

---

## Rulerunner Actions

This document describes the **actions** of each Actions file (.rra/.rrx). After the Table of Contents, it covers these topics.

I.	Introduction.....	4-12
II.	Rulemanager's Rulesets and Actions Libraries.....	4-13
III.	AutoDoc Actions.....	4-15
	1. BlankPagesIDBySize .....	4-16
	2. CalculateOffset.....	4-16
	3. CreateFingerprint.....	4-17
	4. DeleteFingerprint .....	4-17
	5. FindBlackFingerprint.....	4-18
	6. FindFingerprint.....	4-19
	7. FindTemplate.....	4-20
	8. MergeCCOs_ByType.....	4-21
	9. SetFilter_HostName.....	4-21
	10. SetFilter_PageType.....	4-22
	11. SetFingerprint.....	4-22
	12. SetFingerprintDir.....	4-23
	13. SetFingerprintWebServiceURL.....	4-23
	14. SetMaxOffset .....	4-24
	15. SetProblemValue.....	4-25
	16. SetSearchArea.....	4-26
	17. SetTemplateDir .....	4-27
IV.	Barcode_X Actions.....	4-28
	1. GetBarCode .....	4-28
	2. MatchBarCode.....	4-29
	3. ReadBarCode .....	4-29
V.	cco2cco Actions .....	4-30
	1. cco2cco.....	4-30
	2. NormalizeCCO.....	4-32
VI.	CMS Actions.....	4-34
	1. DetectHCFA .....	4-34
	2. ValidQualifier.....	4-34
VII.	DataClip Action .....	4-35

- 1. DCI\_ClipField..... 4-35
- VIII. DCO Actions ..... 4-37
  - 1. ChkConfidence..... 4-37
  - 2. ChkDCOStatus ..... 4-38
  - 3. ChkDCOType..... 4-38
  - 4. ChkIntegrity ..... 4-39
  - 5. ChkLastDCOType ..... 4-39
  - 6. ClearAltText..... 4-40
  - 7. ClearDCO ..... 4-41
  - 8. CopyPD2DD..... 4-41
  - 9. CreateDocuments ..... 4-42
  - 10. CreateFields ..... 4-43
  - 11. DeleteFields..... 4-44
  - 12. PropagateToAltText..... 4-44
  - 13. SetDCOStatus ..... 4-45
  - 14. SetDCOType ..... 4-45
  - 15. SetDocumentType ..... 4-46
  - 16. SetFldConfidence ..... 4-47
  - 17. SetPageFingerprintID ..... 4-48
  - 18. SetPageStatus..... 4-49
  - 19. SetPageTemplateID ..... 4-50
  - 20. SetPageType ..... 4-51
- IX. Equalize Action..... 4-52
  - 1. EqualizeUnbalancedImage ..... 4-53
- X. Export Actions ..... 4-54
  - 1. BatchVariable\_ExportValue..... 4-54
  - 2. BlankFields..... 4-54
  - 3. BlankLines..... 4-54
  - 4. BPilot..... 4-55
  - 5. CloseExportFile..... 4-56
  - 6. DCOProperty ..... 4-56
  - 7. DocumentVariable\_ExportValue ..... 4-57
  - 8. ExportAllFields..... 4-57
  - 9. ExportFieldValue..... 4-58
  - 10. ExportMYValue..... 4-58
  - 11. ExportSmartParameter ..... 4-58
  - 12. ExportToBatchDir ..... 4-59
  - 13. Filler..... 4-59
  - 14. FixedLenLJ..... 4-60
  - 15. FixedLenRJ..... 4-60
  - 16. GetDATE ..... 4-61
  - 17. GetProfileString ..... 4-62
  - 18. GetTime..... 4-63
  - 19. Lineltem\_AddElement ..... 4-64

20. Lineltem_BlankFields .....	4-65
21. Lineltem_ClearElements .....	4-65
22. Lineltem_ExportElements .....	4-66
23. Lineltem_SmartParameter .....	4-67
24. NewLine .....	4-68
25. PageVariable_ExportValue .....	4-68
26. ResetFieldVariables .....	4-69
27. SetFilePathAsVariable .....	4-69
28. SetCSV .....	4-70
29. SetElementSeparator .....	4-70
30. SetExportPath .....	4-71
31. SetExtensionName .....	4-71
32. SetFileName .....	4-72
33. SetFilePathAsVariable .....	4-73
34. SetFill .....	4-73
35. SetFixedLength .....	4-73
36. SetIgnoreFieldStatus .....	4-73
37. SetJustified .....	4-74
38. SetOMR_Separator .....	4-74
39. SetSpaceFill .....	4-75
40. SetZeroFill .....	4-75
41. Text .....	4-75
42. Variable_ExportValue .....	4-76
43. Variable_IsValue .....	4-76
44. <i>Batch Pilot Properties (BPilot Action)</i> .....	4-77
45. <i>Document Hierarchy Properties</i> .....	4-78
<b>XI. ExportDB Actions .....</b>	<b>4-79</b>
1. AddRecord .....	4-80
2. CloseConnection .....	4-81
3. ExportBatchIDToColumn .....	4-82
4. ExportFieldToColumn .....	4-83
5. ExportNodeXMLToColumn changed .....	4-84
6. ExportPropertyToColumn .....	4-85
7. ExportSmartParamToColumn .....	4-86
8. ExportToColumn .....	4-87
9. OpenConnection .....	4-88
10. SetTableName .....	4-89
<b>XII. ExportXML Actions .....</b>	<b>4-90</b>
1. XML_CommitNode .....	4-90
2. XML_NewNode .....	4-91
3. xml_SaveFile .....	4-91
4. xml_SetAttributeValue .....	4-92
5. xml_SetExportPath .....	4-92
6. xml_SetFileName .....	4-93
7. xml_SetNodeValue .....	4-94

- XIII. Grayscale Action ..... 4-95
  - 1. ConvertGraytoBW ..... 4-95
- XIV. icr\_c Actions ..... 4-95
  - 1. RecognizeFieldICR\_C ..... 4-95
  - 2. RecognizeFieldVoteICR\_C ..... 4-96
  - 3. RecognizePageFields2CCO\_ICR\_C ..... 4-97
  - 4. RecognizePageFieldsICR\_C ..... 4-97
  - 5. RecognizePageICR\_C ..... 4-98
- XV. ImageFix Actions ..... 4-99
  - 1. ImageEnhance ..... 4-99
  - 2. LoadSettings ..... 4-100
  - 3. LoadSettings\_FingerprintID ..... 4-100
- XVI. Imprint Actions ..... 4-101
  - 1. ImPrint ..... 4-101
  - 2. SetAdjustedWidth ..... 4-101
  - 3. SetFontName ..... 4-102
  - 4. SetFontSize ..... 4-103
  - 5. SetOpaque ..... 4-103
- XVII. Intellocate Actions ..... 4-104
  - 1. iloc\_AdjustZones ..... 4-104
  - 2. iloc\_AssignPageType ..... 4-105
  - 3. iloc\_SetDetailZones ..... 4-105
  - 4. iloc\_SetZones ..... 4-106
  - 5. IsPageDataMissing ..... 4-106
- XVIII. IOverlay Actions ..... 4-107
  - 1. Overlay ..... 4-107
  - 2. SetBackgroundImage ..... 4-108
  - 3. SetDitheringBackground ..... 4-108
  - 4. SetHaloBackground ..... 4-109
- XIX. Locate Actions ..... 4-110
  - 1. How to Assemble a Keyword List ..... 4-110
  - 2. AddLeadingZeros/Leading Zeros ..... 4-113
  - 3. DefaultValue ..... 4-113
  - 4. FilterIt ..... 4-114
  - 5. FindDBList ..... 4-115
  - 6. FindDBList\_InZone ..... 4-115
  - 7. FindKeyList ..... 4-116
  - 8. FindKeyList\_InZone ..... 4-116
  - 9. FindLastKeyList ..... 4-117
  - 10. FindLastKeyList\_InZone ..... 4-117
  - 11. FindLastRegEx ..... 4-118
  - 12. FindLastRegEx\_InZone ..... 4-118
  - 13. FindLastRegExList ..... 4-119

14. FindLastRegExList_InZone.....	4-119
15. FindLastWord .....	4-120
16. FindLastWord_InZone .....	4-120
17. FindNextDBList .....	4-121
18. FindNextDBList_InZone .....	4-121
19. FindNextKeyList.....	4-122
20. FindNextKeyList_InZone.....	4-122
21. FindNextRegExList .....	4-123
22. FindNextRegExList_InZone .....	4-123
23. FindRegExList.....	4-124
24. FindRegExList_InZone .....	4-124
25. GoAboveWord.....	4-125
26. GoBelowWord .....	4-125
27. GoDownLine .....	4-126
28. GoFirstLine .....	4-126
29. GoFirstWord .....	4-127
30. GoLastLine .....	4-127
31. GoLastWord.....	4-127
32. GoLeftWord .....	4-128
33. GoRightWord.....	4-128
34. GoUpLine .....	4-129
35. GroupWords .....	4-129
36. GroupWordsLeft.....	4-130
37. GroupWordsRight .....	4-130
38. IsAlpha .....	4-131
39. IsCurrency .....	4-131
40. IsDateValue.....	4-132
41. IsNumber.....	4-132
42. IsValue.....	4-133
43. IsValue_RegEx .....	4-133
44. MaxLength .....	4-134
45. MergeWordLF .....	4-135
46. MergeWordRT.....	4-136
47. MinLength .....	4-137
48. RegExFind .....	4-137
49. RegExFindNext.....	4-137
50. RegExFind_InZone.....	4-138
51. RegExFindNext_InZone.....	4-139
52. ScanRT.....	4-140
53. SelectSnippet .....	4-141
54. SetRect.....	4-142
55. UpdateDCOField.....	4-142
56. UpdateField.....	4-143
57. ValueInField .....	4-144
58. ValueInField_Fuzzy .....	4-144

59. ValueInField_RegEx.....	4-145
60. WordFind .....	4-145
61. WordFind_InZone .....	4-146
62. WordFindNext .....	4-146
63. WordFindNext_InZone.....	4-147
64. WordFind_Offset.....	4-147
XX. LookUp Actions .....	4-148
1. ClearLookUpResults.....	4-148
2. CloseConnection .....	4-149
3. ExecuteSQL.....	4-150
4. LookupCurrentValue .....	4-151
5. LookupReturnValue.....	4-152
6. OpenConnection .....	4-153
7. PopulateWithResult .....	4-154
XXI. MC_Identify Actions .....	4-156
1. AutoField.....	4-156
2. FindFields .....	4-156
3. ReadDCOSetup .....	4-157
4. SetFormType .....	4-157
5. SetFSPathName .....	4-158
6. SetMaxTolerantDistance .....	4-158
7. SetWritePosFile.....	4-159
XXII. MC_Validation Actions .....	4-160
1. AddCenturyTo2DigitYear .....	4-160
2. CalculateResult .....	4-160
3. CalculateUB.....	4-161
4. CommonParseAddress .....	4-162
5. CommonValAddress.....	4-163
6. ConvertHyphen .....	4-164
7. InheritSnippets.....	4-164
8. MC_ReadZones .....	4-165
9. Parse31aPhSig.....	4-165
10. Parse58ainsnm.....	4-166
11. Parse58binsnm .....	4-166
12. Parse58cinsnm.....	4-166
13. Parse82name.....	4-167
14. Parse83aname.....	4-167
15. Parse83bname.....	4-167
16. ParseLastFirstIniNames .....	4-168
17. ParseUB_Eighties .....	4-168
18. PopulateFieldPositionFromSetup .....	4-169
19. PopulateFromField .....	4-169
20. StripTrailingAlpha.....	4-169
21. TransformLI .....	4-170

22. ValidateNPI .....	4-172
23. ValidateStateMil.....	4-172
24. ValProcedureCode .....	4-172
25. ValRequiredGroup .....	4-173
<b>XXIII. ocr_s Actions .....</b>	<b>4-174</b>
1. RecognizeDocToPDF.....	4-174
2. RecognizeFieldOCR_S .....	4-175
3. RecognizeFieldVoteOCR_S .....	4-176
4. RecognizeOM_OCR_S.....	4-176
5. RecognizePageFields2CCO_OCR_S.....	4-177
6. RecognizePageFieldsOCR_S.....	4-177
7. RecognizePageOCR_S .....	4-178
8. RecognizePageOCR_S_2TextFile .....	4-178
9. RecognizeToFile_OCR_S .....	4-179
10. RecognizeToPDF.....	4-181
11. Rotatelmage.....	4-182
12. SetFastTradeOffOCR_S.....	4-183
13. SetLegacyDecompositionOCR_S.....	4-183
<b>XXIV. PatternMatch Actions.....</b>	<b>4-184</b>
1. MatchPattern .....	4-184
2. Pat_RecogMatch_ID.....	4-185
3. Pat_RegisterZones.....	4-185
4. Pat_ReleasePageAnchors.....	4-186
5. PatternMatch_Fingerprint .....	4-187
6. PatternMatch_Identify.....	4-188
7. PatternMatch_PageType .....	4-189
8. SetMatchConfidence.....	4-190
<b>XXV. Recog_Shared Actions .....</b>	<b>4-191</b>
1. Analyzelmage .....	4-191
2. CCO_Normalization_OFF .....	4-192
3. CreateTextFile .....	4-193
4. IsBlankPage.....	4-194
5. RecognizeOMRThresh.....	4-195
6. RegisterPageFields.....	4-197
7. Releaselmage .....	4-197
8. RotateTIO .....	4-198
9. SetFingerprintRecogPriority .....	4-199
10. SetFullPageRecogArea.....	4-199
11. SetRecogFailureRetryDelay .....	4-200
12. SnapCCOtoDCO .....	4-200
13. SnapDCOtoCCO .....	4-201
14. UseOutOfProcessingRecog.....	4-201
<b>XXVI. RuleRunner Actions .....</b>	<b>4-202</b>
1. AbortOnError .....	4-202

2. CheckAllIntegrity .....	4-203
3. CheckDocCount .....	4-203
4. CheckPageCount .....	4-204
5. DebugModeOff .....	4-204
6. DebugModeOn .....	4-205
7. GoNextRule .....	4-206
8. PilotMessage_Clear .....	4-206
9. PilotMessage_Set .....	4-207
10. ProcessChildren .....	4-207
11. rr_AbortBatch.....	4-208
12. rr_Append .....	4-208
13. rr_Compare.....	4-209
14. rr_Compare_Not.....	4-210
15. rr_Copy .....	4-211
16. rr_Get .....	4-212
17. rr_Prepend .....	4-213
18. rr_Set.....	4-214
19. rr_WriteNode .....	4-215
20. SetExpectedDocs.....	4-215
21. SetExpectedPages .....	4-216
22. SetTaskStatus .....	4-216
23. SkipChildren.....	4-217
24. Status_Preserve_OFF.....	4-217
25. Status_Preserve_ON .....	4-218
26. Task_NumberOfSplits .....	4-219
27. Task_RaiseConditions .....	4-220
XXVII. Split Action.....	4-221
1. SplitBatch .....	4-221
XXVIII. TM524 Actions .....	4-223
1. TM524 .....	4-223
2. TM4DocIdFormat.....	4-223
XXIX. Validate Actions.....	4-224
1. AddDecimal .....	4-224
2. AddToString .....	4-224
3. AllLowerCase .....	4-225
4. AllowOnlyChars .....	4-225
5. AllUpperCase .....	4-226
6. AppendFromField .....	4-226
7. AppendToField .....	4-227
8. Calculate .....	4-228
9. CheckSubFields .....	4-229
10. ClearFieldValue .....	4-231
11. CompareFields .....	4-232
12. CopyToField .....	4-233



13. DateStamp .....	4-233
14. DefaultValue .....	4-234
15. DeleteALPHA .....	4-234
16. DeleteChars .....	4-235
17. DeleteChildType .....	4-235
18. DeleteMISC .....	4-236
19. DeleteNUMERIC.....	4-236
20. DeleteParent .....	4-236
21. DeletePUNCT .....	4-237
22. DeleteSYSTEM.....	4-237
23. EmptyFIELD .....	4-238
24. FailRuleSet.....	4-238
25. FieldContainsValue .....	4-238
26. FilterValue.....	4-239
27. GetJobID .....	4-239
28. GotoNextRule .....	4-240
29. HasChildOfType .....	4-240
30. Is_Alpha .....	4-241
31. Is_Currency .....	4-241
32. Is_Date .....	4-242
33. Is_DateSetFormat.....	4-242
34. Is_DateUpToToday.....	4-243
35. Is_DateWithinXDays .....	4-243
36. Is_DateWithinRange .....	4-244
37. Is_GreaterThan.....	4-244
38. Is_LessThan .....	4-245
39. Is_NonNumeric.....	4-246
40. Is_Numeric.....	4-246
41. Is_Value.....	4-247
42. IsDateGE .....	4-247
43. IsDateLE.....	4-248
44. IsFIELDEmpty.....	4-248
45. IsFIELDFilled .....	4-249
46. IsHiddenField.....	4-249
47. IsJobID .....	4-250
48. IsThisFieldEmpty .....	4-250
49. IsThisFieldFilled .....	4-251
50. IsVariableEmpty .....	4-251
51. IsVariableFilled.....	4-252
52. LeadingZeros.....	4-252
53. LeftTruncate .....	4-253
54. MakeCurrency .....	4-253
55. MaximumLength.....	4-254
56. MessageBox .....	4-255
57. MinimumLength .....	4-255

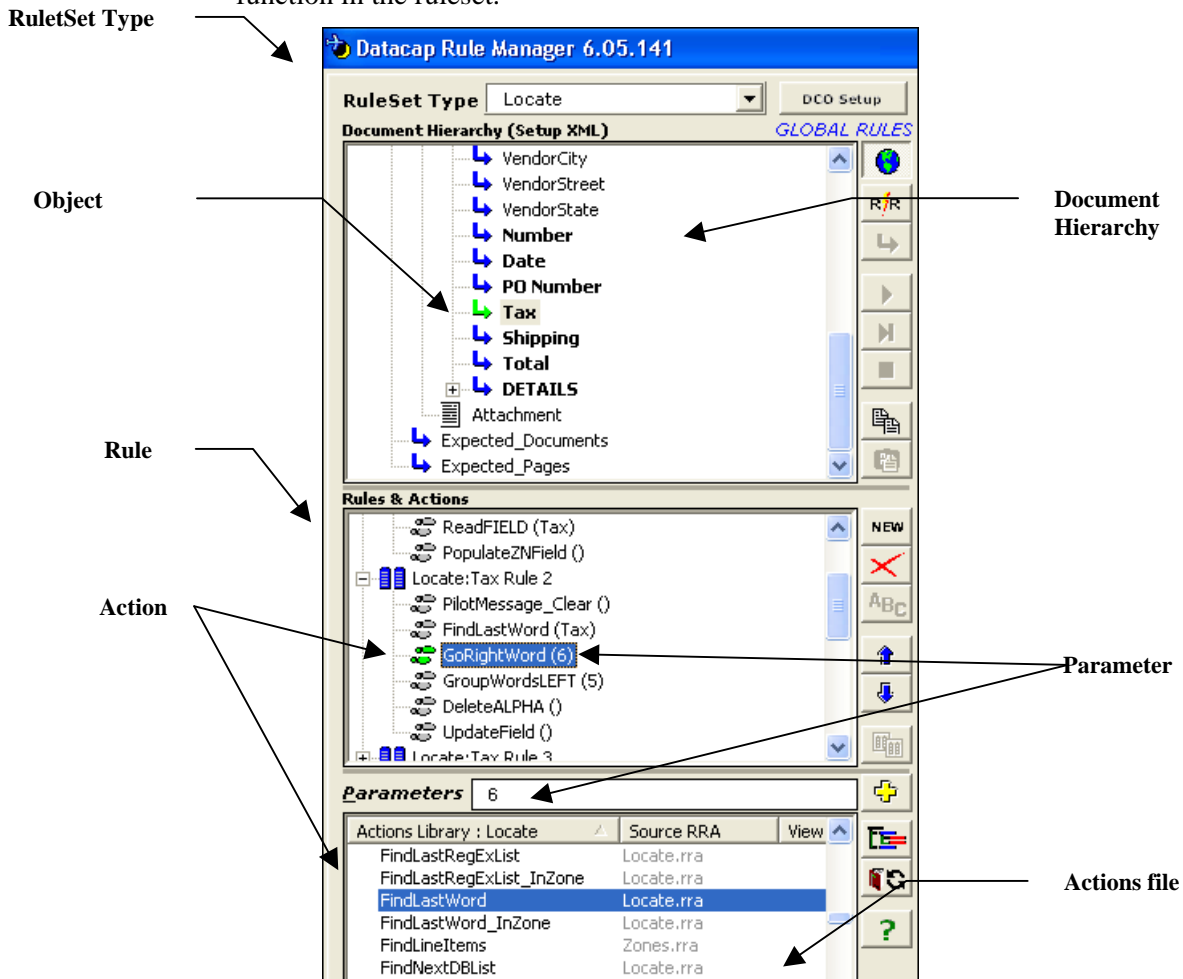
58. Non_OVERRIDEABLE .....	4-256
59. OMR_MaxChecked.....	4-256
60. OMR_MinChecked .....	4-257
61. OVERRIDEABLE .....	4-257
62. PadToLeft .....	4-258
63. PadToRight.....	4-258
64. ParseMultilineAddress .....	4-259
65. ParseName .....	4-259
66. PatternInField .....	4-260
67. ReadBatchVariable .....	4-260
68. ReadCurrentObjVariable .....	4-261
69. ReadDocumentVariable.....	4-261
70. ReadFIELD .....	4-261
71. ReadPageVariable.....	4-262
72. RemoveSpaces .....	4-262
73. ReplaceValue.....	4-263
74. ReplaceValue_AtPosition.....	4-264
75. RightTruncate.....	4-264
76. SaveAsBatchVariable .....	4-265
77. SaveAsCurrentObjVariable .....	4-266
78. SaveAsDocumentVariable .....	4-267
79. SaveAsPageVariable .....	4-268
80. SplitLeft.....	4-268
81. SplitRight .....	4-269
82. SumFIELDS .....	4-270
83. TimeStamp.....	4-271
84. TrailingZeros .....	4-271
85. TrimSpaces.....	4-272
86. WriteField.....	4-272
XXX. Vote Action.....	4-273
1. VoteFld.....	4-273
XXXI. vScan Actions.....	4-274
1. AddDocument .....	4-274
2. CopyFile .....	4-275
3. DeletelImageFile.....	4-275
4. MoveImageFileToDirectory .....	4-276
5. Scan.....	4-276
6. SearchInSubdirectory.....	4-277
7. SetFastMode.....	4-277
8. SetImageType.....	4-278
9. SetMailSourceFolder .....	4-279
10. SetMaxImageFiles.....	4-279
11. SetMultiPageTiff .....	4-280
12. SetSortOrder .....	4-280
13. SetSourceDirectory .....	4-281

XXXII. Zone Actions .....	4-282
XXXIII. Zone Actions Procedural .....	4-283
1. AdjustZonesToImageOffset .....	4-283
2. AnchorPage .....	4-283
3. CalculateLocalOffset .....	4-284
4. CreateBlockCCO .....	4-284
5. FindBlocks_WhiteSpace .....	4-285
6. FindDataBlocks .....	4-285
7. FindLineItems .....	4-286
8. FindRegExBlocks.....	4-286
9. GetZoneText .....	4-287
10. InheritParentPosition.....	4-287
11. LoadBlockCCO .....	4-288
12. LoadZones .....	4-288
13. MCCOPositionAdjust.....	4-289
14. PopulateZNField.....	4-289
15. PopulateZNLinItemField .....	4-290
16. ReadZones .....	4-291
17. RegisterPage .....	4-291
18. ScanDetails .....	4-292
19. ScanDetailsByLine.....	4-292
20. ScanDetailsByVSpace .....	4-293
21. ScanLinItem .....	4-293
22. SetEOL .....	4-294
23. SetEOL_CRLF.....	4-295
XXXIV. Zone Actions Dimensional .....	4-296
1. ZoneBOTTOM_LowerBound .....	4-296
2. ZoneBOTTOM_ImageBottom.....	4-297
3. ZoneBOTTOM_UpperBound .....	4-297
4. ZoneImage_SaveAs .....	4-298
5. ZoneLEFT_ImageLeft.....	4-299
6. ZoneLEFT_LeftBound.....	4-299
7. ZoneLEFT_RightBound .....	4-300
8. ZoneRIGHT_ImageRight.....	4-300
9. ZoneRIGHT_LeftBound .....	4-301
10. ZoneRIGHT_RightBound.....	4-301
11. ZoneTOP_ImageTop .....	4-302
12. ZoneTOP_LowerBound .....	4-302
13. ZoneTOP_UpperBound .....	4-303

# Introduction

An action has these attributes:

- ◆ It is a programming **script** contained within an **Action file** that consists of scripts which share a broad objective.
- ◆ It is effective only when it has been assigned to a *Rulemanager* rule or to a function of a Datacap Studio ruleset. A rule or function contains one or more actions that attempt to carry out a specific procedure when the rule/ruleset is **bound** to an **object** of an application's **Document Hierarchy**.
- ◆ An action can be assigned to one rule, or multiple rules.
- ◆ To be added to a rule or function, many actions need one or more **parameters** – values that gives the action purpose and direction.
- ◆ If an action fails for any reason, the rule does not invoke its next action. Instead, the rule shuts down and *Rulemanager* or *Datacap Studio* runs the *next* rule or function in the ruleset.



Invoice Rulemanager – Rules panel

Chapter 3 examined the fields and tools of the *Rulemanager Window's Rules* panel.

This chapter focuses exclusively on actions. It begins with explanations of the Action files, and of the steps you take to add an Action file to an Actions Library (below). Chapter 4 then describes *all* actions in each category (Page 15).

An action's explanation uses this format:

Actions Library	Locate.rra/.rrx
Description	Moves to a word which is <i>above</i> the current word by <i>n</i> lines.
Parameter	An <i>Integer</i> indicating the number of lines above the current word.
Level	The level or levels of the Document Hierarchy to which the action can be applied.
Returns	<i>True</i> if a word is found. Otherwise, <i>False</i> .
Example	<p>WordFind(Total)  <b>GoAboveWord(1)</b>  GoRightWord(1)  UpdateField()</p> <p>This <i>Taskmaster for APTs</i> sequence finds the word which is the <i>Tax</i> field's <i>entered</i> value ("29.78") when a fingerprint has these fields and values:</p> <p style="padding-left: 40px;"><b>Tax</b> 29.78  <b>Total</b> 234.70</p>
See also...	GoBelowWord

## Rulemanager's Rulesets and Actions Libraries

In Rulemanager, rules of a particular RuleSet Type – **Locate**, for example, or **Validation** – can include only actions which belong to the RuleSet Type's **Actions Library**.

An Actions Library consists of one or more Action files (.rra): an Action file contains scripts for actions in a specific category.

The Actions Library of a typical application's **Locate** RuleSet Type probably includes actions in at least three categories:

- **Locate** actions (**Locate.rra/.rrx**) find fields on a *working* fingerprint, then their *recognized* values, and add these values to the fingerprints' Data file/
- **Zone** actions (**zone.rra**) find fields on a *working* fingerprint that are *zoned* fields on a *permanent* fingerprint, and isolates the values in those fields.
- **Recognition** actions (**rrrunner.dcs**) are housekeeping procedures, and are automatically included in every RuleSet Type's Actions Library.

- ✓ To identify the Action files in a RuleSet Type's library, you can select the RuleSet Type from the drop-down list at the top of the *Rules* panel and scroll through the **Actions** sector at the bottom of the panel (Chapter 3).
- ✓ To further investigate – and even modify - a RuleSet Type's Actions Library, you can click on the sector's **Setup RuleSet Types and Library Files** icon. This step opens the **RuleSet Type** dialog.
  - The **RuleSet Type** dialog:
    - ◆ Lists each Action file included in a RuleSet Type's Actions Library *except* the Recognition file (**rrrunner.dcs**) that is automatically included in every library.
    - ◆ Helps you add an Action file to a library, or remove a file.
    - ◆ Gives you a chance to define a new RuleSet Type, assign one or more Action files to the library, or modify a RuleSet Type's name.
- ✓ The **RuleSet Type** dialog is easy to use, and procedures for defining a RuleSet Type and providing it with Action files are straightforward. Chapter 5 of the *Guide to Taskmaster Rules* describes every aspect of the **RuleSet Type** dialog, and shows you how to construct, test and modify RuleSet Types.

## AutoDoc Actions

**AutoDoc** actions are most commonly combined with actions from the **Recog** Actions Library to automatically identify documents.

The sequence below shows how **AutoDoc** actions combine with actions in the **Recog** library to form a global **Recognition** rule. In this example, the rule applies to current pages such as those represented by the *APT* application's **APT Page** object.

ACTION	Library	PURPOSE
AnalyzeImage()	Recog	Creates a geometric representation of the current image with lists of characters, words, and lines, and stores these details in a <page>.cco file.
CreateFields()	DCO	Refers to the <i>setup</i> Document Hierarchy to create <b>Field</b> objects in a <i>runtime</i> Document Hierarchy – a page's Data file (.xml).
RotateImage()	Recog	Using the results of the <b>AnalyzeImage</b> action, tests to see if the text on the page is sideways and, if necessary, rotates the image to the correct orientation.
RecognizePageOCR_S()	Recog	Reads and recognizes all words on the page and adds the <i>recognized</i> values to the <page>.cco file created by <b>AnalyzeImage</b> .
SetSearchArea(0.5)	AutoDoc	States the percentage of the <i>recognized</i> page, starting from the top of the page, that is to be used during attempts to match the <page>.cco file with a fingerprint in the application's <b>Fingerprint</b> directory (below).
SetProblemValue(0.7)	AutoDoc	Specifies a minimum <b>Matching Tolerance Rating</b> . Values range from 0 to 1, with higher values representing less tolerance.
SetFingerprintDir (C:\ParentDir\Invoice\Fingerprint)	AutoDoc	Locates the directory that holds the application's fingerprints.
FindFingerprint(True)	AutoDoc	Initiates the fingerprint matching procedure and creates a new fingerprint if no match is found.

The tables which follow describe the **AutoDoc** actions.

### BlankPagesIDBySize

Actions Library	AutoDoc.rra/rrx
Description	Uses the Image file's size to determine if the file represents a "blank" page. Any page with an Image file smaller than the <b>minimum size</b> parameter (in bytes) will be assigned the <b>Page Type</b> value you enter as a parameter.
Parameter	A three-part, comma-separated value consisting of:  <i>Numeric</i> value indicating the minimum size in bytes that qualifies a page as a blank page.  <i>String</i> value representing the Page Type of a blank page.  <i>Numeric</i> value (0, 1 or 2) to designate which pages in a multi-page Image file are to be evaluated.  The third parameter is optional. 0=both sides of a two-page Image file. 1=odd pages only. 2=even images only.
Level	Batch, Document, or Page levels.
Returns	<i>False</i> if any parameter is invalid, or the rule with this action is bound to a <b>Field</b> object of the Document Hierarchy. Otherwise, <i>True</i> .
Example	BlankPagesIDBySize(1000,Blank_Page)

### CalculateOffset

Actions Library	AutoDoc.rra/rrx
Description	Sets the standard Offset value to be used when matching <i>source</i> pages to fingerprints.  The actual shift is four times this value in pixels (the default is 24 pixels). Increasing this value improves but slows the matching process: you can use the <b>SetMaxOffset</b> action to limit this value
Parameter	<i>Integer value between 1 and 255.</i>
Level	All levels
Returns	False if the parameter is not valid. Otherwise, True.
Example	<b>CalculateOffset(4)</b>  This action sets the Offset value at 16 pixels (4 * 4), well below the 24 pixel default.



## CreateFingerprint

Actions Library	AutoDoc.rra/rrx
Description	<p>Creates a fingerprint for the current <i>source</i> page.</p> <p>The resulting fingerprint will consist of two files: the page's Image file (.tif) and its Processing file (.cco).</p> <p><b>Alert!</b> A <b>SetFingerprintDir</b> action must precede this action.</p>
Parameters	None
Level	Page level only.
Returns	<i>False</i> if the rule with this action is not bound to a <b>Page</b> object of the Document Hierarchy; if the current page does not have an Image file; or if the fingerprint's two files cannot be created. Otherwise, <i>True</i> .
Example	<pre>SetFingerprintDir(C:\ParentDir\APT\fingerprint) <b>CreateFingerprint()</b></pre>
See also.....	SetFingerprintDir, DeleteFingerprint

## DeleteFingerprint

Actions Library	AutoDoc.rra/rrx
Description	<p>Deletes the Image file (.tif) and Processing file (.cco) of the current page's fingerprint from the application's <b>fingerprint</b> directory, and its record from the Rules database.</p> <p><b>Alert!</b> A <b>SetFingerprintDir</b> action must precede this action.</p>
Parameters	None
Level	Page level only.
Returns	<i>False</i> if action is not applied at the Page level; or if the fingerprint's files cannot be deleted. Otherwise, <i>True</i> .
Example	<pre>SetFingerprintDir(C:\ParentDir\APT\fingerprint) <b>DeleteFingerprint()</b></pre>
See also.....	SetFingerprintDir, CreateFingerprint

## FindBlackFingerprint

Actions Library	AutoDoc.rra/rrx
Description	<p>Attempts to match <b>black HCFA-1500</b> and <b>UB-92</b> medical claims to fingerprints in the <b>fingerprints</b> directory of the <i>Taskmaster for Medical Claims</i> application.</p> <p>If a match does not occur, the action will respond according to the parameter(s) you enter.</p>
Parameters	<p>Two comma-separated values:</p> <p><i>True/False</i>: <i>True</i> if a task is to create a new fingerprint and add the fingerprint's two files (.tif and .cco) to the <b>fingerprint</b> directory if a match does not occur; <i>False</i> if the task is to proceed without creating a new fingerprint.</p> <p>Optional: The Page Type that is to be assigned to the newly created fingerprint if the first parameter is <i>True</i>. If you do not include this parameter, the action will assign the Page Type of the current page (<i>HCFA 1500</i>, for example.)</p>
Level	Page level only
Returns	<i>False</i> if the action is not applied at the Page level or if the first parameter is <i>False</i> and a fingerprint match does <b>not</b> occur. Otherwise, <i>True</i> .
Example	<pre>AnalyzeImage() SetSearchArea(0.5) SetProblemValue(0.7) SetFingerprintDir(\Datacap\MClaims\fingerprint) <b>FindBlackFingerprint(True,hcfa 1500)</b></pre> <p>In this sequence, the <b>FindBlackFingerprint</b> action will only use the top 50% of the fingerprint to search for a match. It will accept a match of 0.7 or higher.</p> <p>If no match is found, the sequence will create a new fingerprint and store it in the location specified by the <b>SetFingerprintDir</b> action.</p> <p>If the parameter has been set to <i>False</i>, no fingerprint will be created if a match is not found.</p>

## FindFingerprint

Actions Library	AutoDoc.rra/rrx
Description	<p>Attempts to match the current page to a fingerprint and creates a new fingerprint if a match does not occur.</p> <p>Include this action <i>after</i> a rule's SetSearchArea, SetProblemValue, and SetFingerprintDir actions.</p>
Parameters	<p>Two comma-separated values:</p> <p>A <i>True/False</i> value: <i>True</i> if a task is to create a new fingerprint and add it to the <b>fingerprint</b> directory if a match does not occur; <i>False</i> if the task is to proceed without creating a new fingerprint to add to the <b>fingerprint</b> directory.</p> <p>Optional: The Page Type that is to be assigned to the newly created fingerprint. If omitted, the current Page Type of the current page will be used.</p>
Level	Page level only.
Returns	<i>False</i> if the action is not applied at the Page level, or if the parameter is <i>False</i> and a fingerprint match does <i>not</i> occur. Otherwise, <i>True</i> .
Example	<pre>AnalyzeImage() RotateImage() RecognizePageOCR_S() SetSearchArea(0.5) SetProblemValue(0.7) SetFingerprintDir(\Datacap\Invoice\Fingerprint) <b>FindFingerprint(True,Invoice_Page)</b></pre> <p>In this sequence, the <b>FindFingerprint</b> action will only use the top 50% of the current page to search for a match. It will accept a match of 0.7 or higher.</p> <p>If no match is found, the sequence will create a new fingerprint and store it in the location specified by the <b>SetFingerprintDir</b> action.</p> <p>If the parameter has been set to <i>False</i>, no fingerprint will be created if a match is not found.</p>

## FindTemplate

Actions Library	AutoDoc.rra/rrx/
Description	Finds the fingerprint that a preceding <b>FindFingerprint</b> action matched to the current <i>source</i> page or created instead (see the explanation of the <b>FindFingerprint</b> action.)
Parameters	<p><i>String</i> boolean value:</p> <p><i>True</i> if the action is to create a new fingerprint if the current <i>source</i> page does not match the target fingerprint.</p> <p><i>False</i> if the action is not to create a new fingerprint if there is not a match.</p>
Level	Page only.
Returns	False if the action is not applied at the Page level.
Example	<pre>RotateImage() &lt;/scr&gt;&lt;br/&gt; RecognizePageOCR_S() &lt;/scr&gt;&lt;br/&gt; SetSearchArea(0.5) &lt;/scr&gt;&lt;br/&gt; SetProblemValue(0.7) &lt;/scr&gt;&lt;br/&gt; SetFingerprintDir(Datacap\1040EZ\Fingerprint) FindFingerprint(True, Invoice_Page) <b>FindTemplate(True)</b></pre>

## MergeCCOs\_ByType

Actions Library	AutoDoc.rra/rrx
Description	Merges the Fingerprint files (.cco) associated with <b>Page</b> objects of the Document Hierarchy.
Parameters	Comma-separated <i>String</i> values indicating the Page Types of the Document Hierarchy objects to be merged.
Level	Document level only.
Returns	<i>False</i> if a Fingerprint file (.cco) for one of the Page Types is not available. Otherwise, <i>True</i> .
Example	MergeCCOs_ByType(Invoice_Page,Invoice_Cont)  The <b>MergeCCOs_ByType</b> action above permits all values of the <i>source</i> pages to be assigned to a single, searchable Fingerprint Processing file (.cco).

## SetFilter\_HostName

Actions Library	AutoDoc.rra/rrx
Description	Sets the Fingerprint Class filter for the identification (matching) algorithm.  The filter will force the fingerprint-matching algorithm to use only fingerprints associated with the specified Fingerprint Class.
Parameters	The <i>String</i> value of the Fingerprint Class you want to use as the filter.
Level	Page level
Returns	Always True.
Example	SetFilter_HostName(TEKSystems)

## SetFilter\_PageType

Actions Library	AutoDoc.rra/rrx
Description	Sets the Page Type filter for the identification (matching) algorithm.  The filter will force the fingerprint-matching algorithm to use only fingerprints associated with that Page Type.
Parameter	The <i>String</i> value of the Page Type you want to use as the filter.
Level(s)	All, but generally at the Page level.
Returns	Always <i>True</i> .
Example	SetFilter_PageType(Invoice_Page)

## SetFingerprint

Actions Library	AutoDoc.rra/rrx
Description	Sets a newly created fingerprint's Fingerprint Class and Fingerprint Class ID values.  After the mandatory Fingerprint Class value and optional Fingerprint Class ID have been manually assigned by a fingerprint creation task, this action places these values into the <b>Host</b> table of the application's Rules database – as RefName and RefID values.
Parameters	A two-part, comma-separated value consisting of:  The name of the <b>Field</b> object of the Document Hierarchy that provides the Fingerprint Class value.  The name of the <b>Field</b> object of the Document Hierarchy that provides the Fingerprint ID value. (This is an optional parameter.)  Both parameters require an @ prefix (see the example below.)
Level	Page level only
Returns	<i>False</i> if either parameter is invalid. Otherwise, <i>True</i> .
Example	SetFingerprint(@VendorName,@VendorID)  In this example, <i>runtime</i> values of the <i>VendorName</i> and <i>VendorID</i> Field objects will populate the <b>Host</b> table of the application's Rules Database.

## SetFingerprintDir

Actions Library	AutoDoc.rra/rrx
Description	Sets the <b>Fingerprint</b> directory of your application. This directory contains the application's fingerprints.
Parameter	A <i>String</i> value specifying the directory's name <i>and</i> its path. <i>Alert!</i> A Drive ID (such as C:\) is <i>optional</i> .
Level(s)	All, but generally applied at the Page level.
Returns	Always <i>True</i> .
Example	SetFingerprintDir(C:\ParentDir\Invoice\Fingerprint) The action above identifies the location and path of the <i>APT</i> application's <b>Fingerprint</b> directory.

## SetFingerprintWebServiceURL

Actions Library	AutoDoc.rra/rrx
Description	Provides the Uniform Resource Locator (URL) that is a link to a pre-established Fingerprint Web Service. Note: this action is not effective unless the Fingerprint Web Service has been installed and configured.
Parameters	<i>String</i> value specifying the URL that is the link to the Fingerprint Web Service (see the example below.)
Level	All
Returns	Always <i>True</i> .
Example	<b>SetFingerprintWebServiceURL(http://www.grandcorp.AccountsReceivable.com/fingerprints)</b>

## SetMaxOffset

Actions Library	AutoDoc.rra/rrx
Description	<p>Sets the Maximum Offset value while matching <i>source</i> pages to fingerprints.</p> <p>The actual shift is four times the maximum offset value in pixels (4 * MaxOffset). Increasing this value improves but slows the matching process.</p> <p>The default value is 6: <math>4 * 6 = 24</math> pixels.</p>
Parameters	<i>Integer</i> value between 1 and 255
Level	All levels.
Returns	<i>False</i> if the parameter is invalid. Otherwise, <i>True</i> .
Example	<p>SetMaxOffset(12)</p> <p>This example results in a Maximum Offset value of 48 pixels, well above the 24 pixel default.</p>



## SetProblemValue

Actions Library	AutoDoc.rra/rrx
Description	<p>Uses the <i>decimal</i> value you supply as a parameter to set a minimum <b>Matching Tolerance Rating</b>.</p> <p><b>Important!</b> A lower rating results in lower tolerance and a greater chance for a match, but also a greater chance for a <i>False</i> match.</p>
Parameters	A <i>decimal</i> value from 0.00 (Lowest Tolerance) to 0.99 (Highest Tolerance).
Level	All, but generally at the Page level.
Returns	<i>False</i> if the parameter is missing or the parameter is not <i>numeric</i> . Otherwise, <i>True</i> .
Example	<pre>AnalyzeImage() CreateFields() RotateImage() RecognizePageOCR-S() SetSearchArea(0.5) <b>SetProblemValue(0.70)</b> SetFingerprintDir(\Datacap\Invoice\Fingerprint) FindFingerprint(True)</pre> <p>In this sequence, the <b>FindFingerprint</b> action assigns a Matching Tolerance Rating that is neither overly restrictive nor unrealistically accepting.</p> <p>If the rule's conditions do not result in a match, and <i>True</i> is used as the parameter for the <b>FindFingerprint</b> action, a new fingerprint will be added to the library.</p>
See also...	SetSearchArea

## SetSearchArea

Actions Library	AutoDoc.rra/rrx
Description	Uses the <i>decimal</i> value(s) you supply to determine the portion of the current page that will be used to find a matching fingerprint.
Parameter	<p>Two parameters:</p> <p>1) A <i>decimal</i> value from 0.01 (1%) to 1.00 (all) to indicate how much of the page is to be matched. If only one parameter is used, the range for fingerprint matching will be 0 to <i>x</i>, where <i>x</i> is the parameter. For example, 0.5 indicates that fingerprint matching will be limited to the first half of the page (0 to 50%).</p> <p>2) <b>Optional:</b> A <i>decimal</i> value from 0.01 (1%) to 1.0 (all) to indicate the end point on the page to be used for fingerprint matching. For example: if the first parameter is 0.6, and the second parameter is 1.0, the last 40% of the page will be used for fingerprint matching (60-100%).</p>
Level(s)	All but generally at the Page level.
Returns	<i>False</i> if the first parameter is missing or is not <i>numeric</i> . Also returns <i>False</i> if the second parameter is not <i>numeric</i> . Otherwise, <i>True</i> .
Example	<p>SetSearchArea(0.5)</p> <p>This example compares lines and words in the upper 50% of the current page to the lines and words in the same portion of each fingerprint.</p> <p>SetSearchArea(0.5,1.0)</p> <p>The example above compares lines and words in the lower 50% of the current page to the lines and words in the same portion of each fingerprint.</p>
See also...	SetProblemValue

## SetTemplateDir

Actions Library	AutoDoc.rra/rrx
Description	Sets the <b>Template</b> directory of your application. This directory contains the application's fingerprint templates.
Parameter	A <i>String</i> value specifying the directory's name <i>and</i> its path. <i>Alert!</i> A Drive ID (such as C:\) is <i>optional</i> .
Level(s)	All, but generally applied at the Page level.
Returns	Always <i>True</i> .
Example	SetTemplateDir(C:\ParentDir\APT\Template) The action above identifies the location and path of the <i>APT</i> application's <b>Template</b> directory.

## Barcode\_X Actions

**Very important!** *Taskmaster Version 7.5* does not include a separate **Barcode\_X.rra** file for *Rulemanager*. The descriptions in the tables below cover actions of Datacap Studio's **Barcode\_X.rrx** file. (You can also review the description of an individual action by right-clicking on its listing in the **Actions Library** area of Datacap Studio's **Rulemanager** page.)

### GetBarCode

Actions Library	Barcode_X. rrx
Description	Returns the value of any barcode within a zone.  A RuleSet with this action can be applied to a <b>Page</b> object or <b>Field</b> object of the Document Hierarchy.  When applied to a <b>Page</b> object, the action reads the first barcode it encounters on the page and creates a <b>Page</b> -level variable ( <b>GetBarCode</b> ), and places the barcode's value in the variable.  When applied to the <b>Field</b> object of a <i>zoned</i> field, the action assigns the barcode's value to the field.
Parameters	None.
Level	Page or Field level.
Returns	<i>False</i> if the action is not applied to a <b>Field</b> object or a <b>Page</b> object. Otherwise, <i>True</i> .
Example	GetBarCode()
See also...	ReadBarCode, MatchBarCode

## MatchBarCode

Actions Library	Barcode_X.rxx
Description	Searches all the barcodes on the current page and checks to see if one of them matches the parameter value.  If a match occurs, the barcode value is placed into a “GetBarCode” page variable.
Parameter	<i>String</i> value that you want to match with a barcode.
Level	Page level only.
Returns	<i>False</i> if the action is not run at the Page level, or if no barcode is found on the page. Otherwise, <i>True</i> .
Example	MatchBarCode()
See also...	ReadBarCode, GetBarCode

## ReadBarCode

Actions Library	Barcode_X. rxx
Description	Checks to see if the current page contains a barcode with the value specified by the parameter.  This action uses the first barcode it encounters.  A RuleSet can use this action to identify a document’s <i>Separator</i> page.
Parameter	The value you want to match.
Level	Page or Field level only.
Returns	<i>True</i> if the parameter’s value matches the actual barcode value; otherwise, <i>False</i> .
Example	<b>ReadBarCode(Separator)</b> SetPageType(Separator)  This example looks for a barcode with the value “Separator”. If found, the second action - a <b>DCO action</b> -establishes the page as a <i>Separator</i> page.
See also...	GetBarCode, MatchBarCode

## cco2cco Actions

The **cco2cco** actions sort the words and lines in a Fingerprint file (.cco) created by a Recognition engine, for use by navigation and pattern match actions.

The **NormalizeCCO** action sorts the words and lines in a Fingerprint file (.cco) created by a Recognition engine, for use by navigation and pattern match actions.

### cco2cco

Actions Library	cco2cco.rra/.rrx
Description	Sorts the words and lines in a Fingerprint file (.cco) created by a Recognition engine, for use by navigation and pattern match actions.  This action is usually required before using <b>Locate</b> actions or the <b>pat_RecogMatch_ID</b> action to find recognized text on a page. For a complete explanation, see the Details section below
Parameter	String value of the path of the folder containing the Action file ( <b>cco2cco.rra</b> ) and two files that it requires: <b>wleft.xml</b> and <b>wbottom.xml</b> . Typically, this would be <b>C:\ParentDir\rrs\rrx</b> .
Level(s)	Page.
Returns	Always True.

continued on next page →

## NormalizeCCO (continued)

Example	<p>For this action, a fingerprint is calculated for a particular image in a batch, as opposed to the Fingerprint database, which contains fingerprints for various page types and layout variations that have been defined for a particular application.</p> <p>There are two types of Fingerprint files. The <b>AnalyzeImage</b> action creates a geometric fingerprint containing lines and "words" based only on the black pixels in the image. Full-page recognition actions (<b>RecognizePageOCR_S</b>, <b>RecognizePageICR_C</b> etc.) create a fingerprint based on the results of recognition, i.e. both geometry and text of the recognized characters, words and lines.</p> <p>In Recognition-based fingerprints, the order of lines and words may appear to be arbitrary, especially if the page contains images, tables, stamps, or blocks of text with varying font sizes. This can cause unpredictable results from <b>Locate</b> actions that navigate geometrically. The word-matching and phrase-matching action <b>pat_RecogMatch_ID</b> also requires well-ordered text to work reliably.</p> <p>The <b>cco2cco</b> action re-orders the words of text in a Recognition-based fingerprint into lines and words in "standard" reading order, from top to bottom and left to right.</p> <p><b>IMPORTANT NOTE:</b> <b>cco2cco</b> discards any "words" or blocks taller than 50 pixels, as well as the <i>less than</i>, <i>greater than</i> and <i>ampersand</i> characters.</p> <p>If the <b>AnalyzeImage</b> action is called before full-page recognition, the recognized text is placed into the geometry created by the action. This hybrid Fingerprint file is not always suitable for <b>cco2cco</b>. To force creation of a pure recognition-based fingerprint, call <b>SetFingerprintRecogPriority(True)</b> before full-page recognition. This guarantees that any existing geometric fingerprint will be ignored, and it applies to OCR_S and ICR_C only.</p> <pre>SetFingerprintPriority(True) RecognizePageOCR_S() cco2cco(C:\ParentDir\rrs\rrx) pat_RecogMatch_ID()</pre>
---------	--

## NormalizeCCO

Actions Library	cco2cco.rra/rrx
Description	<p>Sorts the words and lines in a Fingerprint file (.cco) created by a Recognition engine, for use by navigation and pattern match actions.</p> <p>This action is usually required before using <b>Locate</b> actions or the <b>pat_RecogMatch_ID</b> action to find recognized text on a page. For a complete explanation, see the Example below</p>
Parameter	None
Level	Page
Returns	Always True.
Example	<pre>SetFingerprintPriority(True) RecognizePageOCR_S_ <b>NormalizeCCO()</b> pat_RecogMatch_ID()</pre> <p>This action sorts the words and lines in a Fingerprint file (.cco) created by a Recognition engine, for use by navigation and pattern match actions. The action is usually required before using <b>Locate</b> actions or the <b>pat_RecogMatch_ID</b> action to find recognized text on a page.</p> <p>In this context, the fingerprint is calculated for a particular image in a batch, as opposed to the Fingerprint database, which contains fingerprints for various page types and layout variations that have been defined for a particular application.</p> <p>There are two types of Fingerprint files. <b>The AnalyzeImage</b> action (<b>recog_shared.rra</b>) creates a geometric fingerprint containing lines and "words" based only on the black pixels in the image. Full-page recognition actions (<b>RecognizePageOCR_S</b>, <b>RecognizePageICR_C</b> etc.) create a fingerprint based on the results of recognition, i.e. both geometry and text of the recognized characters, words and lines.</p>

continued on next page →



## NormalizeCCO (continued)

Example (ct)	<p>In Recognition-based fingerprints, the order of lines and words may appear to be arbitrary, especially if the page contains images, tables, stamps, or blocks of text with varying font sizes. This can cause unpredictable results from Locate actions that navigate geometrically. The word-matching and phrase-matching action <b>pat_RecogMatch_ID</b> also requires well-ordered text to work reliably.</p> <p>The <b>cco2cco</b> action re-orders the words of text in a Recognition-based fingerprint into lines and words in "standard" reading order, from top to bottom and left to right.</p> <p><b>IMPORTANT!</b> <b>cco2cco</b> discards any "words" or blocks taller than 50 pixels, as well as the less than , greater than and ampersand characters.</p> <p><b>NOTE:</b> If the <b>AnalyzeImage</b> action is called before full-page recognition, the recognized text is placed into the geometry created by <b>AnalyzeImage</b>. This hybrid Fingerprint file is not always suitable for <b>cco2cco</b>. To force creation of a pure recognition-based fingerprint, call the <b>Recog_Shared</b> file's <b>SetFingerprintRecogPriority(True)</b> action before full-page recognition. This guarantees that any existing geometric fingerprint will be ignored, and it applies to OCR_S and ICR_C only</p>
--------------	--

## CMS Actions

The CMS actions carry out procedures which focus on recognition of *HCFA-1500* claims of the *Taskmaster for Medical Claims* application.

### DetectHCFA

Actions Library	cms.rra/rrx
Description	Checks values in the <i>24cEMG_(n)</i> and <i>24jRefID(n)</i> field of a HCFA page's Service Table to determine if the claim is a <i>CMS 1500</i> claim.
Parameter	None
Level(s)	Page only.
Returns	<i>False</i> if the ruleset with this action is not bound to a <b>Page</b> object, or if the action encounters invalid data types. Otherwise, <i>True</i> .
Example	DetectHCFA()

### ValidQualifier

Actions Library	cms.rra/rrx
Description	Checks the value in the bound <b>Field</b> object of the Document Hierarchy against valid Qualifier codes.
Parameter	None
Level(s)	Field only.
Returns	Always <i>True</i> .
Example	ValidQualifier()

## DataClip Action

The **DCI\_ClipField** action *clips* a specific field in the image of the current *source* page – and

- Generates a separate Image file (.tif) that displays *only* the clipped field and any value it contains;
- Adds a new “page” to the current Page file (.xml). The new page is an adjunct to the *source* page, and contains only the clipped image.

### DCI\_ClipField

Actions Library	dcclip.rra/.rrx
Description	<p>During processing, clips the field on the Image file (.tif) of every <i>source</i> page represented by the <b>bound Field</b> object of the Document Hierarchy – and generates a separate Image file (.tif) displaying the clipped field’s contents only.</p> <p>The action also adds a new “page” representing the new Image file to the current task’s Page file (.xml).</p> <p><b>Alert!</b> The Image ID assigned to the Image file representing the <i>source</i> page has this format: <b>tm000001.tif</b>. The Image ID of the clipped field’s Image file adds one character and has this format: <b>tm00000101.tif</b></p> <p>The second pair in the batch will have these ID’s: <b>tm000002.tif</b>. and <b>tm00000201.tif</b>. (This assumes that a <i>source</i> page has only one clipped field.)</p>
Parameters	<p>Two comma-separated <i>string</i> values:</p> <p>2.) The <b>Page Type</b> that is to be assigned to the Image file containing the clipped field and its value. <b>Remember!</b> The new Image file is represented by a new page listing in the current Page file (.xml). The Page Type value you assign will be used to identify pages with clipped fields.</p> <p>2.) The <b>Page Status</b> to be assigned to pages with clipped images of the bound <b>Field</b> object of the Document Hierarchy. <b>Alert!</b> Be sure that the status you assign conforms to Page Status specifications used throughout your application.</p> <p>Both values will be picked up by the Page file (.xml) generated by the <i>rulerunner task</i> that applies a rule containing this action.</p>
Level	Field only.

### DCI\_ClipField )continued)

Returns	<p><i>False</i> if either parameter is invalid. Otherwise <i>True</i>.</p> <p>If the <b>DCI_ClipField</b> action cannot locate the target field on a <i>source</i> page, the action will not generate an Image file for the clipped field and will not add a corresponding page to the current Page file (.xml).</p>
Example	<p>DCI(OfficePens_Page, 0)</p> <p>In this example, <i>OfficePens</i> is the Page Type value specified in Datacap Studio's <b>Zones</b> page.</p>

## DCO Actions

DCO actions work directly on the structure and objects of an application's Document Hierarchy. They are included in the libraries of rulesets such as **Recognize**, **CreateDocs**, and **Validate**.

**Note:** The *runtime* form of most Verify tasks is usually bound to the **Page** level of the Document Hierarchy; therefore, rules will not run in the Verify task if they are applied at the **Batch** or **Document** level.

The following tables describe the **DCO** actions.

### ChkConfidence

Actions Library	DCO.rra/.rrx
Description	Checks the confidence of all field data on selected pages (selected by Page Status).  If any fields contain Low Confidence data, the page is marked with the status you specify as a parameter.
Parameters	Two or three comma-separated values:  The <i>Numeric</i> value of the minimum confidence required. This value will be superseded on a field-by-field basis, if the field's <b>ReqConf</b> variable is set.  The <i>Numeric</i> Page Status code to assign to any page that has one or more fields with Low Confidence data: if a field's <b>ConfidenceString</b> property contains a value lower than the first parameter. Subfields, line items, etc are included. Typically, "1" ( <i>Problem</i> ) is the value of this parameter.  If only two parameters are specified, only pages with Status=0 will be checked by this action. If a third or additional parameter is supplied, those parameters specify the list of page statuses to be checked.
Level	All levels.  This action operates on the entire batch regardless of the level to which its rule is bound.
Returns	<i>True</i> if all fields in all <i>source</i> pages are High Confidence. <i>False</i> if any field has Low Confidence data, or if the parameters are not <i>Numeric</i> .
Example	ChkConfidence(8,1)
See also	ChkDocStatus

## ChkDCOStatus

Actions Library	DCO.rra/.rrx
Description	Confirms that the status of the Document Hierarchy's current object is identical to the status you enter as the parameter.
Parameter	The <i>Numeric</i> value of the status you're checking.
Level	All levels
Returns	<i>True</i> if a match occurs; otherwise, <i>False</i> .
Example	<p><b>ChkDCOStatus(0)</b> returns <i>True</i> if the current object has a status equal to 0, and <i>False</i> if it does not.</p> <p><b>ChkDCOStatus(48)</b> returns <i>True</i> if the current object has a status equal to 48, and <i>False</i> if it does not.</p>
See also...	ChkDCOType

## ChkDCOType

Actions Library	DCO.rra/.rrx
Description	Confirms that the <b>Type</b> property of the Document Hierarchy's current object is identical to the type you enter as the parameter.
Parameter	The <i>String</i> value of the <b>Type</b> property of the object you're checking.
Level	All levels
Returns	<i>True</i> if a match occurs; otherwise, <i>False</i> .
Example	<p>ChkDCOType(Invoice)</p> <p>Applied at the <b>Page</b> level, the action above returns <i>True</i> if the current object is an <b>APT Page</b> object (using the <b>APT</b> application as an example), and <i>False</i> if it is not.</p>
See also...	ChkDCOStatus, ChkLastDCOType

## ChkIntegrity

Actions Library	DCO.rra/.rrx
Description	Checks to be sure that the integrity of the batch, as detailed in the Page file of the current task, meets the integrity requirements set within the Document Hierarchy (Setup DCO).  “Integrity” refers to the correct types and numbers of pages within each document in the batch and the correct order of the pages in each document.
Parameter	None
Level	Batch and Document levels
Returns	Returns <i>True</i> if no integrity problems are found. Otherwise, <i>False</i> .
Example	ChkIntegrity()

## ChkLastDCOType

Actions Library	DCO.rra/.rrx
Description	Checks that the <b>Type</b> property of the Document Hierarchy’s <i>previous</i> object is identical to the type you enter as the parameter.  The example below is applied at the Page level and checks to see if the previous <b>Page</b> object’s type matches the parameter (Separator).
Parameter	The <i>String</i> value of the <i>previous</i> object’s name.
Level	All
Returns	<i>True</i> if the <b>Type</b> property of the Document Hierarchy’s <i>previous</i> object matches the parameter. Otherwise, <i>False</i> .
Example	<b>ChkLastDCOType(Separator)</b> SetPageType(Invoice) SetPageStatus(1)  Applied at the Page level, this sequence checks to see if the previous <b>Page</b> object was a <i>Separator</i> page. If so, the type of the <i>current</i> page is set to <i>APT</i> , and its status is set to “1.”

## ClearAltText

Actions Library	DCO.rra/.rrx
Description	<p>Clears character and confidence values from the <b>Character Array</b> position specified by the parameter. (A field in the Data file can hold more than one representation of the field's value. Values other than the current, visible value are accessed via an index number. The current value is at index 0, the next value is at index 1. Each additional value also has corresponding character confidences.)</p> <p><b>Note:</b> Most actions only work with characters and confidence values located in the first position of the <b>Character Array</b> (position 0).</p> <p>The <b>ClearAltTex</b> action is used with <i>Taskmaster Web's</i> Advanced Index task, for "Double Blind" data entry.</p>
Parameters	The index in the <b>Character Array</b> where you want to clear the character and confidence values. 0 is the first index, followed by 1, etc.
Level	Field level only
Returns	Always <i>True</i> .
Example	<p>PropagateToAltText(1)  <b>ClearAltText(0)</b></p> <p>In this example, all characters at the first index (0) of the <b>Character Array</b> will be copied to the second index (1). The second action will then clear character and confidence values from the first index in the <b>Character Array</b>.</p>
See also...	PropagateToAltText



## ClearDCO

Actions Library	DCO.rra/.rrx
Description	Clears all objects of the Document Hierarchy which are children of the bound object, and their variables.
Parameters	None.
Level	All.
Returns	<i>True</i> if all child objects and their variables are removed; otherwise <i>False</i> .
Example	<pre>CreateFields() ClearDCO()</pre> <p>Applied at the Page level, the example will first add fields to the page and then remove them.</p>

## CopyPD2DD

Actions Library	DCO.rra/.rrx
Description	Assigns the value in a <b>Page</b> object's <b>PD(Page Data)</b> variable to the <b>Document</b> object's <b>DD (Doc Data)</b> variable.
Parameters	None.
Level	Page level only.
Returns	<i>False</i> if the action is not at the Page level, or if the <b>PD</b> variable has no value. Otherwise, <i>True</i> .
Example	<b>CopyPD2DD ( )</b>

## CreateDocuments

Actions Library	DCO.rra/.rrx
Description	<p>Arranges the contents of a task's Page file (for example, <b>Recognition.xml</b>.) into documents based on the Document Integrity rules (min, max, and order) specified in your application's Document Hierarchy.</p> <p>The action assembles documents from the pages in the batch. Batches containing existing document structures will cause this action to return <i>False</i>, with no affect to the existing document structure.</p> <p><b>Alert!</b> This action is applied at the <b>Batch</b> level, and generally in its own RuleSet (in a <b>CreateDocs</b> rule, for example.)</p>
Parameters	None
Level	Batch level only
Returns	<i>True</i> if successful; otherwise, <i>False</i> .
Example	CreateDocuments()

## CreateFields

Actions Library	DCO.rra/.rrx
Description	<p>Creates the Data file for a page in a batch. The Data file for the first page in the batch, <i>tm000001</i>, as an example, is <b>tm000001.xml</b>.</p> <p>This Data file lists all fields for the current page based on the fields listed in the setup Document Hierarchy. Each field has an <b>ID</b> (for an <i>APT</i> page, for example, the <i>Date</i> field's ID is <b>Date</b>), and three properties with <i>default</i> values: <b>TYPE</b>, <b>Position</b>, and <b>Status</b>.</p> <p>Later, actions of various kinds (<b>Locate</b>, <b>Zone</b>, <b>Validation</b>, <b>DCO</b>, etc.) can assign other values to these properties. These actions can also add properties (variables) and values to the Data file, or remove properties and values.</p>
Parameters	None
Level	Page level only.
Returns	<i>True</i> if successful; otherwise, <i>False</i> .
Example	<pre>AnalyzeImage() RotateImage RecognizePageOCR_S() SetSearchArea(0.5) SetProblemValue(0.3) SetTemplateDir(\Datacap   Invoice\Template) FindFingerprint(True) <b>CreateFields()</b></pre> <p>This <i>APT</i> application's sequence sets up the current page for processing, recognizes the words on the page, associates the page with a fingerprint, and finally creates a Data file with blank fields for the page.</p>
See also...	CreateDocuments

## DeleteFields

Actions Library	DCO.rra/.rrx
Description	<p>Deletes all child fields and characters from calling object of the Document Hierarchy.</p> <p>This action will also remove the Data file (.xml) from the batch if called from a <b>Page</b> object</p>
Parameter	None.
Level	All.
Returns	True. If successfull. Otherwise, False.
Example	<code>DeleteFields()</code>

## PropagateToAltText

Actions Library	DCO.rra/.rrx
Description	<p>Copies the character and confidence values from the first index of the <b>Character Array</b> (index 0) to the index specified by the parameter. (The character “node” of a page’s Data file is an array that can hold many <i>recognized</i> character values, and their corresponding confidence values.)</p> <p><b>Note:</b> Rules will only work with characters and confidence values located in the first position of the <b>Character Array</b> (index 0).</p> <p>The <b>PropagateToAltTex</b> action is used with <i>Taskmaster Web’s</i> Advanced Index task, for “Double Blind” data entry.</p>
Parameters	The index of the <b>Character Array</b> where you want to copy the character and confidence values. 0 is the first index, followed by 1, etc.
Level	Field level only
Returns	Always returns <i>True</i> .
Example	<p>PropagateToAltText(1) ClearAltText(0)</p> <p>All characters in the first index of the <b>Character Array</b> will be copied to the second index. Then, the second action will clear the character and confidence values from the first index.</p>
See also...	ClearAltText

## SetDCOStatus

Actions Library	DCO.rra/.rrx
Description	Assigns a value to the <b>Status</b> property of the <i>current</i> object of the Document Hierarchy.
Level	All
Parameter	An <i>Integer</i> representing the new status.
Returns	Always <i>True</i> .
Example	<p>ChkDCOType(Invoice)  <b>SetDCOStatus(1)</b></p> <p>This sequence checks to see if the current object of the Document Hierarchy is a <b>Page</b> object – in this example, an <i>APT</i> page. If so, the value of the <b>Page</b> object’s <b>Status</b> property is set to “1”.</p>
See also...	SetDCOType, SetPageType

## SetDCOType

Actions Library	DCO.rra/.rrx
Description	Assigns a value to the <b>Type</b> property of the current object of the Document Hierarchy.
Parameter	A <i>String</i> value you’re assigning to the current object’s <b>Type</b> property.
Level	All
Returns	Always <i>True</i> .
Example	<p>ChkLastDCOType(Separator)  <b>SetDCOType(Invoice)</b></p> <p>This sequence checks to see if the previous object of the <i>APT</i> application’s Document Hierarchy was a <b>Page</b> object – in this case, a <i>Separator</i> page. If so, it sets <i>APT</i> as the <b>Type</b> property of the <i>current</i> object.</p>
See also...	SetDCOStatus, SetPageStatus, SetDocumentType

## SetDocumentType

Actions Library	DCO.rra/.rrx
Description	<p>Similar to the <b>SetDCOType</b> action but works at the Document, Page or Field level.</p> <p>The action assigns the Document Type you enter as a parameter to the current <b>Document</b> object of the Document Hierarchy. You can also use a <b>Field</b> object's value to set the Document Type. (Refer to the <b>Parameter</b> section.)</p>
Parameter	<p>The value you want to assign as the <b>Document</b> object's <b>Type</b> property.</p> <p>You can also designate a field in a <b>Page</b> object's Data file, and use its text value to set the Document Type. Simply enter the name of a valid <b>Field</b> object and surround it with single quotes. For example: 'Number'.</p>
Level	Document, Page, and Field levels
Returns	<i>False</i> if there are no <b>Document</b> objects in the Data file, or if the parameter is invalid. Otherwise, <i>True</i> .
Examples	<p>SetDocumentType('Number')</p> <p>or</p> <p>SetDocumentType(Invoice_Document)</p>
See also...	SetDCOType, SetPageType

## SetFldConfidence

Actions Library	DCO.rra/.rrx
Description	For the provided field name, sets the confidence for each character to the same value.
Parameter	None.
Level(s)	Field only.
Returns	Always <i>True</i> . (If an input parameter is invalid, no confidence levels are changed and a message is logged)
Example	<p>This action unconditionally sets the confidence values for each character within a field to a specific confidence. This Action can be useful to change confidence levels if the preceding actions are all successful.</p> <p>For example, it may make sense to perform a set of calculations on one or more fields and based on the outcome, it would imply that the confidence level of those fields should be high regardless of the confidence of the recognition engine. After the success of the calculations, you can call <code>SetFldConfidence()</code> to unconditionally set the values of these fields.</p> <p>The first parameter is the name of the field that will have its confidence level set.</p> <p>The second parameter is the desired confidence level, between 1 and 10. If the second parameter is not passed in, 10 is used as the default.</p> <p>This function supports smart parameters for the field name.</p>

## SetPageFingerprintID

Actions Library	DCO.rra/.rrx
Description	<p>Assigns a value to the <b>FingerprintID</b> property of the selected <b>Page</b> object of the Document Hierarchy.</p> <p><b>Alert!</b> <b>SetPageFingerprintID</b> will create a <b>FingerprintID</b> property of the current <b>Page</b> object if it does not already exist.</p>
Parameter	The <i>String</i> value of the Fingerprint ID.
Level	Page level only
Returns	<i>True</i> if the rule is applied at the Page level. Otherwise, <i>False</i> .
Example	<p>WordFind(MQSW)          SetPageFingerprintID(1010)</p> <p>In this sequence, if the <b>WordFind</b> action locates “MQSW” on the current page, the <b>SetPageFingerprintID</b> action assigns “1010” as the page’s Fingerprint ID.</p> <p>This links the page to a fingerprint with a Fingerprint ID of “1010”.</p>



## SetPageStatus

Actions Library	DCO.rra/.rrx
Description	<p>Similar to the <b>SetDCOStatus</b> action, but works <i>only</i> at the <b>Page</b> level.</p> <p>The action assigns the status you enter as the parameter to the current <b>Page</b> object of the Document Hierarchy.</p>
Parameter	<p><i>Numeric</i> value representing the status.</p> <p>The <i>APTs</i> application (as an example) employs three <i>default</i> Page Statuses:</p> <p>49 = ScanOK  0 = Complete  1 = Incomplete/Not validated</p> <p>You can define your own statuses by using the <i>Filter</i> tab of a task's <b>Task Settings</b> dialog.</p>
Level	Page level only.
Returns	Always <i>True</i> .
Example	<p>A kScan task might assign <i>Other</i> as the <b>Page Type</b> and <i>49</i> as the default <b>Page Status</b> for every successfully scanned image in the batch.</p> <p>The sequence below is an example of a rule that converts <i>Other</i> pages to <i>APT</i> pages, and assigns a Page Status to each:</p> <pre>SetPageType(Invoice) <b>SetPageStatus(1)</b></pre> <p>This combination establishes the page as an <i>APTs</i> page, and gives it a status of <i>1</i>. This means that the page has not yet been validated and must be processed by a task which applies <b>Validation</b> rules (a Recognition or Verification task, for example).</p>
See also...	SetPageType, SetDCOStatus, SetDCOType, SetDocumentStatus

## SetPageTemplateID

Actions Library	DCO.rra/.rrx
Description	Assigns a value to the <b>FingerprintID</b> property of the selected <b>Page</b> object of the Document Hierarchy.  <i>Alert!</i> The <b>SetPageTemplateID</b> action will create a <b>FingerprintID</b> property of the current <b>Page</b> object if it does not already exist.
Parameter	The <i>String</i> value of the Fingerprint ID.
Level(s)	Page only.
Returns	<i>True</i> if the ruleset with this action is applied at the Page level. Otherwise, <i>False</i> .
Example	<pre>WordFind(MQSW) SetPageTemplateID(1010)</pre> <p>In this sequence, if the WordFind action locates "MQSW" on the current page, and the <b>SetPageTemplateID</b> action assigns "1010" as the page's Fingerprint ID. This links the page to a fingerprint with a Fingerprint ID of "1010".</p>

## SetPageType

Actions Library	DCO.rra/.rrx
Description	<p>Similar to the <b>SetDCOType</b> action, but works at the <b>Page</b> or <b>Field</b> level.</p> <p>The action assigns the Page Type you enter as a parameter to the current <b>Page</b> object of the Document Hierarchy. You can also use a <b>Field</b> object's value to set the Page Type. (See the <b>Parameter</b> section below.)</p>
Parameters	<p>A <i>String</i> value representing the Page Type.</p> <p>You can also designate a field in a <b>Page</b> object's Data file, and use its text value to set the Page Type. Simply enter the name of a valid <b>Field</b> object and surround it with single quotes. For example: 'PageCode'.</p>
Level	Page and Field levels
Returns	<i>False</i> if there are no <b>Page</b> objects in the Page file, or if the parameter is invalid. Otherwise, <i>True</i> .
Example	<p>The <i>APT</i> application's kScan task assigns <i>Other</i> as the <b>Page Type</b> and 49 as the default <b>Page Status</b> for every successfully scanned image in the batch.</p> <p>The sequence below is an example of a rule that converts <i>Other</i> pages to <i>APT</i> pages, and assigns a Page Status to each:</p> <pre> <b>SetPageType(Invoice)</b> SetPageStatus(1) </pre> <p>This combination sets the page as an <i>APT</i> page, and gives it a status of <i>1</i>. This means that the page has not yet been validated and must be processed by a task which applies <b>Validation</b> rules (a RuleRunner Task, for example).</p>
See also.....	SetPageStatus, SetDCOStatus, SetDCOType

## Equalize Action

*Taskmaster's* **EqualizeUnbalancedImage** action – aka the **Equalize** action - helps resolve the unequal *vertical/horizontal* resolution of an image.

For example, the resolution of a faxed image may well be approximately **200 dpi** horizontally (*X*), but **100 dpi** vertically (*Y*). This disparity can compromise a task's ability to recognize values within the image, and to display clearly fields within the image, or the overall image.

The **Equalize** action resolves this disparity by enhancing the image so that both the vertical and horizontal resolutions are 200 dpi.

The table on the next page describes the **EqualizeUnbalancedImage** action.

- ✓ First, be sure to review these *important considerations*:
  - For Rulemanager, the **Equalize.rra** Actions file must reside in the **Scripts** sub-folder of your **Datacap** directory's **BPilot** folder. If you are using Datacap Studio, the **Equalize.rrx** Actions file has to be in the **RRX** sub-folder of your **Datacap** directory's **RRS** folder.
  - Equalize procedures are most effective right after the scanning of faxed images is complete. Therefore, the *rulerunner task* that applies a rule with this action will typically come *after* a Main job's vScan task but *before* the task that is responsible for recognition. Chapter 7 of the *Guide to Taskmaster Rules* explores the setup and operation of *rulerunner tasks* that apply the rules you define and assign.
  - This is a **singular action**: it is the only member of the **Equalize.rra** Actions file.

## EqualizeUnbalancedImage

Actions Library	Equalize.rra/.rrx
Description	<p>Resolves differences in the dpi (dots per inch) resolutions along the horizontal (X) and vertical (Y) planes of one ore more <i>faxed</i> images in a batch.</p> <p>The action selects images from the batch in response to the parameter you enter, and produces new images with <b>200 x 200 dpi</b>.</p>
Parameters	<p><i>Integer</i> value which determines the cut-off point for the resolution which should be equalized:</p> <p><code>EqualizeUnbalancedImage ( 0 )</code>, for example, specifies that there is no cut-off point: <i>all</i> images will be subject to equalization.</p> <p><code>EqualizeUnbalancedImage ( 20 )</code> establishes a cut-off point of 200(x)/180(y). This means that the action will equalize <i>all</i> images with this resolution ratio and more (200/180, 200/160 etc.) – but will ignore all images with balance ratios less than 200/180 (in this example.)</p> <p><b><i>Alert!</i></b> Standard Mode fax resolution in Dots per Inch (DPI) is 204/98; Fine Mode fax resolution is 204/196.</p>
Level	Page level only.
Returns	<i>False</i> if the parameter is not <i>numeric</i> or if the rule containing the action is not bound to a <b>Page</b> object of the Document Hierarchy. Otherwise, <i>True</i> .
Examples	<code>EqualizeUnbalanceImage(0)</code>

## Export Actions

The **Export** actions of an **Export** ruleset extract information from the Data file of the current *source* page, optionally combine it with information from other sources such as fixed text, and insert that information into an Export file. Along the way, certain **Export** actions may re-format the data before it is added to the Export file.

**Important!** **Export** actions and ruleset supply information to a **text file**. A separate set of actions (found in **ExportDB.rra/.rrx/rrx**) exports data to an Export **database**. (For explanations of **ExportDB** actions, see Page 79.) A third group – **ExportXML** actions – creates XML formats for the Export data (Page 90).

### BatchVariable\_ExportValue

Actions Library	Export.rra/.rrx
Description	Exports the value contained in the specified batch-level variable.
Parameter	The name of the Batch variable
Level	All
Returns	Always <i>True</i> .
Example	BatchVariable_ExportValue(ED)  This action will export the value located in the <i>ED</i> <b>Batch</b> variable to your Export file.

### BlankFields

Actions Library	Export.rra/.rrx
Description	Inserts <i>n</i> blank fields into the Export file, adjacent to the current field.  The fields are blank; other actions direct the Export task to fill the fields.
Parameter	A number <i>n</i> indicating how many blank fields to add to the Export file.
Level	All
Returns	Always <i>True</i> .
Example	BlankFields(12)

### BlankLines

Actions Library	Export.rra/.rrx
Description	Inserts <i>n-1</i> blank lines into the Export file.
Parameter	A number <i>n</i> indicating how many blank lines to add <i>below</i> the current line.
Level	All
Returns	Always <i>True</i> .
Example	BlankLines(5)  This action inserts <i>four</i> empty lines, leaving the insertion point for the next output on the fifth line. Additional output will begin on the fifth line.
See also...	NewLine

## BPilot

Actions Library	Export.rra/.rrx
Description	Exports the value assigned to the <i>Batch Pilot</i> property that you designate as the parameter.  <i>Alert!</i> A separate topic lists the <i>Batch Pilot</i> properties.
Parameter	The name of the <i>Batch Pilot</i> property whose value is to be included in the Export file. (A separate topic lists the properties you can include as parameters.)
Level	All
Returns	Always <i>True</i> .
Example	NewLine() Text(BatchID: ) <b>BPilot(BatchID)</b>  This sequence adds “ <b>BatchID:</b> ” followed by the current Batch ID into the Export file. For example:  BatchID: 20050019.001

### CloseExportFile

Actions Library	Export.rra/.rrx
Description	<p>Closes the currently opened Export file.</p> <p>This action usually belongs to its own RuleSet (<b>ExportClose</b>, for example), and applies to the <b>Batch</b> object of the Document Hierarchy. However, it can be used at any level.</p>
Parameters	None
Level	All
Returns	Always <i>True</i> .
Example	CloseExportFile()
See also...	OpenExportFile

### DCOProperty

Actions Library	Export.rra/.rrx
Description	<p>Exports the value assigned to the <i>DCO</i> property that you designate as the parameter.</p> <p><b>Alert!</b> A separate topic lists the <i>DCO</i> properties.</p>
Parameter	The name of the <i>DCO</i> property whose value is to be included in the Export file. (A separate topic lists the properties you can include as parameters.)
Level	All
Returns	Always <i>True</i> .
Example	<p>NewLine() Text(Document: ) <b>DCOProperty(ID)</b></p> <p>If this sequence is applied to a <b>Document</b> object, the Export file for document 01 in batch 20050219.057 will look like:</p> <p style="text-align: center;">Document: 20050219.057.01</p>
See also...	BPilot



## DocumentVariable\_ExportValue

Actions Library	Export.rra/.rrx
Description	Exports the value contained in the specified Document-level variable.
Parameter	Document Variable Name
Level	Document, Page, Field
Returns	Always <i>True</i> .
Example	<p>DocumentVariable_ExportValue(PageCount)</p> <p>A number of techniques can add variables to a <b>Document</b> object of the Document Hierarchy. These variables are listed as properties of the object in the <i>Document Hierarchy Setup</i> window.</p> <p>The example above reads the value assigned to the <b>PageCount</b> Document variable and places it in the Export file.</p>

## ExportAllFields

Actions Library	Export.rra/.rrx
Description	Exports all field values in the current page. The action does not export detail line fields.
Parameter	None
Level	Page level only.
Returns	<i>False</i> if the action is not used at the Page level. Otherwise, <i>True</i> .
Example	ExportAllFields()

### ExportFieldValue

Actions Library	Export.rra/.rrx
Description	Exports the specified <b>Field</b> object's value to the Export file.
Parameter	The name of the <b>Field</b> object whose value you want to export.
Level	Page level only.
Returns	<i>False</i> if the parameter is not a <b>Field</b> object's name. Otherwise, <i>True</i> .
Example	ExportFieldValue(Date) ExportFieldValue(Number) ExportFieldValue(Total)  This sequence exports the <i>current</i> values stored in the <i>Date</i> , <i>Number</i> and <i>Total</i> fields.

### ExportMYValue

Actions Library	Export.rra/.rrx
Description	Exports the current field value to the Export file.
Parameters	None
Level	Field level only.
Returns	Always <i>True</i> .
Example	ExportMYValue()

### ExportSmartParameter

Actions Library	ExportDB.rra/.rrx
Description	Exports an evaluated smart parameter value to the Export file.
Parameters	None
Level	Field level only.
Returns	Always <i>True</i> .
Example	ExportSmartParameter(@P\DueDate.Expired)  The example will export the value of a variable named <i>Expired</i> on field <i>DueDate</i> which is a child field on the parent page (@P) of the calling node.

## ExportToBatchDir

Actions Library	Export.rra/.rrx
Description	Sets the path for the Export file to the current Batch Directory.
Parameters	None
Level	All
Returns	<i>False</i> , if the Batch Directory is not accessible. Otherwise, <i>True</i> .
Example	ExportToBatchDir()

## Filler

Actions Library	Export.rra/.rrx
Description	Adds filler to the Export value of the field represented by the bound <b>Field</b> object of the Document Hierarchy.
Parameters	Two comma-separated values.  The first is a number indicating the total length in characters of the filler; the second is the filler's character. This character is optional: if you do not enter a value, the action will use the most recent Global character setting.
Level	Field only.
Returns	Always <i>True</i> .
Example	Filler(12,n)  The action in the example fills the current field with 12 instances of "n".

## FixedLenLJ

Actions Library	Export.rra/.rrx
Description	Indicates how much right-justified data is to be exported from a specified field.
Parameter	Two comma-separated values: <ol style="list-style-type: none"><li>1. The field's name: this is the name of the corresponding <b>Field</b> object of the Document Hierarchy.</li><li>2. The number of characters that are to be exported...counting from the field's left end.</li></ol>
Level	Field.
Returns	False if the action cannot identify left-justified characters, or if the field is empty. Otherwise, True
Example	FixedLenLJ(Volume,8)

## FixedLenRJ

Actions Library	Export.rra/.rrx
Description	Similar to the FixedLenLJ action, this action exports a specified number of characters from a field's right end (right-justified.)
Parameter	Two comma-separated values: <ol style="list-style-type: none"><li>1. The field's name: this is the name of the corresponding <b>Field</b> object of the Document Hierarchy.</li><li>2. The number of characters that are to be exported...counting from the field's right end.</li></ol>
Level	Field.
Returns	False if the action cannot identify right-justified characters, or if the field is empty. Otherwise, True
Example	FixedLenRJ(InvoiceDate,12) This example exports 12 characters from the right end of the <i>InvoiceDate</i> field.

**GetDATE**

Actions Library	Export.rra/.rrx
Description	Exports today's Date in a format you specify as the parameter.
Parameter	<p>The Date's format.</p> <p>“*” stipulates the default <i>mm/dd/yyyy</i> construction. However, you can combine any of the following <i>String</i> values to define a different format:</p> <p>d = day of the month, 1-31  dd = two-digit day, 01-31  yyyy = four-digit year  yy = two-digit year  m = month, 1-12  mm = two-digit month, 01-12  ccyy = four-digit year  y = Julian day of the year</p> <p>”.” and “/” are valid separators</p>
Level	All
Returns	Always <i>True</i> .
Example	<p><b>GetDate( *)</b> inserts today's date into the Export file with this format:</p> <p>11/16/2005</p>
See also...	GetTime

## GetProfileString

Actions Library	Export.rra/.rrx
Description	Accesses a Settings file (.ini) and adds a value in that file to your Export file.  <b>Important:</b> The action assumes that the Settings file resides in the <b>Batches</b> directory.
Parameters	1.) The [Section] within the Settings file. 2.) The Key entry within the section, with the value you want to retrieve. 3.) The name of the Settings file.
Level	All
Returns	<i>True</i> if the Settings file designate by the parameter exists. Otherwise, <i>False</i> .
Example	GetProfileString(General,MyValue,Batch.ini)

## GetTime

Actions Library	Export.rra/.rrx
Description	Exports the current Time in a format you specify as the parameter.
Parameter	<p>The Time's format.</p> <p>“*” stipulates the default HH:MM:SS construction. However, you can combine any of the following <i>String</i> values to define a different format:</p> <ul style="list-style-type: none"> <li>m = minute 1-59</li> <li>s = second 1-59</li> <li>h = 1-23</li> <li>yy = two-digit year</li> <li>mm = two-digit minute, 01-59</li> <li>ss = two-digit second, 01-59</li> <li>hh = two-digit hour, 01-23</li> </ul> <p>”:” is the valid separator.</p>
Level	All
Returns	Always <i>True</i> .
Example	<p><b>GetTime(*)</b> inserts the current time into the Export file with this format:</p> <p>07:08:16</p>
See also...	GetDate

## LineItem\_AddElement

Actions Library	Export.rra/.rrx
Description	<p>Includes the Line Item <b>Field</b> object you specify as an element of a Line Item Array.</p> <p>A Line Item Array accumulates and organizes <i>captured</i> line item values retrieved from the Data file of a particular page.</p> <p><b>Important!</b> A rule using this action should be applied to the <i>parent DETAILS</i> object of the Document Hierarchy.</p> <p>This action is used for exporting Line Item values.</p>
Parameter	The name of the <i>child Field</i> object of the Document Hierarchy (DETAILS, in the case of the <i>APT</i> application, is an example.)
Level	The <i>parent</i> field that contains the <i>child</i> Line Item field. An example is the <i>DETAILS</i> field in the <i>APT</i> application.
Returns	Always <i>True</i> .
Example	<p>LineItem_AddElement(Price)</p> <p>The action above expands the Line Item Array by one field: <b>Price</b>.</p> <p>LineItem_ExportElements will populate this element and other elements of the array with the <i>captured</i> values it finds in a page's Data file before exporting them.</p>
See also...	LineItem_ExportElements, LineItem_ClearElements



## LineItem\_BlankFields

Actions Library	Export.rra/.rrx
Description	Includes the specified number of blank fields as elements of a Line Item Array.
Parameter	The number of blank fields to export as part of the Line Item Array.
Level	The <i>parent</i> field that contains the <i>child</i> Line Item field. An example is the <i>DETAILS</i> field in the <i>APT</i> application.
Returns	Always True.
Example	<p>This action includes the specified number of blank Line Item fields as elements of a Line Item Array.</p> <p><b>Note:</b> A Line Item Array accumulates and organizes captured line item values retrieved from the Data file of a particular page.</p> <p><b>Important!</b> A rule using this action should be applied to the parent <i>DETAILS Field</i> object of the Document Hierarchy.&lt;br/&gt;&lt;br/&gt;</p> <p>LineItem_BlankFields(6)</p> <p>The action above expands the Line Item Array by six blank fields.</p>

## LineItem\_ClearElements

Actions Library	Export.rra/.rrx
Description	Clears the Line Item Array of its values.
Parameters	None
Level	The <i>parent</i> field that contains the <i>child</i> Line Item field. (DETAILS, in the case of the <i>APT</i> application, is an example.)
Returns	Always <i>True</i> .
Example	<p>LineItem_ClearElements()</p> <p>This is a housekeeping function you can use to be sure the array does not contain leftover values.</p>
See also...	LineItem_ExportElements

### LineItem\_ExportElements

Actions Library	Export.rra/.rrx
Description	Exports the <i>captured</i> values in a page's Line Item Array that have been populated with LineItem_AddElement actions.
Parameters	None
Level	The <i>parent</i> field that contains the <i>child</i> Line Item sub-fields. Example: the <i>LINEITEM</i> field in the <i>APT</i> application.
Returns	Always <i>True</i> .
Example	LineItem_ExportElements()  This example exports the values included in the Line Item Array to your Export file.
See also...	LineItem_ClearElements

## LineItem\_SmartParameter

Actions Library	Export.rra/.rrx
Description	<p>Adds a smart parameter algorithm as an element of a Line Item Array.</p> <p>A Line Item Array accumulates and organizes captured line item values retrieved from the Data file(.xml) of a particular page.</p> <p><b>Important!</b> A rule using this action should be applied to the parent DETAILS object of the Document Hierarchy.</p> <p>This action is used for exporting Blank Line Item values.</p>
Parameter	Smart Parameter to be evaluated during processing of the Line Item array.
Level	The <i>parent</i> field that contains the child Line Item field. An example is the <i>DETAILS</i> field in the <i>APT</i> application.
Returns	Always <i>True</i> .
Example	<p><b>LineItem_SmartParameter(@F\Price+@TIME(HH:MM:SS))</b></p> <p>The action above places a child field <i>Price</i> of the calling field node (<i>@F</i>) appended with the current time in format <i>HH:MM:SS</i> to the Export file.</p> <p><b>LineItem_ExportElements</b> will populate this element and other elements of the array with the captured values it finds in a page's Data file before exporting them.</p>
See also...	LineItem_ExportElements, LineItem_ClearElements, LineItem_AddElements, LineItem_BlankFields

### NewLine

Actions Library	Export.rra/.rrx
Description	Starts a new line in your Export file.
Parameter	None
Level	All
Returns	Always <i>True</i> .
Example	<b>NewLine()</b> Text(Export Output)  This sequence starts a new line and adds “Export Output” to the beginning of the line.
See also...	BlankLines

### PageVariable\_ExportValue

Actions Library	Export.rra/.rrx
Description	Exports runtime values assigned to a variable of the bound <b>Page</b> object of the Document Hierarchy.
Parameter	<i>String</i> value of the variable's name.
Level	Page.
Returns	False if the variable does not exist. Otherwise, True.
Example	PageVariable_ExportValue(TemplateID)

## ResetFieldVariables

Actions Library	Export.rra/.rrx
Description	Resets the variables of the bound <b>Field</b> object of the Document Hierarchy.
Parameter	None.
Level	Field.
Returns	Always True.
Example	<p>ResetFieldVariables()</p> <p><b>Note:</b> This action's procedure assigns "blank" to the <b>Fixed Length</b> variable; "Left" to the <b>Justified</b> variable; "1" to the <b>SetFill</b> variable; and no value to the <b>IgnoreFieldStatus</b> variable.</p>

## SetFilePathAsVariable

Actions Library	Export.rra/.rrx
Description	Saves the path and name of your Export file to the variable specified by the parameter. Note: If the variable does not exist, it will be created.
Parameter	<i>String</i> value of the name of the variable that will store the Export file's name and its path..
Level	All
Returns	Always <i>True</i> .
Example	SaveFilePathAsVariable(Export_File)

### SetCSV

Actions Library	Export.rra/.rrx
Description	Ensures that all <i>exported</i> values are delimited by a comma separator.
Parameter	<i>True</i> enables the action; <i>False</i> disables it.
Level	All
Returns	Always <i>True</i> .
Example	<b>SetCSV(True)</b> ExportFieldValue(Date) ExportFieldValue(Number) ExportFieldValue(Total) <b>SetCSV(False)</b>  This sequence will export the <i>captured</i> values of the <i>Date</i> , <i>Number</i> , and <i>Total Field</i> objects into your Export file. A comma will be added after each value to separate the fields.
See also...	SetElementSeparator

### SetElementSeparator

Actions Library	Export.rra/.rrx
Description	Ensures that all <i>exported</i> values are delimited by a separator you designate as the parameter.
Parameter	A field separator character – or a <i>String</i> value of “Off” or “False” to disengage the action..
Level	All
Returns	Always <i>True</i> .
Example	SetElementSeparator()  This action uses “ ” to delimit the Export file’s values.  SetElementSeparator(Off)  turns off the action.
See also...	SetCSV

## SetExportPath

Actions Library	Export.rra/.rrx
Description	Specifies the path to the Export file's location.
Parameter	The <i>complete</i> path to the <b>Export</b> folder.
Level	All
Returns	<i>True</i> if the path specified by the parameter exists. Otherwise, <i>False</i> .
Example	SetExportPath(C:\ParentDir\Invoice\Export)

## SetExtensionName

Actions Library	Export.rra/.rrx
Description	Assigns an extension to the current Export file. The default value is <b>.txt</b> .
Parameter	The file extension you want to use.
Level	All
Returns	Always <i>True</i> .
Example	SetExtensionName(.dat) In this example, the Export file will have a <b>.dat</b> extension.
See also...	SetFileName

### SetFileName

Actions Library	Export.rra/.rrx												
Description	Assigns a name to the current Export file.												
Parameter	<p>The file's name (without an extension).</p> <p>Using commas as delimiters, you can assemble a name with the variables below:</p> <table><tr><td>@BatchID</td><td>Batch ID</td></tr><tr><td>@ID</td><td>ID, current object</td></tr><tr><td>@Status</td><td>Status, current object</td></tr><tr><td>@Type</td><td>Type, current object</td></tr><tr><td>@Date</td><td>Today's date</td></tr><tr><td>@Time</td><td>Current time</td></tr></table>	@BatchID	Batch ID	@ID	ID, current object	@Status	Status, current object	@Type	Type, current object	@Date	Today's date	@Time	Current time
@BatchID	Batch ID												
@ID	ID, current object												
@Status	Status, current object												
@Type	Type, current object												
@Date	Today's date												
@Time	Current time												
Level	All												
Returns	Always <i>True</i> .												
Examples	<p>SetFileName(Export_,@BatchID) SetExtensionName(.txt)</p> <p>This sequence establishes a series of Export files with names such as</p> <p>Export_20021231.001.txt</p> <p>Export_20021231.002.txt</p> <p>In contrast,</p> <p><b>SetFileName(@BatchID)</b> SetExtensionName(.txt)</p> <p>will establish a series of Export files with Batch IDs only:</p> <p>20021231.001.txt</p> <p>20021231.002.txt</p>												
See also...	SetExtensionName												



## SetFilePathAsVariable

Actions Library	Export.rra/.rrx
Description	Saves the path and name of your Export file to the variable specified by the parameter
Parameter	Variable name specifying where the Export file name and path will be stored
Level	All
Returns	Always True
Example	This action saves the path and name of your Export file to the variable specified by the parameter. If the variable does not exist, it will be created.  <b>SaveFilePathAsVariable(Export_File)</b>

## SetFill

Actions Library	Export.rra/.rrx
Description	Sets a single <b>character</b> as the global filler value.
Parameter	Single <i>String</i> character to be used as the global filler value.
Level	Any
Returns	<i>False</i> if more than one character is used as a parameter. Otherwise, <i>True</i> .
Example	<b>SetFill(\$)</b>

## SetFixedLength

Actions Library	Export.rra/.rrx
Description	Uses the <i>Numeric</i> value you enter as a parameter to establish a fixed length of a value exported from the current field.
Parameter	<i>Numeric</i> value indicating the field's length.
Level	Field.
Returns	False if the parameter is not <i>Numeric</i> ; Otherwise, True.
Example	SetFixedLength(12)

## SetIgnoreFieldStatus

## Export Actions

---

Actions Library	Export.rra/.rrx
Description	Directs the task that processes a rule with this action to ignore the STATUS value assigned to a field represented by the bound <b>Field</b> object of the Document Hierarchy.
Parameter	<i>Numeric</i> value of the Field status to be ignored..
Level	Field.
Returns	False if the parameter is not <i>Numeric</i> ; Otherwise, True.
Example	SetIgnoreFieldStatus(1)

### SetJustified

Actions Library	Export.rra/.rrx
Description	Right-justifies or left-justifies a field's exported values according to the parameter you enter.
Parameter	An upper case R(right-justified) or L(left-justified). <i>Alert!</i> In either case, the code must be preceded by a space.
Level	Field.
Returns	False if the parameter is not upper case, or if it is not preceded by a space. Otherwise, True.
Example	SetJustified( R)

### SetOMR\_Separator

Actions Library	Export.rra/.rrx
Description	For multi-punch OMR fields, uses the parameter's value as the separator character.
Parameter	The separator character you want to use.
Level	All
Returns	Always <i>True</i> .
Example	SetOMR_Separator(;

## SetSpaceFill

Actions Library	Export.rra/.rrx
Description	Specifies the use of the ASCII 32 <b>space</b> as the global filler value.
Parameter	None
Level	Any
Returns	Always <i>True</i>
Example	<p><b>SetSpaceFill()</b></p> <p><b>Note</b> that the action’s syntax does not include a value inside the parentheses that surround the parameter. Instead, the parentheses enclose an ASCII 32 “space.”</p>

## SetZeroFill

Actions Library	Export.rra/.rrx
Description	Sets the use of the ASCII 48 <b>zero</b> as the global filler value.
Parameter	None
Level	All
Returns	Always True.
Example	<p><b>SetZeroFill</b></p> <p><b>Note</b> that the action’s syntax does not include any parentheses whatsoever. This apparent gap is intentional because it specifies the use of the ASCII 48 “zero” filler.</p>

## Text

Actions Library	Export.rra/.rrx
Description	Specifies the path to the Export file’s location.
Parameter	The <i>complete</i> path to the <b>Export</b> folder.
Level	All
Returns	<i>True</i> if the path specified by the parameter exists. Otherwise, <i>False</i> .
Example	SetExportPath(C:\ParentDir\Invoice\Export)

### Variable\_ExportValue

Actions Library	Export.rra/.rrx
Description	Exports the value assigned to a variable of the current object of the Document Hierarchy.
Parameter	The name of the variable with the value you want to export.
Level	All
Returns	Always <i>True</i> .
Example	Variable_ExportValue(ID)  This action exports the value assigned to the <b>ID</b> property of the current object of the Document Hierarchy.

### Variable\_IsValue

Actions Library	Export.rra/.rrx
Description	Checks to see if the parameter value matches the value assigned to a variable of the current object of the Document Hierarchy.
Parameters	1.) The name of the variable with the value you want to compare. 2.) The value you want to match with the variable's value.
Level	All
Returns	<i>True</i> if the variable's value matches the parameter's value. Otherwise, <i>False</i> .
Example	Variable_IsValue(Invoice,Yes)  This action returns <i>True</i> if the value of the current <b>Page</b> object's <b>Invoice</b> variable is "Yes".

### **Batch Pilot Properties (BPilot Action)**

The **BPilot** action can use any property of the **Batch Pilot** object as a parameter, including those listed below:

BatchDir	The name and location of the application's <b>Batches</b> directory.
BatchID	The Batch Number of the current batch ( <i>20020072.003</i> , for example.)
JobName	The name of the current <i>Taskmaster</i> job ( <i>Main</i> , for example.)
Operator	The User ID of the operator currently processing the batch.
PagesInBatch	A count of all pages in the batch.
DocsInBatch	A count of all documents in the batch. <b>Remember:</b> in most configurations, a Recognition task re-organizes a batch into a series of documents and their pages.
Priority	The processing priority assigned to the current batch ("10" = Low, "1" = High, "5" = Default). A task selects batches from its queue first according to <b>Priority</b> .
Station	The Station ID of the workstation currently processing the batch.
TaskName	The name of the task with the batch in its queue.
XtraBatchFieldValue	The value in a custom field you've added to the <b>Job Monitor's</b> Batch Information Table.

### ***Document Hierarchy Properties***

The **DCO** action can use any property of the **DCO** Object as a parameter, including those listed below:

ID	The value of an object's <b>ID</b> property. For a <b>Batch</b> object, this might be <i>20020072.003</i> . You can apply this action at any level(s).
ImageName	The name and location of a <b>Page</b> object's Image file: <i>C:\ParentDir\APT\batches\20030145.001\TM000001.tif</i> , for example.
Status	The value assigned to an object's <b>Status</b> property. You can apply this action at any level(s).
Type	The value assigned to an object's <b>Status</b> property. You can apply this action at any level(s).

## ExportDB Actions

These actions move a batch's verified and validated information from the Page and Data files directly to a target database.

An **ExportDB** rule can work with a **Page** object of the Document Hierarchy, or a **Field** object:

- A rule linked to a **Page** object uses **ExportFieldToColumn** actions to store *captured* field values to a database column.
- A rule linked directly to a **Field** object typically uses an **ExportToColumn** action to store that the *captured* value for that **Field** object only.

## AddRecord

Actions Library	ExportDB.rra/.rrx
Description	<p>Inserts <i>assembled</i> data into the database table specified by a previous <b>SetTableName</b> action.</p> <p><i>Alert!</i> This action runs after earlier actions gather data, open the database, and access the correct table (see the example).</p>
Parameters	None
Level	All, but generally used at the Page or Field level.
Returns	<p><i>False</i> if there is no connection to the database; if an error occurs when the action attempts to add the record to the database; or if a <b>SetTableName</b> action was not previously used.</p> <p>Otherwise, <i>True</i>.</p>
Example	<pre>SetTableName(Invoice) ExportFieldToColumn   (VendorID,db_Vendor) ExportFieldToColumn   (Number,db_Number) ExportFieldToColumn   (Total,db_Total) <b>AddRecord()</b></pre> <p>This <b>ExportDB</b> rule applies to a <b>Page</b> object of the Document Hierarchy.</p> <p>The actions open the database and direct the rule's attention to the <b>Invoice</b> table. The rule then sets up a record with three values – <b>Vendor ID</b>, <b>Invoice Number</b> and <b>Total</b>. Afterwards, the <b>AddRecord</b> action updates the table with new information.</p>
See also...	SetTableName, ExportFieldToColumn



## CloseConnection

Actions Library	ExportDB.rra/.rrx
Description	<p>Closes an open connection to your Export database.</p> <p>Usually, this action is placed in a RuleSet –that is separate from the RuleSet that opens the connection and stores the data.</p> <p><b>ExportDBClose</b> RuleSet, for example, is run at the Batch level after all data has been exported from the batch to the specified database.</p>
Parameters	None
Level	All, but generally used as part of a separate RuleSet at the Batch level.
Returns	<i>False</i> if the connection is already closed. Otherwise, <i>True</i> .
Example	<p>CloseConnection()</p> <p>This action closes the previously opened connection to the Export database.</p> <p>This action is usually part of a separate RuleSet that prevents the need to repeatedly open the connection to the Export database. (You can open the connection once in the first RuleSet, export data from all documents and pages in the batch, then close the connection once in the second RuleSet.)</p>
See also...	OpenConnection

## ExportBatchIDToColumn

Actions Library	ExportDB.rra/.rrx
Description	Exports the current Batch ID to the database column specified by the parameter.
Parameters	Two parameters: <ol style="list-style-type: none"><li>1) The name of the column you are exporting to.</li><li>2) <b>Optional:</b> Setting the second parameter to “1” will cause the action to fail if the first parameter is invalid.</li></ol>
Level	All, but generally at the Page or Field level.
Returns	<i>False</i> if the second parameter is set to “1” and the parameter does not identify a valid database column. Otherwise, <i>True</i> .
Example	SetTableName(Export_Results) <b>ExportBatchIDToColumn(db_BatchID,1)</b> ExportFieldToColumn(Date,db_Date) ExportFieldToColumn(Total,db_Total) AddRecord()
See also...	AddRecord, ExportFieldToColumn, ExportPropertyToColumn

## ExportFieldToColumn

Actions Library	ExportDB.rra/.rrx
Description	<p>A <i>page-level action</i> that extracts the <i>captured</i> value of a <b>Field</b> object from the Data file of the current page, and specifies its target location within a table of the Export database.</p> <p>Within a rule, this action should run before an <b>AddRecord</b> action, which commits the data to the database.</p>
Parameters	<ol style="list-style-type: none"> <li>1.) The name of the <b>Field</b> whose value you want to export.</li> <li>2.) The name of the <b>Column</b> in your database table where you want to place the data.</li> </ol>
Level	Page level only.
Returns	<p>False if:</p> <ol style="list-style-type: none"> <li>1.) There is no connection to the database; or</li> <li>2.) The database column specified as a parameter does not exist.</li> <li>3.) The <b>Field</b> object identified by the parameter does not exist.</li> <li>4.) A <b>SetTableName</b> action was not used previously.</li> </ol> <p>Otherwise, <i>True</i>.</p>
Example	<pre>SetTableName(Export_Results) ExportFieldToColumn(VendorID,db_Number) ExportFieldToColumn(Date,db_Date) ExportFieldToColumn(Total,db_Total) AddRecord()</pre> <p>This action exports the <i>captured</i> value of three <b>Field</b> objects from the Data file of the current page, to corresponding columns of the <b>Export_Results</b> table, within the <b>InvoiceExport</b> database.</p> <p><b>Important!</b> Make sure you use the <b>OpenConnection</b> action to establish a connection to your Export database. This is usually accomplished by a rule at the Batch level.</p>
See also...	SetTableName, ExportToColumn

## ExportNodeXMLToColumn changed

Actions Library	ExportDB.rra/.rrx
Description	Exports the value of the XML property of the bound object (node) of the Document Hierarchy to a column of the Export database.
Parameters	Two comma-separated values:  1.) The smart parameter path to the bound object of the Document Hierarchy. This object's <b>XML</b> property will be added to a designated column of the Export database.  2.) The <i>String</i> value of the name of the target column in the Export database. The action will add the value of the calling object's <b>XML</b> property to this column.
Level	All, but generally at the Batch level.
Returns	False if:  1.) There is no connection to the database.  2.) The column identified by the parameter does not exist.  3.) A <b>SetTableName</b> action was not previously used.  4.) The smart parameter path does not point to a valid object of the Document Hierarchy
Example	<b>ExportNodeXMLToColumn(@P\MyField.MYDBCOLUMN)</b>

## ExportPropertyToColumn

Actions Library	ExportDB.rra/.rrx
Description	Adds the value of a property (variable) of the selected object to a column of the Export database
Parameters	1.) <i>String</i> value of the property's name 2.) <i>String</i> value of the column's name
Level	All, but generally used at the Page or Field level.
Returns	False if: 1.) There is no connection to the export database. 2.) The column of the database does not exist. 3.) The property (variable) identified by the parameter does not exist. 4.) A <b>SetTableName</b> action was not used previously. Otherwise, <i>True</i> .
Example	<pre>SetTableName(Export_Results) ExportFieldToColumn(VendorID,db_Number) <b>ExportPropertyToColumn(Status,db_PageStatus)</b> ExportFieldToColumn(Total,db_Total) AddRecord()</pre> <p>This sequence updates the <b>db_PageStatus</b> column of the <b>Export_Results</b> table with the value of the selected object's <b>Status</b> property.</p>
See also...	ExportToColumn, ExportFieldToColumn, ExportBatchIDToColumn

## ExportSmartParamToColumn

Actions Library	ExportDB.rra/.rrx
Description	Adds the evaluated value of a smart parameter to a column of the Export databaset Hierarchy and assign them to a column of the Export database.
Parameters	1.) A smart parameter that locates the value to be assigned to the <b>Text</b> property of the bound object (aka, the "calling" object) of the Document Hierarchy.  2.) The <i>String</i> value of a column name of the Export database. The action will add the <b>Text</b> property to this table.
Level	All.
Returns	<i>False</i> if:  1.) There is no connection to the database.  2.) The column identified by the parameter does not exist.  3.) A <b>SetTableName</b> action was not previously used.  Otherwise, <i>True</i> .
Example	<code>ExportSmartParamToColumn(@P\MyField,EXPDBCOLUMN)</code>

## ExportToColumn

Actions Library	ExportDB.rra/.rrx
Description	A <i>field-level action</i> that exports the <i>captured</i> value of the current <b>Field</b> object from the page's Data file to a target column within a previously-designated table of an open Export database.
Parameters	The name of the column in the database table where you want to place the data.
Level	Field level only.
Returns	False if: <ol style="list-style-type: none"> <li>1.) There is no connection to the database.</li> <li>2.) The column identified by the parameters does not exist.</li> <li>3.) A <b>SetTableName</b> action was not previously used.</li> </ol> Otherwise, <i>True</i> .
Example	<p>Batch Level:</p> <pre>OpenConnection(InvoiceExport) SetTableName(Export_Results)</pre> <p>Current field:</p> <pre>ExportToColumn(db_Date)</pre> <p>Last Field:</p> <pre>AddRecord( )</pre> <p>This example exports the <i>captured</i> value of the <b>Field</b> object to which the rule applies, from the Data file of the current page to the <b>db_Date</b> column of the <b>Export_Results</b> table, within the <b>InvoiceExport</b> database.</p>
See also...	ExportFieldToColumn, ExportPropertyToColumn

## OpenConnection

Actions Library	ExportDB.rra/.rrx
Description	Opens a connection to the database you specify as the parameter.  A rule containing this action can apply to any object of the Document Hierarchy, but it is most often used at the batch level.
Parameter	The Data Source Name (DSN) or Connection String of the target Export database.  <i>Alert!</i> If the action is establishing a connection with an Oracle database, or a SQL Server database using SQL Server Authentication, be sure to expand the DSN parameter by adding the correct Provider, user ID and Password. For example:  <b>Oracle →</b> OpenConnection(PROVIDER=ODBCORACLE;DSN=1040Look;CATALOG=;DBNTA=;UID=Admin;PWD=Admin;)  <b>SQL Server Authentication →</b> OpenConnection(PROVIDER=ODBCMSSQL;DSN=1040Look;CATALOG=;DBNTA=;UID=Admin;PWD=Admin;)
Level	All, but generally used at the Batch level.
Returns	<i>True</i> if the connection opens. Otherwise, <i>False</i> .
Example	OpenConnection(InvoiceExport)  <i>Alert!</i> This action must come first, before any other ExportDB actions.
See also...	CloseConnection



## SetTableName

Actions Library	ExportDB.rra/.rrx
Description	Sets the name of the table in your database to which the data is to be exported. This action needs to be used before the <b>AddRecord</b> action.
Parameter	Two parameters: <ol style="list-style-type: none"> <li>1) The name of the table you are exporting to.</li> <li>2) <b>Optional:</b> Setting the second parameter to “1” will cause the action to fail if the first parameter is invalid.</li> </ol>
Level	All
Returns	<i>False</i> if the second parameter is set to “1” and the parameter does not identify a valid table name. Otherwise, <i>True</i> .
Example	<b>SetTableName(Export_Results,1)</b> ExportFieldToColumn(MyDate, db_Date) AddRecord()
See also...	AddRecord

## ExportXML Actions

The **ExportXML** actions “wrap” their values in XML syntax before adding them to an Export file or an Export database.

### XML\_CommitNode

Actions Library	ExportXML.rra/.rrx
Description	Commits the XML node specified by the parameter's Tag ID.
Parameters	<i>String</i> value of the XML Tag.
Level	All
Returns	Always True.
Example	<p>This action commits (closes) the current xml node with the XML tag value of the parameter. The action allowing a new node with the same tag to be created at the same hierarchical level in the <b>output xml</b> file.</p> <pre>xml_CommitNode(LineTotal)</pre> <p>The example above (closes) the current XML node with a Tag: <i>LineTotal</i>.</p>

**XML\_NewNode**

Actions Library	ExportXML.rra/.rrx
Description	Adds a new node as a <i>child</i> element of an existing node in the Export XML file.
Parameters	Comma-separated <i>String</i> values of::  <ol style="list-style-type: none"> <li>1. the NodeID (tag name) of a new <i>child</i> node; and...</li> <li>2. the NodeID (tag name) of the <i>parent</i> node.</li> </ol>
Level	Document, Page or Field.
Returns	True if the parent node exists. False if a duplicate root node is declared or parent Node ID does not exist.
Example	<p>The new NodeID followed by the parent NodeID creates a new Node in the Export XML file.</p> <p>The first <b>xml_NewNode</b> action in a ruleset will create the root node of the XML file using new child NodeID, leaving the parent NodeID blank.</p> <pre>xml_NewNode(ClaimsData,HCFA)</pre> <p>Duplicated Node ID's will cause the prior node of the same ID to commit to its specific parent node and will no longer be available for modification.</p> <p><b>Note:</b> Adding a second child NodeID with the same Tag name of the root node will cause the action to return false.</p>

**xml\_SaveFile**

Actions Library	ExportXML.rra/.rrx
Description	Commits all unsaved nodes and saves the XML file to disk.
Parameters	None
Level	All
Returns	True if the file is created successfully. Otherwise, False.
Example	xml_SaveFile()

### xml\_SetAttributeValue

Actions Library	ExportXML.rra/.rrx
Description	Assigns attributes to a specific node.
Parameters	Three comma-separated values: <ol style="list-style-type: none"><li>1. The NodeID.</li><li>2. The attribute's name.</li><li>3. The smart parameter value to be assigned to the attribute.</li></ol>
Level	All.
Returns	False if the node does not exist. Otherwise, True.
Example	The example below assigns the current Page's <i>Number</i> field value to the <b>Xpage</b> node's <b>Number</b> attribute.  xml_SetAttributeValue(Xpage,Number,@P\Number)

### xml\_SetExportPath

Actions Library	ExportXML.rra/.rrx
Description	The full path to the directory or folder that will contain the output file – without the file's name.
Parameters	The full path to the directory or folder that will contain the output file – without the file's name. <b>Note:</b> the parameter defaults to the <b>Batch</b> directory.
Level	All
Returns	False if the path is not valid. Otherwise, True.
Example	xml_SetExportPath()  In this case, the action will automatically add the output file to the current batch.

**xml\_SetFileName**

Actions Library	ExportXML.rra/.rrx
Description	<p>Assigns a filename to the XML output file.</p> <p><b>Note:</b> as a default, the action assigns the current Batch ID with an “.xml” extension.</p>
Parameters	<p><i>String</i> value of the output file’s filename.</p> <p>Using commas as delimiters, you can assemble a name with the variables below:</p> <p>@BatchID    Batch ID  @ID            ID, current object  @Status      Status, current object  @Type        Type, current object  @Date        Today’s date  @Time        Current time</p>
Level	All
Returns	Always True.
Example	xml_SetFileName(@BatchID)
See also...	SetFileName action of the Export.rra/.rrx file.

### xml\_SetNodeValue

Actions Library	ExportXML.rra/.rrx
Description	Assigns a value to a specific node.
Parameters	Two comma-separated <i>String</i> values: <ol style="list-style-type: none"><li>1. The NodeID (tag name).&gt;</li><li>2. The smart parameter value to be assigned to the node.</li></ol>
Level	All
Returns	False if the node does not exist. True otherwise.
Example	<p>The example below shows the action assigning a smart parameter string value of <i>To Infinity and Beyond!</i> to a node with a tag name of <b>Rocket</b>.</p> <pre>xml_SetNodeValue(Rocket,To Infinity and Beyond!)</pre> <p><b>Note:</b> the action defaults to the <b>CurrentObj</b> value if no value is assigned.</p>

## Grayscale Action

This action makes it possible to convert **Grayscale** Image files (.tif) to **Black and White** Image files (.tif).

### ConvertGraytoBW

Actions Library	Grayscale.rra/.rrx
Description	Converts Grayscale TIFF files into Black and White TIFF files. The action will rename the original Grayscale image using the same base filename, but replacing the .tif filename extension with “.tis”.
Parameters	None
Level	Batch, Document, or Page level.
Returns	<i>True</i> if the conversion is successful; otherwise <i>False</i> .
Example	ConvertGraytoBW()

## icr\_c Actions

### RecognizeFieldICR\_C

Actions Library	icr_c.rra/.rrx
Description	A field-level action that retrieves a <i>zoned</i> field’s settings from the <i>ICR/C</i> tab of the <i>Recognition Options Setup</i> dialog, and uses these settings to recognize the field’s value.
Parameters	None
Level	Field level only.
Returns	<i>False</i> if the action is not run at the Field level or if recognition fails. Otherwise, <i>True</i> .
Example	TaxpayerSSN Rule 1 <b>RecognizeFieldICR_C()</b>  In this example, the rule uses this action to retrieve and apply settings from the <i>ICR/C</i> tab of the <i>Recognition Options Setup</i> dialog...settings that have been previously assigned to a Document Hierarchy’s <i>zoned TaxpayerSSN Field</i> object.

## RecognizeFieldVoteICR\_C

Actions Library	icr_c.rra/rrx
Description	<p>A field-level action that initiates a voting procedure that first uses specifications in the <i>ICR/C</i> tab of the <b>Recognition Options Setup</b> dialog to recognize the field's <b>characters</b>.</p> <p>When this action stores the results of recognition, it first determines if the corresponding <b>Field</b> object of the Document Hierarchy contains a value.</p> <p>If a value is present, the action compares the existing value with the field's recognition results - character by character. If a character's values match, the Confidence Rating for the character is raised to the maximum level. If the values do <i>not</i> match, the Confidence Rating for the character is lowered to the minimum.</p> <p>Note that when using this voting procedure, the second Recognition engine is <i>secondary</i> and its results are <i>never</i> assigned. Instead, the action changes the Confidence Ratings on the basis of results provided by the first Recognition engine.</p> <p>If there are no recognition results before the action is called, this action will perform like a <b>RecognizeFieldOCR_S</b> action.</p>
Parameters	None
Level	Field level only.
Returns	<i>False</i> if the action is not run at the Field level or if recognition fails. Otherwise, <i>True</i> .
Example	RecognizeFieldOCR_S() RecognizeFieldVoteICR_C()
See also...	RecognizeFieldVoteOCR_S



## RecognizePageFields2CCO\_ICR\_C

Actions Library	icr_c.rra/rrx
Description	Uses the matching skills and procedures of the <b>cco2cco</b> action to recognize and store values of a <i>source</i> page.
Parameters	None
Level	Page only.
Returns	<i>False</i> if the ruleset with this action is not bound to a <b>Page</b> object of the Document Hierarchy. Otherwise, <i>True</i> .
Example	RecognizePageFields2CCO_ICR_C(0)  <b>Note:</b> For more about this action, be sure to check the description for the <b>cco2cco</b> action.

## RecognizePageFieldsICR\_C

Actions Library	icr_c.rra/rrx
Description	A page-level action that recognizes all fields on the page that have been configured for ICR/C recognition (see the <i>ICR/C</i> tab of the <i>Recognition Options Setup</i> .  Individual <i>field-level</i> recognition actions will overwrite the results from this <i>page-level</i> action.  This action will not recognize a <i>zoned</i> field if the <b>Skip Recognition</b> checkbox in the <i>ICR/C</i> tab of the <i>Recognition Options Setup</i> dialog has been selected.
Parameters	None
Level	Page level only.
Returns	<i>False</i> if the action is not run at the Page level or if recognition fails. Otherwise, <i>True</i> .
Example	RecognizePageFieldsICR_C()

## RecognizePageICR\_C

Actions Library	icr_c.rra/rrx
Description	<p>Refers to settings in the <i>ICR/C</i> tab of the <b>Recognitions Options Setup</b> dialog to recognize <i>all</i> characters on a page, and populates the page's CCO file with the recognition results.</p> <p><b>Alert!</b> If a CCO file does not exist at the time this action is called, the action will create one.</p>
Parameters	None
Level	Page level only.
Returns	<i>False</i> if the action is not run at the <b>Page</b> level of the Document Hierarchy, or if recognition is not successful. Otherwise, <i>True</i> .
Example	<p>AnalyzeImage()            RotateImage()  <b>RecognizePageICR_C()</b></p> <p>This sequence creates a CCO file and checks to see if rotation of the image is needed.</p> <p><b>Full-page</b> recognition then takes place in response to settings in the <i>ICR/C</i> tab of the <b>Recognition Options Setup</b> dialog. The recognition results are stored in the page's CCO file.</p>
See also...	RecognizePageOCR_S, CreateTextFile

## ImageFix Actions

These actions are used to improve image quality, in response to specifications in a pre-defined ImageFix Settings file (**imagefix.ini**, for example), located in an application's **Process** directory.

Alternatively, a rule with these actions can specify a fingerprint-specific ImageFix Settings file (see Chapter 3 of the *Rulemanager Reference*.) The file is stored in the application's **fingerprint** folder and its name must include the Fingerprint ID – **1044.ini**, for example.

Since actions can be called conditionally according to Page Type or other properties, it is possible to tailor the image processing to be very specific to different images within a batch. For example, fingerprint-specific settings can be applied to images.

Image processing Settings files are created in *Rulemanager* and loaded by the appropriate actions.

## ImageEnhance

Actions Library	ImageFix.rra/.rrx
Description	Initiates image processing. Include this action after a LoadSettings or LoadSettings_FingerprintID action.
Parameters	The file extension that the action is to assign to the backup of the original Image file. For example: <b>tio</b> The extension should be 3 or 4 <i>alphanumeric</i> characters.
Level	Page or Field Level.
Returns	<i>False</i> if the parameter is not 3 or 4 <i>alphanumeric</i> characters.
Example	LoadSettings(C:\ParentDir\Invoice\Process\ImageFix.ini) <b>ImageEnhance(tio)</b> Keep in mind in this example that the ImageFix settings specified in ImageFix.ini will be applied to every page in the batch.

### LoadSettings

Actions Library	ImageFix.rra/.rrx
Description	Loads the settings that ImageFix will use to process all images in the current batch.
Parameters	The name of the ImageFix Settings file (.ini). The action will search for a file of this name in the application's <b>Process</b> directory.  If this file is not in your application's <b>Process</b> directory, the file's full path is required.
Level	All
Returns	<i>False</i> if the ImageFix Settings file that you specify as a parameter is not found; otherwise, <i>True</i> .
Example	LoadSettings(C:\ParentDir\Invoice\Process\ImageFix.ini) ImageEnhance(tio)

### LoadSettings\_FingerprintID

Actions Library	ImageFix.rra/.rrx
Description	Loads the specific ImageFix Settings file (.ini) that corresponds to the Fingerprint ID of the current page.  The action will search the application's <b>fingerprint</b> folder for a fingerprint-specific ImageFix Settings file. Settings for these files are assigned during the Image Enhancement phase of Fingerprint Definition, using tools of <i>Rulemanager's Image Processing Setup</i> dialog. (Chapter 3 of the <i>Rulemanager Reference</i> shows you how to define a fingerprint-specific ImageFix Settings file.)  <b>Important!</b> The name of a fingerprint-specific ImageFix Settings file is limited to the Fingerprint ID with the ".ini" extension: <b>1044.ini</b> , for example.
Parameters	None
Level	Page level only.
Returns	<i>False</i> if a fingerprint-specific Settings file does not exist; otherwise <i>True</i> .
Example	LoadSettings(C:\ParentDir\Invoice\fingerprint\1044.ini) ImageEnhance(tio)

## Imprint Actions

**Imprint** actions place text values onto the current image.

### ImPrint

Actions Library	Imprint.rra/.rrx
Description	<p>Imprints the text you specify onto the current page's image file.</p> <p>By default, the text's font size is 12, and font's style "Times New Roman", with an adjusted width of 30.</p>
Parameters	<p>Three parts, separated by semicolons(;</p> <p>1) Comma-delimited Endorsement string:</p> <p style="padding-left: 40px;">You can combine several options to construct your Endorsement String:</p> <p>Text</p> <p>Variables:</p> <p>1) @BatchID     Batch ID  2) @ID            ID, current object  3) @Status        Status, current object  4) @Type          Type, current object  5) @Date          Today's date  6) @Time          Current time  7) @Variable,ObjectType/Variable Name</p> <p>For Object type in Option #7 above:  0 = Batch, 1 = Document,  2 = Page, 3 = Field</p> <p>(Example: @Variable,3/TemplateID)</p> <p>Field Values: #&lt;field name&gt;. Example: #SSN</p> <p>2) X coordinate for the starting position of the text.  3) Y coordinate for the starting position of the text.</p>
Level	Page level only.
Returns	<i>False</i> if parameter(s) are missing, or the X or Y parameter is not <i>Numeric</i> .
Example	<pre>SetFontName(Arial) SetFontSize(10) SetAdjustedWidth(100) <b>ImPrint(@BatchID,Page:,@ID;0;0)</b></pre> <p>This example will place the Batch ID and Page ID at the top of the image.</p>

### SetAdjustedWidth

## Imprint Actions

---

Actions Library	Imprint.rra/.rrx
Description	Adjust the width of the imprinted text.  This action is <i>optional</i> . If the action is not used, the adjusted width will default to 30. If your text is being cut off, increase the parameter value.
Parameters	<i>Numeric</i> value of the maximum character length of the string to be imprinted.
Level	Page level only.
Returns	<i>False</i> if the parameter is not numeric; otherwise, <i>True</i> .
Example	SetFontName(Arial) SetFontSize(10) <b>SetAdjustedWidth(100)</b> Imprint(@BatchID,Page:,@ID;0;0)

## SetFontName

Actions Library	Imprint.rra/.rrx
Description	Specifies the font style that will be used.  This action is <i>optional</i> . If not used, the font style will default to <b>Times New Roman</b> .
Parameters	<i>String</i> value of the font's name.
Level	Page level only.
Returns	<i>False</i> if the parameter is missing; otherwise <i>True</i> .
Example	<b>SetFontName(Arial)</b> SetFontSize(10) SetAdjustedWidth(100)    Imprint(@BatchID,Page:,@ID;0;0)

## SetFontSize

Actions Library	Imprint.rra/.rrx
Description	Specifies the font size that will be used.  This action is <i>optional</i> . If you do not use the action, the font size will default to <b>12</b> .
Parameters	<i>Numeric</i> value of the font's size.
Level	Page level only.
Returns	<i>False</i> if the parameter is not <i>Numeric</i> ; otherwise <i>True</i> .
Example	SetFontName(Arial) <b>SetFontSize(10)</b> SetAdjustedWidth(100)    Imprint(@BatchID,Page:,@ID;0;0)

## SetOpaque

Actions Library	Imprint.rra/.rrx
Description	Sets the opacity/transparency of rectangles that cover text, or images beneath the text, resulting from Imprint actions.
Parameters	Integer value: 1 or 0.  SetOpaque(1) indicates full opacity and is typical.  SetOpaque(0) results in subsequent rectangles that are translucent (transparent.)
Level	All, but generally at the Page level
Returns	False if the parameter is not an Integer. Otherwise, True
Example	<b>SetOpaque ( 1 )</b>

## Intellocate Actions

**Intellocate** actions assign zone position information to **Field** objects of the Document Hierarchy when an application sets up new fingerprints.

For example, when the *Taskmaster for APTs* application encounters an image of an invoice without a fingerprint, an operator uses the CreateFP task of the VerifyVendor job to generate a new fingerprint, and store the fingerprint's Image file (.tif) and Processing file (.cco) in the application's **fingerprint** directory.

The operator uses the job's next task – VerifyAddV – to manually zone some or all of the fields on the *source* page – the invoice, in this case. (**Page** objects in a Document Hierarchy have one or more **Fields** objects.). Zoning procedures cover Line Item sub fields, as well.

After the VerifyAddV task has finished, the job's Intellocate task applies rules containing **Intellocate** actions to transfer the fingerprint's zone position information (generated manually by the operator) to the Document Hierarchy file (<**application.xml**>).

At this point, a new fingerprint has been established and the Document Hierarchy file (.xml) contains the fingerprint's zone position information. Further rules can now be run on this previously unmatched *source* page to capture its data. And the next time that the application processes a page with identical fields in the same locations, the page can be matched to the existing fingerprint.

✓ When a *rulerunner task* of a Main job successfully matches a *source* page to the new fingerprint, position information for zoned fields is added to the page's Data file (.xml).

### iloc\_AdjustZones

Actions Library	IntelLocate.rra/.rrx/.rrx
Description	Updates fingerprint-specific position coordinates for <b>Field</b> objects in the Document Hierarchy based on the locations listed for the current <i>source</i> page's Data file (.xml)  Because the action applies to a <b>Page</b> object, it updates coordinates for all fields with zones.
Parameters	None
Level	Page level only.
Returns	<i>False</i> if a fingerprint match has not occurred and a Template ID value has not been assigned to the current page, or if the Document Hierarchy file cannot be saved. Otherwise, <i>True</i> .
Example	iloc_AdjustZones()



**iLoc\_AssignPageType**

Actions Library	IntelLocate.rra/.rrx/.rrx
Description	Assigns a <b>required</b> Page Type value to a newly created fingerprint.  It will be stored as a tp_PageType value in the <b>Template</b> table of the application's Rules database.).
Parameters	<i>Numeric</i> value for the Page Type to be assigned to the <b>current</b> page – and to the new fingerprint that will be based on the page.  <b>Alert!</b> Page Type values are in the <b>Page Type</b> table of an application's Rules database.
Level	Page level only.
Returns	<i>False</i> if the parameter is not <i>Numeric</i> . Otherwise, <i>True</i> .
Example	Iloc_AssignPageType(2)  In the <b>Page Type</b> table of the <i>1040EZ</i> application's Rules database, "2" specifies the <i>1040EZ_Page</i> Page Type.

**iLoc\_SetDetailZones**

Actions Library	IntelLocate.rra/.rrx/.rrx
Description	Writes the position coordinates of a new fingerprint's <b>Detail Line</b> fields <b>from</b> a page's Data file <b>to</b> the <b>Pos</b> properties of the corresponding <b>Detail Line Field</b> objects of the application's Document Hierarchy..
Parameters	None
Level	Page level only.
Returns	<i>False</i> if the Document Hierarchy file (.xml) cannot be found; if a fingerprint match has not occurred; or if the Document Hierarchy file (.xml) cannot be saved. Otherwise, <i>True</i> .
Example	iLoc_SetDetailZones()

### iloc\_SetZones

Actions Library	IntelLocate.rra/.rrx/.rrx
Description	Writes the position coordinates of a new fingerprint's <i>zoned</i> fields <i>from</i> a page's Data file <i>to</i> the <b>Pos</b> properties of the corresponding <b>Field</b> objects in the Document Hierarchy file (.xml).
Parameters	None
Level	Page level only.
Returns	<i>False</i> if a fingerprint match has not occurred, or if the Document Hierarchy file (.xml) cannot be saved. Otherwise, <i>True</i> .
Example	iloc_SetZones ()

### IsPageDataMissing

Actions Library	IntelLocate.rra/.rrx/.rrx
Description	Checks to see that the current page has a valid Data file ( <b>TM00001.xml</b> , for example).  The <b>Intellocate</b> actions will place the new zone information for the current page into the Data file.
Parameters	None
Level	Page level only
Returns	<i>False</i> if the current page does not have a valid Data file. Otherwise, <i>True</i> .
Example	IsPageDataMissing()

## IOverlay Actions

These actions overlay the image file you specify on the image of the current page.

### Overlay

Actions Library	IOverlay.rra/.rrx
Description	<p>Combines the current image with the image file specified by the <b>SetBackgroundImage</b> action into a new image replacing the current image.</p> <p>This action is used to reinstate a form background that was “dropped out” during scanning.</p> <p><i>Alert!</i> Use this action after a <b>SetBackgroundImage</b>, <b>SetDitheringBackground</b> or <b>SetHaloBackground</b> action.</p>
Parameters	None.
Level	Page level only.
Returns	<i>False</i> if the action is not applied to a <b>Page</b> object, or if the action encounters an error. Otherwise, <i>True</i> .
Example	<pre>SetBackgroundImage(C:\ParentDir\mclaims\process\hcfat\hcfat.tif) SetDitheringBackground(True) SetHaloBackground(True) <b>Overlay()</b></pre>

### SetBackgroundImage

Actions Library	IOverlay.rra/.rrx
Description	Designates the Image file that will overlay the current page's image.
Parameters	Full path to the overlay Image file.
Level	All
Returns	Always <i>True</i> .
Example	SetBackgroundImage(C:\ParentDir\mclaims\process\hcfa \hcfat.tif) SetDitheringBackground(True) SetHaloBackground(True) Overlay()

### SetDitheringBackground

Actions Library	IOverlay.rra/.rrx
Description	Enables or disables dithering of the background image.  Dithering will make the background image appear lighter than the current image's information so that visually it appears less prominent.
Parameters	<i>String</i> value to enable dithering ("True") or disable dithering ("False").
Level	All
Returns	Always <i>True</i> .
Example	SetBackgroundImage(C:\ParentDir\mclaims\process\hcfa \hcfat.tif) <b>SetDitheringBackground(True)</b> SetHaloBackground(True) Overlay()

## SetHaloBackground

Actions Library	IOverlay.rra/.rrx
Description	<p>Enables or disables a halo of white pixels around any black pixels from the current image where they would otherwise touch pixels from the background.</p> <p>This makes the foreground information easier to read.</p>
Parameters	<i>String</i> value to enable a halo (“True”) or prevent a halo (“False”).
Level	All
Returns	Always <i>True</i> .
Example	<pre>SetBackgroundImage(C:\ParentDir\mclaims\process\hcfat.tif) SetDitheringBackground(True)           <b>SetHaloBackground(True)</b> Overlay()</pre>

## Locate Actions

A rule using a **Locate** action is a directive to find a **field value** on an image.

Typically, if this rule finds the value it is searching for, the rule will save that value, and the location on the image of that value, in the data file.

When you construct a *new* rule using Locate actions, combining just a few of the actions below will often suffice. As an example, the *APT* application uses the following global rule to locate a Purchase Order Number on an invoice (*comments are in italics.*)

```
WordFind(Purchase Order) / Looks for a "Purchase Order" title. If
successful...
GoRightWord(1) / ...moves right one word.
IsNumber(30) / Checks to be sure the word is a number that is 30% numeric.
UpdateField() / Saves the value to the Data file.
```

But suppose the invoice you're working with doesn't have a label, "Purchase Order." What if the invoice – and the corresponding fingerprint - use "P.O. Number" instead. You can craft a custom rule to replace the global rule's **WordFind** action with:

```
WordFind(P.O.Number) / Looks for "P.O. Number" as the title
```

Of course, since there are many ways a vendor may label the purchase order number field, it can be even more effective to use the **FindKeyList** action that will simultaneously search for many different potential values.

Searching actions, such as **WordFind** and **FindKeyList** always start at the top left of the page. If you need to find subsequent occurrences of a word, then **WordFindNext** is a handy action.

The topics in this section describe the standard **Locate** actions. As you review the explanations, keep in mind that in order for **Locate** actions to work, the image must first be processed with **Recog** actions. Remember, too, that **Locate** rules do *not* distinguish between words representing the titles of fields on a page – *static* values - and words representing the temporary *entered values* entered by a form-filler.

✓ Explanations of the **Locate** actions use the term "**current page**" to designate a page represented by both an Image file (.tif) *and* the associated Connected Components Object file (.cco). Information in a Components file identifies the **characters, words, and the lines** that hold these words.

## How to Assemble a Keyword List

Some **Locate** actions search a page for a **keyword** such as "Invoice Number" or "P.O. Number." The list of keywords resides in a Keyword file; the file's name (without an extension) is the action's parameter. If the action matches a word in the file with a word

on the current page, the action returns a *True* condition, and the rule continues with the next action.

An example action that looks for an “Invoice Number” keyword in a keyword list file, *InvNum.key*, has this form:

FindKeyList(InvNum)

The *APT* application, as an example, includes these Keyword files in its **Process** directory:

- InvDate.Key**            “Invoice Date” and its variations
- InvNum.Key**            “Invoice Number”
- IsDate.Key**            Alternative Date formats
- PONumber.Key**        “Purchase Order Number”
- TotalDue:Key**        “Total Due”

The relatively lengthy **TotalDue:Key** file has these keywords:

Amount Due	TOTAL
AMOUNT DUE	Total \$
Balance	TOTAL \$
BALANCE	Total Amount
Balance Due	TOTAL AMOUNT
BALANCE DUE	TOTAL AMOUNT DUE
CREDIT AMOUNT	TOTAL AMOUNT DUE:
INV....[ ]*To	Total Amount Due:
INV....[ ]*TO	Total Charges
Invoice Amount	Total Charges \$
INVOICE TOTAL	Total Due
Invoice total	TOTAL DUE
Pay This Amount	Total Due >>>
PAY THIS AMOUNT	Total Invoice
PLEASE PAY THIS	TOTAL INVOICE

## Locate Actions

---

AMOUNT	
Please Pay This Amount	TOTAL INVOICE AMOUNT
Please pay this amount	TOTAL REMIT
Total	USD \$

To add a keyword to an existing list, use *WordPad* or other text file editing program to open the file. Type in the new keyword and save the file.

To assemble a new file:

---

Step	Action
1.	In the <i>WordPad Window</i> , select <b>New</b> from the <b>File</b> menu.
2.	Enter the file's keywords and regular expressions. (Don't worry! You can return later to add more words.)
3.	Select <b>Save</b> from the <b>File</b> menu; save the Keyword file in the application <b>Process</b> directory as a <i>Text Document</i> (.txt).
4.	Open the <b>Process</b> directory. Be sure the new file is part of the directory, then re-name it by replacing <b>.txt</b> with <b>.key</b> as the file's extension.
5.	Double-click on the file's listing to review and modify its contents.
6.	You can use "expressions" as keywords. For example, the <b>INV....[ ]*To</b> keyword is really a regular expression. For details, consult your VBScript Programmer's Reference.

The tables which follow describe the **Locate** actions.



## AddLeadingZeros/Leading Zeros

Actions Library	Locate.rra/Validations.rrx  <i><b>Important!</b></i> This <i>Rulemanager Locate</i> action is the <b>LeadingZeros</b> action in Datacap Studio's <b>Validations.rrx</b> file.
Description	Adds one or more zeros to the left (beginning) of a <i>located</i> word's first character – until the value's total number of characters equals the parameter you specify.
Parameter	The <i>Numeric</i> value of the word's maximum length after padding with zeros
Level	Field level only.
Returns	Always <i>True</i> .
Example	<b>AddLeadingZeros (8)</b> directs its rule to append zeros to the beginning of the recognized value until the word has a total of eight characters:  345.67 becomes 00345.67
See also...	AddTrailingZeros, PadRight

## DefaultValue

Actions Library	Locate.rra/.rrx
Description	Assigns a default value to the current field.
Parameter	The <i>String</i> value you're assigning to the field.
Level	Field level only.
Returns	Always <i>True</i> .
Example	DefaultValue(Bill Paid)  or  DefaultValue(PastDue!)

### FilterIt

Actions Library	Locate.rra/.rrx
Description	Removes <i>all</i> instances of the character(s) you enter as a parameter from the located word.
Parameter	A <i>String</i> containing the character(s) to be removed.  <i>Alert!</i> This action removes every instance of the character or characters.
Level	Field level only.
Returns	Always <i>True</i> .
Example	FilterIt(-)  31-Dec-01 becomes 31Dec01  FilterIt(1)  31-Dec-01 becomes 3-Dec-0
See also...	DeleteString

## FindDBList

Actions Library	Locate.rra/.rrx
Description	Finds and opens the Keyword Database you enter as a parameter, and attempts to match words in the current source page's Fingerprint file (.cco) with keywords in the database's Keyword tables.
Parameter	The name of the Keyword Database. Typically, this is an Access database file ( <b>KeyDB.mdb</b> , for example) in the application's <b>Process</b> directory.
Level	Page or Field.
Returns	False if the page's Fingerprint file (.cco) is not available, if the file is empty, or if the database does not exist.
Example	<e>FindDBList(Doctors)

## FindDBList\_InZone

Actions Library	Locate.rra/.rrx
Description	Attempts to match a word in a zoned field, or words in a zoned block, against keywords in tables of keyword database.  When a match occurs, the action provides the Fingerprint file (.cco) of the <i>source</i> page with the word's locating coordinates.
Parameter	<i>String</i> value of the database name and path.
Level	Field.
Returns	False if the source page's Fingerprint file (.cco) is not available or if it is empty. Otherwise, True.
Example	FindDBList_InZone(c:\MClaimsApp\Databases\Physician_Data)

## FindKeyList

Actions Library	Locate.rra/.rrx
Description	<p>Finds and opens the Keyword file you specify as a parameter - <i>then</i> checks the words on the current page against the keywords in the list, looking for a match.</p> <p>To improve the possibility of matching in spite of recognition inaccuracies, <b>FindKeyList</b> automatically adjusts the search criteria to allow for common character substitutions. For example, if the list includes “will” and the recognition read “wi11”, a match will still occur.</p>
Parameter	The name of the Keyword file <i>without</i> the <b>.key</b> extension. This file is expected to be in the application’s <b>Process</b> folder.
Level	Field level only.
Returns	<i>False</i> is the Keyword file is not found <i>or</i> if none of the keywords is in the list. Otherwise, <i>True</i> .
Example	<p>FindKeyList(InvNum)</p> <p>This action searches the current page for the first keyword in the <i>APT Number</i> Keyword file (<b>invnum.key</b>). If successful, the search stops; if not, the action continues with the next keyword until a match is found, or until there are no more keywords.</p>

## FindKeyList\_InZone

Actions Library	Locate.rra/.rrx
Description	<p>Finds and opens the Keyword file you enter as a parameter, and attempts to match words in the file with keywords within the bound zone of the <i>source</i> page's Fingerprint file (.cco).</p> <p>The action returns the value of each match.</p>
Parameter	<i>String</i> value of the keyword file's name - without a <b>.key</b> extension.
Level	Field.
Returns	False if the action cannot open the Fingerprint file (.cco) of the <i>source</i> page. Otherwise, True
Example	<p>FindKeyList_InZone(Line_Items)</p> <p>In this example, if the zone encompasses a Line Item Table, the action might well match multiple key words in the "Line_Items" list with words in the zone.</p>

## FindLastKeyList

Actions Library	Locate.rra/.rrx
Description	Searches the Keyword file you enter as a parameter for the last instance of a match between a keyword in the file and a word in the Fingerprint file (.cco) of the current <i>source</i> page
Parameter	<i>String</i> value of the Keyword file's name - without a ".key" extension
Level	Field
Returns	False if the action cannot find the page's CCO file or if the file has no values. Otherwise, True.
Example	FindLastKeyList(Date)  In this example, the action returns the location of the last instance of a match between a word in the <b>Date</b> keyword file and a value on the current page.

## FindLastKeyList\_InZone

Actions Library	Locate.rra/.rrx
Description	Searches the Keyword file you enter as a parameter for the last instance of a match between a keyword in the file and a word in the bound zone of the current <i>source</i> page.
Parameter	<i>String</i> name of the Keyword file, without a .key extension.
Level	Field.
Returns	False if the action cannot load the page's Fingerprint file (.cco) or if the file is empty. Otherwise, True.
Example	FindLastKeyList_InZone(ClaimsData)  This could be a helpful action if the zone is made up of data for Medical Claims and the task is looking for the end of that zone - just to speed processing.

## FindLastRegEx

Actions Library	Locate.rra/.rrx
Description	In response to the Regular Expression you enter a parameter, locates the last instance of a particular value on the current page.
Parameter	Regular Expression consisting of Regular Expression characters.
Level	Page.
Returns	False if the page's Fingerprint file (.cco) cannot be loaded, or if it is empty. Otherwise, True.
Example	<pre>FindLastRegEx("invoice"\b\s"number")</pre> <p>This action looks for and returns the final location of "Invoice Number" on the current <i>source</i> page.</p>

## FindLastRegEx\_InZone

Actions Library	Locate.rra/.rrx
Description	Uses the Regular Expression you enter as a parameter to locate the final instance of a particular value within a zone.
Parameter	Regular Expression consisting of valid Regular Expression characters.
Level	Page
Returns	False if the Fingerprint file (.cco) of the current <i>source</i> page is not available, or if it is empty. Otherwise, True.
Example	<pre>FindLastRegEx_InZone("Total"\b\s"Amount")</pre> <p>This action performs well if the <i>source</i> page has a zone with one or more <i>Total Amount</i> fields: the action will locate the last.</p>

## FindLastRegExList

Actions Library	Locate.rra/.rrx
Description	Uses the Regular Expression you enter as a parameter to locate the last instance of a Keyword list in a specific Keyword file.
Parameter	Regular Expression that identifies the Keyword file, and consists of valid Regular Expression characters.
Level	Page
Returns	False if the Fingerprint file (.cco) of the current source page is unavailable or empty. Otherwise, True.
Example	<pre>FindLastRegExList("""Doctor"\b\s\"Names"\b\s\"Address""")</pre> <p>The Regular Expression in this action locates the <b>Doctor Names Address</b> Keyword file. The action then finds the last Keyword list in the file - whatever it happens to be.</p>

## FindLastRegExList\_InZone

Actions Library	Locate.rra/.rrx
Description	Uses the Regular Expression you enter as a parameter to locate the last instance of a Keyword list in a specific Keyword file, and attempts to match a keyword in the list with a word in the zone.
Parameter	Regular Expression that identifies the Keyword file, and consists of valid Regular Expression characters.
Level	Page
Returns	False if the action cannot load the Knowledge file (.cco) of the current source page, or if the file is empty. Otherwise, True.
Example	<pre>FindLastRegExList_InZone("""Doctor"\b\s\"Names"\b\s\"Address""")</pre> <p>The Regular Expression in this action locates the <b>Doctor Names Address</b> Keyword file, and opens the last Keyword list in the file. The action then attempts to match a keyword in the list with a word in the zoned field of the <i>source</i> page.</p>

## FindLastWord

Actions Library	Locate.rra/.rrx
Description	Locates the last instance of the word you enter as a parameter, within the image of the current page, and adds the word's location coordinates to the page's Fingerprint file (.cco).
Parameter	<i>String</i> value of the word.
Level	Page.
Returns	False if the action cannot load the CCO file or if the file is empty. Otherwise, True.
Example	<b>FindLastWord(Total)</b> GoRightWord(1) IsCurrency()  This sequence finds the last instance of "Total" on the current page; moves right one word; and checks to be sure that the word has a Currency value.

## FindLastWord\_InZone

Actions Library	Locate.rra/.rrx
Description	Checks a zoned field (or block of fields) for the last instance of the word you enter as a parameter; and returns the word's locating coordinates to the page's Fingerprint file (.cco).
Parameter	<i>String</i> value of the target word.
Level	Page.
Returns	False if the CCO file is not available, or is empty. Otherwise, True.
Example	<b>FindLastWord_InZone(Total)</b> GoRightWord(1) IsCurrency()  This sequence attempts to find the last example of <i>Total</i> in a zone. It then locates and validates a Currency amount for the <i>Total</i> field.



## FindNextDBList

Actions Library	Locate.rra/.rrx
Description	As a follow-up to a <b>FindDBList</b> action, finds and opens another Keyword Database designated by the parameter, and attempts to match words in the current <i>source</i> page's Fingerprint file (.cco) with keywords in this database's Keyword tables.
Parameter	<i>String</i> value of the Database name.
Level	Page.
Returns	False if the page's CCO file is not available or if it is empty. Otherwise, True.
Example	<p><b>FindDBList(Doctors)</b> FindNextDBList(Physicians)</p> <p>In this example, the <b>FindDBList(Doctors)</b> action searches Keyword tables of the <i>Doctors</i> database, looking for a match.</p> <p>If there is no match, the <b>FindNextDBList(Physicians)</b> action directs the rule to access tables of the <i>Physicians</i> Keyword database.</p>

## FindNextDBList\_InZone

Actions Library	Locate.rra/.rrx
Description	As a follow-up to a <b>FindDBList_InZone</b> action, finds and opens a second Keyword Database designated by the parameter, and attempts to match words in the bound zone with keywords in the second database's Keyword tables.
Parameter	<i>String</i> value of the database's name.
Level	Page.
Returns	False if the action cannot load the <i>source</i> page's Fingerprint file (.cco) or if the file is empty. Otherwise, True
Example	<p>In the example below, if the first action fails to match a keyword in the <i>Inventory</i> database, the rule will immediately search tables of the <i>Parts</i> database.</p> <p>FindDBList_InZone(Inventory) FindNextDBList_InZone(Parts)</p>

## FindNextKeyList

Actions Library	Locate.rra/.rrx
Description	Opens a second Keyword file as a direct follow-up action to a <b>FindKeyList</b> action.
Parameter	<i>String</i> value of the file's name - without a .key extension.
Level	Field.
Returns	False if the Keyword file cannot be found; otherwise, True.
Example	<p>In the example below, if the <b>FindKeyList(Tax)</b> action cannot find a match, the rule calls <b>FindNextKeyList(Tax)</b> instantly.</p> <pre>FindKeyList(Tax) FindNextKeyList(IRS)</pre>

## FindNextKeyList\_InZone

Actions Library	Locate.rra/.rrx
Description	<p>Opens a second Keyword file as a direct follow-up action to a <b>FindKeyList_InZone</b> action.</p> <p>The combined actions attempt to match words for a zoned field or zoned block with Keyword values in either Keyword file</p>
Parameter	<i>String</i> value of the next Keyword file's name - without a .key extension.
Level	Field.
Returns	False if the Fingerprint file(.cco) of the current <i>source</i> page has not been loaded, or is empty. Otherwise, True.
Example	<p>In the example below, the first action attempts to match values of a zoned field, or a zoned block, with values in the <b>Inventory.key</b> file.</p> <p>If successful, the action will provide the page's Fingerprint file (.cco) with the field's location coordinates.</p> <p>If a match does not occur, the rule will try the <b>Parts.key</b> file by calling the <b>FindNextKeyList_InZone(Parts)</b> action.</p> <pre>FindKeyList_InZone(Inventory) <b>FindNextKeyList_InZone(Parts)</b></pre>

## FindNextRegExList

Actions Library	Locate.rra/.rrx
Description	Uses the parameter's Regular Expression to specify a Keyword file that is to be opened and explored immediately after the file designated by a preceding <b>FindRegExList</b> action.
Parameter	A Regular Expression that uses valid Regular Expression characters to identify the "next" file.
Level	Field.
Returns	False if the <i>source</i> page's Fingerprint file (.cco) is not available or if it is empty. Otherwise, True.
Example	FindRegExList("Inventory"\b\s\ "Parts") FindNextRegExList(("Inventory"))

## FindNextRegExList\_InZone

Actions Library	Locate.rra/.rrx
Description	Uses the parameter's Regular Expression to indicate the Keyword list to be accessed when the list opened by a preceding <b>FindRegExList_InZone</b> action does not contain a matching keyword.  A rule with this action <i>must</i> be bound to a zoned field or zoned block of fields.
Parameter	A Regular Expression that specifies the follow-up Keyword list. The Regular Expression must be made up of valid Regular Expression characters.
Level	Field
Returns	False if the <i>source</i> page's Fingerprint file (.cco) is not available, or if it is empty.
Example	FindRegExList_InZone("Inventory"\b\s\ "Parts") FindNextRegExList(("Inventory"))

## FindRegExList

Actions Library	Locate.rra/.rrx
Description	Uses the parameter's Regular Expression to specify a Keyword file with keyword values that can be matched to words on the current <i>source</i> page.  If a match occurs, the action will update the page's Fingerprint file(.cco) with the location of the matched word.
Parameter	Regular Expression identifying and locating the Keyword file.
Level	Field.
Returns	False if the CCO file is not available or if it is empty. Otherwise, True.
Example	FindRegExList("Inventory"\b\s\ \"Parts")

## FindRegExList\_InZone

Actions Library	Locate.rra/.rrx
Description	Uses the parameter's Regular Expression to specify a Keyword file with keyword values that can be matched to a word in the bound zoned field or block on the current <i>source</i> page.  If a match occurs, the action will update the page's Fingerprint file (.cco) with the location of the matched word.
Parameter	A Regular Expression specifying the name of the Keyword file.
Level	Field.
Returns	False if the <i>source</i> page's CCO file has not been loaded, or is empty.
Example	FindRegExList_InZone("Inventory"\b\s\ \"Parts")

## GoAboveWord

Actions Library	Locate.rra/.rrx
Description	Moves to a word which is <i>above</i> the current word by <i>n</i> lines. This action is always called after a search action has been used.
Parameter	An <i>Integer</i> indicating the number of lines above the current word.
Level	Field level only.
Returns	<i>True</i> if a word is found. Otherwise, <i>False</i> .
Example	<p>WordFind(Total)  <b>GoAboveWord(1)</b>  GoRightWord(1)  UpdateField()</p> <p>This <i>APT</i> sequence finds the word which is the <i>Tax</i> field's <i>entered</i> value ("29.78") when a fingerprint has these fields and values:</p> <p><b>Tax</b> 29.78  <b>Total</b> 234.70</p>
See also...	GoBelowWord, FindWord, FindKeyList

## GoBelowWord

Actions Library	Locate.rra/.rrx
Description	Moves to a word which is <i>below</i> the current word by <i>n</i> lines. This action is always called after a search action has been used.
Parameter	An <i>Integer</i> indicating the number of lines below the current word.
Level	Field level only.
Returns	<i>True</i> if a word is found. Otherwise, <i>False</i> .
Example	<p>WordFind(Tax)  <b>GoBelowWord(1)</b>  GoRightWord(1)  UpdateField()</p> <p>This <i>APT</i> sequence finds the word which is the <i>Total</i> field's <i>entered</i> value ("234.70") when a fingerprint has these fields:</p> <p><b>Tax</b> 29.78  <b>Total</b> 234.70</p>
See also...	GoAboveWord, FindWord, FindKeyList

## GoDownLine

Actions Library	Locate.rra/.rrx
Description	Moves to the first word of a line that is <i>n</i> lines below the current line. This action is always called after a search action has been used.
Parameters	An <i>Integer</i> indicating the number of lines below the current line.
Level	Field level only.
Returns	<i>True</i> if a line is found. Otherwise, <i>False</i> .
Example	<pre>FindKey(Invoice) <b>GoDownLine(1)</b> GoRightWord(1) UpdateField()</pre> <p>This sequence finds the word which is the <i>Number</i> field's <i>entered</i> value ("10034") when a fingerprint has these fields:</p> <pre>INVOICE Number: 10034</pre>
See also...	GoUpLine, FindWord, FindKeyList

## GoFirstLine

Actions Library	Locate.rra/.rrx
Description	Moves to the first line of the current zone, or to the first line of the <i>source</i> page if there is no zone.
Parameters	None.
Level	Page or field.
Returns	False if the page's Fingerprint file (.cco) is unavailable or empty. Otherwise, True.
Example	GoFirstLine()

## GoFirstWord

Actions Library	Locate.rra/.rrx
Description	Navigates to the first word on the current line.
Parameters	None.
Level	Page.
Returns	False if the page's Fingerprint file (.cco) is not available or if it is empty. Otherwise, True.
Example	GoFirstWord()

## GoLastLine

Actions Library	Locate.rra/.rrx
Description	Navigates to the last line in the current zone, or to the last line in the current page if a zone is not present.
Parameters	None.
Level	Page or field.
Returns	False if the <i>source</i> page's Fingerprint file (.cco) is unavailable or empty. Otherwise, True.
Example	GoLastLine()

## GoLastWord

Actions Library	Locate.rra/.rrx
Description	Navigates to the last word in the current line. If the source page's Fingerprint file (.cco) does not have a line position, the action defaults to the first line in the current zone or page, if no zone is present.
Parameters	None.
Level	Page or field.
Returns	False if the <i>source</i> page's Fingerprint file (.cco) is unavailable or empty. Otherwise, True.
Example	GoLastWord()

### GoLeftWord

Actions Library	Locate.rra/.rrx				
Description	Moves <i>n</i> words to the left of the current word, in the same line. This action is always called after a search action has been used.				
Parameters	An <i>Integer</i> indicating the number of words to move leftwards from the current position.				
Level	Field level only.				
Returns	<i>True</i> if a word is found. Otherwise, <i>False</i> .				
Example	<p>WordFind(Total)  <b>GoLeftWord(1)</b>          GoBelowWord(1)          UpdateField()</p> <p>If the fingerprint has a table with columns such as <b>Total</b> and <b>Tax</b>, the actions above will locate the <i>Tax</i> amount below (“344.76”):</p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;"><b>Tax</b></td> <td style="text-align: center;"><b>Total</b></td> </tr> <tr> <td style="text-align: center;">344.76</td> <td style="text-align: center;">13,774.00</td> </tr> </table>	<b>Tax</b>	<b>Total</b>	344.76	13,774.00
<b>Tax</b>	<b>Total</b>				
344.76	13,774.00				
See also...	GoRightWord, FindWord, FindKeyList				

### GoRightWord

Actions Library	Locate.rra/.rrx				
Description	Moves <i>n</i> words to the right of the current word, in the same line. This action is always called after a search action has been used.				
Parameter	An <i>Integer</i> indicating the number of words to move to the right.				
Level	Field level only.				
Returns	<i>True</i> if a word is found. Otherwise, <i>False</i> .				
Example	<p>WordFind(Tax)  <b>GoRightWord(1)</b>          GoBelowWord(1)          UpdateField()</p> <p>If the fingerprint has a table with columns such as <b>Total</b> and <b>Tax</b>, the actions above will locate the <i>Total</i> amount below (“13,774.00”):</p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;"><b>Tax</b></td> <td style="text-align: center;"><b>Total</b></td> </tr> <tr> <td style="text-align: center;">344.76</td> <td style="text-align: center;">13,774.00</td> </tr> </table>	<b>Tax</b>	<b>Total</b>	344.76	13,774.00
<b>Tax</b>	<b>Total</b>				
344.76	13,774.00				
See also...	GoLeftWord, FindWord, FindKeyList				



## GoUpLine

Actions Library	Locate.rra/.rrx
Description	Moves to the first word of a line that is <i>n</i> lines above the current line. This action is always called after a search action has been used.
Parameter	An <i>Integer</i> indicating the number of lines above the current line.
Level	Field level only.
Returns	<i>True</i> if a line is found. Otherwise, <i>False</i> .
Example	<pre>FindKey(Invoice) <b>GoUpLine(1)</b> GoRightWord(1) UpdateField()</pre> <p>This sequence finds and extracts the <i>entered</i> value of the <i>Date</i> field (“2/13/03”), in a fingerprint with these words:</p> <pre>Date: 2/13/03 INVOICE</pre>
See also...	GoDownLine, FindWord, FindKeyList

## GroupWords

Actions Library	Locate.rra/.rrx
Description	Groups any words to the left and right of a located word <i>if</i> the target words are themselves separated by a character width equal to or less than the character width you specify as a parameter.
Parameters	<i>Long</i> value of the maximum character width separating words to the right and left of the current word.
Level	Field level only.
Returns	Always <i>True</i> .
Example	<pre>WordFind(Treasury) <b>GroupWords(1.5)</b></pre> <p>If a line contains these words...</p> <pre>20 000 U S Treasury 7 33</pre> <p>the <b>GroupWords</b> action will merge “20” with “000”; “U” with “S” and “7” with “33” to produce:</p> <pre>20000 US Treasury 733</pre>

See also...	GroupWordsLeft, GroupWordsRight
-------------	---------------------------------

### GroupWordsLeft

Actions Library	Locate.rra/.rrx
Description	Groups only words to the left of the located word <i>if</i> the target words are separated by a character width equal to or less than the character width you specify as a parameter.
Parameter	<i>Long</i> value of the maximum character width separating words to the left of the current word.
Level	Field level only.
Returns	Always <i>True</i> .
Example	WordFind(000) <b>GroupWordsLeft(2)</b>  If a line contains these words...  000 US Treasury  the <b>GroupWordsLeft</b> action will merge “20” with “000”  20000 US Treasury
See also...	GroupWords, GroupWordsRight

### GroupWordsRight

Actions Library	Locate.rra/.rrx
Description	Groups words to the right of the located word <i>if</i> the target words are separated by a character width equal to or less than the character width you specify as a parameter
Parameter	<i>Long</i> value of the maximum character width separating words to the right of the current word.
Level	Field level only.
Returns	Always <i>True</i> .

Example	<p>WordFind(20)  <b>GroupWordsRight(2)</b></p> <p>If a line contains these words...</p> <p style="padding-left: 40px;">20 000 US Treasury</p> <p>the <b>GroupWordsRight</b> action will merge “20” with “000” to produce:</p> <p style="padding-left: 40px;">20000 US Treasury</p>
See also...	GroupWords, GroupWordsLeft

## IsAlpha

Actions Library	Locate.rra/.rrx
Description	Determines if the characters in a located word are <i>n% alphabetic</i> .
Parameter	An <i>Integer</i> (0-100) indicating the percentage necessary for the action to return a <i>True</i> condition.
Level	Field level only.
Returns	<i>True</i> if the value specified by the parameter is <i>alpha</i> . Otherwise, <i>False</i> .
Example	<p>If the located word’s <i>recognized</i> value is: ABC1</p> <p style="padding-left: 40px;"><b>IsAlpha(75)</b> returns <i>True</i></p> <p style="padding-left: 40px;"><b>IsAlpha(80)</b> returns <i>False</i></p>
See also...	IsCurrency, IsDateValue IsNumber

## IsCurrency

Actions Library	Locate.rra/.rrx
Description	<p>Determines if the value of the located word is 100% <i>numeric and</i> includes a two-digit decimal amount.</p> <p><b>Alert!</b> When determining the <i>numeric</i> percentage, the action ignores spaces and punctuation such as commas and decimal points.</p>
Parameters	None
Level	Field level only.
Returns	<i>True</i> if the located value is <i>currency</i> . Otherwise, <i>False</i> .

## Locate Actions

---

Example	If the located word's <i>recognized</i> value is: 12.00  <code>IsCurrency()</code> returns <i>True</i>  If the located word's <i>recognized</i> value is: 1200  <code>IsCurrency()</code> returns <i>False</i>  If the located word's <i>recognized</i> value is: \$12.00  <code>IsCurrency()</code> returns <i>True</i>
See also...	IsAlpha, IsDateValue IsNumber

## IsDateValue

Actions Library	Locate .rra
Description	Determines if the <i>recognized</i> value of the located word has an acceptable Date format.
Parameters	None
Level	Field level only.
Returns	<i>True</i> if the located value is an acceptable Date. Otherwise, <i>False</i> .
Example	If the located word's <i>recognized</i> value is: 01/01/2003  <code>IsDateValue()</code> returns <i>True</i>  If the located word's <i>recognized</i> value is: January 01,2003  <code>IsDateValue()</code> returns <i>True</i>  If the located word's <i>recognized</i> value is: 13/13/2003  <code>IsDateValue()</code> returns <i>False</i>
See also...	IsAlpha, IsCurrency, IsNumber

## IsNumber

Actions Library	Locate.rra/.rrx
Description	Determines if the characters in a located word are <i>n%</i> numeric
Parameter	An <i>Integer</i> (0-100) indicating the percentage necessary for the action to return a <i>True</i> condition.
Level	Field level only.
Returns	<i>True</i> if the located value meets the parameter's requirement for an <i>Integer</i> . Otherwise, <i>False</i> .

Example	If the located word's <i>recognized</i> value is: #755  <b>IsNumber ( 75 )</b> returns <i>True</i>  <b>IsNumber ( 80 )</b> returns <i>False</i> .
See also...	IsAlpha, IsCurrency, IsDateValue

## IsValue

Actions Library	Locate.rra/.rrx
Description	Determines if the <i>recognized</i> value of a word of the current page is identical to the value you enter as a parameter.
Parameter	<i>String</i> value to be compared to the object's <i>recognized</i> value.
Level	Field level only.
Returns	<i>True</i> if the located value matches the parameter's value. Otherwise, <i>False</i> .
Example	WordFind(Houston) GoRightWord(2) <b>IsValue(77770)</b>  This sequence confirms that the current page's <i>recognized</i> value for Houston's ZIP code is "77770".  The action returns a <i>Boolean</i> value: <i>True</i> if the values are the same, <i>False</i> if they are not.

## IsValue\_RegEx

Actions Library	Locate.rra/.rrx
Description	Determines if the recognized value of a word of the current page is identical to the Regular Expression parameter you enter.
Parameter	Regular Expression made up of valid Regular Expression characters, specifying the value to be checked against the recognized value.
Level	Field.
Returns	<i>True</i> if the located value matches the parameter's value. Otherwise, <i>False</i> .
Example	IsValue_RegEx(("Total"))

## MaxLength

Actions Library	Locate.rra/.rrx
Description	Compares the number of characters in the located word to a maximum number you supply as the parameter.
Parameter	An <i>Integer</i> specifying the maximum number of characters the word can contain.
Level	Field level only.
Returns	<i>False</i> if the parameter is not <i>Numeric</i> , or if the actual number of characters exceeds the parameter. Otherwise, <i>True</i> .
Example	If the located word's <i>recognized</i> value is: ANYTHING  <b>MaxLength(14)</b> returns <i>True</i>  <b>MaxLength(8)</b> returns <i>True</i>  <b>MaxLength(3)</b> returns <i>False</i> .
See also...	MinLength

## MergeWordLf

Actions Library	Locate.rra/.rrx
Description	<p>Merges the located word with one or more words to the left, on the same line.</p> <p>A “word” in this context is a string of characters, which may include spaces. This action is used when the value searched for may have spaces in it.</p>
Parameter	An <i>Integer</i> indicating the number of words to be merged.
Level	Field level only.
Returns	Always <i>True</i> .
Example	<p>WordFind(2000)  <b>MergeWordLf(1)</b>  UpdateField()</p> <p>Given the following sequence:</p> <p style="padding-left: 40px;">Invoice Date: Jan 2000</p> <p><b>FindWord</b> will locate the highlighted value:</p> <p style="padding-left: 40px;">Invoice Date: Jan <b>2000</b></p> <p><b>MergeWordLf</b> will consolidate it with the text “Jan”:</p> <p style="padding-left: 40px;">Invoice Date: <b>Jan 2000</b></p> <p>...so that the <b>UpdateField</b> action will save the entire value, “Jan 2000”.</p>
See also...	MergeWordRt

## MergeWordRT

Actions Library	Locate.rra/.rrx
Description	Merges the located word with one or more words to the right, on the same line.
Parameter	An <i>Integer</i> indicating the number of words to be merged.
Level	Field level only.
Returns	Always <i>True</i> .
Example	<p>WordFind(Jan) <b>MergeWordRt(1)</b> UpdateField()</p> <p>Given the following sequence:</p> <p style="padding-left: 40px;">Invoice Date: Jan 2000</p> <p>The result of <b>WordFind</b> is to locate the highlighted word:</p> <p style="padding-left: 40px;">Invoice Date: <b>Jan</b> 2000</p> <p><b>MergeWordRt</b> will consolidate that value with the year value to its right:</p> <p style="padding-left: 40px;">Invoice Date: <b>Jan 2000</b></p> <p><b>UpdateField</b> saves the entire value, "Jan 2000".</p>
See also...	MergeWordLf



## MinLength

Actions Library	Locate.rra/.rrx
Description	Compares the number of characters in the located word to a minimum number you supply as the parameter.
Parameter	An <i>Integer</i> specifying the minimum number of characters the word can contain.
Level	Field level only.
Returns	<i>False</i> if the parameter is not <i>Numeric</i> , or if the actual number of characters is less than the parameter. Otherwise, <i>True</i> .
Example	If the located word's <i>recognized</i> value is: ABC1  <b>MinLength(4)</b> returns <i>True</i>  <b>MinLength(3)</b> returns <i>True</i>  <b>MinLength(6)</b> returns <i>False</i> .
See also...	MaxLength

## RegExFind

Actions Library	Locate.rra/.rrx
Description	Uses the parameter's Regular Expression to locate a word with a particular value on the current <i>source</i> page.  If successful, the action updates the page's Fingerprint file (.cco) with the word's location.
Parameter	A Regular Expression made up of valid Regular Expression characters.
Level	Field.
Returns	False if the page's Fingerprint file (.cco) is not available or if it is empty. Otherwise, <i>True</i> .
Example	<b>RegExFind("Date")</b> GoRightWord(1) IsDateValue() UpdateField()  This sequence tries to find a date on the current page. If successful, it updates the bound field's <b>Text</b> property with the date.

## RegExFindNext

## Locate Actions

---

Actions Library	Locate.rra/.rrx
Description	Uses the parameter's Regular Expression to locate a word with a particular value and update the page's Fingerprint file (.cco) with the word's location.  <i><b>Important!</b></i> This is an immediate follow-up to a <b>RegExFind</b> action.
Parameter	Regular Expression with the "next" target word's value, and consisting of valid Regular Expression characters.
Level	Field.
Returns	False if the page's Fingerprint file (.cco) is not available or is empty. Otherwise, True.
Example	RegExFind("ItemID") RegExFindNext("Desc")  In this sequence, the first action looks for <i>ItemID</i> . If the search fails, the follow-up action looks for <i>Desc</i> .

## RegExFind\_InZone

Actions Library	Locate.rra/.rrx
Description	Uses the parameter's Regular Expression to locate a word with a particular value within a zoned field or block.  The action then updates the page's Fingerprint file (.cco) with the word's location.
Parameter	Regular Expression with the "next" target word's value, and consisting of valid Regular Expression characters.
Level	Field.
Returns	False if the page's Fingerprint file (.cco) is not available or is empty. Otherwise, True.
Example	RegExFind_InZone("ItemID") RegExFindNext("Desc")  In this sequence, the first action looks for <i>ItemID</i> . If the search fails, the follow-up action looks for <i>Desc</i> .

## RegExFindNext\_InZone

Actions Library	Locate.rra/.rrx
Description	<p>Uses the parameter's Regular Expression to locate a word with a particular value within a zoned field or block.</p> <p>The action then updates the page's Fingerprint file (.cco) with the word's location. <b>Important!</b> This is an immediate follow-up to a RegExFind_InZone action.</p>
Parameter	Regular Expression with the "next" word's value, consisting of valid Regular Expression characters.
Level	Field.
Returns	False if the source page's Fingerprint file (.cco) is unavailable or empty. Otherwise, True.
Example	<pre>RegExFind_InZone("Price") RegExFindNext_InZone("LineTotal")</pre> <p>In this example, the first action locates <i>Price</i> inside a zone such as a Line Item Table, while the follow-up action finds the <i>LineTotal</i> word.</p>

## ScanRT

Actions Library	Locate.rra/.rrx
Description	Looks for a word or words in positions that are slightly above or below the line on which the current word rests.
Parameter	<i>Numeric</i> value of the number of words to be evaluated to the right of the current word.
Level	Field level only.
Returns	<i>True</i> if a word is found. Otherwise, <i>False</i> .
Example	WordFind(Number) <b>ScanRight(1)</b>  This sequence first finds “Number” on the current page, then moves one word to the right as it searches for a value.  To compensate for the possibility that this value may be printed slightly above or below the line on which “Number” was printed, the <b>ScanRight</b> action expands the area in which the target value can reside.
See also...	GoRightWord

## SelectSnippet

Actions Library	Locate.rra/.rrx
Description	<p>Used in conjunction with directional actions, this action will populate a Snippet field with the <i>recognized</i> value of the located word.</p> <p>This action is usually the last rule of a <b>Locate</b> RuleSet, and is included as a last resort when the probably location of the field has been found by finding the label, but no likely values have been found.</p>
Parameters	<p>1.) The character that is to appear in the Snippet field if a value is not available. “~” is the default.</p> <p>2.) The magnification of the value’s image. “1” = 100% and is the default.</p>
Level	Field level only.
Returns	Always <i>True</i> .
Example	<p>FindKeyInList(InvNum) GoRightWord(1) <b>SelectSnippet(~,1)</b></p> <p>This sequence first tries to locate an <i>APT Number</i> keyword in the current page. If successful, the next action attempts to lock onto a word to the right of the located word.</p> <p>If that word is present, the <b>SelectSnippet</b> action will place the image of the word’s <i>recognized</i> value into the Snippet of the applicable <b>Field</b> object. The Data Entry operator can then determine if the Snippet contains the correct value and can enter the data into the accompanying Data Edit field in the <i>Data Entry Panel</i>.</p>

### SetRect

Actions Library	Locate.rra/.rrx
Description	Specifies the position of the image associated with the current field.
Parameters	Four coordinates designating the rectangle's size and location: X1, X2, Y1 and Y2.
Level	Field level only.
Returns	Always <i>True</i> .
Example	<p>SetRect(0,0,100,200)</p> <p>This action creates a blue rectangle at coordinates 0, 0, 100 and 200 when a user stops on this field in the verify task.</p> <p>This action is useful if you wish to highlight a specific area of the image where you know your value resides.</p>

### UpdateDCOField

Actions Library	Locate.rra/.rrx
Description	<p>Updates the size and position coordinates of the <b>Field</b> object of the Document Hierarchy representing the field identified by the parameter.</p> <p><b>Important!</b> This action does not update a field's <b>Text</b> value. Instead, it modifies the size and location parameters of a field or zone. Do not confuse this action with the <b>UpDateField</b> action, which updates <b>Text</b> values.</p>
Parameter	<i>String</i> value of the target field's name, as a <b>Field</b> object of the Document Hierarchy.
Level	Field.
Returns	False if: the <i>source</i> page's Fingerprint file (.cco) is not available or is empty; if the target field cannot be found; or if there is no information about the target field's starting and ending sizes and locations. Otherwise, True.
Example	UpdateDCOField(Preparer_Name)

## UpdateField

Actions Library	Locate.rra/.rrx
Description	<p>Updates the appropriate field in the current page's Data file with the recognized (and possibly formatted!) value of the located word.</p> <p><b>Important!</b> An <i>entered</i> value that the <b>UpdateField</b> action places in a Data file becomes a <i>captured</i> value, and can be processed by <b>Validation</b> and <b>Export</b> RuleSets.</p>
Parameters	None
Level	Field level only.
Returns	Always <i>True</i> .
Example	<pre>FindKey(Date) GoRightWord(1) IsDateValue() <b>UpdateField()</b></pre> <p>The first action in the sequence finds a word that identifies the <i>Date Field</i> object of the current page. This is the field's <i>static</i> value – probably its title.</p> <p>The next action moves right one word to locate the field's <i>entered</i> value – a recognized date such as 12/31/2002. The third action checks to be sure the value has an acceptable Date format.</p> <p>The concluding <b>UpdateField</b> action takes place only if the others are successful. It adds the field's <i>entered</i> value to the current page's Data file, where it is a <i>captured</i> value awaiting the attention of upcoming rules with <b>Validate</b> and <b>Export</b> actions.</p>

### ValueInField

Actions Library	Locate.rra/.rrx
Description	Checks if the parameter you enter is within the value of the current field represented by the bound <b>Field</b> object of the Document Hierarchy.
Parameters	The <i>precise</i> value that is to be matched with some or all of the value in the current field.
Level	Field level only
Returns	<i>False</i> if no match occurs. Otherwise, <i>True</i> .
Example	ValueInField(Invoice)

### ValueInField\_Fuzzy

Actions Library	Locate.rra/.rrx
Description	Checks if there is a “fuzzy” match of the parameter’s value with the value in the current field.
Parameters	String value to be matched to the current field’s value, using fuzzy matching procedures.
Level	Field level only
Returns	<i>False</i> if no match occurs. Otherwise, <i>True</i> .
Example	ValueInField_Fuzzy(Invoice)



## ValueInField\_RegEx

Actions Library	Locate.rra/.rrx
Description	Checks if the Regular Expression you specify as the parameter is equivalent to the value of the current field.
Parameters	The Regular Expression to search for in the current field's value.
Level	Field level only.
Returns	<i>False</i> if no match occurs. Otherwise <i>True</i> .
Example	ValueInField_RegEx([\^\b\s\n\r]Inv[oOO][iItl1]ce[\b\s]*)

## WordFind

Actions Library	Locate.rra/.rrx
Description	Uses a keyword approach to search for the <i>first</i> time the word you supply as a parameter appears in the <i>recognized</i> page.
Parameter	The word that is the object of the search.  This case-sensitive value can contain up to 255 characters – <i>alpha</i> , <i>numeric</i> and <i>alphanumeric</i> – and can include multiple words (see the example.)
Level	Field level only.
Returns	<i>True</i> if the word's value is located. Otherwise, <i>False</i> .
Example	<b>WordFind(Sales Tax)</b> GoRightWord(1) IsCurrency() UpdateField  In this example, the <b>WordFind</b> action looks for the first occurrence of "Sales Tax" within the current page.
See also...	WordFindNext, FindKeyList

## WordFind\_InZone

Actions Library	Locate.rra/.rrx
Description	<p>Searches a zoned field or block within the Fingerprint file (.cco) of the current <i>source</i> page for a word with the value you enter as a parameter.</p> <p>If the search is successful, the action updates the file with the word's location.</p>
Parameters	<i>String</i> value of the word's value.
Level	Page or Field.
Returns	False if the <i>source</i> page's Fingerprint file (.cco) is not available or is empty. Otherwise, True.
Example	<pre>WordFind_InZone(Sub-Total) GoRightWord(1) IsCurrency()</pre>

## WordFindNext

Actions Library	Locate.rra/.rrx
Description	Finds the next instance of the keyword you supply as the parameter in this action <i>and</i> in a preceding <b>WordFind</b> action.
Parameter	The word that is the object of the continuing search.
Level	Field level only.
Returns	<i>True</i> if the parameter is located. Otherwise, <i>False</i> .
Example	<pre>WordFind(Total) <b>WordFindNext(Total)</b> <b>WordFindNext(Total)</b> GoRightWord(1) IsCurrency() UpdateField</pre> <p>In this sequence, the <b>Locate</b> rule locks onto the third instance of the word "Total" in the current page.</p>
See also...	WordFind, FindKeyList

## WordFindNext\_InZone

Actions Library	Locate.rra/.rrx
Description	<p>Searches a zone within the Fingerprint file (.cco) of the current <i>source</i> page for a word with the value you enter as a parameter.</p> <p>If the search is successful, the action updates the file with the word's location. <b><i>Alert!</i></b> This is a follow-up action to the <b>WordFind_InZone</b> action.</p>
Parameters	<i>String</i> value of the "next" word's value.
Level	Field.
Returns	False if the <i>source</i> page's Fingerprint file (.cco) is not available or is empty. Otherwise, True.
Example	<pre>WordFind_InZone(ItemID) WordFindNext_InZone(Desc)</pre>

## WordFind\_Offset

Actions Library	Locate.rra/.rrx
Description	Generates an ImageOffset value based on the difference between keywords on an IDENTIFIED live image and the IDENTIFIED Fingerprint image.
Parameters	<i>String</i> value of a keyword that the action is to find on both images
Level	Page.
Returns	Always True.
Example	<pre>WordFind_Offset(Invoice_Number)</pre>

### LookUp Actions

**LookUp** actions can accompany **Validate** actions as part of a **Validation** RuleSet, or can be part of an independent RuleSet Type you define.

- ✓ The *Taskmaster for APTs* application supplies a skeleton LookUp database: you'll find **InvoiceLook.mdb** in the application's **Process** directory. You can alter a stock database to suit your needs, or create a new LookUp database.

### ClearLookUpResults

Actions Library	Lookup.rra/.rrx
Description	Clears the stored results returned from a previous lookup.
Parameter	None
Level	All
Returns	Always True.
Example	<pre>OpenConnection(InvLook) SelectSQL("SELECT NAME, ADDRESS FROM Vendor;") PopulateWithResult(1) <b>ClearLookupResults()</b></pre> <p>Here, <b>ClearLookupResults()</b> clears the stored results of the previously retrieved Vendor Name and Address data.</p>

## CloseConnection

Actions Library	Lookup.rra/.rrx
Description	<p>Closes an open connection to your Lookup database.</p> <p>Usually, this action is placed in a RuleSet that is separate from the RuleSet that opens the connection and performs the lookups.</p> <p><b>LookUpDBC</b> RuleSet, for example, is run at the Batch level after all lookups have been performed.</p>
Parameter	None
Level	All
Returns	Always <i>True</i> .
Example	<pre>OpenConnection(InvoiceLook) LookUpReturnValue(hs_HostID,Host,hs_RefID) PopulateWithResult() <b>CloseConnection()</b></pre> <p>This sequence opens the database connection, performs a lookup, then closes the database connection.</p> <p>This action is usually part of a separate RuleSet that prevents the need to repeatedly open the connection to the Lookup database. (You can open the connection once in the first RuleSet, perform the required lookups, then close the connection once in the second RuleSet.)</p>
See also...	OpenConnection

## ExecuteSQL

Actions Library	Lookup.rra/.rrx
Description	<p>Executes a SQL statement.</p> <p>This action does not use stored field values, nor does it populate <b>Field</b> objects. It will return <i>True</i> if the SQL statement executes successfully, <i>False</i> if it does not.</p>
Parameter	The SQL statement you want to execute.
Level	All
Returns	<i>True</i> if the SQL statement executes successfully; <i>False</i> if it does not.
Example	<pre>OpenConnection(InvoiceLook) ExecuteSQL(INSERT INTO Vendor (CompanyCode,Type) VALUES ('MQSW','New'))</pre> <p>This sequence opens a connection to the <b>InvoiceLook</b> database. Next, it inserts values into the <b>CompanyCode</b> and <b>Type</b> columns of the <b>Vendor</b> table.</p>

## LookupCurrentValue

Actions Library	Lookup.rra/.rrx
Description	Queries an <i>open</i> LookUp database to see if a <b>Field</b> object's <i>captured</i> value is also a value in the database.
Parameters	<p>Four clauses of a SQL SELECT statement (see the example):</p> <ul style="list-style-type: none"> <li>• SELECT clause: designates the column or columns to be checked.</li> <li>• FROM clause: identifies the source table within the previously opened database.</li> </ul> <p>WHERE clause: specifies the matching criteria.</p> <p>AND clause (optional): limits the statement's scope.</p>
Level	Field level only.
Returns	<i>False</i> if the action is not applied at the Field level; if the database connection was not opened; or the captured value is not in the database. Otherwise, <i>True</i> .
Example	<pre>OpenConnection(InvoiceLook) LookupCurrentValue(PONumber, Vendor, PO   Number) CloseConnection()</pre> <p>This sequence looks in the <b>Vendor</b> table of the <b>InvoiceLook</b> database to see if the <i>captured</i> value of the current <b>Field</b> object is also a value in the database.</p> <p>The <b>LookupCurrentValue</b> action uses its parameters to assemble a SQL SELECT statement with these clauses:</p> <pre>SELECT PONumber (column) FROM Vendor (table) WHERE PONumber (captured value)</pre> <p>The action returns <i>True</i> if the <b>PONumber</b> column in the <b>Vendor</b> table contains a value equal to the <i>captured</i> value of the current <b>Field</b> object.</p>
See also...	LookUpReturnValue

## LookupReturnValue

Actions Library	Lookup.rra/.rrx
Description	<p>Uses the <i>captured</i> value associated with the current <b>Field</b> object to determine if a table in an <i>open</i> LookUp database contains a corresponding value.</p> <p>If there is a match (<i>True</i>), the <b>PopulateWithResult</b> action updates the Data file of the current page with the value.</p>
Parameters	<p>Four elements of a SQL SELECT statement (see the examples):</p> <ul style="list-style-type: none"> <li>• SELECT clause: designates the target <b>column</b> or columns.</li> <li>• FROM clause: identifies the source <b>table</b> within the previously opened database.</li> <li>• WHERE clause: specifies the <b>matching criteria</b>.</li> </ul> <p>AND clause (optional): limits the statement's <b>scope</b>.</p>
Level	Field level only.
Returns	<i>False</i> if the action is not applied at the Field level; the database connection was not opened; or the <i>captured</i> value is not in the database. Otherwise, <i>True</i> .
Example	<pre>OpenConnection(InvoiceLook) LookupReturnValue(VendorName, Vendor, VendorID) PopulateWithResult(1, FALSE) CloseConnection()</pre> <p>This sequence looks in the <b>Vendor</b> table of the <b>InvoiceLook</b> database, for a <i>VendorID</i> value that matches the <i>captured</i> value of the current <i>VendorID</i> <b>Field</b> object.</p> <p>If the search is successful (<i>True</i>), the follow-up <b>PopulateWithResult</b> action assigns the <i>Vendor Name</i> value to the Data file of the current page, as the <i>VendorName</i> field's <i>captured</i> value.</p>
See also...	LookupCurrentValue, PopulateWithResult



## OpenConnection

Actions Library	Lookup.rra/.rrx
Description	Uses the Data Source Name (DSN) or Connection String you provide as the parameter to open a connection to your LookUp database.
Parameter	<p>The <i>String</i> value of the Data Source Name or Connection String.</p> <p><b>Alert!</b> If the action is establishing a connection with an Oracle database, or a SQL Server database using SQL Authentication, be sure to expand the DSN parameter by adding the correct Provider, user ID and Password. For example:</p> <p><b>Oracle:</b></p> <pre>OpenConnection( PROVIDER=ODBCORACLE ; DSN=1040Look ; CATALOG= ; DBNTA= ; UID=Admin ; PWD=Admin ; )</pre> <p><b>SQL Server Authentication:</b></p> <pre>OpenConnection(PROVIDER=ODBCMSSQL;DSN=1040Look;CATALOG=;DBNTA=;UID=Admin;PWD=Admin;)</pre>
Level	All
Returns	<i>True</i> if the action results in a connection to the database; otherwise, <i>False</i> .
Examples	<pre>OpenConnection(InvoiceLook)</pre> <p>or</p> <pre>OpenConnection(PROVIDER=ODBCMSSQL; DSN=1040Look; CATALOG=; DBNTA=; UID=Admin; PWD=Admin;)</pre>
See also...	CloseConnection

## PopulateWithResult

Actions Library	Lookup.rra/.rrx
Description	<p>Populates a <b>Field</b> object with a database value retrieved by a <b>SelectSQL</b> or <b>LookupReturnValue</b> action.</p> <p>This action allows multiple rules to populate multiple <b>Field</b> objects with data from a <i>single</i> database record (see the example below.)</p>
Parameters	<p>1.) A number <i>n</i> indicating which value in a record retrieved by an earlier <b>SelectSQL</b> or <b>LookupReturnValue</b> action is to be assigned to the current Field object (and added to the Data file of the current page.)</p> <p>“1” refers to the first column in a recordset, “2” refers to the second column, etc. (see the example.)</p> <p>2.) <i>True</i> or <i>False</i>. <i>True</i> causes the action to <i>fail</i> if the action returns a recordset with multiple lines. <i>False</i> permits the action to accept a recordset with multiple lines but to use values in the <i>first</i> record of the recordset.</p>
Level	Field level only.
Returns	<p><i>True</i> if the second parameter is “True” and a previous <b>SelectSQL</b> (or <b>LookupReturnValue</b>) action finds a recordset with only one record.</p> <p><i>True</i> if the second parameter is “False” and a previous <b>SelectSQL</b> (or <b>LookupReturnValue</b>) action finds a recordset with one or more records.</p> <p>Otherwise, <i>False</i>.</p>

PopulateWith Result (continued)

<p>Example</p>	<p>(Field#1)</p> <p>OpenConnection(InvoiceLook)          SelectSQL("Select * From Vendor Where VendorID = %s;",VendorID)   <b>PopulateWithResult(1,FALSE)</b></p> <p>(Field#2)</p> <p>PopulateWithResult(2,FALSE)</p> <p>In the example, the <b>SelectSQL</b> action of the RuleSet applied to <i>Field#1</i> retrieves the recordset (if it exists). The <b>PopulateWithResult</b> action places the value of the <i>first</i> record's <i>first</i> column into the field where the rule has been applied.</p> <p>The <b>PopulateWithResult</b> action of a rule applied to <i>Field #2</i> populates the field with the value of the <i>first</i> record's <i>second</i> column.</p> <p><i>False</i> means that the action can accept a recordset with multiple records but will extract value(s) from the first record only.</p>
<p>See also...</p>	<p>LookupCurrentVaue, LookupReturnValue, SelectSQL</p>

## MC\_Identify Actions

These actions are used by the *Taskmaster for Medical Claims* application to identify red *HCFA-1500* and red *UB-92* forms.

### AutoField

Actions Library	MC_Identify.rra/.rrx
Description	Identifies red <i>HCFA-1500</i> or red <i>UB-92</i> forms.  The action must be placed after a rule's <b>SetMaxTolerantDistance</b> , <b>SetFormType</b> , <b>SetFSPathName</b> , and <b>SetWritePosFile</b> actions (see the example below.)
Parameters	None
Level	Page level only.
Returns	<i>False</i> if the action is not applied at the Page level; otherwise, <i>True</i> .
Example	SetMaxTolerantDistance(60) SetFormType(0) SetFSPathName(\Datacap\MClaims\process\HCFA.fs) SetWritePosFile(True) <b>AutoField()</b>

### FindFields

Actions Library	MC_Identify.rra/.rrx
Description	Sets up a Data file (.xml) for the current page, and supplies the Data file with field position information.  Typically, this action is part of a <i>Medical Claims ID_PageFix</i> ruleset.
Parameters	None
Level	Page level only.
Returns	<i>False</i> if rule with this action is not bound to a <b>Page</b> object of the Document Hierarchy, or if a <i>source</i> page represented by the Page object is unavailable. Otherwise, returns <i>True</i> .
Example	SetMaxTolerantDistance(60) SetFormType(0) SetFSPathName(\Datacap\MClaims\process\HCFA.fs) SetWritePosFile(True) <b>FindFields()</b>

## ReadDCOSetup

Actions Library	MC_Identify.rra/rrx
Description	An important preliminary action that identifies objects at each level of the Document Hierarchy (aka the setup DCO).  The action also reads values assigned to an object's properties and variables.
Parameters	<i>String</i> value of the Document Hierarchy's file name.  Typically, the file name is the Application ID with an .xml extension: "HCFA.xml", for example, is the name of the <i>HCFA</i> application's Document Hierarchy file.
Level	All, but usually the Batch level.
Returns	Always True.
Example	ReadDCOSetup(HCFA.xml)

## SetFormType

Actions Library	MC_Identify.rra/rrx
Description	Sets the Form Type value that will be used by <b>Autofield</b> .
Parameters	<i>String</i> value indicating the Form Type:  For <i>HCFA-1500</i> : <b>0</b> or <b>hcfa</b>  For <i>UB-92</i> : <b>1</b> or <b>ub92</b>
Level	All
Returns	<i>False</i> if the parameter is invalid; otherwise, <i>True</i> .
Example	SetMaxTolerantDistance(60) <b>SetFormType(0)</b> SetFSPathName(\Datacap\MClaims\process\HCFA.fs) SetWritePosFile(True) AutoField()

### SetFSPathName

Actions Library	MC_Identify.rra/.rrx
Description	Sets the name and path of the FormSpec that will be used by <b>AutoField</b> .
Parameters	The full name and path of the FormSpec file (.fs).
Level	All
Returns	<i>False</i> if the parameter is invalid; otherwise, <i>True</i> .
Example	<pre>SetMaxTolerantDistance(60) SetFormType(0) <b>SetFSPathName(\Datacap\MClaims\process\HCFA.fs)</b> SetWritePosFile(True) AutoField()</pre>

### SetMaxTolerantDistance

Actions Library	MC_Identify.rra/.rrx
Description	Sets the tolerance level that <b>AutoField</b> will use to match <i>HCFA-1500</i> or red <i>UB-92</i> forms (depending on the form specified using the <b>SetFormType</b> action).
Parameters	Maximum Tolerant Distance – <i>Integer</i> from 1 (Lowest Tolerance) to 100 (Highest Tolerance).
Level	All
Returns	<i>False</i> if the parameter is not an integer between 1 and 100; otherwise, <i>True</i> .
Example	<pre><b>SetMaxTolerantDistance(60)</b> SetFormType(0) SetFSPathName(\Datacap\MClaims\process\HCFA.fs) SetWritePosFile(True) AutoField()</pre>

## SetWritePosFile

Actions Library	MC_Identify.rra/.rrx
Description	Turns on POS file creation.
Parameters	<i>String</i> value of “True” or “1” (without the quotation marks) to activate POS file creation.
Level	All
Returns	Always <i>True</i> .
Example	SetMaxTolerantDistance(60) SetFormType(0) SetFSPathName(\Datacap\MClaims\process\HCFA.fs) <b>SetWritePosFile(True)</b> AutoField()

## MC\_Validation Actions

These actions are used by the *Taskmaster for Medical Claims* applications to parse, modify, and validate data.

- ✓ *Parsing* breaks up a value into logical components. Parsing the value assigned to a “Name” field typically breaks up the full value into sub-values for its First Name, Last Name and Middle Initial components.

### AddCenturyTo2DigitYear

Actions Library	MC_Validation.rra/.rrx
Description	Converts two-digit <b>Year</b> values to four-digit <b>Year</b> values.  All dates are assumed to be before today’s date, with a format of <b>mmddy</b> . If today is 150507, and this action is applied to a field with a value of 221095, the date is assumed to be 22101995.
Parameters	None
Level	Field level only.
Returns	<i>False</i> if value is not a valid date in the mmddy format, or if the action is not applied at the Field level. Otherwise, <i>True</i> .
Example	AddCenturyTo2DigitYear()

### CalculateResult

Actions Library	MC_Validation.rra/.rrx
Description	Determines if the calculation provided by the parameter is <i>True</i> or <i>False</i> .
Parameters	<i>String</i> value of the equation that is the basis for the action’s calculation.  Field values can be used in the string by adding the Field name surrounded by single quotation marks ( ‘ ’ ).
Level	Field level only.
Returns	<i>False</i> if the calculation results in a validation error, or if the action is not applied at the Field level. Otherwise, <i>True</i> .
Example	CalculateResult(‘29Amt_Pd+’30BalDue=’28TotChg’)



## CalculateUB

Actions Library	MC_Validation.rra/.rrx
Description	<p>A <i>UB92</i> action that determines if the amounts in the following fields sum correctly:</p> <p>47attchg,47bttchg,47cttchg,47dttchg,47ettchg,47fittchg,  47gttchg,47httchg,47jttchg,47kttchg,47mттchg,47nttchg,  47pttchg,47qттchg,47rttchg,47sttchg,47ттtchg,47uttchg,  47vттchg,47wттchg,47xttchg,47yttchg,47zттchg</p>
Parameters	None
Level	Field level only
Returns	<i>False</i> if the validation fails or if the action is not applied at the Field level. Otherwise, <i>True</i> .
Example	CalculateUB()

### CommonParseAddress

Actions Library	MC_Validation.rra/.rrx
Description	<p>Parses addresses in the following fields into appropriate subfields:</p> <p><i>HCFA</i> – Facility Address (Field 32), or Physician Address (Field 33)</p> <p><i>UB92</i> – Patient Address (Field 13), Facility Address (Field 1), or Responsible Party Address (Field 38).</p>
Parameters	<p>For <i>HCFA</i>: <i>String</i> value of:</p> <ol style="list-style-type: none"> <li>1) <b>HCField32Object</b>: for parsing Facility Address field (field 32).</li> <li>2) <b>HCField33Object</b>: for parsing Physician Address field (field 33).</li> </ol> <p>For <i>UB92</i>: <i>String</i> value of:</p> <ol style="list-style-type: none"> <li>1) <b>UBField13Object</b>: for parsing Patient Address field (field 13).</li> <li>2) <b>UBField1Object</b>: for parsing Facility Address field (field 1).</li> <li>3) <b>UBField38Object</b>: for parsing responsible party Name and Address field (field 38).</li> </ol>
Level	Page level only.
Returns	<i>False</i> if the parameter is invalid, or if the action is not at the Page level. Otherwise, <i>True</i> .
Example	CommonParseAddress(HCField32Object)

## CommonValAddress

Actions Library	MC_Validation.rra/.rrx
Description	<p>Validates address values:</p> <ol style="list-style-type: none"> <li>1) <b>First Name:</b> value can start with “Ms”, “Mr”, “Miss”, “Dr” (Salutations). The remaining values have to be <i>alphanumeric</i> with no special characters. Punctuation is allowed only after the salutation.</li> <li>2) <b>Last name:</b> same requirements for first name</li> <li>3) <b>Street:</b> <i>alphanumeric</i>, upper or lower case. Can include punctuation and the # character.</li> <li>4) <b>City:</b> characters from A to Z, upper or lower case, comma, period, space, and &amp; character.</li> <li>5) <b>State:</b> must be <i>two alphanumeric</i> characters.</li> <li>6) <b>Zip Code:</b> must be between <i>five</i> and <i>nine numeric</i> characters. This value is checked against the <b>State</b> value (above).</li> <li>7) <b>Phone Number:</b> The Area Code is checked against <b>State</b> and <b>Zip Code</b> values (above).</li> </ol>
Parameters	Comma-delimited <i>String</i> with a list of field names with address values to be validated.
Level	Page level only.
Returns	<i>False</i> if not placed at Page level; otherwise, <i>True</i> .
Example	<p>CommonValAddress(Insured,4InsFNam,4InsLNam, 7IAddStr,,7IAddCty,7IAddSta,7IAddZip,)</p> <p>or</p> <p>CommonValAddress(Description,12plname,12pfname, 13paddr1,13paddr2,13padcit,13padsta,13padzip,)</p>

### ConvertHyphen

Actions Library	MC_Validation.rra/.rrx
Description	Remove spaces and commas, converts hyphens. “1,2,3,4” becomes “1234”; “1-2-3” becomes “123”.
Parameters	None
Level	Field level only.
Returns	<i>False</i> if not at the Field level; otherwise, <i>True</i> .
Example	ConvertHyphen()

### InheritSnippets

Actions Library	MC_Validation.rra/.rrx
Description	Assigns the snippet position information of the current <b>Field</b> object to the <b>Field</b> objects specified in the parameter.
Parameters	Names of the fields that will inherit the same snippet information as the current <b>Field</b> object. For example: <i>2paLname, 2PaFname, 2PaMInit</i>
Level	Field level only
Returns	<i>False</i> if the action is not applied at the Field level, or if a parameter is incorrect. Otherwise, <i>True</i> .
Example	InheritSnippets(2paLname, 2PaFname,2PaMInit)

**MC\_ReadZones**

Actions Library	MC_Validation.rra/.rrx
Description	Copies all zone positions to the current <i>page</i> .  <i>Alert!</i> This action handles Autofield-based OMR zone detection for a <i>Medical Claims</i> application. The action is <i>not</i> compatible with standard rules-based OMR zone detection procedures.
Parameters	None
Level	Page level only.
Returns	<i>False</i> if not applied at the Page level. Otherwise, <i>True</i> .
Example	MC_ReadZones()

**Parse31aPhSig**

Actions Library	MC_Validation.rra/.rrx
Description	Parses field <i>31aPhSig</i> of the <i>HCFA</i> application.
Parameters	None
Level	Field level only.
Returns	Always <i>True</i> .
Example	Parse31aPhSig()

**Parse58ainsnm**

Actions Library	MC_Validation.rra/.rrx
Description	Parses field <i>58ainsnm</i> of the <i>UB92</i> application.
Parameters	None
Level	Field level only.
Returns	Always <i>True</i> .
Example	Parse58ainsnm()

**Parse58binsnm**

Actions Library	MC_Validation.rra/.rrx
Description	Parses field <i>58binsnm</i> of the <i>UB92</i> application.
Parameters	None
Level	Field level only.
Returns	Always <i>True</i> .
Example	Parse58binsnm()

**Parse58cinsnm**

Actions Library	MC_Validation.rra/.rrx
Description	Parses field <i>58cinsnm</i> of the <i>UB92</i> application.
Parameters	None
Level	Field level only.
Returns	Always <i>True</i> .
Example	Parse58cinsnm()

**Parse82name**

Actions Library	MC_Validation.rra/.rrx
Description	Parses field <i>82name</i> of the <i>UB92</i> application.
Parameters	None
Level	Field level only.
Returns	Always <i>True</i> .
Example	Parse82name()

**Parse83aname**

Actions Library	MC_Validation.rra/.rrx
Description	Parses field <i>83aname</i> of the <i>UB92</i> application.
Parameters	None
Level	Field level only.
Returns	Always <i>True</i> .
Example	Parse83aname()

**Parse83bname**

Actions Library	MC_Validation.rra/.rrx
Description	Parses field <i>83bname</i> of the <i>UB92</i> application.
Parameters	None
Level	Field Level Only.
Returns	Always <i>True</i> .
Example	Parse83bname()

### ParseLastFirstIniNames

Actions Library	MC_Validation.rra/.rrx
Description	<p>Parses the name information in the first line of an <b>address superfield</b>.</p> <p>The action will parse the full name's value into the Last, First, and Middle Name/Initial fields specified by the parameter.</p>
Parameters	<p>Three comma separated parameters:</p> <p>The name of the <i>Last Name</i> <b>Field</b> object.</p> <p>The name of the <i>First Name</i> <b>Field</b> object.</p> <p>The name of the <i>Middle Name</i> or <i>Middle Initial</i> <b>Field</b> object.</p>
Level	Field level only.
Returns	<i>False</i> if the parameter values are invalid; Otherwise, <i>True</i> .
Example	ParseLastFirstIniNames(2PaLName,2PaFname,2paMinit)

### ParseUB\_Eighties

Actions Library	MC_Validation.rra/.rrx
Description	<p>Special action to parse fields 82 (Attending Physician ID) and 83 (Other Physician ID) of the <i>UB92</i> application.</p> <p>Because each field has two values on the <i>UB-92</i> form – a physician's ID <b>and</b> Name - this action parses the full value and assigns the name to one field and the ID to another.</p>
Parameters	<p>Three comma-separated <i>String</i> parameters. 1) The name of the <b>Field</b> object that is to be parsed; 2) The name of the <b>Field</b> object that will contain the physician's Name value; 3) The name of the <b>Field</b> object that will contain the physician's ID value.</p> <p>Examples:</p> <ol style="list-style-type: none"> <li>1) 82apid,82name,82id</li> <li>2) 83aopid,83aname,83aid</li> <li>3) 83bopid,83bname,83bid</li> </ol>
Level	Field level only.
Returns	Always, <i>True</i> .
Example	ParseUB_Eighties(82apid,82name,82id)



## PopulateFieldPositionFromSetup

Actions Library	MC_Validation.rra/.rrx
Description	Transfers a field's zone positions from the Document Hierarchy to a task's Page file.
Parameters	None
Level	Field level only.
Returns	<i>False</i> if position information is missing from the <b>Field</b> object in the document hierarchy. Otherwise, <i>True</i> .
Example	PopulateFieldPositionFromSetup()

## PopulateFromField

Actions Library	MC_Validation.rra/.rrx
Description	Copies the value from the field specified by the parameter into the current field.
Parameters	The name of the field whose value is to be assigned to the current field.
Level	Field level only.
Returns	<i>False</i> if the parameter is invalid, or if the action is not applied at the Field level. Otherwise, <i>True</i> .
Example	PopulateFromField(24aDtFr1)

## StripTrailingAlpha

Actions Library	MC_Validation.rra/.rrx
Description	Removes all alpha characters from the captured value, except any in the first character position.
Parameters	None
Level	Field level only
Returns	Always <i>True</i> .
Example	StripTrailingAlpha()

## TransformLI

Actions Library	MC_Validation.rra/.rrx
Description	Remaps the Line Item Table fields into a hierarchical structure.
Parameters	None.
Level	Document or Page.
Returns	True if the assignment of values for all fields in all lines of the Line Item Table was successful. False if the transfer of values for one or more fields fails.
Example	<p>Each line in the table is remapped into a repeating set of fields. These fields have a <i>parent</i> field with a unique ID. All data in a source field is retained in the new target field, with the exception of the new fields type and ID. The linear field structure is replaced with a structure based on a parent field within the claim page called DETAILS.</p> <p>Each set of fields in the <b>row</b> of table data is then placed within a LINEITEM type field: the ID of the field is patterned on the index of its insertion in the DETAILS field. The first Line Item field will be called LINEITEM1, the second LINEITEM2, etc. Each Line Item field contains an identical set of Field Type and ID as outlined in the values list below.</p> <p>As an example, the HCFA Line Item Table has twelve <i>24nDtf</i>r fields. During processing, the <b>TransformLI</b> action assigns recognized values for these fields to a <i>DateFrom</i> <b>Field</b> object that is a child of the Line Item Table's <i>Details</i> new parent <b>Field</b> object. Ultimately, 12 recognized values can be assigned to the <i>DateFrom</i> <b>Field</b> object.</p> <p>Values associated with rows of the <i>HCFA-1500</i> table will be assigned to the Field types below: (<i>n</i> is a value from 1 to 12):</p> <p>24aDtf<math>n</math> = DateFrom</p> <p>24aDtTon = DateThru</p> <p>24bPlac<math>n</math> = PlaceOfService</p> <p>24cTypen = TypeOfService</p> <p>24cEMG_<math>n</math> = EMG_C</p>

<p>Example (continued)</p>	<p>24dCPT_ <i>n</i> = CPT_Code                  24dMod<i>n</i> = Modifiers                  24eDiagn = DiagPointer                  24fChgs<i>n</i> = Charges&lt;br/&gt;                  24gDays<i>n</i> = Days_Units                  24hEPSt<i>n</i> = EPSD                  24iQual<i>n</i> = Qualifier                  24jRefID<i>n</i> = ReferenceId                  24iEMG_ <i>n</i> = EMG_I                  24jCOB<i>n</i> = COB                  24kLoc<i>n</i> = LocalUse                  24Infor<i>n</i> = Info&lt;br/&gt;                  Values associated with rows of the UB92/UB04 table will be assigned to these child <b>Field objects</b>: (<i>n</i> is a value from a <b>to z</b>):&lt;br/&gt;&lt;br/&gt;                  42&lt;b&gt;n&lt;/b&gt;revcd = RevCode&lt;br/&gt;                  43&lt;b&gt;n&lt;/b&gt;dsrt = Description&lt;br/&gt;                  44&lt;b&gt;n&lt;/b&gt;hcpcs = HCPCS&lt;br/&gt;                  44&lt;b&gt;n&lt;/b&gt;Mod = Modifiers&lt;br/&gt;                  45&lt;b&gt;n&lt;/b&gt;srvdt = ServiceDate&lt;br/&gt;                  46&lt;b&gt;n&lt;/b&gt;srvun = Units&lt;br/&gt;                  47&lt;b&gt;n&lt;/b&gt;ttchg = Charges&lt;br/&gt;                  48&lt;b&gt;n&lt;/b&gt;ncchg = NonCovered&lt;br/&gt;                  49&lt;b&gt;n&lt;/b&gt; = LocalUse&lt;br/&gt;&lt;br/&gt;                  &lt;b&gt;Note:&lt;/b&gt; This action will convert all pages in a document if called from a &lt;b&gt;Document&lt;/b&gt; object. The expected page types are &lt;b&gt;HCFA 1500&lt;/b&gt;, &lt;b&gt;UB92&lt;/b&gt;, and &lt;b&gt;UB04_Page&lt;/b&gt;</p>
----------------------------	--

### ValidateNPI

Actions Library	MC_Validation.rra/.rrx
Description	Validates aclaim's National Provider Identifier codes
Parameters	None
Level	Page or Field
Returns	False if the action cannot validate the code. Otherwise, True.
Example	ValidateNPI()

### ValidateStateMil

Actions Library	MC_Validation.rra/.rrx
Description	Checks to see if the value in the field represented by the bound <b>Field</b> object is a valid two-character State abbreviation.
Parameters	None
Level	Field level only.
Returns	<i>False</i> if the field's value does not contain two characters, or if the value is not a valid State abbreviation. Otherwise, <i>True</i> .
Example	ValidateStateMil()

### ValProcedureCode

Actions Library	MC_Validation.rra/.rrx
Description	Validates the <b>Procedure Code</b> fields of a <i>HCFA-1500</i> form.
Parameters	None
Level	Field level only
Returns	<i>False</i> if not applied at the Field level, or if the procedure code is invalid. Otherwise, <i>True</i> .
Example	ValProcedureCode()

## ValRequiredGroup

Actions Library	MC_Validation.rra/.rrx
Description	Checks that all fields in a designated group are filled with data. This action is used for <i>HCFA-1500</i> forms only.
Parameters	Names of the fields in the group.
Level	Field level only.
Returns	<i>False</i> if the parameters are invalid, or if any of the parameter fields does not contain data. Otherwise, <i>True</i> .
Example	<code>ValRequiredGroup(24aDtFr1,24aDtTo1, 24adCPT_1,24fChg1,24gdays1)</code>

## ocr\_s Actions

## RecognizeDocToPDF

Actions Library	ocr_s.rra/rrx
Description	<p>Converts all pages in a document to Adobe PDF format and places them in a PDF file (.pdf).</p> <p>Any rule containing this action <i>must</i> be applied to a <b>Document</b> object of the Document Hierarchy.</p>
Parameters	<p>A single Numeric Value between 1 and 4 where:</p> <p>1 = "Adobe PDF with image on text." The generated PDF file contains one image for each page in the document, and also contains the recognized characters underneath. Displaying the generated PDF file in a PDF reader results in a look that is very similar to the original document. Note: the text can be searched.</p> <p>2 = "Adobe PDF." The generated PDF file contains the recognized characters in the same positions as in the original. Displaying the generated PDF file in a PDF reader results in a look that is very similar to the original document. Note: the text can be searched.</p> <p>3 = "Adobe PDF with image substitutes." Similar to #2 (Adobe PDF), problematic recognition cases are handled by the inclusion of smaller images of smaller image snippets in the output file taken from the original image. Image snippets are also exported for the following cases: a) words containing suspect characters; b) words not approved by the checking subsystem ("non-dictionary" words); c) words containing rejection symbols; d) words containing missing symbol(s).</p> <p>4 = "Adobe PDF image only" The generated PDF file contains one image for each page in the document. The file will not contain any characters, so the text cannot be searched.</p>
Level	Document level only.
Returns	<i>False</i> if the action is not applied at the <b>Document</b> level of the Document Hierarchy, or if conversion is not successful. Otherwise, <i>True</i> .
Example	RecognizeDocToPDF(3)
See also...	RecognizeToPDF

**RecognizeFieldOCR\_S**

Actions Library	ocr_s.rra/rxx
Description	A field-level action that retrieves a <i>zoned</i> field's settings from the <i>OCR/S</i> tab of the <b>Recognition Options Setup</b> dialog, and uses these settings to recognize the field's value.
Parameters	None
Level	Field level only.
Returns	<i>False</i> if the action is not run at the Field level or if recognition fails. Otherwise, <i>True</i> .
Example	<p>TaxpayerSSN Rule 1 <b>RecognizeFieldOCR_S()</b></p> <p>In the example, the rule uses the action to retrieve and apply settings in the <i>OCR/S</i> tab of the <b>Recognition Options Setup</b> dialog...settings that have been previously assigned to a Document Hierarchy's <i>zoned</i> field.</p>

## RecognizeFieldVoteOCR\_S

Actions Library	ocr_s.rra/rrx
Description	<p>A field-level action that initiates a voting procedure that first uses specifications in the <i>OCR/S</i> tab of the <b>Recognition Options Setup</b> dialog to recognize the field's <b>characters</b>.</p> <p>When this action stores the result of recognition, it first determines if the corresponding <b>Field</b> object of the Document Hierarchy contains a value.</p> <p>If a value is present, the action compares the field's existing value with the recognition results - character by character. If a particular character's values match, the Confidence Rating for the character is raised to the maximum level. If the values do <i>not</i> match, the Confidence Rating for the character is lowered to the minimum.</p> <p>Note that when using this voting procedure, the second Recognition engine is <i>secondary</i> and its results are <i>never</i> assigned. Instead, the action changes the Confidence Ratings on the basis of results provided by the first Recognition engine.</p> <p>If there are no recognition results previous to this action, then it will act just like the <b>RecognizeFieldOCR_S</b> action.</p>
Parameters	None
Level	Field level only.
Returns	<i>False</i> if the action is not run at the Field level or if recognition fails. Otherwise, <i>True</i> .
Example	RecognizeFieldICR_C() RecognizeFieldVoteOCR_S()

## RecognizeOM\_OCR\_S

Actions Library	ocr_s.rra/rrx
Description	This field-level action is a shortcut to <b>Optical Mark Recognition</b> procedures that are carried out in response to settings in the <i>OCR/S</i> tab of the <b>Recognition Options Setup</b> dialog (Chapter 3).
Parameters	None
Level	Field level only.
Returns	<i>False</i> if the action is not run at the Field level or if recognition fails. Otherwise, <i>True</i> .
Example	RecognizeOM_OCR_S()



## RecognizePageFields2CCO\_OCR\_S

Actions Library	ocr_s.rra/rrx
Description	<p>Performs recognition on fields that have been designated for OCR/S recognition.</p> <p>The action then transfers the <b>Zonal OCR_S</b> recognition values to the page's CCO file.</p>
Parameters	None.
Level	Page level only.
Returns	<i>False</i> if the action is not run at the Page level or if recognition fails. Otherwise, <i>True</i> .
Example	RecognizePageFields2CCO_OCR_S()

## RecognizePageFieldsOCR\_S

Actions Library	ocr_s.rra/rrx
Description	<p>A page-level action that recognizes all fields on the page that have been configured for OCR/S recognition (see the <i>OCR/S</i> tab of the <i>Recognition Options Setup</i> dialog.)</p> <p>Individual <i>field-level</i> recognition actions will overwrite the results from this <i>page-level</i> action.</p> <p>This action will not recognize a <i>zoned</i> field if the <b>Skip Recognition</b> checkbox in the <i>OCR/S</i> tab of the <i>Recognition Options Setup</i> dialog has been selected.</p>
Parameters	None
Level	Page level only.
Returns	<i>False</i> if the action is not run at the Page level or if recognition fails. Otherwise, <i>True</i> .
Example	<p>ReadZones()</p> <p>RecognizePageFieldsOCR_S()</p>

## RecognizePageOCR\_S

Actions Library	ocr_s.rra/rrx
Description	<p>Refers to settings in the <i>OCR/S</i> tab of the <b>Recognition Options Setup</b> dialog to recognize <i>all</i> characters on a page, and populates the page's CCO file with the recognition results.</p> <p><b>Alert!</b> If a CCO file does not exist at the time this action is called, the action will create one.</p>
Parameters	None
Level	Page level only.
Returns	<i>False</i> if the action is not run at the <b>Page</b> level of the Document Hierarchy, or if recognition is not successful. Otherwise, <i>True</i> .
Example	<p>AnalyzeImage()          RotateImage()  <b>RecognizePageOCR_S()</b></p> <p>This sequence creates a CCO file for the current page, and checks to see if rotation of the image is needed.</p> <p><i>Full-page recognition</i> then takes place in response to settings in the <i>OCR/S</i> tab of the <b>Recognition Options Setup dialog</b>. The recognition results are stored in the CCO file.</p>
See also...	RecognizePageICR_C, CreateTextFile

## RecognizePageOCR\_S\_2TextFile

Actions Library	ocr_s.rra/rrx
Description	<p>Generates a Text file (.txt) that contains the raw recognition results for each page in the batch, and adds the file to the current batch.</p> <p>This action does <i>not</i> create or populate a page's CCO file with the recognition results.</p>
Parameters	None
Level	Page level only.
Returns	<i>False</i> if the action is not run at the <b>Page</b> level of the Document Hierarchy. Otherwise, <i>True</i> .
Example	RecognizePageOCR_S_2TextFile()
See also...	CreateTextFile

## RecognizeToFile\_OCR\_S

Actions Library	ocr_s.rra/rxx
Description	<p>Performs OCR recognition on the image of a source page, and stores the output of the OCR/S recogniton engine in a file.</p> <p>The output file is in one of 22 alternative formats (see the Parameters section below.). Because <i>Taskmaster</i> does not actually process the file in the format you specify, this action is useful primarily for debugging the engine, of if you need raw (unverified) OCR output in that format.</p>
Parameters	<p>The action requires a <i>Numeric</i> parameter from 1-22 to specify a combination of recognition targets and output formats. <b>Important!</b> "Image" refers to the image of the bound <b>Page</b> object of the Document Hierarchy. "Filename" is the string portion of a file's name that precedes its extension.</p> <p>1 - Recognize an image of the <b>Page</b> object of the DOCUMENT Hierarchy embedded in a PDF file. Output filename = the image filename plus a ".pdf" file extension (PDF ImageOn Text).</p> <p>2 - Recognize a single image with a ".pdf" extension. Output filename = the image filename with a ".pdf" extension (PDF).</p> <p>3 - Recognize a replacement image of the <b>Page</b> object of the Document Hierarchy, when the image is in a PDF file. Output filename = the replacement image filename with a ".pdf" extension (PDFImageSubst).</p> <p>4 - Recognize an image included in a Text file with a ".pdf" extension. Output filename = filename and a ".pdf" extension(PDF Text).</p> <p>5 - Recognize an HTML image of the bound <b>Page</b> object of the Document Hierarchy. Output filename = image filename with an ".html" extension (HTML140).</p> <p>6 - Recognize an image of the bound <b>Page</b> object of the Document Hierarchy in an Excel file. Output filename = image filename with an ".xls" extension (Excel2000.)</p> <p>7 - Recognize any image of the bound <b>Page</b> object of the Document Hierarchy in a Word2000 file with a ".doc" extension. Output filename = image filename and ".doc" extension (Word2000).</p> <p>8 - Recognize any image of the bound <b>Page</b> object of the Document Hierarchy in a WordML file. Output filename = image filename with a ".doc" extension (WordML).</p>

## RecognizeToFile\_OCR\_S (continued)

Parameters (continued)	<p>9 - Recognize any image of the bound <b>Page</b> object of the Document Hierarchy in a Word97 file. Output filename = image filename with a ".doc" extension (Word97)</p> <p>10 - Recognize any image of the bound <b>Page</b> object of the Document Hierarchy in a RTF2000SWord file. Output filename = image filename with an ".rtf" extension (RTF2000SWord)</p> <p>11 - Recognize any image of the bound <b>Page</b> object of the Document Hierarchy in an RTF2000 file. Output filename = image filename with an ".rtf" extension (RTF2000).</p> <p>12 - Recognize the image of the bound <b>Page</b> object of the Document Hierarchy in a Text file with an ".RTF6" extension. Output filename = image filename with an ".rtf" extension (Rich Text).</p> <p>13 - Recognize the image of the <b>Page</b> object of the Document Hierarchy in a Text file with an "RTF6" extension. Output filename = image filename with an ".rtf" extension (Rich Text).</p> <p>14 - Recognize the image of the <b>Page</b> object of the Document Hierarchy in a Text file with an ".Text" extension. Output filename = image filename plus ".txt" extension (Text).</p> <p>15 - Recognize the image of the <b>Page</b> object of the Document Hierarchy in a Text file with an ".Csv" extension. Output filename = image filename with a ".txt" extension (CSV - Comma Separated Variable).</p> <p>16 - Recognize the image of the <b>Page</b> object of the Document Hierarchy in a Text file with a ".FormattedTxt" extension. Output filename = image filename with ".txt" extension (Formatted Text).</p> <p>17 - Recognize the image of the <b>Page</b> object of the Document Hierarchy in a Text file with a ".UText" extension. Output filename = image filename with a ".txt" extension (Text).</p> <p>18 - Recognize the image of the <b>Page</b> object of the Document Hierarchy in a Text file with an ".UCSV" extension. Output filename = image filename with a ".CSV" extension.</p> <p>19 - Recognize the image of the <b>Page</b> object of the Document Hierarchy in a Text file with an ".UFormattedText" extension. Output filename = image filename with ".txt" extension.</p> <p>20 - Recognize the image of the <b>Page</b> object of the Document Hierarchy in a Text file with an ".Audio" extension. Output filename = image filename with ".aud" extension (Text).</p> <p>21 - Recognize the image of the <b>Page</b> object of the Document Hierarchy in a Text file with a ".WordPad" extension. Output filename = image filename with an ".rtf" extension (Rich Text for WordPad).</p> <p>22 - Recognize the image of the <b>Page</b> object of the Document Hierarchy in a Text file with an ".XML" extension. Output filename = image filename with ".xml" extension (XML).</p>
Level	Page, Document or Batch
Returns	False if a ruleset with this action is bound to a <b>Field</b> object of the Document Hierarchy, or if the parameter is not numeric. Otherwise, True.
Example	<b>RecognizePageOCR_S_2TextFile(21)</b>

## RecognizeToPDF

Actions Library	ocr_s.rra/rrx
Description	Converts a scanned Image file (.tif) to an Adobe Portable Document Format (PDF) file.
Parameters	<p><i>Numeric</i> value of 1 – 4 where:</p> <p>1 = Adobe PDF with image on text</p> <p>2 = Adobe PDF: Graphics, KeepBold, KeepItalic and KeepUnderline have an effect</p> <p>3 = Adobe PDF with image substitutes. Graphics,KeepBold, KeepItalic and KeepUnderline have an effect only.</p> <p>4 = Adobe PDF wit image on text. Graphics,KeepBold, KeepItalic and KeepUnderline have an effect only.</p> <p>5. Adobe PDF image only.</p>
Level	Page level only.
Returns	<i>False</i> if the action is not run at the <b>Page</b> level of the Document Hierarchy, or if the parameter is not a value of 1 through 4. Otherwise, <i>True</i> .
Example	<b>RecognizeToPDF(3)</b>
See also...	RecognizeDocToPDF

## RotateImage

Actions Library	ocr_s.rra/rrx
Description	<p>After an action that creates a page's Fingerprint file (.cco), this action checks to see if the scanned Image file (.tio) needs to be rotated by 90, 180, or 270 degrees.</p> <p>If rotation is necessary, the action saves the Image file (.tio) in the new, correct position – and updates the CCO file with the correct position coordinates.</p> <p><b>Alert!</b> Within a rule, a <b>RotateTIO</b> action with a <i>True</i> parameter must precede the <b>RotateImage</b> action (see the example).</p>
Parameters	None
Level	Page level only.
Returns	<i>False</i> if the action is not run at the <b>Page</b> level of the Document Hierarchy. Otherwise, <i>True</i> .
Example	<pre>AnalyzeImage() RotateTIO(True) <b>RotateImage()</b> RecognizePageICR_C()</pre> <p>In this example, a CCO file that corresponds to an existing Image file (.tio) is created by the <b>AnalyzeImage</b> action. The <b>RotateTIO</b> action checks to see if rotation of the image is appropriate, and by how much.</p> <p>If applicable, the subsequent <b>RotateImage</b> action rotates the image in response to the value provided by the RotateTIO action, and provides the new CCO file with correct position coordinates.</p>
See also.....	RotateTIO

## SetFastTradeOffOCR\_S

Actions Library	ocr_s.rra/rrx
Description	Increases the speed of the <b>RecognizePageOCR_S</b> action. <i>Alert!</i> This action's "tradeoff" may be accuracy for speed.
Parameters	None
Level	All
Returns	Always <i>True</i> .
Example	AnalyzeImage() Rotate Image() <b>SetFastTradeOffOCR_S()</b> RecognizePageOCR_S()  This action speeds the <i>word recognition</i> process of the <b>RecognizePageOCR_S</b> action.
See also...	RecognizePageOCR_S

## SetLegacyDecompositionOCR\_S

Actions Library	ocr_s.rra/rrx
Description	"Decomposes" an image to prepare it for <i>field</i> or <i>page</i> recognition.  This action intensifies gradients between and within words on the current page. The action will increase recognition time and should only be used as needed.
Parameters	<i>String</i> value of the Fingerprint ID.
Level	All
Returns	Always <i>True</i> .
Example	SetLegacyDecompositionOCR_S(1010) RecognizePageOCR_S()  This combination creates a CCO file for the current page, intensifies gradients between and within words on the page, then uses settings in the <i>OCR/S</i> tab of the <b>Recognition Options Setup dialog</b> to carry out <i>word</i> recognition of the page.

## PatternMatch Actions

**PatternMatch** actions can be used to:

- Identify pages by matching the current page with a fingerprint in your application's **Fingerprint** library.
  - Adjust the locations (register) of fields on a page by comparing the location of "Anchor-fields" on the current image with the location of those fields in the fingerprint.
- ✓ A "current page" results when an **AnalyzeImage** action converts an **image** generated by a Scan task to a **page** ready for recognition.

### MatchPattern

Actions Library	PatternMatch.rra/.rrx
Description	Seaches on the current image in a zone associated with the current field for a match to the pattern specified for this field in a fingerprint.  The fingerprint will be one with the current image's Fingerprint ID - or the Global Fingerprint ID if the current image is not identified.
Parameters	None
Level	Field level.
Returns	False if:  The anchor position returned is not <i>Numeric</i> ; or  No image is found; or  The accuracy of the match is below the set Confidence Value; or  An anchor match does not occur.  Otherwise, <i>True</i> .
Example	MatchPattern()



## Pat\_RecogMatch\_ID

Actions Library	PatternMatch.rra/.rrx
Description	<p>Establishes a dictionary of unique patterns within the Image file of the current source page.</p> <p>It also links the dictionary to the Image file of the page's fingerprint, and to the fingerprint's Fingerprint file(.cco). As a result, "fingerprint matching" can match a dictionary of patterns in a new source page with pattern dictionaries values in existing fingerprints</p>
Parameters	None
Level	Page only
Returns	False if a ruleset with this action is not bound to a Page object. Otherwise, True.
Example	<b>Pat_RecogMatch_ID( )</b>

## Pat\_RegisterZones

Actions Library	PatternMatch.rra/.rrx
Description	<p>Registers and adjusts the positions of all fields on the image of the current <i>source</i> page, according to the positions of the designated Anchor field(s).  <b>Note:</b> In Datacap Studio, a <b>Field</b> object of the Document Hierarchy is designated as an Anchor by setting its <b>PatternMatch</b> variable to "1" in the object's <i>Properties</i> tab.</p> <p>The expected positions of the Anchor fields on the image (taking into account the Fingerprint classification) are compared with the recognized positions of those Anchor fields - or the Anchor positions set manually by a Fixup task's operator.</p> <p>If one Anchor is found, the field positions are all shifted by the same amount.</p> <p>If two or more Anchors are found, the field positions are shifted by different amounts, depending on their distance from from each Anchor. This process is called <i>Interpolation</i>.</p>
Parameters	None
Level	Page only
Returns	True if the ruleset with this action is bound to a <b>Page</b> object of the Document Hierarchy, and if the action can find all designated Anchor fields. Otherwise, False.
Example	<b>Pat_RegisterZones( )</b>

### **Pat\_ReleasePageAnchors**

Actions Library	PatternMatch.rra/.rrx
Description	
Parameters	
Level	
Returns	
Example	

## PatternMatch\_Fingerprint

Actions Library	PatternMatch.rra/.rrx
Description	<p>Identifies a page from a specified list of fingerprints.</p> <p>The action will load all patterns associated with fields on this page that have been specified as “anchors” from the specified fingerprints, then will search the current image for each of the patterns.</p> <p>When the action finds a match, it will set the matching Fingerprint ID and Page Type.</p> <p>The action will also create page-level fields and update the Anchor fields with anchor-specific pattern offset values in a field level <b>Image_Offset</b> variable.</p>
Parameters	<p>A comma-separated list of one or more Fingerprint ID’s.</p> <p>You use the <i>Fingerprints &amp; Zones</i> panel of the <i>Rulemanager Window</i> to assign Fingerprints ID’s to fingerprints. Chapter 4 of the <i>Guide to Taskmaster Rules</i> explains the process.</p>
Level	Page level only.
Returns	<p><i>False</i> if the rule containing this action was not applied to a Page object of the Document Hierarchy; if a parameter is invalid; if a match does not occur; or if one or more of the specified fingerprints do not exist. Otherwise, <i>True</i>.</p>
Example	<p><b>PatternMatch_Fingerprint(1024,1034,1035,1036)</b></p> <p>This example compares the current page to the four fingerprints specified by their IDs.</p>

## PatternMatch\_Identify

Actions Library	PatternMatch.rra/.rrx
Description	<p>Identifies a page and set its Page Type using pattern matching.</p> <p>The action will load all patterns associated with this field from the fingerprint library, then will search on the current image for each of the patterns. When it finds a match, it sets the matching Fingerprint ID and Page Type. <b>Important!</b> You can use this action together with, or instead of, the <b>FindFingerprint</b> action of the <b>AutoDoc</b> Actions file.</p> <p>When this action finds a match, it sets the matching Fingerprint ID and Page Type. Will also create page-level fields and update the Anchor fields with anchor-specific pattern offset values in a field-level <b>Image_Offset</b> variable.</p>
Parameters	None
Level	Page level only.
Returns	<i>False</i> if the rule containing this action was not applied to a <b>Page</b> object of the Document Hierarchy; if a pattern match is not found; or if fingerprints do not exist. Otherwise, <i>True</i> .
Example	PatternMatch_Identify()

## PatternMatch\_PageType

Actions Library	PatternMatch.rra/.rrx
Description	<p>Identifies a page according to its Page Type.</p> <p>The action loads all patterns associated with fields on the current page that have been specified as “anchors” from the fingerprints associated with the specified Page Type, then searches the current image for each of the patterns.</p> <p>When the action finds a match, it assigns the matching Fingerprint ID and Page Type.</p> <p>The action also creates page-level fields, and update the Anchor fields with anchor-specific pattern offset values in a field-level <b>Image_Offset</b> variable.</p> <p>Finally, the action attempts to match the current page to one more fingerprints, according to the values of their <b>Page Type</b> properties.</p>
Parameters	<p>The value of a <b>Page</b> object’s <b>Type</b> property.</p> <p><b>Important:</b> Usually, this is the <i>name</i> of a <b>Page</b> object, as it appears in the listing of the application’s Document Hierarchy, in the <b>Document Hierarchy Setup</b> window. <i>HCFA 1500</i>, for example, is the name of a <b>Page</b> object in the <i>Medical Claims</i> application – <b>and</b> is the value of its <b>Page Type</b> property.</p>
Level	Page level only.
Returns	<i>False</i> if the rule containing this action was not applied to a <b>Page</b> object of the Document Hierarchy; if the parameter is invalid; if a match does not occur; or if fingerprints do not yet exist. Otherwise, <i>True</i> .
Example	<p>PatternMatch_PageType(HCFA 1500)</p> <p>This action looks for a match among the inventory of fingerprints that have a page type of “HCFA 1500”.</p>

## **SetMatchConfidence**

Actions Library	PatternMatch.rra/.rrx
Description	Sets the confidence threshold for pattern matching.
Parameters	<p>The value of the confidence threshold.</p> <p>The value must be between 0 (lowest confidence) and 9 (highest confidence).</p> <p>Higher values require fewer differences between the compared areas to return a positive match value.</p>
Level	All
Returns	<i>False</i> if the parameter is not a number between 0 and 9. Otherwise, <i>True</i> .
Example	PatternMatch(9)

## Recog\_Shared Actions

### AnalyzeImage

Actions Library	recog_shared.rra/rrx
Description	<p>Converts the Image file (.tif) that represents the current page to a Fingerprint file (.cco) for the page.</p> <p><b>Note:</b> A page's Fingerprint file is also known as its CCO file.</p> <p>A rule using this action should be applied to a <b>Page</b> object (such as the <i>APT</i> application's <i>source APT</i> page.)</p> <p>This action is <i>not</i> required if <i>full-page</i> recognition takes place using actions such as <b>RecognizePageOCR_S</b> or <b>RecognizePageICR_C</b>.</p>
Parameters	None
Level	Page level only.
Returns	<i>False</i> if the action is not run at the <b>Page</b> level of the Document Hierarchy; otherwise, <i>True</i> .
Example	<p><b>AnalyzeImage()</b>          RotateImage()          SetProblemValue(0.7)          SetSearchArea(0.5)          FindFingerprint(True)</p> <p>This sequence generates a CCO file for the current page , then checks to see if rotation of the image is needed.</p> <p>Next, the sequence attempts to match the current page with a fingerprint. (For more about the matching process, see the descriptions of the <b>AutoDoc</b> actions. )</p>

## CCO\_Normalization\_OFF

Actions Library	recog_shared.rra/rrx
Description	<p>Prevents the automatic running of NormalizeCCO procedures after a full-page recognition action has run.</p> <p>A full-page recognition action such as <b>RecognizePageICR_C</b> automatically calls the thorough but time-consuming <b>NormalizeCCO</b> action after recognition is complete (this action is part of the <b>cco2cco.rrx</b> file.)</p> <p>To by-pass this procedure, place <b>CCONormalization_OFF</b> right after the recognition action.</p>
Parameters	None
Level	Page only.
Returns	False if the action does not run at the Page level; otherwise, True
Example	<pre>RecognizePageICR_C ( ) CCONormalization_OFF ( )</pre>



## CreateTextFile

Actions Library	recog_shared.rra/rrx
Description	<p>Creates a Text file for the current page; adds the page's <i>recognized</i> values to the file; and places the file in the current batch, in your application's <b>Batches</b> directory.</p> <p><b>Alert!</b> The Text file generated by this action is handy for debugging purposes – to see what recognition is placing into the page's CCO file.</p> <p>This action should follow a <i>full-page</i> recognition action such as <b>RecognizePageOCR_S</b>, in a rule applied to a <b>Page</b> object of the Document Hierarchy.</p>
Parameters	None
Level	Page level only.
Returns	<p><i>False</i> if the action is not run at the <b>Page</b> level of the Document Hierarchy, or if there is not a CCO file for the processed image.</p> <p>Otherwise, <i>True</i>.</p>
Example	<pre>RecognizePageOCR_S() <b>CreateTextFile()</b> SetProblemValue(0.7) SetSearchArea(0.5) FindFingerprint(True)</pre> <p>After the <i>full-page</i> recognition action (<b>RecognizePageOCR_S</b>), the <b>CreateTextFile()</b> action places the <i>recognized</i> values into a Text file that it sets up for the page, and adds the file to the current batch, in the application's <b>Batches</b> directory. The text file that is created has the same filename as the image, but is assigned a <b>.txt</b> filename extension.</p>
See also...	AnalyzeImage, RecognPageOCR_S, RecogPageICR_C, RecognizePageOCR_S_2TextFile

## IsBlankPage

Actions Library	recog_shared.rra/rrx
Description	<p>Counts the number of words in the CCO file of the current page and returns <i>True</i> if the count is less than or equal to the number you enter as the parameter.</p> <p>A rule containing this action should apply to a <b>Page</b> object; within the rule, this action should come after one of the actions that creates a fingerprint, such as <b>AnalyzeImage</b>, <b>RecognizePageOCR_S</b>, or <b>RecognizePageICR_C</b>.</p>
Parameter	An <i>Integer</i> indicating the maximum words a “blank” page can contain.
Level	Page level only.
Returns	<i>False</i> if the action is not run at the <b>Page</b> level of the Document Hierarchy, or if there is not a CCO file for the processed image. Otherwise, <i>True</i> .
Example	<p>AnalyzeImage()  <b>IsBlankPage(5)</b>  SetPageType(Separator)</p> <p>This sequence uses <b>AnalyzeImage</b> to create a CCO file, then checks to see if the file contains <i>less than</i> six words. If so, the <b>IsBlankPage(5)</b> action returns <i>True</i>.</p> <p>The final action, a <b>DCO</b> action, establishes the page as a <i>Separator</i> page.</p>

## RecognizeOMRThresh

Actions Library	recog_shared.rra/rrx
Description	<p>Performs OMR checkbox recognition by counting black pixels within each OMR box area in a Field with one or more OMR boxes:</p> <p><b>Text Boxes</b> The action sets the text value of the field to a string of 0's and 1's (one digit per OMR box); and assigns a Confidence String to the string of digits: '4' for Low Confidence up to '9' for High Confidence.</p> <p><b>Density String and Confidence Value</b> The action also establishes a "DensityString" variable for the Character String, indicating percentage-filled, from ASCII 48 ('0') through 148.</p> <p>For each possible OMR box, there will be a character. The character's ASCII value minus 48 is the percentage-filled. If the Density String=0X, the first OMR field was blank, and the second was 40% filled. The ASCII value for X is 88. 88 minus 48 = 40.</p> <p><b>MultiPunch and Confidence Values</b></p> <p>If the MultiPunch setting is set to 1 and multiple OMR's were filled beyond the threshold, the one that was filled the most will be marked and set to Low Confidence.</p> <p>If the percentage-filled is below the second parameter, the OMR box will not be selected and the confidence will be high.</p> <p>If the percentage-filled is between the two parameters, the OMR box will not be selected and the confidence will be low.</p> <p>If the percentage-filled is above the first parameter and below double the first parameter minus the second parameter, the OMR box will be selected and the confidence will be low.</p> <p>If the percentage-filled is above double the first parameter minus the second parameter, the OMR box will be selected and the confidence will be high. Note: The <b>RecognizeOMRThresh</b> action works best on dropout boxes, but with an appropriate background value can work effectively with boxes that are visible in the scanned image.</p> <p>This action requires <i>dcimage.ocx.v.6.03.22</i> or above.</p>

RecogOMRThresh (continued)

Parameters	<p>Two comma-separated <i>Floating Point</i> or <i>Integer</i> values that specify the count of black pixels in OMR boxes:</p> <ol style="list-style-type: none"> <li>1. <b>Threshold:</b> the percentage of pixels in the zone - the field zone not the printed box that should be considered "checked", i.e. the lightest box that is not just noise, but should be considered a check mark.</li> <li>2. <b>Background:</b> the percentage of pixels in the zone that might be due to scanner noise and/or the border of the printed box. This value also controls the range on either side of the Threshold value that is "low confidence".</li> </ol> <p>Please note: The action also accepts parameters that are fractional percentages which are needed to detect marks in very large zones. You cannot use commas to separate the decimal and fractional portions of the number. Allowed: <code>RecogOMRThresh(0.5, 0.25)</code>. Not allowed: <code>RecogOMRThresh(0,5,0,25)</code>.</p> <p>The parameters should be experimentally adjusted on real-world scanned forms. First, determine the Threshold value that correctly identifies a light mark as "checked", and correctly identifies noisy zones as "unchecked". Second, adjust the Background parameter's value to achieve an acceptable confidence interval.</p>
Level	Field only.
Returns	Always True.
Examples	<p><b>RecogOMRThresh ( 10 , 5 )</b></p> <p>For a small to medium size zone, "10% filled" may be considered a deliberate mark. Anything below 5% (Background) is definitely <i>not</i> a mark. Anything above 15% (Threshold + (Threshold - Background)) is a high confidence mark. This would work with a non-dropout OMR field where the box's printed outline takes up less than 5% of the zone area. It would also work for dropout forms.</p> <p><b>RecogOMRThresh ( 2 , 0 )</b></p> <p>This is for a signature line, or a large zone where the percentage-filled is usually much lower than for a small zone. This assumes very little background or noise.</p>

## RegisterPageFields

Actions Library	recog_shared.rra/rrx
Description	Returns the field position for all <i>zoned</i> fields of the current page. This action is similar to the <b>ReadZones</b> action. <b>Alert! ReadZones</b> from the <b>zones.rra</b> file is the <i>preferred</i> action.
Parameters	None.
Level	Page level only.
Returns	<i>False</i> if the page's CCO file does not exist, or the action is not run at the Page level. Otherwise, <i>True</i> .
Example	RegisterPageFields()
See also...	ReadZones(Zones.rra)

## ReleaseImage

Actions Library	recog_shared.rra/rrx
Description	Releases an image locked by a preceding action of the recognition process.
Parameters	String value to initiate the action's procedure (True) or prevent it (False).
Level	Page only
Returns	Always True.
Example	<b>ReleaseImage ( True )</b>

## RotateTIO

Actions Library	recog_shared.rra/rrx
Description	<p>Checks to see if an Image file processed by the <b>ImageFix</b> action that assigns a <b>.tio</b> extension to the file needs to be rotated by 90, 180, or 270 degrees.</p> <p>If rotation is necessary, the action saves the Image file with the same <b>.tio</b> extension.</p>
Parameters	A <i>String</i> value that initiates rotation ( <i>True</i> ) or prevents rotation ( <i>False</i> ).
Level	Page level only.
Returns	Always <i>True</i> .
Example	AnalyzeImage() <b>RotateTIO(True)</b> RotateImage() RecognizePageICR_C()
See also...	RotateImage

## SetFingerprintRecogPriority

Actions Library	recog_shared.rra/rrx
Description	<p>Sets the option that controls whether a full-page recognition action is to create a CCO file for the current page.</p> <p>When this option is <i>On</i>, processing is faster because the call to <b>AnalyzeImage</b> is eliminated.</p> <p>This action should be placed before a full-page recognition action.</p>
Parameters	<i>True</i> to turn the action <i>On</i> and create the CCO file, <i>False</i> to turn the action <i>Off</i> .
Level	All
Returns	Always <i>True</i> .
Example	SetFingerprintRecogPriority(True)

## SetFullPageRecogArea

Actions Library	recog_shared.rra/rrx
Description	<p>An <i>optional</i> action that sets the area of the current page that will be the target of recognition procedures when a <b>full-page</b> recognition action is called.</p> <p>For example:</p> <p>“0.1” indicates that the first 10% of the page is to be recognized;</p> <p>“1.00” indicates that the entire page is to be recognized.</p> <p>This action is helpful if you know that a page’s values will always be in a particular location on the page, and recognition of the entire page is not necessary.</p>
Parameters	<p><i>Numeric</i> value of the Recognition area: 0.00 – 1.00.</p> <p>The <i>default</i> value is 1.00.</p>
Level	All
Returns	<i>False</i> if the parameter is not a number from 0.00 through 1.00. Otherwise, <i>True</i> .
Example	SetFullPageRecogArea(0.5)
See also...	RecogPageOCR_S, RecogPageICR_C

## SetRecogFailureRetryDelay

Actions Library	recog_shared.rra/rrx
Description	Sets the amount of seconds to wait before the <b>UseOutOfProcess</b> action intervenes to re-start failed procedures of a full-page recognition action.
Parameters	<i>Numeric</i> value indicating the number of seconds that the <b>UseOutOfProcess</b> is to wait before re-activating the preceding recognition action. <b>Note:</b> When the <b>UseOutOfProcess</b> action starts, it converts seconds into milliseconds.
Level	All
Returns	Always True
Example	<b>SetRecogFailureRetryDelay(10)</b> UseOutOfProcessRecog( True ) RecognizePageOCR_S( )

## SnapCCOtoDCO

Actions Library	recog_shared.rra/rrx
Description	Transfers the recognition results of the current page's CCO file to the appropriate <b>Field</b> objects of the Document Hierarchy (DCO).
Parameters	None.
Level	Page level only.
Returns	Returns <i>False</i> if the action is not run at the Page level, or if a CCO file does not exist for the current page. Otherwise, <i>True</i> .
Example	SnapCCOtoDCO()
See also...	SnapDCOtoCCO



## SnapDCOtoCCO

Actions Library	recog_shared.rra/rrx
Description	<p>Transfers the recognition results assigned to <b>Field</b> objects of the Document Hierarchy (DCO) to the current page's CCO file.</p> <p>If <i>zonal</i> recognition is used instead of <i>full-page</i> recognition, this action will populate the current page's CCO file with the results of <i>zonal</i> recognition. Then, when the Verify task runs, a user can use the <b>ClickNKey</b> option to populate fields.</p>
Parameters	None.
Level	Page level only.
Returns	<i>False</i> if the action is not run at the Page level, or no CCO exists for the current page. Otherwise, <i>True</i> .
Example	SnapDCOtoCCO()
See also...	SnapCCOtoDCO

## UseOutOfProcessingRecog

Actions Library	recog_shared.rra/rrx
Description	<p>This action re-starts recognition launched by a preceding full-page Recognition action that runs into trouble. The action is placed after a full-page action such as <b>RecognizePageOCR_S</b>.</p> <p>It is also directly tied to the <b>SetRecogFailureRetryDelay</b> action, which determines how long (in seconds) the <b>UseOutOfProcessRecog</b> waits before starting up the preceding recognition action.</p>
Parameters	<i>String</i> value indicating whether the action should take effect if recognition stops (True) or should ignore the problem(False).
Level	All
Returns	False if the parameter passed to the function is not True or False. Otherwise, True.
Example	<pre>UseOutOfProcessRecog ( True ) SetRecogFailureRetryDelay ( 5 ) RecognizePageOCR_S ( )</pre>

## RuleRunner Actions

**RuleRunner** actions carry out fundamental procedures that govern the structure and contents of a batch and its documents – and determine how a task will respond if the integrity of a batch is compromised.

As a result, the actions in this category are automatically included in the libraries of *all* RuleSet Types, and the RuleRunner Actions file (**rrrunner.dcs**) does *not* appear as an option to add to a RuleSet Type Actions Library.

### AbortOnError

Actions Library	rrunner.dcs/rrrunner.rrx
Description	Determines whether a task will abort if it encounters scripting errors.
Parameters	“True” if the task is to abort when errors are encountered, or “False” if the task is to continue when it encounters an error.
Level	All
Returns	<i>False</i> if the parameter is invalid; otherwise, <i>True</i> .
Example	AbortOnError(True)

## CheckAllIntegrity

Actions Library	rrunner.dcs/rrunner.rrx
Description	Checks that documents in the batch contain the correct type and number of pages, in line with the Document Integrity requirements of the Document Hierarchy.
Parameters	None
Level	Batch level only.
Returns	<i>True</i> if the Document Integrity of the current batch meets the requirements as defined in setup of the Document Hierarchy. Otherwise, <i>False</i> .
Example	<p>CreateDocuments() CheckAllIntegrity()</p> <p>These actions are part of a rule applied to the <b>Batch</b> object of the Document Hierarchy. The first assembles documents from the pages in the batch; the second ensures that the makeup of each document is valid.</p>

## CheckDocCount

Actions Library	rrunner.dcs/rrunner.rrx
Description	<p>Confirms that the number of documents found in the current task's Page file (.xml) matches the count of <i>expected</i> documents.</p> <p>The expected number of documents is usually collected at scan time and stored in the batch-level variable <b>ED</b>.</p>
Parameters	None
Level	Batch level only.
Returns	<i>True</i> if the two counts are equal; <i>False</i> if they are not.
Example	<p>CheckDocCount()</p> <p>The number of <i>expected</i> documents is usually provided by the operator of a job's Scan task. This very handy action can compare the actual amount to the estimate at any time after a <b>CreateDocuments</b> action has assembled the documents in the batch.</p>
See also...	<b>CheckPageCount</b>

### CheckPageCount

Actions Library	rrunner.dcs/rrunner.rrx
Description	Confirms that the number of individual images (“pages”) listed in the current task’s Page file (.xml) matches the count of <i>expected</i> pages.  The expected number of pages is stored in the batch-level variable, <b>EP</b> .
Parameters	None
Level	Batch level only.
Returns	<i>True</i> if the two counts are equal; <i>False</i> if they are not.
Example	CheckPageCount().
See also...	<b>CheckDocCount</b>

### DebugModeOff

Actions Library	rrunner.dcs/rrunner.rrx
Description	Turns off <i>enhanced</i> logging procedures that were turned on by an earlier <b>DebugMode_On</b> action.  Enhanced logging expands the scope and depth of a processing log’s information, and of the logs that <i>Rulemanager</i> generates when you are testing a rule and its actions.  This feature also increases the size of a Log file significantly, and should only be used when you are testing the impact of an action and rule on the application’s workflow.
Parameters	None
Level	All
Returns	Always <i>True</i> .
Example	DebugMode_OFF()  This action disables the enhanced logging of task activity <i>and</i> of <i>Rulemanager</i> ’s tests of rules and actions.
See also...	<b>DebugMode_ON</b>

## DebugModeOn

Actions Library	rrunner.dcs/rrunner.rrx
Description	<p>Turns on an application's <i>enhanced</i> logging procedures.</p> <p>This is usually the first action in the rule or RuleSet you are testing.</p> <p><b>Alert 1!</b> This action can enable enhanced logging only if you have previously enabled a task's <i>standard</i> logging procedures. For more information on Logging, consult Chapter 3 of the <i>Guide to Batch Pilot</i>.</p> <p><b>Alert 2!</b> Enhanced debugging continues until it is disabled by a <b>DebugMode_OFF</b> action.</p> <p><b>Alert 3!</b> This feature may also increase the size of a Log file significantly, and should only be used when you are testing the impact of an action and rule</p>
Parameters	None
Level	All
Returns	Always <i>True</i> .
Example	<p><b>DebugMode_ON()</b>  OpenConnection(ExportDB)  SetTableName(Invoice)  ExportFieldToColumn(Number,  db_Number)  AddRecord  DebugMode_OFF</p> <p>This rule shows enhanced logging during the course of several actions.</p> <p>If this rule is run by a <i>RuleRunner</i> Export task and you have previously used the <i>Log</i> tab of the RuleRunner's <b>Task Settings</b> dialog to activate <i>standard</i> logging, you will find additional information about the actions in the rule between the <b>DebugMode_ON</b> and <b>DebugMode_OFF</b> actions..</p>
See also...	Status_Preserve_OFF

### GoNextRule

Actions Library	runner.dcs/runner.rrx
Description	Ignores the actions in the current rule and carries forward to the next rule in a ruleset.
Parameters	None
Level	All
Returns	Always False.
Example	<code>GoToNextRule()</code>

### PilotMessage\_Clear

Actions Library	runner.dcs/runner.rrx
Description	Removes the <i>runtime</i> variable "MESSAGE" from the bound object of the Document Hierarchy.
Parameters	None
Level	All
Returns	Always <i>True</i>
Example	<code>PilotMessage_Clear()</code>
See also.....	<code>PilotMessage_Set()</code>

## PilotMessage\_Set

Actions Library	rrunner.dcs/rrunner.rrx
Description	Provides the bound object of the Document Hierarchy with a <i>runtime MESSAGE</i> variable and places the parameter's value in this variable.  A <i>Batch Pilot Data Entry</i> form will display values stored in the <b>MESSAGE</b> variable in the form's status bar when an operator tabs into a field with a <b>MESSAGE</b> variable.
Parameters	String value that you want to place in the <b>MESSAGE</b> variable.
Level	All
Returns	Always <i>True</i>
Example	PilotMessage_Set("Field Value not Valid")
See also.....	PilotMessage_Clear()

## ProcessChildren

Actions Library	rrunner.dcs/rrunner.rrx
Description	A follow-up action that initiates the processing of elements represented by the bound object, and all its children.
Parameters	A two-part, comma-separated specification of a Condition and a Command.  The Condition is any valid VBScript expression. The Command is the VB executable that results from the Condition.
Level	All
Returns	False if the number or sequence of the arguments are invalid. Otherwise, True.
Example	<b>ProcessChildren(1,Exit)</b>

### rr\_AbortBatch

Actions Library	runner.dcs/runner.rrx
Description	Stops processing the current batch and sets the status of the batch to <i>Abort</i>
Parameters	None
Level	All
Returns	Always True.
Example	<code>rr_AbortBatch()</code>

### rr\_Append

Actions Library	runner.dcs/runner.rrx
Description	Uses Smart Parameters to place the first argument's value at the trailing end of the target object's value.
Parameters	Two comma-separated Smart Parameters:  1.) The source value.  2.) A reference to the target object.  The action retrieves the value of the source object, and "appends" it to the target value.
Level	All, but Target must be a <b>Field</b> object
Returns	False if the calling object and target object are the same, or if the action cannot locate the target object's variable. Otherwise, True.
Example	<code>rr_Append(@D.DocID,@F)</code>  This action inserts the current calling object's parent <b>DocID</b> variable value and appends it to the calling field's value.



**rr\_Compare**

Actions Library	runner.dcs/runner.rrx
Description	Uses the Smart Parameters that you enter as the parameter to locate and compare the values of two object's variables. If the values are not the same, the action returns False.
Parameters	<p>A comma-separated String consisting of two Smart Parameters.</p> <p>1.) A reference to the variable of an object that is the first value's source.</p> <p>2.) A reference to the variable of an object that is the second value's source.</p> <p><b>Note:</b> Either reference - but not both - can specify a variable of the calling object (the bound object of the Document Hierarchy.) Alternatively, both references can identify a variable of an object that is a parent or child of the calling object.</p>
Level	All
Returns	False if the compared values do not match. Otherwise, True.
Example	<p><b>rr_Compare(@Tot_Pages,@B.Tot_Pages)</b></p> <p>This example shows how a value is solicited from the <i>Tot_Pages</i> variable of the calling object and the <b>Batch</b> object. The two values are then compared: the action returns False if the values are not the same.</p>

## rr\_Compare\_Not

Actions Library	rrunner.dcs/rrunner.rrx
Description	Uses the Smart Parameters that you enter as the parameter to locate and compare the values of two object's variables. If the values are not the same, the action returns True. This action is the negation of <b>rr_Compare</b> .
Parameters	<p>A comma-separated String consisting of two Smart Parameters.</p> <ol style="list-style-type: none"><li>1.) A reference to the variable of an object that is the first value's source.</li><li>2.) A reference to the variable of an object that is the second value's source.</li></ol> <p><b>Note:</b> Either reference - but not both - can specify a variable of the calling object (the bound object of the Document Hierarchy.) Alternatively, both references can identify a variable of an object that is a parent or child of the calling object.</p>
Level	All
Returns	True if the compared values do not match. Otherwise, False
Example	<p><b>rr_Compare_Not (@Tot_Pages ,@B.Tot_Pages )</b></p> <p>This example shows how a value is solicited from the <i>Tot_Pages</i> variable of the calling object and the <b>Batch</b> object. The two values are then compared: the action returns True if the values are not the same.</p>

**rr\_Copy**

Actions Library	rrunner.dcs/rrunner.rrx
Description	Uses the Smart Parameters you enter as a parameter to copy the value of a source field object to a target field object. This action is unusual in that it is intended to work only on field objects
Parameters	<p>Two comma-separated Smart Parameters:</p> <ol style="list-style-type: none"> <li>1.) A reference to the source field.</li> <li>2.) A reference to the target field.</li> </ol> <p>The action retrieves the value, confidence and image references of the source field object, and copies it to the target field object.</p>
Level	Field level
Returns	False if the action cannot retrieve the target or source object. Otherwise, True.
Example	<p><b>rr_Copy(@B\OPERATOR,@P\OPERATOR)</b></p> <p>This example copies the Batch field <i>OPERATOR</i> value to the <i>OPERATOR</i> field of the bound object of the Document Hierarchy - whatever that happens to be</p>

**rr\_Get**

Actions Library	rrunner.dcs/rrunner.rrx
Description	<p>Uses the Smart Parameter string that you insert as a parameter to locate the value of a target variable, and assign it to the Text property of the calling object.</p> <p>The "rr_" actions use Smart Parameters such as @BatchID, @ID, @VALUE, and @JOBID. A rule containing a Smart Parameter action is bound to an object of the Document Hierarchy: this is the rule's <b>calling object</b>. The rule solicits a value from a variable of a <b>target object</b> or from a processing variable, and assigns the value to the calling object's <b>Text</b> property.</p> <p>You can also use Navigation and Concatenation tools to assemble your own Smart Parameter.</p>
Parameters	<p><i>String</i> Smart Parameter that identifies the target variable that will supply a value to the <b>Text</b> property of the bound object of the Document Hierarchy - the "calling" object. For example:</p> <p>rr_get(@B.Operator) retrieves the value of the <b>Batch</b> object's <b>OperatorID</b> variable and assigns it to the <b>Text</b> property of the calling object.</p> <p>rr_Get(@Operator), on the other hand, assigns the current Operator ID (a processing variable) to the <b>Text</b> property of the calling object.</p>
Level	All
Returns	False if the "calling" object of the Document Hierarchy and a "target" object you specify as a parameter are identical - or if the action cannot locate the target object. Otherwise, True.
Example	<p><b>rr_Get (@B.OPERATOR)</b></p> <p>This example retrieves the value of the <b>Batch</b> object's <b>Operator</b> property and assigns it to the calling object's <b>Text</b> property.</p> <p><b>rr_Get (@DICT_WORD ( . . \MONTH ) )</b>  <b>rr_Set (@DICT_VALUE ( .. \MONTH ) )</b></p> <p>This example shows how Smart Parameters help with the Word and Value settings of an OMR field's Dictionary - in this case, a MONTH dictionary made up of 12 months (January-December) and 12 corresponding values (0-11). The <b>rr_Get</b> action assigns the calling object's <b>Text</b> property to the Dictionary <b>WORD</b> variable of the OMR field 'MONTH', which is a sibling of the calling object.</p> <p>The <b>rr_Set</b> action assigns the calling object's <b>Text</b> property to the corresponding <b>VALUE</b> variable of the OMR field's SEX dictionary.</p> <p>As a result, the OMR field's MONTH dictionary will end up with a new Value and Word pair.</p>

**rr\_Prepend**

Actions Library	runner.dcs/runner.rrx
Description	Uses the Smart Parameters to place the first argument's value at the beginning of the target object's value.
Parameters	<p>Two comma-separated Smart Parameters:</p> <ol style="list-style-type: none"> <li>1.) The source value.</li> <li>2.) A reference to the target object.</li> </ol> <p>The action retrieves the value of the source object, and "prepends" it to the target field value.</p>
Level	All but Target must be a <b>Field</b> object.
Returns	False if the calling object and target object are the same, or if the action cannot locate the target object's variable. Otherwise, True.
Example	<p><b>rr_Prepend(@D.DocID,@F)</b></p> <p>This action inserts the current calling object's parent <b>DocID</b> variable value and prepends it to the calling field's value.</p>

**rr\_Set**

Actions Library	rrunner.dcs/rrunner.rrx
Description	Uses the parameter's elements - two Smart Parameters you insert - to locate the value of a source object's variable, and assign it to a specific variable of a second, receiving object.
Parameters	<p>A comma-separated String consisting of two Smart Parameters:</p> <ol style="list-style-type: none"> <li>1. A reference to the variable of an object that will be the value's source. This object is the parent of the calling object in the Document Hierarchy. (For Smart Parameters, the "calling" object is the bound object of the Document Hierarchy.)</li> <li>2. A reference to the receiving object and its variable.</li> </ol> <p><b>Note:</b> A Smart Parameter can designate a level above or below that of the calling object, and a variable of objects at that level.  <b>@B.Tot_Pages</b> , for example, uses the "@B." syntax to specify the <i>Tot_Pages</i> variable of the <b>Batch</b> object.</p>
Level	All
Returns	False if the calling object and source object are identical - or if the action cannot locate the target object.
Example	<p><b>rr_set (@D.Tot_Pages ,@B.Tot_Pages )</b></p> <p>This example assumes that the calling object is a child of a <b>Document</b> object -- a <b>Page</b> object or <b>Field</b> object. It locates the value in the parent document's <i>Tot_Pages</i> variable and assigns it to the <i>Tot_Pages</i> variable of the parent <b>Batch</b> object.</p> <pre>rr_Get (@DICT_WORD( . . \MONTH ) ) rr_Set (@DICT_VALUE( . . \MONTH ) )</pre> <p>This example shows how Smart Parameters help with the Word and Value settings of an OMR field's Dictionary - in this case, a MONTH dictionary with 12 words (January - December) and twelve corresponding values (0-11)...a value for each word. The <b>rr_Get</b> action assigns the calling object's <b>Text</b> property to the Dictionary <b>WORD</b> variable of the OMR field 'MONTH', which is a sibling of the calling object.</p> <p>The <b>rr_Set</b> action assigns the calling object's <b>Text</b> property to the corresponding <b>VALUE</b> variable of the OMR field's MONTH dictionary.</p> <p>As a result, the OMR field's MONTH dictionary will end up with a new Value and Word pair.</p>

**rr\_WriteNode**

Actions Library	runner.dcs/runner.rrx
Description	Sets up a separate XML data file element for the calling object during rulerunner processing.
Parameters	None.
Level	All.
Returns	Always True.
Example	<b>rr_WriteNode()</b>

**SetExpectedDocs**

Actions Library	runner.dcs/runner.rrx
Description	<p>Assigns a value to the <b>Expected Docs</b> property of the Document Hierarchy's <b>Batch</b> object.</p> <p>This value is stored in the <b>ED</b> variable in the batch's Page file. This value is usually set by the Scan or vScan task.</p>
Parameters	<i>Numeric</i> value indicating the number of <i>expected</i> documents in the batch.
Level	Batch level only.
Returns	<i>False</i> if the parameter is not <i>Numeric</i> ; otherwise, <i>True</i> .
Example	SetExpectedDocs(10)
See also...	SetExpectedPages, CheckDocCount

## SetExpectedPages

Actions Library	rrunner.dcs/rrunner.rrx
Description	<p>Assigns a value to the <b>Expected Pages</b> property of the Document Hierarchy's <b>Batch</b> object.</p> <p>This value is stored in the <b>EP</b> variable of the batch's Page file. This value is usually set by a Scan or vScan task.</p>
Parameters	<i>Numeric</i> value indicating the number of <i>expected</i> pages in the batch.
Level	Batch level only.
Returns	Returns <i>False</i> if the parameter is not <i>Numeric</i> ; otherwise, <i>True</i> .
Example	<b>SetExpectedPages ( 50 )</b>
See also...	SetExpectedDocs, CheckPagecount

## SetTaskStatus

Actions Library	rrunner.dcs/rrunner.rrx
Description	Sets the current task's status value that is to be returned to <i>Taskmaster</i> when the task finishes processing.
Parameters	<p><i>Numeric</i> value representing the status that the task is to return to <i>Taskmaster</i>. The statuses include:</p> <ul style="list-style-type: none"> <li>1 – Cancelled</li> <li>0 – Abort</li> <li>2 – Finished</li> <li>4 – Hold</li> <li>8 – Pending</li> </ul>
Level	All
Returns	<i>False</i> if the parameter is not <i>Numeric</i> ; otherwise, <i>True</i> .
Example	SetTaskStatus(4)



## SkipChildren

Actions Library	rrunner.dcs/rrunner.rxx
Description	Prevents rules applied to <i>child</i> objects of the current <i>parent</i> object from being run.  This action is used to optimize the execution of rules by eliminating the need to visit every field on every page.
Parameters	None
Level	All
Returns	Always <i>True</i> .
Example	SkipChildren()

## Status\_Preserve\_OFF

Actions Library	rrunner.dcs/rrunner.rxx
Description	Changes the <i>Status Preserve</i> condition of a <b>Page</b> object of the Document Hierarchy- and its <b>Field</b> objects - from <i>On</i> to <i>Off</i> .  The <i>Off</i> condition allows the actions of a <b>Validate</b> RuleSet to assign a “problem” status to any <b>Field</b> object with an invalid <i>captured</i> value. The Verification task’s <i>Data Entry</i> panel will then surround the value with a pink background, alerting the operator to the problem
Parameters	None
Level	All
Returns	Always <i>True</i> .
Example	Status_Preserve_OFF()  This action is probably the only action of a single <b>Validate</b> rule that is applied to a <b>Page</b> object of a <i>source</i> page such as the <i>APT</i> application’s <i>APT</i> page.  The application of the rule to the <b>Page</b> object means that subsequent <b>Validate</b> rules and their actions do not have to preserve a <b>Field</b> object’s current status if the <i>captured</i> value for the field is unacceptable.
See also...	Status_Preserve_ON

### Status\_Preserve\_ON

Actions Library	rrunner.dcs/rrunner.rrx
Description	<p>Changes the <i>Status Preserve</i> condition of a <b>Page</b> object - and its <b>Field</b> objects - from <i>Off</i> to <i>On</i>.</p> <p>The <i>On</i> condition prevents a rule and its actions from assigning a “problem” status to a <b>Field</b> object – even if the object’s <i>captured</i> value fails validation.</p> <p><b>Alert!</b> <i>Status_Preserve_ON</i> is the default condition for all RuleSets.</p>
Parameters	None
Level	All
Returns	Always <i>True</i> .
Example	Status_Preserve_ON
<b>See also...</b>	Status_Preserve_OFF

## Task\_NumberOfSplits

Actions Library	rrunner.dcs/rrunner.rrx
Description	<p>Specifies how many times sub-batches have been created from the current batch.</p> <p>This action is used to communicate changes that have been made to the batch to <i>Taskmaster</i>. The action <b>does not</b> actually create sub-batches or split the batch in any way.</p>
Parameters	<i>Integer</i> value of the number of splits. In most cases you will want to use “1” as the parameter
Level	All
Returns	<i>False</i> if the parameter you enter is not <i>Numeric</i> . Otherwise, <i>True</i> .
Examples	<p><b>Task_NumberOfSplits(1)</b> Task_RaiseCondition(0,0)</p> <p>In this example, <i>Taskmaster</i> is alerted to create one sub-batch entry and to raise the first <b>child</b> job condition for this sub-batch entry.</p> <p>or</p> <p><b>Task_NumberOfSplits(1)</b> Task_RaiseCondition(0,1)</p> <p>In this example, <i>Taskmaster</i> is alerted to create one sub-batch <b>entry</b> and to raise the second <b>child</b> job condition for this sub-batch entry.</p> <p>Child Job Conditions are established in the <i>Batch Pilot Task Settings</i> dialog, which is accessible when the task is opened in Setup mode via the <i>Taskmaster Administrator</i>. For example: the <i>1040EZ</i> demo’s PageID task lists one condition: “Requires Fixup”. This condition was setup in the <i>Batch Pilot Task Settings</i> dialog’s <i>General</i> tab.</p>
See also...	Task_RaiseConditions

## Task\_RaiseConditions

Actions Library	rrunner.dcs/rrunner.rrx
Description	<p>Assigns the correct “Child Job Condition” to the correct “sub-batch” <i>entry</i> created by the <b>Task_NumberOfSplits</b> action.</p> <p>In the examples below, one sub-batch <i>entry</i> is created, and a condition is assigned to this entry.</p> <p>For more information on “Child Jobs”, please refer to Chapter 3 of the <i>Guide to Batch Pilot</i> and Chapter 10 of the <i>Guide to Taskmaster Rules</i>.</p>
Parameters	<p><i>Integer</i> value of the sub-batch index; 0 is the first sub-batch, 1 is the second, etc. The <b>Task_NumberOfSplits</b> action determines how many sub-batch entries are created.</p> <p><i>Integer</i> value designating the “Child Job Condition” that should be assigned to the specified sub-batch. 0 is the first Child Job Condition, 1 is the second, etc.</p>
Level	All
Returns	<i>False</i> if either parameter is not <i>Numeric</i> . Otherwise, <i>True</i> .
Example	<p>Task_NumberOfSplits(1) Task_RaiseCondition(0,0)</p> <p>In this example, <i>Taskmaster</i> is alerted to create one sub-batch <i>entry</i> and to raise the first Child Job Condition for this sub-batch entry.</p> <p>or</p> <p>Task_NumberOfSplits(1) Task_RaiseCondition(0,1)</p> <p>In this example, <i>Taskmaster</i> is alerted to create one sub-batch <i>entry</i> and to raise the second Child Job Condition for this sub-batch entry.</p> <p>Child Job Conditions are setup via the <i>Batch Pilot Task Settings</i> dialog, which is accessible when the task is opened in setup mode via the <b>Taskmaster Administrator</b>. For example: the <i>1040EZ</i> demo’s PageID task lists one condition called “Requires Fixup”. This condition was setup via the <i>Batch Pilot Task Settings</i> dialog’s <i>General</i> tab.</p>
See also...	Task_NumberOfSplits

## Split Action

### SplitBatch

Actions Library	Split.rra/.rrx
Description	<p>Splits documents or pages in the batch into sub-batches, according to the value of the smart parameter you enter.</p> <p>If a page is part of a document, it goes along with all other pages in that parent document.</p> <p>Additional considerations:</p> <p>Batches are currently numbered xxx.1, xxx.2...xxx.99 (no leading zero attached). There is only one "job routing condition" raised by this action: it is the first one in the task's list of conditions.</p> <p>The task's <i>Task Setup/Task Settings</i> screen <b>must</b> be configured as <i>Job Router</i> , and a single condition defined (by convention, call it <b>Split</b> .)</p> <ol style="list-style-type: none"> <li>1.) Any and all <b>child batches</b> will be routed via this single condition.</li> <li>2.) If the application wants to treat the individual buckets differently, then the first step in the workflow after splitting can check the same smart parameter value and branch or re-route the child batch using that value.</li> <li>3.) All the structure and variables, etc. that were in the parent batch docs/pages are retained in the child batches.</li> <li>4.) In addition, new variables <b>ParentBatch</b> and <b>ParentBatchDir</b> are added.</li> </ol>

continued on next page →

### SplitBatch (continued)

Parameters	<p>A <b>smart parameter</b> which specifies a variable to be evaluated for every page/document. <b>Important:</b> The resulting values of the variable are grouped into "buckets":</p> <ol style="list-style-type: none"><li>1.) Identical values go into the same bucket.</li><li>2.) If there are multiple buckets, all pages/documents that share the same value will split to the same child batch.</li><li>3.) There can be only <b>one child batch for each unique bucket value</b>.</li></ol> <p>Child batches are named like the parent batch, but with a suffix such as .01, .02, etc.</p> <p>The action's smart parameters were formerly known as Meta Words. For example you might use for splitting:</p> <p><b>@D.Inbox.</b> If there is an Inbox variable in each document, this will split by the value of the Inbox. Important! Any document or loose page (outside of the document) that does not have this value set will remain in the parent batch.</p>
Level	Batch only.
Returns	<p>True if the operation succeeds, False if it does not.</p> <p>Each child batch split off will generate a condition, which should be configured for <b>Split</b> in the workflow. Any page or document with a blank value for the splitting value will remain in the original "parent" batch.</p>
Example	<pre>rr_Set (@BTaskID, @D.TaskID) SplitBatch (@D.TaskID)</pre> <p>This example shows how the Meta Word parameters of the <b>rr_Set</b> action assign values of TaskID to each document's <b>TaskID</b> variable, and how the <b>SplitBatch</b> action splits a batch when it encounters that value.</p>

## TM524 Actions

These actions make it possible to create workflows that include both new-style tasks that use rules and old-style *Taskmaster* tasks, such as *Multitask*, *DocEdit* and *Paper Keyboard* tasks.

### TM524

Actions Library	TM524.rra/.rrx
Description	<p>Makes changes to the Page file for the current batch so that it is backward-compatible with non-rules-based tasks, such as <i>Recog2K</i>, or <i>ExportTX</i>.</p> <p>These tasks expect certain batch-, document-, and page-level variables that are not used in the <b>rules-based</b> environment. After this action is run, old-style tasks will run as if the batch was run entirely with that type of task.</p> <p>Rules that employ this action must be part of a RuleSet that is bound to the Document Hierarchy's <b>Batch</b> object.</p>
Parameters	None
Level	Batch level only.
Returns	Always <i>True</i> .
Example	TM524()

### TM4DocIdFormat

Actions Library	TM524.rra/.rrx
Description	<p>Formats all <b>Document</b> object ID's with a <i>Taskmaster 4</i> layout style.</p> <p>Rules that employ this action must be part of a RuleSet that is bound to the Document Hierarchy's <b>Batch</b> object.</p>
Parameters	None
Level	Batch level only.
Returns	<i>False</i> if the action is not applied at the Batch level; otherwise, <i>True</i> .
Example	TM4DocIdFormat()

## Validate Actions

**Validate** actions can format and validate data before Verification takes place. In addition, they can combine with **Lookup** actions to confirm the accuracy and integrity of a page's *captured* values – values in the page's Data file.

### AddDecimal

Actions Library	validations.rra/rrx
Description	Places a <i>decimal</i> character in the <i>captured</i> value, at the character position you specify as a parameter
Parameter	A number <i>n</i> indicating the character position at which to place the decimal.
Level	Field level only.
Returns	Always <i>True</i> .
Example	AddDecimal(3)  324556 becomes 324.556

### AddToString

Actions Library	validations.rra/rrx
Description	Inserts a character, or string of characters, into the <i>captured</i> value, one or more times.
Parameters	1.) The characters or character string to be inserted; defaults to a space ' ' 2.) A number <i>n</i> indicating the target position within the <i>captured</i> value; defaults to the end of the value. 3.) The number of insertions; defaults to 1.
Level	Field level only.
Returns	Always <i>True</i> .
Example	AddToString(=\$,1,1)  345.67 becomes =\$345.67  AddToString(=\$,1,2)  345.67 becomes =\$=\$345.67



## AllLowerCase

Actions Library	validations.rra/rrx
Description	Converts any Upper Case characters in a <b>Field</b> object's <i>captured</i> value to lower case characters.
Parameters	None
Level	Field level only.
Returns	Always <i>True</i> .
Example	AllLowerCase()  To ensure that the characters in all Product ID's are lower case, a <b>Validate</b> rule that applies to a Document Hierarchy's <i>Item Field</i> object would include this action.
See also...	AllUpperCase

## AllowOnlyChars

Actions Library	validations.rra/rrx
Description	Removes all characters from the current word that are not designated and permitted by the parameter's Regular Expression value.
Parameters	A Regular Expression that specifies permitted characters in the current word.
Level	Field only
Returns	Always <i>True</i> .
Example	AllowOnlyChars(ABCDEFG.)  HELLO DOLLY. becomes ED.

### AllUpperCase

Actions Library	validations.rra/rrx
Description	Converts the lower case characters in a <b>Field</b> object's <i>captured</i> value to Upper Case characters.
Parameters	None
Level	Field level only.
Returns	Always <i>True</i> .
Example	AllUpperCase()  A <b>Validate</b> rule with this action, if applied to a <i>State Field</i> object which accepts only abbreviations, would be sure the <i>captured</i> values contain Upper Case letters (AZ, AL, etc.)
See also...	AllLowerCase

### AppendFromField

Actions Library	validations.rra/rrx
Description	Appends the <i>captured</i> value of the specified <b>Field</b> object to the <i>captured</i> value of the current <b>Field</b> object.  You can also apply this action at the Page level. A <b>Text</b> page-level variable with the appended value will be added to the page's Data file.
Parameter	The name of the <b>Field</b> object whose text value is to be appended to the current field's value.
Level	Page or Field Level.
Returns	<i>False</i> if the parameter is not a <b>Field</b> object's name. Otherwise, <i>True</i> .
Example	AppendFromField(Number)

## AppendToField

Actions Library	validations.rra/rrx
Description	Appends the <i>captured</i> value of the current <b>Field</b> object to the <i>captured</i> value of the <b>Field</b> object specified by the parameter.
Parameter	The name of the <b>Field</b> object to which the value is to be appended
Level	Field level only.
Returns	<i>False</i> if the parameter is not a <b>Field</b> object's name. Otherwise, <i>True</i> .
Example	AppendToField(FirstName)  If the current <b>Field</b> object is <i>MiddleInitial</i> , a rule with this action would append the value of the <i>Middle Initial</i> field to the <i>FirstName</i> <b>Field</b> object.

## Calculate

Actions Library	validations.rra/rrx
Description	<p>Calculates the equation you enter as a parameter and compares the result to the current <b>Field</b> object's <i>captured</i> value.</p> <p><b>Note:</b> If the result does not match the equation's result, all fields involved in the equation will receive a <i>Failed</i> status – and will appear pink in the applicable field of the <i>Data Entry</i> panel.</p>
Parameter	<p>The equation that is the basis for the calculation.</p> <p>You can use a <b>Field</b> object's name or numeric values with any arithmetic operator (+, -, *, /, ^). To use a <b>Field</b> object's name, surround the field name with single quotation marks ( ' ). A null is treated as a "0".</p>
Level	Field level only.
Returns	<i>True</i> if the expression is valid. <i>False</i> if a field's value is not <i>numeric</i> or if the expression is not valid.
Examples	<p>Calculate ('SubTotal' + 'Shipping' + 'Tax' = 'Total')</p> <p style="text-align: center;">or</p> <p>Calculate(('SubTotal' + 'Shipping' + 'Tax') – 0.05 = 'Total')</p>

## CheckSubFields

Actions Library	validations.rra/rrx
Description	<p>Validates an instance of a parent <b>Field</b> object by confirming the presence (or absence) of <i>captured</i> values for one or more of its child fields. (In the <i>APT</i> application, as an example, child fields of the <i>LINEITEM</i> parent include: <i>ItemID</i>, <i>ItemDesc</i>, <i>Quantity</i>, <i>Price</i> and <i>LineTotal</i>.)</p> <p>This action usually runs in its own RuleSet (for example, in a <b>Filter</b> RuleSet) and would be applied to the <i>DETAILS Field</i> object in the <i>APT</i> example.</p>
Parameter	An “expression” that specifies which child fields are to be checked for the presence or absence of <i>captured</i> values. Within the expression, each child <b>Field</b> object’s name needs to be surrounded with single quotation marks ( ‘ ’ ). You can also use parentheses ( ) in your expression.
Level	Field level only.
Returns	Always <i>True</i> .

### CheckSubFields (continued)

Examples	<p><b>Example 1:</b> The <i>captured</i> values for a <i>LINEITEM</i> field are:</p> <p>ItemID = 12345 ItemDesc = Price = 12.00 LineTotal =</p> <p>The action's parameter contains this expression: CheckSubFields(('ItemID' OR 'ItemDesc') AND ('Price' OR 'LineTotal'))</p> <p>In this example the action returns <i>True</i> and the current <i>LINEITEM</i> object is <i>Valid</i>.</p> <p><b>Example 2:</b> The <i>captured</i> values for the <i>LINEITEM</i> field are:</p> <p>ItemID = 12345 ItemDesc = Thank you for your order Price = LineTotal =</p> <p>The action's parameter contains this expression: CheckSubFields(('ItemID' OR 'ItemDesc') AND ('Price' OR 'LineTotal'))</p> <p>In this example the action returns <i>False</i> and the current <i>LINEITEM</i> object is <i>Invalid</i>. The field and the values of its child fields are deleted from the Data file.</p> <p><b>Example 3:</b> The <i>captured</i> values for the <i>LINEITEM</i> field are:</p> <p>ItemID = 12345 LineTotal = gonetolunch</p> <p>The action's parameter contains this expression: CheckSubFields( ('ItemID' ) AND( 'LineTotal' ) )</p> <p>In this case, the action returns <i>True</i> and validates the <i>LINEITEM</i> field – despite the presence of a nonsense entry in the <i>LineTotal</i> child field!</p>
----------	--

## ClearFieldValue

Actions Library	validations.rra/rrx
Description	<p>Deletes the selected <b>Field</b> object's <i>captured</i> value.</p> <p><b>Note:</b> This action does not clear any Alt-Text values associated with the field; instead you use the <b>ClearAltText</b> action of the <b>DCO.rra/rrx</b> Actions Library for this procedure.</p>
Parameters	None
Level	Field level only.
Returns	Always <i>True</i> .
Example	<p>ClearFieldValue()</p> <p>This action typically belongs to a follow-up <b>Validate</b> rule that deals with a <i>False</i> response to an action such as IsDate() in the previous rule.</p> <p>If the field's value is not a date (in this example), the <b>ClearFieldValue</b> action will remove the value that is there.</p>

## CompareFields

Actions Library	validations.rra/rrx
Description	<p>Locates values in two fields specified by the first two parameters.</p> <p>If values are present in both fields, the action compares the values according to the matching criteria of the last three parameters.</p>
Parameters	<p>Five comma-separated values:</p> <p><i>String</i> value of the source field's name. This is the field with a value to be compared.</p> <p><i>String</i> value of the target field's name. This is the field with a value to be compared to.</p> <p><i>String</i> value: Y or Yes; N or No. Alternatively, a <i>Numeric</i> value: 0 = No or 1=Yes. "Yes" (or Y or 1) allows the action to carry out a <b>fuzzy</b> rather precise comparison of the fields' values.</p> <p><i>Numeric</i> value of the percentage of required <b>precision</b>. Numbers less than "100" permit increasing fuzziness.</p> <p><i>String</i> value: Y or Yes; N or No. Alternatively, a <i>Numeric</i> value: 0 = No or 1=Yes. "Yes" (or Y or 1) directs the action to compare values in the fields <b>word-by-word</b>.</p>
Level	All
Returns	<i>False</i> if the designations of <b>Field</b> objects in the first two parameters are not valid, or if the Data Types of the values in the first two fields do not match. Otherwise, <i>True</i> .
Example	CompareFields(Invoice_Date, Due_Date, Yes, 100, Yes)



## CopyToField

Actions Library	validations.rra/rrx
Description	Copies the <i>captured</i> value of the current <b>Field</b> object to the <b>Field</b> object you designate as the action's parameter.
Parameter	The name of the target <b>Field</b> object of the Document Hierarchy.
Level	Field level only.
Returns	<i>False</i> if the parameter does not match a <b>Field</b> object's name. Otherwise, <i>True</i> .
Example	If the current field's value is "1/1/05"  CopyToFIELD(Date)  assigns "1/1/05" (without the quotation marks!) to the Document Hierarchy's <i>Date</i> field.
See also...	WriteField

## DateStamp

Actions Library	validations.rra/rrx
Description	Updates the current <b>Field</b> object with today's date.
Parameter	One of two Date formats: MM/DD/YYYY or DD/MM/YYYY, using two-year or four-year values. <b>Alert!</b> Do not use a <i>Numeric</i> string.  The parameter defaults to MM/DD/YYYY.
Level	Field level only.
Returns	Always <i>True</i> .
Example	<b>DateStamp(*)</b> produces 01/20/2007  <b>DateStamp(dd/mm/yy)</b> produces 20/01/05

### DefaultValue

Actions Library	validations.rra/rrx
Description	Sets the <i>captured</i> value of the current <b>Field</b> object to the <i>String</i> value you enter as a parameter.
Parameter	The <i>String</i> value you want to use.
Level	Field level only.
Returns	Always <i>True</i> .
Examples	DefaultValue(Bill Paid) DefaultValue(Past Due)

### DeleteALPHA

Actions Library	validations.rra/rrx
Description	Deletes all <i>alphabetic</i> characters from the current <b>Field</b> object's <i>captured</i> value.
Parameters	None
Level	Field level only.
Returns	Always <i>True</i> .
Example	DeleteALPHA()  JAN2003 becomes 2003
See also...	DeleteChars, DeleteMISC, DeleteNUMERIC, DeletePUNCT

## DeleteChars

Actions Library	validations.rra/rrx
Description	Deletes specific characters from the current <b>Field</b> object's <i>captured</i> value
Parameter	<p>1.) The character or string of characters to be deleted; defaults to the character at the <i>target position</i> parameter (#2 below).</p> <p>2.) A number <i>n</i> indicating the <b>target position</b> within the value; defaults to first location of the <i>target String</i>.</p> <p>3.) The number of times the character or character string is to be deleted from the value. The default is "1" and "*" deletes <i>all</i> instances.</p>
Level	Field level only.
Returns	Always <i>True</i> .
Example	<p>DeleteChars(-, *,*)</p> <p>223-56-7669 becomes 223567669</p>
See also...	DeleteALPHA, DeleteMISC, DeleteNUMERIC, DeletePUNCT

## DeleteChildType

Actions Library	validations.rra/rrx
Description	Deletes child objects of the type you designate as a parameter from the Document Hierarchy.
Parameter	<p><i>String</i> value of the child object Type.</p> <p>"Field", for example, will eliminate all <b>Field</b> objects. In the <i>APT</i> application, however, "LineItem" will remove child fields of the parent <i>Details</i> parent <b>Field</b> object.</p>
Level	All except Batch.
Returns	False if the child objects do not exist. Otherwise, True.
Example	DeleteChildType(Field)

### DeleteMISC

Actions Library	validations.rra/rrx
Description	Removes all characters with ASCII values 123 through 191 from the current <b>Field</b> object's <i>captured</i> value.
Parameter	None
Level	Field level only.
Returns	Always <i>True</i> .
Example	DeleteMISC()  {Hello} becomes Hello
See also...	DeleteALPHA, DeleteChars, DeleteNUMERIC, DeletePUNCT

### DeleteNUMERIC

Actions Library	validations.rra/rrx
Description	Removes all <i>Numeric</i> characters from the current <b>Field</b> object's <i>captured</i> value
Parameters	None
Level	Field level only.
Returns	Always <i>True</i> .
Example	DeleteNUMERIC()  JAN2003 becomes JAN
See also...	DeleteALPHA, DeleteChars, DeleteMISC, DeletePUNCT

### DeleteParent

Actions Library	validations.rra/rrx
Description	Deletes the parent of the Document Hierarchy object to which a rule with this action is bound.
Parameters	None
Level	Document, Page, and Field
Returns	<i>False</i> if a parent or grandparent object cannot be found, or if the deletion of the parent object fails. Otherwise, <i>True</i> .
Example	DeleteParent()

**DeletePUNCT**

Actions Library	validations.rra/rrx
Description	Removes all characters with ASCII values 33-47,58-64,91-96,and 123-191 from the current <b>Field</b> object's <i>captured</i> value
Parameters	None
Level	Field level only.
Returns	Always <i>True</i> .
Example	DatePUNCT()
See also...	DeleteALPHA, DeleteChars, DeleteMISC, DeleteNUMERIC

**DeleteSYSTEM**

Actions Library	validations.rra/rrx
Description	Removes all characters with ASCII values 0-31 from the current <b>Field</b> object's <i>captured</i> value
Parameters	None
Level	Field level only.
Returns	Always <i>True</i> .
Example	DeleteSystem()  Hello Dolly becomes HelloDolly.

### EmptyFIELD

Actions Library	validations.rra/rrx
Description	Clears the text value in the field represented by the <b>Field</b> object of the Document Hierarchy specified by the parameter.
Parameters	The name of the <b>Field</b> object that is to be emptied.
Level	Page or Field level.
Returns	<i>False</i> if the field specified by the parameter does not exist. Otherwise, <i>True</i> .
Example	EmptyFIELD(Date)  A rule with this action can be bound to a <b>Page</b> object as long as the page has a <i>Date</i> field.

### FailRuleSet

Actions Library	validations.rra/rrx
Description	Causes the entire RuleSet to fail.
Parameters	None
Level	All.
Returns	Always <i>False</i> .
Example	Is_Value(False) <b>FailRuleSet()</b>

### FieldContainsValue

Actions Library	validations.rra/rrx
Description	Checks if a field represented by the bound object of the Document Hierarchy contains the parameter's text value.
Parameters	The text that the action is looking for.
Level	All, but generally at the Field level.
Returns	<i>False</i> if the value is not in the object's <b>CurrentObj.Text</b> variable. Otherwise, <i>True</i> .
Example	FieldContainsValue(Invoice)

## FilterValue

Actions Library	validations.rra/rrx
Description	Removes <i>all</i> instances of the character(s) you enter as the parameter, from the current <b>Field</b> object's <i>captured</i> value
Parameter	A <i>String</i> containing the character(s) to be removed.  Every instance of the character(s) will be removed from the <i>captured</i> value.
Level	Field level only.
Returns	Always <i>True</i> .
Example	FilterValue(0)  11002900 becomes 1129
See also...	DeleteChars

## GetJobID

Actions Library	validations.rra/rrx
Description	Assigns the Job ID of the current <i>Taskmaster</i> job – the job's <b>Pilot.JobID</b> property – to the <b>CurrentObj.Text</b> variable of the bound object of the Document Hierarchy.
Parameters	None
Level	All
Returns	<i>False</i> if the action cannot find a JobID value. Otherwise, <i>True</i> .
Example	GetJobID()

### GotoNextRule

Actions Library	validations.rra/rrx
Description	Returns a <i>False</i> condition so that the next rule in the RuleSet can run
Parameters	None
Level	All
Returns	Always <i>False</i> .
Example	<p>Is_Value(Skip) <b>GotoNextRule()</b></p> <p>If the condition in the first action is met, the sequence assigns a <i>False</i> status to the second action and to the rule of which it is a part. As a result, the RuleSet automatically summons the next <i>rule</i>.</p>

### HasChildOfType

Actions Library	validations.rra/rrx
Description	<p>Determines if the bound object of the Document Hierarchy has a child or children of the type specified by the parameter.</p> <p>The action can also determine if a <i>runtime</i> variable you specify as a parameter has been assigned to the bound object.</p>
Parameters	The name of a level of the Document Hierarchy (Batch, Document, Page, Field) or of a <i>runtime</i> variable.
Level	All.
Returns	<i>False</i> if the bound object does not include a child or children specified by the parameter, or a variable identified by the parameter. Otherwise, <i>True</i> .
Examples	<p>HasChildOfType(Page)</p> <p>This example determines if the bound object is the parent of one or more pages.</p> <p>HasChildOfType(IGNORE)</p> <p>In this example, the action determines if <b>IGNORE</b> is a <i>runtime</i> variable of the bound object.</p>



## Is\_Alpha

Actions Library	validations.rra/rrx
Description	Determines if the characters in the current <b>Field</b> object's <i>captured</i> value are <i>n%</i> alphabetic.
Parameter	A number (0-100) indicating the percentage necessary to return a <i>True</i> condition.
Level	Field level only.
Returns	<i>True</i> if the parameter's requirement is met. Otherwise, <i>False</i> .
Example	Is_Alpha(50)  #RPR-1421 returns <i>False</i>  Is_Alpha(30)  #RPR1421 returns <i>True</i>
See also...	Is_Numeric, Is_Value

## Is\_Currency

Actions Library	validations.rra/rrx
Description	Determines if the current <b>Field</b> object's <i>captured</i> value is 100% <i>Numeric</i> <b>and</b> includes a two-digit decimal amount.  When determining the <i>Numeric</i> %, the action ignores spaces and punctuation such as commas and decimal points.
Parameters	None
Level	Field level only.
Returns	<i>True</i> if the action's specifications are met. Otherwise, <i>False</i> .
Example	Is_Currency()  \$1,200 returns <i>False</i>  \$1,200.00 returns <i>True</i>

### Is\_Date

Actions Library	validations.rra/rrx
Description	Ensures that the captured value of the field represented by the bound <b>Field</b> object has an acceptable Date format.  This action will accept any valid date from 01/01/0001 through 12/31/9999.
Parameters	None
Level	Field level only.
Returns	<i>True</i> if the action's specifications are met. Otherwise, <i>False</i> .
Examples	Is_Date ()  April 6, 1944 returns <i>True</i> 04/06/44 returns <i>True</i> 30.6.44 returns <i>True</i> Feb 31,2003 returns <i>False</i>

### Is\_DateSetFormat

Actions Library	validations.rra/rrx
Description	Confirms that a field's data is a valid date and then formats (or re-formats) its Date value according to the format you enter as the parameter.
Parameter	<b>Parameter:</b> The <i>Date</i> format you want to use.  mm/dd/yyyy ( <i>Default</i> ) mm/dd/yy dd/mm/yy mm.dd.yy, etc.
Level	Field level only.
Returns	False if the parameter is invalid, or the current field value is not a valid date. Otherwise returns True.
Example	<b>Is_DateSetFormat ( * )</b> June 3, 2002 becomes 06/03/2002  <b>Is_DateSetFormat (mm.dd.yy)</b> June 3, 2002 becomes 06.03.02

## Is\_DateUpToToday

Actions Library	validations.rra/rrx
Description	Checks that the current <b>Field</b> object's <i>Date</i> value is today's date or earlier.
Parameters	None
Level	Field level only.
Returns	<i>False</i> if the field's value is not a valid date or if the date is after "Today"; otherwise, <i>True</i> .
Example	<p>Is_Date() Is_DateUpToToday()</p> <p>This sequence confirms that a value is a date and that it is the same as, or earlier than, today's date.</p>
See also...	Is_Date, DateStamp

## Is\_DateWithinXDays

Actions Library	validations.rra/rrx
Description	Checks that current <b>Field</b> object's <i>captured</i> Date value is within <i>n</i> days of the number you enter as a parameter.
Parameter	A number <i>n</i> that specifies how many days make up the review period.
Level	Field level only.
Returns	<i>False</i> if the field's value is not a valid date or if the date is older than the number of days in the parameter; otherwise, <i>True</i> .
Example	<p>Is_Date() Is_DateWithinXDays(30)</p> <p>This sequence checks that a value is a date within 30 days of today's date.</p>
See also...	Is_Date, Is_DateUpToToday

## Is\_DateWithinRange

Actions Library	validations.rra/rrx
Description	<p>Checks that the value assigned to the <b>Text</b> property of the bound object is a valid Date.</p> <p>If so, the action confirms that the Date is within the range specified by the parameters.</p>
Parameters	<p>Comma-separated Dates that define the range:</p> <p>1) Start Date</p> <p>2) End Date.</p> <p><b>TODAY</b> can represent the current Date.</p>
Level	All, but generally at the Field Level.
Returns	<i>False</i> if the value assigned to the current object's <b>Text</b> property is not a valid date; if either parameters is invalid; or the <b>Text</b> property's Date value is not within the range specified by the parameters.
Example	<p>Is_DateWithinRange(1/1/2006, 1/31/2006)</p> <p>Is_DateWithinRange(1/1/2006,TODAY)</p>

## Is\_GreaterThan

Actions Library	validations.rra/rrx
Description	Determines if the <i>captured</i> value of the current <b>Field</b> object is greater than (or equal to) the value you enter as a parameter. If the field's value is not <i>Numeric</i> or <i>currency</i> , the action will return a <i>False</i> condition.
Parameter	The <i>Numeric</i> or <i>Currency</i> value which is the basis for comparison.
Level	Field level only.
Returns	<i>False</i> if the parameter or <b>Field</b> object's <i>captured</i> value is not <i>Numeric</i> - or if the result does not meet the action's requirements. Otherwise, <i>True</i> .
Example	<p>Is_GreaterThan(624)</p> <p>Returns <i>True</i> if the <b>Field</b> object's value is 625.00</p> <p>Returns <i>False</i> if the value is 623.99.</p> <p>Returns <i>True</i> if the value is 624.00.</p>
See also...	Is_LessThan

## Is\_LessThan

Actions Library	validations.rra/rrx
Description	Determines if the <i>captured</i> value of the current <b>Field</b> object is less than (or equal to) the value you enter as a parameter. If the field's value is not <i>Numeric</i> or <i>Currency</i> , the action will return a <i>False</i> condition.
Parameter	The <i>Numeric</i> or <i>Currency</i> value you want to compare against.
Level	Field level only.
Returns	<i>False</i> if the parameter or <b>Field</b> object's <i>captured</i> value is not <i>Numeric</i> - or if the result does not meet the action's requirements. Otherwise, <i>True</i> .
Example	<p>Is_LessThan(625)</p> <p>Returns <i>True</i> if the <b>Field</b> object's value is 624.99.</p> <p>Returns <i>False</i> if the value is 625.01</p> <p>Returns <i>True</i> if the value is 625.00</p>
See also...	Is_GreaterThan

### Is\_NonNumeric

Actions Library	validations.rra/rrx
Description	Determines if the characters in the <i>captured</i> value of the current <b>Field</b> object are <i>n%</i> non-numeric.
Parameter	A number (0-100) indicating the percentage that results in a <i>True</i> condition. The default percentage is 100.
Level	Field level only.
Returns	<i>False</i> if the parameter is non-numeric, or if the field's value exceeds the parameter's percentage of <i>numeric</i> characters. Otherwise, <i>True</i> .
Example	If the current value is A33c...  <code>Is_NonNumeric(50)</code> returns <i>True</i>  <code>Is_NonNumeric(60)</code> returns <i>False</i>
See also...	Is_Alpha, Is_Value

### Is\_Numeric

Actions Library	validations.rra/rrx
Description	Confirms that the field contains no alphabetic or special characters
Parameter	A number (0-100) indicating the percentage that results in a <i>True</i> condition. The default percentage is 100.
Level	Field level only.
Returns	<i>True</i> if the parameter's condition is met. Otherwise <i>False</i> , including if the field is empty.
Example	If the current value is -3.7  <code>Is_Numeric(50)</code> returns <i>True</i>  <code>Is_Numeric(60)</code> returns <i>False</i>
See also...	Is_Alpha, Is_Value

## Is\_Value

Actions Library	validations.rra/rrx
Description	Determines if the value you've entered as the parameter is identical to the <i>captured</i> value of the current <b>Field</b> object.
Parameter	The value to be checked against the <b>Field</b> object's value.
Level	Field level only.
Returns	<i>True</i> if the action's requirement is met. Otherwise, <i>False</i> .
Example	If the <b>Field</b> object's value is 525.00  Is_Value(525.00) returns <i>True</i>  <b>Is_Value(525)</b> returns <i>False</i>
See also...	Is_Alpha, Is_Numeric

## IsDateGE

Actions Library	Validation.rra
Description	Checks that the Date value in the current field represented by the bound <b>Field</b> object of the Document Hierarchy is greater than or equal to the Date value in the field you specify as the parameter.
Parameters	The name of the <b>Field</b> object of the Document Hierarchy to be compared with the current field's Date value.
Level	Field level.
Returns	<i>False</i> if the action is not applied at the Field level, or either field does not contain a valid Date. Otherwise, <i>True</i> .
Example	IsDateGE(24aDtFr1)
See also...	IsDateLE

## IsDateLE

Actions Library	Validation.rra
Description	Checks that the date in the current field represented by the bound <b>Field</b> object of the Document Hierarchy is less than or equal to the Date value in the field you specify as the parameter.
Parameters	The name of the <b>Field</b> object of the Document Hierarchy to be compared with the current field's Date value.
Level	Field Level.
Returns	<i>False</i> if the action is not applied at the Field level, or if either field does not contain a valid Date value. Otherwise, <i>True</i> .
Example	IsDateLE (36aospth)
See also...	IsDateGE

## IsFIELDEmpty

Actions Library	validations.rra/rrx
Description	Checks to see if the <b>Field</b> object you designate as a parameter does <i>not</i> currently have a <i>captured</i> value.
Parameter	The <b>Field</b> object's name.
Level	All.
Returns	<i>False</i> if the <b>Field</b> object's name does not exist - or if the field contains a <i>captured</i> value. Otherwise, <i>True</i> .
Example	<p><b>IsFIELDEmpty(Shipping)</b>                      DefaultValue(NoShipping)</p> <p>In this example, if the <i>Shipping</i> <b>Field</b> object's <i>captured</i> value is \$921.11, this action will return a <i>False</i> condition – and therefore terminate the rule.</p> <p>If the <i>Shipping</i> <b>Field</b> object does not have a value (<i>True</i>), the <b>DefaultValue</b> action enters its parameter (“NoShipping”) as the <b>Field</b> object's <i>captured</i> value.</p>
See also...	DefaultValue, IsFIELDFilled



**IsFIELDFilled**

Actions Library	validations.rra/rrx
Description	Determines if a <b>Field</b> object contains a <i>captured</i> value, or is empty.
Parameter	The name of the <b>Field</b> object.
Level	All
Returns	<i>False</i> if the <b>Field</b> object's name does not exist - or if the field does not contain a <i>captured</i> value. Otherwise, <i>True</i> .
Example	IsFIELDFilled(PaymentDue)  If the action returns <i>True</i> – if the field does contain a value – the rule invokes its next <i>action</i> .  If the action returns <i>False</i> , the rule closes and the task applies the next <i>rule</i> , which might include a <b>CopyToFIELD</b> action.
See also...	DefaultValue, IsFIELDEmpty

**IsHiddenField**

Actions Library	validations.rra/rrx
Description	Determines if the STATUS value of the <b>Field</b> object representing the current field is -1 (Hidden).  Typically, a field with a -1 STATUS is not part of a Verify panel.
Parameter	None
Level	Field only.
Returns	True if the <b>Field</b> object's STATUS = -1; Otherwise False .
Example	<b>IsHiddenField()</b>

### IsJobID

Actions Library	validations.rra/rxx
Description	Checks that the Job ID of the current <i>Taskmaster</i> job matches the parameter's Job ID value.
Parameters	<i>String</i> value of the Job ID to be compared to the current Job ID.
Level	All
Returns	<i>True</i> if the current Job ID matches the parameter's value. Otherwise, <i>False</i> .
Example	IsJobID(Main)

### IsThisFieldEmpty

Actions Library	validations.rra/rxx
Description	Checks to see if the current field has no captured value.
Parameter	None
Level	Field level only.
Returns	<i>False</i> if not applied to the Field level, or if the current field has a text value. Otherwise, <i>True</i> .
Example	IsThisFieldEmpty()
See also...	IsThisFieldFilled, IsFIELDEmpty

**IsThisFieldFilled**

Actions Library	validations.rra/rrx
Description	Checks to see if the current field has a <i>captured</i> value.
Parameter	None
Level	Field level only.
Returns	<i>False</i> if not applied to the Field level, or if the current field has no text value. Otherwise, <i>True</i> .
Example	IsThisFieldFilled()
See also...	IsThisFieldEmpty, Is FIELDFilled

**IsVariableEmpty**

Actions Library	validations.rra/rrx
Description	Checks to see if the variable specified by the parameter does not contain a value.  This action only checks variables of the current object.
Parameter	Name of the current object's variable to be checked.
Level	All
Returns	<i>False</i> if the parameter is invalid, or if the variable contains a value. Otherwise, <i>True</i> .
Example	IsVariableEmpty(TemplateID)
See also...	IsVariableFilled

### IsVariableFilled

Actions Library	validations.rra/rrx
Description	Checks to see if the variable specified by the parameter contains a value.  This action only checks variables of the current object.
Parameter	The name of the current object's variable to be checked.
Level	All
Returns	False if the parameter is invalid, or if the variable does not contain a value. Otherwise, <i>True</i> .
Example	IsVariableFilled(TemplateID)
See also...	IsVariableEmpty

### LeadingZeros

Actions Library	validations.rra/rrx
Description	Adds zeros ("0") to the beginning of the current <b>Field</b> object's <i>captured</i> value until the value's total length reaches the maximum <i>n</i> you specify as the parameter.
Parameter	A number <i>n</i> which is the value's maximum length.
Level	Field level only.
Returns	<i>False</i> if the parameter you enter is not <i>numeric</i> ; otherwise, <i>True</i> .
Example	LeadingZeros(10)  2240.00 becomes 0002240.00
See also...	TrailingZeros, MaximumLength, PadToLeft

## LeftTruncate

Actions Library	validations.rra/rrx
Description	Deletes characters from the start of the current <b>Field</b> object's <i>captured</i> value until the value's length equals the length indicated by the parameter.
Parameter	A number <i>n</i> that is the value's maximum length.
Level	Field level only.
Returns	<i>False</i> if the parameter is not <i>Numeric</i> . Otherwise, <i>True</i> .
Example	LeftTruncate(6) EU0002240 becomes 002240
See also...	RightTruncate, MaximumLength

## MakeCurrency

Actions Library	validations.rra/rrx
Description	Formats a field's text value as a currency value.
Parameters	None
Level	All, but generally at the Field level.
Returns	<i>True</i> if the text value is <i>numeric</i> and greater than one character. Otherwise, <i>False</i> .
Example	MakeCurrency()

## MaximumLength

Actions Library	validations.rra/rrx
Description	Checks the character length of the current <b>Field</b> object's <i>captured</i> value to see if its length is equal to <i>or</i> shorter than the parameter you set.
Parameter	A number <i>n</i> designating the value's maximum length.
Level	Field level only.
Returns	<i>False</i> if the parameter you enter is not <i>numeric</i> , or if the number of characters exceeds the parameter's value. Otherwise, <i>True</i> .
Example	MaximumLength(6) EU2240 returns <i>True</i> EU002240 returns <i>False</i>
See also...	MinimumLength

## MessageBox

Actions Library	validations.rra/rrx
Description	<p>Causes a <b>Message Box</b> with two buttons, labeled “Yes” and “No,” to appear on the operator’s screen during the Verification task, and displays a message you enter as the parameter.</p> <p><b>Alert!</b> This action should only be placed in a rule that is run interactively. If a background station runs this rule, it will stop processing until a user responds to the message box.</p> <p>If the Data Entry operator clicks on the Yes button in the <b>Message Box</b>, the action returns <i>True</i> and the rule continues. If the operator chooses the No button, the action returns <i>False</i> and the rule fails.</p>
Parameter	The message you want to display to the Data Entry operator.
Level	All
Returns	<i>False</i> if the current task is not interactive – or if the operator clicks on the <b>Message Box’s</b> No button. Otherwise, <i>True</i> .
Example	<pre>IsFIELDEmpty(Date) MessageBox(Date missing!Continue?)</pre> <p>After checking to be sure the <i>Date</i> field does not have a value, the <b>MessageBox</b> action issues a warning, and asks the operator to respond by clicking on the Yes or No button.</p>

## MinimumLength

Actions Library	validations.rra/rrx
Description	Checks the character length of the current <b>Field</b> object’s <i>captured</i> value to see if its length is equal to <i>or</i> longer than a number <i>n</i> .
Parameter	A number <i>n</i> designating the value’s minimum length.
Level	Field level only.
Returns	<i>False</i> if the parameter you enter is not <i>numeric</i> , or if the number of characters is less than the parameter’s value. Otherwise, <i>True</i> .
Example	<pre>MinimumLength(6) EU2240 returns True EU22 returns False</pre>
See also...	MaximumLength

### Non\_OVERRIDEABLE

Actions Library	validations.rra/rrx
Description	<p>Specifies that if the validation fails for the current object, that it is <i>non-overrideable</i> by the user.</p> <p><b>Important!</b> This status prevents an operator from overriding a field's validations and then continuing on to subsequent pages.</p>
Parameters	None
Level	All
Returns	Always <i>True</i> .
Example	<p>Non_OVERRIDEABLE() Is_Numeric(100)</p> <p>In this sequence, if the field's <i>captured</i> value is not 100% <i>Numeric</i>, an operator cannot override a Validation rule's subsequent rejection of the value.</p>
See also...	OVERRIDEABLE

### OMR\_MaxChecked

Actions Library	validations.rra/rrx
Description	Indicates the maximum number of checkboxes that can contain a value (a check, for example).
Parameters	An <i>Integer</i> value specifying this maximum.
Level	Field level only.
Returns	<p><i>False</i> if the parameter you enter is not <i>numeric</i>, or the field is not an OMR field.</p> <p><i>True</i> if the number of OMR boxes checked is <i>less than or equal</i> to the parameter you entered.</p>
Example	OMR_MaxChecked(1)
See also...	OMR_MinChecked



## OMR\_MinChecked

Actions Library	validations.rra/rrx
Description	Indicates the minimum number of checkboxes that can contain a value (a check, for example).
Parameters	An <i>Integer</i> value specifying this minimum.
Level	Field level only.
Returns	<i>False</i> if the parameter you enter is not <i>numeric</i> , or the field is not an OMR field.  <i>True</i> if the number of OMR boxes checked is <b><i>greater than or equal</i></b> to the parameter you entered.
Example	OMR_MinChecked(1)
See also...	OMR_MaxChecked

## OVERRIDEABLE

Actions Library	validations.rra/rrx
Description	Specifies that if the validation fails for the current object, that it is <b><i>overrideable</i></b> by the user, allowing them to continue on to subsequent pages.  <b><i>Important!</i></b> This status allows an operator to override the field's validations.  <i>Overrideable</i> is the default for all fields.
Parameter	None
Level	All
Returns	Always <i>True</i> .
Example	<b>OVERRIDEABLE ( )</b> Is_Numeric(100)  In this sequence, if the field's <b><i>captured</i></b> value is not 100% <i>Numeric</i> , an operator can override this Validation rule.
See also...	Non_OVERRIDEABLE

## PadToLeft

Actions Library	validations.rra/rrx
Description	Pads the current <b>Field</b> object's <i>captured</i> value with spaces to the left of the first character.
Parameter	A number <i>n</i> indicating the maximum permissible length of the value. If the action finds that a value's length is less than this number, it will insert spaces until the maximum length is reached.
Level	Field level only.
Returns	Always <i>True</i> .
Example	<b>PadToLeft (12)</b> uses spaces to expand a value with less than 12 characters. For example:  RSJ-112 becomes _ _ _ _ RSJ-112
See also...	PadToRight

## PadToRight

Actions Library	validations.rra/rrx
Description	Pads the current <b>Field</b> object's <i>captured</i> value with spaces to the right.
Parameter	A number <i>n</i> indicating the maximum permissible length of the value. If the action finds that a value's length is less than this number, it will insert spaces until the maximum length is reached.
Level	Field level only.
Returns	Always <i>True</i> .
Example	<b>PadToRight (10)</b> uses spaces to expand a value with less than 10 characters. For example:  456.11 becomes 456.11_ _ _ _
See also...	PadToLeft

## ParseMultilineAddress

Actions Library	validations.rra/rrx
Description	<p>Parses the Multiline Address value in the current field represented by the bound <b>Field</b> object of the Document Hierarchy.</p> <p>The action will place the appropriate parsed value into the fields specified by the parameters.</p>
Parameters	<p>Comma-separated <i>String</i> values of the names of fields which will hold the parsed data, in the following order:</p> <p>Name, AddressLine1, AddressLine2, City ,State ,Zip , Phone</p>
Level	Field level only.
Returns	<i>False</i> if the parameters are invalid, or parsing cannot take place. Otherwise, <i>True</i> .
Example	ParseMultilineAddress(VendorName,VenAddress1,VenAddress2,VenCity,VenState,VenZip,VenPhone)

## ParseName

Actions Library	validations.rra/rrx
Description	<p>Parses a <i>name</i> <b>Field</b> object's <i>captured</i> value.</p> <p>Applied to a "Name" field, the action will parse the Last, First, and Middle names into the fields specified by the parameter.</p>
Parameter	<p>Three comma separated parameters:</p> <p>The name of the <i>Last Name</i> <b>Field</b> object.</p> <p>The name of the <i>First Name</i> <b>Field</b> object.</p> <p>The name of the <i>Middle Name</i> or <i>Middle Initial</i> <b>Field</b> object.</p>
Level	Field level only.
Returns	<i>False</i> if not placed at the Field level; if the current field contains no data; or if the parameters are invalid. Otherwise, <i>True</i> .
Example	<p>ParseName(LastName,FirstName,MidName)</p> <p>Bound to a <i>Name</i> <b>Field</b> object which includes values for all three names, the action will place the Last name into the "LastName" field, the First name into the "FirstName" field, and the Middle name (or middle initial) into the "MidName" field.</p>
See also...	ParseMultilineAddress

### PatternInField

Actions Library	validations.rra/rrx
Description	Uses the VBScript <b>Regular Expression Pattern</b> you enter as a parameter to search for a matching pattern in the current object's Text value.
Parameters	<i>String</i> value of the Regular Expression. The expression can include any Regular Expression characters (see the example.)
Level	All, but generally at the Field level.
Returns	<i>True</i> if a match occurs. Otherwise, <i>False</i> .
Example	PatternInField(([\^\b\s\n\r]Inv[oOO][iItl1]ce[\b\s]*))

### ReadBatchVariable

Actions Library	validations.rra/rrx
Description	Assigns the value of a variable of the <b>Batch</b> object to the current <b>Field</b> object.  If the action is not placed at the field level, then a variable called Text will be created on the current object, and the found value will be written to this variable.
Parameter	The name of the <b>Batch</b> object's property.
Level	All
Returns	<i>False</i> if the parameter is not a valid <b>Batch</b> -level variable; otherwise, <i>True</i>
Example	ReadBatchVariable(ScanDate)  A <b>Validate</b> rule with this action could transfer the Date on which the batch was created – its <i>ScanDate</i> value – to a comparable and selected <i>ScanDate</i> <b>Field</b> object.
See also...	ReadDocumentVariable, ReadPageVariable

## ReadCurrentObjVariable

Actions Library	validations.rra/rxx
Description	<p>Reads the Text value of the bound object of the Document Hierarchy.</p> <p>For a <b>Field</b> object, this is the value of the object's <b>Text</b> property. For objects at other levels, the value is in a <i>runtime</i> variable of the bound object (see the examples.)</p>
Parameters	The name of the bound object's property or variable.
Level	All
Returns	<i>False</i> if the property or variable specified by the parameter is not found. Otherwise, <i>True</i> .
Example	<p>ReadCurrentObjVariable(Text)</p> <p>ReadCurrentObjVariable(TemplateID)</p>

## ReadDocumentVariable

Actions Library	validations.rra/rxx
Description	<p>Assigns the value of a property of the <b>Document</b> object to the current <b>Field</b> object.</p> <p>If the action is not placed at the field level, then a variable called Text will be created in the current object, and the found value will be written to this variable.</p>
Parameter	The name of the <b>Document</b> object's variable.
Level	All
Returns	False if the parameter is not a valid <b>Document</b> -level variable. Otherwise, <i>True</i> .
Example	<p>ReadDocumentVariable(PageCount)</p> <p>A <b>Validate</b> rule with this action could transfer the number of pages in the document to the selected <i>TotalPages</i> <b>Field</b> object.</p>
See also...	ReadBatchVariable, ReadPageVariable

## ReadFIELD

Actions Library	validations.rra/rxx
-----------------	---------------------

## Validate Actions

---

Description	Retrieves the <i>captured</i> value from a sibling <b>Field</b> object you specify as the parameter, and assigns the value to the current <b>Field</b> object.  If the action is not bound at the Field level, then a variable called Text will be created in the current object, and the found value will be written to this variable.
Parameter	The name of the source <b>Field</b> object.
Level	All
Returns	<i>False</i> if the parameter does not match a <b>Field</b> object's name. Otherwise, <i>True</i> .
Example	ReadFIELD(Date)  This action assigns the <i>captured</i> value in the <i>Date</i> field to the <i>working</i> fingerprint's current field.
See also...	WriteFIELD

## ReadPageVariable

Actions Library	validations.rra/rrx
Description	Assigns the value of a variable of the <b>Page</b> object to the selected <b>Field</b> object. If the action is not placed at the field level, then a variable called Text will be created in the current object, and the found value will be written to this variable.
Parameter	The name of the <b>Page</b> object's variable.
Level	All
Returns	False if the parameter is not a valid <b>Page</b> -level variable; otherwise, <i>True</i> .
Example	ReadPageVariable(FieldCount)  A <b>Validate</b> rule with this action could transfer the number of fields in the <b>Page</b> object that is the "parent" of a selected <b>Field</b> object such as <i>TotalFields</i> .  <b>Important!</b> A Document Hierarchy may well include multiple <b>Page</b> objects – and certain variables that are the same for both pages. This action refers to variables of the <b>Page</b> object to which the selected <b>Field</b> object belongs.
See also...	ReadBatchVariable, ReadDocumentVariable

## RemoveSpaces

Actions Library	validations.rra/rrx
Description	Deletes characters that are spaces.
Parameters	None
Level	All, but generally at the Field level.
Returns	<i>False</i> if the action is unable to isolate the spaces in a field's value. Otherwise, <i>True</i> .
Example	RemoveSpaces()

## ReplaceValue

Actions Library	validations.rra/rrx
Description	Replaces a character or string of characters in the current <b>Field</b> object's <i>captured</i> value with a <i>String</i> you enter as one of the parameters.
Parameter	<ol style="list-style-type: none"> <li>1.) The character or string of characters to be replaced; defaults to a space</li> <li>2.) The character(s) of the replacement <i>String</i>.</li> <li>3.) The number of times replacement is to occur. The default is "1" and "*" replaces <i>all</i> instances.</li> </ol>
Level	Field level only.
Returns	Always <i>True</i> .
Example	<p>ReplaceValue(./,*)</p> <p>01.02.2005 becomes 01/02/2005</p>

## ReplaceValue\_AtPosition

Actions Library	validations.rra/rrx
Description	Replaces the value at a specific position with a replacement string, or deletes the value at the designated position
Parameter	Two-part comma-separated value:  1.) The position that contains the value to be replaced.  2.)The replacement string; this parameter defaults to "" indicating a deletion.
Level	Field only.
Returns	True if the character replacement is successful; Otherwise, False .
Example	<b>ReplaceValue_AtPosition(3, /)</b>

## RightTruncate

Actions Library	validations.rra/rrx
Description	Deletes characters from the end of the current <b>Field</b> object's <i>captured</i> value until the value's length equals the length specified by the parameter.
Parameter	A number <i>n</i> that is the value's maximum length.
Level	Field level only.
Returns	<i>False</i> if the parameter is not <i>Numeric</i> . Otherwise, <i>True</i> .
Example	<b>RightTruncate(8)</b> reduces the following value:  3,344.001 becomes 3,344.00
See also...	LeftTruncate



## SaveAsBatchVariable

Actions Library	validations.rra/rrx
Description	Assigns a value to a variable of the <b>Batch</b> object.
Parameter	<p>1.) The name of the <b>Batch</b> object's variable;</p> <p>2.) The <i>String</i> value to be assigned.</p> <p>These are comma-separated parameters, and the second is <i>optional</i>. If you do not include this parameter, the action will automatically assign the value of the selected <b>Field</b> object.</p>
Level	All
Returns	Always <i>True</i> .
Examples	<p><b>SaveAsBatchVariable(FormID)</b></p> <p>SaveAsBatchVariable(FormID,SodaSurvey)</p> <p>The first example assigns the <i>captured</i> value of the selected <b>Field</b> object to the <b>Batch</b> object's <i>FormID</i> variable.</p> <p>The second example assigns "SodaSurvey" to the variable.</p>
See also...	ReadBatchVariable

## SaveAsCurrentObjVariable

Actions Library	validations.rra/rrx
Description	Assigns a value to a variable of the current object.
Parameter	<p>1.) The name of the variable;</p> <p>2.) The <i>String</i> value to be assigned.</p> <p>These are comma-separated parameters, and the second is <i>optional</i>. If you do not include this parameter, the action will automatically assign the <i>captured</i> value of the selected <b>Field</b> object to the variable.</p>
Level	All
Returns	Always <i>True</i> .
Examples	<p>SaveAsCurrentObjVariable(Type)</p> <p>SaveAsCurrentObj(Type,Front)</p> <p>The first example assigns the <i>captured</i> value of the selected <b>Field</b> object to that object's <i>Type</i> variable.</p> <p>The second example assigns "Front" to the <i>Type</i> variable of the selected object.</p>

## SaveAsDocumentVariable

Actions Library	validations.rra/rrx
Description	Assigns a value to a variable in the current <b>Document</b> object
Parameter	<p>1.) The name of the variable;</p> <p>2.) The <i>String</i> value to be assigned.</p> <p>These are comma-separated parameters, and the second is <i>optional</i>. If you do not include this parameter, the action will automatically assign the <i>captured</i> value of the selected <b>Field</b> object to the variable.</p>
Level	Document, Page, or Field.
Returns	Always <i>True</i> .
Examples	<p>SaveAsDocumentVariable(DocName)</p> <p>SaveAsDocumentVariable(DocName,SurveyOne)</p> <p>The first example assigns the <i>captured</i> value of the current <b>Field</b> object to the <b>Document</b> object's <i>DocName</i> variable.</p> <p>The second example assigns "SurveyOne" to the variable.</p> <p><b>Remember:</b> If the Document Hierarchy has multiple <b>Document</b> objects with identical variables, the variable you specify is a property of the <b>Document</b> in which the current <b>Field</b> is located.</p>
See also...	ReadDocumentVariable

## SaveAsPageVariable

Actions Library	validations.rra/rrx
Description	Assigns a value to a variable of the current <b>Page</b> object.
Parameter	1.) The name of the variable; 2.) The <i>String</i> value to be assigned.  These are comma-separated parameters, and the second is <i>optional</i> . If you do not include this parameter, the action will automatically assign the <i>captured</i> value of the selected <b>Field</b> object to the variable.
Level	Page or Field Level.
Returns	Always <i>True</i> .
Examples	SaveAsPageVariable(Amount) SaveAsPageVariable(Amount,15000)  The first example assigns the <i>captured</i> value of the current <b>Field</b> object to the current <b>Page's</b> <i>Amount</i> variable.  The second example assigns "15000" to the variable.
See also...	ReadPageVariable

## SplitLeft

Actions Library	validations.rra/rrx
Description	Splits a <b>Field</b> object's captured value at the first instance of the character you specify as a parameter.  The action deletes all characters to the <i>right</i> of the separating character, leaving the separating character in the field's value.
Parameter	<i>String</i> value of the separating character.
Level	Field level only.
Returns	Always <i>True</i> .
Example	SplitLeft(c)  If the object's value is "Description," this action truncates it to "Desc".
See also...	SplitRight

## SplitRight

Actions Library	validations.rra/rrx
Description	<p>Splits a <b>Field</b> object's captured value at the first instance of the character you specify as a parameter.</p> <p>The action deletes all characters to the <i>left</i> of the separating character, as well as the separation character. All text to the right of the separating character remains.</p>
Parameter	<i>String</i> value of the separating character.
Level	Field level only.
Returns	Always <i>True</i> .
Example	<p>SplitRight(=)</p> <p>If the object's value is "InvNumber=A1234," this action truncates it to "A1234".</p>
See also...	SplitLeft

## SumFIELDS

Actions Library	validations.rra/rrx
Description	<p>Sums <i>captured</i> values of any “child” <b>Field</b> if a child object’s <b>Type</b> property is identical to the Type you specify as a parameter.</p> <p>Alternatively, the actions sums values assigned to a variable of the child <b>Field</b> objects – a variable which is the same as the variable you enter as a parameter.</p> <p><b>Remember:</b> This action must be applied to the parent <b>Field</b> object.</p>
Parameter	<i>String</i> value of a <b>Field</b> object’s <b>Type</b> property <i>or</i> the name of a variable.
Level	Field level only.
Returns	Always <i>True</i> .
Examples	<p>SumFIELDS(Detail)</p> <p>SumFIELDS(LineTotal)</p> <p>The first action above sums the <i>captured</i> values of <i>Detail</i> <b>Field</b> objects which are children of the current <b>Field</b> object.</p> <p>The second action sums values assigned to the <i>LineTotal</i> variable of child <b>Field</b> objects.</p> <p>The result, in both cases, is a <i>Long</i> number.</p>

## TimeStamp

Actions Library	validations.rra/rrx
Description	Updates the current <b>Field</b> object with the current Time.
Parameter	A time format (for example, HH:MM:SS, or HH:MM.) * defaults to HH:MM:SS
Level	Field level only.
Returns	Always <i>True</i> .
Examples	<b>TimeStamp(*)</b> produces 09:20:02  <b>TimeStamp(HH:MM)</b> produces 09:20
See also...	DateStamp

## TrailingZeros

Actions Library	validations.rra/rrx
Description	Adds zeros (“0”) to the end of the current <b>Field’s captured</b> value until the value’s length reaches the maximum <i>n</i> you enter as the parameter.
Parameter	A number <i>n</i> which is the value’s maximum length.
Level	Field level only.
Returns	<i>False</i> if the parameter you enter is not <i>numeric</i> ; otherwise, <i>True</i> .
Example	TrailingZeros(10) 2240.00 becomes 2240.00000
See also...	LeadingZeros

### TrimSpaces

Actions Library	validations.rra/rrx
Description	Deletes extra spaces at the beginning and end of the current <b>Field</b> object's <i>captured</i> value.
Parameters	None
Level	Field level only.
Returns	Always <i>True</i> .
Example	TrimSpaces()  456.11_ _ _ _ becomes 456.11
See also...	RightTruncate

### WriteField

Actions Library	validations.rra/rrx
Description	Assigns the current <b>Field</b> object's <i>captured</i> value to a sibling <b>Field</b> object you specify as the parameter.
Parameter	The name of the target <b>Field</b> object.
Level	Field level only.
Returns	<i>False</i> if the parameter does not match a <b>Field</b> object's name. Otherwise, <i>True</i> .
Example	WriteFIELD (Date)  This action places the <i>captured</i> value of the current field into the <i>Date</i> field.
See also...	ReadField



## Vote Action

This action is used for second or third pass data entry.

### VoteFld

Actions Library	Vote.rra
Description	<p>Checks to see if the data entered for by the first Data Entry operator matches the data enter by the second Data Entry operator.</p> <p><i>Failed</i> (mismatched) values set the confidence for the entire string to '1', therefore flagging the field as Low Confidence. <i>Positive</i> matches set the entire string's confidence to '9' (High Confidence).</p> <p>This action needs to run after the second Data Entry task has been completed.</p> <p>This action is used for workflows using third pass data entry.</p>
Parameters	None
Level	Field
Returns	<i>False</i> if the values do not match. Otherwise, <i>True</i> .
Example	VoteFld()

## vScan Actions

*Rulemanager's vScan* actions initiate “virtual” scanning procedures that load image files from previously scanned images. Usually, the images are .tif file - or other image files that reside temporarily in a processing directory, where they await the attention of a vScan task.

A vScan task is a *batch creation* task: after processing a set of images, it places the files in a batch it has just defined. A RuleSet using vScan actions must run at the **Batch** object level of the application's Document Hierarchy.

The following tables describe the vScan actions.

### AddDocument

Actions Library	vscan.rra
Description	<p>Places all scanned pages into a single, batch-wide document.</p> <p>The action also adds <b>ED</b>, <b>AD</b>, <b>EP</b> and <b>AP</b> properties to a task's Page file (.xml).</p> <p>This action allows a VScan task to create a batch that has all the same characteristics of a batch that is created using a standard Taskmaster scan task such as kScan or iScan.</p>
Parameters	None
Level	Batch level only.
Returns	<i>False</i> if the Document Hierarchy contains one or more <b>Document</b> objects when this action is invoked. In this case, the action will not add a document. Otherwise, <i>True</i> .
Example	AddDocument()

## CopyFile

Actions Library	vscan.rra
Description	A housekeeping action that copies the current source image file to a location you specify <i>and</i> specifies the copied file's extension.
Parameters	1) The <i>String</i> value of the name of the target file system folder. 2) The file extension you wish to use for each file. The action defaults to <b>tif</b> if this parameter is blank.
Level	Batch level only.
Returns	Always <i>True</i> .
Example	CopyFile(C:\ParentDir\Invoice\Images\copies\tif) Scan()  This example copies the source image file to the "copies" folder and then adds it to the batch.  It does <i>not</i> delete the image from the source folder.
See also...	Scan

## DeleteImageFile

Actions Library	vscan.rra
Description	A housekeeping action that deletes the current source image file.
Parameters	None
Level	Batch level only.
Returns	Always <i>True</i> .
Example	<b>DeleteImageFile()</b> Scan()  This sequence deletes the image files from their current location after they have been added to the current batch folder.

## MoveImageFileToDirectory

Actions Library	vscan.rra
Description	A housekeeping action that moves the current Image file to a location you specify.
Parameters	The full path to the target location of the Image file.
Level	Batch level only.
Returns	<i>False</i> if the parameter is not a valid directory, or if permission to access / write to the directory is denied. Otherwise, <i>True</i> .
Example	<pre>MoveImageFileToDirectory(C:\ParentDir\Invoice\   backup) Scan()</pre> <p>This sequence copies the source image files to the current batch, then moves the Image files to the specified directory.</p>

## Scan

Actions Library	vscan.rra
Description	<p>Scans a set of waiting Image files, according to the parameters set by earlier actions (see the example.)</p> <p>This is usually the last action in a <b>vScan</b> rule.</p>
Parameters	None
Level	Batch level only.
Returns	<i>False</i> if a <b>SetSourceDirectory</b> action does not precede this action. Otherwise, <i>True</i> .
Example	<pre>SetImageType(.tif) SetSourceDirectory(C:\ParentDir\Invoice\Images) SetMaxImageFiles(100) <b>Scan()</b></pre> <p>These are the elements of a sample <b>vScan Rule</b>. The <b>Scan</b> action will load the specified number of images (if available) from the specified folder into the current batch folder.</p>

## SearchInSubdirectory

Actions Library	vscan.rra
Description	Looks for source files in sub-directories of the directory you designated with a <b>SetSourceDirectory</b> action.
Parameters	None
Level	Batch level only.
Returns	Always <i>True</i> .
Example	<p>SetSourceDirectory(C:\ParentDir\Invoice\Images)  <b>SearchInSubdirectory()</b></p> <p>If the <b>Images</b> directory in this example includes sub-directories, this action will direct the <b>Scan</b> action to process the contents of the sub-directories.</p>
See also...	SetSourceDirectory

## SetFastMode

Actions Library	vscan.rra
Description	<p>Increases speed by preventing the <b>Scan</b> action from opening the files it is scanning.</p> <p><b>Alert!</b> This action must be used if a task is scanning PDF files.</p>
Parameters	None
Level	Batch level only.
Returns	Always <i>True</i> .
Example	SetImageType(.pdf)                      SetFastMode()                      Scan()
See also	SetImageType, Scan

## SetImageType

Actions Library	vscan.rra
Description	Uses the value of a file extension to specify the type of Image files the task will scan.  This is an <i>optional vScan</i> action: the task will scan .tif files by default.
Parameter	<i>String</i> value of the file's identifying extension – <b>tif</b> , <b>.tif</b> , <b>gif</b> , <b>.gif</b> , <b>jpg</b> , or <b>.jpg</b> .  If you do not include a period (“.”), the action will add it automatically.
Level	Batch level only.
Returns	<i>True</i> if the parameter is one of the values specified above. (These values are <i>not</i> case-sensitive.) Otherwise, <i>False</i> – even if the parameter is a null value.
Examples	SetImageType(.jpg)  SetImageType(jpg)  Either format is acceptable.

## SetMailSourceFolder

Actions Library	vscan.rra
Description	Searches the location you specify for e-mails that contain Image file attachments.
Parameters	<i>String</i> value of the Outlook folder you want to search.
Level	Batch level only.
Returns	Always <i>True</i> .
Example	<p>SetMailSourceFolder(Inbox/Images) Scan()</p> <p>This example will search the <b>Images</b> subfolder for any e-mails with attachments, and copy these Image files into the newly-created batch.</p>

## SetMaxImageFiles

Actions Library	vscan.rra
Description	Limits the number of Image files the vScan task will add to a single batch.
Parameters	<p><i>String</i> value specifying the maximum number of files.</p> <p>If you do <b>not</b> enter a value, the task will scan all images in the target folder – and this action will return <i>True</i> (below).</p>
Level	Batch level only.
Returns	<i>False</i> if the parameter is not <i>Numeric</i> . Otherwise, <i>True</i> .
Example	<p>SetMaxImageFiles(50)</p> <p>Sets the maximum number of files to add to a batch at fifty.</p> <p><b>Remember!</b> A vScan task is a <i>Batch Creation</i> task: it sets up a new batch each time it scans Image files.</p>

## SetMultiPageTiff

Actions Library	vscan.rra
Description	Permits the use of multipage source image files.
Parameters	None
Level	Batch level only.
Returns	Always <i>True</i> .
Example	<pre>SetImageType(.tif) <b>SetMultiPageTiff()</b> Scan()</pre> <p>If the Scan action in this sequence encounters a multipage .tif file, then it will read each one into the current batch as a separate image, thereby “bursting” the multipage file into individual images.</p>

## SetSortOrder

Actions Library	vscan.rra
Description	Sets the order in which Image files will be imported, in response to the parameters you enter.
Parameter	<p>Two comma-separated <i>String</i> values:</p> <p>1) Designation of the images’ sorting field: ImageName, Type, DateCreated, DateLastAccessed, DateLastModified, or Size.</p> <p>2) <i>Optional</i>: ASC (Ascending) or DESC (Descending). If you do not include this parameter, the action defaults to ASC.</p>
Level	Batch level only.
Returns	<i>False</i> if the parameters are not valid. Otherwise, <i>True</i> .
Example	<b>SetSortOrder(Name,ASC)</b> Scan()



## SetSourceDirectory

Actions Library	vscan.rra
Description	<p>Indicates the name and path of the directory containing the Image files to be scanned.</p> <p>This is a <b>required</b> action; a task which employs a <b>vScan</b> rule cannot process images unless it has this key locator.</p>
Parameter	<i>String</i> value of the directory's name and path.
Level	Batch level only.
Returns	<i>False</i> if the parameter is blank or the directory does not exist. Otherwise, <i>True</i> .
Example	SetSourceDirectory(C:\ParentDir\Invoice\Images)

## Zone Actions

**Zone** actions capture information from *zoned* fields. These are the **Field** objects on a fingerprint that you have submitted to the zoning procedures of the **Rule** fingerprint with parameters that specify a field's size and position.

- ✓ You can apply *Rulemanager's* zoning technology to individual **Field** objects within the fingerprint – to the *APT* application's *Number* or *Total* field, for example – or to **Field** objects containing child objects. In the *APT* example, this includes the **Field** objects of Line Item data: *DETAILS* and *LINEITEM*, and *ItemID*, *ItemDesc*, *Quantity*, *Price* and *LineTotal*.

A **Locate** rule uses **Zone** actions to solicit a field's zone specifications...its size and location. The rule can then search the current page for a field of comparable size and location. If the search is successful, the application can use the zone specifications to **register** the field more precisely within the fingerprint; to extract its *recognized* value; and to add the value to the Data file of the current page.

These actions require that you have already set up zones on the fingerprint that correspond with **Field** objects of the Document Hierarchy (Chapter 3).

A **Locate** RuleSet can contain rules that use **Zone** actions and rules that contain **Locate** actions. For example, the RuleSet might begin with a **Locate** rule using **Zone** actions followed by a **Locate** rule with **Locate** actions. If the first rule fails for some reason (perhaps the rule can't find an "anchor" field or zones were not previously established) the subsequent rule with **Locate** actions will be run.

**Zone** actions fall into two categories: *Procedural* (on the next page) and *Dimensional* (Page 296).

## Zone Actions Procedural

### AdjustZonesToImageOffset

Actions Library	Zones.rra
Description	Adjusts the zone offsets for all children of the calling component.  The offset adjustment is based on the data calculated from matching the current page or a Pattern area to a fingerprint. The resulting offset amount must be greater than 0.
Parameters	None
Level	Page or Field level
Returns	Always <i>True</i> .
Example	AdjustZonesToImageOffset()

### AnchorPage

Actions Library	Zones.rra
Description	Uses the size and location parameters of a <i>zoned</i> Anchor field on the fingerprint to identify a matching Anchor field on a current page.
Parameters	None
Level	Page level.
Returns	<i>True</i> if the current page contains an <b>Anchor</b> field. Otherwise, <i>False</i> .
Example	AnchorPage()  Unlike the <b>RegisterPage</b> action, this action does <i>not</i> require a matching Anchor word. Instead, if it locates an Anchor field within the zone coordinates supplied by the fingerprint, the action will use zone data to register all fields of the current page.  This advanced version of the <b>RegisterPage</b> action has advantages and disadvantages. Because it does not look for an Anchor word, matching can be more rapid and less frustrating.  However, <b>AnchorPage</b> compromises safety because it surrenders what might be a valuable check on the matching process.
See also...	RegisterPage

### **CalculateLocalOffset**

Actions Library	Zones.rra
Description	Calculates the X and/or Y offset amount for the calling field.
Parameters	X and/or Y  The action uses these parameters to calculate a new parent page Image_Offset value by comparing the fingerprint zone for the calling field against the current field zone.
Level	Field only.
Returns	Always True.
Example	<b>CalculateLocalOffset (XY)</b>

### **CreateBlockCCO**

Actions Library	Zones.rra
Description	Creates a temporary in memory CCO object, containing only the words and lines in the calling fields zone position - using the source page's CCO file.  The action is useful when searching images where the data is located asymetrically across the page. Using this action focuses the CCO to only look at the words and lines within the defining zone for all future searches for this page.  This does not affect the runtime CCO. Reloading the page will reload the CCO bound to the page, releasing this temporary CCO.
Parameters	None.
Level	Field
Returns	Always true
Example	<b>CreateBlockCCO ( )</b>

## FindBlocks\_WhiteSpace

Actions Library	Zones.rra
Description	<p>Uses a vertical white space (pixels) to find blocks of data within the current source page.</p> <p>The action returns each block's position assigned to a series of repeating fields based on the first child field of the calling object.</p>
Parameters	A String value indicating the number of pixels between lines.
Level	Field with one child field.
Returns	False if the action cannot divide create any child field/zones. Otherwise, True
Example	FindBlocks_WhiteSpace(27)

## FindDataBlocks

Actions Library	Zones.rra
Description	<p>Uses Start and End key words to find blocks of data within the current source page.</p> <p>The action returns each block's position assigned to a series of repeating fields based on the first child field of the calling object</p>
Parameters	Key word Start Value, and its End Value.
Level	Page.
Returns	True if the action can locate the Data block indicated by the parameter. Otherwise, False.
Example	FindDataBlocks(FROM,THRU)

## FindLineItems

Actions Library	Zones.rra
Description	Uses settings for a previously-established zone covering a <b>block</b> of Line Item Detail, and zones for individual <b>Line Items</b> , to assemble a portion of a current page, limited to Line Item Detail
Parameter	Boolean.  <i>True</i> if the action is to include both Line Item fields and sub-fields in the current page.  <i>False</i> if a Line Item is to show primary fields only
Level	Field level
Returns	<i>False</i> if there is no zone or position information for the Line Item block and its sub-fields, <i>or</i> if the parent block does not contain Line Item children. Otherwise, <i>True</i> .
Example	FindLineItems(True)  This action retrieves <b>zonal</b> information from the fingerprint about a block of Line Item Detail, individual rows, and their fields and sub-fields.  The action applies this data to a full-length current page as it constructs a separate fingerprint containing just Line Item Detail

## FindRegExBlocks

Actions Library	Zones.rra
Description	Uses a Regular Expression to find blocks of data within the current source page.  The action returns each block's position assigned to a series of repeating fields based on the first child field of the calling object.
Parameters	Regular Expression that contains the data block's Start Value, and its End Value.  Start and End values can be adjusted up or down by x lines, using the 3 <sup>rd</sup> (adj top) and 4 <sup>th</sup> (adj bottom) CSV positions.
Level	Page level
Returns	True if the action can locate the Data block indicated by the parameter. Otherwise, False.
Example	FindRegExBlocks(/bFROM/b,/bTHRU/b)

## GetZoneText

Actions Library	Zones.rra
Description	Retrieves the text in a <i>zoned</i> object of the current page.
Parameters	None.
Level	Field level
Returns	Always <i>True</i> .
Example	<p><b>GetZoneText ( )</b></p> <p>This action assumes that you have established zone parameters for this <b>Field</b> object of the Document Hierarchy.</p>

## InheritParentPosition

Actions Library	Zones.rra
Description	Provides the bound child object of the Document Hierarchy with the zone parameters of a parent object identified by the parameter.
Parameters	<p>Case-sensitive string value of the parent object's name.</p> <p><i>Details</i>, for example, is the name of a parent <b>Field</b> object in the <i>Invoices</i> application. If the bound object is <i>LINEITEM</i>, the action will provide this child object with the zone parameters of the parent.</p>
Level	Field.
Returns	False if the action cannot locate the parent field. Otherwise, True.
Example	<b>InheritParentPosition(Details)</b>

## LoadBlockCCO

Actions Library	Zones.rra
Description	Replaces the current page with the <i>LineItemBlock</i> fingerprint you created with the <b>FindLineItems</b> action.
Parameter	None
Level	Page or Field level
Returns	<i>True</i> if a Fingerprint file (.cco) file containing block information can be loaded. Otherwise, <i>False</i> .
Example	<p>FindLineItems(True)  <b>LoadBlockCCO()</b></p> <p><i>Be careful!</i> This deceptively simple combination can have quite an impact because it limits the scope of the current page to Line Item Detail, and then retains the abridged version as the fingerprint's file (.cco).</p>

## LoadZones

Actions Library	Zones.rra
Description	<p>Loads position information for each node in the calling object and children.</p> <p>The action pre-adjust's these values based on offset information stored in an 'Image_Offset' variable at any node level.</p> <p>Position information is based on the Document Hierarchy position for the Fingerprint ID passed as a parameter.</p> <p>The offset value is applied to all child objects of a node where an 'Image_Offset' variable is found; unless overwritten by a child node also having an Image_Offset value to apply.</p>
Parameters	String value of the Fingerprint ID.
Level	Page or Field
Returns	<i>True</i> if a Fingerprint file (.cco) file containing block information can be loaded. Otherwise, <i>False</i> ..
Example	<b>LoadZones ( 1013 )</b>



## MCCOPositionAdjust

Actions Library	Zones.rra
Description	<p>Wraps additional pages of a multi-page document within the first page.</p> <p>It is an important action for any application that processes multi-page forms because it ensures that the data on every page invoice becomes part of a combined single-page document.</p> <p>Functionally this normalizes all zones for multipage documents to the images for which they are sourced.</p>
Parameters	None
Level	Page
Returns	False if the current document does not consist of more than one <i>source</i> page. Otherwise, True.
Example	<code>MCCOPositionAdjust()</code>

## PopulateZNField

Actions Library	Zones.rra
Description	<p>Populates the fingerprint's Data file with the <i>recognized</i> value contained inside the <b>Unit Field</b> object's Zone. This action should be used with Unit Fields such as the <i>APT</i> application's <i>Total</i> or <i>Number</i> field.</p>
Parameters	None
Level	Field level.
Returns	<i>True</i> if a fingerprint's .cco file containing block information can be found. Otherwise, <i>False</i> .
Example	<p>PopulateZNField()</p> <p>Remember that you have to "zone" the <b>Unit Field</b> object before a rule with this action can operate effectively.</p>
See also...	PopulateZNLineItemField

## PopulateZNLIneltemField

Actions Library	Zones.rra
Description	<p>Populates the fingerprint's Data file with the <i>recognized</i> value contained inside the zone of a child <b>Field</b> object of a <i>LINEITEM</i> parent field.</p> <p>This action should only be used with sub-fields of the <i>LINEITEM</i> field - <i>ItemID</i>, <i>ItemDesc</i>, <i>Quantity</i>, <i>Price</i> in the <i>APT</i> application.</p>
Parameters	None
Level	Field level
Returns	True if the bound <b>Field</b> object is the <i>child</i> of a <i>parent Field</i> object - <b>and</b> if a fingerprint's .cco file containing block information can be found. Otherwise, <i>False</i> .
Example	<p>PopulateZNLIneltemField()</p> <p>Remember that you have to “zone” the sub-field before a rule with this action can operate effectively.</p>
See also...	PopulateZNField

## ReadZones

Actions Library	Zones.rra
Description	Copies all zone positions from the fingerprint to the current page.
Parameters	None
Level	Page or Field level
Returns	Always <i>True</i> .
Example	<p><b>ReadZones()</b> RegisterPage()</p> <p>This sequence provides a rule with the fingerprint's 'default' zone information that can be used if the Anchor field and search word are not found.</p>

## RegisterPage

Actions Library	Zones.rra
Description	<p>Links the current page with zone information stored in the fingerprint.</p> <p><i>Alert!</i> A rule with this action should apply to a <b>Page</b> object of the Document Hierarchy.</p>
Parameters	None
Level	Page level
Returns	Always <i>True</i> .
Example	<p>RegisterPage()</p> <p>This action checks the fingerprint for each <b>Field</b> object's zone specifications, and uses these specifications to register corresponding fields in the current page.</p>
See also...	AnchorPage

## ScanDetails

Actions Library	Zones.rra
Description	<p>This action searches the <b>DETAILS Field</b> object's zone for instances of a <b>LINEITEM Field</b> object.</p> <p>A <b>ScanDetails</b> action should be applied to the <b>DETAILS Field</b> object (in the <i>APT</i> application).</p>
Parameters	None
Level	Field level
Returns	<i>True</i> if the bound <b>Field</b> object contains lines of data. Otherwise, <i>False</i> .
Example	<p>ScanDetails()</p> <p>This action will capture <i>all</i> potential <b>LINEITEM</b> rows within the parent <b>DETAILS</b> field – even rows that may not fit your criteria for content.</p> <p><b>Alert!</b> You will need to run additional RuleSets to delete ineligible Line Items. (In the <i>APT</i> application, these are the <b>Clean</b> and <b>Filter</b> RuleSets.)</p>
See also...	ScanLineItem

## ScanDetailsByLine

Actions Library	Zones.rra
Description	Searches the <b>DETAILS Field</b> object's zone for instances of a <b>LINEITEM Field</b> object. Captures the data based on the number of lines (parameter) in a Line Item table, row by row.
Parameters	Number of lines in each lineitem.
Level	Field level.
Returns	True if the bound <b>Field</b> object contains lines of data. Otherwise, False
Example	<p><b>ScanDetailsByLine(2)</b></p> <p>This action will capture all potential <b>LINEITEM</b> rows (consisting of 2 lines each) within the parent <b>DETAILS</b> field – even rows that may not fit your criteria for content.</p> <p><b>Alert!</b> You will need to run additional RuleSets to delete ineligible Line Items. (In the <i>Invoices</i> application, these are the <b>Clean</b> and <b>Filter</b> RuleSets.)</p>

## ScanDetailsByVSpace

Actions Library	Zones.rra
Description	Searches the DETAILS <b>Field</b> object's zone for instances of a <i>LINEITEM Field</i> object – and captures the data based on the number of vertical pixels (parameter) in a Line Item table, row by row
Parameters	Number of vertical pixels in each lineitem.
Level	Field level.
Returns	True if the bound <b>Field</b> object contains lines of data. Otherwise, False.
Example	<p><b>ScanDetailsByLines(45)</b></p> <p>This action will capture all potential LINEITEM rows (consisting of 45 pixels lines each) within the parent DETAILS field – even rows that may not fit your criteria for content.</p> <p><b>Alert!</b> You will need to run additional RuleSets to delete ineligible Line Items. (In the <i>Invoices</i> application, these are the <b>Clean</b> and <b>Filter</b> RuleSets.)</p>

## ScanLineItem

Actions Library	Zones.rra
Description	This action searches a <i>LINEITEM Field</i> object's zone for instances of <b>child Field</b> objects such as <i>ItemID</i> , <i>ItemDesc</i> , <i>Quantity</i> , <i>Price</i> , <i>LineTotal</i> (in the <i>APT</i> application.)
Parameters	None
Level	Field level
Returns	Always <i>True</i> .
Example	<p>ScanLineItem()</p> <p>This action will capture each sub-field within the <b>parent LINEITEM</b> row.</p>
See also...	ScanDetails

## SetEOL

Actions Library	Zones.rra
Description	Sets the “End Of Line” character that will be used to separate data from a zone with multiple lines of text.  If this action is not used, the default character is a space.
Parameters	String Value
Level	All
Returns	Always <i>True</i> .
Example	SetEOL()  Sets the End of Line character to the ' ' (pipe) character. A capture zone with two lines of text will have the <i>captured</i> value separated by this new character.
See also...	SetEOL_CRLF

**SetEOL\_CRLF**

Actions Library	Zones.rra
Description	<p>Sets the “End Of Line” character that will be used to separate data from a zone with multiple lines of text to the carriage return and Line Feed characters (ASCII values 13 &amp; 10).</p> <p>If this action is not used, the default character is a space.</p>
Parameters	None
Level	All
Returns	Always <i>True</i> .
Example	<p>SetEOL_Crlf()</p> <p>Sets the End of Line character to the Carriage Return and Line Feed characters. A capture zone with two lines of text will have the captured value separated by these characters.</p>
See also...	SetEOL()

## Zone Actions Dimensional

The *Dimensional* actions manipulate specific *zonal* values of a field in the fingerprint to more accurately reflect the size and location of a current page's value for this field.

The *Dimensional* actions can be very helpful if you are able to anticipate difficulties with the length and placement of a field's actual data.

Keep in mind that hand-printed entries are far more likely to stray beyond the boundaries of their zones. If your form permits data of this kind, take time to become familiar and comfortable with the *Dimensional Zone* actions.

Another approach uses the positioning coordinates of a value's **image** to adjust the location of the field's zone for the current page.

**Remember!** Some *Dimensional* actions rely on a field's word to modify *zonal* properties; others use a field's image.

### ZoneBOTTOM\_LowerBound

Actions Library	Zones.rra
Description	Uses the lower boundary of the current field's <b>word</b> to specify the bottom of the field's <b>zone</b> .  This can result in a taller or shorter zone, depending on the location of the word's upper boundary.
Parameters	None
Level	Field level
Returns	Always <i>True</i> .
Example	ZoneBOTTOM_LowerBound()  If you find that an unacceptably high percentage of values for a particular field are entered a line too high, consider the use of this action in a follow-up <b>Locate</b> rule to search for the misplaced value.
See also...	ZoneBOTTOM_UpperBound



## ZoneBOTTOM\_ImageBottom

Actions Library	Zones.rra
Description	<p>Uses the <i>upper</i> boundary of the current field's <b>image</b> to designate the <i>top</i> edge of the field's <b>zone</b>.</p> <p>This can extend or contract the zone's width depending on the placement of the image's left side.</p>
Parameters	None
Level	Field level
Returns	Always <i>True</i> .
Example	ZoneBOTTOM_ImageBottom()
See also...	ZoneTOP_ImageTop

## ZoneBOTTOM\_UpperBound

Actions Library	Zones.rra
Description	<p>Uses the lower boundary of the current field's <b>zone</b> to specify the top of the field's <b>word</b>.</p> <p>This can result in a taller or shorter word, depending on the location of the zone's lower boundary.</p>
Parameters	None
Level	Field level
Returns	Always <i>True</i> .
Example	<p>ZoneBOTTOM_UpperBound()</p> <p>This action is helpful when experience shows that a certain number of values in this field extend beyond the zone's lower limit. A follow-up <b>Locate</b> rule with this action can find those values.</p>
See also...	ZoneBOTTOM_LowerBound

## ZoneImage\_SaveAs

Actions Library	Zones.rra
Description	<p>Saves the current <b>Objects Zone</b> area of an image as a separate tif Image file.</p> <p>The file is always placed in the <b>Batches</b> directory; saving to the current page will create a multi-page tif with the additional image being the zone.</p> <p>Formatting allows the following special options:</p> <ul style="list-style-type: none"> <li>+@BATCHID' Adds BatchID to Zone Image Name")</li> <li>+@ID' Adds Object ID to Zone Image Name")</li> <li>+@STATUS' Adds Object Status to Zone Image Name</li> <li>+@TYPE' "adds Object Type to Zone Image Name</li> <li>+@DATE+mm/dd/yyyy' adds date stamp to name - required trailing date format argument shows default, also '+' can be used</li> <li><u>+@TIME+HH:MM:SS</u>' Adds Time Stamp Value to Name - required trailing time format argument shows default, also '+' can be used</li> <li>+@VALUE' Adds Object Text to Zone Image Name</li> </ul> <p><b>Note:</b> +#name' appends value of child name to imagename</p>
Parameters	File name to save as zone image.
Level	Field or Page level
Returns	<i>False</i> if an image cannot be saved. Otherwise, <i>True</i> .
Example	<p>Rulemanager Example: (Called on 'DETAILS' field)</p> <p style="padding-left: 40px;">ZoneImage_SaveAs("SAMMY"+@TYPE+@DATE+JJJ)</p> <p>Creates this file: "..\SAMMYDETAILS243.tif"</p>

## ZoneLEFT\_ImageLeft

Actions Library	Zones.rra
Description	<p>Uses the left boundary of the current field's <b>image</b> to define the left edge of the field's <b>zone</b>.</p> <p>This can extend or contract the zone's width depending on the placement of the image's left side</p>
Parameters	None
Level	Field level
Returns	Always <i>True</i> .
Example	<p>ZoneLEFT_ImageLeft()</p> <p><b>Be careful.</b> Images are often larger than a field's zone, and almost always larger than the field's word.</p>
See also...	ZoneRIGHT_ImageRight

## ZoneLEFT\_LeftBound

Actions Library	Zones.rra
Description	<p>Uses the left boundary of the current field's <b>word</b> to define the left edge of the field's <b>zone</b>.</p> <p>This extends the zone's width to accommodate the current word.</p>
Parameters	None
Level	Field level
Returns	Always <i>True</i> .
Example	<p>ZoneLEFT_LeftBound()</p> <p>If the field's <i>recognized</i> values occasionally reach beyond the zone's limits, use this action to expand the zone appropriately.</p>
See also...	ZoneLEFT_RightBound

## ZoneLEFT\_RightBound

Actions Library	Zones.rra
Description	Uses the <b>right</b> boundary of the current field's <b>word</b> to designate the <b>left</b> edge of the field's <b>zone</b> .  This extends the zone's width to accommodate the current word.
Parameters	None
Level	Field level
Returns	Always <i>True</i> .
Example	ZoneLEFT_RightBound()  This unusual action cleans the zone by ensuring that it does not contain a value.
See also...	ZoneLEFT_LeftBound

## ZoneRIGHT\_ImageRight

Actions Library	Zones.rra
Description	Uses the right boundary of the current field's <b>image</b> to define the right edge of the field's <b>zone</b> .  This can extend or contract the zone's width depending on the placement of the image's right side
Parameters	None
Level	Field level
Returns	Always True.
Example	ZoneRight_ImageRight()  <b>Be careful.</b> Images are often larger than a field's zone, and almost always larger than the field's word.
See also...	ZoneLeft_ImageLeft

## ZoneRIGHT\_LeftBound

Actions Library	Zones.rra
Description	Uses the <i>left</i> boundary of the current field's <b>word</b> to designate the <i>right</i> edge of the field's <b>zone</b> .  <i>Important!</i> This action also excludes the word from the zone.
Parameters	None
Level	Field level
Returns	Always <i>True</i> .
Example	ZoneRIGHT_LeftBound()  This action can clean the zone by eliminating the word it contains.
See also...	ZoneRIGHT_LeftBound

## ZoneRIGHT\_RightBound

Actions Library	Zones.rra
Description	Uses the <i>right</i> boundary of the current field's <b>word</b> to designate the <i>right</i> edge of the field's <b>zone</b> .
Parameters	None
Level	Field level
Returns	Always <i>True</i> .
Example	ZoneRIGHT_RightBound()  This is a convenient action if you find that a certain percentage of the field's entered values extend beyond to the zone's right edge.
See also...	ZoneRIGHT_LeftBound

### ZoneTOP\_ImageTop

Actions Library	Zones.rra
Description	Uses the <i>upper</i> boundary of the current field's <b>image</b> to designate the <i>top</i> edge of the field's <b>zone</b> .  This can extend or contract the zone's width depending on the placement of the image's left side. .
Parameters	None
Level	Field level
Returns	Always <i>True</i> .
Example	ZoneTOP_ImageTop()
See also...	ZoneBOTTOM_ImageBottom

### ZoneTOP\_LowerBound

Actions Library	Zones.rra
Description	Uses the upper boundary of the current field's <b>word</b> – its <i>recognized</i> value - to specify the <i>bottom</i> of the field's <b>zone</b> .  <i>Important!</i> This action can expand or contract the zone's height, depending on the placement of the word's lower boundary within the current page.
Parameters	None
Level	Field level
Returns	Always <i>True</i> .
Example	ZoneTOP_LowerBound()  If you find that an unacceptably high percentage of values for a particular field are entered a line too high, consider the use of this action in a follow-up <b>Locate</b> rule to search for the misplaced value.
See also...	ZoneTOP_UpperBound

## ZoneTOP\_UpperBound

Actions Library	Zones.rra
Description	<p>Uses the upper boundary of the current field's <b>word</b> to indicate the top of the field's <b>zone</b>.</p> <p>This can result in a taller or shorter zone, depending on the location of the word's upper boundary.</p>
Parameters	None
Level	Field level
Returns	Always <i>True</i> .
Example	<p>ZoneTOP_LowerBound()</p> <p>This action is helpful when experience shows that a certain number of values in this field will be measurably shorter than the zone's height. A follow-up <b>Locate</b> rule with this action will find those values..</p>
See also...	ZoneTOP_LowerBound

