8

Testing Your Application

Tools and procedures of the **Test Page** help you systemically evaluate whether the rules you have constructed on the **Rulemanager Page** process documents as you intend. They can also be used as debugging tools: if you run across problems in a production environment, the **Test Page** can help you trace the source of the problem.

Chapter 5 describes these tools and procedures as it covers the following topics:

8.1. Introduction 8-2

8.1 What is a Task Profile? 8-4

8.1.1 How to Construct a Task Profile 8-4

8.1.2 Running Task Profiles 8-6

8.2 Debugging your Application 8-14

8.2.1 Setup Tabs 8-14

8.2.2 Diagnostic Tabs 8-16

8.2.2.1 Breakpoints tab 8-16

8.2.2.2 Runtime State Tab 8-18

8.3 Test Page Reference 8-25

8.3.1 Test Page Toolbar 8-25

8.3.2 Runtime Batch Hierarchy Tab 8-27

8.3.2.1 Victim Objects 8-30

8.1. Introduction

The **Test Page** consists of the three sections illustrated on the next page.

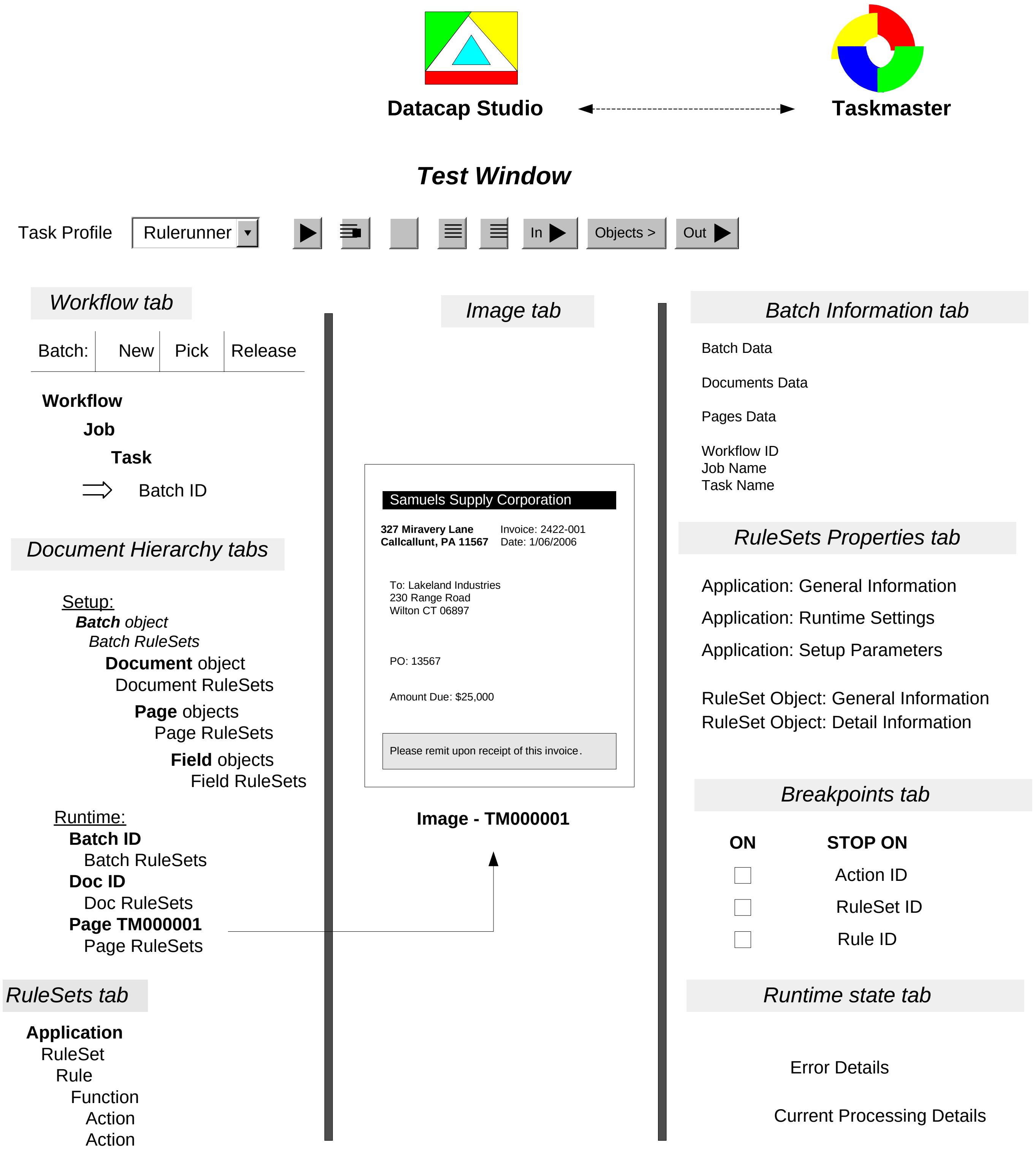
Testing combines the toolbar's icons with buttons, settings and to carry out tests of your Task Profiles, Ruleset/Rule combinations, rules, and fingerprints.

Images and the text values of the pages being processed.

Information, including object properties, current status of processing, and logging information.   Additional information is available about the application itself – including lists of Ruleset/Rule combinations assigned to individual Task Profiles.

By default, these three sections appear from left to right inside the **Test Page**. Those locations, however, are flexible and can be reorganized by the user.

✓ Chapter 8's principal focus is on the first section…on Datacap Studio's testing procedures. After a review of preparatory steps – using the *1040EZ* application for examples – the chapter describes sequences that test each ***setup*** and ***runtime*** element. Where applicable, these descriptions refer to steps you would take to test components of other applications, including ***your*** application.



Datacap Studio ◄----------► Taskmaster

Test Window

Task Profile | Rulerunner ▾ | ▶ | In ▶ | Objects > | Out ▶

Workflow tab

Batch: | New | Pick | Release

Workflow

Job

Task

⇒ Batch ID

Document Hierarchy tabs

Setup:

***Batch** object*

Batch RuleSets

Document object

Document RuleSets

Page objects

Page RuleSets

Field objects

Field RuleSets

Runtime:

Batch ID

Batch RuleSets

Doc ID

Doc RuleSets

Page TM000001

Page RuleSets

RuleSets tab

Application

RuleSet

Rule

Function

Action

Action

Image tab

Samuels Supply Corporation

327 Miravery Lane
Callcallunt, PA 11567

Invoice: 2422-001
Date: 1/06/2006

To: Lakeland Industries
230 Range Road
Wilton CT 06897

PO: 13567

Amount Due: $25,000

Please remit upon receipt of this invoice.

Image - TM000001

Batch Information tab

Batch Data

Documents Data

Pages Data

Workflow ID
Job Name
Task Name

RuleSets Properties tab

Application: General Information

Application: Runtime Settings

Application: Setup Parameters

RuleSet Object: General Information

RuleSet Object: Detail Information

Breakpoints tab

ON	**STOP ON**
□	Action ID
□	RuleSet ID
□	Rule ID

Runtime state tab

Error Details

Current Processing Details

8.1 What is a Task Profile?

A Task Profile lists the Rulesets in the order in which they will be processed when submitted to the Rulerunner Service.

8.1.1 How to Construct a Task Profile

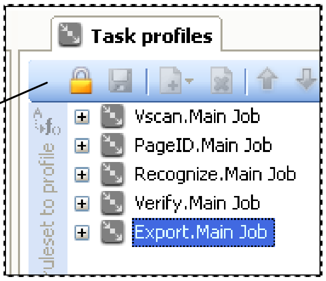
Task Profiles are created in the Task Profiles tab that shows up on the Rulemanager and Test Pages of Datacap Studio.

To Assemble a Task Profile:

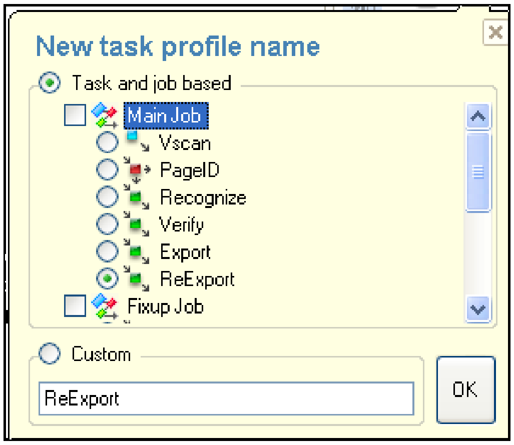
Step **Action**

1. Open the *Task Profiles* tab and toggle the **Lock** icon (!).

Toggle this icon.

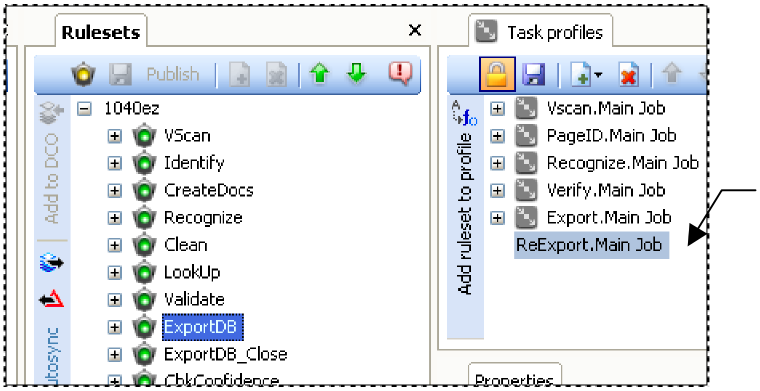


2. Press the + icon on the toolbar. Select the **Task and Job Based** option in the ***New Task Profile Name*** dialog, and the Job ID.



3. Enter the task’s name in the field at the bottom and press the OK button.

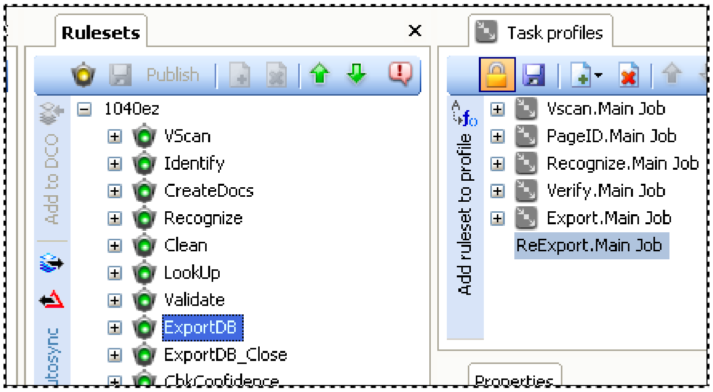
4. Confirm that the Task/Job ID appears in the Task Profiles list.



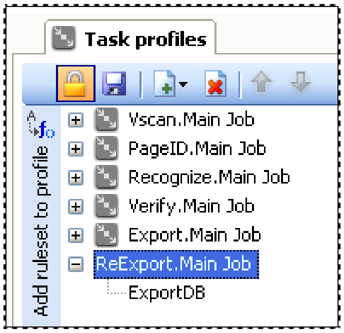
Task/Job ID

To Assemble a Task Profile (continued)

Step	Action
5.	Open the *Task Profiles* tab and toggle the **Lock** icon (!).
6.	In the *Rulesets* tab, highlight the task's opening Ruleset and press the **Add Ruleset to Profile** icon in the *Task Profiles* tab.



7.	Confirm that the Ruleset is part of the Task Profile.



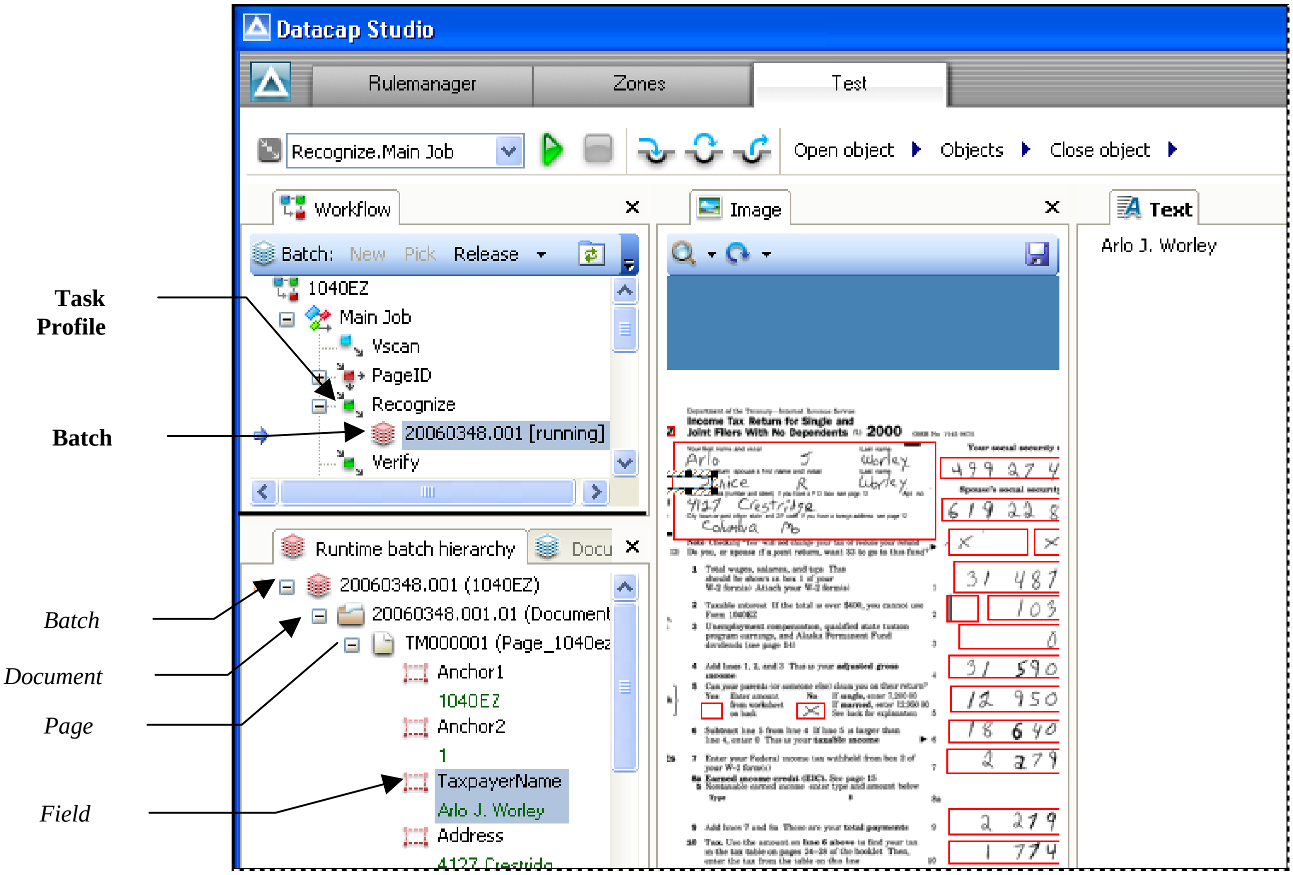
8.	Repeat Step # 6 to add another Ruleset to the Task Profile.

8.1.2 Running Task Profiles

Procedures for running Task Profiles are simple and straightforward.

Task Profile operations involve four tabs of the **Test Page**:

- The *Workflow* tab lists jobs and their Task Profiles.
- The *Runtime Batch Hierarchy* tab shows the contents and organization of the batch after processing by the current Task Profile.
- The *Image* tab displays the image of a page you select from the *Runtime Batch Hierarchy* tab.
- The *Text* tab reveals the text value of a field you select after a Recognition or Verify task has processed the batch.



Datacap Studio – *Test Page*

The *Workflow* tab lists the *Taskmaster* jobs and tasks which you can run to test rules..

✓ Although Rulerunner Service can be run from *non-Taskmaster* environments, in order to test in the **Test Page**, you need at least a single *Taskmaster* job and task.

- To start a task, highlight its Task ID and press the **Run** icon ▶ in the **Test Page** toolbar. (To stop a task, click on the **Pause** ■ icon.)
- Once you are underway, you will find batch details in the *Workflow* tab and in the *Runtime Batch Hierarchy* tab.
- If you select a Page ID, its image will appear in the *Image* tab.
- After you have processed a Task Profile such as Recognize in the example above, the *Text* tab will display a field's text value.

The **Select Task Profile** drop-down list on the left edge of the toolbar moves a batch to another Task Profile for testing.

✓ If the task name and the Task Profile name correspond, exactly, then Datacap Studio will automatically match them up, so that when a batch advances from one task to the next, the Task Profile associated with that batch will also advance. Of course, if they do *not* correspond, then it is up to the user to select the correct Task Profile.

✓ Non-Taskmaster users can simply advance the Task Profile without advancing the task in the *Taskmaster* job. It will work exactly the same from Datacap Studio – but *not* from *Taskmaster.*

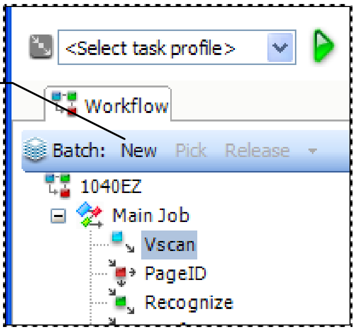
Icons of the *Workflow* tab help manage the testing of Task Profiles, and include:

Icon	Description
Batch	An inactive icon indicating that tests of Task Profiles run at the batch level.
New	Assigns a new *Taskmaster* batch to the selected Task. The batch creation task, e.g. Scan, VScan, should be selected. ***Very important!*** DStudio will not generate an actual batch until you follow up by clicking on the **Run** icon ▶ in the **Test Page** toolbar.
Pick	Activates a batch for further testing by a Task Profile. When multiple batches are in line for processing by a Task Profile, you use this tool to identify the batch you want to test.
Release	Assigns a status to a batch, corresponding to Taskmaster statuses: *Advance:* DStudio will automatically move the batch to the next task with a status of *Pending*. *Hold:* usually for batches that have started but not completed processing. *Pending:* the batch is available for processing. *Finish:* completes the batch and advances it to the next task, if there is one. This is equivalent to Advance in most situations. *Running:* assigned by DStudio to the batch that you have picked for active testing. *Abort :*sets the batch aside and prevents it from further processing in Datacap Studio. *Cancel:* similar to Abort, though the batch becomes invisible to *Taskmaster* as well..
Update View	Refreshes the view of jobs and tasks from *Taskmaster*.
Prompt Messages	Selections that generate optional prompts during testing. *Prompt on Task Profile* results in a message that alerts you if a Task Profile is applying rules to a batch for a second (or third, etc.) time. *Batch Release Prompt* directs DStudio to ask if you would like to advance the batch to the next Task Profile – after the test has finished with the current Task Profile.

To run a batch through a set of Task Profiles (using the *1040EZ* application for practice), take these steps:

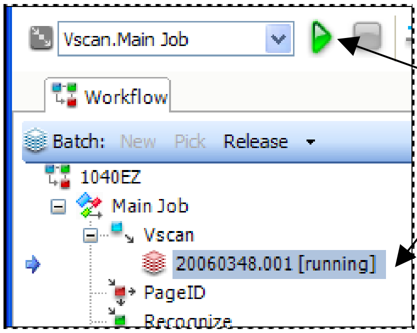
Step **Action**

1. In the *Workflow* tab, select the Workflow ID (1040EZ), Job ID(Main Job) and Task ID (Vscan) – in that order. Confirm that the tab's **New** icon is active.



New icon

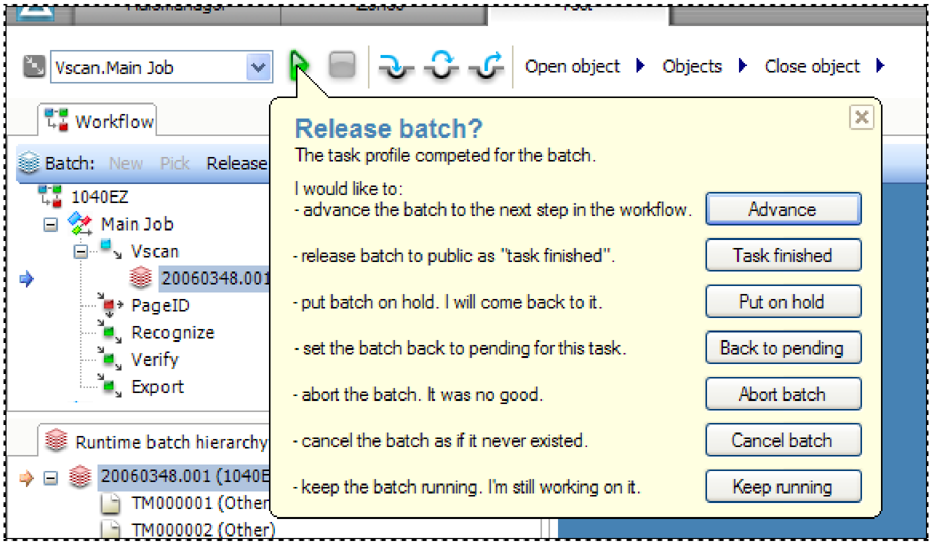
2. Press the **New** icon to add the shell of a new batch to the Vscan Task Profile.



Click here.

New batch

3. Press the **Run** icon ▶ to initiate the Ruleset/Rule combinations of the VScan Task Profile.



Release batch?

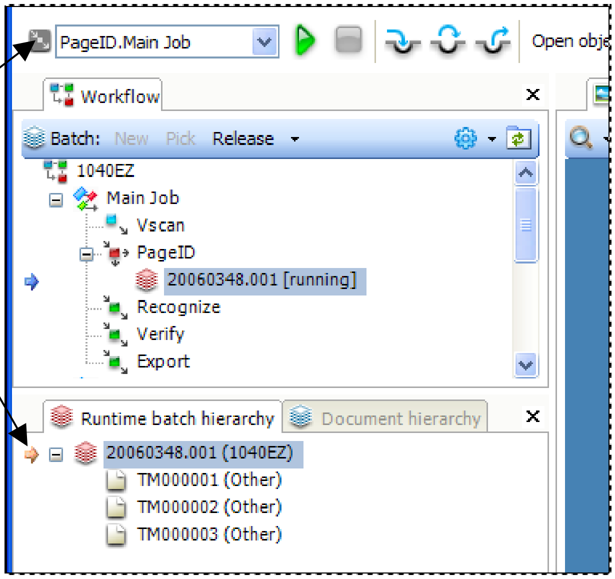
The task profile competed for the batch.

I would like to:

- advance the batch to the next step in the workflow. Advance
- release batch to public as "task finished". Task finished
- put batch on hold. I will come back to it. Put on hold
- set the batch back to pending for this task. Back to pending
- abort the batch. It was no good. Abort batch
- cancel the batch as if it never existed. Cancel batch
- keep the batch running. I'm still working on it. Keep running

To Run Task Profiles (continued)

Step	Action
4.	When the *Release Batch?* dialog appears, select a processing option. Unless there are problems, you will probably select *Advance* after you have finished with the Vscan Task Profile. This automatically moves the batch to the Page ID Task Profile.
5.	The **Select Task Profile** drop-down list shows the Task ID of the Task Profile that is line to process the batch. The *Runtime Batch Hierarchy* tab provides details of the batch and its *Other* pages. (Typically, the Page Type of all pages in a batch is *Other* until a recognition task has a chance to identify individual pages.)



Ready for PageID

The batch and its *Other* pages

Step	Action
6.	Highlight a Page ID to be sure its image appears in the *Image* tab.

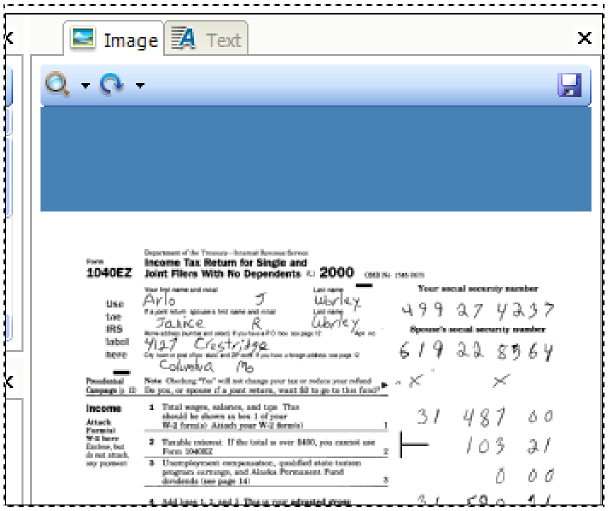
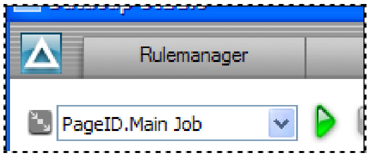


Image
Text
Form 1040EZ
Income Tax Return for Single and Joint Filers With No Dependents
2000

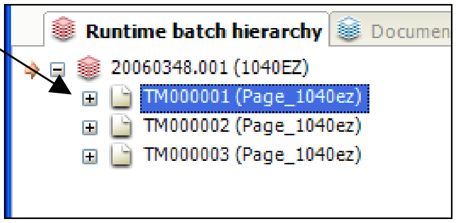
To Run Task Profiles (continued)

Step **Action**

7. Press the **Run** icon to initiate processing by Rulesets of the PageID Task Profile (in this example.)

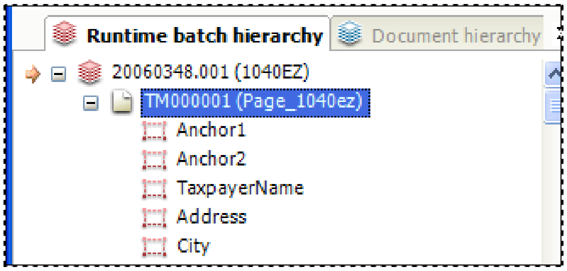


8. If the task is successful, the *Runtime Batch Hierarchy* tab will change the Page Types of the *Other* pages to *Page_1040ez* pages.

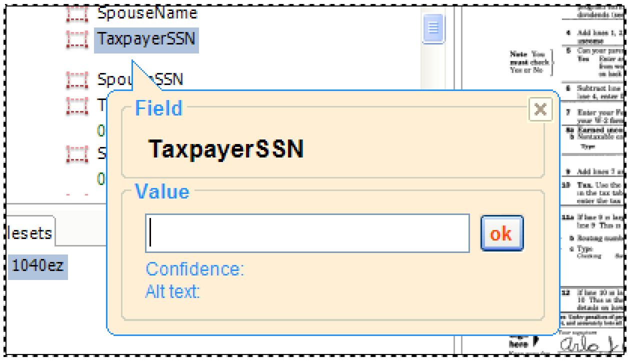


Open for a list of fields.

9. Now you can open a page for a first look at its fields.



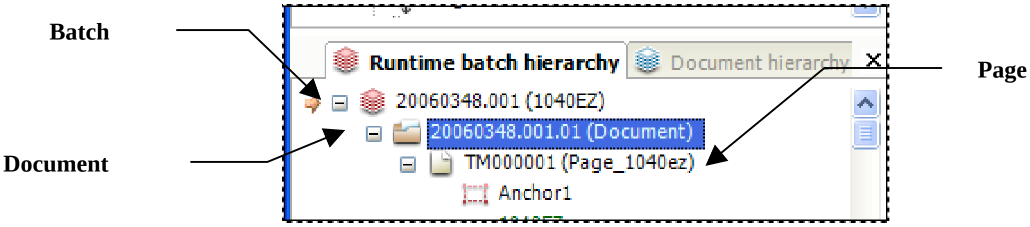
10. For another "first look", you can double-click on a Field ID to access its unique but still empty ***Data Entry*** panel.



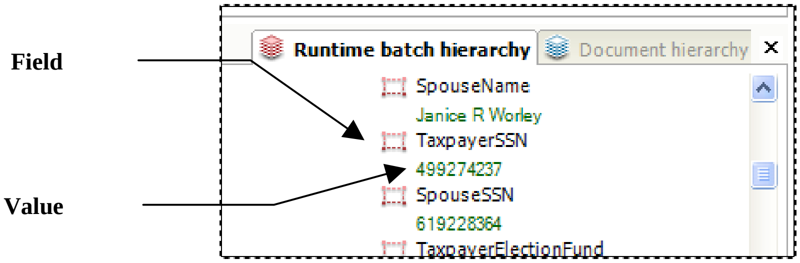
11. Press the **Run** icon to initiate processing by the numerous Rulesets of the Recognize Task Profile (in this example).

To Run Task Profiles (continued)

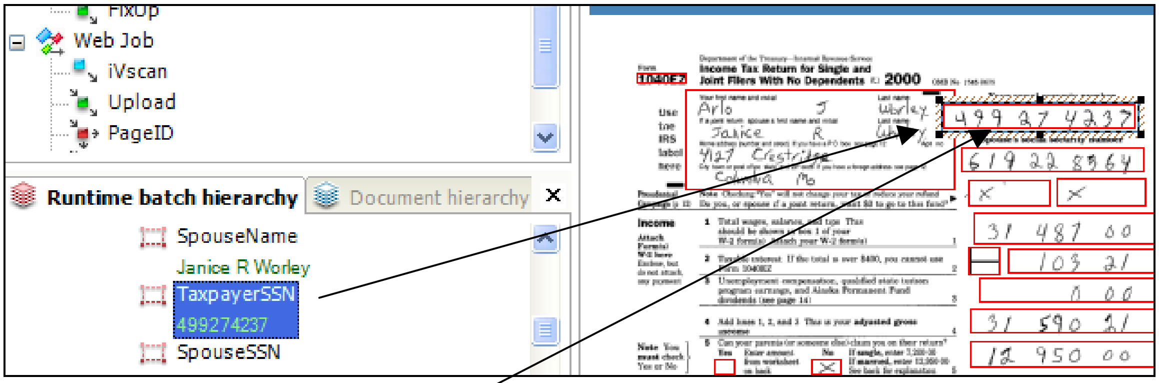
Step	Action
12.	Confirm, first, that the task has successfully re-organized the batch into a series of documents and ***source*** pages.



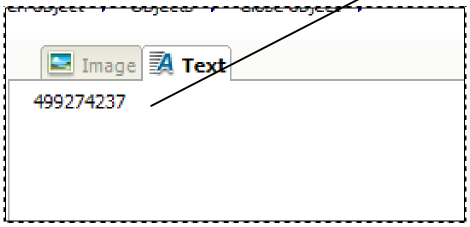
13. Check that recognized values appear with specific fields.



14. Select a field with a value to confirm that the field is highlighted in the page's image.

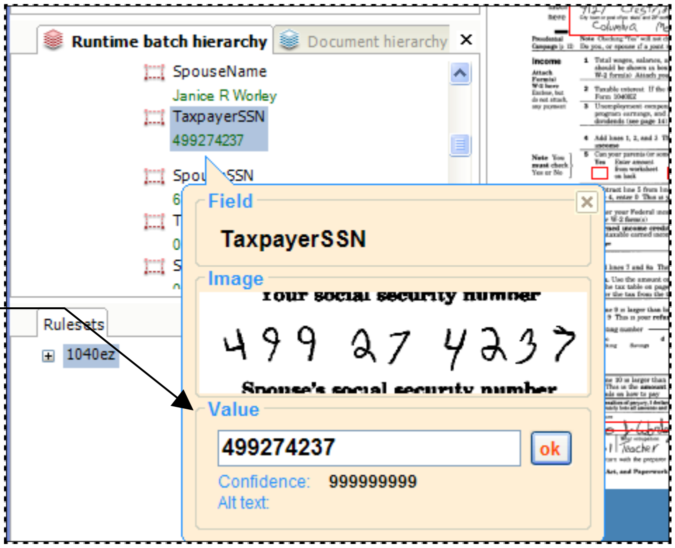


15. Check, too, that a field's recognized value now appears in the *Text* tab.



To Run Task Profiles (continued)

Step	Action
16.	Double-click on a field to open its ***Data Entry*** panel to be sure it includes the recognized value.



SpouseName
Janice R Worley
TaxpayerSSN
499274237

Field: TaxpayerSSN

Image

Value: 499274237

Confidence: 999999999

Alt text

Value field

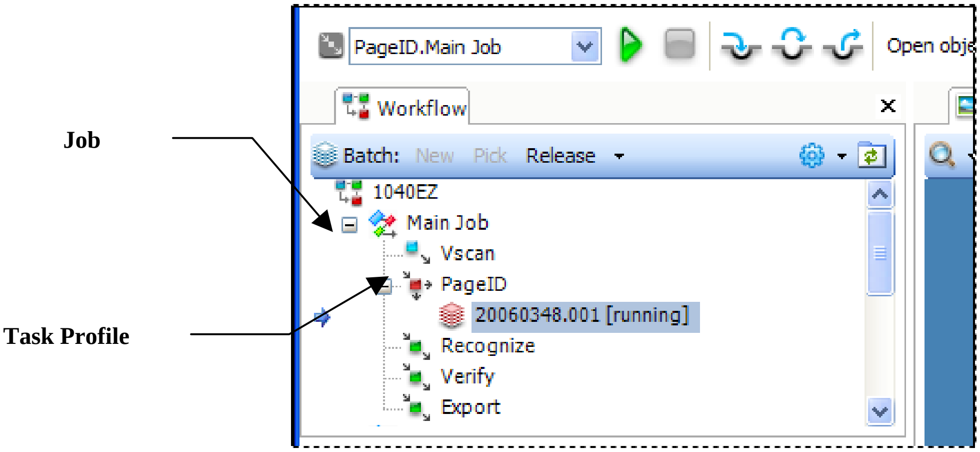
17.	If necessary, use the panel’s interactive **Value** field to modify the recognized value and press the OK button.
18.	Advance the batch to the Verify Task Profile.
19.	Press the **Run** icon to initiate processing by Rulesets of the Verify Task Profile. If the task runs into validation problems, you can use a field’s ***Data Entry*** panel to revise and update its value.
20.	Advance the batch to the Export Task Profile.
21.	Press the **Run** icon to initiate processing by Rulesets of the Export Task Profile.
22.	Afterwards, select the Task Finished button in the ***Release Batch?*** dialog to remove the batch from the test of Task Profiles.

8.2 Debugging your Application

The tabs of the **Test Page** combine with specific icons to contribute to the scope and success of your debugging procedures.

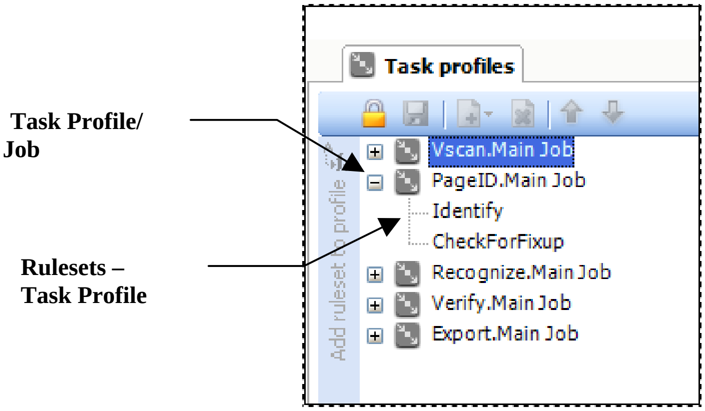
8.2.1 Setup Tabs

The *Workflow* tab shows how Task Profiles are associated with *Taskmaster* jobs, and the order in which they will create and process a batch and its contents:



DStudio Test Page - *Workflow tab*

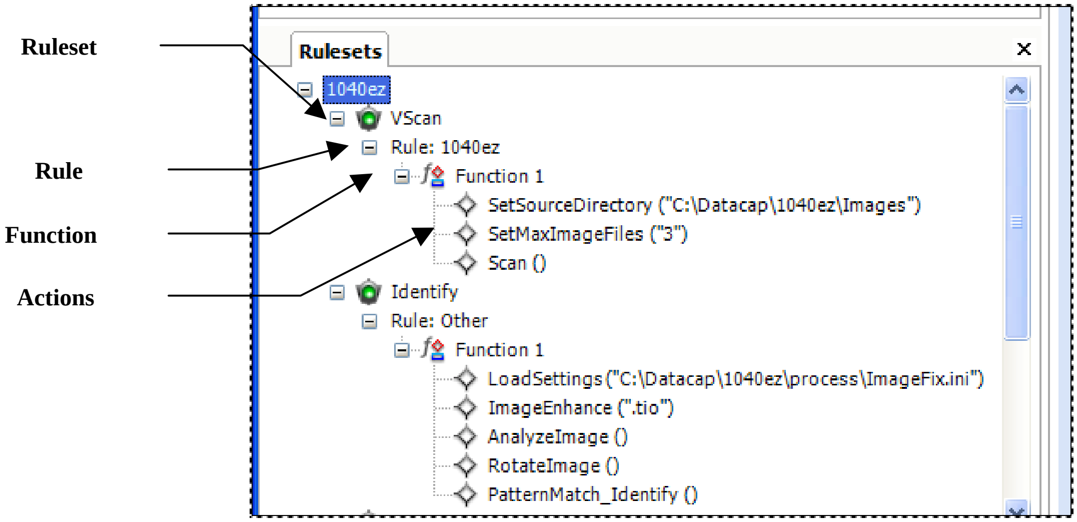
The *Task Profiles* tab identifies the Rulesets that are part of each Task Profile:



DStudio Test Page – *Task Profiles tab*

The *Rulesets* tab (shown on the next page) breaks down individual Rulesets into hierarchies of:

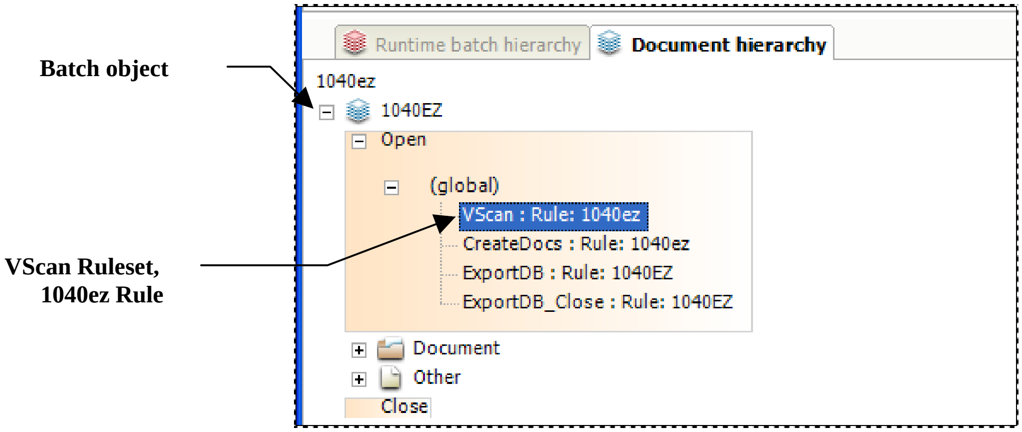
- Rules that belong to a Ruleset
- A rule's function(s)
- Actions that make up a function.



```
1040ez
  VScan
    Rule: 1040ez
      Function 1
        SetSourceDirectory ("C:\Datacap\1040ez\Images")
        SetMaxImageFiles ("3")
        Scan ()
  Identify
    Rule: Other
      Function 1
        LoadSettings ("C:\Datacap\1040ez\process\ImageFix.ini")
        ImageEnhance (".tio")
        AnalyzeImage ()
        RotateImage ()
        PatternMatch_Identify ()
```

Test Page – *Rulesets Tab*
VScan and Identify Ruleset/Rule Combinations

The *Document Hierarchy* tab lists the Ruleset/Rule combinations assigned to specific objects. As a result, the **Test Page** can formally evaluate these connections.



```
1040ez
  1040EZ
    Open
      (global)
        VScan : Rule: 1040ez
        CreateDocs : Rule: 1040ez
        ExportDB : Rule: 1040EZ
        ExportDB_Close : Rule: 1040EZ
    Document
    Other
    Close
```

Test Page – Document Hierarchy tab

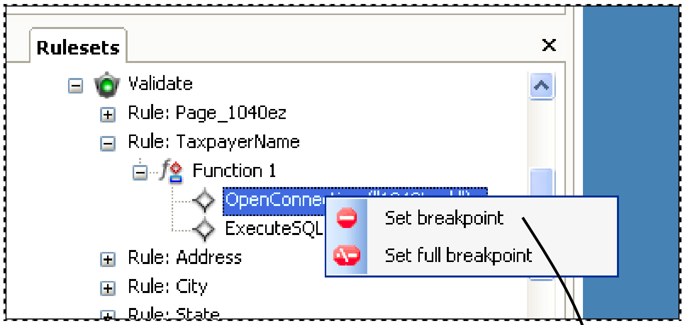
8.2.2 Diagnostic Tabs

The **Test Page** uses three tabs (and their tools) to isolate and report on problems.

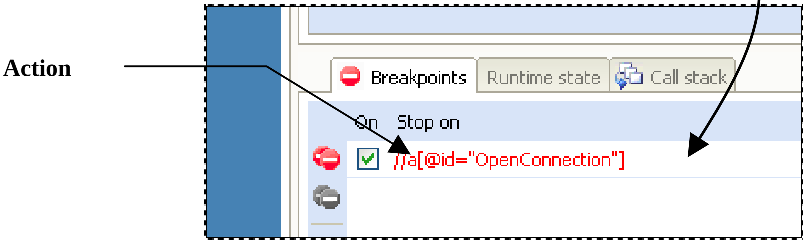
8.2.2.1 Breakpoints tab

The *Breakpoints* tab lists the breakpoints you set in the *Rulesets* tab.

A **breakpoint** intentionally halts a task's processing activity at a specific Ruleset, at a member of the Ruleset, or when a Task Profile is processing a specific page or field. A breakpoint is set in the *Rulesets* or *Task Profiles* tab, and managed in the *Breakpoints* tab.



Breakpoint options: OpenConnection Action

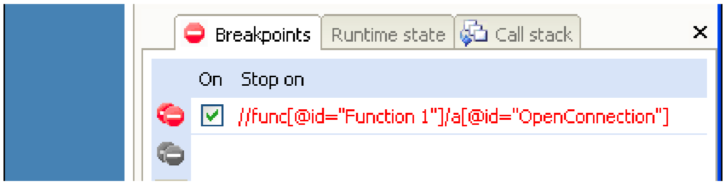


☑ //a[@id="OpenConnection"]

Action Breakpoint

In the example above, the syntax indicates a "single" breakpoint for the **OpenConnection** action.

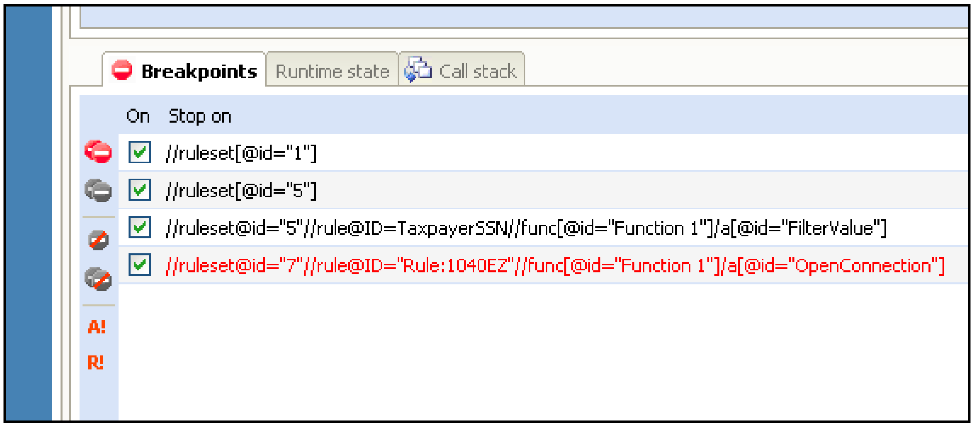
> ***Very important!*** A single breakpoint isolates and responds to this action no matter where – or how often – the action appears in the application's Ruleset/Rule combinations.
>
> The "full breakpoint" is specific: it designates an action or function within the context of the parent Ruleset/Rule combination:



☑ //func[@id="Function 1"]/a[@id="OpenConnection"]

Full Breakpoint

The *Breakpoints* tab has these features:



On	Stop on
☑	//ruleset[@id="1"]
☑	//ruleset[@id="5"]
☑	//ruleset@id="5"//rule@ID=TaxpayerSSN//func[@id="Function 1"]/a[@id="FilterValue"]
☑	//ruleset@id="7"//rule@ID="Rule:1040EZ"//func[@id="Function 1"]/a[@id="OpenConnection"]

Test Page – Breakpoints Tab

A check in a check box enables a breakpoint: removing the check disables it.

Icons on the on the tab's left edge (from top to bottom) include:

⊖ (Red) Enables all unchecked breakpoints

⊖ (Gray) Disables all checked breakpoints.

⊕ (Gray) Deletes checked breakpoints

⊕ (Gray) Deletes all breakpoints – checked and unchecked

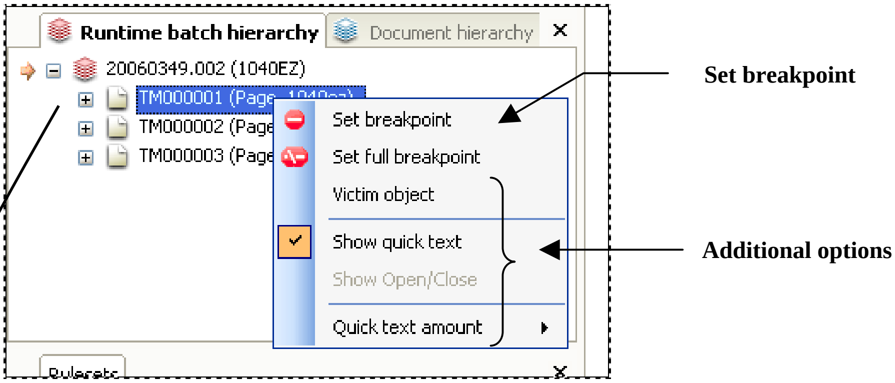
A! (Red) Directs the test to stop whenever it encounters a failed action, and clearly identifies the action.

R! (Red) Directs the test to stop whenever it encounters a failed rule, and clearly identifies the rule.

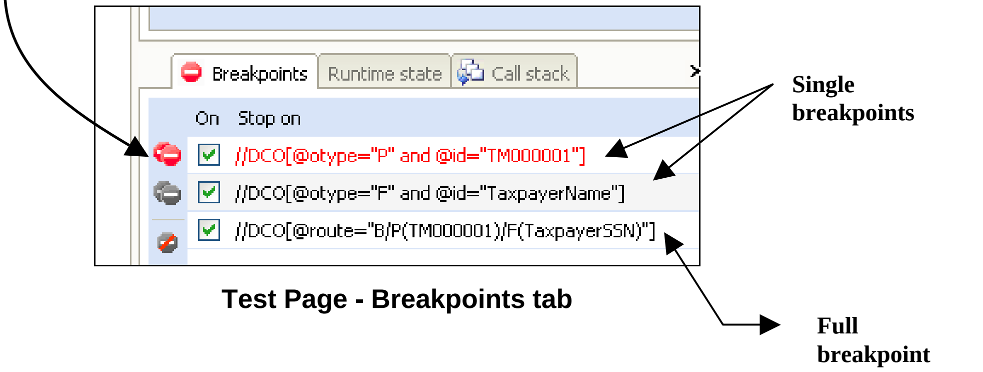
☛ **Other breakpoints.** After you initially test a Task Profile such as the *1040EZ* application's Page ID or Recognize task, you can set breakpoints for recognized components of the Document Hierarchy that have been found on a processed page.

On the next page there are two single breakpoints: one designates a Page ID (TM000001), and other a Field ID (TaxpayerName). These can be very helpful if you are trying to determine if a field's value on a particular page is accurate.

This *Breakpoints* tab also shows a full breakpoint that designates the *Taxpayer SSN* field when it is part of a *TM000001* page.



Set Breakpoint – Page TM000001



On	Stop on
☑	//DCO[@otype="P" and @id="TM000001"]
☑	//DCO[@otype="F" and @id="TaxpayerName"]
☑	//DCO[@route="B/P(TM000001)/F(TaxpayerSSN)"]

Test Page - Breakpoints tab

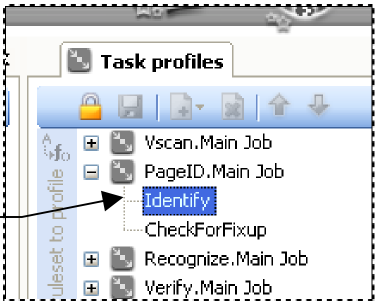
8.2.2.2 Runtime State Tab

The *Runtime State* tab provides details of successes and failures as Ruleset/Rule combinations of a Task Profile process a batch and its contents.

✓ To work with this tab effectively, you should be thoroughly familiar with the application’s basic Rule Maps (Chapter 6). You need to know, for example, the ID’s of the Rulesets.

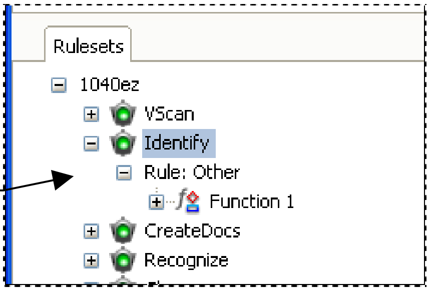
The illustration at the top of the next page indicates that the Main Job’s Page ID Task Profile starts out by applying rules of the **Identify** Ruleset. But it does not give you the Ruleset ID. However, a look at the *Rulesets* tab shows that this Ruleset is second in line, with an ID of “2”.

The third illustration in this sequence shows a breakpoint at this Ruleset. As a result, `//Ruleset[@id=2”]` designates this Ruleset. When the checkbox next to the Ruleset ID is selected, a Task Profile with the Ruleset will stop when it begins to apply the Ruleset’s rules. On the next page, an arrow points to the target Ruleset; in addition, the activated breakpoint takes on a deep crimson color.



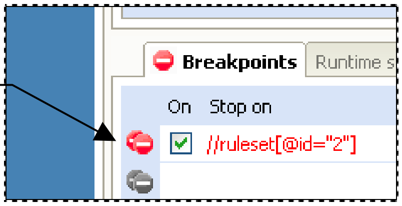
Identify Ruleset

Task Profiles tab



Ruleset ID=2

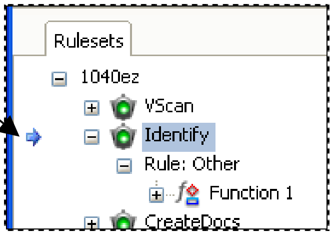
Rulesets Tab



☑ //ruleset[@id="2"]

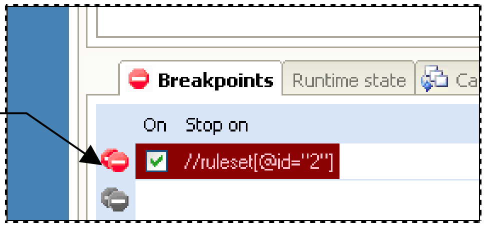
Set breakpoint

Breakpoints tab
Ruleset 2 (Identify)



Activated breakpoint

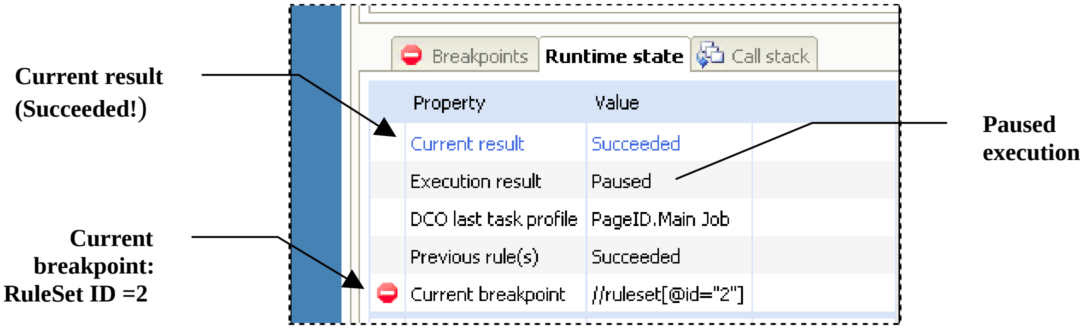
Activated Breakpoint



☑ //ruleset[@id="2"]

Activated breakpoint

Activated Breakpoint



Property	Value
Current result	Succeeded
Execution result	Paused
DCO last task profile	PageID.Main Job
Previous rule(s)	Succeeded
Current breakpoint	//ruleset[@id="2"]

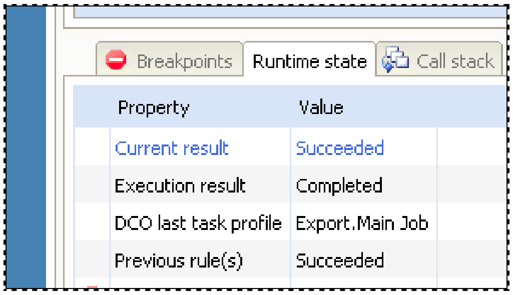
Current result (Succeeded!)

Current breakpoint: RuleSet ID =2

Paused execution

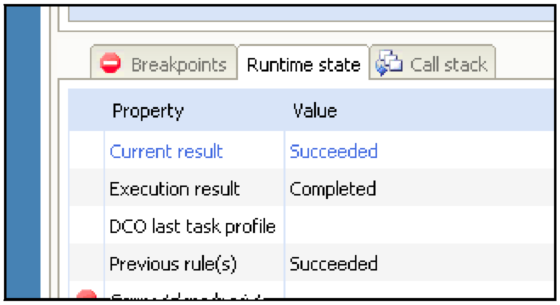
Runtime State tab

After you remove the breakpoint – or step out of it – the *Runtime State* tab will report on the progress of additional tasks until the batch has been completely processed by the job's tasks.



Property	Value
Current result	Succeeded
Execution result	Completed
DCO last task profile	Export.Main Job
Previous rule(s)	Succeeded

Runtime State tab –Export complete (current batch)



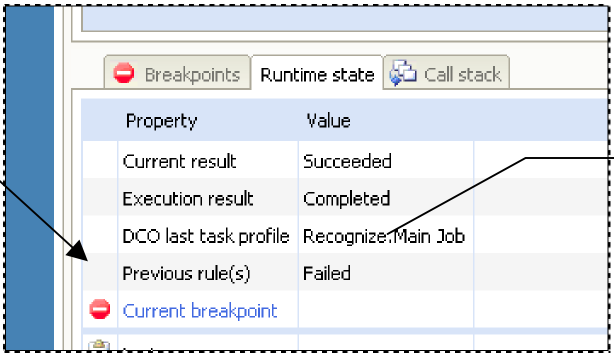
Property	Value
Current result	Succeeded
Execution result	Completed
DCO last task profile	
Previous rule(s)	Succeeded

Runtime State tab – Job complete (current batch)

✓ The *Runtime State* tab also provides details about problems uncovered by a test. The following page traces these details.

The *1040EZ* application's VScan and PageID Task Profiles typically process a batch without difficulty.

If you look carefully, however, the *Runtime State* tab almost always highlights a problem in this sample application– even if you have not called for a breakpoint. Here is an example:



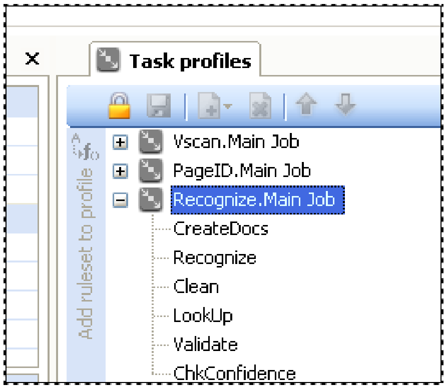
Property	Value
Current result	Succeeded
Execution result	Completed
DCO last task profile	Recognize.Main Job
Previous rule(s)	Failed
Current breakpoint	

Failed rule!

Recognize Task Profile

Runtime State tab

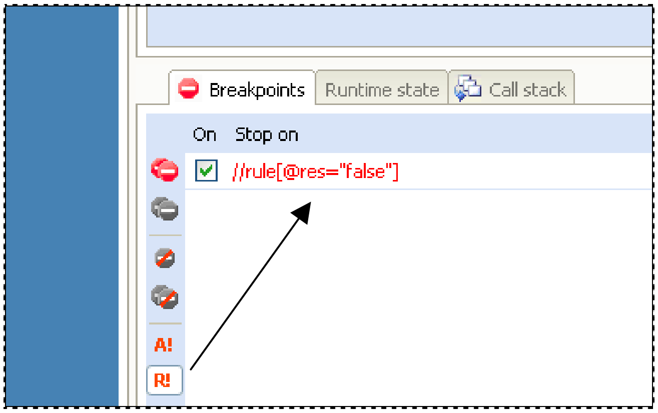
✓ Because the *Runtime State* tab specifies a failed rule in the Recognize Task Profile, the first step is to review the Rulesets applied by the Recognize Task Profile (Page 4):



- Vscan.Main Job
- PageID.Main Job
- Recognize.Main Job
  - CreateDocs
  - Recognize
  - Clean
  - LookUp
  - Validate
  - ChkConfidence

Rulesets of the Recognize Task Profile

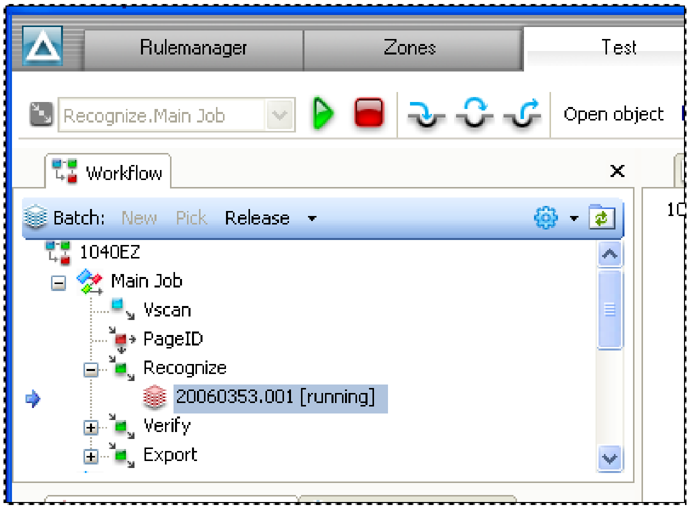
✓ At this point, you can take an aggressive but simple step to identify a problem rule within any of the Rulesets. Just select the *Stop on a Failed Rule* breakpoint option in the Breakpoints tab (illustrated on the next page,)



On	Stop on
☑	//rule[@res="false"]

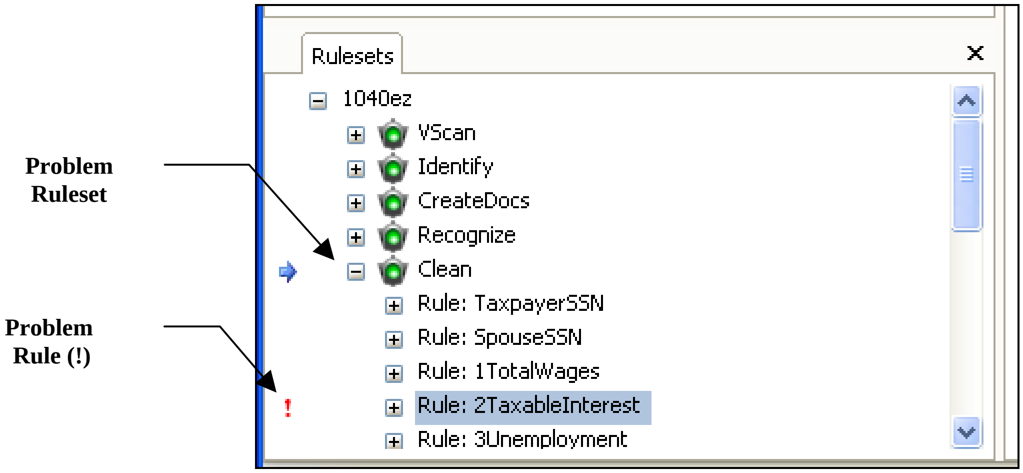
Test Page - ***Breakpoints Tab***

As a result, when the Task Profiles run, DStudio's test procedures will alert you to problem Rulesets and rules.



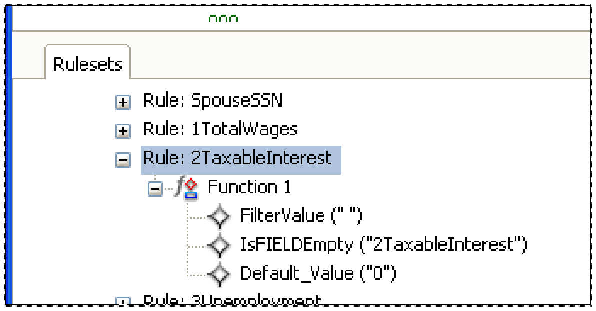
1040EZ
Main Job
Vscan
PageID
Recognize
20060353.001 [running]
Verify
Export

Recognize Task – Batch 20060253.001



1040ez
VScan
Identify
CreateDocs
Recognize
Clean
Rule: TaxpayerSSN
Rule: SpouseSSN
Rule: 1TotalWages
Rule: 2TaxableInterest
Rule: 3Unemployment

Rulesets tab – *Problem Rule*

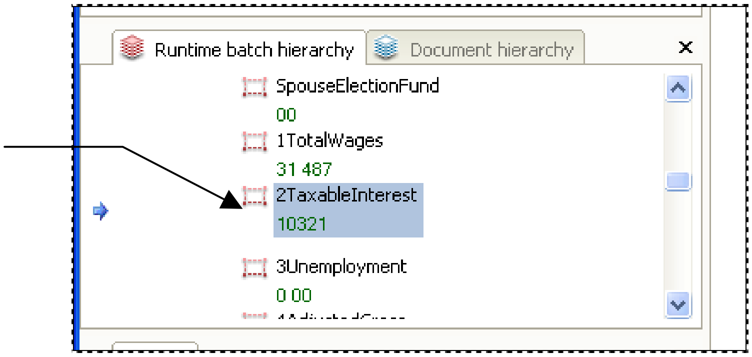


Rule: SpouseSSN
Rule: 1TotalWages
Rule: 2TaxableInterest
Function 1
FilterValue (" ")
IsFIELDEmpty ("2TaxableInterest")
Default_Value ("0")

Rulesets tab – *Problem Rule and Actions*

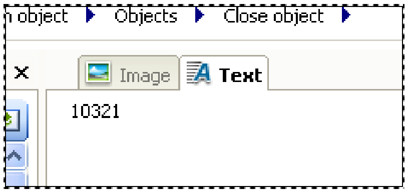
Below, the diagnostics go a step further as they point out the field of the current ***source*** page that ends up with a questionable value because of the incorrect rule. The *Test* tab, too, displays this value.

Correct value?



SpouseElectionFund
00
1TotalWages
31 487
2TaxableInterest
10321
3Unemployment
0 00

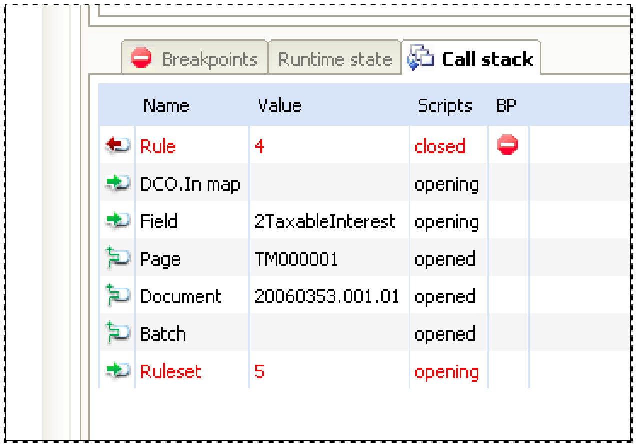
Runtime Batch Hierarchy – *Problem Object*



10321

Text tab – Problem Value

✓ When you run into trouble (intended or not) be sure to spend time with the *Call Stack* tab - a helpful summary of test information.



Name	Value	Scripts	BP
Rule	4	closed	⛔
DCO.In map		opening	
Field	2TaxableInterest	opening	
Page	TM000001	opened	
Document	20060353.001.01	opened	
Batch		opened	
Ruleset	5	opening	

Call Stack tab

8.3 Test Page Reference

The graphic on Page 3 displays the **Test Page's** three sections:

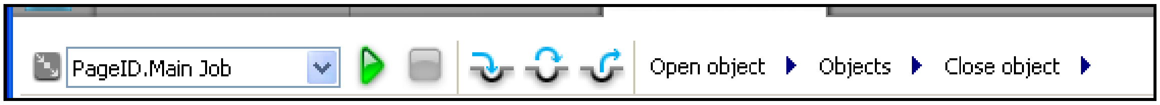
Testing tabs occupy the column on the left-hand side. Its tools and settings combine with the toolbar's icon to initiate and manage testing contents and flow.

The **Image** tab displays the image of a current ***source*** page, as well as the zoned fields of a matched fingerprint. The accompanying **Text** tab presents the recognized value of a selected field, or of a field subject to a breakpoint.

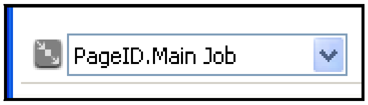
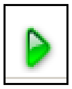
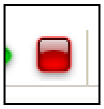
Information, on the right, provides ongoing data about the application and its test components, and about testing results.

8.3.1 Test Page Reference

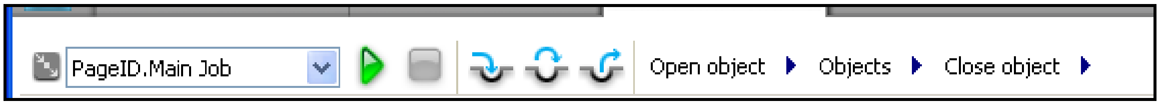
The toolbar's settings and icons operate in conjunction with the procedures in the **Testing** section.



The table below describes the toolbar's components.

Icon	Description
	A drop-down list of Task Profiles (Page 4) – typically. ***Very important!*** As you move through the Task Profiles of the Main job, be sure to select the current task from this list – even if the automatic advance mechanism is not active.
 **Debug Task Profile**	Launches the debugging of the Task Profile you have designated in the **Task Profile** field, and a test of its Ruleset(s) and rules.
 **Pause**	Intervenes to temporarily end the current "debugging session." This icon is only active if the current task has stopped because it has encountered a **breakpoint** in a rule. To resume the test, click on this icon and on the Task Processing and Debugging icon (above).

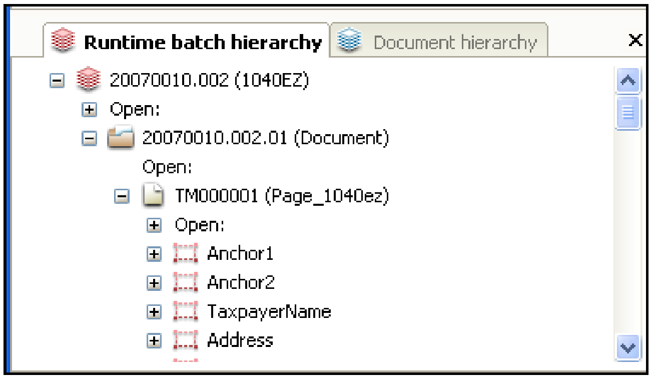
Step Into	"Steps into" lower levels – sequentially - of an element that you have set up as a breakpoint and executes that line of code as well as its follow-up code. (Stepping into a function, for example, debugs the functions and its actions.)
Step Over	"Steps over" the current element and onto the next element at the same level sequentially. As a result, if a breakpoint stops at a rule's first function, clicking on this icon debugs the first function of the next rule.
Step Out	Debugs all remaining lines of the current procedure.
Open object ▶ **Debug Open Rulesets**	Debugs the *Open* Ruleset/Rule combinations assigned to the current Task Profile ***and*** bound to a specific object of the Document Hierarchy that you have selected in the *Runtime Batch Hierarchy* tab.
Objects ▶ **Debug Children Objects**	Debugs the *Open* Ruleset/Rule combinations assigned to the current Task Profile ***and*** bound to child objects of the object you have selected in the *Runtime Batch Hierarchy* tab.
Close object ▶ **Debug Closing Rulesets**	Debugs the *Close* Ruleset/Rule combinations assigned to the current Task Profile ***and*** bound to a specific object of the Document Hierarchy that you have selected in the *Runtime Batch Hierarchy* tab.



8.3.2 Runtime Batch Hierarchy Tab

The *Runtime Batch Hierarchy* tab is Datacap Studio's central debugging environment because:

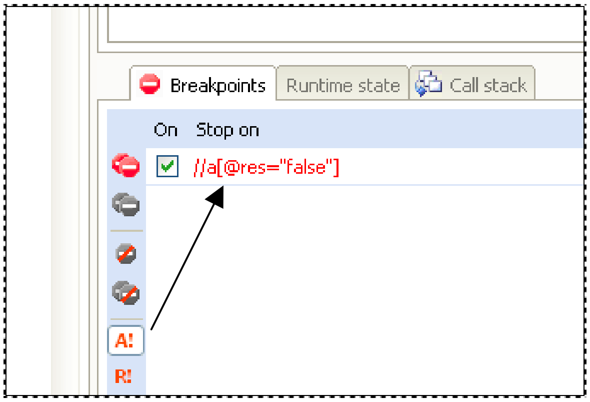
- The tab and its alternative procedures are easy to set up and use.
- It has productive links to other tabs.
- The processing information it provides is clear.
- Its diagnostic data is just as friendly.



```
⊟ 20070010.002 (1040EZ)
  ⊞ Open:
  ⊟ 20070010.002.01 (Document)
      Open:
    ⊟ TM000001 (Page_1040ez)
      ⊞ Open:
      ⊞ Anchor1
      ⊞ Anchor2
      ⊞ TaxpayerName
      ⊞ Address
```

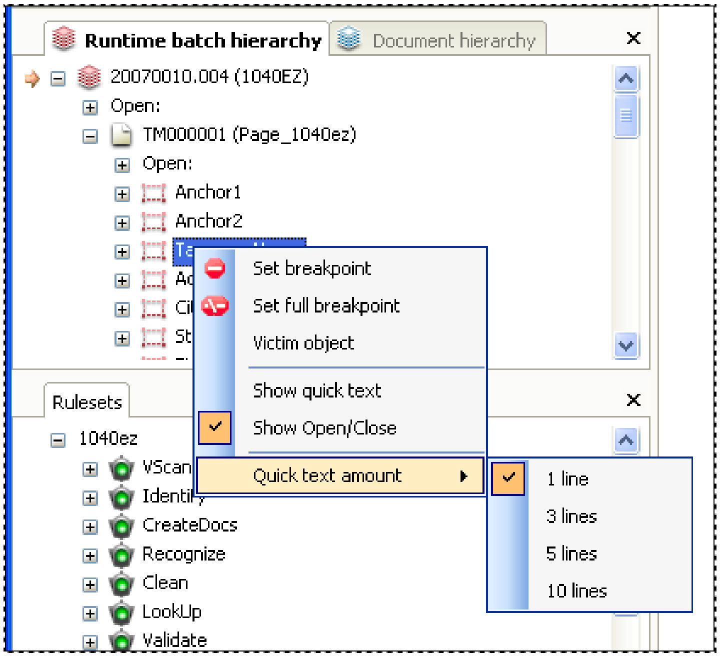
Runtime Batch Hierarchy tab

For a quick, accurate look at the features of the *Runtime Batch Hierarchy* tab, take these steps:

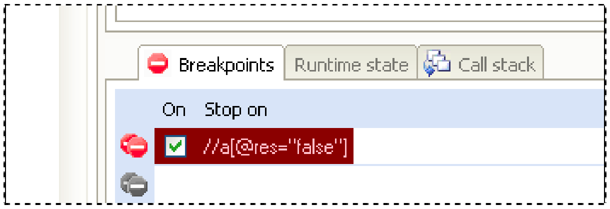
Step	Action
1.	Open the *1040EZ* application's Datacap Studio and turn to the **Test Page.**
2.	Set up a breakpoint that *stops on a failed action*.  Breakpoints Runtime state Call stack On Stop on A! R! ☑ //a[@res="false"]
3.	Follow the steps on Page 6 to create and scan a new batch.
4.	Process the batch with the Page ID task.

How to Explore the Runtime Batch Hierarchy Tab (continued)

Step	Action
5.	Before you run the Recognize task, right-click on a Field ID in one of the pages.



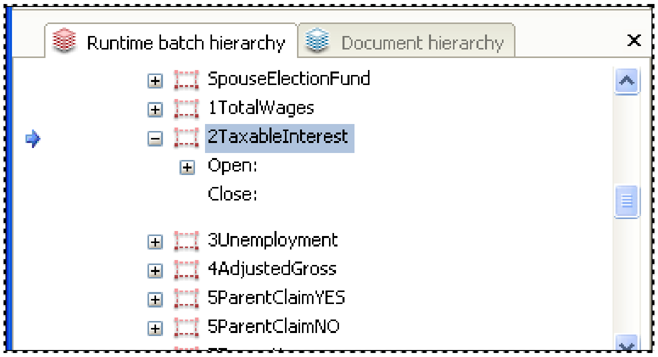
Step	Action
6.	Select **Show Quick Text** if you want DStudio to display ***every*** field's recognized value just below the Field ID. ***Very important!*** Whether or not you select this option, a field's recognized value will automatically appear in the *Text* tab during processing.
7.	If you select **Show Quick Text,** use the **Quick Text Amount** selection to specify how many lines of text can appear beneath a Field ID.
8.	Select **Show Open/Close** to direct Datacap Studio to list all *Open* and *Close* Rulesets that have been assigned to all objects (Chapter 5.) This can be helpful if you intend to use the **Open object** icon. You cannot select both **Show Quick Text** and **Show Open/Close** .
9.	Run the Recognize task!
10.	Note, first, the modified color of the breakpoint's specification in the *Breakpoints* tab (illustrated on the next page.)



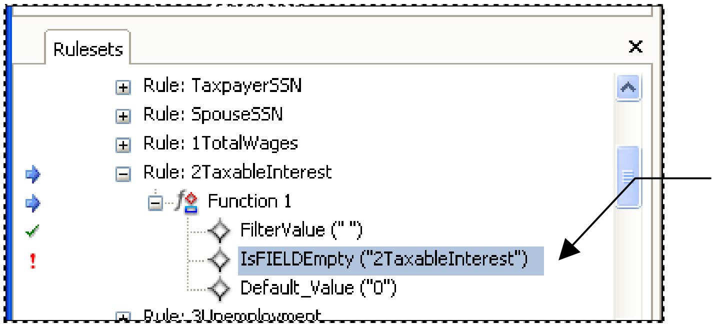
How to Explore the Runtime Batch Hierarchy Tab (continued)

Step **Action**

23. Check the *Runtime Batch Hierarchy* tab for the identity of the problem object – in this case, the *2TaxableInterest* field.



24. Look, now, at the *Rulesets* tab.

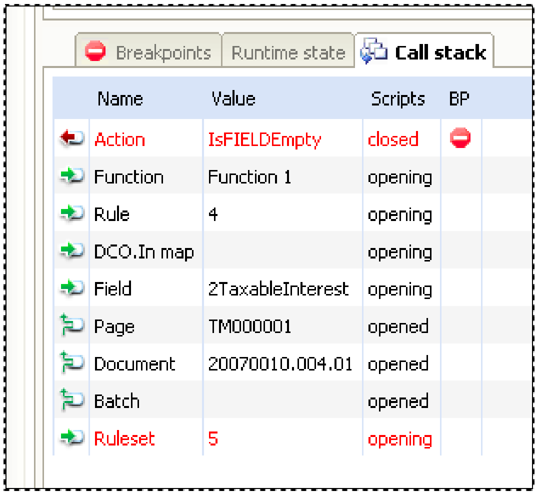


Problem action!

25. Check, as well, the supporting information in the important but easily overlooked *Call Stack* tab.

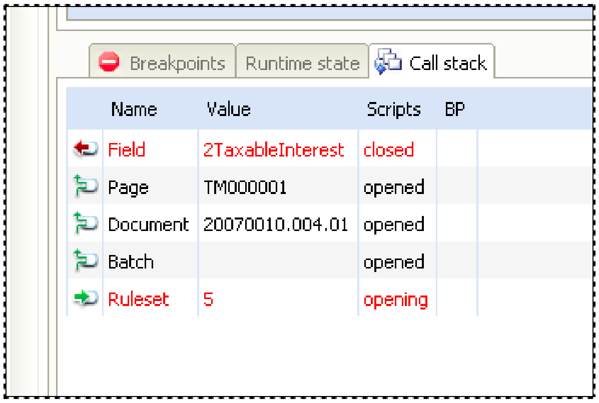
26. After you fix or jump over the problem, the *Call Stack* tab's information will resemble the second example on the next page.

Alert! The diagnostic settings for one batch are not removed when you process additional batches.



Name	Value	Scripts	BP
Action	IsFIELDEmpty	closed	⛔
Function	Function 1	opening	
Rule	4	opening	
DCO.In map		opening	
Field	2TaxableInterest	opening	
Page	TM000001	opened	
Document	20070010.004.01	opened	
Batch		opened	
Ruleset	5	opening	

Call Stack tab – Ruleset Hierarchy



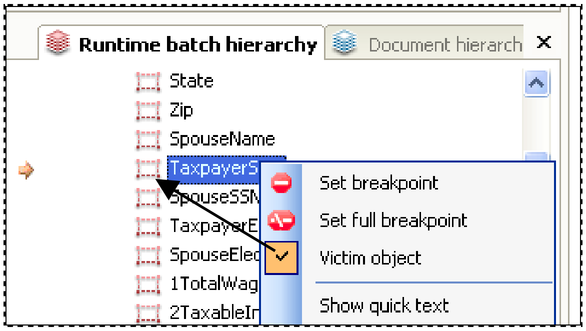
Name	Value	Scripts	BP
Field	2TaxableInterest	closed	
Page	TM000001	opened	
Document	20070010.004.01	opened	
Batch		opened	
Ruleset	5	opening	

Call Stack tab – Batch Hierarchy

8.3.2.1 Victim Objects

To set up and benefit from the tab's **Victim** criteria:

- Right-click on any processed object in the *Runtime Batch Hierarchy* tab and select *Show Open/Close*.
- Right-click on a specific processed object in the *Runtime Batch Hierarchy* tab…this becomes the victim object.



When a task such as Recognize runs, it will process ***only*** the victim. This gives you a chance to explore the object’s Ruleset/Rule combinations, and any problems they may encounter.

✓ ***Important!*** Be sure to release the object’s victim status before you continue to process the batch.