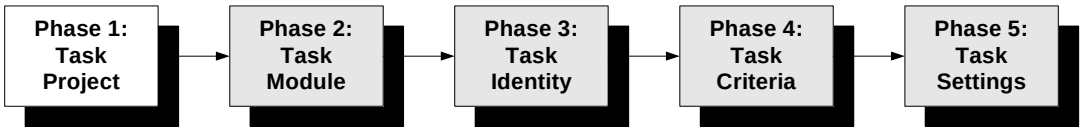


kFixUp Setup Specifications – *FixUp and Rescan Modes*

✓ The *Application Wizard Guide* shows you how to use the **Taskmaster Application Wizard** to install and develop a new application such as *MQSW*.

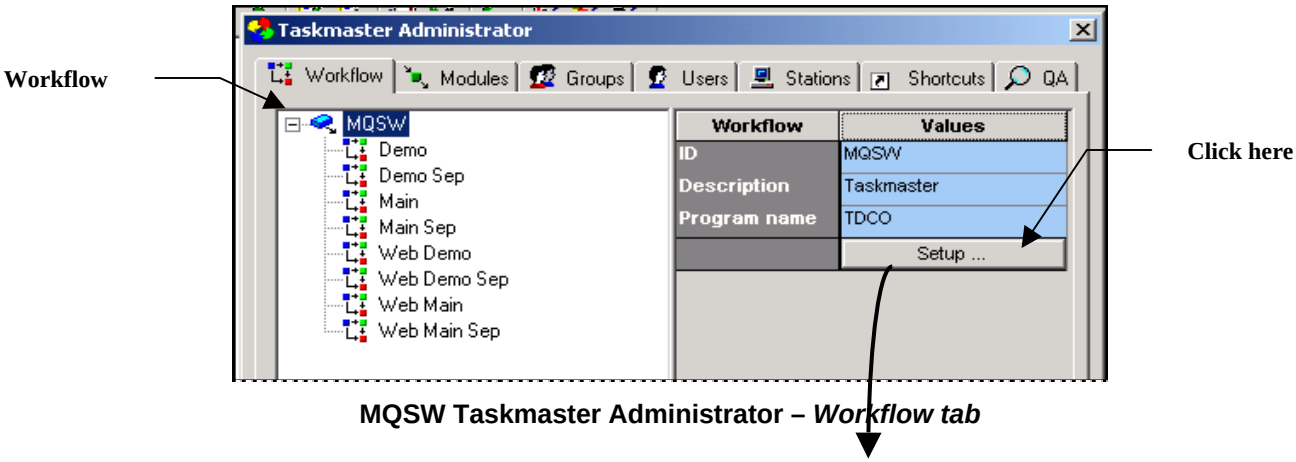
kFixUp Setup – Phase 1: Task Project

The definition of a kFixUp task begins with its **Task Project.**

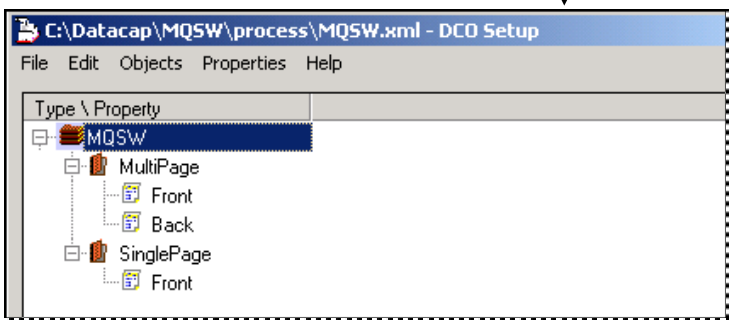


The Task Project contains the task's ***setup*** and ***runtime*** forms – as well as the software that runs the task. The Task Project is also a file (.bpp) – a file that includes the criteria and settings you assign in Phase 4 and Phase 5, as well as the forms.

1. ***Important!*** Your application's Document Hierarchy is a ***required*** property of a Task Project. By the time you start setting up FixUp tasks, the Document Hierarchy file (.xml) will be firmly in place – and immediately available to you (see Chapter 3 and Chapter 6).



MQSW Taskmaster Administrator – *Workflow tab*



Document Hierarchy Setup Window
MQSW.xml

To Assemble a kFixUp Task Project

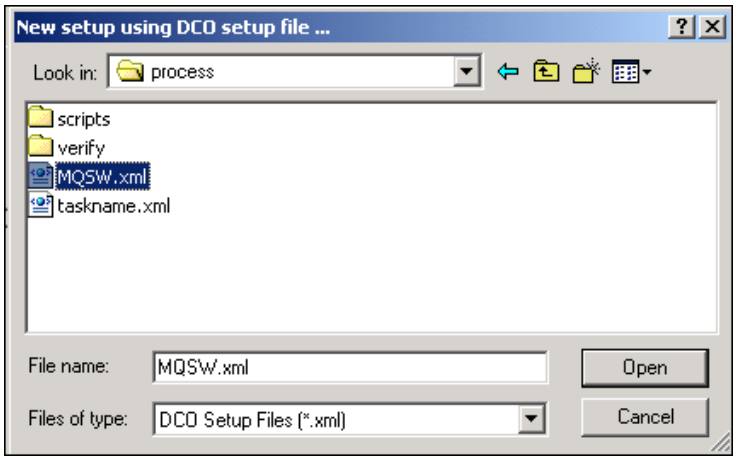
Step	Action

1. Be sure that the **Workflow** component of the Workflow Hierarchy that will contain the task includes a Document Hierarchy file (.xml).

2. Select **Datacap Taskmaster** from your Windows Start button's **Programs** options.

3. To open the ***Batch Pilot Window***, double-click on the **Batch Pilot** icon in the **Batch Pilot** folder. (*Batch Pilot* is the workshop you'll use to put together the Task Project in this phase, and assign task properties in Phase 4 and Phase 5.)

4. Select **New Project** from the window's **File** menu. *Batch Pilot* will instantly ask you to enter the name and path of the Document Hierarchy file in the ***Open File*** dialog. Use the window's **Form** menu to be sure the project is ***not*** in **Design** mode.)



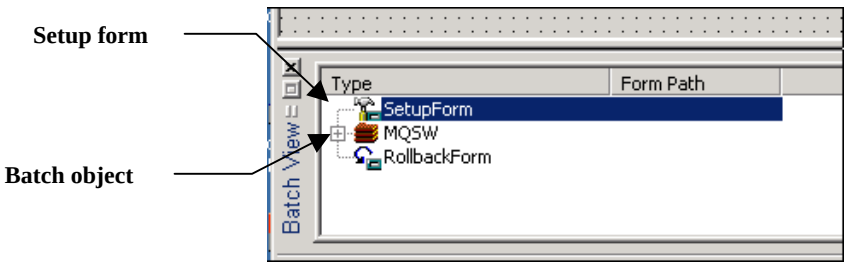
New Project Setup -
Document Hierarchy file

To Assemble a kFixUp Task Project (continued)

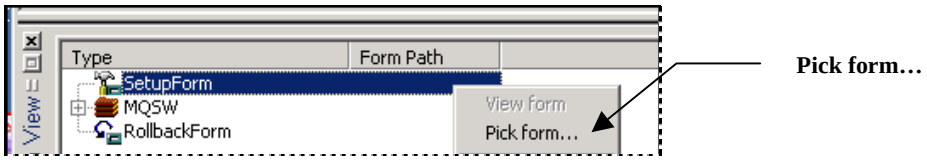
Step	Action

5. 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***. (For thorough explanations of *Batch Pilot* and the ***Batch Pilot Window***, you can click on the Help button at the top of the window, or refer to the *Guide to Batch Pilot*.)

6. 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.)



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

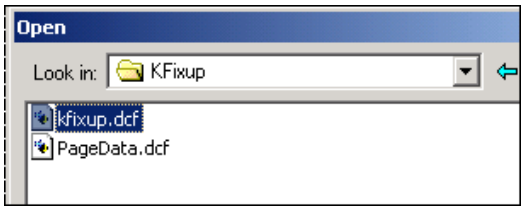


To Assemble a kFixUp Task Project (continued)

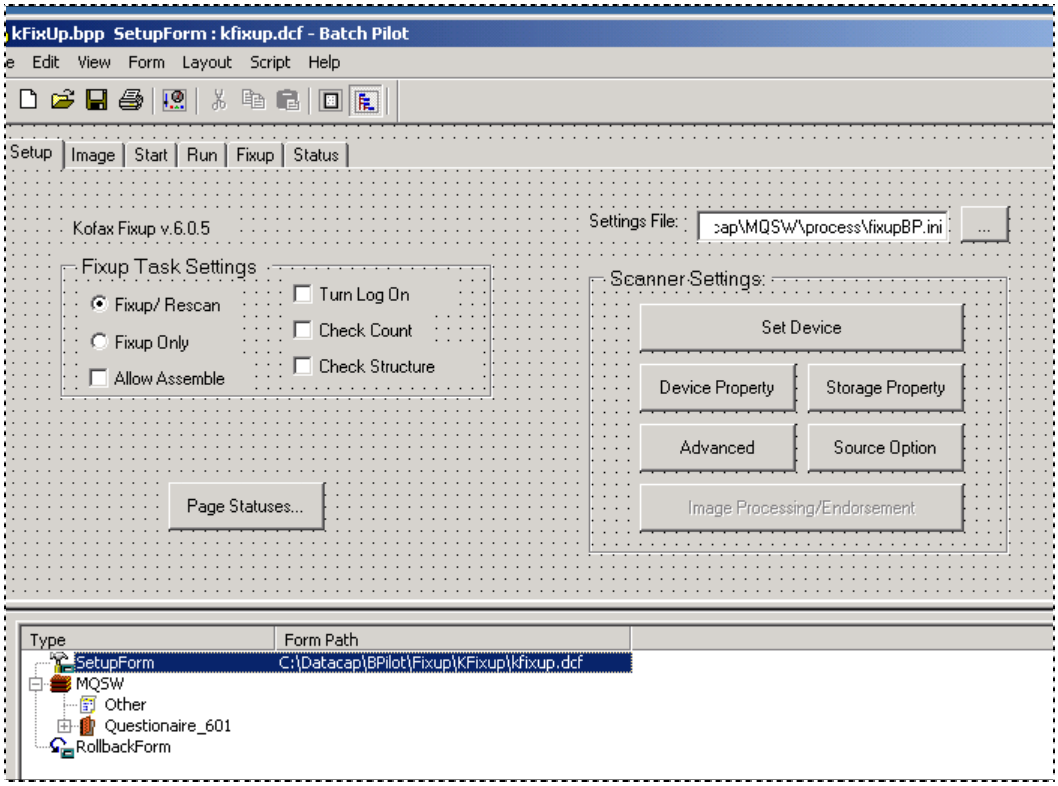
Step	Action

8. Use the ***Open File*** dialog to navigate to the **Datacap** directory's **BPilot** folder.

9. Select **kfixup.dcf** from the **FixUp** folder's **kFixUp** sub-folder, and press the dialog's **Open** button.



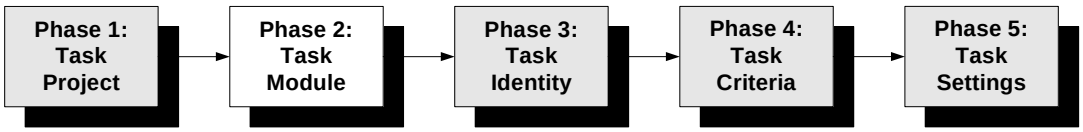
10. When you return to the ***Batch Pilot Window***, check the file name and path in the **Form Path** column of the **Batch View** area.



11. Select **Save As** from the **File** menu to save this Task Project in your application's **Process** directory - and make it ***instantly available*** to a kFixUp Task Definition (Page 31).

kFixUp Setup – Phase 2: Task Module

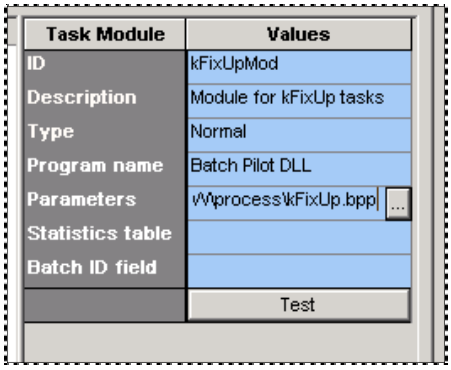
A Task Module connects the kFixUp Task Definition to its Task Project.



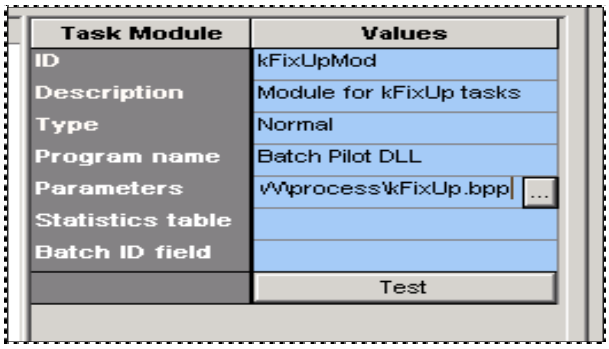
When you're setting up a new kFixUp task – and do not yet have a Task Module – you'll take the steps below steps to define it. ***Remember!*** You cannot define a Task Module until a Task Project is firmly in place (Page 25).

Step	Action

1. Open the *Modules* tab of your application's ***Taskmaster Administrator***.
2. Click on the Add button to clear the fields in the **Values** area on the right.
3. Enter a unique Module **ID** and a brief but important **Description** of the module.



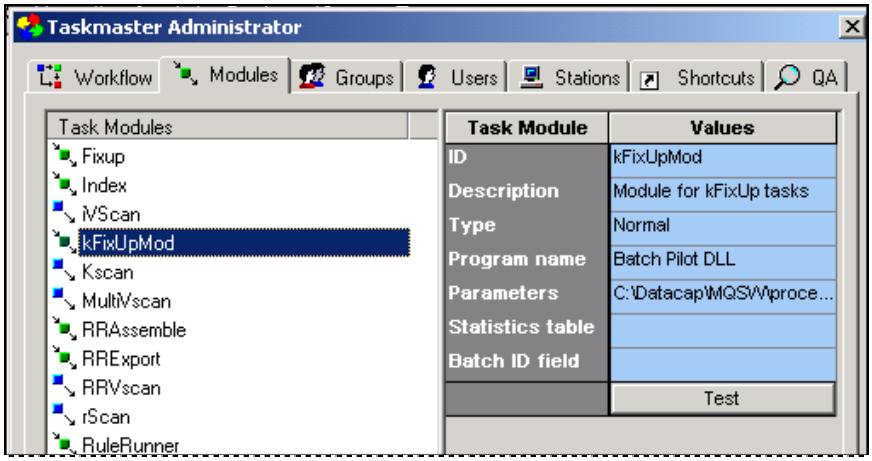
4. Select *Normal* from the **Type** drop-down list.
5. Select *Batch Pilot DLL* from the **Program Name** drop-down list.
6. Click once in the **Parameters** field to display the field's Browse button. Click on the Browse button to retrieve the ***Open File*** dialog and select the Task Project file (.bpp) you assembled in Phase 1 (Page 25).



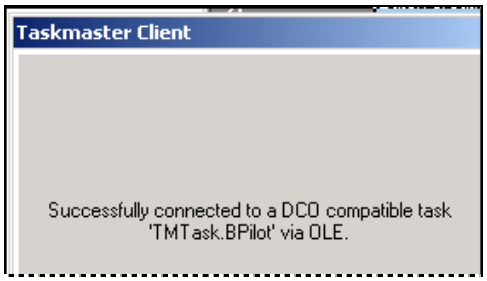
To Define a kFixUp Task Module (continued)

Step	Action

7. 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.

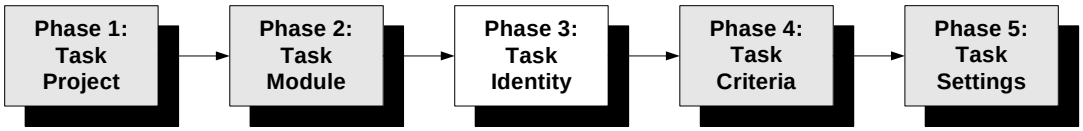


8. Press the Test button. If the connection between the Task Module and Task Project is secure, you will receive this technical message:



kFixUp Setup – Phase 3: Task Identity

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



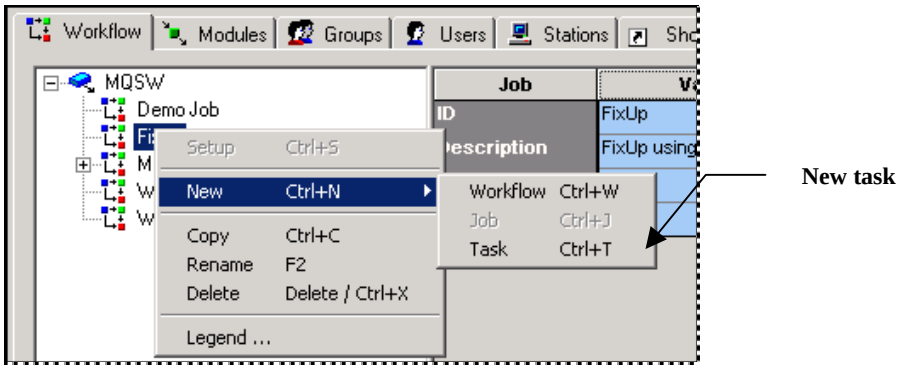
To provide a kFixUp task with its identity:

Step	Action

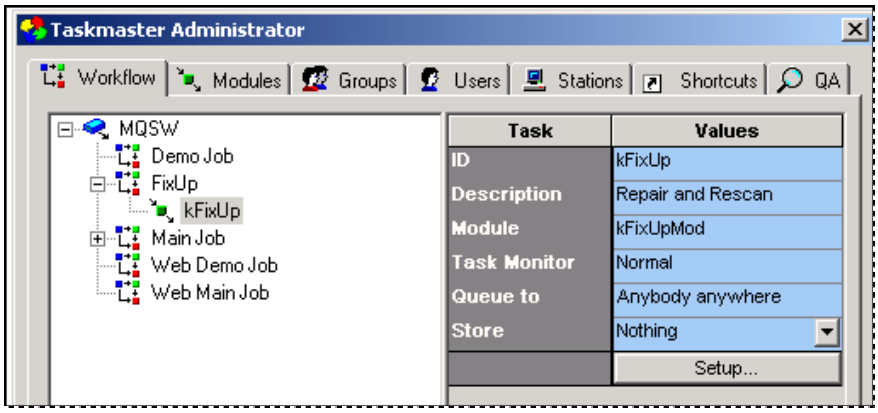
1. Open the ***Taskmaster Administrator's*** *Workflow* tab.

2. Right-click on the ***child*** job that you previously defined (Page 16) –and which will contain the kFixUp task (*FixUp*, in this example.)

3. Select **New** and **Task** from the options.



4. Enter a ***unique*** Task ID in the open space below the Job ID (illustrated on the next page). Be sure this value appears in the **ID** field of the **Values** area as well.



To Provide a Task Identity (continued)

Step	Action

5. In the **Values** area, enter a brief but ***important*** **Description** of this kFixUp task.

6. From the **Module** drop-down list, select the ID of the Task Module you defined in Phase 2 (Page 29).

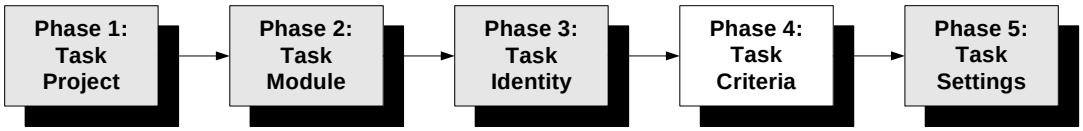
7. Do not alter the ***default*** value of the **Queue to** property. If operations of subsequent tasks within this ***child*** job are to be restricted to the operator or workstation that launches ***this*** task, select one of the values in the **Store** property's drop-down list. (Chapter 6 and *Taskmaster* Help explain the **Queue to** and **Store** properties.)

9. Press the ***Taskmaster Administrator's*** Apply button to save the identifying properties of the kFixUp Task Definition.

You can find complete explanations of 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 its **Queue to** and **Store** properties.

kFixUp Setup – Phase 4: Setup Specifications

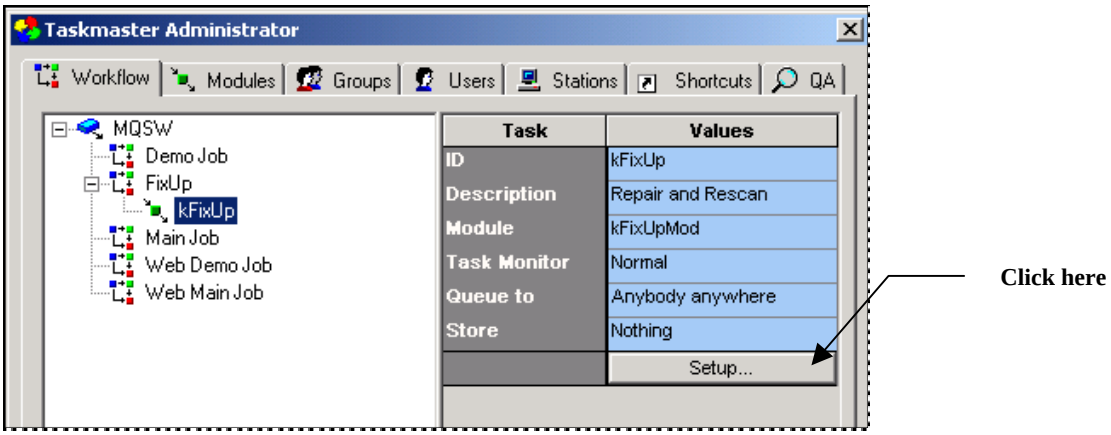
Phase 4 uses the ***kFixUp Setup*** dialog to determine the task's scope – *FixUp Only* or *FixUp and Rescan*.



If the kFixUp task can rescan pages, you'll use this phase to review and possibly modify the scanner settings of the ***parent*** job's kScan task – and to assign new scanner settings.

For either processing mode, you'll evaluate and possibly select specifications that belong only to a kFixUp task.

To begin this phase, highlight the Task ID you established in Phase 4 and press the Setup button in the ***Task Master Administrator's*** *Workflow* tab:



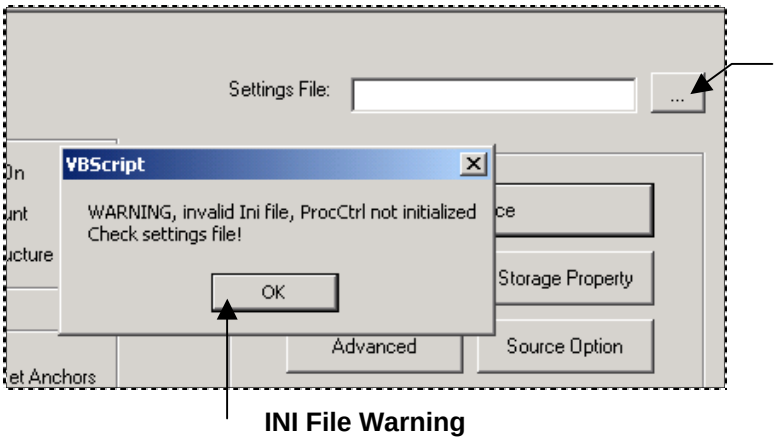
MQSW Taskmaster Administrator – *Workflow tab*

☛ Pay close attention to the **Description** in this example: the Administrator intends to set up this task to remedy images and batch organization ***and***, if necessary, to rescan portions of the batch – or maybe the entire batch.

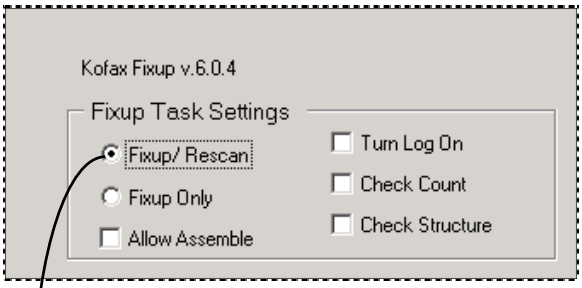
When you first press the Setup button, the warning on the next page will appear on your screen:

To operate, the kFixUp task has to retrieve and update parameters in the **`[ScanCtrl]`** sector of a Settings file (.ini) established for the task's workflow (*MQSW*, in this example.) The Settings file – **mqsw.ini** - is in the application's **Process** directory.

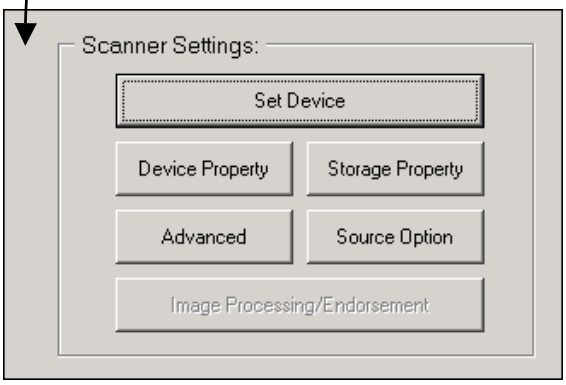
The setup process will give you full access to the ***Setup*** dialog as soon as you enter the file's name and path in the **Settings File** field.



After you have identified the application's Settings file, indicate whether the kFixUp task can carry out FixUp and Rescan procedures (*FixUp/Rescan*) – or if it will be restricted to FixUp procedures (*FixUp Only*). These are radio buttons: you ***must*** select an option.



FixUp Task Scope



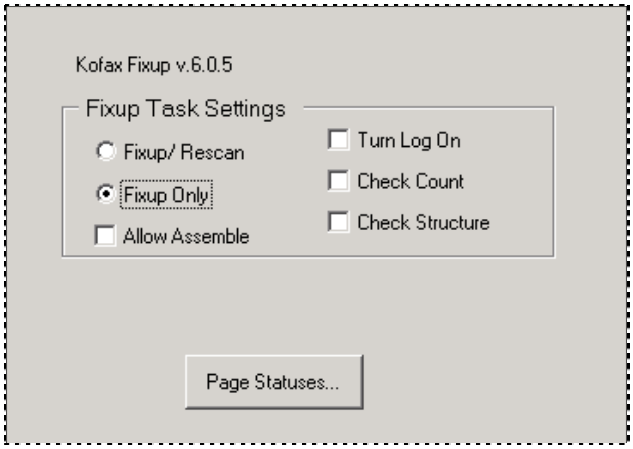
Scanner and Source Device

✓ ***Very important!*** To operate in *FixUp/Recan* mode, the task needs a scanner and a Kofax Source Device. During setup, however, you cannot select a Source Device, or parameters in the **Scanner Settings** area, until you have defined the device (Chapter 7) ***and*** have connected Taskmaster Client to the scanner.

kFixUp Options

The radio buttons in this section determine whether the task will be responsible for both FixUp ***and*** Rescan procedures, or just FixUp.

- **Fixup** activities adjust and enhance pages; move pages round within the batch; re-constitute documents, and re-organize the batch.
- **Rescan** is a physical process that scans one or more pages and replaces existing Image files with the new files.



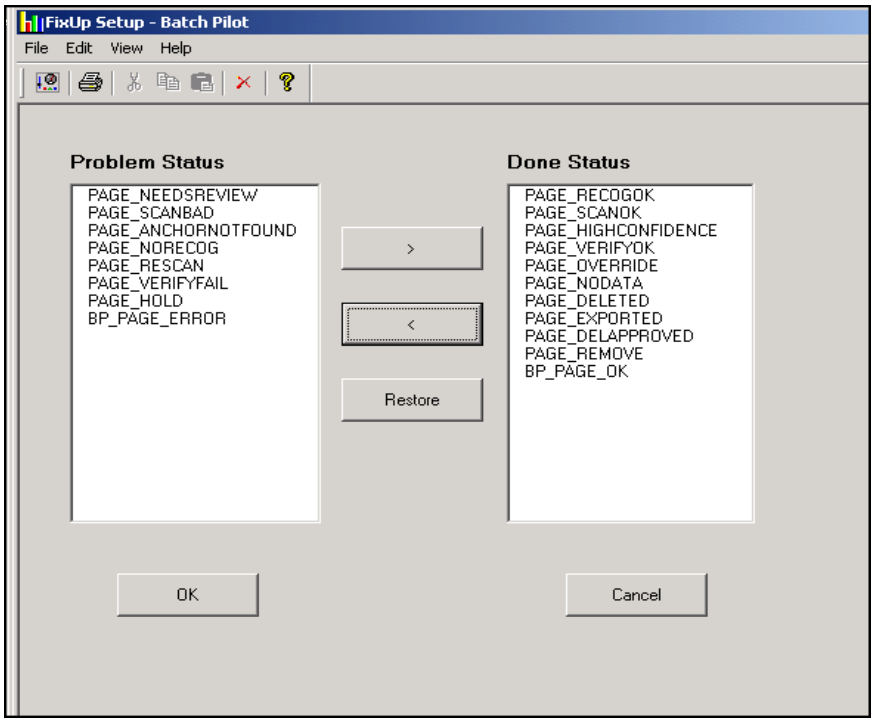
Four additional but *optional* settings in this area can expand the task's scope:

- **Allow Assemble** permits the task to re-assemble the ***batch and its documents*** after FixUp and possible Rescan operations are complete. This option places a Rebuild button on the task's processing dialog (Page 40) and can resolve apparent Document Integrity problems without delay.
- **Turn Log On** instructs the task to create a log of its activities, according to specifications you provide in the ***Log*** tab of the ***Task Settings*** dialog (Chapter 6). A log can be a very helpful tool as you evaluate the task's performance and contributions.
- **Check Count** counts the *actual* number of documents and pages in the batch; compares the totals to the *expected* counts; and alerts the FixUp operator if there are discrepancies. ***Alert!*** This option ***must*** be checked if a **Document and Page Counts** rule is in place (Page 13_.
- **Check Structure** assesses the structure of each document in terms of the types of pages it contains; the total number of pages; and the number of each Page Type. ***Alert!*** This option ***must*** be checked if a **Docoument Integrity** rule is in place (Page 11).

Page Status Button

The Page Statuses button opens a dialog with two fields:

- **Problem Statuses** lists the statuses of pages with "problems." A page with a Problem Status ***cannot*** be processed by other tasks until kFixUp has resolved the problem and the Fixup operator has assigned a *Done* status to ***all*** pages in the batch (Page 40). **Done Statuses** are Page Statuses that an operator assigns to indicate how a problem has been resolved – and to make the current page available for further processing.



kFixUp Setup – Page Statuses

✓ Directional buttons in the ***Page Status*** dialog move statuses from one side to the other. When you press the OK button, the dialog's current listings update the task's Settings file (Page 34).

Alert! A task assigns a status to every page it processes. Placing a Page Status in the **Problem Status** list means that a page with that status ***cannot*** be released from the kFixUp task until the task has taken care of the problem and its operator has assigned a new **Done** status – or the problem page has been deleted. Be careful when you compile the list: too many "problems" can hamper productivity.

Scanner Settings

The right-hand portion of the ***kFixUp Setup*** dialog assigns a Kofax scanner device and scanner criteria to be used exclusively by the kFixUp task's re-scan procedures.

✓ The source device you select and the scanner settings you assign may or may not be the same as those of the kScan task (Chapter 7). However, you must define the kFixUp device before you add these specifications. ***Be sure***, too, that you have connected Taskmaster Client to the scanner, and that the scanner is on, before you attempt to access these settings.

☛ To supply your kFixUp task with a Source Device, you'll take the steps below. Again, you ***cannot*** proceed until you have connected the scanner (with its Kofax image controls) to your computer ***and*** have turned on the scanner.

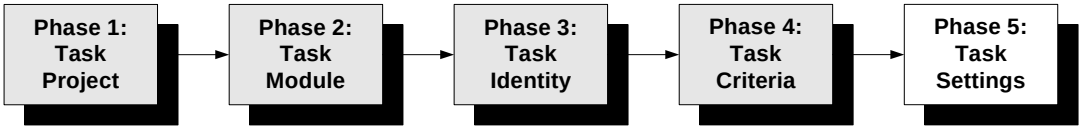
Step	Action
1.	Follow the instructions on Page 33to open the ***kFixUp Setup Specifications*** window (illustrated on the previous page.)
2.	Click on the Set Device button: the ***Select Scan Source*** dialog will appear.
3.	Select the applicable Kofax Source Device from the drop-down list. ***Alert!*** Be sure the Source Device you choose includes the scanner you're working with, as a component.
4.	Click on the OK button at the bottom of the ***Select Scan Source*** dialog: *Taskmaster* will assign the device to the task ***and*** will update the task's Scanner Settings file (.ini) with device information. (For an explanation of this file, see Chapter 7).
5.	Click on the OK button at the bottom of the ***kFixUp Setup Specifications*** window; when the ***Taskmaster Administrator's*** *Workflow* tab appears on your screen, press the Apply button then the Done button.

At this point, if you re-open the ***kFixUp Task Setup*** dialog and click on the Set Device button, the ***Select Scan Source*** dialog will automatically display the device name in its drop-down list. If you press the Properties button, a read-only edition of the ***Select Scanner*** dialog will appear, with information about the scanner and engine.

If you press the Advanced button, the ***Advanced Source Properties*** dialog will appear. Chapter 7 fully explains the settings and options of this dialog.

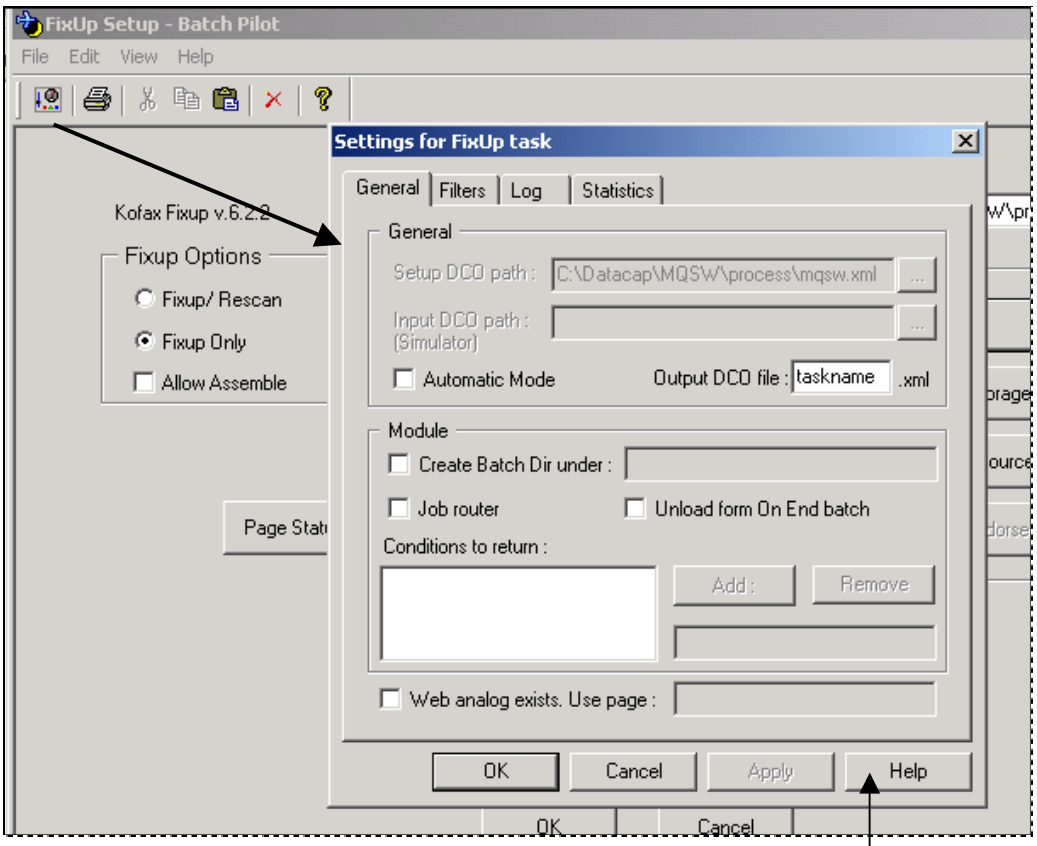
kFixup Setup – Phase 5: Task Settings

The closing phase of a kFixup task's setup involves specifications in the tabs of the ***Task Settings*** dialog.



✓ To access this dialog, select **Task Settings** from the ***Batch Pilot Window's*** **File** menu after you have entered Task Criteria in Phase 4 (Page 33).

The ***Task Settings*** dialog of a typical kFixUp task contains nothing exceptional (see Chapter 6 for a full explanation of this dialog.)

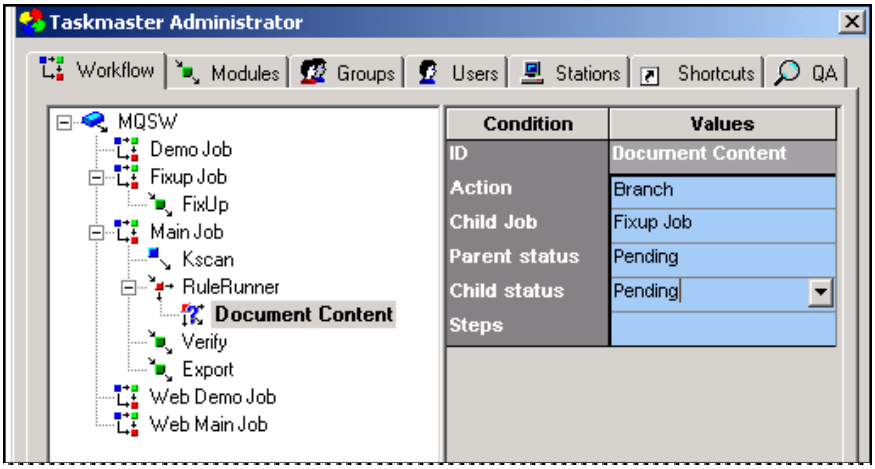


kFixUp Task Settings dialog

kFixUp: Task Parameters in the Parent Job

Alert! Although the kFixUp task has been fully assembled, the task cannot operate until you link its job – a FixUp ***child*** job (Page 16) - to a task of a ***parent*** job.

In the example below, the *MQSW* RuleRunner task belongs to the ***parent*** Main job. The *MQSW* Administrator has taken steps to assign a **Document Content** processing condition to the task – and to define the condition itself (Page 5).



MQSW Taskmaster Administrator – *Branching to FixUp*

To set up a link between the RuleRunner **task** and the FixUp **job** when the task encounters the *Document Content* condition involves these steps:

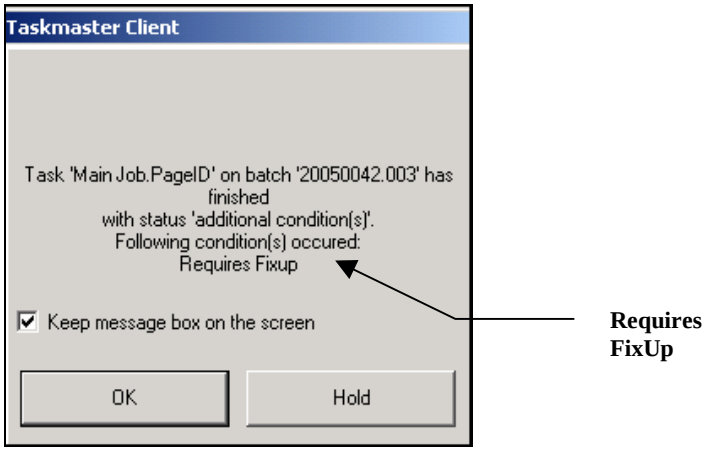
- Highlight the processing condition in the components list on the left.
- Select an **Action** from the drop-down list on the right (*Branch*, in the example). This defines the RuleRunner task's response when it encounters the *Document Content* processing condition.
- From the **Child Job** dorp-down list, select the job to which the batch will be diverted if the task in the ***parent*** job runs into trouble.
- Select the processing status to be assigned to the batch as it awaits further attention by tasks of the ***parent*** job.
- Select the processing status to be assigned to the batch as it awaits the immediate attention of the ***child*** job's kFixUp task.
- If the batch will ***not*** return to the task that sent it to the kFixUp task, use the Steps field to indicate the target task in the **parent** job. "1", for example, would send the corrected batch to the task that's one position after the diverting task – Verify, in the example above.

kFixUp Operations

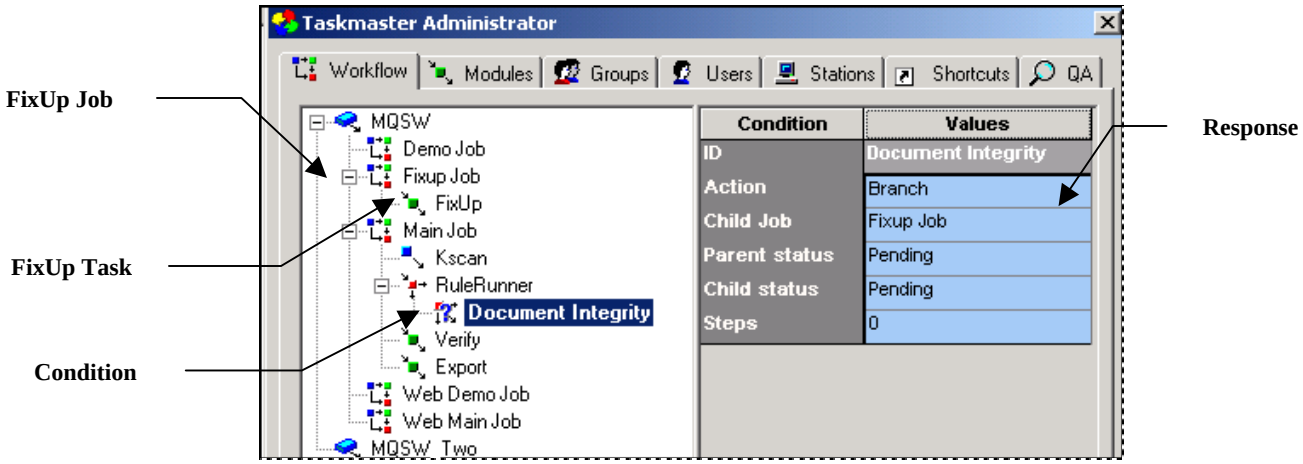
A task in a ***parent*** job – a Recognition task, for example - will dispatch a batch to a FixUp ***child*** job and its kFixUp task ***whenever*** the Recognition task encounters processing conditions that were

- Identified in the ***Task Settings*** dialog (Page 5).
- Linked directly to the FixUp job by settings in the *Workflow* tab of the ***Taskmaster Administrator*** (Page 16).
- Linked to a Fixup rule (Page 9) or
- Linked to a Hot Key of a ***Data Entry*** panel (Page 15).

The first indication of trouble comes when the ***parent*** job's task attempts to process the problem batch, and a warning similar to this appears on the operator's screen:



In the example which follows, an (intentionally!) small batch processed by the RuleRunner task of the *MQSW* application's Main Job has detected a **Document Count** problem (Page 13). As a result, the task branched the batch to the FixUp job and its kFixUp task.



MQSW Task Master Administrator – *Workflow tab*

✓ The ***kFixUp*** panel appears as soon as the FixUp operator launches the task by double-clicking on the applicable Job-Task shortcut in the application's ***Operations*** window.

The first illustration on the next page depicts the full scale of the panel; other illustrations highlight specific portions of the panel.

The panel has three sections:

- **Image View** displays the image of the current page – the page that is now under consideration. (A close look at the example reveals that it is ***not*** the image of a Marketing Survey!) To open this area, select a page in the **Batch View** area (below) or **Image View** from the window's **View** menu.
- **Batch View** lists current ***runtime*** components of the Document Hierarchy at three levels: **Batch**, **Document**, and **Page**. The illustration depicts a batch (*20050084.001*) with four documents (*20050084.001.**01** - 20050084.001.**04***). Each document has one page, numbered sequentially (*TM000001- TM000004*). To open this area, select **Setup Tree** from the **View** menu.
- **FixUp Actions** contains the fields and buttons you use to repair a batch or dispatch an image to the task's rescan procedures (Page 40).

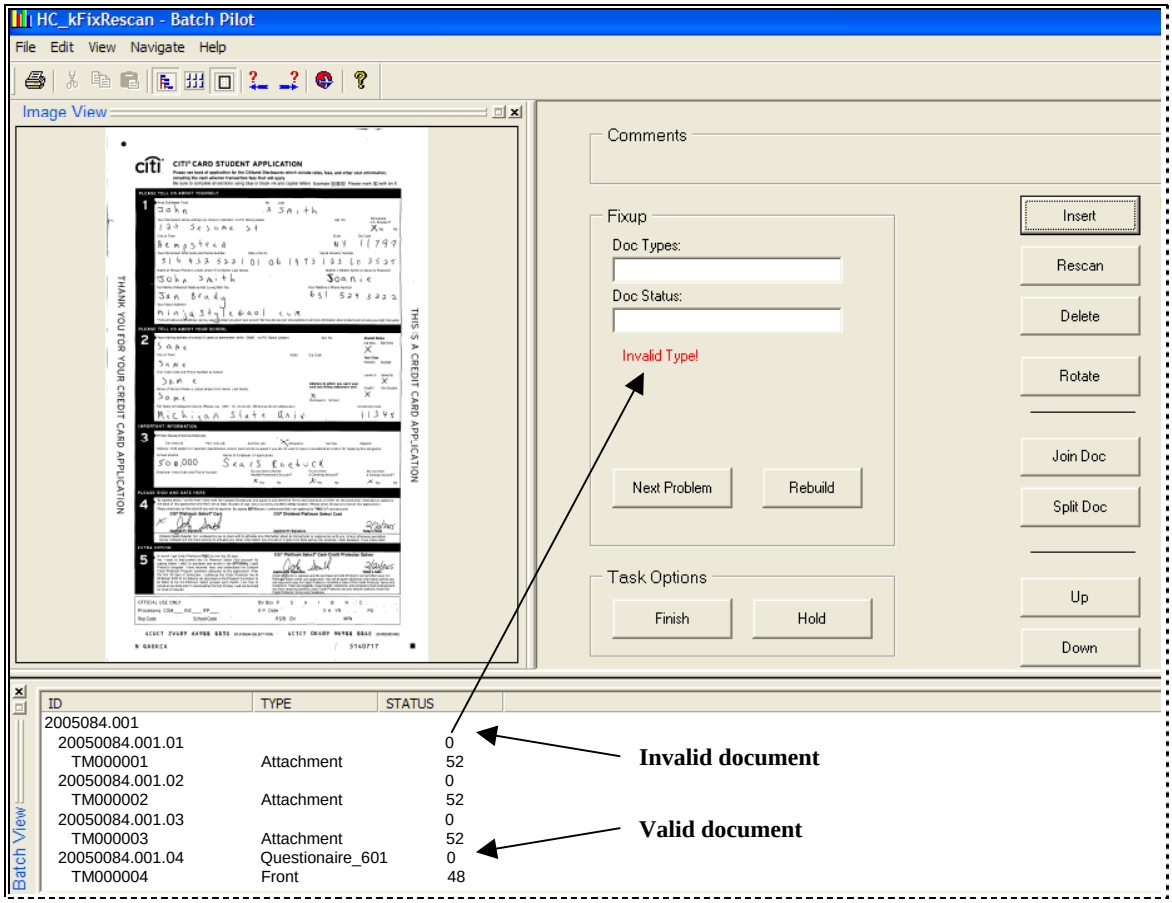
kFixUp Panel – Batch View Area

This is the panel's batch assessment area ***and*** its point-and-click staging arena.

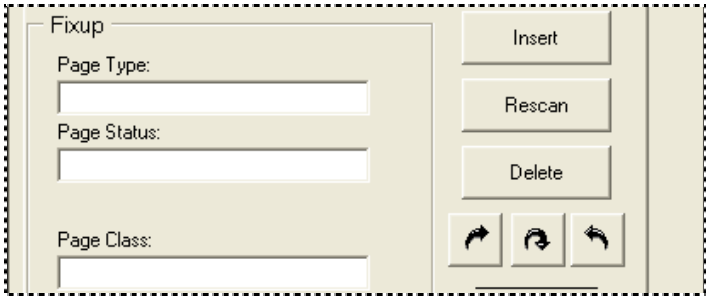
When something is wrong with the batch, you can probably identify the nature of the problem(s) right here – once you've had a chance to learn the values of the application's Page Status and Document Status codes (for a list of default statuses, see Page 74)

In the *MQSW* example, a first look at the items in the hierarchy shows that only the fourth document has a **Type** value (indicates that the claim document has just one *HCFA-1500* ***source*** page.) What about the other documents? Well, each has an *Attachment* page but does not have a *HCFA 1500* ***source*** page – and therefore does not meet the Document

Hierarchy's **Document Integrity** requirements that an *HC Single* document must have one, but only one, *HCFA 1500* page. (Unlimited *Attachments* are permitted as long as the document has that one ***source*** page.) ***Very Important!*** kFixUp operations can handle Document Integrity problems ***only*** if you have selected the **Check Structure** option in the task's ***Setup*** dialog (Page 35).



kFixUp Panel – Document Repair



Page Repair

The **Batch View** area also uses **Status** values to help diagnose problems. Above, the processing status of each *Attachment* page is "52" (*DoesntNeedVerification*). The Page Status of the *Questionaire_601* document's *Front* page is "48"(*Recognition OK*) – an indication of success.

Because the batch has serious flaws, rules governing the RuleRunner task automatically assigned "0" as the Document Status for each document: you'll use the **Doc Status** drop-down list in the **Actions** area to update these values.

kFixUp Panel – Image View Area

Suppose that the first page *Attachment* page is a legitimate *Front* page (instead of a Credit Card application), but that the page was placed in the scanner's tray upside down.

Although the *MQSW* RuleRunner task can have remarkable image adjustment powers, it may still consider the page as unrecognizable, and assign *Attachment* as its Page Type.

However, when the FixUp operator highlights page *TM00001* in the **Batch View** area, the operator will quickly identify it as a legitimate source page – and can use the Rotate button in the **Actions** area to flip the page vertically.

✓ This area provides the operator with essential before-and-after reviews of the pages in a document, and all pages in a batch.

kFixUp Panel – Repair Tools Area

The fields and tools of the **Repair Tools** area work in concert with the listings and values in the **Batch View** area to repair a batch and its contents. ***Alert!*** The roles and titles of certain features change in response to your selection of a document or page in the **Batch View** area.

The **Repair Tools** area of the ***kFixUp*** panel includes:

Comments

This field displays messages that further explain the nature of an existing problem.

A message appears in this field when the FixUp operator clicks on the Next Problem button (below).

FixUp

These first-level tools give the FixUp operator a chance to intervene without delay to carry out basic remedial procedures

Doc Types/Page Types is a drop-down list of the Document Hierarchy's **Document** objects or **Page** objects (Page 10). Remedying a problem may only mean assigning the correct **Type** value to a page or document that has been highlighted in the panel's **Batch View** area.

Doc Status/Page Status is a drop-down list of the application's Document Statuses or Page Statuses. Again, fixing a batch may only require the assignment of a valid processing status – a status that will allow a task in the ***parent*** job to continue working with the page or document, and with the overall batch. Page 36 describes these statuses.

Page Class is a drop-down list of fingerprint classes (Chapter 4). If a Recognition or Verify task in the ***parent*** job has failed to match a page with a fingerprint, the FixUp operator can help out by designating the application fingerprint class for the page.

Problem Alert. A message in this field displays the nature of a problem associated with a particular document or page that the operator has highlighted in the **Batch View** area. (In the example, *Invalid Type* warns the operator that Document 20050084.001.01 lacks a valid **Type** specification.)

Next Problem button. Clicking on this button moves the operator immediately to the next problem document or page. ***Important!*** In a typical batch with hundreds of pages, this button saves the FixUp operator considerable time and frustration.

Rebuild button. Re-builds the batch to accommodate the operator's adjustments. The button can be used at any point in the FixUp process; a series of messages indicate the success or failure the re-building procedures. ***Alert!*** This button is available ***only*** if the **Allow Assemble** option has been selected in the ***FixUp*** dialog (Page 35).

Task Options

Buttons in this area manage the kFixUp task itself.

Finish button. Confirms that all problems have been resolved before assigning a *Finished* status to the batch – or issues an Error Message if problems remain.

Hold button. Places the batch on *Hold.*

Scan Buttons

✓ ***Important!*** These buttons operate on individual pages and are ***not*** available if the task configuration does not include scanning capabilities (Page 35).

Insert scans a new page and ***adds*** it after the page that has been highlighted in the **Batch View** area. Clicking on this button opens a dialog that asks the operator for a **Page Data Prefix** value, and a **Counter Start** value.

Rescan scans a page and uses it to replace the current, highlighted page.

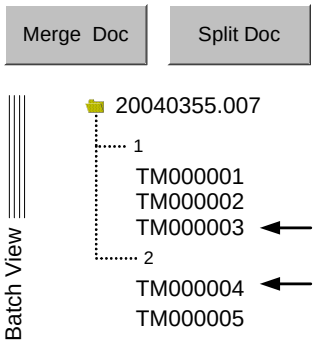
Page and Document FixUp Buttons

After a proper warning, the **Delete** button removes a highlighted page.

The **Rotate** button rotates the image of the highlighted page 90^0.

Join and Split Buttons

If the task of the ***parent*** job has organized the batch into a series of documents – each with its own pages – you can **merge** a document with the document ***above*** it in the **Batch/Doc/Page** hierarchy. In the example, if you select the **TM000004** page and click on the Join button, kFixUp will merge the first two documents into one.



Alternatively, you can split a document into two. In the example, if you highlight *TM0003* and press the Split Doc button, kFixUp will re-organize the batch into three documents. Document 1 will have two pages; Document 2 will have one; and Document 3 will have its original two pages.

✓ Although the Join and Split buttons require practice, they can add significantly to an operator's skills and just as significantly reduce the time he or she needs to repair a batch.

Up and Down Buttons.

These buttons move a page up or down within the hierarchy.

iFixUp Tasks – Setup and Operation

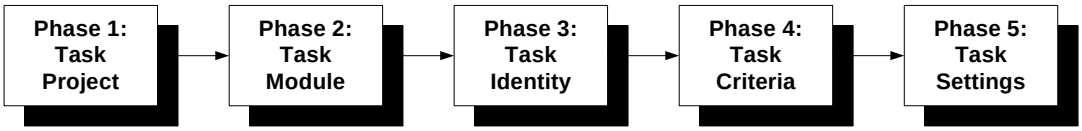
Like a kFixUp task, an **iFixUp** task has a ***setup*** component and a distinctly different ***runtime*** component. You work with the ***setup*** component when you define the task or modify its Task Definition. A qualified FixUp operator uses the ***runtime*** component when he or she reviews, repairs and sometimes re-processes problem batches.

✓ ***Important!*** An iFixUp task is closely related to a workflow's iScan task (Chapter 7). Many features are identical; for example, if you assign **Fixup/Rescan** capabilities to the iFixUp task, the task may employ the same source device and probably the same scanner as the iScan task. Other ***setup*** properties differ, however, and are described in the following sections.

iFixUp Tasks - Setup

Like most tasks, an iFixUp **Task Definition** needs a Task Identity, a Task Project, a Task Module and basic setup specifications. (For a complete review of Task Definition components, see Chapter 6.)

The configuration of a new iFixUp task is a process with five phases. ***Alert!*** Chapter 6 describes details of this process. This section examines those elements of task configuration that are associated with the setup of iFixUp tasks.



Phase 1 (on the next page) adds an iFixUp **Task Project** (.bpp) to your application's **Process** directory.

During Phase 2 (Page 53), you'll define the **Task Module** that will connect your iFixUp task to the Task Project you established in Phase 1.

Phase 3 provides the iFixUp Task Definition with a formal identity and assigns key properties to the task itself (Page 55).

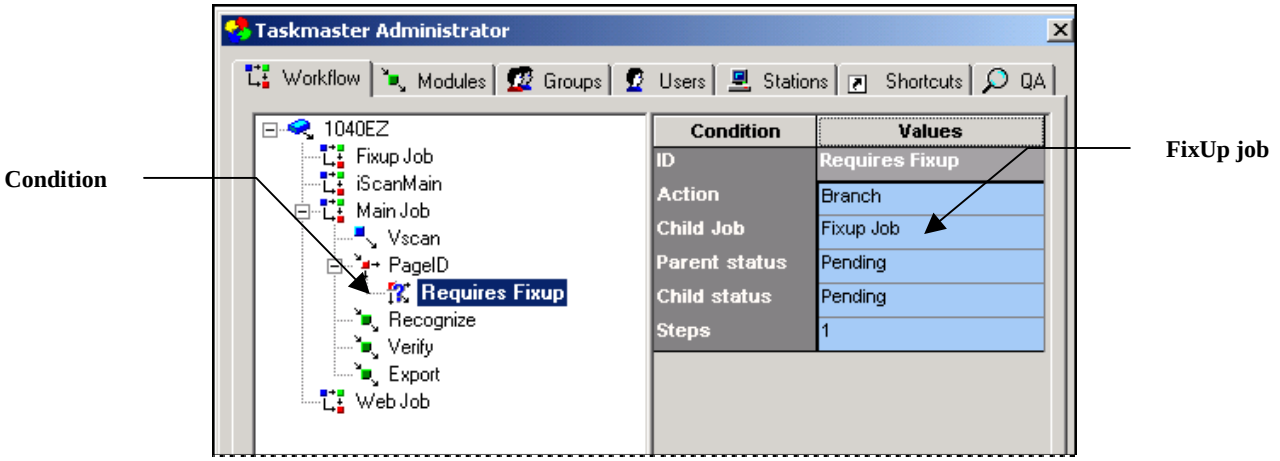
Phase 4 uses the ***iFixUp Setup*** dialog to assign an ISIS-certified scanner to the task – if the task can rescan problem pages - and operating specifications to the task and scanner, if applicable (Page 57).

Phase 5 reviews and modifies values in the tabs of the ***Task Settings*** dialog (Page 57).

The ***setup*** specifications for an iFixUp task will be part of its Task Project (.bpp). You can add them to the Task Project after you have a Task Module to connect the Task Definition to the Task Project, and have the established a Task Identity that designates the Task Module as a key property (Page 53).

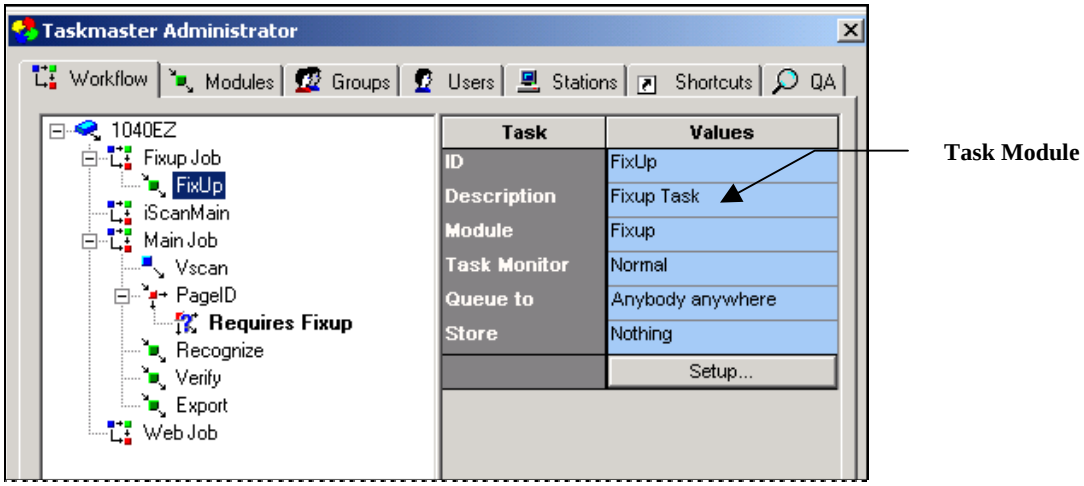
The following illustrations show how the PageID task of the pre-configured *1040EZ* application's Main Job will branch to the FixUp Job if the task encounters a "Requires FixUp" condition.

✓ ***Remember!*** Processing conditions can combine rules that have been established by the application's *Rule Manager* with specifications in the ***Task Settings*** dialog of the ***parent*** job's task – PageID, in the example below. For a full explanation of processing conditions, see Page 5.



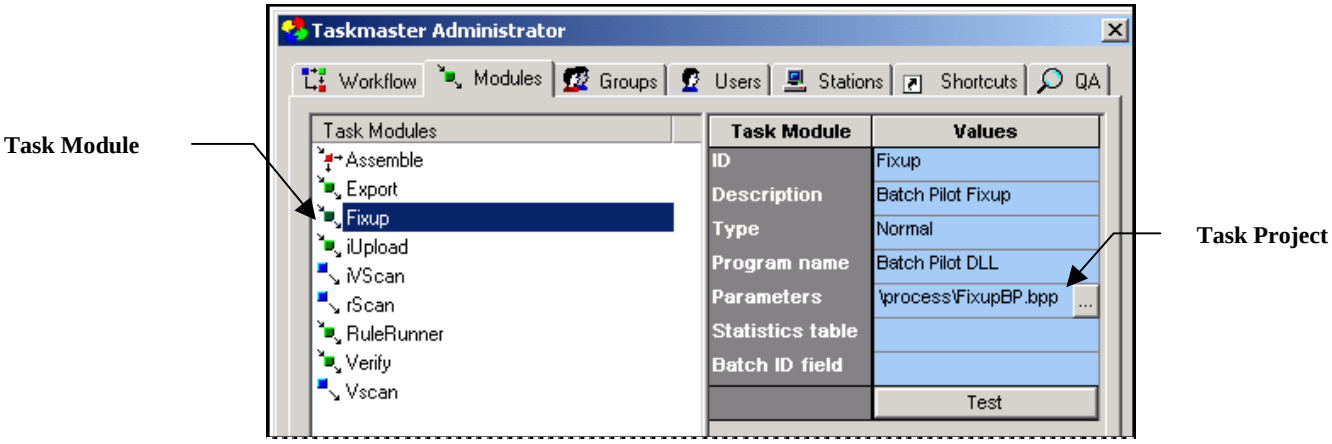
1040EZ Taskmaster Administrator – *Workflow tab*

In this example, the FixUp **Job's** FixUp **task** employs a FixUp **Task Module:**



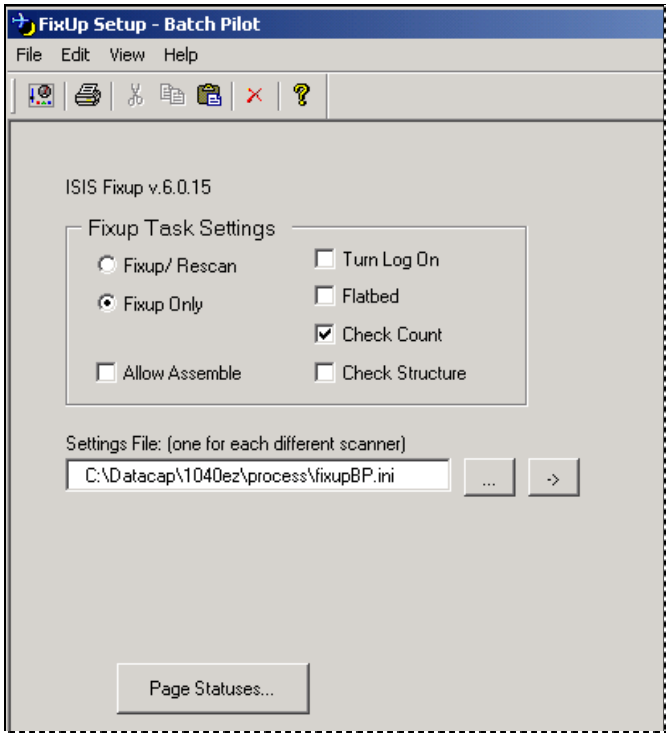
1040EZ Taskmaster Administrator – *Workflow tab*

The Task Module, in turn, links the iFixUp Task Definition to the iFixUp Task Project (.bpp):



1040EZ Taskmaster Administrator – *Modules tab*

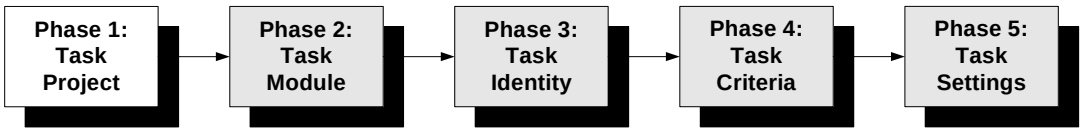
If you highlight the *FixUp* Task ID in the **Components** area of the ***Taskmaster Administrator's*** *Workflow* tab, and click on the ***Setup*** button in the **Properties** area, the Task Project's ***Setup*** dialog will appear on your screen. Or, to be more precise, the dialog's left-hand portion will appear: if you select **FixUp/Rescan** instead of **FixUp Only** in the **FixUp Task Settings** area, the right-hand side will display various scanner settings (Chapter 7).



iFixUp Setup Specifications - *FixUp Only*

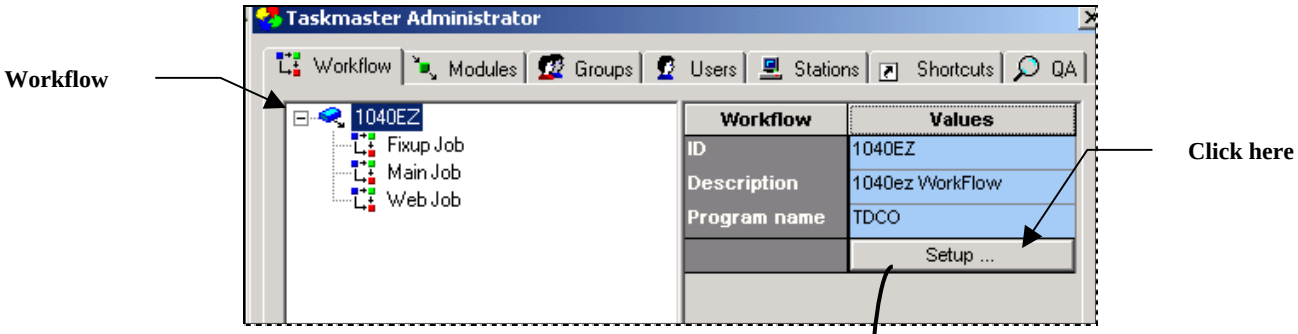
iFixUp Setup – Phase 1: Task Project

The definition of an iFixUp task begins with its **Task Project.**

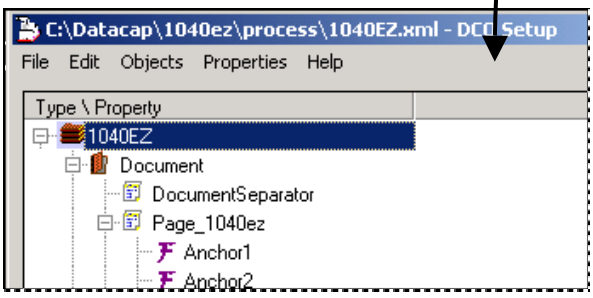


The Task Project contains the task's ***setup*** and ***runtime*** forms – as well as the software that runs the task. The Task Project is also a file (.bpp) – a file that includes the criteria and settings you assign in Phase 4 and Phase 5, as well as the forms.

2. ***Important!*** Your application's Document Hierarchy is a ***required*** property of a Task Project. By the time you start setting up FixUp tasks, the Document Hierarchy file (.xml) will be firmly in place – and immediately available to you (see Chapter 3 and Chapter 6).



1040EZ Taskmaster Administrator – *Workflow tab*



Document Hierarchy Setup Window
1040EZ.xml

To Assemble an iFixUp Task Project

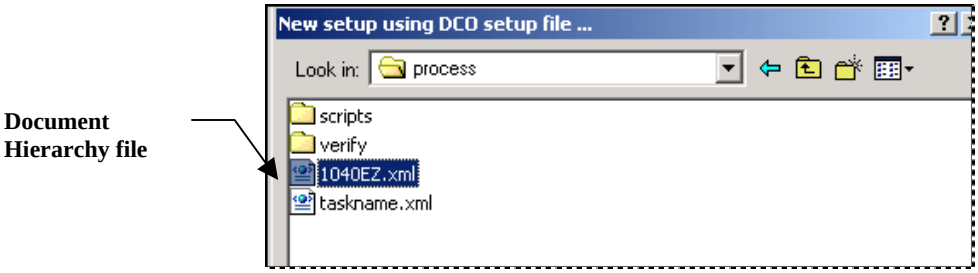
Step	Action

1. Be sure that the **Workflow** component of the Workflow Hierarchy that will contain the task includes a Document Hierarchy file (.xml).

2. Select **Datacap Taskmaster** from your Windows Start button's **Programs** options.

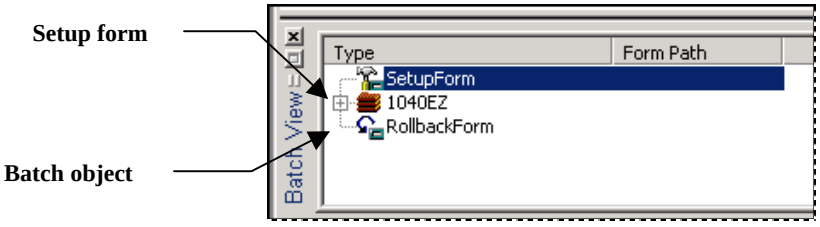
3. To open the ***Batch Pilot Window***, double-click on the **Batch Pilot** icon in the **Batch Pilot** folder. (*Batch Pilot* is the workshop you'll use to put together the Task Project in this phase, and assign task properties in Phase 4 and Phase 5.)

4. Select **New Project** from the window's **File** menu. *Batch Pilot* will instantly ask you to enter the name and path of the Document Hierarchy file in the ***Open File*** dialog. Use the window's **Form** menu to be sure the project is ***not*** in **Design** mode.)



5. 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***. (For thorough explanations of *Batch Pilot* and the ***Batch Pilot Window***, you can click on the Help button at the top of the window, or refer to the *Guide to Batch Pilot*.)

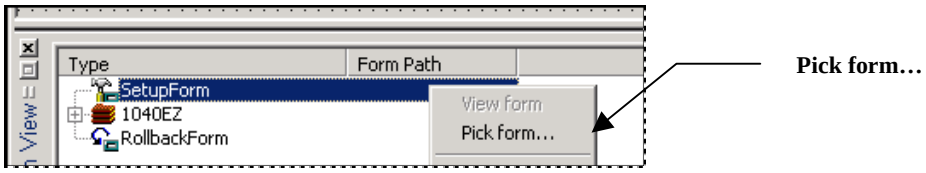
6. Confirm that the **Batch View** area at the bottom of the window displays a **Setup** form item in the **Type** column, as well as the **Batch** object of the Document Hierarchy you've specified (*MQSW* in the example below.)



To Assemble an iFixUp Task Project (continued)

Step	Action

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



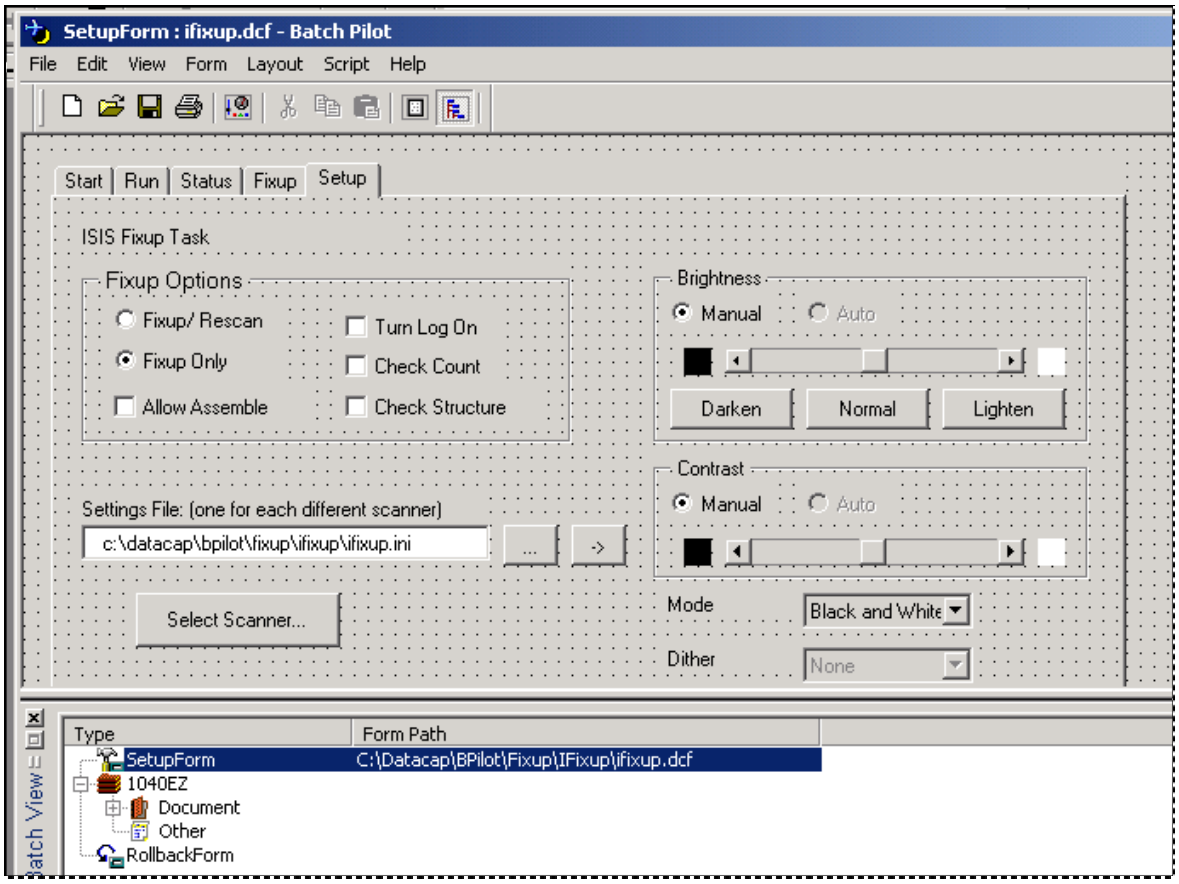
8. Use the ***Open File*** dialog to navigate to the **Datacap** directory's **BPilot** folder.

9. Select **ifixup.dcf** from the **FixUp** folder's **iFixUp** sub-folder, and press the dialog's **Open** button.

10. When you return to the ***Batch Pilot Window***, check the file name and path in the **Form Path** column of the **Batch View** area.

11. Select **Save Project As – *not* Save Project -** from the **File** menu to save this Task Project in your application's **Process** directory - and make it instantly available to the iFixUp Task Definition (Page 55).

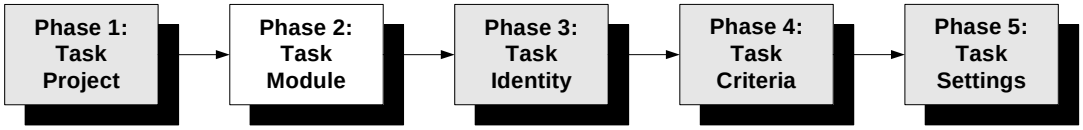
✓ The illustration at the top of the next page highlights elements of the new Task Project.



Batch Pilot Window - *iFixUp Task Project*

iFixUp Setup – Phase 2: Task Module

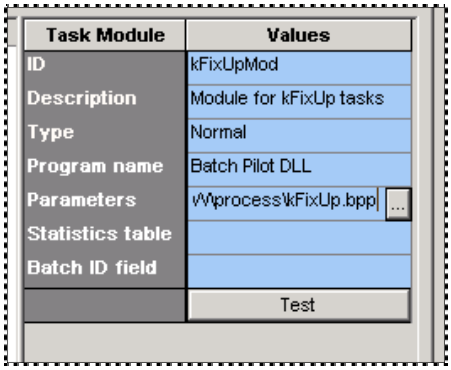
A Task Module connects the iFixUp Task Definition to its Task Project.



When you're setting up a new iFixUp 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 (Page 53).

Step	Action

1. Open the *Modules* tab of your application's ***Taskmaster Administrator***.
2. Click on the Add button to clear the fields in the **Values** area on the right.
3. Enter a unique Module **ID** and a brief but important **Description** of the module.
4. Select *Normal* from the **Type** drop-down list.
5. Select *Batch Pilot DLL* from the **Program Name** drop-down list.

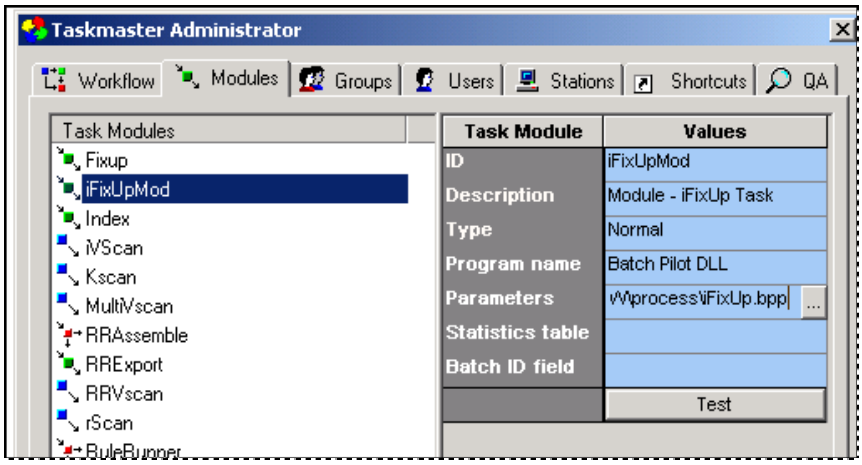


6. Press the Apply button at the bottom of the ***Taskmaster Administrator.*** Click once in the **Parameters** field to display the field's Browse button. Click on the Browse button to retrieve the ***Open File*** dialog and select the Task Project file (.bpp) you assembled in Phase 1 (Page 25).

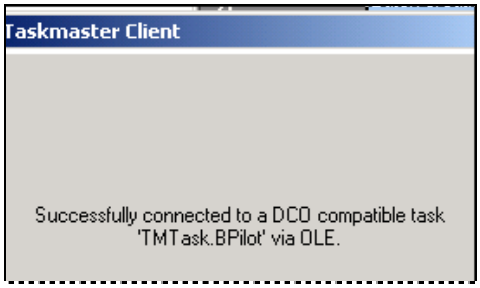
To Define an iFixUp Task Module (continued)

Step	Action

7. Confirm that the new module's ID is now part of the **Task Modules** list on the left-hand side.

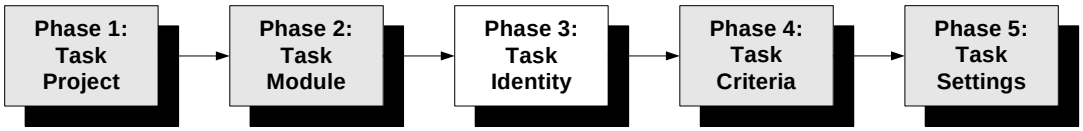


8. Press the Test button. If the connection between the Task Module and Task Project is secure, you will receive this technical message:



iFixUp Setup – Phase 3: Task Identity

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

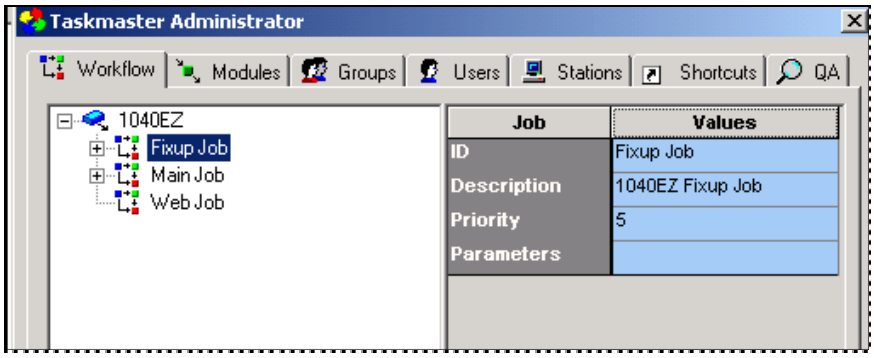


To provide an iFixUp task with its identity:

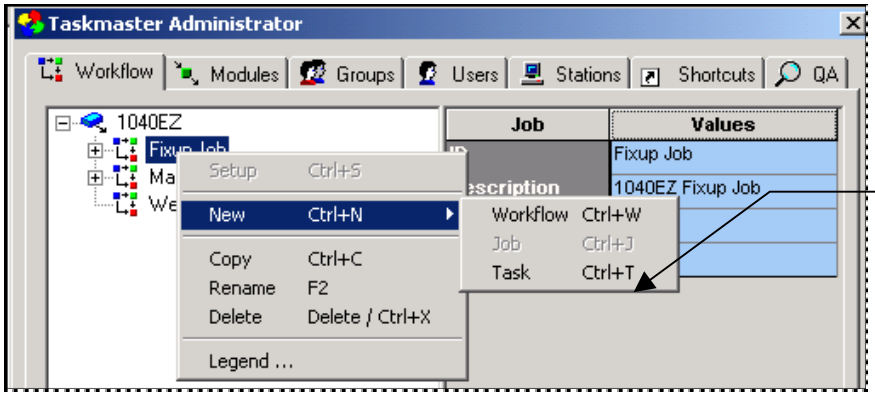
Step	Action

1. Open the ***Taskmaster Administrator's*** *Workflow* tab.

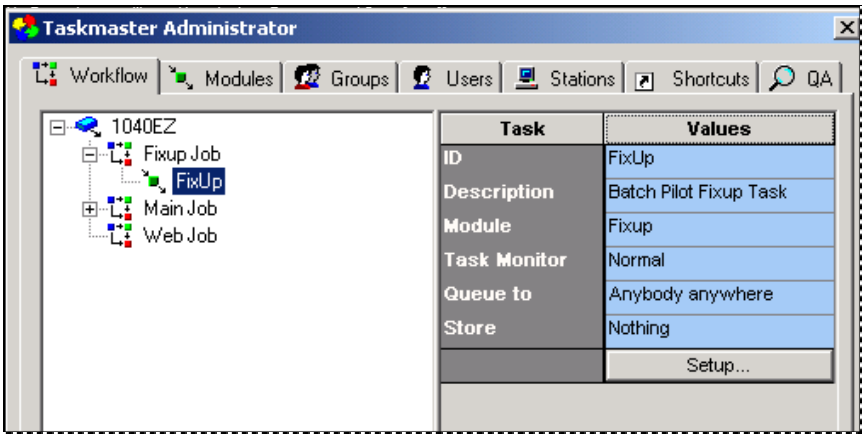
2. Right-click on the ***child*** job that you previously defined (Page 16) –and which will contain the iFixUp task (*FixUp*, in this example.)



3. Select **New** and **Task** from the options.



4. Enter a ***unique*** Task ID in the open space below the Job ID (illustrated on the next page). Be sure this value appears in the **ID** field of the **Values** area as well.



To Provide a Task Identity (continued)

Step	Action

5. In the **Values** area, enter a brief but ***important*** **Description** of this iFixUp task.

6. From the **Module** drop-down list, select the ID of the Task Module you defined in Phase 2 (Page 53).

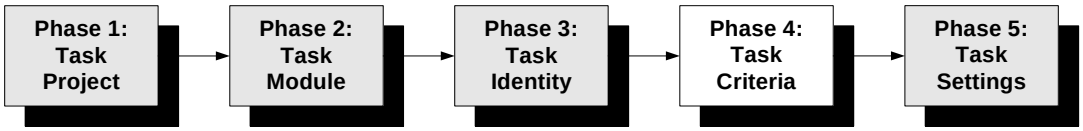
7. Do not alter the ***default*** value of the **Queue to** property. If operations of subsequent tasks within this ***child*** job are to be restricted to the operator or workstation that launches ***this*** task, select one of the values in the **Store** property's drop-down list. (Chapter 6 and *Taskmaster* Help explain the **Queue to** and **Store** properties.)

9. Press the ***Taskmaster Administrator's*** Apply and Done buttons to save the identifying properties of the iFixUp Task Definition.

You can find complete explanations of 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 its **Queue to** and **Store** properties.

iFixUp Setup – Phase 4: Task Specifications

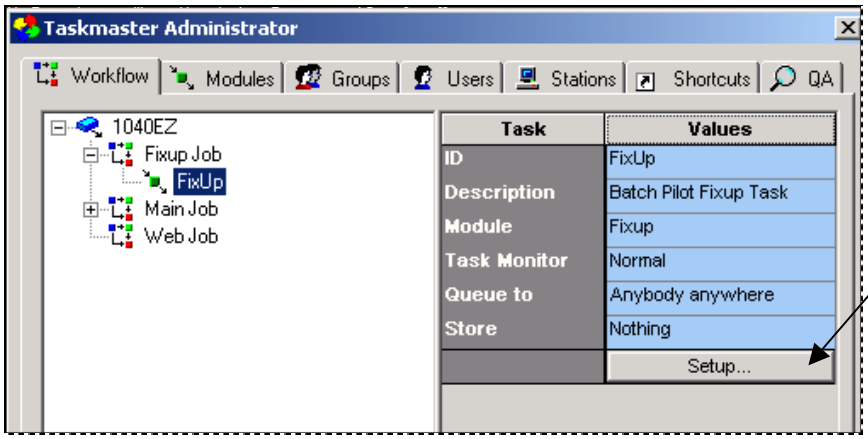
Phase 4 uses the ***iFixUp Setup*** dialog to determine the task's scope – *FixUp Only* or *FixUp and Rescan*.



If the iFixUp task can rescan pages, you'll use this phase to review and possibly modify the scanner settings of the ***parent*** job's iScan task – and to assign new scanner settings.

For either processing mode, you'll evaluate and possibly select specifications that belong only to an iFixUp task.

3. ***Very important!*** During this phase, you'll work with the ***setup*** form of the iFixup Task Project you defined in Stage 1 (Page 49). This form - and the nature of its settings – are linked directly to the **ISIS Source Device** of the scanner used by the ***parent*** job's iScan task scan. When it's time to assign values to the specifications of the ***iFixUp Task Setup*** dialog:
    - Your scanner must be attached to the computer from which you are working ***if*** iFixUp will be operating in its **FixUp/Rescan** mode (see the next page). The scanner must be "On" and the Source Device in place while you set up the task.
    - If iFixUp is to operate in **FixUp Only** mode, you can assign setup specifications ***without*** connecting the scanner.
4. To begin this phase, highlight the Task ID you established in Phase 4 and press the Setup button in the ***Task Master Administrator's*** *Workflow* tab:

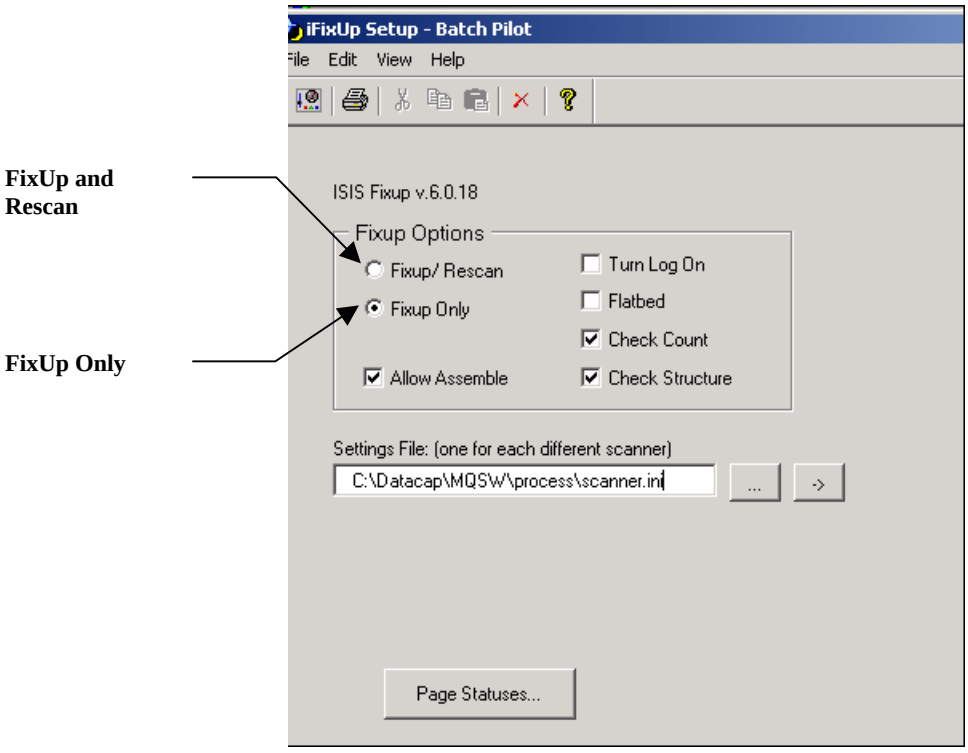


MQSW Taskmaster Administrator – *Workflow tab*

When you first press the Setup button, a warning may appear on your screen. To operate, the iFixUp task has to retrieve and update parameters in the **`[ScanCtrl]`** sector of a

Settings file (.ini). The Settings file – **scanner.ini** - is in the application's **Process** directory.

After you have identified the application's Settings file, indicate whether the iFixUp task can carry out FixUp ***and*** Rescan procedures (*FixUp/Rescan*) – or if it will be restricted to FixUp procedures (*FixUp Only*). These are radio buttons that have a direct impact on the content of the ***iFixUp Operations*** dialog (Page 58) so you ***must*** select an option.



iFixUp - Task Scope

iFixUp Options

The radio buttons in this section (illustrated on the previous page) determine whether the task will be responsible for both FixUp and Rescan procedures, or just FixUp.

- **Fixup** activities adjust and enhance pages; move pages round within the batch; re-constitute documents, and re-organize the batch.
- **Rescan** is a physical process that scans one or more pages and replaces existing Image files with the new files. Rescan procedures are ***closely allied*** to the scanning activities of the Main job's iScan task, and use many of the same components.

Five additional but *optional* settings in this area can expand the task's scope:

- **Allow Assemble** displays the Rebuild button in the ***Operations*** dialog and permits the task to re-assemble the batch and its documents after FixUp and

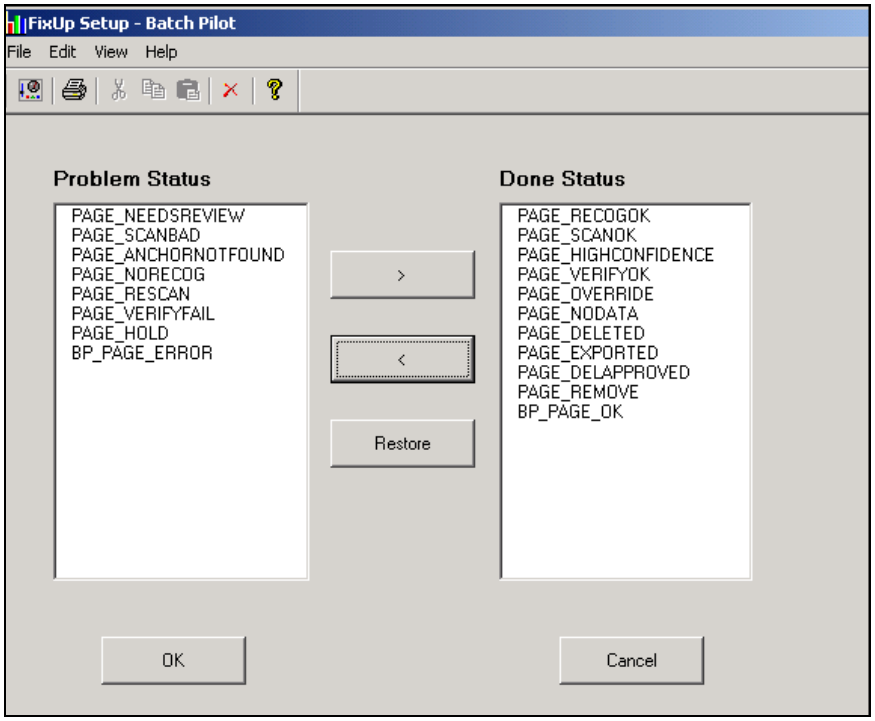
possible Rescan operations are complete. This option can resolve apparent Document Integrity problems without delay.

- **Turn Log On** instructs the task to create a log of its activities, according to specifications you provide in the ***Log*** tab of the ***Task Settings*** dialog (Chapter 6). A log can be a very helpful tool as you evaluate the task's performance and contributions.
- **Flatbed** allows the operator to rescan a single page by placing the paper on the scanner's glass "flatbed".
- **Check Count** counts the *actual* number of documents and pages in the batch; compares the totals to the *expected* counts; and alerts the FixUp operator if there are discrepancies.
- **Check Structure** assesses the structure of each document in terms of the types of pages it contains; the total number of pages; and the number of each Page Type.

Page Status Button

The Page Statuses button opens a dialog with two fields:

- **Problem Statuses** lists the statuses of pages with "problems." A page with a Problem Status ***cannot*** be processed by other tasks until iFixUp has resolved the problem and the Fixup operator has assigned a status in the **Done Status** list to the page.
- **Done Statuses** are Page Statuses that an operator assigns to indicate how a problem has been resolved – and to make the current page available for further processing.



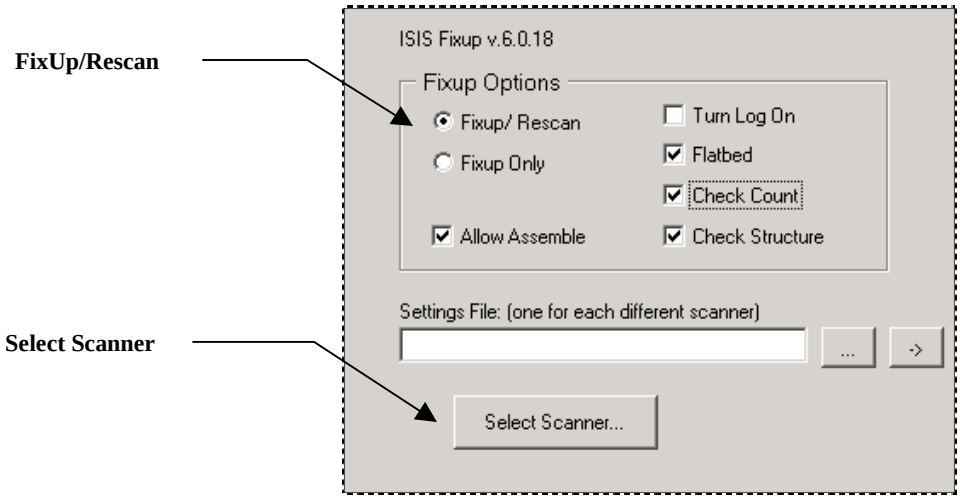
iFixUp Setup – Page Statuses

✓ Directional buttons in the ***Page Status*** dialog move statuses from one side to the other. When you press the OK button, the dialog's current listings update the task's Settings file (Page 34).

Alert! A task assigns a status to every page it processes. Placing a Page Status in the **Problem Status** list means that a page with that status ***cannot*** be released from the iFixUp task until the task has taken care of the problem and its operator has assigned a new, Done status. Be careful when you compile the list: too many "problems" can hamper productivity.

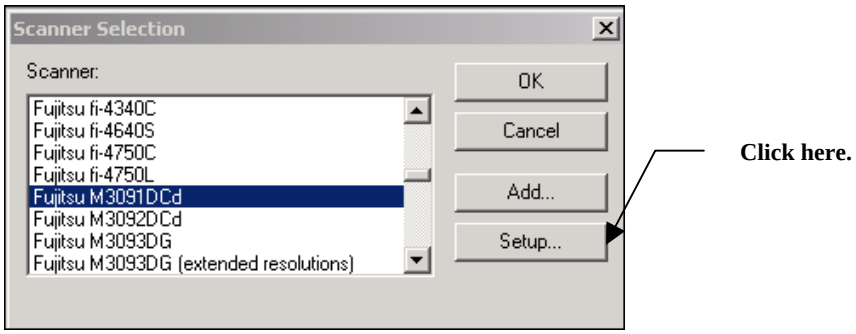
FixUp/Rescan Settings

If you assign the **FixUp/Rescan** mode to the iFixUp task, a ***very important*** button appears towards the bottom of the ***Setup*** dialog's left-hand side, along with a series of scanner settings on the right.



The Select Scanner button links the task to its scanner…and to the Source Device that you have previously defined. ***Remember:*** Be sure that the scanner is on and that the Source Device has been installed – and that the scanner is connected to the computer you're using to assign these Setup criteria.

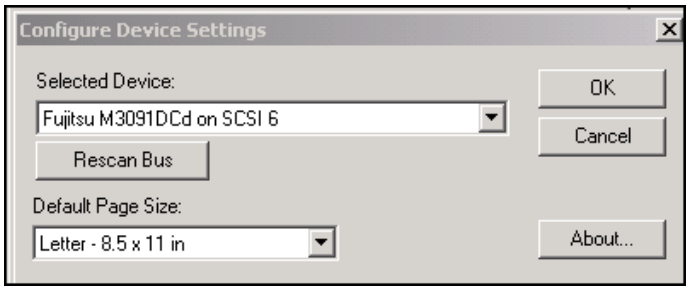
When you press the Select Scanner button, the ***Scanner Selection*** dialog appears. Select your scanner from the drop-down list, and click on the Setup button.



Scanner Selection dialog

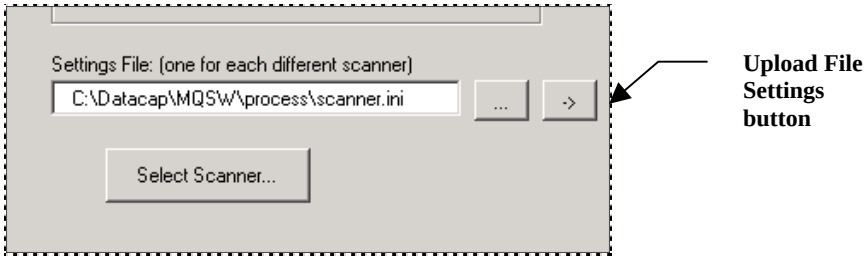
When the ***Configure Device Settings*** dialog appears:

- Select the computer's Source Device from the **Selected Device** drop-down list;
- Choose a **Default Page Size**;
- Click on the OK button.



Configure Device Settings dialog

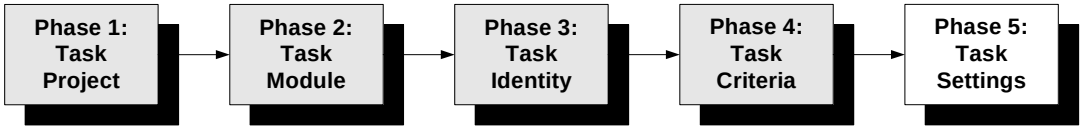
✓ The iFixUp task uses a Settings file (.ini) to retain details of the scanner's specifications. A ***default*** file - **scanner.ini** – is in the **Process** directory of each pre-configured application (*1040EZ,* for example), and in the **NewApp** sub-folder of the Datacap directory's **Support** folder. You can re-name the file and place it in any location as long as you enter its name and path in the ***Setup*** dialog's Settings file field.



Scanner Settings File Designation

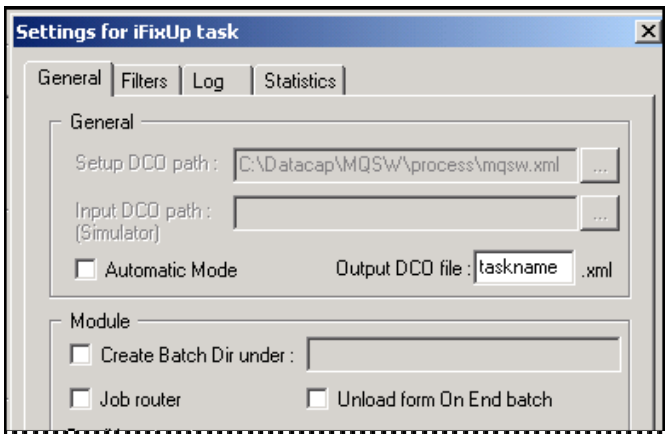
iFixup Setup – Phase 5: Task Settings

The closing phase of an iFixup task's setup involves specifications in the tabs of the ***Task Settings*** dialog.



✓ To access this dialog, select **Task Settings** from the ***Batch Pilot Window's*** **File** menu after you have entered Task Criteria in Phase 4 (Page 57).

The ***Task Settings*** dialog of a typical iFixUp task has nothing exceptional (see Chapter 6 for a full explanation of this dialog.) In the General tab, be sure that you do ***not*** select **Automatic Mode**: iFixUp is a task that requires an operator's participation.



iFixUp Task Settings dialog – *General tab*

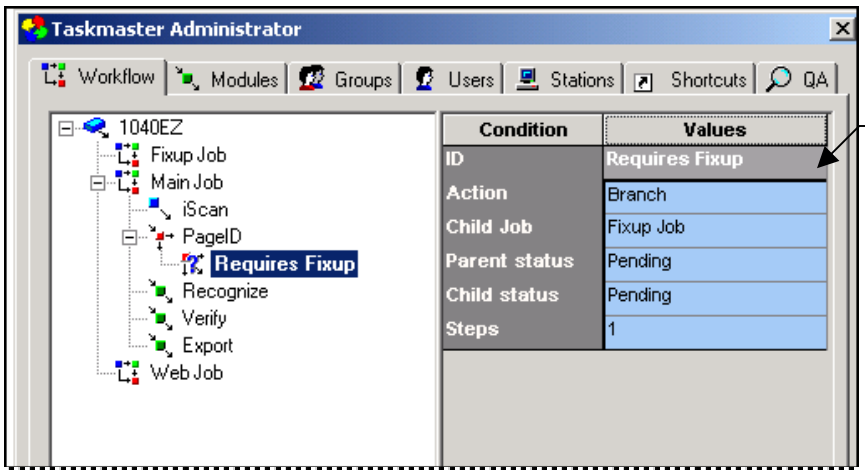
iFixUp - Additional Setup Considerations

The setup of an iFixUp task is incomplete until you have linked the ***child*** job that contains it to a task in the ***parent*** job (below). In addition, even a fully-configured task cannot run until you take the Security steps that will allow individual operators and workstations to launch the task and its job (Page 67).

Task Parameters in the Parent Job

Although the iFixUp task has been fully assembled, the task cannot operate until you link its job – a FixUp ***child*** job (Page 16) - to a task of a ***parent*** job.

In the example below, the *1040EZ* PageID task belongs to the ***parent*** Main Job. The *1040EZ* Administrator has taken steps to assign a **Requires FixUp** processing condition to the task – and to define the condition itself (Page 5).



1040EZ Taskmaster Administrator – *Branching to FixUp*

To set up a link between the PageID **task** and the FixUp **job** when the task encounters the *Requires FixUp* condition involves these steps:

- Be sure that the rules defining the condition have been set up in *Rule Manager*.
- Confirm that the task of the ***parent*** job – PageID, in this example – will process the RuleSet that includes these rule.
- Highlight the task in the ***parent*** job…and the Condition ID. Check that the Condition ID appears in the lists on the tab right side.
- Selection the Action that the task in the ***parent*** job is to take if it encounters the condition.
- Identify the **Child Job** that will be responsible for reviewing and repairing the problem.
- Indicate the status of the batch as it awaits processing by the ***parent*** and ***child*** jobs.

Click on the Apply and Done buttons at the bottom of your ***Task Master Administrator***.

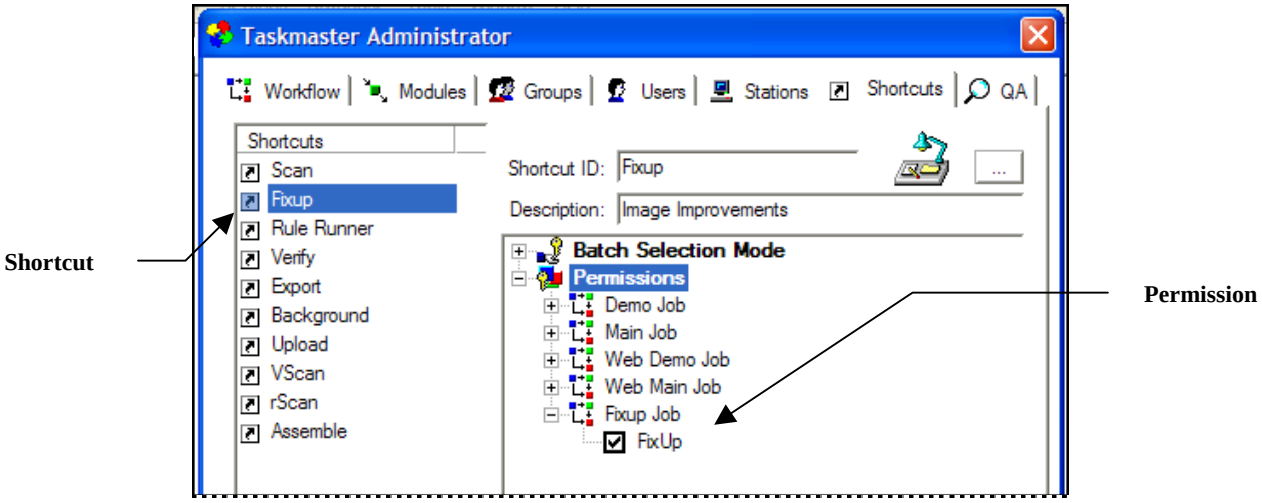
Task Security

The section that begins on Page 67 examines the operation of an iFixUp task. However, the task - technically, the **Job/Task Combination** – cannot run into you assign the following Security parameters:

Authorizations: Shortcut, Station and Operator

FixUp is a specialized endeavor requiring skill and patience – ***and*** Security parameters for at least one FixUp operator, a FixUp workstation, and a FixUp Job-Task Shocrtuct icon.

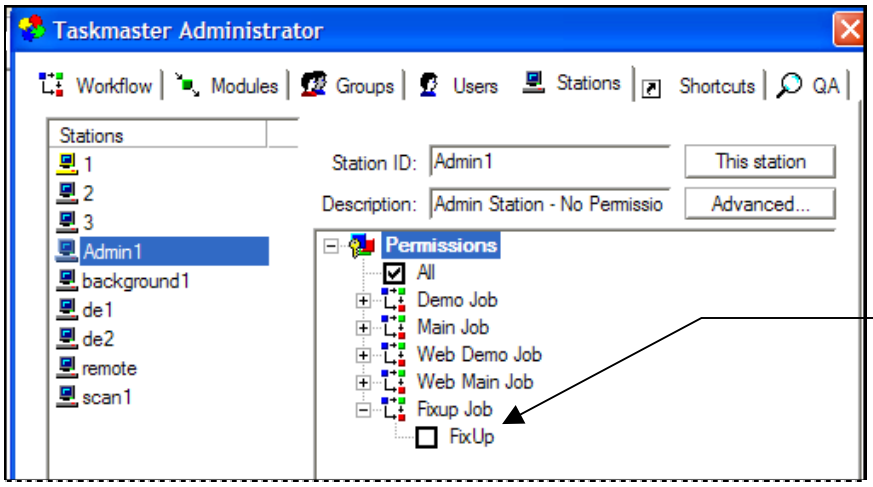
The *Application Wizard* (Chapter 2) automatically includes a *FixUp* Job-Task Shortcut Icon with permission to launch the FixUp Job.FixUp Job/Task Combination:



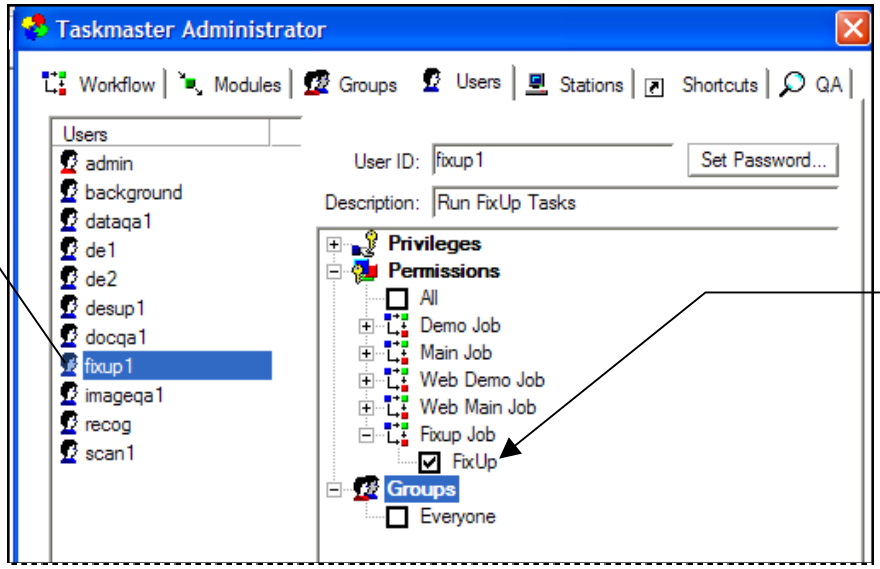
Taskmaster Administrator – Shortcuts tab
FixUp Shortcut Icon

The wizard also supplies the new application with a default *FixUp1* User Definition that gives an operation permission to run the FixUp task (see the illustration on the next page.)

✓ ***However,*** the utility does ***not*** include a Station Definition that authorizes a specific workstation to carry out the FixUp task. ***Very important!*** A Job/Task Combination (such as FixUp Job.FixUp) cannot run unless the operator, station ***and*** shortcut all have permission to run that Job/Task Combination. Chapter 5 of the Taskmaster Administrator's Guide explains all aspects of Application Security.



Taskmaster Administrator – *Stations tab*



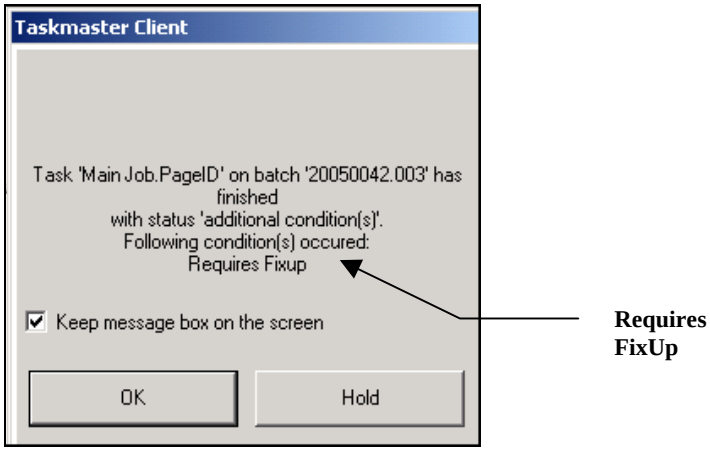
Taskmaster Administrator – *Users tab*
FixUp1 Operator

iFixUp Operations

A task in a ***parent*** job – a Recognition task, for example - will dispatch a batch to a FixUp ***child*** job and its iFixUp task ***whenever*** the Recognition task encounters processing conditions that were

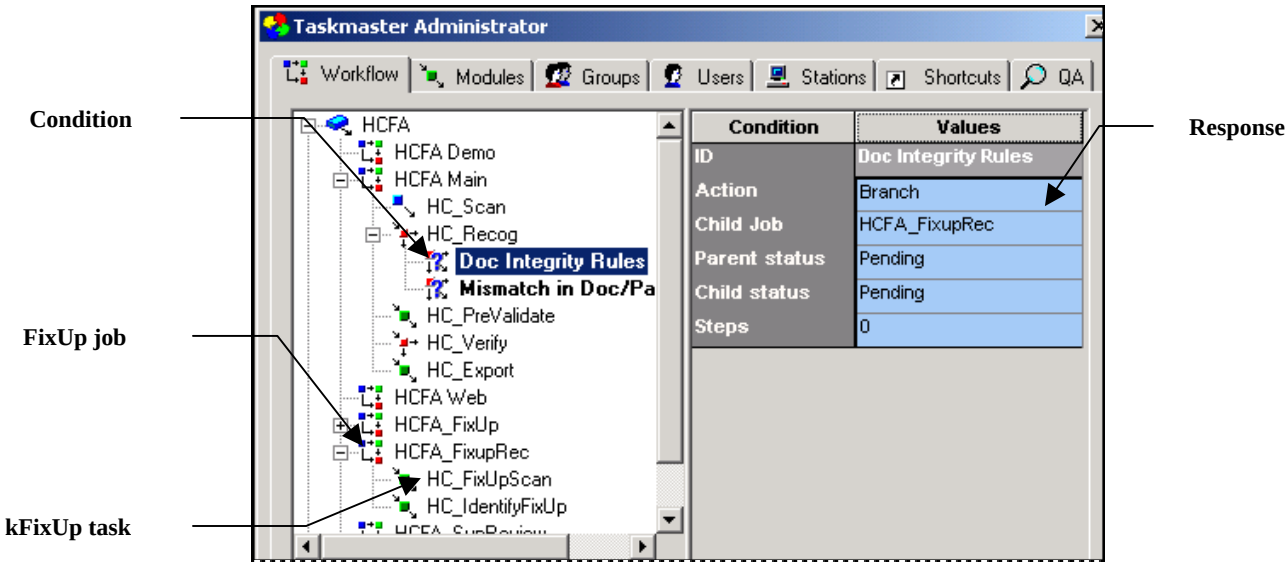
- Defined in the ***Task Settings*** dialog (Page 63)
- Linked to the actions of specific rules (Page 9) or Hot Keys (Page 15).
- Linked directly to the FixUp job by settings in the *Workflow* tab of the ***Taskmaster Administrator*** (Page 64).

The first indication of trouble comes when the ***parent*** job's task attempts to process the problem batch, and a warning similar to this appears on the operator's screen:



In the example which follows, an (intentionally!) small batch processed by the HC_Recog task of the *Taskmaster for Medical Claims* application has detected a **Document Integrity** problem: three out of four documents in the batch do not contain a ***source*** page…in this case, a *HCFA 1500* page. As a result, the task is unable to identify the three documents according to their Document Types – and doesn't know what to do with the *Attachment* pages that have been assigned to the documents.

So the HC_Recog task diverts the batch to the HCFA FixUpRec **job –**and to its opening HC_FixUpScan **task** – an iFixUp task that can repair batch structures ***and,*** if necessary, re-scan pages.



Medical Claims Task Master Administrator – *Workflow tab*

✓ The ***iFixUp*** panel appears as soon as the FixUp operator launches the task by double-clicking on the applicable Job-Task shortcut in the application's ***Operations*** window.

The first illustration on the next page depicts the full scale of the panel; other illustrations highlight specific portions of the panel.

The panel has three sections:

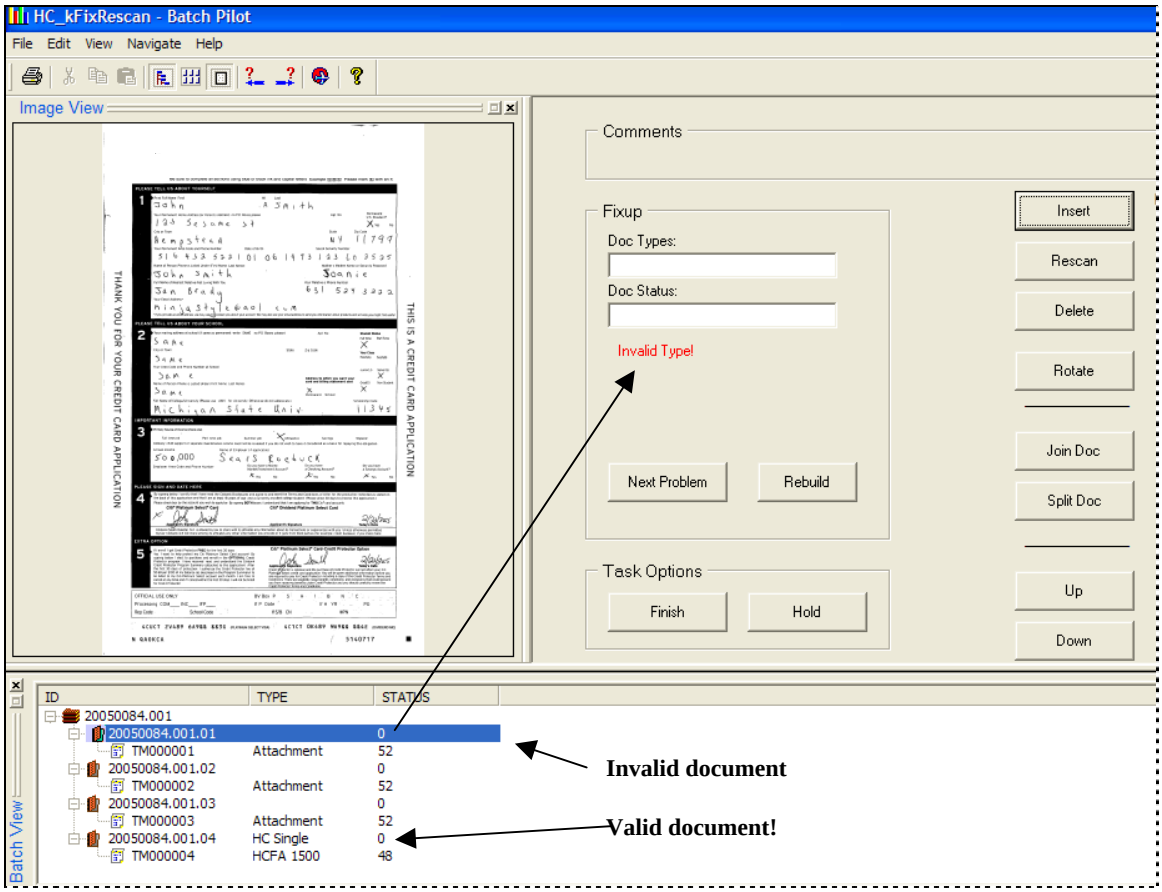
- **Image View** displays the image of the current page – the page that is now under consideration. (A close look at the example reveals that it is ***not*** the image of a *HCFA-1500* health claim!) To open this area, select a page in the **Batch View** area (below) or **Image View** from the window's **View** menu…or click on the **Image View** icon in the window's toolbar.
- **Batch View** lists current ***runtime*** components of the Document Hierarchy at three levels: **Batch**, **Document**, and **Page**. The illustration depicts a batch (*20050084.001*) with four documents (*20050084.001.**01** - 20050084.001.**04**)*. Each document has one page, numbered sequentially (*TM000001- TM000004*). To open this area, select **Setup Tree** from the **View** menu or click on the **Batch View** icon in the window's toolbar.
- **FixUp Actions** contains the fields and buttons you use to repair a batch or dispatch an image to the task's rescan procedures (Page 71).

iFixUp Panel – Batch View Area

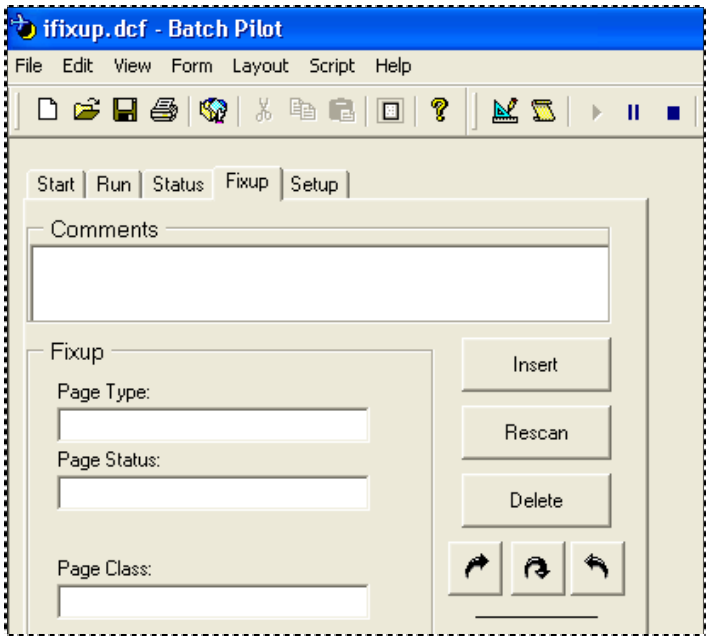
This is the panel's batch assessment area ***and*** its point-and-click staging arena.

When something is wrong with the batch, you can probably identify the nature of the problem(s) right here – once you've had a chance to learn the values of the application's Page Status and Document Status codes (for a list of default statuses, see Page 74)

In the *Medical Claims* example, a first look at the items in the hierarchy shows that only the fourth document has a **Type** value (*HC Single* indicates that the claim document has just one *HCFA-1500* ***source*** page.) What about the other documents? Well, each has an *Attachment* page but does not have a *HCFA 1500* ***source*** page – and therefore does not meet the Document Hierarchy's **Document Integrity** requirements that an *HC Single* document must have one, but only one, *HCFA 1500* page. (Unlimited *Attachments* are permitted as long as the document has that one ***source*** page.)



iFixUp Panel – *Document Repair*



iFixUp Panel - *Page Repair*

The **Batch View** area also uses **Status** values to help diagnose problems. Above, the processing status of each *Attachment* page is "52" (*Doesn'tNeed Veriication*). The Page Status of the *HC Single* document's *HCFA 1500* page is "48"(*Recognition OK*) – an indication of success.

Because the batch has serious flaws, rules governing the HC_Recog task automatically assigned "0" as the Document Status for each document: you'll use the **Doc Status** drop-down list in the **Actions** area to update these values.

iFixUp Panel – Image View Area

Suppose that the first page *Attachment* page is a legitimate *HCFA 1500* page (instead of a Credit Card Application!), but that the page was placed in the scanner's tray upside down.

Although the *Medical Claims* HC Recog task has remarkable image adjustment powers, it may still consider the page as unrecognizable, and assign *Attachment* as its Page Type.

However, when the FixUp operator highlights page *TM00001* in the **Batch View** area, the operator will quickly identify it as a legitimate source page – and can use the Rotate button in the **Actions** area to flip the page vertically.

✓ This area provides the operator with essential before-and-after reviews of the pages in a document, and all pages in a batch.

iFixUp Panel – Repair Tools Area

The fields and tools of the **Repair Tools** area work in concert with the listings and values in the **Batch View** area to repair a batch and its contents. ***Alert!*** The roles and titles of certain features change in response to your selection of a document or page in the **Batch View** area.

The **Repair Tools** area of the ***iFixUp*** panel includes:

Comments

This field displays messages that further explain the nature of an existing problem.

A message appears in this field when the FixUp operator clicks on the Next Problem button (below).

FixUp

These first-level tools give the FixUp operator a chance to intervene without delay to carry out basic remedial procedures

Doc Types/Page Types is a drop-down list of the Document Hierarchy's **Document** objects and **Page** objects (Page 10). Remedying a problem may only mean assigning the correct **Type** value to a page or document that has been highlighted in the panel's **Batch View** area.

Doc Status/Page Status is a drop-down list of the application's Document Status (or Page Statuses). Again, fixing a batch may only require the assignment of a valid processing status – a status that will allow a task in the ***parent*** job to continue working with the page or document, and with the overall batch.

Page Class is a drop-down list of fingerprint classes (Chapter 4). If a Recognition or Verify task in the ***parent*** job has failed to match a page with a fingerprint, the FixUp operator can help out by designating the application fingerprint class for the page.

Problem Alert. A message in this field displays the nature of a problem associated with a particular document or page that the operator has highlighted in the **Batch View** area. (In the example, *Invalid Type* warns the operator that Document 20050084.001.01 lacks a valid **Type** specification.)

Next Problem button. Clicking on this button moves the operator immediately to the next problem document or page. ***Important!*** In a typical batch with hundreds of pages, this button saves the FixUp operator considerable time and frustration.

Rebuild button. Re-builds the batch to accommodate the operator's adjustments. The button can be used at any point in the FixUp process; a series of messages indicate the success or failure the re-building procedures. ***Alert!*** This button is available only if you have selected the **Allow Assemble** option in the ***iFixUp Setup*** dialog (Page 58).

Task Options

Buttons in this area manage the iFixUp task itself.

Finish button. Confirms that all problems have been resolved before assigning a *Finished* status to the batch – or issues an Error Message if problems remain.

Hold button. Places the batch on *Hold.*

Scan Buttons

✓ ***Important!*** These buttons work only with pages and are not available if the task configuration does not include scanning capabilities (Page 35).

Insert scans a new page and ***adds*** it after the highlighted in the **Batch View** area. Clicking on this button opens a dialog that asks the operator for a **Page Data Prefix** value, and a **Counter Start** value.

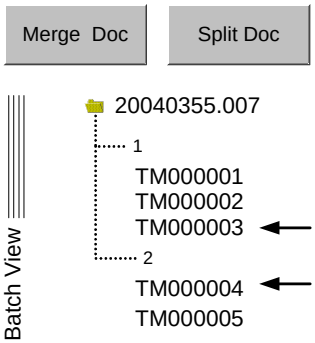
Rescan scans a page and uses it to replace the highlighted page in the **Batch View** area.

Page and Document FixUp Buttons

After a proper warning, the **Delete** button removes a highlighted page. The **Rotate** buttons rotate the image of the highlighted page 90^0 in various directions.

Join and Split Buttons

If the task of the ***parent*** job has organized the batch into a series of documents – each with its own pages – you can **merge** a document with the document ***above*** it in the **Batch/Doc/Page** hierarchy. In the example, if you select the **TM000004** page and click on the Join button, iFixUp will merge the first two documents into one.

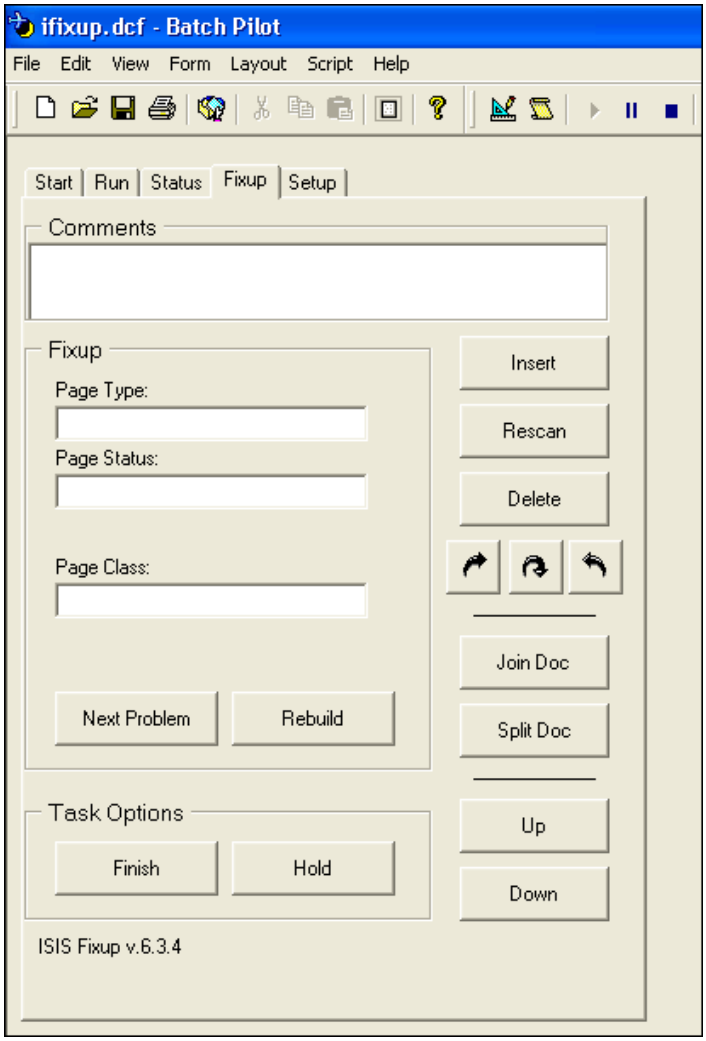


Alternatively, you can split a document into two. In the example, if you highlight *TM0003* and press the Split Doc button, iFixUp will re-organize the batch into three documents. Document 1 will have two pages; Document 2 will have one; and Document 3 will have its original two pages.

✓ Although the Join and Split buttons require practice, they can add significantly to an operator's skills and just as significantly reduce the time he or she needs to repair a batch.

Up and Down Buttons.

These buttons move the highlighted page up or down within the hierarchy.



iFixUp Operations Panel

Page Statuses and Document Statuses

The table below reviews the Page Statuses you're most likely to encounter in a Page file. The list is in alphabetical order.

Page Status	Code	Description
Cannot Find Anchors	51	Indicates that the workflow's Recognition task has failed to identify the anchor fields of the page. A batch containing a page with this status usually ends up in the hands of a FixUp operator (Chapter 9).
Deletion Approved	77	Indicates that an operator or supervisor has approved the deletion of the page ***and*** document.
Deleted Page	75	Marks a page ***and*** its parent document for deletion and removes their association with the sponsoring batch. This status does not, however, directly affect the corresponding Image file.
Export Done	76	Indicates that the ***entire*** workflow has successfully processed a page.
No Data	74	Ensures that none of the data on the page will be forwarded to an Export task – in effect, deleting the page. The assignment of this status to a page, usually by an operator or supervisor, results in a *Pending* status for the document (140).
Page on Hold	72	Suspends the processing of a page, often when a page requires special attention, and places its document on *Hold* as well (130). If any page remains on *Hold* after the task processes the other pages in a batch, *Taskmaster* automatically assigns a *Hold* status to the batch itself.
RecogDoneOK	48	Indicates a page that has been successfully processed by the Recognition task and is ready for Verification. This status does ***not*** mean that data in all fields on the page is complete and accurate.
Recognition Not Done	65	Indicates that although Recognition successfully identified the anchors on the page, a processing error prevented further Recognition steps. *Taskmaster* generates this status automatically; pages with this status should be treated as problems..

Page Status	Code	Description
Remove Page	78	"Removes" the page from the workflow and its data from further consideration. ***It also removes the document containing the page.*** In most cases, a page with this status is beyond repair.
Rescan Page	70	Marks a page for re-scanning (Chapter 9).
Review Page	79	"Branches" a page to a child job for review and possible repair.
Scan Bad	50	Indicates that unsuccessful scanning has resulted in a problem page.
Scan OK	49	Indicates that a Scan task has successfully scanned a page, and is prepared to send the page to Recognition. *Taskmaster* will automatically assign this status to any page that scans or rescans correctly.
Verification Done	66	Indicates that a page has been successfully processed by a FixUp2k task operating in the *Verify* mode (Chapter 9). With rare exception, this status is assigned only by *Taskmaster* when the Verification task completes its work.
Verification Failed	75	Indicates that a FixUp2k task in the *Verify* mode has been unable to process the page, and an operator or supervisor has intervened to assign this status. As a result, the batch is usually placed on *Hold*.

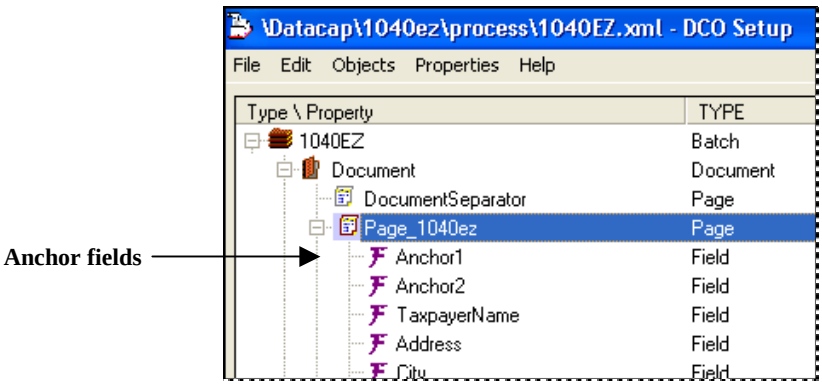
The ***Document*** Statuses you'll most likely encounter include:

Page Status	Code	Description
Complete Doc	141	Indicates that the Recognition task has been able to assemble a document with the correct number and type of pages.
Deleted Doc	128	Indicates that an authorized operator or Supervisor has removed a document from the internal organization of the batch.
Incomplete Doc	147	Denotes a document with the wrong number or type of pages.
Needs Review Doc	66	Sets aside the document and its pages for review by a remedial FixUp task.
On Hold Doc	130	Places the document and its pages on hold until an authorized operator or Supervisor releases it.

Page Status	Code	Description
Override Doc	75	Indicates that an authorized operator or Supervisor has "overridden" the steps a Verification or Validation task has taken to halt processing on the document and its pages.
Pending Doc	140	Indicates that the document and its pages are ready for additional processing.

Anchor Fields – FixUp

The Document Hierarchy of the *1040EZ* application contains two **Field** objects - *Anchor1* and *Anchor2* – that represent "Anchor" fields on a ***source*** page:

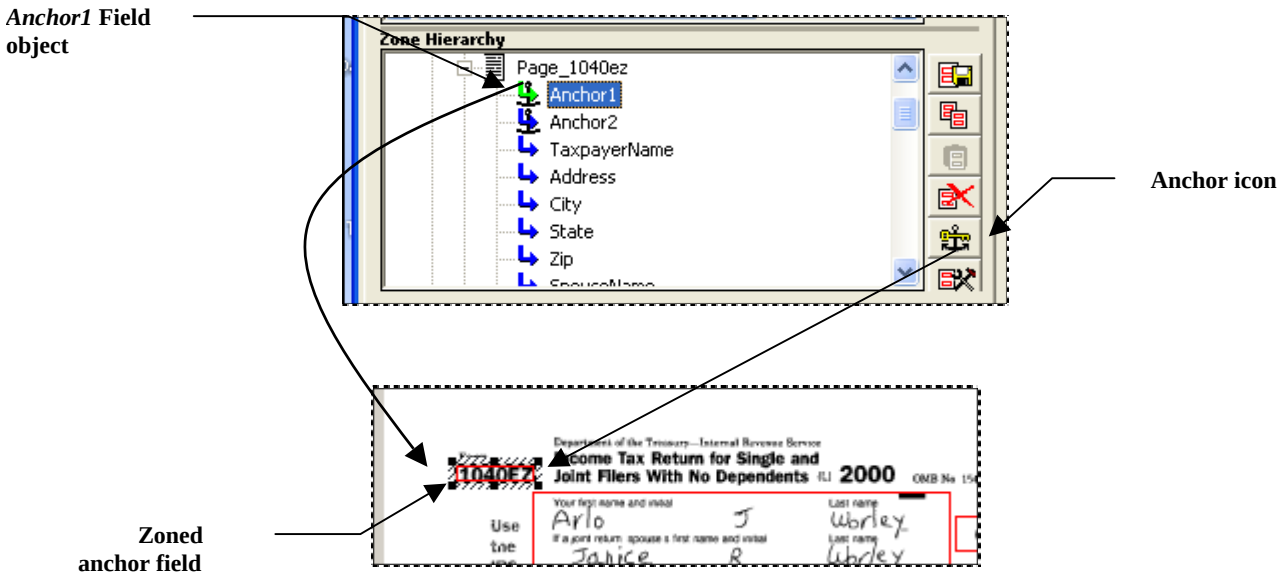


1040EZ Document Hierarchy

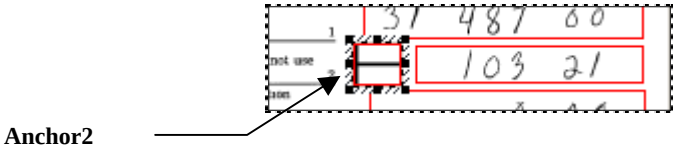
The **Anchor** icon in the *1040EZ* ***Rule Manager Window*** has been used to locate and zone these very important fields on the image of the *1040EZ* fingerprint, and to assign an anchoring

role to each field.

Below, the pre-printed **1040EZ** caption on the 1040EZ IRS form serves as one anchor: it is an important point of reference for the application's PageID task as the task attempts to identify the pages in a batch according to their Page Types.



A second anchor might be helpful because it can confirm the PageID task's identification of a ***source*** page. The default setup of the *1040EZ* application singles out the symbol to the left of the **Taxable interest** amount to serve as the second anchor:



✓ Often, however, this symbol is not as dark as it needs to be. The following paragraphs show you how to select a different item in the fingerprint's image to be the second anchor, and how to establish FixUp procedures to deal with **missing anchors.**

How to Configure Anchor Fields

The chart at the top of the next page identifies the components of an application that require your attention if you intend to:

- Add an Anchor **Field** object to the application's Document Hierarchy;
- Zone the Anchor field on a fingerprint;
- Define rules to guide a *RuleRunner* task's attempts to locate the anchor field on a ***source*** page;
- Define a processing condition that leads the task to turn the current batch over to a FixUp job when it cannot locate an anchor on the ***source*** page.

Step 1: Add an Anchor Field Object to the Document Hierarchy

To add an Anchor field to your application, follow the procedures outlined in Chapter 3 to add a **Field** object to the application's Document Hierarchy: this **Field** object will represent the fingerprint's Anchor field. ***Be sure*** that the field is a child of the ***source*** **Page** object. (In the illustration on the previous page, *Anchor1* and *Anchor2* are children of the *1040EZ_Page* **Page** object.)

Step 2: Assign a "Required" Status to the Anchor Field Object

The **Field** object that represents an Anchor field must have a **Required** property with "1" as its value.

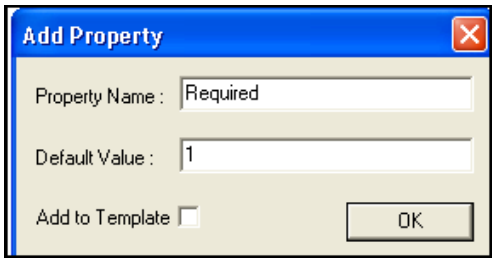
Alert! Because **Required** is not a default property of a Document Hierarchy's **Field** objects, take these steps to include the property (see Chapter 3 for details):

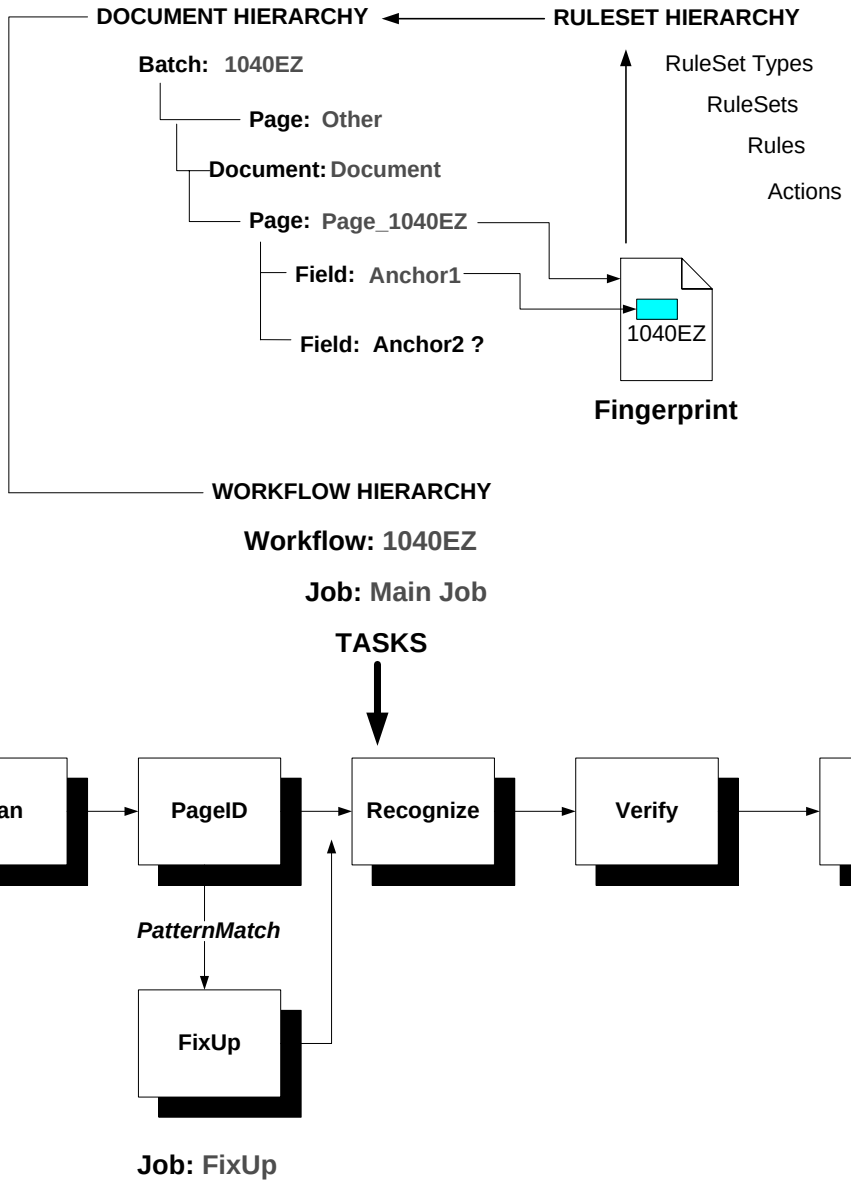
a. Open the ***Document Hierarchy Setup*** window.

b. Highlight the Anchor field's name.

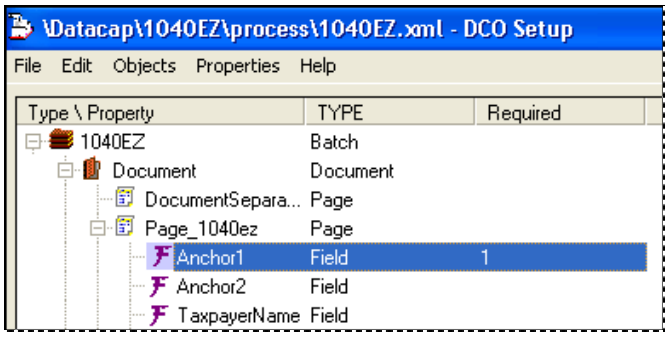
c. Select **New** from the **Properties** menu.

d. Enter "Required" as the **Property Name** (without the quotation marks!):



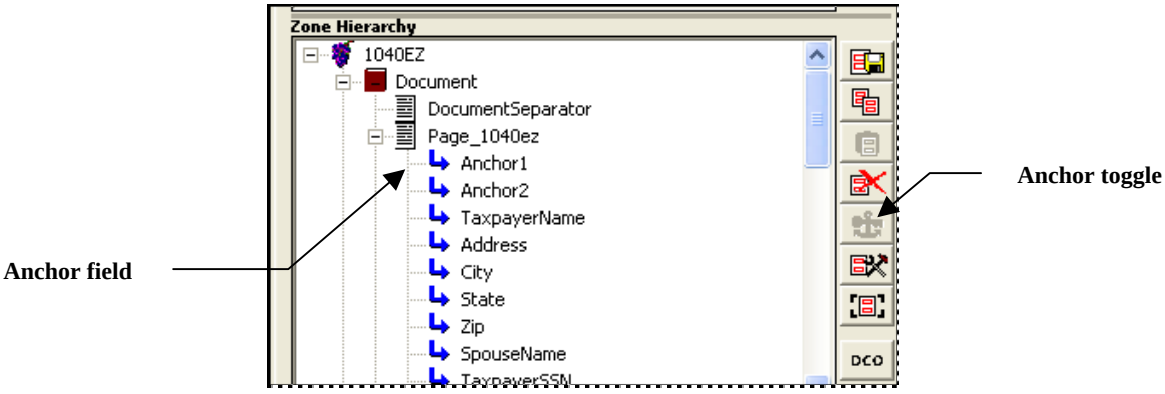


✓ Check to be sure that the **Field** object now includes a **Required** property with *1* as its value, and repeat these steps for any other Anchor fields:



Step 3: Zone the Anchor Field

As soon as you have added the **Field** object to the Document Hierarchy, open the *Fingerprints & Zones* panel of your application's ***Rule Manager Window***…and the image of the fingerprint that will be zoned with one or more Anchor fields.



1040EZ Rule Manager Window -
Fingerprints & Zones panel

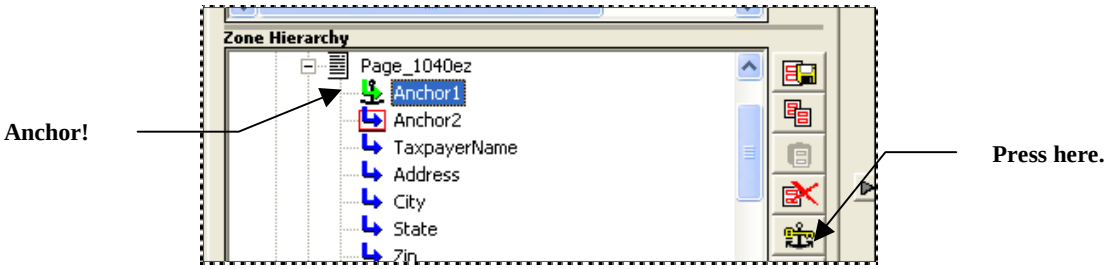
Pay close attention to two items in the example above:

- The *Anchor1* and *Anchor2* fields in the **Zone Hierarchy** area are still identified by standard ↳field icons.
- The **Anchor** toggle icon on the right edge of the **Zone Hierarchy** area is not yet available.

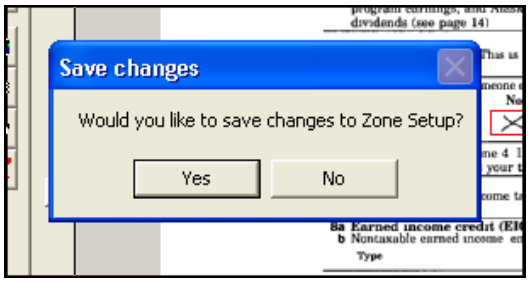
To assign anchor qualities to a specific field:

Step	Action

1. Highlight the Anchor field's name in the **Zone Hierarchy** area.
2. Check that the **Anchor** toggle icon on the right-hand edge is active and available.
3. Depress the **Anchor** toggle icon
4. Confirm that the Anchor field's identifying icon on the left changes to an "anchor"



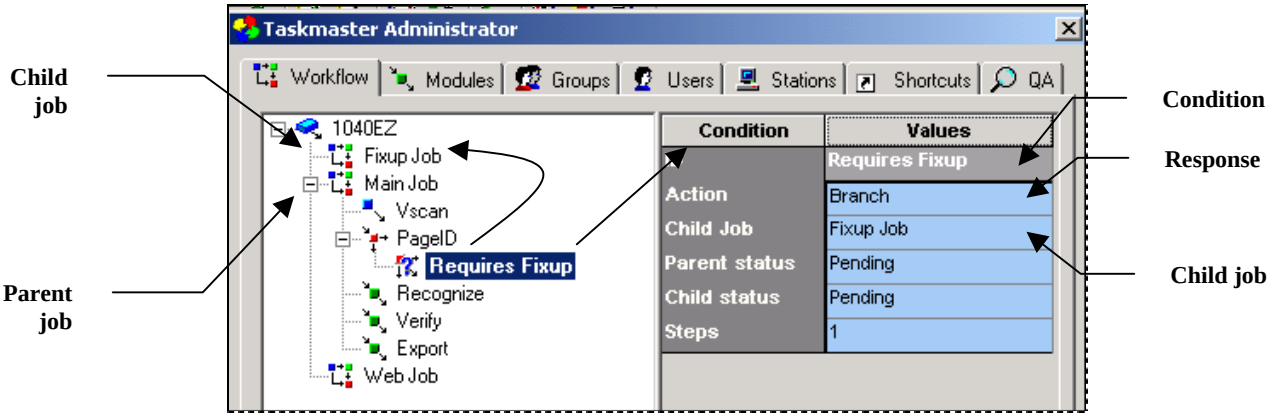
5. While the **Anchor** toggle is still depressed, zone the Anchor field on the fingerprint's image. ***Important!*** You can use the **Anchor** toggle icon to assign and remove a field's Anchor qualities.
6. Un-toggle the **Anchor** icon and click on the Yes button of the ***Save Changes?*** dialog.



7. Press the OK button of the ***Zone Setup*** dialog.

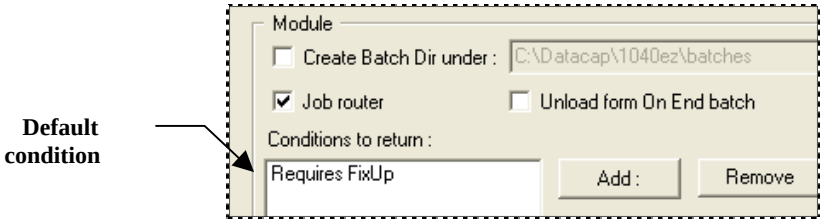
How to Define Anchor Field Conditions and Rules

The default setup of the Main Job of the*1040EZ* application branches to its FixUp ***child*** job if the PageID task encounters a pre-defined *Requires FixUp* condition:

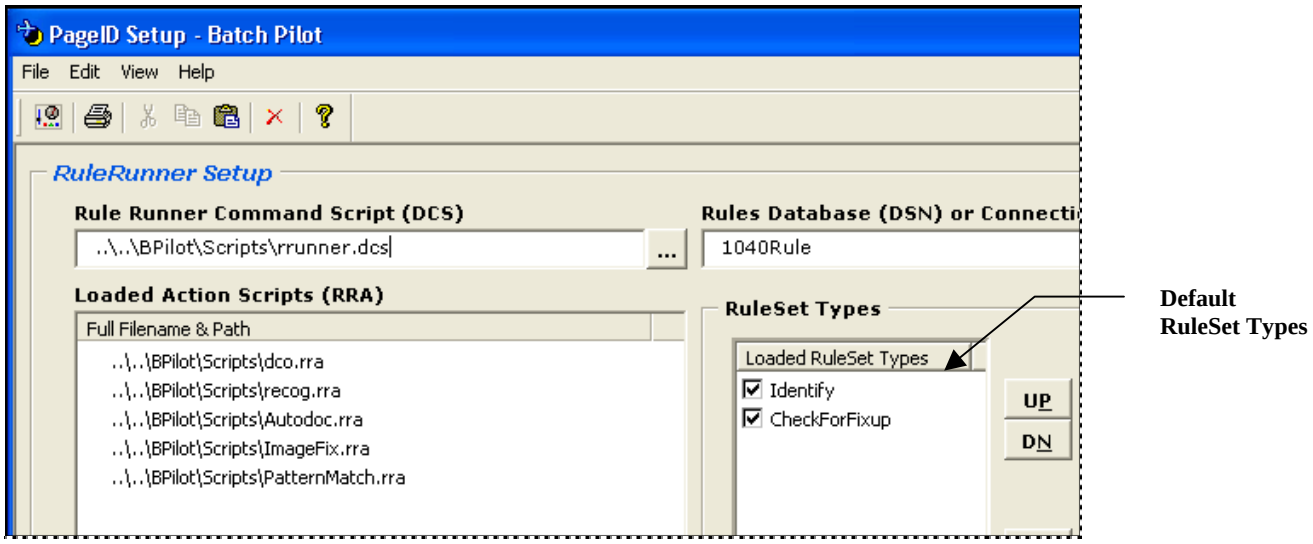


Condition Properties – *Requires Fixup*

This condition is set forth in the Task Definition's ***Task Settings*** dialog; rules of the two RuleSet Types that govern the PageID task's response are defined by the *1040EZ* application's *Rule Manager*, and assigned to the task during Task Definition:

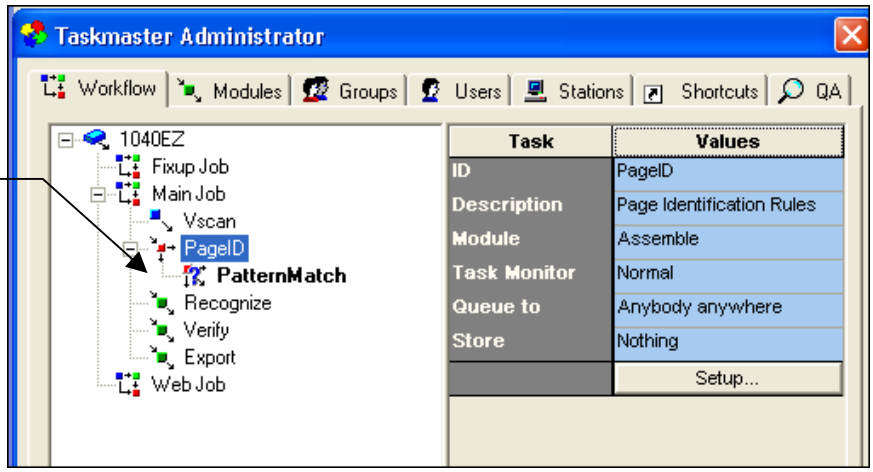


PageID Task Settings dialog – *General tab*



PageID Task Setup dialog – *PageID Task*

✓ You can modify the *1040EZ* application without difficulty so that its PageID task will look for the Anchor fields you designated previously (Page 80) – and divert the batch to the FixUp job if the search is unsuccessful. To begin, you'll need a new processing condition for the PageID task:

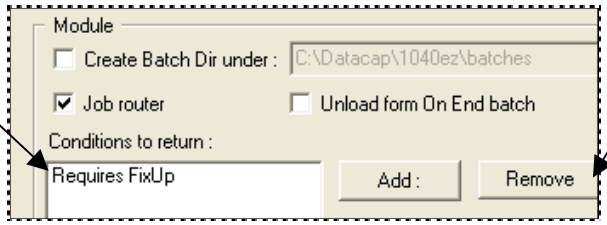


New condition

To replace the existing condition with a new condition ("*PatternMatch*", perhaps):

Step	Action

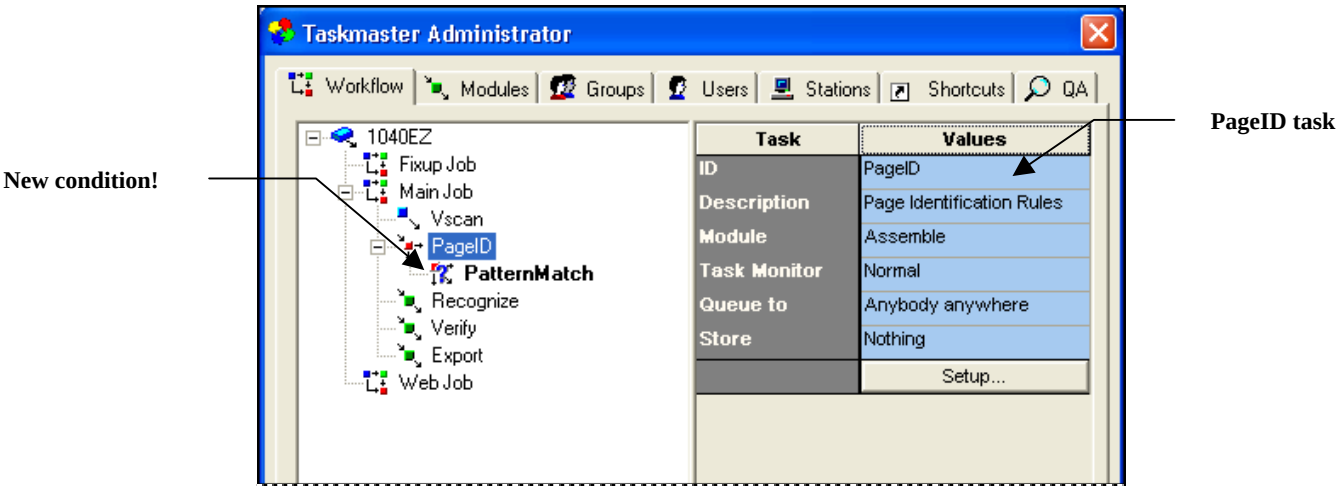
1. Open the *Workflow* tab of the application's ***Taskmaster Administrator***.
2. Highlight the Task ID of the branching task in the parent job.
3. Press the Setup button. When the ***Task Setup*** dialog appears, select **Task Settings** from the **File** menu: the *General* tab of the ***Task Settings*** dialog will appear on your screen.
4. Highlight the condition you intend to replace in the **Condition to Return** field and press the Remove button.



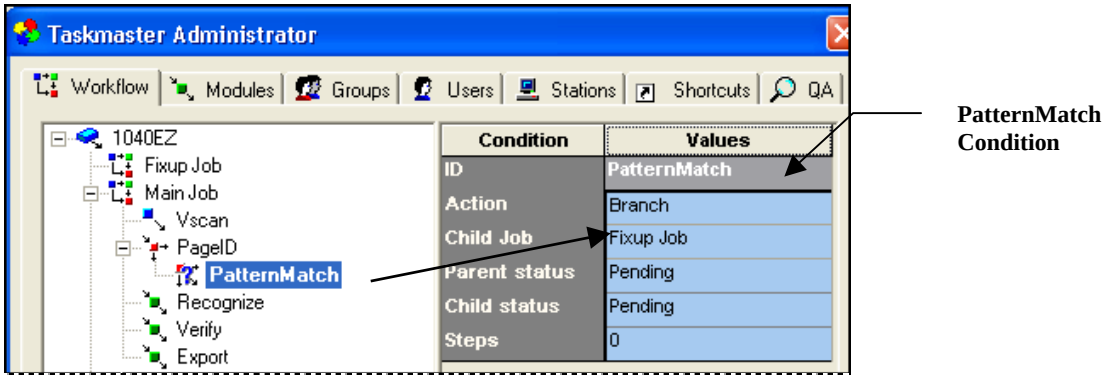
Highlight this condition.

Click here.

5. Click on the OK button to close the ***Task Settings*** dialog. Press the Done button of the ***PageID Task Setup*** dialog. When the ***Taskmaster Administrator's*** *Workflow* tab returns, press the Apply button, and the Done button.
6. Compress the job listings in the *Workflow* tab.
7. Open the job listings, and highlight the Main Job's PageID task (illustrated on the next page.)



1040EZ Taskmaster Administrator – *Workflow tab*
PageID Task Properties

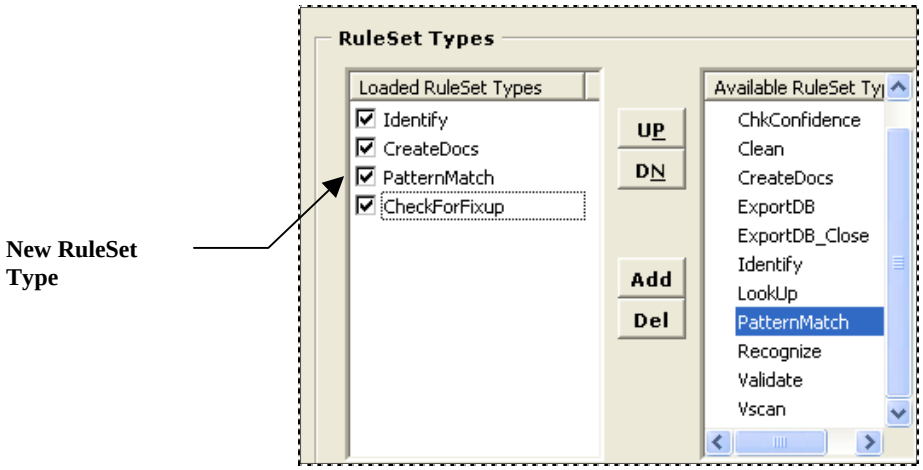


1040EZ Taskmaster Administrator – *Workflow tab*
PatternMatch Condition Properties

✓ Instead of the two RuleSet Types assigned to the PageID task for its basic batch review procedures (illustrated on Page 82), the extended search for Anchor fields also involves the actions in rules of two additional RuleSet Types: **CreateDocs** and **PatternMatch.**

As a result, the Task Definition of the PageID task will include four RuleSet Types listed in the processing order shown on the next page: **Identify**,**CreateDocs**, **PatternMatch**, and **CheckforFixUp.**

- The **Identify**, **CreateDocs** and **CheckForFixUp** RuleSet Types are already available. However, you have to move the **CreateDocs** RuleSet Type from the **Available RuleSet Types** list to the **Loaded RuleSet Types** list – and a step up to a position below **Identify**.
- The **PatternMatch** RuleSet Type does not yet exist. Just follow the instructions in Chapter 4 to set up the new RuleSet Type, or take the steps outlined on the next page.



PageID Task Setup dialog

How to Define an Anchor Field's RuleSet Type

To construct the new **PatternMatch** RuleSet Type (you can give it any name!):

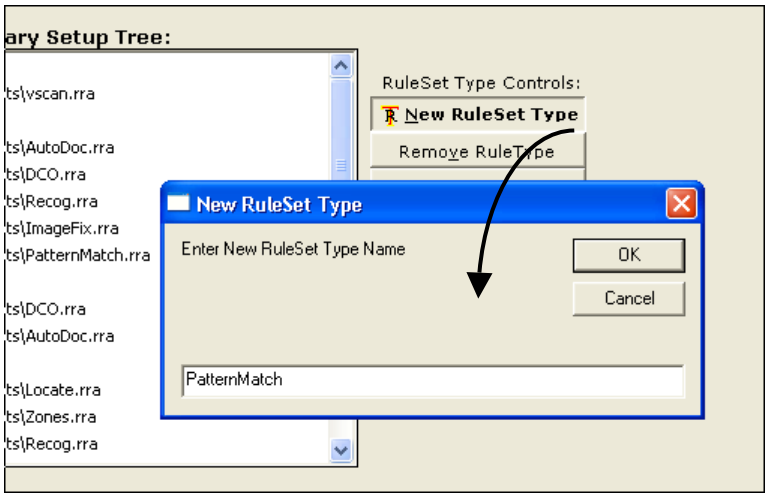
Step	Action

1. Open the *Rules* panel of your application's ***Rule Manager Window.***

2. Press the **RuleSet Types and Actions Libraries** icon along the right edge of the **Actions Library** area.

3. Click on the New RuleSet Type button in the ***RuleSet Types*** dialog.

4. Use the ***New RuleSet Type*** dialog to enter the new **RuleSet Type Name**; click on the OK button (illustrated on the next page.)

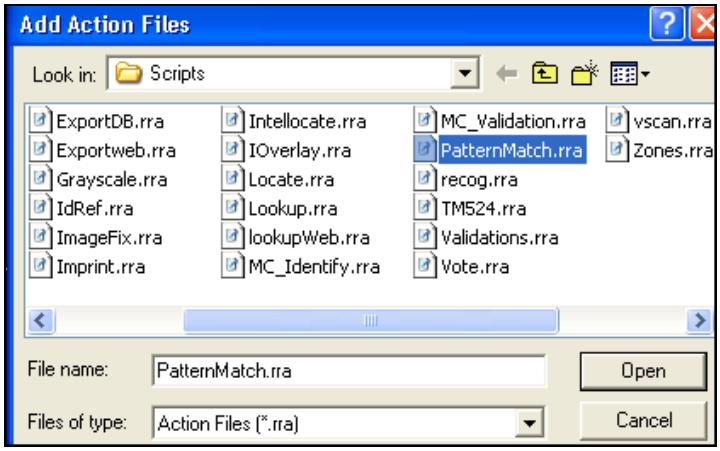


To Construct the New PatternMatch RuleSet Type (continued)

Step	Action

5. Highlight the RuleSet Type's name in the ***RuleSet Types*** dialog and click on the Add RRA button. This will allow you to assign the **PatternMatch** Actions file (.rra) to the new RuleSet Type.

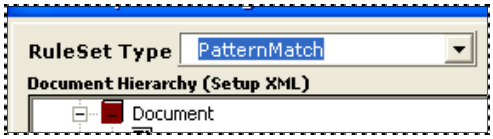
6. Use the ***Add Actions Files*** dialog to add the **PatternMatch** Actions file (.rra) to the RuleSet Type.



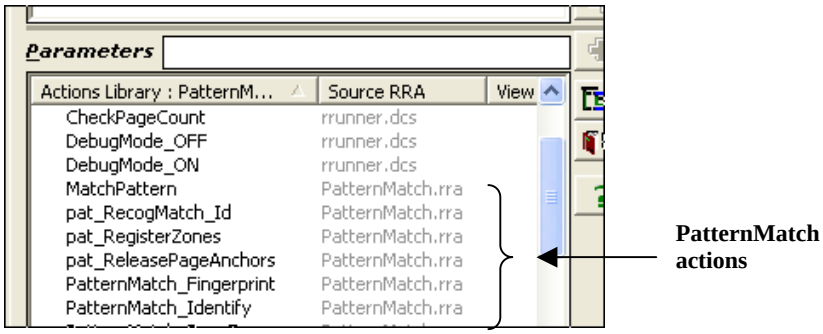
To Construct the New PatternMatch RuleSet Type (continued)

Step	Action

7. Close the ***RuleSet Types*** dialog. Confirm the listing of the new RuleSet Type in the *Rules* panel of the ***Rule Manager Window.***



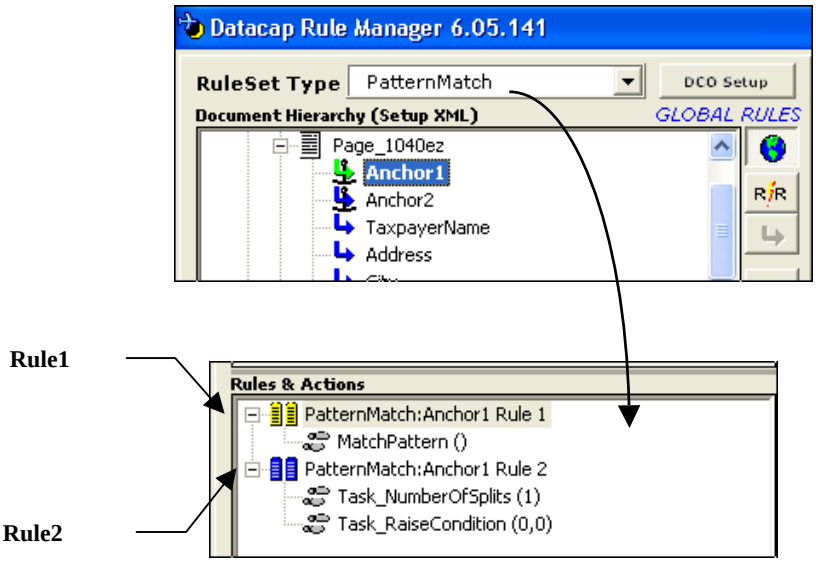
8. Check that the contents of the **PatternMatch.rra** Actions file are listed in the **Actions Library** area when you select the new RuleSet Type's name:



PatternMatch RuleSet

To search for an Anchor field and divert the current batch to the FixUp ***child*** job if the search is unsuccessful, the PageID task has to apply a very simple but powerful **PatternMatch** RuleSet.

The RuleSet is bound to the applicable Anchor **Field** object of the Document Hierarchy – *Anchor1*, for example – and consist of two rules:



The **MatchPattern** action of the first rule directs the task to search the current ***source*** page for the Anchor field represented by the *Anchor1* **Field** object of the Document Hierarchy.

If the search is unsuccessful, Rule2 confirms that the task has encountered a problem; indicates the problem is caused by condition "0" (*PatternMatch*, in this case); and delegates the batch to the ***child*** job identified with a "0" – the FixUp job.