
Task Projects

A *Batch Pilot Task Project* is the foundation for a *Taskmaster* task.

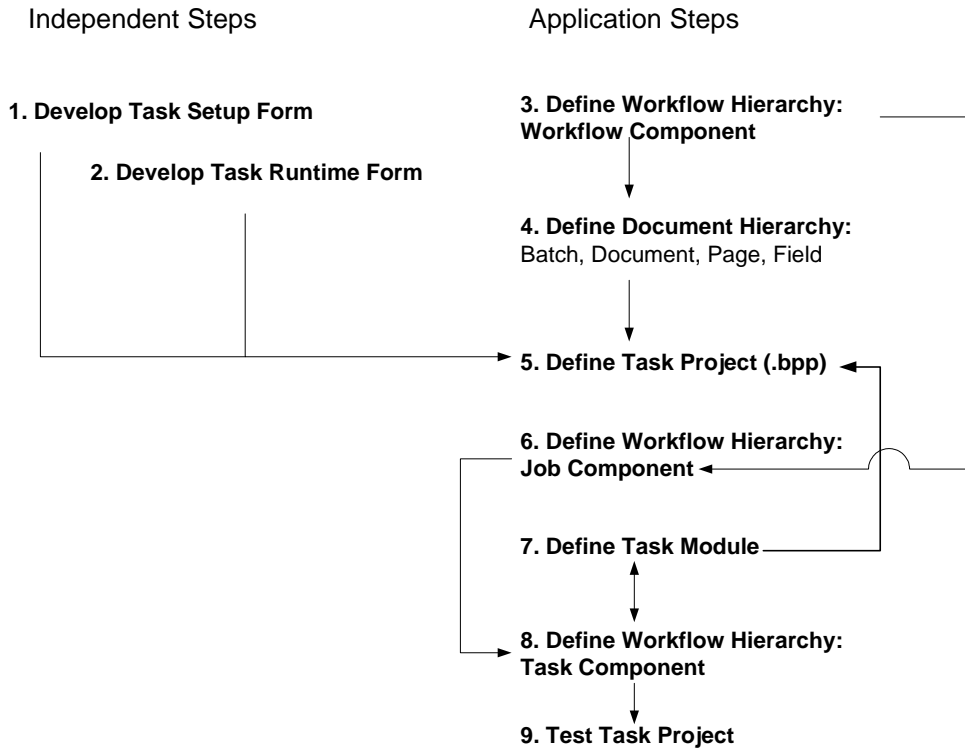
Chapter 4 examines the nature of a Task Project; the assignment of *setup* and *runtime* forms to the project; the content of a Task Project file (.bpp); and the steps you take to test a project. Chapter 4 also introduces you to the *Task Settings* dialog.

Chapter 4 covers these topics:

I. Introduction	4-2
A. Task Projects of the 1040EZ Application	4-3
II. Settings of a Task Project	4-4
i. The Task Project File	4-8
III. How to Define a Task Project.....	4-12
A. Auditing the New Project	4-17
B. Task Projects and Task Definitions	4-18
i. Task Settings Dialog.....	4-18
IV. How to Test a Task Project	4-26
A. Testing Setup Procedures	4-26
B. Testing Runtime Procedures.....	4-27
i. Logs of Task Project Runtime Tests.....	4-28

Introduction

Inside a **Datacap Taskmaster** application, the process of Task Definition takes this path:



The initial *independent* steps construct the *setup* and *runtime forms* that become the dialogs that the task uses to assign specifications to the Task Definition, and to interact with a user when the task runs. (Chapter 3 describes the nature and development of a project's forms.)

On the right, Step 3 establishes a formal **workflow** as a component of the *Taskmaster* Workflow Hierarchy, and assigns values to the workflow.

Step 4 sets up the application's **Document Hierarchy**, and places objects at four levels. A typical Document Hierarchy has a single **Batch** object at the top; one or more **Document** objects that belong to the batch on the second level; one or more **Page** objects for each document on the next level; and **Field** objects within most pages at the bottom.

Step 5 uses *Batch Pilot* to assemble the **Task Project**. The Document Hierarchy is a required component of the Task Project, along with a *setup* and *runtime* form.

Step 6 defines a *Taskmaster job* as a child of the workflow you put together in Step 3.

Step 7 sets up a **Task Module** that links the Task Definition to the Task Project.

Step 8 defines the task itself. Much of this step involves the use of a dialog derived from the Task Project's *setup* form, and the assignment of specifications in tabs of *Batch Pilot's Task Settings* dialog.

Step 9 tests the Task Project, using *Batch Pilot's Task Simulator*.

Although this chapter refers to and explains the relationships between the steps, the chapter concentrates on Step 5, and on the content and use of the *Task Settings* dialog.

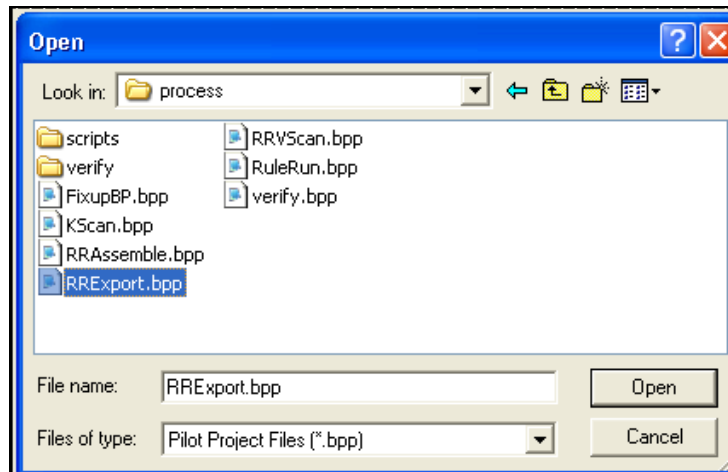
- ✓ **Very important!** The Task Project for an application's Verification task usually has an additional responsibility – the construction of the task's *Data Entry* panel. Chapter 5 examines this feature in depth.

Task Projects of the 1040EZ Application

1040EZ is an almost indestructible training application that is immediately available to you as you learn about existing Task Projects, and learn how to set up and test new projects.

To access an existing *1040EZ* Task Project, take these steps:

Step	Action
1.	Open <i>Batch Pilot</i> .
2.	Select Open Project from the <i>Batch Pilot Window's</i> File menu. (Don't select Open Form instead.)
3.	Use the tools of the Open File dialog to access the <i>1040EZ</i> application's Process directory.



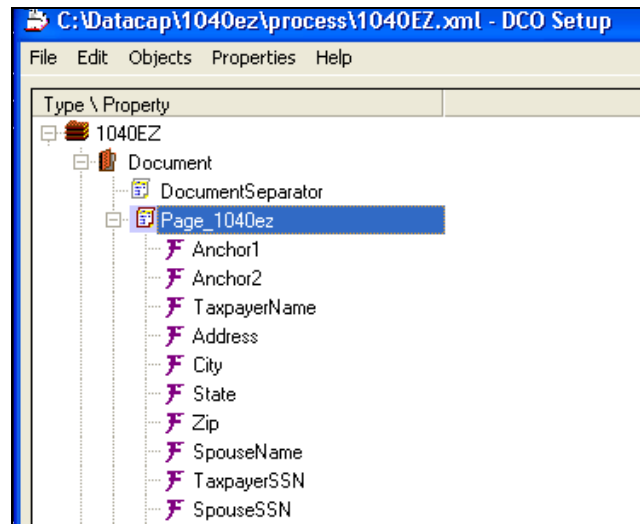
4. Open a Task Project such as **RRExport.bpp**: this is the project responsible for the application's Export tasks.
5. Toggle the window's **Design** icon to get a good look at the project's *setup* form.

Settings of a Task Project

A *Batch Pilot* Task Project is a file (<task>.bpp) with settings that provide a conduit between a task's setup procedures and its runtime operations.

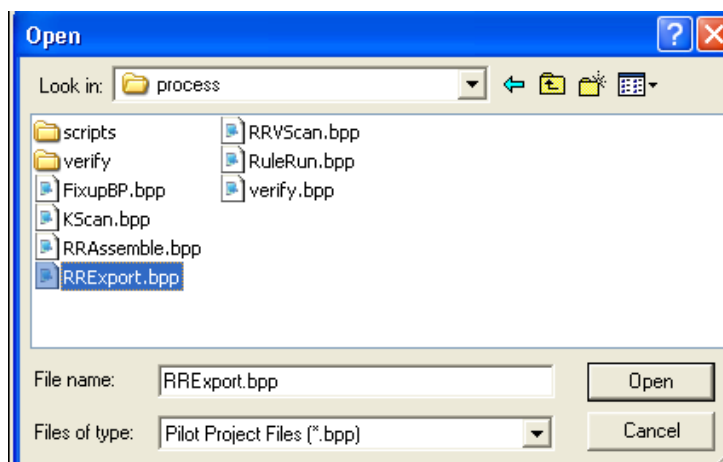
Each Task Project is more than just application-specific; it is task-specific within the application.

The graphics below and on the next few pages depict the components of the *1040EZ* application's **Export.bpp** Task Project:



1040EZ Document Hierarchy

A Document Hierarchy (above) is an application's foundation, with objects at four levels: **Batch**, **Document**, **Page** and **Field**. In this example, the *1040EZ* Document Hierarchy is a *required* component of the application's **RRExport.bpp** Task Project...and of all *1040EZ* Task Projects.



**RRExport Task Project File:
1040EZ/Process/RRExport.bpp**

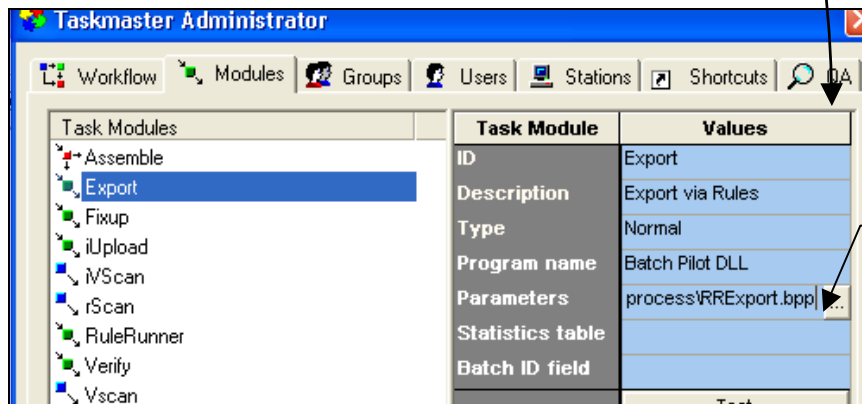
The Task Project is a file (.bpp) that usually resides in an application's **Process** directory (illustrated on the previous page.)

In this example, the *Export* Task Module connects the Main job's Export task to the **RRExport.bpp** Task Project. This Task Module is a *required* property of the Export task.



1040EZ Export Task Definition

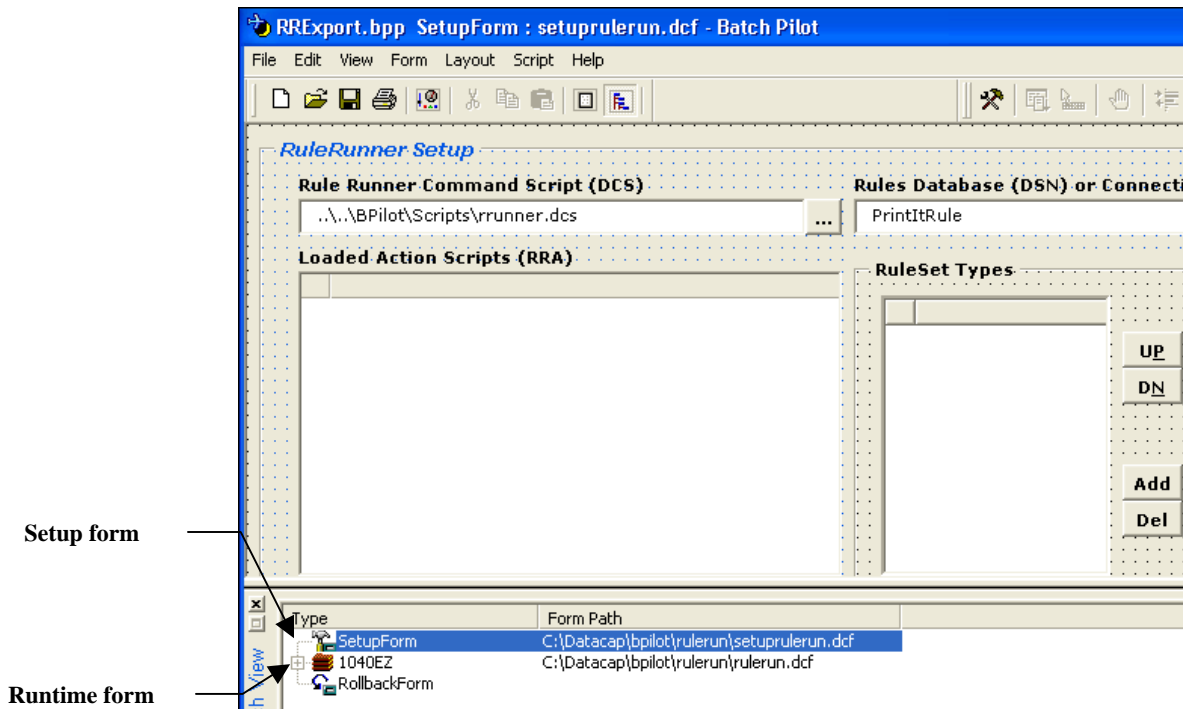
Export module



Export Task Module

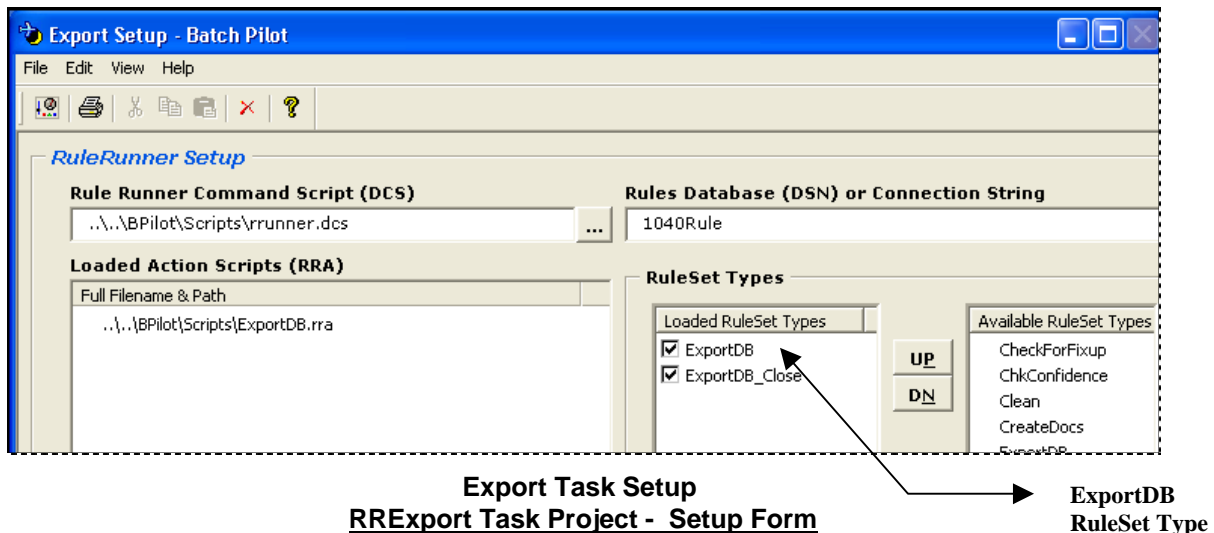
Link to RRExport.bpp

Settings of a Task Project



RRExport Task Project(.bpp)

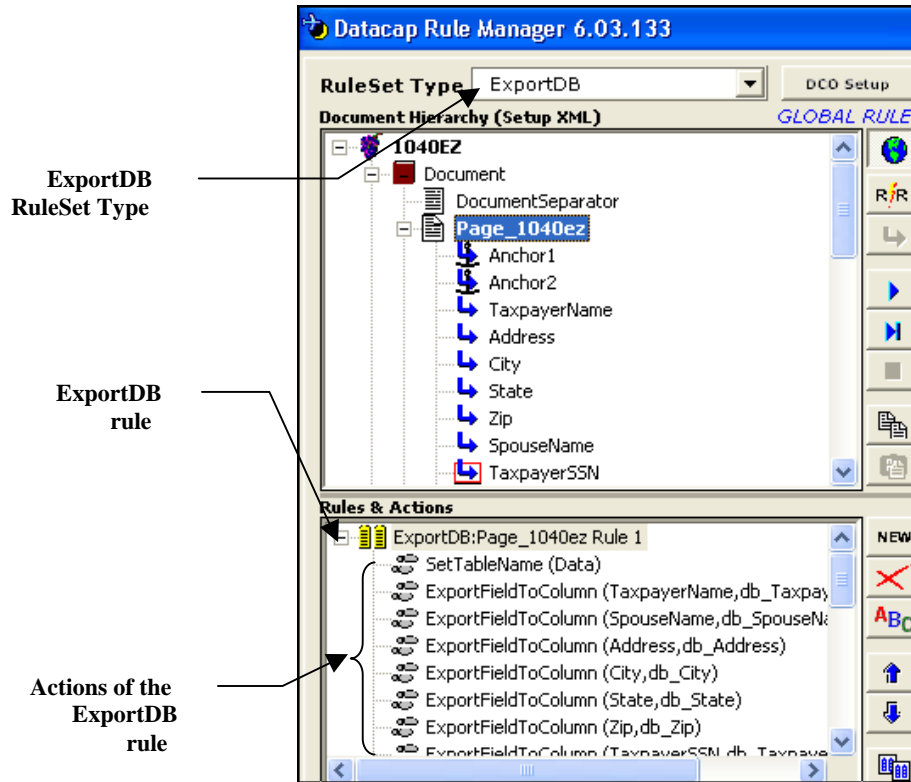
The **RRExport.bpp** Task Project has a *setup* form and a *runtime* form (Chapter 3). These forms reside in the applicable sub-folders of the **Datacap** directory's **BPilot** folder and must be assigned to the project (Page 12).



**Export Task Setup
RRExport Task Project - Setup Form**

**ExportDB
RuleSet Type**

During Export Task Definition, the *setup* form becomes a *Setup* dialog for the task and displays the fields illustrated above. Here, the dialog's **Loaded RuleSet Types** field lists the types of pre-defined rules that will guide the Export task's operations (see the example on the next page.)



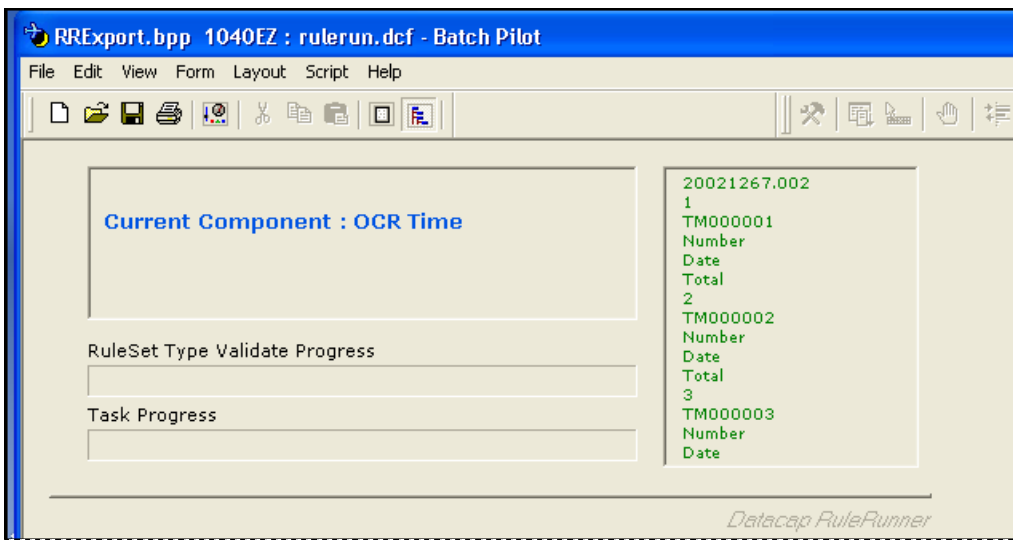
ExportDB RuleSet Type

ExportDB rule

Actions of the ExportDB rule

1040EZ Rule Manager Window - Rules panel

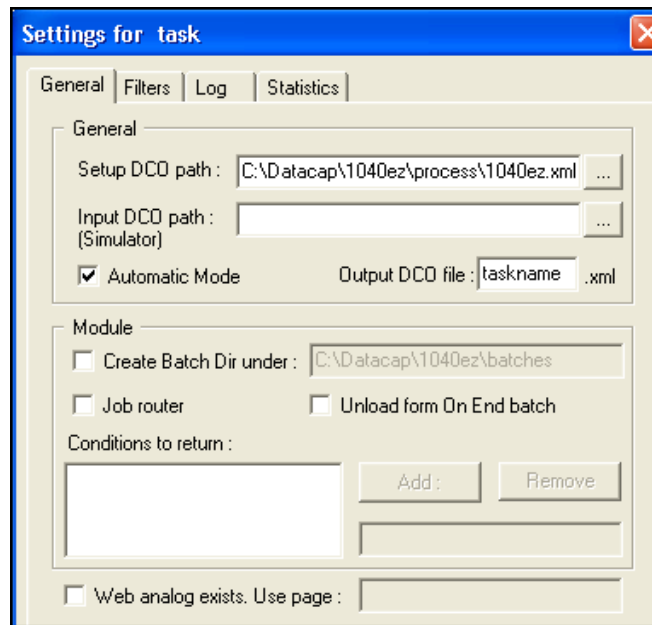
When the Export task operates, the Task Project's *runtime* form becomes the task's processing dialog, and might or might not appear in the processing background, according to a setting in the *Setup* dialog.



Export Task Operations
RRExport Task Project - Runtime dialog

Settings of a Task Project

The Task Project includes a *Task Settings* dialog with four tabs; you can assign values to settings in these tabs during the setup of the project *or* during the definition of a task based on the project:



Task Settings dialog – General tab

- ✓ Page 18 describes the settings in each tab.

The Task Project File

Again, the Task Project is a file (**taskproject.bpp**) that combines Task Definition specifications with certain Task Project specifications. You can review the file's contents with a text editor such as *Notepad*. Here is a brief excerpt from the RRExport Task Project file:

Setting and Value	Description
[General]	This section includes essential task and project values. Most specifications in this section are items in the <i>General</i> tab of the <i>Task Settings</i> dialog (for details, see Page 18 and <i>Batch Pilot</i> Help).
RootBatchDir=..\batches	Directs the task to look for batches in the application's batches directory. Although ...\batches is a default project value, the Task Definition can override it.

Task Project File (continued)

Setting and Value	Description
DCOSetup=1040EZ.xml	The identity of the Task Project's Document Hierarchy file (.xml). Alert! The Document Hierarchy is a <i>required</i> property of the Task Project and <i>cannot</i> be changed during the definition of a task based on this project.
TaskDCOFile=taskname.xml	The identity of the Page file (.xml) that the task will generate as it processes a batch. taskname.xml is a default value indicating that the task's name will also be the Page file's name – and that ".xml" will be the file's extension.
RuleDSN=1040Rule	The Data Source Name of the application's Rules database. This value is assigned during task setup.
EnableUI=True	Indicates if the <i>runtime</i> dialog will appear on the operator's screen (<i>True</i>) or will be hidden (<i>False</i>). This value is assigned during task setup.
JobRouter=0	Indicates whether or not a task based on this Task Project can route batches that encounter special conditions to <i>child</i> jobs for review and possible repair. "0" = No; "1" = Yes. (See the [Conditions] section below.) Alert! This value is assigned during task setup <i>and</i> must be reinforced by a <i>Job Router Type</i> value in the definition of the task's Task Module (see Page 5).
[iCap]	Items in this section indicate how tasks based on this Task Project can be used in a <i>Taskmaster Web</i> environment
Enabled=0	Specifies that the task is not yet available to <i>Taskmaster Web</i> .
[ReturnCond]	These settings identify the pre-defined conditions that will lead a task based on this Task Project to divert the current batch to a <i>child</i> job for additional attention. Most tasks do <i>not</i> use this feature: for details, see Chapter 10 of the <i>Guide to Taskmaster Rules</i> describes this feature.

Task Project File (continued)

Setting and Value	Description
Conditions=0	<p>Indicates that the Export task (in this example) does not encounter special conditions.</p> <p>If there were conditions, this row would be followed by rows specifying the conditions.</p>
[Batch]	<p>The settings in this section identify the Task Project forms that are bound to the project at the highest level of the Setup Tree: <i>SetupForm</i> or the <i>1040EZ Batch</i> object of the Document Hierarchy.</p>
Types=2	<p>Indicates that two forms are bound to the RRExport Task Project at the uppermost level – and are used by the project’s Export tasks (in this case.)</p> <p>Alert(1)! If a Task Project employs a Multi-tab Form with multiple tabs (Chapter 3), this value is “1”.</p> <p>Alert(2)! The Task Project file of the <i>1040EZ Verify</i> task’s project (verify.bpp) specifies one type at this level and one form (below) - the project’s setup form. This is because the runtime form is bound to the Document Hierarchy’s <i>Page_1040EZ Page</i> object (Chapter 5).</p>
Type0Form=..\bpilot\ rulerun\ rulerun.dcf	<p>The name and path of the project’s runtime form if the form is bound to the Batch object of the Document Hierarchy.</p> <p>If applicable, this setting would identify a project’s Multi-tab Form (Chapter 3).</p>
Type1Form=..\bpilot\ setuprulerun.dcf	<p>The name and path of the project’s setup form.</p>
[scripts]	<p>This specification identifies the high-level script that the Export task (and all other <i>RuleRunner</i> tasks) use to apply rules and process a batch.</p> <p>This script also contains RRunner actions that are included in the Actions Library of every RuleSet Type.</p> <p>Although this is a default value of any project that uses the RuleRunner setup form (setuprulerun.dcf), the Task Definition process can modify it.</p>
RRunner=..\BPilot\Scripts\ runner.dcs	<p>The name and path of the <i>RuleRunner</i> script.</p>

Task Project File (continued)

Setting and Value	Description
[ActionFiles]	<p>This section lists the files that contain actions that make up the rules applied by the Export task (in this example.)</p> <p>A Task Definition identifies the Action files in the <i>Task Setup</i> dialog's Loaded Actions Scripts field (illustrated on Page 6).</p> <p>Important! The list does not specify the rrunner.dcs file even though that file does include an essential set of actions.</p>
File1=..\BPilot\Scripts\ ExportDB.rra	<p>The Actions file for Export rules of the Export task.</p> <p>(Rules applied by the <i>1040EZ</i> Recognize task contain actions from multiple .rra files: each is listed in this section of the Task Project file (rulerun.bpp)).</p>
[RRuleType]	<p>This section lists all RuleSet Types with rules that the task applies. A Task Definition specifies these values in the RuleSet Types fields of the <i>Task Setup</i> dialog (illustrated on Page 6).</p>
RuleTypes=2	<p>The number of RuleSet Types used by the Export task.</p>
RuleType1=ExportDB	<p>The Export task's first RuleSet Type (in this case.)</p>
RuleType2=ExportDBCclose	<p>The Export task's second RuleSet Type.</p>

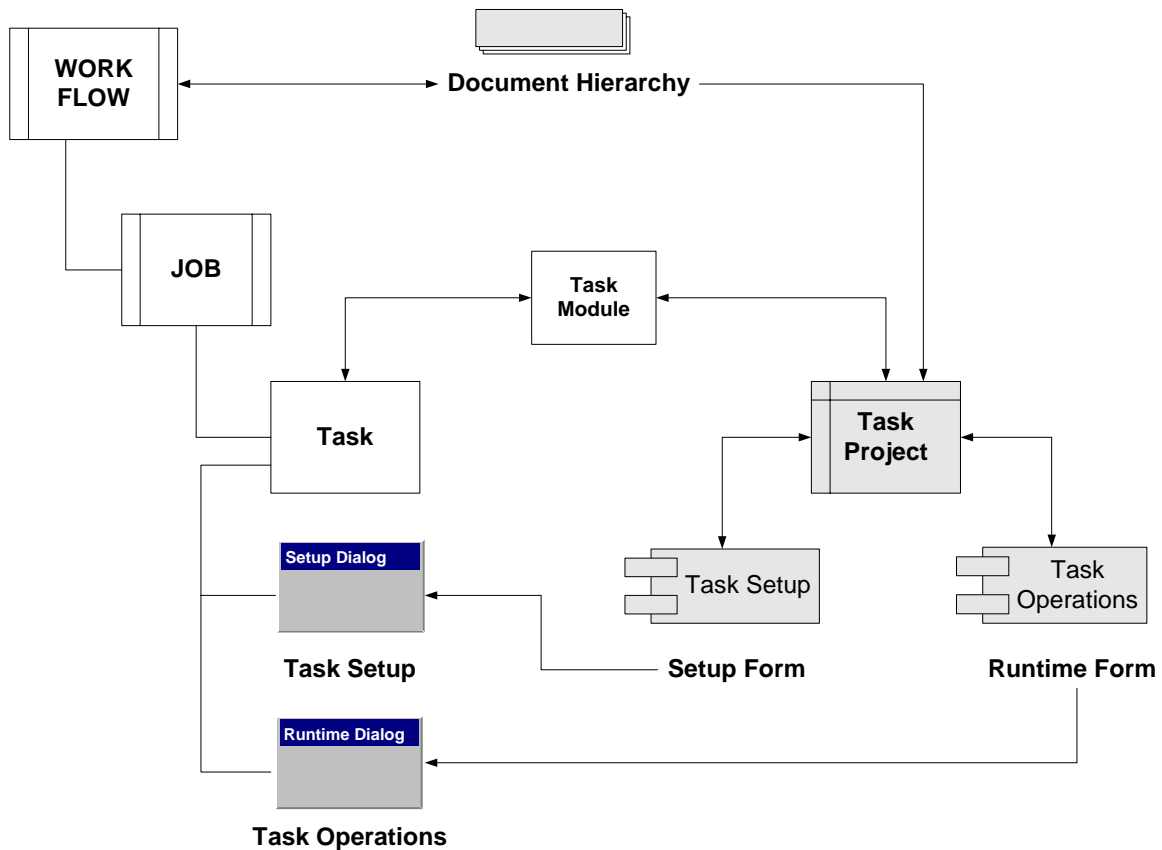
- ✓ The remainder of the file deals with the placement of the *Runtime* dialog within the operator's window when the Export task processes a batch. These are *project* rather than *task* specifications.

Please note: The table above describes certain core settings in the *RRExport* Task Project file. Some Task Projects files are considerably longer, and others are shorter. Do not hesitate to explore these files. However, we **strongly recommend** that you do not modify the values in the files; instead, change settings in the Task Project or Task Definition interfaces.

How to Define a Task Project

The chart below emphasizes the critical role of the Task Project as a conduit between a formal Task Definition that assigns a task to a job, and the task's two essential dialogs: the *setup* dialog that determines the task's scope and dimensions; and the *runtime* dialog for task operations. Alternatively, as Chapter 3 explains, a Task Project can employ a single Multi-tab Form with setup and runtime tabs.

The chart also highlights the contribution of the application's Document Hierarchy to the Task Project.

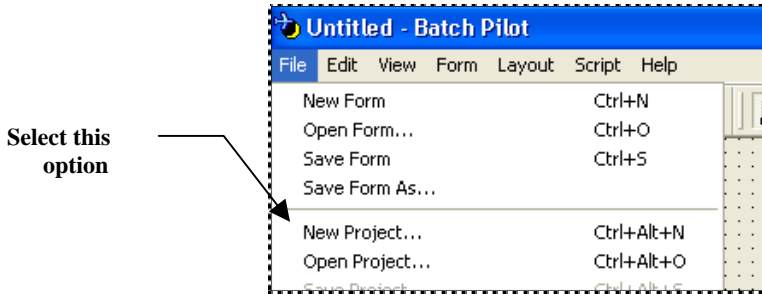


As soon as the Document Hierarchy and the applicable forms are in place, defining a Task Project is straightforward – just follow the steps on the next page.

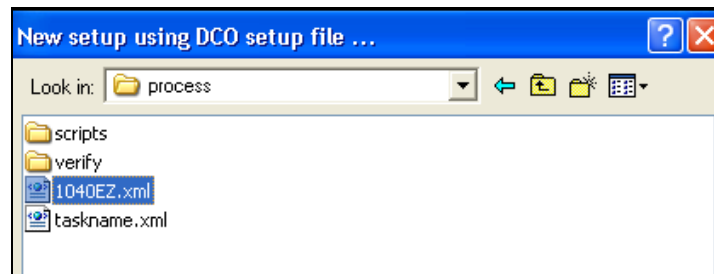
- ✓ This discussion refers to a new *1040EZ* task based on a Task Project you assemble to export the recognized and verified values in a batch to an Export **file** (.txt) rather than to an Export **database**.

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

1. Open *Batch Pilot*.
2. Select **New Project** from the *Batch Pilot Window's* **File** menu.

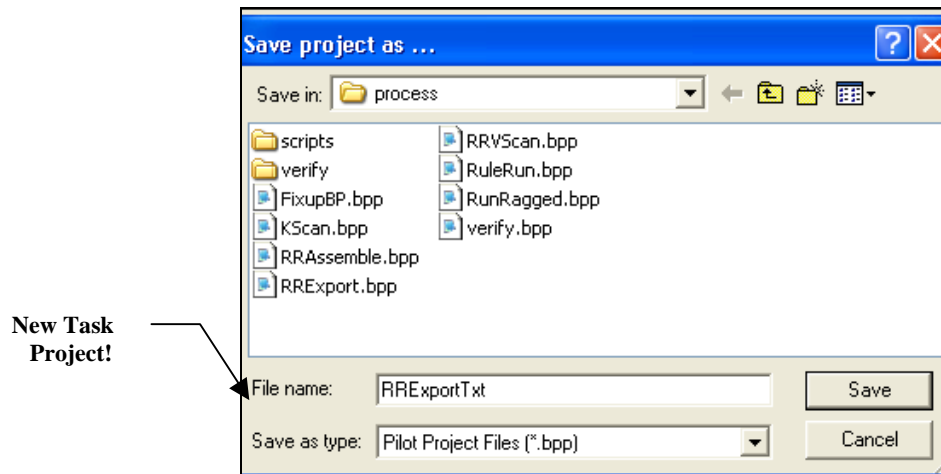


3. Use the tools of the *Open File* dialog to locate and open the XML file that contains your application's Document Hierarchy (**1040.xml**, in the example). This important step completes an invaluable link between your application and this Task Project well *before* you begin to define the task.



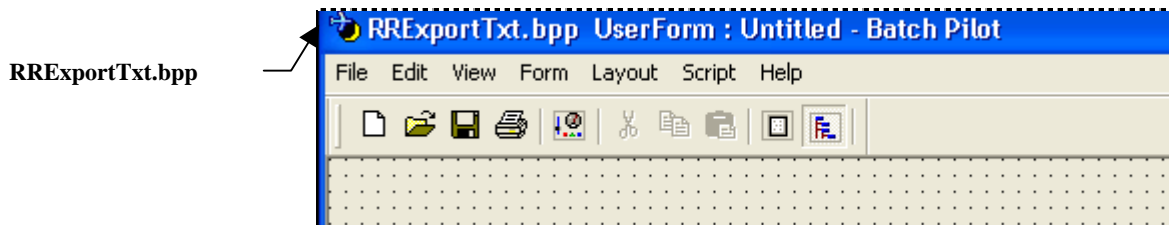
4. When the *Batch Pilot Window* returns to your screen, select the **File** menu's **Save Project as...** option to save the new Task Project file (.bpp) – typically in the application's **Process** directory (illustrated on the next page.)

How to Define a Task Project

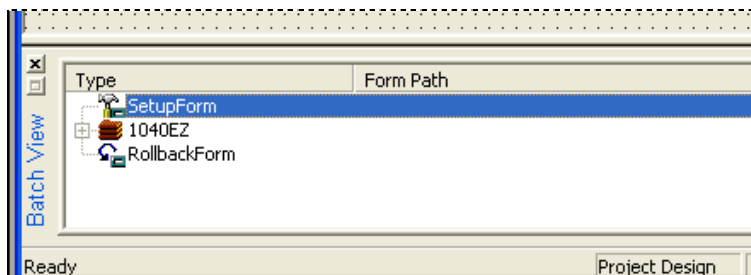


How to Define a Task Project (continued)

Step	Action
5.	Confirm that the <i>Batch Pilot Window's</i> Title Bar displays the Task Project's name.



6. If the **Batch View** area at the bottom of the window is *not* visible, select **Setup Tree** from the **View** menu. *Alert!* At this early stage, the Setup Tree displays three obvious rows:



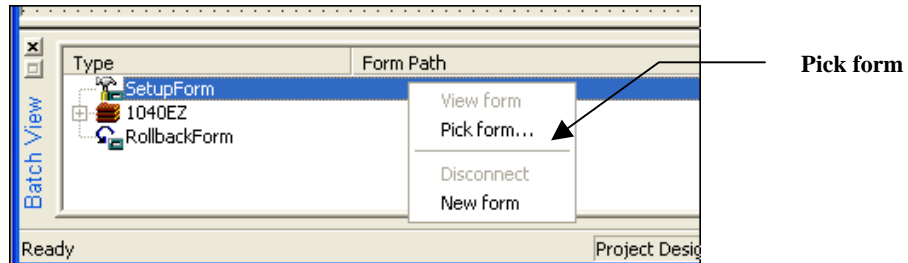
SetupForm sits alone at the top and will accept the name and path of the *setup* form you assign to the project (Step #8). *1040EZ* is at the **Batch** level of the Document Hierarchy – and will accept the name and path of the project's *runtime* form in this case (Step #11). (Remember, too, that this row would be used for the assignment of a single Multi-tab Form.)

RollbackForm is inactive.

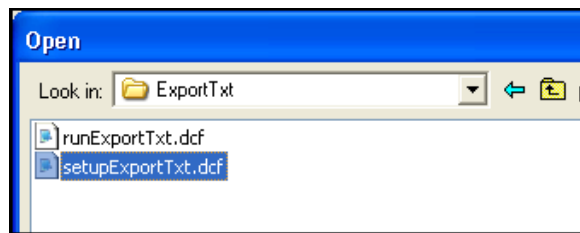
How to Define a Task Project (continued)

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

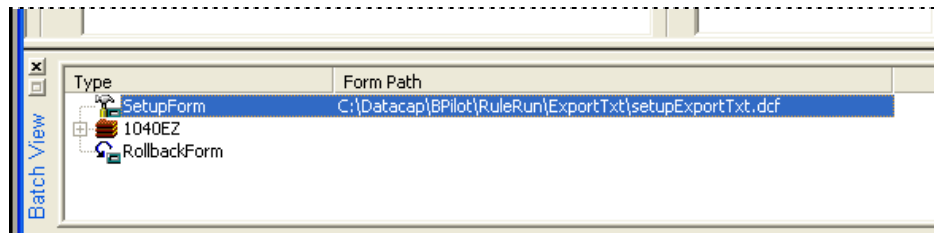
- Right-click on the *SetupForm* row and select **Pick form**.



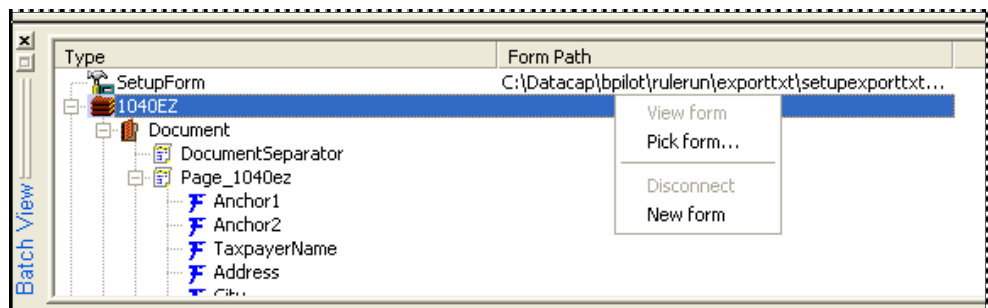
- Use the tools of the *Open File* dialog to locate and open the *setup* form (.dcf) you intend to assign to the Task Project.



- Confirm that the form's name and path appear in the *SetupForm* row of the Setup Tree - and un-toggle the **Form** menu's **Design** option to get a good look at the *setup* form.



- Right-click on the *1040EZ* row (in this example) and select **Pick form**.

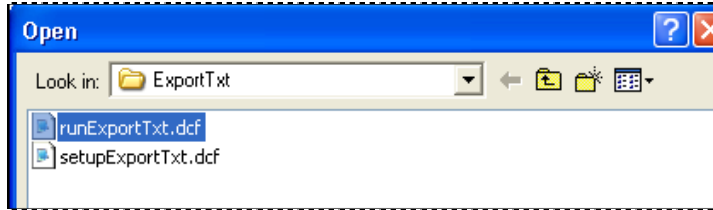


How to Define a Task Project

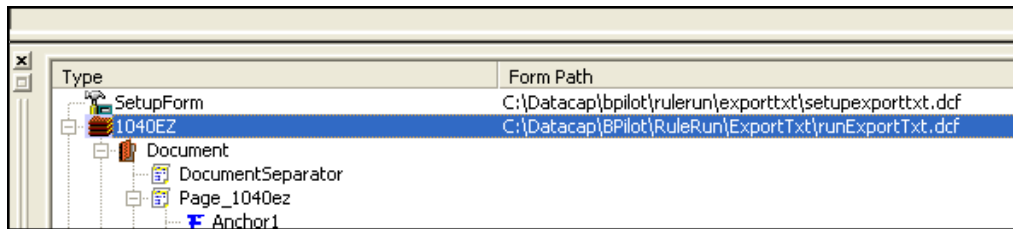
How to Define a Task Project (continued)

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

11. Use the *Open File* dialog to locate and open the Task Project's *runtime* form.

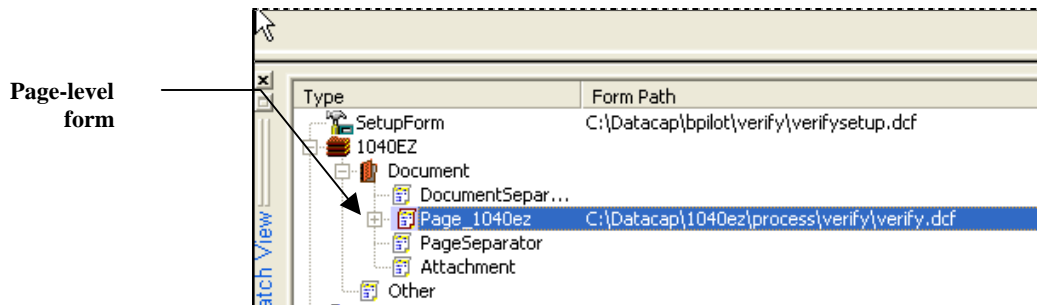


12. Make sure that the form's name and path appear in the *1040EZ* row of the Setup Tree— and un-toggle the **Form** menu's **Design** option to get a good look at the form.



13. Select **Save Project** from the **File** menu to save the new **Task Project**!

Very important! The Task Project that is the foundation for a typical Verify task assigns its *runtime* form to a **Page** object of the Document Hierarchy. This is because the form is a **Data Entry** panel that processes recognized values for each page in a batch...page after page. Here's an example:



Chapter 5 examines Verify Task Projects and *Data Entry* forms.

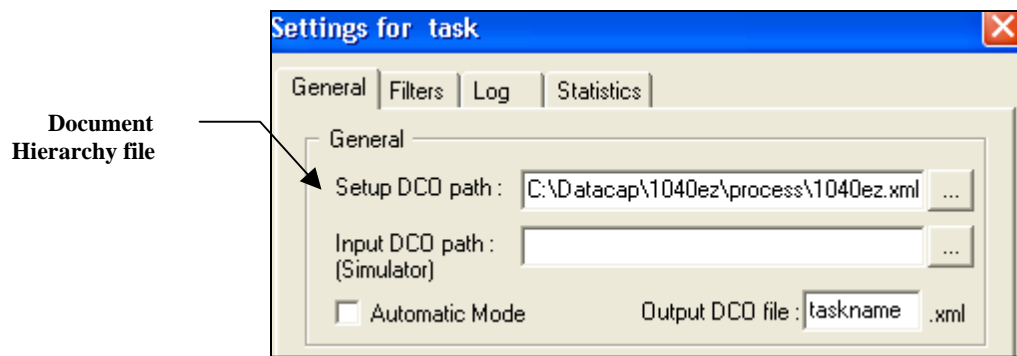
Auditing the New Project

There are two ways to check on a new Task Project before you attempt to define a task that employs the project and its forms.

Task Settings

After you open the Task Project in the *Batch Pilot Window*, select **Task Settings** from the **File** menu to access the *Task Settings* dialog. (The next page describes this dialog.)

Although the majority of settings and values in the tabs of this dialog are assigned when the task is defined, one setting belongs exclusively to the Task Project: the identity of the Document Hierarchy file (aka “Setup DCO”) that you assigned in Step #3.



Task Settings dialog – *General tab*

Task Project File

The new Task Project file – **RRExportTxt.bpp**, in this case – should have specifications associated with the project. For example:

[General]

DCOSetup=1040ez.xml The identity of the application’s Document Hierarchy file (.xml)

[Batch]

Types=2 The number of forms assigned to the Document Hierarchy’s Batch level.

Type0=SetupForm The name of the object at the Batch level to which the first form has been assigned.

Type0Form=..\..\bpilot\
rulerun\exporttxt\
setupexporttxt.dcf The name and path of the *setup* form’s file (.dcf).

Type1=1040EZ The name of the object at the Batch level to which the second form has been assigned.

How to Define a Task Project

```
Type1Form=..\..\bpilot\  
rulerun\exporttxt\  
runexporttxt.dcf
```

The name and path of the *runtime* form's file (.dcf).

Task Projects and Task Definitions

As the chart on Page 12 illustrates, a Task Module connects a task to its Task Project and, therefore, to the project's *setup* and *runtime* forms.

Task Definition is a process with these stages:

- Stage 1 assembles the Task Project.
- Stage 2 defines a Task Module: its **Parameter** property is the link to the Task Project.
- Stage 3 provides the task with a unique identify – and with the identity of its Task Module.
- Stage 4 uses the *Task Setup* dialog and the *Task Settings* dialog to complete the Task Definition.

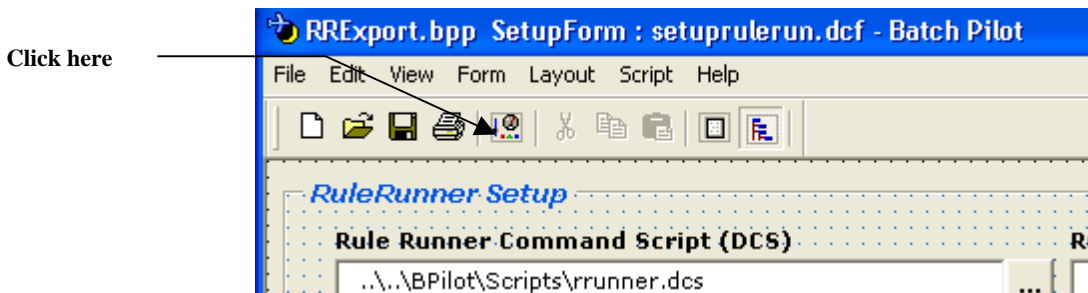
Chapter 6 of the *Guide to Taskmaster Rules* fully describes this process. Other chapters of that text concentrate on the setup and operation of tasks in these categories: Scanning, RuleRunner (Page Identification, Recognition and Export), Verification and FixUp.

Task Settings Dialog

The *Task Settings* dialog has three exceptional attributes:

- It is a *Batch Pilot* device.
- It gathers limited information exclusively for a particular Task Project;
- It gathers additional information for a task based on the Task Project.

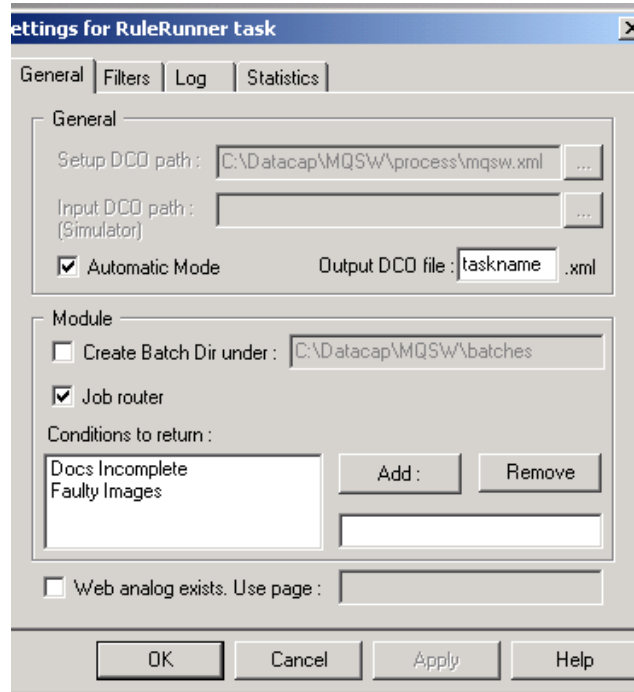
You can access this dialog by clicking on the **Task Settings** icon in the *Batch Pilot Window's Main* toolbar; by selecting **Task Settings** from the **File** menu; or by pressing the **Ctrl+T** keyboard combination.



Batch Pilot Window – *Main toolbar*

In addition, you can access the dialog when you are defining a Task Project - or a task based on the project. However, we ***strongly recommend*** that you limit your involvement with the ***Task Settings*** dialog to the Task Definition process. Chapter 6 of the *Guide to Taskmaster Rules* describes this process, and provides detailed explanations of specifications in the tabs of the ***Task Settings*** dialog. Tables on the next few pages summarize these explanations; *Batch Pilot* Help also describes the dialog and its settings.

Task Settings dialog: General tab

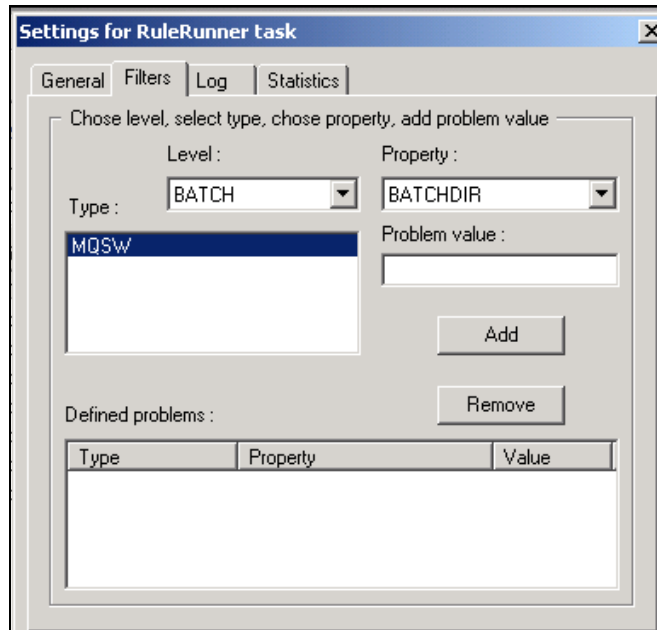


Task Settings Window – RuleRunner Task

Setting	Explanation
Setup DCO Path	The name and path of the underlying Task Project’s Document Hierarchy.
Input DCO Path	The name and path of a task’s Page file (.xml) that can be used for testing the Task Project. (Page 26).
Automatic Mode	A checkbox which, if selected, allows the task to run in the processing background, without operator participation.
Output DCO file	The name of the Page file (.xml) generated by a task based on this Task Project. The default “taskname” value automatically combines the task’s name with the .xml extension.
Module area	Settings in this area supplement the default characteristics that result when an exceptional Task Module is assigned to the task .

Task Settings dialog – *General tab (continued)*

Setting	Explanation
Create Batch Dir under:	A checkbox which, if selected, designates the name and path of the application's Batch Directory in the accompanying field.
Job Router	<p>A checkbox which, if selected, indicates that the Task Definition has assigned a Task Module with <i>Job Router</i> capabilities.</p> <p>This means that the task can route batches that encounter exceptional conditions to a <i>child</i> job for review and possible repair.</p> <p>For details, see Chapter 10 of the <i>Guide to Taskmaster Rules</i>.</p>
Add button	Adds a processing Condition you have entered in the data field below to the list of Conditions to Run .
Remove button	Deletes a processing Condition you have entered in the data field below.
Conditions to Return	<p>A list of processing conditions that the task might encounter.</p> <p>You can insert conditions <i>only</i> if you select the Job Router option and do not select the Create Batch Dir option (see Chapter 10 of the <i>Guide to Taskmaster Rules</i>.)</p>
Web analog exists. Use page:	<p>A checkbox which, if selected, indicates that the task runs in a <i>Taskmaster Web</i> environment using the Active Server Page (asp) you specify in the Use page field.</p> <p>For a close-up look at an example, highlight the pre-configured <i>Upload</i> task of the <i>1040EZ</i> Web Job. Click on the Setup button, and open the <i>General</i> tab of the Task Settings dialog.</p> <p>If you select this option, be sure to enter the name of the Active Server Page (.asp) file in the accompanying field.</p>
OK button	Saves current specifications and closes the Task Settings dialog.
Cancel	Closes the Task Settings dialog without saving new or modified specifications.
Apply	Saves current specifications but does not close the Task Settings dialog.

Task Settings dialog – *Filters tab*

Task Settings dialog: Filters tab

Settings in the *Filters* tab provide parameters the task can use to filter the contents of a batch *before* processing begins – or even to decide whether or not to process the batch itself. (For a full explanation, see Chapter 6 of the *Guide to Taskmaster Rules*.)

Alert! If you use this tab to construct a filter, the tab will process only those objects of the Document Hierarchy that meet the filter’s criteria. For example, a Verification task might only select **documents** with a processing **Status** of “146”.

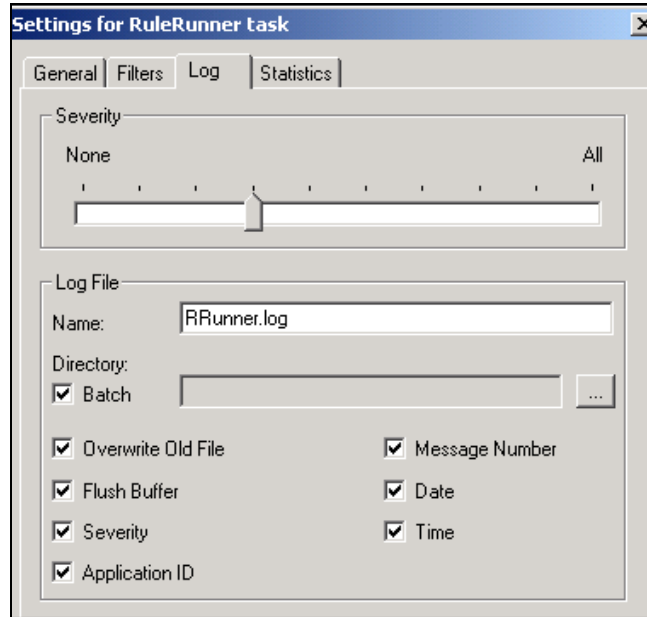
Setting	Explanation
Level	<p>A drop-down list of Document Hierarchy levels: <i>Batch</i>, <i>Document</i>, <i>Page</i>, <i>Field</i>.</p> <p>Important! Although you can select any level, properties for that level will appear in the Property list to the right only if you have bound a project form to an object in the Batch View area’s Setup Tree.</p>
Type	Confirms the identity of the Setup Tree level you’ve chosen.

Task Settings dialog – Filter tab continued

Setting	Explanation
Property	<p>A drop-down list of the properties of Document Hierarchy objects at the level to which you have bound the project's <i>Runtime</i> or <i>Setup</i> form.</p> <p>There are three fundamental considerations:</p> <p>Although filtering involves one level of the Document Hierarchy, you can establish multiple criteria at that level.</p> <p>The properties of a particular level become available only after you bind a form to that level.</p> <p>During Task Definition, an Administrator can review and modify the filtering parameter in the tab's Defined Problems area.</p>
Problem Value	<p>A value for the property you've selected from the Property drop-down list.</p> <p>The task will use this value as the basis for an Accept/Reject processing decision.</p>

Task Settings dialog: Log tab

Logs can be exceptionally helpful, especially when you're setting up and testing a project's tasks. The settings in this tab determine the nature of a log, its location (typically within a batch) and its contents.



Task Settings dialog – Log tab

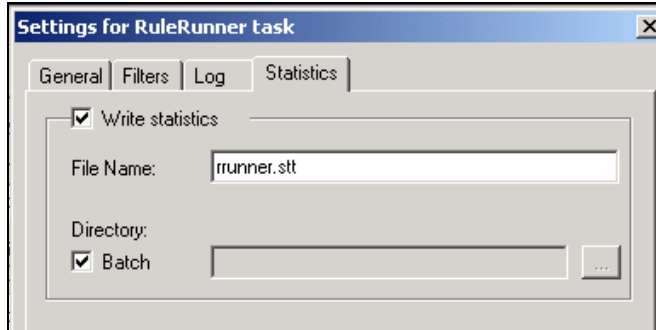
Setting	Explanation
Severity	<p>This continuum measures the relative importance of problems that the task is likely to encounter and report on in the Log file it provides for each batch.</p> <p>As the pointer moves to the right, severity drops while the scope and number of reported problems increases.</p> <p>If you move the pointer leftwards, only the more important problems are considered and, steadily, the number of logged problems diminishes.</p>
Log File	The name of the Log file that the task will add to a batch.
Directory	<p>A check box which, if activated, directs <i>Taskmaster</i> to place Log files in the Batch folders of the application's Batches directory.</p> <p>If you do not select this option, you <i>must</i> enter the name and path of the directory that is to hold Log files in the accompanying field.</p>

Task Settings dialog - Log tab (continued)

Setting	Explanation
Overwrite Old Files	A check box which, if activated, directs <i>Taskmaster</i> to replace an earlier Log file with the current file if they share a name and location.
Flush Buffer	<p>A check box which, if activated, means that <i>Taskmaster</i> will continuously transfer messages from a mid-stream buffer to the log as processing continues.</p> <p>If you do not select this feature, <i>Taskmaster</i> will store messages in a buffer for a particular period. This alternative increases processing speed but compromises safety: if an error occurs, crashes, you may lose the messages in the buffer.</p>
Severity	Adds a Severity Code to each message in the log.
Application ID	Adds an application code to each message.
Message Number	Includes a unique Message Number with each message in the log.
Date	Adds the Date.
Time	Adds the Time.

Task Settings dialog: Statistics Tab

Settings in this tab authorize the task to generate a special Statistics file (.sts) when it completes its work with a batch.



Task Settings dialog – Statistics tab

Setting	Explanation
Write Statistics	A checkbox which, if activated, directs the task to compose a Statistics file with details of its operations with a batch.
File Name	The name and extension of the Statistics file.
Directory	A check box which, if activated, directs <i>Taskmaster</i> to place Log files in the Batch Folders of your application’s Batches directory. If you do not select this option, use the accompanying field to enter the name and path of the directory that is to hold Statistics files.

How to Test a Task Project

A fully configured Task Project consists of:

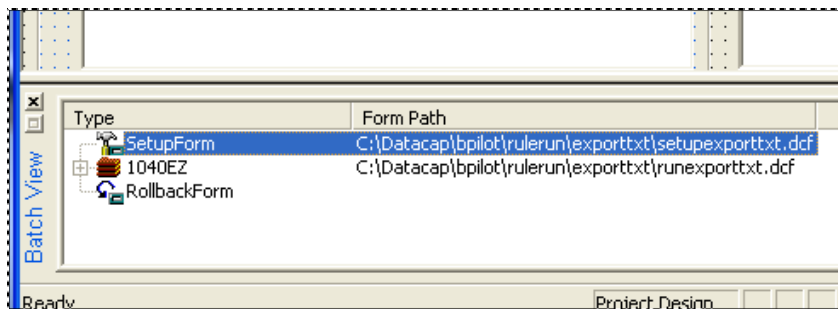
- The designation of the application’s Document Hierarchy as the project’s key property
 - A Task Project file (.bpp) - Page 8
 - The project’s *setup* form (.dcf) (or a Multi-tab form) - Page 6
 - The project’s *runtime* form (dcf) – Page 6
 - A Task Module to connect a Task Definition to the project – Page 18
 - A Task Definition, including specifications in the *Task Settings* dialog – Page 18.
- ✓ You cannot test the Task Project completely and effectively until all these elements are in place.

When you test a Task Project, you can separately assess its task *setup* mechanism, and its *runtime* procedures.

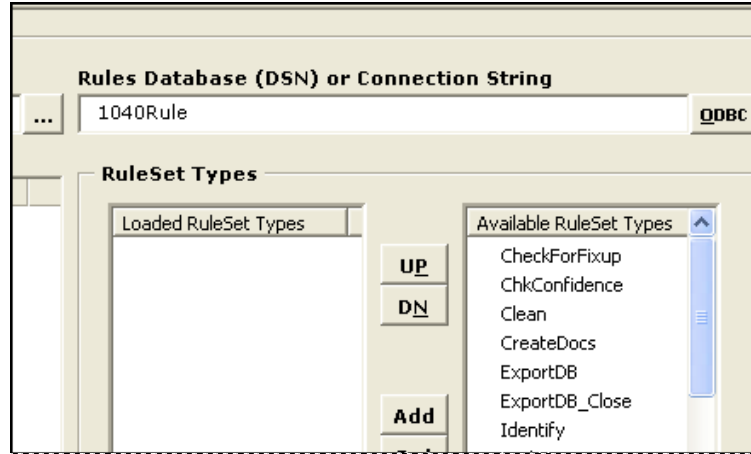
Testing Setup Procedures

To test the project’s ability to setup a task that is based on the project, take these steps:

Step	Action
1.	Open the project in the <i>Batch Pilot Window</i> .
2.	Open the Setup Tree in the Batch View area.
3.	Highlight the Setup Form row. In the case of a Multi-tab Form, highlight the form bound to the Document Hierarchy’s Batch object (<i>1040EZ</i> , in the example.)
4.	Select Task Setup from the <i>Batch Pilot Window’s File</i> menu.



If the setup elements of the Task Project are intact, *Batch Pilot* will display a copy of the task’s *Task Setup* dialog, with each control both available and working. At this point, you should be able to “mimic” the task’s setup steps, using these controls.



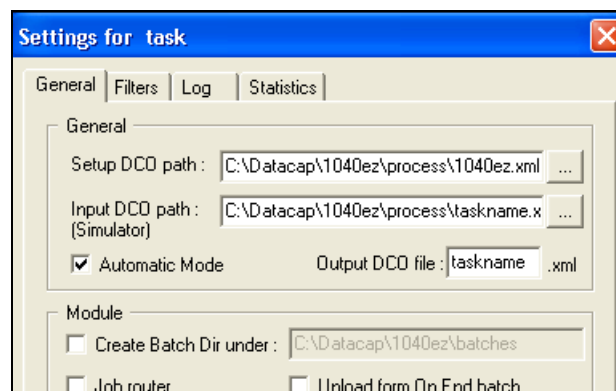
RRExportTX Task Project – Setup

Testing Runtime Procedures

Details of this procedure depend on the nature of the Task Project you’re testing – and on values you’ve placed in the *General* tab of the *Task Settings* dialog.

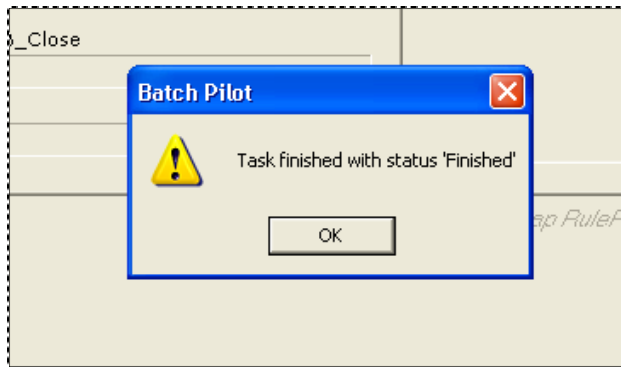
To test the project’s *runtime* settings and abilities:

Step	Action
1.	Open <i>Batch Pilot</i> and the <i>Batch Pilot Window</i> .
2.	Select Open Project from the File menu to open a Task Project such as the <i>1040EZ</i> application’s RRExport.bpp project.
3.	Click on the Task Settings icon to open the <i>Task Settings</i> dialog form this project.



How to Test a Task Project's Runtime Settings (continued)

- | Step | Action |
|------|--|
| 4. | Be sure that the name and Setup DCO Path value is accurate (Page 6). |
| 5. | The Input DCO Path identifies a default Page file (.xml) that the Task Project can use for testing. Alternatively, you can use the Ellipsis button to locate and select an actual Page file in a specific batch (if you have run the application's workflow through its Verification task, in this case.) |
| 6. | Press the OK button at the bottom of the <i>General</i> tab. |
| 7. | Select Simulate Task from the <i>Batch Pilot Window's</i> File menu. |
| 8. | Press the OK button when <i>Batch Pilot</i> asks if you would like to save the project. |
| 9. | Pause briefly while <i>Batch Pilot</i> conducts the test. |
| 10. | If everything goes well, <i>Batch Pilot</i> will let you know with a message similar to this: |



Logs of Task Project Runtime Tests

Batch Pilot will generate a log of your runtime tests if you take these steps:

- | Step | Action |
|------|--|
| 1. | Use a Text Editor such as <i>Notepad</i> to open the bpilot.ini file in your Datacap directory's BPilot folder. |
| 2. | In the [log] sector, enter <code>Severity=8</code> . |
| 3. | In the [log] sector, enter the name and path of the Log file. For example:
<code>Path=C:\Datacap\1040EZ\Process\TestLog.</code> |

