

# docWizz

## *ReferenceBook*



# Contents

<b>1</b>	<b>Introduction.....</b>	<b>16</b>
<b>2</b>	<b>dW Requirements .....</b>	<b>17</b>
<b>3</b>	<b>dW System Environment.....</b>	<b>18</b>
3.1	docWizz Specifications.....	19
3.1.1	Document types.....	19
3.1.1.1	Document Structures and Configuration in docWizz.....	19
3.1.2	Image File Formats.....	19
3.1.3	East Asian or Arabic Languages Support .....	20
3.1.4	Limitations.....	20
3.1.5	Image size calculation .....	21
3.1.6	Performance .....	22
3.1.7	Interactive Clients .....	23
3.1.7.1	Monitors.....	23
3.1.8	Processing Scenarios.....	24
3.2	docWizz environment .....	25
3.3	Access Requirements.....	25
3.4	MS SQL .....	26
3.5	Log and Pool Database on SQL.....	27
3.5.1	Switch Databases for Log files to SQL Databases.....	27
3.6	dWClient (Workstation).....	29
<b>4</b>	<b>dWSrv .....</b>	<b>30</b>
4.1	docWizz Environment.....	30
4.2	Access requirements .....	32
4.3	Installation dWServer .....	32
4.4	dWfileServer .....	33
4.5	customershortname-dwsrv.ini.....	34
<b>5</b>	<b>docWizz share.....</b>	<b>37</b>
5.1	Document folders .....	37
5.1.1	IMAGELINK1 .....	37
5.1.2	RESCAN.....	37
5.1.3	SCANNER_CALIBRATION.....	37
5.1.4	BACKUP .....	38
5.1.5	RESTORE .....	38
5.1.6	EXPORT .....	38
5.2	System environment folder.....	39
5.2.1	CONFIG.....	39
5.2.1.1	BasicCfg .....	40
5.2.1.2	MiscCfg.....	40

5.2.1.3	PVSCFG .....	41
5.2.1.4	QA.....	41
5.2.2	MAINTENANCE .....	41
5.2.3	INSTALL .....	42
5.2.4	WORK.....	42
5.2.4.1	ClientConfig .....	43
5.2.4.2	Extratasks .....	43
5.2.4.3	FTPClient.....	43
5.2.4.4	OCRSERVICE .....	43
5.2.4.5	Lock .....	43
5.2.4.6	Service Status.....	44
5.2.5	DOCUMENTATION.....	44
5.2.6	RemoteQAMngr / RemoteQALoader .....	44
5.3	Processing configuration folders .....	45
5.3.1	SCRIPT.....	45
5.3.1.1	dwx .....	48
5.3.1.2	epub.....	48
5.3.1.3	export.....	49
5.3.1.4	Export_to_dw_doc.....	49
5.3.1.5	ImageProcessing.....	50
5.3.1.6	lib .....	50
5.3.1.7	MLHelpers .....	51
5.3.1.8	reject .....	51
5.3.1.9	remoteQAScripts .....	52
5.3.1.10	Samples.....	52
5.3.1.11	Setup .....	52
5.3.1.12	task .....	52
5.3.2	SCHEMA .....	53
5.3.3	PROJECT-CFG .....	55
5.3.3.1	Default .....	55
5.3.3.2	Clipping.....	56
5.3.3.3	rdy.....	56
5.3.3.4	dsl .....	56
5.3.3.5	epub.....	57
5.3.3.6	metadata.....	57
5.3.3.7	PDF.....	57
5.3.3.8	PDFImportCfg.....	57
5.3.3.9	reject .....	59
5.3.4	LISTS.....	60
5.3.5	SCANNER-INFO .....	62
5.3.6	SCANNER_CALIBRATION.....	63

5.3.7	TRAINING.....	63
5.3.8	MetadataDB.....	63
5.3.9	DICT .....	63
<b>6</b>	<b>IN.....</b>	<b>64</b>
<b>7</b>	<b>OUT.....</b>	<b>65</b>
<b>8</b>	<b>POOL .....</b>	<b>66</b>
<b>9</b>	<b>Help Menu.....</b>	<b>67</b>
<b>10</b>	<b>Configuration of projects .....</b>	<b>69</b>
10.1	Workflow overview.....	69
10.2	Add a new configuration .....	70
10.3	docWizz RDY files .....	70
10.4	Defining RDY files .....	72
10.4.1	Sample RDY file .....	72
10.4.2	DESCRIPTION element .....	73
10.4.3	PROCESSING element.....	74
10.4.3.1	Attributes.....	75
10.4.3.2	STOPAT .....	77
10.4.3.3	ALLOWSPPLIT .....	77
10.4.3.4	STITCHING .....	77
10.4.3.5	IMAGE_PREPROCESSING.....	77
10.4.3.6	IMAGE_PREPROCESSING: Despeckle.....	78
10.4.3.7	IMAGE_PREPROCESSING: Deskew.....	79
10.4.3.8	IMAGE_PREPROCESSING: Printspace .....	79
10.4.3.9	IMAGE_ENHANCE.....	80
10.4.3.10	OCR.....	80
10.4.3.11	STRUCTURE.....	82
10.4.3.12	PAGE_LINKING .....	82
10.4.3.13	METADATA .....	82
10.4.3.14	SKIP_LA .....	82
10.4.3.15	ZOOM_LEVEL.....	82
10.4.3.16	CLIPPING .....	82
10.4.4	OCR_CONFIG_PATH .....	83
10.4.5	LA_CONFIG_PATH.....	83
10.4.6	EXPORT element.....	83
10.5	dWSrv configuration per project .....	84
10.6	Implement substitution tables (long S) .....	85
10.7	Disable/Enable functions .....	87
10.7.1	EDITING .....	88
10.7.2	EDITINGTASK.....	88
10.7.3	PROCESSING .....	89
10.7.4	POOL.....	90

10.7.5	ITEMTRACKING .....	90
10.7.6	DOUBLEKEYING .....	90
10.7.7	LIST .....	90
10.8	Define MODS or DC .....	91
10.9	Project cfg to overwrite Default cfg.....	92
10.10	Stitching .....	93
10.11	PDF / OCR comparison .....	95
10.12	Keep selection in list view.....	97
10.13	Black & White algorithm usage.....	98
10.14	Enable/Disable certain cleaning algorithms .....	100
10.15	Edit info-boxes for tasks .....	100
10.16	Expand tools automatically.....	101
10.17	Change colors of frames for Cropping step.....	101
10.18	Customize Index view.....	102
10.19	Set LA quality for very close zones .....	103
10.20	Divided Document Pool .....	104
10.21	Including IgnoreTypes.dsl to a project.....	107
<b>11</b>	<b>Special Tasks.....</b>	<b>109</b>
11.1	Scope and details of Tasks .....	109
11.2	ExtraTasks.....	111
11.3	BackupConfig Task .....	112
11.4	Task: ExtrataskWithOCR.....	112
11.5	Reduce-, Restore Pool data .....	113
11.6	ComputeMissingOCR.....	115
<b>12</b>	<b>Update procedures .....</b>	<b>116</b>
<b>13</b>	<b>Auto-Update .....</b>	<b>117</b>
13.1	Folder structure .....	117
13.2	Routine calling customized shell scripts.....	119
13.3	Main files to check and start the Update process .....	119
13.4	Prevent computer from updating .....	120
13.5	Example how to prepare an execute a regular update .....	120
13.6	DWOCRREMOTE auto-update functionality.....	121
13.7	Background Information .....	121
<b>14</b>	<b>Manual updates .....</b>	<b>122</b>
14.1	Removing Files.....	122
14.2	Local Binaries .....	122
14.3	Updating services (dWSrv + dWSystemServices) .....	123
14.4	OCR.....	123
14.5	GUI .....	124
14.6	Post Update scripts .....	124
14.7	Notes .....	125

<b>15</b>	<b>Notes for Windows 7 or Server 2008 systems .....</b>	<b>127</b>
<b>16</b>	<b>PDFcfg.xml.....</b>	<b>128</b>
16.1	Configuration file for PDF creation (PDFcfg_xml) .....	128
16.2	Configuration setting description .....	129
16.2.1	PDF Version .....	129
16.2.2	Hidden Text .....	129
16.2.3	PartialImages.....	129
16.2.4	GenerateBookmarks.....	129
16.2.5	SinglePage .....	130
16.2.6	PageMode .....	131
16.2.7	PageLayout.....	131
16.2.8	OpenAction .....	131
16.2.9	PDFBookmarkLanguage .....	131
16.2.10	PDFBookmarkCoords.....	132
16.2.11	LinearizedPDF .....	132
16.2.12	UserAccessPermissions .....	133
16.2.13	ImageResolution.....	133
16.2.14	PartialImageResolution .....	133
16.2.15	ImageQualityJPG .....	133
16.2.16	ImageQualityJP2K.....	134
16.2.17	UseBWPages .....	134
16.2.18	AddPageThumbnail .....	134
16.2.19	AddPageLabel .....	134
16.2.20	FontType.....	135
16.2.21	TextRenderingMode .....	135
16.2.22	TextObjectsAsAnsi .....	135
16.2.23	UseCustomizedSettings .....	136
16.2.24	TaggedPDF .....	136
16.2.25	PDFAType .....	136
16.2.26	PDFFontPath .....	136
16.2.27	EmbedFont .....	137
16.2.28	FontMinimize .....	137
16.2.29	FontTextShrinkPercent.....	137
16.2.30	BiDirectionalText .....	137
16.2.31	EncapsulateTextSelection .....	138
16.2.32	PDFBookmarkTypes .....	138
16.2.33	PDFGroupBookmarks generate .....	139
16.2.34	Additional Bookmarks.....	139
16.2.35	Import PDF files as image .....	139
16.2.36	PDF Metadata.....	140
16.2.36.1	Detailed Explanation of Date .....	140

16.2.36.2	Single Page PDFs .....	141
16.2.36.3	Multi Page PDFs.....	142
16.3	Sample PDF configuration file .....	144
16.4	Sample File metadata.xml .....	148
16.5	Sample File XMP Metadata .....	149
<b>17</b>	<b>EPUB Configuration .....</b>	<b>150</b>
17.1	EPUB Generating export .....	150
17.2	EPUB configuration file.....	151
17.3	Generating EPUB in postProcess (METS, ALTO and page images) .....	151
17.4	EPUB document specific correction rules .....	152
17.5	EPUB on various devices .....	153
<b>18</b>	<b>Document definition .....</b>	<b>156</b>
18.1	Document structure definition.....	156
18.2	Create custom classes file.....	158
18.3	Create specific page classes (feature codes) .....	158
18.4	Create specific zone types .....	159
18.5	Change sequence of zone types .....	160
18.6	Create specific structure or hierarchy types .....	160
18.7	Defining specific colors for zone types .....	161
18.8	Sort zone types list .....	162
<b>19</b>	<b>OCR .....</b>	<b>165</b>
19.1	OCR configuration .....	165
19.2	OCR settings .....	168
19.2.1	Auto replacement not hard coded anymore .....	168
19.2.2	project-cfg\Default\Default-OCRCfg.xml .....	169
19.2.2.1	Font type and language settings .....	169
19.2.2.2	OCR related to the characters.....	169
19.2.2.3	OCR related to the ABBYY detection using patterns .....	170
19.2.2.4	OCR for specific zone type .....	171
19.2.2.5	Image processing settings - Job settings .....	172
19.2.2.6	Enhanced image settings .....	172
19.2.2.7	Post OCR processing .....	174
19.2.2.8	OCR settings .....	175
19.2.2.9	Zone Type.....	175
19.2.3	Project based / customized dictionary .....	176
19.3	OCR confidence computing in docWizz .....	177
19.4	OCR accuracy statistics .....	178
19.4.1	Choose Random Samples.....	178
19.4.2	Attain Number of Errors.....	178
19.4.3	Evaluate Statistics .....	179
19.5	OCR character counting .....	180

19.6	OCR Languages with different reading rules .....	180
19.7	Abbyy Finreader 11/12 .....	182
19.8	Abbyy Finereader license backup and recovery .....	183
19.8.1	Preparing License Backup.....	183
19.8.2	Recovering from your License Backup.....	183
19.8.3	Recovering without earlier created License Backups.....	183
19.9	Abbyy Finereader and dual core licenses .....	185
19.10	Abbyy license server .....	185
19.11	Abbyy Fraktur (Gothic) OCR licenses .....	185
19.12	Tesseract OCR .....	186
19.13	OmniPage 20.....	187
19.14	OmniPage 21 .....	188
19.15	Google Vision .....	189
19.16	Transkribus .....	191
19.17	Project based auto replacements .....	193
19.18	Compare two texts and report differences (using Levenshtein distance) .....	194
19.19	Hunspell dictionaries .....	195
<b>20</b>	<b>Export configuration .....</b>	<b>201</b>
20.1	TCL export configuration file.....	201
20.2	Vendor Information .....	201
<b>21</b>	<b>XML export configuration file .....</b>	<b>202</b>
21.1	General configuration .....	202
21.2	Path and name configuration.....	202
21.3	Image creation configuration .....	203
21.3.1	Partial images .....	203
21.3.2	Image creation script .....	204
21.3.3	File format identifier .....	205
21.3.4	JP2 settings in export.xml and export.tcl .....	205
21.4	TCL procedure calls on export .....	206
21.5	XML Output File Configuration .....	210
<b>22</b>	<b>Automated QA: Reject Conditions .....</b>	<b>211</b>
22.1	Explanation of rejects .....	211
22.2	Enable / disable rejects .....	212
22.3	Reject types .....	212
22.4	Reject status .....	213
22.5	Reject conditions .....	214
22.6	Configurations .....	216
22.7	Setting and activating Reject conditions.....	218
22.8	How to deal with Reject conditions (example).....	219
22.9	Defining new Reject conditions .....	220
22.10	Seperate rejects by Monograph or Newspaper.....	221



22.11	User interface .....	222
22.12	List of files and folders affected by the ReleaseByEnt feature .....	224
22.13	Example: Merge to Advertisement .....	224
<b>23</b>	<b>Default rejects .....</b>	<b>225</b>
23.1	CheckArticleHeading .....	225
23.2	CheckBlockRectangle .....	225
23.3	CheckDocLanguage .....	226
23.4	CheckDocPageCount .....	226
23.5	CheckDocPrintSpace .....	227
23.6	CheckEditionsTitle .....	227
23.7	CheckEmptyIssue .....	228
23.8	CheckForDifferentResolutions.....	228
23.9	CheckForDirtyOCR.....	229
23.10	CheckForDuplicateBlocks .....	229
23.11	CheckForInvalidPages .....	230
23.12	CheckForMissingOCROnPage.....	230
23.13	CheckForMultipageImage .....	231
23.14	CheckForNotAvailable .....	231
23.15	CheckForOCRTextvsDict .....	232
23.16	CheckForOriginImage .....	232
23.17	CheckForRepeatedSpecialPages .....	233
23.18	CheckForResetedOCROnManager .....	233
23.19	CheckForSpecialPages .....	234
23.20	CheckForUncorrectedOCR .....	234
23.21	CheckForWordCoord.....	235
23.22	CheckFrameSizes .....	235
23.23	CheckHyphenatedWord .....	236
23.24	CheckIllustrationLikeStructs .....	236
23.25	CheckImageMemorySize .....	237
23.26	CheckImagePrintSpace .....	237
23.27	CheckImageResolution .....	238
23.28	CheckImageSize .....	238
23.29	CheckIssueContainsPages .....	239
23.30	CheckIssueMetadata .....	239
23.31	CheckLangAndLanguageTerm.....	240
23.32	CheckMissingFrames .....	240
23.33	CheckMoreThanOneFrameReScan .....	241
23.34	CheckOCRConfidence .....	241
23.35	CheckPageNumber .....	242
23.36	CheckpageNumberToBeReseted.....	242
23.37	CheckPNErrors.....	243

23.38	CheckSetupImportPath .....	243
23.39	CheckStructureErrorsH .....	243
23.40	CheckStructureErrorsP .....	244
23.41	CheckSupplementPagesSizes .....	244
23.42	CheckXMLandMetadata .....	245
23.43	CheckXMLTextRectangle .....	245
23.44	CheckZonesOnlyOnPage .....	246
23.45	Doc_EvenPages .....	246
23.46	Enable / disable rejects .....	247
23.47	Explanation of rejects .....	247
23.48	InvalidBlockType .....	247
23.49	Issue_ContinuousPages.....	248
23.50	Issue_EvenPages.....	248
23.51	Reject status .....	249
23.52	Reject types .....	249
23.53	RejectMoreIssuesOnSamePage .....	250
23.54	Task_CheckEmptyReTyping.....	250
23.55	Task_CheckEmptyTyping.....	251
23.56	TestPrintSpace .....	251
23.57	TestRedFrameOnPage .....	252
23.58	ValidateDocument .....	252
23.59	ValidateMetadata.....	252
<b>24</b>	<b>RejectByEnt.xml .....</b>	<b>253</b>
24.1	Critical Condition .....	258
24.2	Timing Investigation.....	258
24.3	SkipInteractive .....	259
24.4	Reject per Editing Task .....	260
<b>25</b>	<b>Manual QA: The QA Work Step.....</b>	<b>261</b>
25.1	Working with the QA Mode.....	261
25.2	Setting up the QA Mode .....	261
25.3	Report result of Double Keying .....	262
25.4	Document Pool .....	263
<b>26</b>	<b>Editing/QA Tasks .....</b>	<b>265</b>
26.1	QA Tasks Configuration .....	267
26.1.1	Default-Setting.xml .....	268
26.1.2	Task Definition .....	268
26.1.3	Default-QA-Tasks.xml .....	268
26.1.4	User management .....	271
26.1.5	Task storage .....	271
26.1.6	Assigning tasks to documents .....	271
26.1.7	Workflow .....	271

26.1.8	Dynamic Tasks .....	272
26.1.9	Log .....	272
26.1.10	Script commands .....	272
26.1.11	XML structure .....	272
26.1.12	Rejects by editing tasks .....	273
26.2	Structure .....	274
26.3	ComputeMissingOCR .....	274
26.4	OCR .....	274
26.5	Double Keying .....	275
26.5.1	Double Keying configuration .....	276
26.6	Keying – First variant – compare different text input .....	277
26.7	Keying – Second variant – compare input text with detected text .....	279
26.8	Metadata .....	280
26.9	Clipping .....	280
26.10	FinalQA .....	281
<b>27</b>	<b>Remote QA .....</b>	<b>282</b>
27.1	Handling Remote QA documents overview .....	282
27.2	Remote QA environment .....	284
27.2.1	Overview .....	285
27.2.2	Collect and prepare data for Remote QA .....	286
27.2.3	Main system .....	287
27.2.3.1	IN, OUT and COMMANDS folders .....	287
27.2.3.2	Commands .....	288
27.2.3.3	RemoteQAManagerCfg.xml .....	289
27.2.3.4	FTP Client .....	294
27.2.3.5	Setting up FTP, FTPS and SFTP .....	299
27.2.4	Loader System .....	302
27.2.4.1	RemoteQALoaderCfg.xml .....	303
27.2.5	Script for filtering out finalQA Documents .....	304
27.2.6	Error on FTP upload (Known issue) .....	305
27.3	Maintenance Remote System .....	306
27.4	Disable/Enable functions .....	307
27.5	Storage and data transfer calculation .....	309
27.5.1	Typical File Sizes per Page .....	310
27.5.2	RQA images compression ratio .....	310
<b>28</b>	<b>Final QA Definition .....</b>	<b>312</b>
<b>29</b>	<b>dWOCRRemote .....</b>	<b>313</b>
29.1	Overview .....	313
29.2	Setup a configuration for using remote OCR .....	314
29.3	How to register services .....	315
29.4	Remote OCR configuration .....	316

29.5	DWOCRRUN.exe .....	319
29.6	DWOCRREMOTE auto-update functionality .....	319
29.7	Remote OCR for multiple environments .....	320
29.8	Stop, Kill or Shutdown commands .....	320
<b>30</b>	<b>Configuration Menu .....</b>	<b>321</b>
30.1	Change Login .....	321
30.2	System configuration .....	322
30.2.1	System Card .....	322
30.2.2	Pool Card .....	323
30.2.3	User Card .....	324
30.2.4	Document Types Card .....	325
30.2.5	Tasks Card .....	327
30.2.6	Lists Card .....	327
30.2.7	ODBC-Connection Card .....	329
30.2.8	Languages Card .....	329
30.2.9	Path Card .....	330
30.2.10	Dialog Boxes Card .....	331
30.2.10.1	New button .....	331
30.2.10.2	Delete button .....	331
30.2.10.3	Edit button .....	331
30.2.10.4	Properties of the Dialog Boxes .....	337
30.2.11	Zone Classes Card .....	340
30.2.12	Dictionaries Card .....	340
30.2.13	Script Card .....	341
30.3	Maintenance .....	342
30.3.1	Support and Error log .....	342
30.3.2	Volume Report .....	343
30.3.3	Backups to Restore .....	344
30.3.4	Recover Documents .....	345
30.3.5	Auto-save .....	346
30.4	Select scanner .....	347
30.5	Attach MS SQL-Server Express .....	348
30.6	Design mode .....	349
30.7	Script Editor .....	353
30.8	Script debugger .....	356
30.9	Encrypted passwords can be used .....	358
30.10	SuspiciousBlock .....	358
30.11	Black & White conversion configuration .....	358
30.12	Grid configuration .....	359
30.13	Re-Scan as well from Scan job .....	360
30.14	Local processing folder configuration .....	361

30.15	Set up a Test Configuration.....	362
<b>31</b>	<b>Add or edit new "NonSortWords" .....</b>	<b>363</b>
<b>32</b>	<b>docWizz ControlCenter .....</b>	<b>365</b>
32.1	License key.....	365
32.2	Services status .....	366
32.2.1	docWizz service levels .....	366
32.2.2	Monitor progress.....	368
32.2.3	Monitor progress for RemoteQA/Remote OCR.....	369
32.3	Pool management .....	370
32.3.1	Pool overview .....	370
32.3.2	Project, Job and Status .....	371
32.3.3	Priorities.....	371
32.3.4	Document pool.....	373
32.3.4.1	Reduce-, Restore Pool data .....	374
32.3.4.2	Properties - History .....	377
32.3.4.3	Custom filters.....	377
32.4	Storage capacity.....	379
32.5	Environment control.....	380
32.5.1	Statistics .....	380
32.5.2	Health status.....	381
32.5.3	Error log.....	382
32.5.4	Environmental functions .....	382
32.5.5	Volume report .....	383
32.6	Import document.....	385
32.7	Configuration .....	386
32.7.1	Projects.....	386
32.7.2	Settings files .....	387
32.7.2.1	General .....	389
32.7.2.2	Import.....	390
32.7.2.3	Cropping .....	390
32.7.2.4	Zoning and Structure .....	392
32.7.2.5	OCR settings .....	393
32.7.2.6	Metadata .....	398
32.7.2.7	Output .....	398
32.7.3	Document definition .....	399
32.7.4	Processing.....	400
32.7.5	Custom control.....	401
32.7.6	Maintenance .....	405
32.7.6.1	Maintenance Mode and its Interaction with the docWizz Services .....	405
32.7.6.2	Memory Handling for dWCtrlDlg.....	407
32.7.6.3	Information for Updating Process .....	407

<b>33</b>	<b>Auto-Power-Off .....</b>	<b>408</b>
33.1	Shutting down .....	410
33.2	Service behavior .....	412
33.3	Wake-Up.....	412
33.4	Manual actions .....	412
33.5	Configuration .....	413
33.6	Long Idle .....	413
33.7	Load Balancing .....	413
<b>34</b>	<b>Maintenance.....</b>	<b>414</b>
34.1	Maintenance Client (MTN).....	415
34.2	How to use the MTNClient.....	416
34.2.1	Starting the MTNClient .....	416
34.2.2	Surface of the MTNClient .....	416
34.2.3	MTN Package Generation .....	417
34.2.4	Sending the data package to CCS .....	418
34.3	MTN Modules .....	420
34.4	MTN Module "common functionalities" .....	421
34.4.1	System Data .....	421
34.4.2	Configuration Data.....	421
34.4.3	Directory List.....	422
34.4.4	Screenshots.....	422
34.4.5	EventLog.....	423
34.4.6	Customer Attachments .....	423
34.4.7	Shortcuts.....	423
34.5	MTN Module docWizz .....	424
34.5.1	Backup documents .....	424
34.5.2	Exported documents.....	424
34.5.3	Pool documents.....	424
34.6	MTNClient - Results .....	425
34.6.1	Results of the MTN Module "common functionalities" .....	425
34.6.2	Result: System Data .....	425
34.6.3	Result: Directory List .....	426
34.6.4	Result: Screenshots, Eventlog, Customer attachments.....	426
34.6.5	Result: EventLog .....	427
34.7	Processing Statistics .....	429
<b>35</b>	<b>EDS error list.....</b>	<b>430</b>
35.1	General errors .....	430
35.2	File access errors .....	430
35.3	Directory errors.....	431
35.4	Property errors.....	431
35.5	Function parameter errors .....	431

35.6	Device errors .....	431
35.7	Stream errors.....	432
35.8	Communication errors .....	432
35.9	Camera UI lock/unlock errors .....	433
35.10	STI/WIA errors.....	433
35.11	Other general errors .....	433
35.12	PTP errors .....	433
35.13	TakePicture errors .....	434
35.14	ED-SDK Generic Error IDs .....	434
<b>36</b>	<b>Statistics .....</b>	<b>439</b>
36.1	Configuration of the Tree .....	442
36.2	Further functionalities .....	443
36.3	Behavior in case of errors.....	443
<b>37</b>	<b>Document processing Log (BatchResults table).....</b>	<b>444</b>
37.1	Active / Idle status .....	446
<b>38</b>	<b>Abbreviations.....</b>	<b>447</b>

# 1 Introduction

**docWizz**

Last updated: 01.03.2022



This manual is dedicated to the advanced user.

We describe here technical oriented information and configuration details. You need system administrator permissions to use most of the described functions. The structure of the program and the construction of the operational elements are the reasons why identical functions can be called up in different ways either via a menu, an icon button or a hotkey. Dialog boxes and certain elements of functionality can be configured to meet the specific needs of docWizz users.

The following description and diagrams refer to a sample configuration. It is not only possible, but also probable that your user interface will look different.

If you should find any inconsistencies in this manual, if you require further explanations, or you find that key questions are inadequately dealt with, we would be very grateful to hear your suggestions. Your suggestions are important to us for the improvement of our manuals.



## 2 dW Requirements

For details, see [docWizz\\_SystemRequirements.pdf](#)

### 3 dW System Environment

For an environment covering both mass production (1 million pages per year) and detailed conversion projects, we recommend an environment of

- 1 file server / network share
- 2 docWizz processing servers
- 3 docWizz QA workstations

The performance on average material is reflected in the following calculation:

#### 1 docWizz service performs 24/7

100 pages/h
2.400 pages/d
72.000 pages/m
864.000 pages/y

#### 1 docWizz workstation operator performs

(strongly depends on defined mark-up/level of structure and documents properties and operator experience)

200 pages/h	
1.200 pages/d	(assuming 6 operating hours per day)
20.400 pages/m	(assuming 17 work days per month)
240.000 pages/y	(assuming 200 work days per year)

## 3.1 docWizz Specifications

General specifications of the **docWizz** input and output as well as restrictions and limitations:

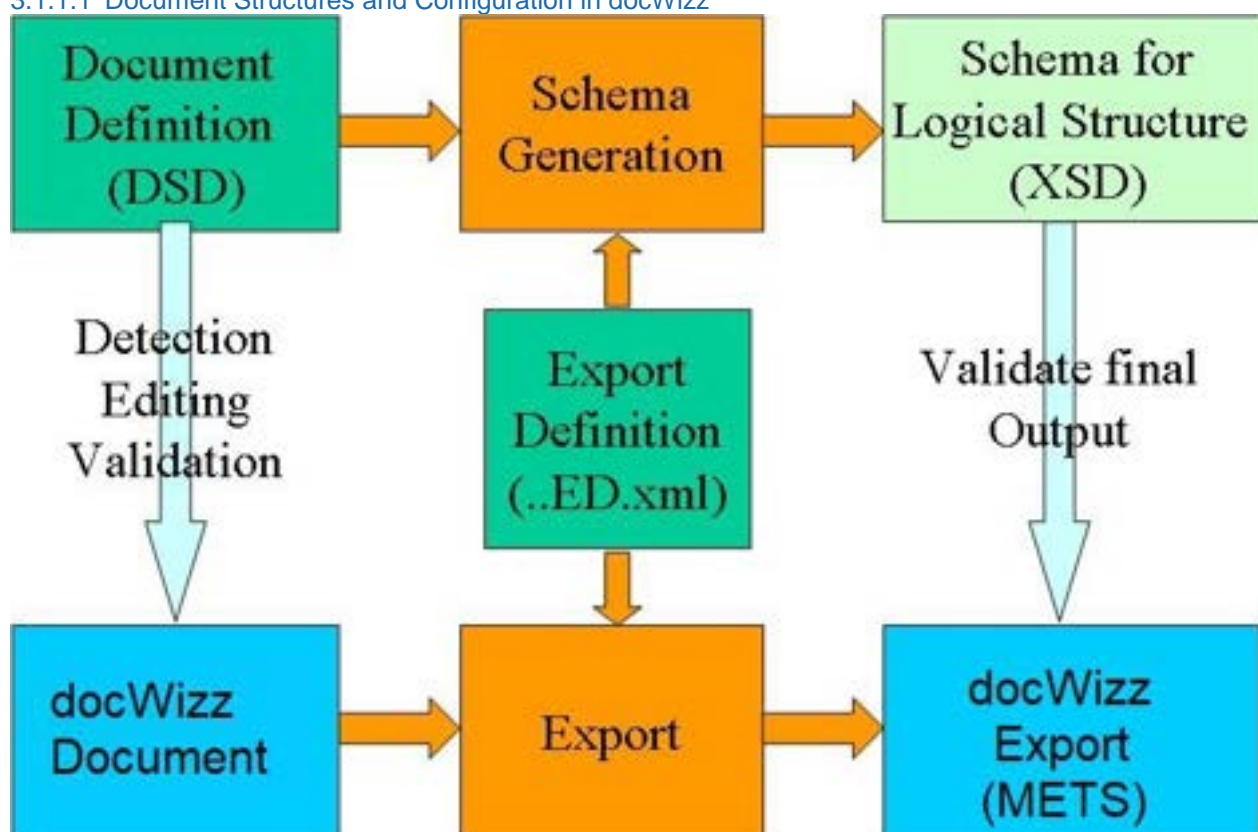
### 3.1.1 Document types

**docWizz** has a flexible system of different document types.

The current rule set and configuration provides support for:

- Newspaper
- Monograph
- Serial

#### 3.1.1.1 Document Structures and Configuration in docWizz



### 3.1.2 Image File Formats

**docWizz** is able to process the following type of images:

- bitonal
- gray scale 8 bit
- color 24 bit

Other formats will be converted by **docWizz** automatically to the nearest matching type.

The output of **docWizz** will always be one of those types.

**docWizz** can process the following file formats as input:

- tiff (uncompressed, group3, group4, packbits, lzw)

- jpg
- jp2000
- bmp
- gif
- pdf
- cr2

As output can be created:

- tiff
- jpg
- jp2000
- pdf

### **Dynamic JPEG 2000 compression**

Images can be compressed dynamically instead of firm compression. Thus we reach a substantially smaller data amount with continuously high quality.

### **Raw-Scan-Import**

Cameras of a Kirtas scanner deliver high level Raw-Scans. These are quite better than default used JPEG images. We have developed a special import for these Raw-Scans.

### **3.1.3 East Asian or Arabic Languages Support**

**docWizz** supports Arabic language.

A special license for Finereader and Omnipage is required.

### **3.1.4 Limitations**

Due to the fact that docWizz needs to keep the complete structure of a document in memory, the size of documents that can be processed within docWizz is restricted.

### **Page sizes**

Image resolution:	The resolution of images must be in range of 200 to 600 dpi. Higher resolutions are also possible, but using higher resolutions requires configuration changes (adapt reject conditions). Only symmetric resolution (x, y) is supported.
Metric image size:	The overall size of images is limited by A1 size.
Image data size:	The official limits for image size are defined for 600 MB memory size (uncompressed files size), internal tests are going up to 1 GB images. ( <b>dW</b> version 6.5 and higher)

### **Document size**

The capacity of docWizz is limited to approx. 10 million characters per document. This leads to a maximum size of 100 pages per newspaper and up to 2000 pages for a typical book. Exceeding the maximum number of characters might cause docWizz to run out of memory.

## Overall capacity

Number of pages: **dW** is designed to be able to handle up to 5 million pages in parallel.

Number of documents: **dW** can handle up to 100.000 documents in parallel.

In smaller production environments a MS Access database is sufficient whereas for mass digitization an active database like SQL Server is mandatory. There is no limit in storage capacity as long as performance is sufficient.

## Count image size

Size	Factor
A4	1
A3	2
A2	4
A1	8
A0	16

Color	Factor
B & W	1
Gray	8
24Bit color	24

Edition	Factor
300	1
400	2
600	4
1200	8

One must insert and multiply each of the 3 factors to receive the size in MB. That is valid for uncompressed, but that is the main one for docWizz processing. LZW, JP2 etc. is only interesting for the disk space.

### 3.1.5 Image size calculation

To provide a figure on which formats / image formats can be processed now by docWizz, here is a calculation for uncompressed images sizes, according to "DIN A" formats. docWizz can process images with the maximum size of 600 MB (marked in green color).

DIN format	A4	A3	A2	A1	A0
height x width (cm)	29,8 x 21	42 x 29,8	59,6 x 42	84 x 59,6	119,2 x 82
height x	11,73	16,54	23,46	33,07	46,93

		width (inch)	x 8,27	x 11,73	x 16,54	x 23,46	x 33,07
<b>resolution 300 dpi</b>	<b>color depth</b>		ca. 8525,32 MP	ca. 17050,64 MP	ca. 34101,28 MP	ca. 68202,55 MP	ca. 136405,12 MP
<b>b/w</b>	2		2,08 Mb	4,16 Mb	8,33 Mb	16,65 Mb	33,30 Mb
<b>gray</b>	8		8,33 Mb	16,65 Mb	33,30 Mb	66,60 Mb	133,21 Mb
<b>color</b>	24		24,98 Mb	49,95 Mb	99,91 Mb	199,81 Mb	399,62 Mb
<b>color (BMP) *</b>	32		33,30 Mb	66,60 Mb	133,21 Mb	266,42 Mb	532,83 Mb

<b>resolution 400 dpi</b>	<b>color depth</b>		ca. 15156,13 MP	ca. 30312,25 MP	ca. 60624,45 MP	ca. 121248,99 MP	ca. 242497,98 MP
<b>b/w</b>	2		3,70 Mb	7,40 Mb	14,80 Mb	29,60 Mb	59,20 Mb
<b>gray</b>	8		14,80 Mb	29,60 Mb	59,20 Mb	118,41 Mb	236,81 Mb
<b>color</b>	24		44,40 Mb	88,81 Mb	177,61 Mb	355,22 Mb	710,44 Mb
<b>color (BMP) *</b>	32		59,20 Mb	118,41 Mb	236,81 Mb	473,63 Mb	947,26 Mb

<b>resolution 600 dpi</b>	<b>color depth</b>	ca.	34101,28 MP	68202,56 MP	136405,12 MP	272810,23 MP	ca. 545620,47 MP
<b>b/w</b>	2		8,33 Mb	16,65 Mb	33,30 Mb	66,60 Mb	133,21 Mb
<b>gray</b>	8		33,30 Mb	66,60 Mb	133,21 Mb	266,42 Mb	532,83 Mb
<b>color</b>	24		99,91 Mb	199,81 Mb	399,62 Mb	799,25 Mb	1598,50 Mb
<b>color (BMP) *</b>	32		133,21 Mb	266,42 Mb	532,83 Mb	1065,66 Mb	2131,33 Mb

\*) BMP has 8 bit reserved per pixel. This is used by MS operating system - and due to that - is also needed in some cases in docWizz (e.g. GUI)

### 3.1.6 Performance

docWizz is designed as a system for mass production. Therefore performance is an important issue.

Although performance is highly dependent on the type of material and its quality, there are some general statements:

- Whenever the user takes an action in browsing or correction, the result should be displayed immediately. In normal cases the waiting time must not exceed one second.
- Any time consuming operation should be postponed and moved to automated processing or background processing.
- Parallel network traffic should be reduced to a minimum.
- For fully automated processing of documents, CPU power is the major limitation of throughput. Therefore, docWizz code is optimized in speed and custom scripts needs also to be optimized in speed. As well new hardware technologies (Dual Core, Quad Core) are supported.
- Memory management: Automatically free image memory will be executed whenever "Memory Depleted" has been detected.

- Memory Management: Before a PostExport script runs, memory allocated for export is cleaned. This enlarges memory available for additional processing like PDF creation.

### 3.1.7 Interactive Clients

Human resources are the most expensive parts during mass digitization. Any kind of human interaction should be reduced of a minimum.

We differentiate between two issues:

- The amount of necessary human interactions should be reduced to a minimum.
- Whenever human interaction is required, the system should respond instantly and any kind of time consuming processing

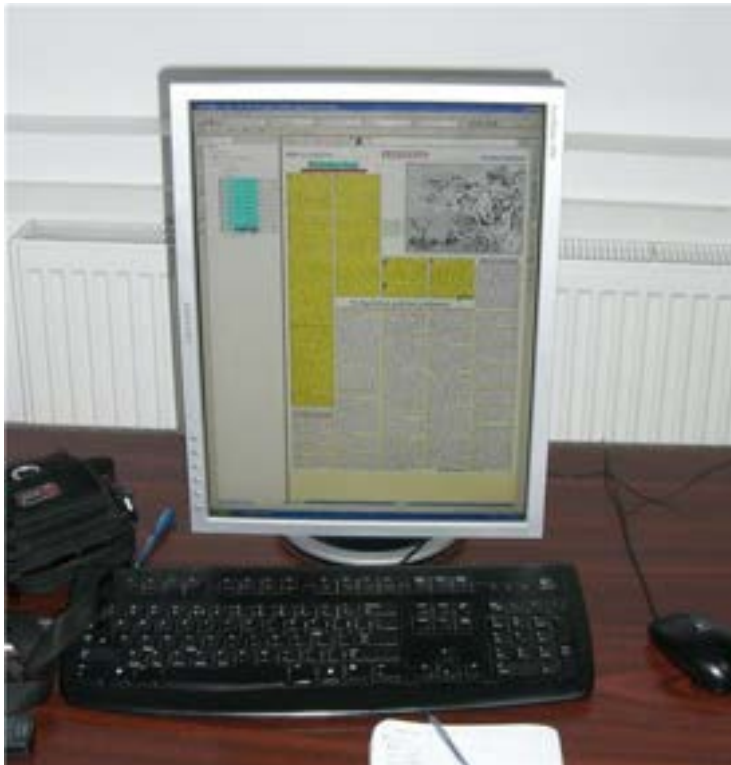
#### 3.1.7.1 Monitors

Screen resolution:

- minimum: 1280 x 720
- recommended: 1920 x 1080 or UHD (4K)

As user interface language, only English is available; other languages are possible, but not tested.

docWizz does not support multiple monitors. For correction of newspapers, it is recommended to use monitors that can be rotated 90 degrees (Pivot display).



### 3.1.8 Processing Scenarios

In general there are two different processing scenarios covered by docWizz:

- fully automated processing with no or limited human interaction
- semi-automated processing with full verification and correction by humans

#### **Fully automated Processing**

In fully automated processing, by default no human interaction is required. Conversion is done automatically and the final output is created. In addition, accuracy of detection is determined. Reject conditions may be defined to break processing of documents in case of unexpected input or results or in case of insufficient accuracy.

The quality of output is controlled just by formal checking. Accuracy is controlled by automated measurement of confidence.

#### **Semi automated Processing**

In semi-automated processing, automated conversion is always verified and corrected by users. A dedicated policy guarantees a consequent high quality and accuracy of the generated output.



## 3.2 docWizz environment

The docWizz environment is based on the server / client architecture.

The dWfileServer is the central location where all configuration data, schema data, script data, and databases will be stored. This ensures that any update of these files has only been done on this machine, and any workstation uses the same configuration and databases.

Each dWServer and dWClient must be able to connect to this shared directory.

In case of only 1 dWService, the dWfileServer can be merged with the dWServer.

In case of further dWServer it is recommended to separate the shared data from the executed dWServices for best performance results.

## 3.3 Access Requirements

To enable docWizz processing server to access the files and databases on the file server, a domain user is needed which is able to execute a local service on the docWizz processing server (local policy) and also has network permissions to connect to the docWizz share on the dWfileServer (read/write).

The docWizz service will be configured to log on using this user. It is recommended to add a domain user having these access rights.

**Note:** docWizz will not work in a Windows workgroup.

If no password is configured for the admin user, the system always logged in as admin, if previous login was as administrator. The user will always be queried in this situation.

## 3.4 MS SQL

### Database server requirements

CPU	Minimum: Intel® Xeon® E3 (or similar) Recommended: Intel® Xeon® E5 (or similar)
RAM	12 GB RAM recommended for combined server (8 GB for dW processing + 4 GB for domain services and database services) 500 MB RAM required for large dW databases.
NIC	Minimum: 1 Gbit/s
Screen	No local screens are required on these machines (except for administrative work).
OS *	Versions: Windows Server Minimum: 2012 Recommended: 2016  Processor architecture: 64 bit Version type: Standard or higher Language: English
Domain	SQL Server must be member of the Microsoft domain
SQL server	MS SQL server version: Minimum 2012 Recommended 2014 / 2016  Version type: For dW edition Basic / Pro : Express Edition (for free) For dW edition Enterprise: Standard Edition or higher required (see comment above)
Firewall	MS Firewall needs to be disabled or opened for full ODBC and UNC connection access
dW environment	All machines / domain user need permanent access to this SQL server

Note: For installations and updates a completely updated operating system is required. All current Windows service packs have to be installed.

## 3.5 Log and Pool Database on SQL

By default setting docWizz log files and pool information are saved on Microsoft Access Databases.

For larger production environments (more than approx. 4 processing servers running) with multiple access on databases it is strongly recommended to switch to dynamic databases such as SQL.

In case of using SQL databases, the use of across two domains is possible, e.g. the docWizz environment is located in one domain and the SQL database is located in another domain.

Recovery models for LOG and POOL database has to be set to simple (in the properties). So all transactions were kept in database and memory.

### 3.5.1 Switch Databases for Log files to SQL Databases

On the SQL server the system administrator must create two empty databases, named dw\_pool and dw\_log. Any other setting and tables will be added by docWizz automatically. When finished close the database.

After setting up the databases some changes need to be performed in the docWizz configuration.

Firstly, close all docWizz applications and docWizz services on all processing servers and QA workstations. Then perform the following changes:

Edit the [DOCUMENT.POOL] and [ERRORLOGDB] section of the customername-docWizz-GLBL.ini to

```
[DOCUMENT.POOL]
POOLDIR=***POOL***
DEPTH=2
OLEDB=1
DISTRIBUTED=0
LOCALDIR=C:\NEWSWORK\DISTPOOL
EXPORTEDDRIVE=NWPOOL
STORETHUMBNAİL=0
THUMBNAİLSIZE=200:200
DATABASENAME=dw_pool
SERVERNAME=servername
DATABASETYPE=SQLSERVER
DBTYPE=2
COLORASJPEG=0
ZIPANALYSIS=1
SAVECURRENTDATE=1
USERNAME=
PASSWORD=
CONNECTIONSTRING=
```

If a dedicated user has been set to the pool database, add username and associated password in clear text. docWizz will encrypt the password at the first start.

```
[ERRORLOGDB]
```

```
DATABASENAME=dw_log
SERVERNAME= servername
DATABASETYPE=SQLSERVER
USERNAME=
PASSWORD=
CONNECTIONSTRING=
```

Training database (containing mainly keywords used in docWizz for various classification tasks) should be also set as an SQL database. Once configured as described below, restart docWizz and database will be populated with default values that can be later altered by specific tools.

```
[DEFAULTTRNDB]
DATABASENAME=TrainingMETaE
SERVERNAME=servername
DATABASETYPE=SQLSERVER
USERNAME=
PASSWORD=
CONNECTIONSTRING=
```

When finished you can restart all services and applications.

Before continuing with production: Open docWizz application on one of your machines, change login to admin, docWizz main menu:

Configuration --> change login  
and select from the main menu:

Configuration, Maintenance and click on Recover Documents.

Database entries for Pool will be recovered into the new databases.

If [ERRORLOGDB] is empty that means automatically ACCESS MDB in the directory.

### 3.6 dWClient (Workstation)

Workstations hold the program files for docWizz METAe and the OCR service, but no configuration data, databases or scripts. These are held on the dWfileServer.

(Only in case of a single workstation environment these data will be on the same machine.)

From any workstation, any document which has been processed to a certain stage can be loaded. On a workstation any interactive quality control or corrections can be done.

See [docWizz\\_Usermanual](#), chapter [Processing Documents](#), for details.

## 4 dWSrv

### **docWizz Server**

The docWizz Server (dWSrv) is a machine where docWizz runs as a Microsoft® Windows based service.

The dWSrv is developed for running automatic steps of the digitization processes of docWizz. Dependent on the configuration, the time-consuming and automatic jobs are taken by the Services and running until defined breakpoints or logical errors will be reached.

### 4.1 docWizz Environment

The dWfileServer/network share is the core location where all configuration data, schema data, script data, and databases will be stored. This ensures that any update of these files needs to be done on this machine only. Any workstation uses the same configuration and databases. Each dWServer and dWClient must be able to connect to this shared directory. Therefore it is necessary to use UNC names in all docWizz configuration files. Mappings and substitutions do not work!

#### **dWfileServer**

The dWfileServer is the core location where all configuration data, schema data, script data, and databases will be stored. This ensures that any update of these files needs to be done on this machine only. Any workstation uses the same configuration and databases. This is essential especially for the training database, which holds all string list entries. String list entries intend to optimize detection of certain structure elements (for example Preface could be indicated by a string 'Preface' [English] or 'Vorwort' [German]).

Also, this machine can hold all intermediate results (pool data), stored in the document pool, which could be stored on a different machine, too.

These files will be XML files as well as image files, if docWizz changed the source files for any reason (deskew, despeckle).

Note: docWizz will never modify Source files, but a copy will be created for changed images in the pool directory.

By default, this machine is also export destination. All export (XML and PDF) and backup files would be stored on this file server. It is also possible to change the export directory to another share in the network or to use several export locations for different projects.

Typically this machine contains all source images and other source data such as XML-files providing any metadata related to these files. However, these source files can be imported from another destination in the network.

Each dWServer and dWClient must be able to connect to this dWfileServer and especially its shared directory.

#### **dWService**

The docWizz/Service (dWSrv) is a Microsoft® Windows based service.

The dWSrv is developed for processing the automatic jobs of the conversion in docWizz, operating 24/7.

Anytime a document is set up (for example images loaded, basic parameter set) the job is queued to be picked up by docWizz service. As the service runs all the time, it will continuously process automated jobs in the workflow.

These automated jobs may be:

- Import of new images
- Detect Double Page Frames\*
- Split Double Page Frames\*
- Image processing and enhancement, like crop, align, skew and despeckle.
- Detect Layout Elements and Build Page Sequence (detect text, tables, illustrations, vertical text, formulas, page numbers, logical page sequence)
- Build Pages Hierarchy (Front/Main/Back, detect headlines, authors,...)
- Build Hierarchy (build Chapter Hierarchy, detect structure elements such as TitlePage, TableofContent, Preface, Lists,...)
- Export (validate vs. pre-defined grammar for monograph/serial, validate metadata (Dublin Core or MODS, export METS/ALTO, METS-text, PDF, Backup files\*\*)

*\* only if double pages are processed*

*\*\* backup files in order to re-launch documents into docWizz to change anything anytime afterwards*

## 4.2 Access requirements

To enable docWizz Service to access the files and databases on the file server, a domain user is needed which is able to execute a local service on the dWServers (local policy) and also has the network permissions to connect to the docWizz share on the dWfileServer (read/write). The docWizz Service will be configured to log on using this user.

It is recommended to add a domain user which guarantees the needed access rights.

Note: docWizz will not work in a Windows workgroup.

## 4.3 Installation dWServer

The docWizz server setup will install the selected binaries automatically. All files must be installed on local disk (mappings and substitutions do not work).

On the dWServer the following directories will be installed by the dWServer Setup Wizard:

The setup automatical register the OCR libraries and install the necessary OCR Dongle. Therefore follow the instructions of the setup wizard.

Next the docWizz Service needs to know where to find the configuration files. This is stored in the global.ini in the 'BIN' subdirectory of the '**DWService**'. The link in the global.ini is pointing to the **dWfileServer** with the shared docWizz directory. It is declared in the following way:

```
EXTINI=\\dWServer\docWizz-Share\config\docWizz
```

Further details to the path see **dWfileServer** chapter.

After this action, the service window will be opened automatically.

Select the docWizz Service ("dWSrv") and open the properties dialog. Now change the user, on which the service will run. The start-up type shall be set to 'Automatic'.

dWSrv properties: change service user and startup type



The service user need the rights to connect to the remote systems where the document pool, the export directory and the configuration files will be placed.

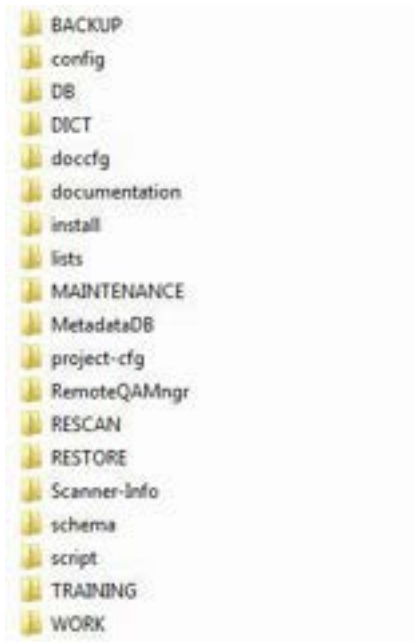
After this step, the docWizz Service is ready to be started.



## 4.4 dWfileServer

On the dWfileServer the following directories will be installed by the setup:

Directory structure of dWfileServer:



These directories will be used by docWizz instances, which are redirected by the global.ini.

So all docWizz instances (for dWSrv the user which is configured to use services) must have read/write-permission on the share, subfolders and files. By default these permissions are given to 'everybody'.

Each dWSrv is re-directed to the dWfileServer to read the global configuration (folders in screen shot above). The linking will be done by the file global.ini which is located on each dWSrv.

Use this pattern (only UNC paths are allowed) to connect the dWSrv to dWfileServer:

```
EXTINI=\\<dWfileServer>\<docWizz-Share>\config\docWizz
```

<dWfileServer> - Name of the physical dWfileServer machine, e.g. dWProcessing

<docWizz-Share> - shared docWizz directory on dWfileServer, e.g. dWShare

Example:

```
EXTINI=\\dWProcessing\dWShare\config\docWizz
```

All dWSrv's will be controlled by another docWizz service. This service will also be executed as a Windows service and is called docWizz Service Manager (**DWSrvManager**). It will be executed once on each machine. This service is a manager program and re-starts the dWSrv's in case of an unexpected error.

dWSrvManager monitors also the Start/Stop/Kill and Shutdown commands for each dWSrv.

## 4.5 customershortname-dwsrv.ini

This file is located in config directory on your dWfileServer. The customer short name was chosen during installation.

The file can be used to configure the look of the tree in docWizz ControlCenter (dW Control Center). It is necessary to adapt some settings after installation of docWizz to reach the best possible performance.

[NIGHTMODE]	basic configuration for nightmode
Groups=NMGroup01,NMGroup02,NMGroup03	the groups of computers that are to be shutdown
LongIdleDuration=100;seconds	awaits the computer to go into idle state for the specified time
CriticalApps=dwcl.exe,dwsrvmanager.exe	when one application that is defined here is running, nightmode shutdown is not going to shut down the computer
NightModeTimer=20000;milliseconds	the time that the computer waits to start processing a job or a task, if overpass the computer is shutdown
TimeToWaitForCriticalAppRestart=500;seconds	waiting for critical applications to start if crash or close by mistake

[NMGroup01]	this name comes from 'Groups' in section [NIGHTMODE]
Enabled=1	1 to enable nightmode shutdown and wakeon lan 0 to disable it
StartDaytime=06:00:00	time interval in which the nightmode shutdown is disabled (working time)
EndDaytime=19:00:00	time interval in which the nightmode shutdown is disabled (working time)
Computers=<Comp01>,<Comp02>,<CompXX>,...	Comp01,Comp02 – computers that are set for nightmode shutdown
WakeableComputers=<Comp01>,<Comp02>,<CompXX>,<CompXX>,...	wakeable computer list

[dWControlCenter]	configuration for dW ControlCenter GUI
TreeTimer=1000	Used to update the services icons in tree ( state : Idle, processing, stopped, etc.)
AllServicesValidDuration=50	update of the dW services status after that period (milliseconds)
MaxNoOfEntriesToAutomaticallyOpenTheGroup=8	open group automatically in tree on max. entries

[SERVICES]	configuration of instances per service group
<Comp01>=1a	First instance of dWSrv on Comp1 will be in group 1a

<Comp01>:DWSrv2=1b	Second instance of dWSrv on Comp1 will be in group 1b
<Comp01>:DWSrv3=1c	Third instance of dWSrv on Comp1 will be in group 1c
<Comp01>:DWSrv4=1d	Fourth instance of dWSrv on Comp1 will be in group 1d
<Comp02>:DWRemoteQAManager=RQA	dW remote QA manager service of Comp02 will be in group RQA
<Comp02>:DWFTPCClient-XXX=RQA	dW FTP Client service of Comp02 will be in group RQA
<Comp03>:DWOCRRemote=OCR	OCR remote service of Comp3 will be in group OCR
...	Add more if necessary

[DEFAULTGROUPS]	definition to which group a dW instance belongs if no group is defined in [SERVICES] for this instance
DWSRV=1a	First instance will belong to group 1a
DWSRV2=1b	Second instance will belong to group 1b
DWSRV3=3	Third instance will belong to group 3
DWSRV4=G	Fourth instance will belong to group G

[PROJECTSSELECTION]	Definition of project based groups
MyProjects01=Books,Newspapers,CustomerProject01	Naming of group can be chosen (no spaces). Add project name(s).
MyProjects01=CustomerProject02	Naming of group can be chosen (no spaces). Add project name(s).

[GROUPS]	Definition how to name the node in the tree of dW Control Center
1a=Blades-dWSrv	Choose a name for the group(s).
1b=Blades/NoOCR_Export	A meaningful naming is very helpful, especially when you use a lot of processing computers with 4 instances on each machine. Spaces and special characters are allowed.
1c=Blades/NoOCR	
1d=Blades/NoOCR_Import	
OCR=RemoteOCR	
RQA=RemoteQA	

[JOBSSELECTION]	Definition of groups with jobs
AllJobs=Import, DetectPageFrames, BatchRestoreVerifyPageFrames, ModifyPages, DetectLayoutElements, DetectPageNumbers, BuildPagesHierarchy, BuildHierarchy, Export, ExportXML	Define a name you'd like to use (e.g. AllJobs) and add the jobs should be done by this group. Separate the entries with comma.
ExportJob=Export, ExportXML	These jobs are possible:

NoOCR=Import, DetectPageFrames, ModifyPages	Import, DetectPageFrames, BatchRestore VerifyPageFrames, ModifyPages, DetectLayoutElements, DetectPageNumbers, BuildPagesHierarchy, BuildHierarchy, Export, ExportXML
---	--

[TASKSSELECTION]	Definition of groups with tasks
NOOCRTASKS=IMPORTIMAGES, WAKEUPCOMPUTERS, BACKUPCONFIGS, MNGRCOLLECTDATA, MNGRUPDATEDATA, EXTRATASKS, CLEANPOOL, RESTORE, FTPPRIO, RESTOREPOOL	Define a name you'd like to use (e.g. ALLTASKS) and add the tasks should be done by this group. Separate the entries with comma.
NOOCRTASKS_PLUS=IMPORTIMAGES, WAKEUPCOMPUTERS, MNGRCOLLECTDATA, MNGRUPDATEDATA, RESTORE, EXTRATASKS, BACKUPCONFIGS, UPDATEPOOLSTATUSMNGR, CLEANPOOL, RESTOREPOOL, LOGUNITMERGE	There are many pre-defined tasks which can be used. Also user defined tasks can be configured here.
RQATASKS=WAKEUPCOMPUTERS, MNGRCOLLECTDATA, MNGRUPDATEDATA, UPDATEPOOLSTATUSMNGR	Example: WAKEUPCOMPUTERS – executes the task to start machines by wake on lan (configured in [NIGHTMODE] section).
DELIVERYTASKS=WAKEUPCOMPUTERS, BACKUPCONFIGS	

[GROUP.1a]	Configuration what a group should do (jobs, tasks, include/exclude projects)
JOBSSSELECTION= ExportJob	Which jobs should be done by this group. Configuration was done in [JOBSSSELECTION] section.
TASKSSELECTION=NOOCRTASKS	Which tasks should be done by this group. Configuration was done in [TASKSSELECTION] section.
PROJECTSINCLUDE=JUSTBAMS	Include a project which should be done by this group. Configuration was done in [PROJECTSSELECTION] section. Please note that only one project name is allowed here. Comma separation and several projects listed will not work!
PROJECTSEXCLUDE=JUSTBAMS	Exclude a project which should not be done by this group. Configuration was done in [PROJECTSSELECTION] section. Please note that only one project name is allowed here. Comma separation and several projects listed will not work!

## 5 docWizz share

Where are all the files? What do the directories contain? How is docWizz file system organized?

A little overview on the most important directories of a docWizz default installation.

The folders of docWizz share are organized as:

- Document folders
- System environment folders
- Processing configuration folders

Headlines in capitals are docWizz path macros as defined in `***CONFIG***\sitename-docWizz-glbl.ini`

For better reading they are displayed without the asterisks (\*).

### 5.1 Document folders

The Document folders of docWizz share are organized as:

- IMAGELINK1
- RESCAN
- SCANNER\_CALIBRATION
- BACKUP
- RESTORE
- EXPORT

#### 5.1.1 IMAGELINK1

default: `DATA\dwIn`

Default location for scanned images to import into docWizz.

#### 5.1.2 RESCAN

default: `DWshare\Rescan`

Default location for scans of pages which are added/replaced to the document during processing.

For example missing pages, insufficient quality for OCR, ...

If documents were marked for rescan, docWizz creates a folder for the document inside. This folder contains a xml file with information about the document/page(s) need to be re-scanned.

The location of the RESCAN folder can be configured in `XYZ-docWizz-glbl.ini`.

#### 5.1.3 SCANNER\_CALIBRATION

default: `IMAGELINK1\Scanner_Calibration`

Folder for target scans for scanner calibration.

## 5.1.4 BACKUP

default: `DWshare\BACKUP`

Default location to store backups of documents in processing (dwb-files).

Those backups contains all files of a docWizz document inclusive images which were changed by docWizz (crop, align, double page splitting) and analysis results which were generated during processing.

A backup can be loaded on the same of another docWizz system.

The location of the BACKUP folder can be configured in `XYZ-docWizz-glbl.ini`.

## 5.1.5 RESTORE

default: `DWshare\RESTORE`

Contains document backups (dwb) for automatic restore by services.

If a document backup (\*.dwb) need to be restored to document pool, it will be copied to this folder automatically and restored.

The location of the RESTORE folder can be configured in `XYZ-docWizz-glbl.ini`.

## 5.1.6 EXPORT

default: `DATA\EXPORT`

Contains the final documents with all files images, METS/ALTO analysis and optional formats (like pdf, epub) for deployment.

Image files containing the page images will also be exported. After successful processing you may delete the current document from the docWizz user interface.

It is also possible to configure project based EXPORT folders to export data to several locations.

The location of the Export folder(s) can be configured in `XYZ-docWizz-glbl.ini`.

## 5.2 System environment folder

The System environment folders of docWizz share are organized as:

CONFIG  
MAINTENANCE  
INSTALL  
WORK  
DOCUMENTATION  
RemoteQAManager / RemoteQALoader

### 5.2.1 CONFIG

default: [DWshare\config](#)

This folder is the starting point for environment configuration. Here are the global site settings like machines, folder locations, database connections, services as well as user interface customizations.

DetailTreeVerifyHierarchy.DWX	Detail Tree Verify Hierarhy
docWizz-dw.DWX	docWizz
docWizz-dw.INI	Configuration of docWizz
docWizz-dwsrv.ini	Configuration of installed services, e.g. which computer, which groups
docWizz-glbl.ini	Main configuration settings for everything that is not project based
Exported.DWX	Exported
LowDisk.xml	Configuration of warning and critical space levels on docWizz shares
PrepareImport.DWX	Prepare Import
Re-Scan.DWX	Re-Scan
Scan.DWX	Scan
SiteName.txt	Name
TRANSLATE.MDB	Not in use at the moment
UpdateConfig.ini	Handles system wide docWizz updates. Disable/enable, include/exclude computers from update
userlabels.xml	Contains user defined lables which can be used to categorize documents in POOL dialog
VerifyDbIPageFrames.DWX	Verify Double Page Frames
VerifyHierarchy.DWX	Verify Hierarchy
VerifyLayoutElements.DWX	Verify Layout Elements
VerifyPageFrames.DWX	Verify Page Frames
VerifyPageNumbers.DWX	Verify Page Numbers
VerifyPages.DWX	Verify Pages
VerifyPagesHierarchy.DWX	Verify Pages Hierarchy
MultipleDocument.dsl	Multiple document

Multivolume_Monograph.dsl	Multivolume Monograph
Newspaper-Basic-PageClasses.dsl	Newspaper Basic Page Classes
Newspaper-BasicExtension-VLE.dsl	Newspaper Basic Extension Verify Layout Elements
Newspaper-Export.dsl	Newspaper Export
Newspaper-Extension-VH.dsl	Newspaper Extension Verify Hierarchy
Newspaper-Extension-VLE.dsl	Newspaper Extension Verify Layout Elements
Newspaper-VH.dsl	Verify Hierarchy for newspapers
Newspaper-VLE.dsl	Verify Layout Elements for newspapers
Newspaper-VPH.dsl	Verify Page Hierarchy for newspapers
Newspaper-VPN.dsl	Verify Page Names for newspapers
Newspaper.dsl	Newspaper
Serial.dsl	Serial
Statistics.dsl	Statistics

\*.dsd - document structure definition or document style definition, contains links to the \*.dsl

\*.dsl - document structure library

\*.dwx - definition of masks, jobs or scripts called from a button

\*.ini - configuration for docWizz client (application)

#### 5.2.1.1 BasicCfg

default: [DWshare\config\BasicCfg](#)

This subfolder contains definitions for the structures and entities used in the hierarchy tree and METS/ALTO files. (MODS, DublinCore, Monographs, Newspapers).

#### 5.2.1.2 MiscCfg

default: [DWshare\config\MiscCfg](#)

Contains export definition files for various document types (like monograph.xml, serial.xml).

ARIALUNI.TTF	Arial font type
export.xml	Export definition
job_workflow.xml	Job workflow definition. What is allowed to route in which step
MonographED.xml	Monograph export definition
MultiVolumeMonographED.xml	Multi Volume Monograph export definition
NewspaperED.xml	Newspaper Export Definition
OnlyMD-MonographED.xml	Only Multiple Document Monograph export definition
OnlyMD-MultiVolumeMonographED.xml	Only Multiple Document Multi Volume export definition
OnlyMD-NewspaperED.xml	Only Multiple Document Newspaper export definition
OnlyMD-SerialED.xml	Only Multiple Document Serial export definition
profiles.xml	Profiles
scan_job_workflow.xml	Scan job workflow



scan_workmodes.xml	Scan processing steps
SerialED.xml	Serial export definition
statistic.xml	Statistics definition
uidcounter.id	Page counter
WINDOWS-1257.enc	8 bit encoding of Windows operating system. Used to support the Estonian, Latvian and Lithuanian languages.
workmodes.xml	Processing steps

### 5.2.1.3 PVSCFG

default: [DWshare\config\PVSCFG](#)

Property View Style Configuration - contains configurations for appearance and properties of the user interface.

Book-DW.xml	Book document types step configuration for UI actions, lists, toolbars, etc
Book-DW_Types.xml	Book document types file used for zone types ordering
docWizz-Gen.xml	Basic docWizz job configuration
docWizz-VSCfg.xml	docWizz visual style configuration file for UI
docWizz-VSMngr.xml	ViewStyleToUse - docWizz views per step ( outline / detail )
FastOCRTyping.xml	Definition of fast OCR typing lists
Newspaper-DW.xml	Newspaper document types step configuration for UI actions, lists, toolbars, etc
Newspaper-DW_Types.xml	Newspaper document types file used for zone types ordering
outline list names.xml	A guide to unique list names – not in use at the moment
pvs.xsd	Property View Style. Merge of custom PVS is supported.

### 5.2.1.4 QA

default: [DWshare\config\QA](#)

This QA subfolder of config contains configuration for RemoteQA. FTP settings as well as Manager and Loader configuration.

XYZ-TransferConfig.tpl	
ftp_prio_config.tpl	Priority settings for RQA FTP transfer
FTPConfig.tpl	
FTPConfig.xml	Standard FTP configuration
RemoteQALoaderCfg.tpl	
RemoteQAManagerCfg.tpl	
RemoteQAManagerCfg.xml	Remote QA Manager configuration

## 5.2.2 MAINTENANCE

default: [DWshare\MAINTENANCE](#)

This folder contains tools, data and scripts and backup files for environment backup, restore and updates.

## 5.2.3 INSTALL

default: `DWshare\Install`

Here are the binaries, runtime redistributables, and scripts of the environments latest version.

The automatic update feature uses this for updating clients and services.

The folder contains the subfoldersdWBin (for binaries update), dWOOCR (for ocr update) and dWScript with scripts.

## 5.2.4 WORK

default: `DWshare\work`

This folder represents all activities within the dW-environment (status of services and running client, priorities, logs, extratasks). This information can be accessed with dWControlCenter.

autoimport.txt	Timestamp file for IMPORTIMAGES task
CleanPool.txt	Timestamp file for CLEANPOOL task
DocsPrio.txt	Timestamp file for DOCPRIO task
<TimedTask>.txt	Empty files whose timestamp indicate the last run of the specific task. (For IMPORTIMAGES, CLEANPOOL, RESTORDOCUMENT)
docWizz Scanning Interface.\$a\$	Indicates a running docWizz ScanClient. Can also have extension *.\$b\$, *.\$c\$, etc. if more instances in use.
docWizz.\$a\$	Indicates a running docWizz client. Can also have extension *.\$b\$, *.\$c\$, etc. if more instances in use.
DWControlCenter.\$a\$	Indicates a running docWizz Control Center. Can also have extension *.\$b\$, *.\$c\$, etc. if more instances in use.
DWFTPClient.\$a\$	Indicates a running docWizz FTP Client. Can also have extension *.\$b\$, *.\$c\$, etc. if more instances in use.
DWOCCRRemote.\$a\$	Indicates a running remote OCR. Can also have extension *.\$b\$, *.\$c\$, etc. if more instances in use.
DWRemoteQAManager.\$a\$	Indicates a running docWizz Remote QA Manager. Can also have extension *.\$b\$, *.\$c\$, etc. if more instances in use.
DWSrv.\$a\$	Indicates a running docWizz service. Can also have extension *.\$b\$, *.\$c\$, etc. if more instances in use. Also DWSrv2, DWSrv3 and DWSrv4 are possible.
DWSrvManager.\$a\$	Indicates a running docWizz service manager. Can also have extension *.\$b\$, *.\$c\$, etc. if more instances in use.
FTP_PRIO.txt	Timestamp file for FTPPRIO task
macs.ini	Saved MAC addresses of computers which are configured to run nightmode (configurable in xyz-docWizz-dwsrv.ini)
PG.CNT	Page counter

Priorities.xml	Priority settings which can be configured in docWizz Control Center.
RestoreDocument.txt	Timestamp file for RESTORE task.
wakeup.txt	Timestamp file for WAKEUPCOMPUTERS task
<project>-RQA-Sharing.wrk	Handles the RQA sharing of documents. Defines the RQA location for the next document. No manual changes are necessary on this file. docWizz will automatically change the file based on settings made in ...\\project-cfg\\<project>-RQA-Sharing.xml

#### 5.2.4.1 ClientConfig

default: [DWshare\\work\\ClientConfig](#)

This directory contains client configuration files, user-, machine- and ini-dependant settings.

#### 5.2.4.2 Extratasks

default: [DWshare\\work\\extratasks](#)

Folder for custom tcl scripts executed by the ExtraTasks-feature. WORK\\Extratasks\\OCRTasks this folder for custom tcl scripts executed by the ExtraTasks-feature run on services providing OCR.

#### 5.2.4.3 FTPClient

default: [DWshare\\work\\FTPClient](#)

This directory contains log files of FTP transfer to different RQA locations.

#### 5.2.4.4 OCRSERVICE

default: [DWshare\\work\\ClientConfig](#)

Shows the OCR license and is structured like:

WSC-04	.	5620	.	FR10	.	antiqua	.	SV-PC	.	0	.	DWSrv	.	732	.	\$LF
<u>serverMachineName</u>		<u>ocrVersion</u>				<u>clientMachineName</u>		<u>instanceName</u>								<u>extension</u>
		<u>clientPid</u>		<u>type</u>				<u>retryCounter</u>							<u>serverPid</u>	

#### 5.2.4.5 Lock

default: [DWshare\\work\\Lock](#)

This directory contains files for each service and document which use these to lock them.

<documentID>.\$xm	Indicates a document which is currently in use. Computer and used instance can be found inside the file.
-------------------	--

<instance>.\$a\$.txt

Indicates a running docWizz instance. Values can be docWizz Scanning Interface, docWizz, DWControlCenter, DWFTPCClient, DWOCRRemote, DWRemoteQAManager, DWSrv and DWSrvManager.

Computer and used instance can be found inside the file.

#### 5.2.4.6 Service Status

`work\ServicesStatus\<MachineName>\DWSrv<X>\CMD`

This directory contains command (start, stop, restart, shutdown, kill) files for the different docWizz services; those files will be written on execution of the different commands in docWizz ControlCenter (a log file will be stored here which saves the information about the document processing (document ID, job, project name, document title, duration, result etc.)).

`work\ServicesStatus\<MachineName>\DWSrv<X>\Srv\<MachineName>.txt`

Is the latest status message of the service.

`work\ServicesStatus\<MachineName>\DWSrv<X>\Log\<MachineName>.txt`

Are the Log files of the service.

### 5.2.5 DOCUMENTATION

default: `DWshare\documentation`

This folder contains manuals for users and administrators along with copyright information on third party software.

#### 5.2.6 RemoteQAMngr / RemoteQALoader

default: `DWshare\RemoteQAMngr`

default: `DWshare\RemoteQALoader`

These folders contain data/command files and logs for RemoteQA.

## 5.3 Processing configuration folders

The Processing configuration folders of docWizz share are organized as:

- SCRIPT
- SCHEMA
- PROJECT-CFG
- LISTS
- SCANNER-INFO
- SCANNER\_CALIBRATION
- TRAINING
- MetadataDB
- DICT

### 5.3.1 SCRIPT

default: `DATA\script`

This folder contains project independent script files for processing customization. Project based scrips are located in the `project-cfg` folder.

`...bk -` for books  
`...NW -` for newspapers

AdaptiveThreshold.TCL	Enhanced black&white image
AddDWScanIssuesStartFlags.tcl	Script used to gather information about issues from docWizz Scan to docWizz document
AddDWScanMetadata.tcl	Script used to gather the metadata information from docWizz Scan to docWizz document
AdjustRDYChanges.tcl	Click update configuration
ApplyRescan.tcl	Script file to handle ApplyRescan job
ApplyShowOrigin.tcl	Script file to apply show on origin
BasicCropPostDPF.tcl	
BuildBWImage.tcl	Script used to generate the BlackWhite image for docWizz documents that will be used for analysis
BuildHierarchy.tcl	Build Hierarchy
BuildHierarchy_BK.tcl	Build Hierarchy books
BuildHierarchy_NW.tcl	Build Hierarchy newspaper
BuildPagesHierarchy.tcl	Build Pages Hierarchy
BuildPagesHierarchy_BK.tcl	Build Pages Hierarchy books
BuildPagesHierarchy_NW.tcl	Build Pages Hierarchy newspaper
ClippingArticle.TCL	For clipping functionality
ColumnMerge.tcl	Experimental script not released
Delete.tcl	Delete

DetectFingers.TCL	Script used to detect operator's fingers on scanned documents
DetectLayoutElements.tcl	Detect Layout Elements
DetectLayoutElements_AA.tcl	Analysis script that will run on documents with the onlyAA profile
DetectLayoutElements_bk.tcl	Detect Layout Elements books
DetectLayoutElements_CRD.tcl	Analysis script that will run on documents with Card document type
DetectLayoutElements_NW.tcl	Detect Layout Elements newspaper
DetectPageFrames.tcl	Detect Page Frames
DetectPageNumbers.tcl	Detect Page Numbers
DetectPageNumbers_BK.tcl	Detect Page Numbers books
DetectPageNumbers_NW.tcl	Detect Page Numbers newspaper
docWizz_ItemTracking.tcl	Item tracking script - related to the remote item tracking and docWizz collaboration
docWizz_Main.tcl	Main script
docWizz_QA.tcl	Script used for QA Tasks (Editing tasks), mainly for double keying feature of docWizz
docWizz_RejectLIB.tcl	Default for reject checks
docWizz_UI.tcl	Only for Client user interface
Export.tcl	Export
Export_BK.tcl	Export books
Export_NW.tcl	Export newspaper
GenerateSchema.tcl	Create *.xsd from schema definitions
Import.tcl	Import
InOutMetadata.tcl	Script that performs actions on the metadata database.
KirtasDPF.tcl	For the use of Kirtas scanners
KirtasDPFFct.tcl	For the use of Kirtas scanners
KirtasDPFSP.tcl	For the use of Kirtas scanners
ModifyPages.tcl	Modify Pages
NW_GetFastFramesDoublePage.tcl	For the use of Kirtas scanners
NW_GetFastFramesOnPageLib.tcl	Script used to store functions for the newspaper FastFrame detection algorithm.
NW_GetFastFramesSingePage.tcl	For the use of Kirtas scanners

OCRDetectLanguage.tcl	Try to autodetect proper language from an extended list of languages. Is not very efficient specially for engines with payment per page usage, since language is detected based on text statistics after running OCR multiple times for each zone, with different language settings. Might be useful when engine autodetection do not work properly, or if document contains texts in more than three languages.
OCREmpty.tcl	Script used when is needed to skip OCR detection on a particular step (f.e. for Fraktur languages, when in order to not use too many pages, OCR is skipped for most of the jobs, and only performed on export).
OCRHelpers.tcl	Script used to store functions for OCR - gathering different info about OCR - bad confidence, minimum confidence, average confidence, etc.
OCRImageEnh.tcl	Today is not used too much in practice. It performs OCR on default image and if quality is not good enough for some individual zones (based on several confidence criteria), perform OCR on enhanced images for those zones. Algorithms for enhanced images are defined in dwShare\config\ConfigTool\ocr_values.xml, and procedures are defined in OCRHelpers.tcl.
OCRImageEnhAnFrAnMerge.tcl	Use BW vs GS images and Fraktur vs Antiqua font and use best result in terms of confidence.
OCRImageEnhForce.tcl	Today is not used too much in practice. It is similar to OCRImageEnh.tcl, but will not try first with original image - will run once on enhanced images.
OCR-PDF.tcl	Script used when page source is a PDF file. Tries first to get text from PDF and in case there is not text available, performs regular OCR for that particular zone.
OCR-PDFComp.tcl	Similar with OCR-PDFVote.tcl, but additionally compare the results and toggle the lines, so that best option is used as default one (based on different typical issues of PDF text found during the time). For PDF-Vote.tcl, always version from PDF is considered the good one.
OCR-PDFVote.tcl	Run OCR and extract text from PDF; set both texts using variant mechanism in docWizz, so that the user choose the best option from both options (OCR vs PDF).
OCRScript.tcl	Not used in present. One of the first OCR scripts, trying different options of OCR (different font types, different source images - GS/BW, etc); it is used only as sample how to implement such a script.
OCRScriptFRorAN.tcl	Not used in present; old script used when Abbyy FR was not able to autodetect the font type.
OCRScriptFRorAN_Tesseract.tcl	Script performs OCR with both settings and combine the results based on confidences (which are calibrated to work with Tesseract).
OCR-TessComp.tcl	Not used in present; old script, not possible to run with last OCR implementation (using old TesseractWrapper)
OCRTesseract.tcl	Not used in present; old script, not possible to run with last OCR implementation (using old TesseractWrapper)

PageAddFixFrame.tcl	
PageEdge_BlackMargin.tcl	
PageEdge_BlackMarginFct.tcl	
PDFBuilder.tcl	Script for generating PDFs by PDFBuilder.dll
PDFBuilderFct.tcl	Script for generating PDFs by PDFBuilder.dll
PDFBuilderSampleCall.tcl	Script for examples of calling the TCL solution of PDFBuilder.  This script uses the functions of "PDFBuilder.tcl"
PrepareImportUI.tcl	Script used for PrepareImport user interface. Control actions and events (update controls, enable disable, fill, selection).
RecognizeLayout.tcl	
RecognizeStructure.tcl	
RecognizeText.tcl	
Reject.tcl	Under special conditions articles can be rejected
RemoveFingers.TCL	For the use of Kirtas scanners
RestoreVleZones.tcl	
ScriptErrorCodes.tcl	Script used to define the script error codes by modules. the error codes must be unique.  Reported error codes in errorlog for scripts are from 1000 to 1999.
StatisticDictionaryHelper.TCL	Script used to store functions for OCR quality check and spellcheck

#### 5.3.1.1 dwx

default: DATA\script\dw\...

\_Readme.txt  
 ComputeOCR.dwx  
 ComputeOCR.tcl  
 ComputeOCR-docWizz-dw.INI  
 ExportToDoc.dwx  
 ExportToDoc.tcl  
 ExportToDoc-docWizz-DW.ini

#### 5.3.1.2 epub

default: DATA\script\epub\...

Standard script files and xml configuration to handle ePUB export.



chapter_template.xml	
EPUB_Library.tcl	File containing most of the script files used for epub generation
EPUB_Main.tcl	Scripts used to generate EPUB files (called from KExportXML::CreateEpubFiles function)
EPUBConfig.xml	EPUB configuration file to configure image quality and bookmarks of the EPUB file
EPUB-GenFromMets.tcl	Scripts used to generate EPUB files from Exported documents
EPUB-GenFromMetsLibrary.tcl	Used to read the configuration file and initialize the global variables for generating EPUB files from Exported documents

#### 5.3.1.3 export

default: DATA\script\export\...

BasicExport.tcl  
BasicExportCfg.tcl

#### 5.3.1.4 Export\_to\_dw\_doc

default: DATA\script\Export\_to\_dw\_doc\...

This folder contains scripts and configuration to create docWizz documents from exported files (METS/ALTO, images). This is useful, if no document backup (\*.dwb) is available and additional work on the files is necessary.

Note: Some zone types are not able to restore from exported files and will not be available in restored document.

Create\_Extra\_Task.TCL  
Run\_Without\_Extra\_Task.TCL  
Sample\_Task.tcl  
Add\_Profile\_To\_doc.tcl  
AddDmdSecMetadata.tcl  
export\_to\_doc\_cfg.xml  
GenerateScripts.tcl  
Mets\_Parse.tcl

### 5.3.1.5 ImageProcessing

default: DATA\script\ImageProcessing\...

Image processing scripts can be used in Scan mask using "Image Enh." drop down..

The scripts contain settings which were taken from Photoshop and can be adapted to your personal needs if necessary.

Complex.tcl	Image processing for complex images
ComplexNoisy.tcl	Image processing for complex and noisy images
Simple.tcl	Image processing for simple images
SimpleNoisy.tcl	Image processing for simple and noisy images

### 5.3.1.6 lib

default: DATA\script\lib\...

DHSHelpers.tcl	
DoOCR.tcl	Library script for OCR actions
Ent.tcl	Library script for entities actions
FillPageNumberSeries.tcl	Library script for Fill Page Number series actions
ini.tcl	
MarcToDcMods.tcl	Library script for converting MARC metadata to DC/ MODS
Math.tcl	Library of math functions
MetadataProcs_v1.1.tcl	Library for metadata functions
NavigateTree.tcl	Library script for navigating docWizz tree
ns_Actions.tcl	
ns_BinaryToHexa.tcl	
ns_buildBWImageAlg.tcl	Library script used for generating BW image used by docWizz
ns_Database.tcl	Library script for database actions
ns_Datetime.tcl	Library script for date and time actions
ns_dWWebService.tcl	
ns_Excel.tcl	
ns_Export.tcl	Library script for export actions
ns_File.tcl	Library script for file object actions
ns_Frame.tcl	Library script for frame actions
ns_Issue.tcl	Library script for Issue actions
ns_Metadata.tcl	Library script used for metadata actions
ns_Mets.tcl	Library script for mets file actions
ns_OcrCfg.tcl	Library script for OCR configuration actions
ns_Tesseract.tcl	

ns_Validate.tcl	Library script for validation of output ( JHOve validation, etc)
ns_Xml.tcl	Library script used for xml file actions
ns_XMLText.tcl	Library script for XMLText object actions
ns_Z39.tcl	Library script for Z39 metadata records actions
SetOcrPlace.tcl	Library script used for OCRPlace actions
TCLProcs.tcl	Library script for various TCL scripts used in docWizz
ToCorrectXML.tcl	Library script used for XML actions

#### 5.3.1.7 MLHelpers

default: DATA\script\MLHelpers\...

DHS\_LAHelpers.tcl  
UsageSample.tcl

#### 5.3.1.8 reject

default: DATA\script\reject\...

Script files to handle reject conditions.

AutoFixHelpers_VH.tcl	
AutoFixPageDimensions.tcl	Script for autofixing page dimensions
Reject_Autofix_OCR.tcl	Script for OCR autofixing - compute missing OCR
Reject_Autofix_VH.tcl	Script for autofixing various problems in Z-RSTstep
Reject_Autofix_VPF.tcl	Script for autofixing various problems in VPF step
Reject_InitPlaceForZones.tcl	Reject initialization of place where reject to be triggered
RejectByEnt_Actions.tcl	Script to enable some actions on reject list ( Do Antiqua/ DoFraktur )
RejectByEnt_Cond.tcl	Reject script for default reject definitions
RejectByEnt_CondOCR.tcl	Rejects for various OCR problems
RejectByEnt_CondOCRTxtvsDict.tcl	Script used for various language checks
RejectByEnt_CondX.tcl	Various reject calls (zone only on page )
RejectByEnt_Main.tcl	Main reject run script
RejectByEnt_MetadataValidation.tcl	Reject scrips for metadtaa validation
RejectByEnt_Task_Cond.tcl	Rejects for editign tasks ( when going from one step to another )
RejectByEnt_UI.tcl	UI reject actions

#### 5.3.1.9 remoteQAScripts

default: DATA\script\remoteQAScripts\...

Only available in connection with Remote QA.

CollectFromClientPool.TCL	Collect documents from loader side to send back on manager
DocImages.tcl	Script for RQA image manipulations
MngrCollectData.tcl	Collect documents from manager to send on Loader
PoolStatusResLdr.tcl	Script collects document processing data and writes to log database
PoolStatusTaskLdr.tcl	Script collects document processing data and writes to log database
PoolStatusTaskMngr.tcl	Script collects document processing data and writes to log database
QAFctHelpers.tcl	Helper functions
SetDocOnClientPool.TCL	Update existing documents/ add new documents on loader
SetDocOnManagerPool.TCL	Update documents on manager with loader data

#### 5.3.1.10 Samples

default: DATA\script\Samples\...

In use to place sample script files.

PDFA2bXMPSample.xml	
PDFBuilderSampleCall.tcl	
Sample-GeneratePDFTasks.TCL	Generates task command files for recreation of PDF files from exported METS

#### 5.3.1.11 Setup

default: DATA\script\Setup\...

PostSetup.tcl

#### 5.3.1.12 task

default: DATA\script\task\...

Script files for task handling.

_Delay_Task_.tcl	Script that sets the delay between running the same task
_Interrupt_Task_.tcl	Script that handles when a task is cancelled
AutoImport.tcl	IMPORTIMAGES task
BackupConfigs.tcl	BACKUPCONFIGS task
CleanPool.tcl	CLEANPOOL task

CreateDocumentBackup.tcl	
CreatePriorityImpFileForFtp.tcl	FTPPRIO task
CreateValidatorExtraTasks.tcl	Script used to generate validation extratasks
DictionaryInsertExtraTask.tcl	Script used to generate dictionary insert words extratasks
DocsPrio.tcl	DOCPRIO task
ExtraTask.TCL	EXTRATASKS and EXTRATASKSWITHOCR tasks
ExtraTaskDoOCR.tcl	Extra task script
ExtraTaskDoOCREditing.tcl	Extra task script
ExtraTaskDoReject.tcl	Extra task script
ExtraTaskGenerate.TCL	Extra task script
ExtraTaskGenerateRestoreBackups.tcl	Extra task script
ExtraTaskWithOCR.TCL	Extra task script
ExtraTaskWithOCRGenerate.TCL	Extra task script
Restore.tcl	RESTORE task
ValidateDocument.tcl	Script used to generate validation extra tasks files
Wakeup.tcl	WAKEUPCOMPUTERS task

### 5.3.2 SCHEMA

default: `DATA\schema`

This folder contains schemas to check whether the final output METS/ALTO contain valid structures according to existing standards of w3.org (defining xml or html) or the Library of Congress (defining MODS, METS, ...). Additionally it contains stylesheets to transform.

The subfolder DATA\Schema\offlineschema will be used as local validation if internet connection is not accessible.

\*.xsd - XML schema definition  
 \*.xsl - Extensible stylesheet language for style transformations

alto-1-1.xsd	ALTO (Analyzed Layout and Text Object) is an extension schema to METS, describing the layout and content of for example single pages.
alto-1-2.xsd	ALTO version 1.2
alto-1-4.xsd	ALTO version 1.4
alto.xsd	ALTO
ALTO2TEXT.xsl	Transformation ALTO to text
dc.xsd	Dublin Core - Interoperable metadata standard
DIVPDFBuilder.xsl	Stylesheet for PDF generation
docWizz_DocDefExport.xsd	Export schema definition.
docWizz_Export.xsd	Export schema definition.

docWizz_ExportValidator.xsd	Export schema definition for docWizz Export Validator.
docWizz_Import.xsd	Validation schema of workflow files (*.xml files can be found in project-cfg\rdy folder
ExportValidator.xml	Export Validator configuration
LogicalStruct.xsd	Schema definition for logical struct map in METS file.
MARC21slim.xsd	MARC (Machine-Readable Cataloguing)
marctomods.xsl	MARC to MODS transformation
MARCTXMLtoDC.xsl	MARC to Dublin Core transformation
MARCTXMLtoDCUtils.xsl	MARC to Dublin Core Utils transformation
METAe.xsl	Transformation, text output
mets-metae.xsd	CCS validation
mets.xsd	Original METS - Metadata Encoding And Transmission Standard
mix.xsd	NISO Metadata for Images in XML Schema This is a standard for technical metadata for digital still images
mix_jp2.xsd	MIX, Jpeg 2000
mods-3-0.xsd	MODS - Metadata Object Description Schema Is a part of the MARC convention
mods-3-1.xsd	MODS
mods-3-1.xsd	MODS
monograph.xsd	document type specific
MultipleDocument.xsd	used to specify and validate logical structmaps in METS files
Multivolume_Monograph.xsd	used to specify and validate logical structmaps in METS files
newspaper.xsd	used to specify and validate logical structmaps in METS files
OCRConfig.xsd	Schema definition for OCR configuration.
OnlyMD-Multivolume_Monograph.xsd	Schema definition used to specify and validate logical structmaps in METS files for document type Multivolume_Monograph.
OnlyMD-Newspaper.xsd	Schema definition used to specify and validate logical structmaps in METS files for document type Newspaper.
OnlyMD-Serial.xsd	Schema definition used to specify and validate logical structmaps in METS files for document type Serial.
rdf.xsd	resource document framework for Dublin Core
serial.xsd	document type
transTypeToTagName.xsl	creates logical structure.xml from METS
xlink.xsd	XML linking schema

### Reference Schema separately

The following schemas are referenced individually according to export configuration in order to handle different versions of each schema:

METS  
ALTO  
MIX  
MODS  
DC  
MARK  
RDF  
Xlink

Schema directory contains folders containing a set of above schema, each in standard version (a package of schemas). There is a default schema with current versions of schemas.

For future, older packages need to be kept in order to keep current productions running.

### 5.3.3 PROJECT-CFG

default: `DATA\project-cfg`

This directory contains standard configuration for certain types of material (monograph, newspaper, serial), as well project specific configurations and scripts.

#### 5.3.3.1 Default

default: `project-cfg\Default`

This directory contains standard configuration files which are taken into account if not existing explicitly in PROJECT\_CFG subfolder.

Default-DSD.txt	In this file are mentioned the "document style definitions" valid for this project i.e. Monograph
Default-externMD.tcl	External metadata file - can be used for customization of the metadata on project
Default-GetTextCtrlActionsList.tcl	Script used for actions ( right click, menu ->Run..), can be used for customization per project
Default-import.tcl	Default project script used for import can be used for customization per project
Default-LACfg.xml	Layout analysis configuration file
Default-OCRCfg.xml	OCR configuration file for OCR settings
Default-OUTLIST_init_proc.tcl	Script used for populating Outline ListView lists
Default-procs.tcl	Script used for customizing procedures ( Re-Scan / metadata etc )
Default-QA-Tasks.xml	Editing tasks configuration file ( define and configure editing tasks)
Default-Setting.xml	Project settings file to enable/disable some features( rqa settings, editing tasks settings, allow delete page )
FrequentlyUsedUnicodeChars.txt	File used for text view insert special character dialog - to place first these characters
Default-LowDisk.xml	Configuration of warnings if system has low disk space. Configures when services should stop. Shown in the docWizz Control Center. Only default settings, not project specific.

<Default>-post<Job>.tcl	Custom processing done before the <Job> is executed.
RQA-Sharing.xml	Handles number of documents to go to each defined RQA location.

### 5.3.3.2 Clipping

Default-clipping-export.tcl	Post process export for Clipping
Default-clipping-export.xml	Settings for clipping export
Default-clipping.tcl	Clipping export script commands
	Possible values:
	"NOARRANGE"
	"HEADLINE"
	"BLOCK"
	"COMPACT"
	"SIMPLE"
	could be entered as return value.
	In the Configuration always:
	<METHOD TYPE="NOARRANGE"> is used, because these are the special parameter for the 1:1 clipping
Default-clipping.xml	Configuration file for docWizz clipping
Default-Clipping-Verify.tcl	This is used for verification of clippings in VH/OLV.

### 5.3.3.3 rdy

Here are the several "ready files" which contain the processing information for this project. The xml files will be copied as rdy into the document folder in the In-Share to initiate the import process.

docWizz checks, if the \*.rdy file has correct syntax.

Ready files are used to automate the workflow process. You may predefine one or more RDY files for each project and store it in a subfolder of the special project configuration called "RDY". The files must have RDY as extension.

Monograph.xml	Workflow file for Monograph processing.
Newspaper.xml	Workflow file for Newspaper processing.
Serial.xml	Workflow file for Serial processing.

### 5.3.3.4 dsl

This file references document style definitions in doccfg.

METAe-Extension-VLE.dsl	additional zone types
METAe-PageClasses.dsl	page types
METAe-VH-GenBook-StructureClasses.dsl	generic for all book types
METAe-VH-Monograph-StructureClasses.dsl	Document style definition for monograph document types (chapter, list, and other hierarchy structures)



METAe-VH-Multivolume_Monograph-StructureClasses.dsl	Document style definition for multivolume monograph document types ( chapter, list, and other hierarchy structures)
METAe-VH-Serial-StructureClasses.dsl	Document style definition for serial document types ( chapter, list, and other hierarchy structures)
Newspaper-PageClasses.dsl	for newspapers
Newspaper-VH-StructureClasses.dsl	for newspapers
Newspaper-Extension-VLE.dsl	Document style definition for newspaper document types to define zone types.
project-cfg/default/dsl/*-ignoretypes.dsl	Reduce the types based on project cfg - there are projects where 3-4 types of zones and two types of structures are needed.

#### 5.3.3.5 epub

Contains settings for additional ePub export.

book.css	CSS style sheets
EPUBConfig.xml	Configuration for ePub file export
titlepage.css	CSS style sheets for title page

#### 5.3.3.6 metadata

InTable_cfg.xml	configuration file for intup table document metadata DB creation/handling
OutTable_cfg.xml	configuration file for output table document metadata DB creation/handling

#### 5.3.3.7 PDF

The file "PDFCf.xml" file must be located in the subfolder 'PDF' of the project configuration.

If this file exist there, those settings will be loaded by the function 'LoadExternalVariables' of the TCL script file "PDFBuilderFct.tcl".

When loading a corrupt or password protected PDF file in scan job, a warning message box appears.

If this file is missing, the configuration of `project-cfg\Default\PDF\PDFCf.xml` will be taken.

PDFCf.xml	File contains settings for PDF creation like PDFType, Resolution, ImageQuality, Bookmarks
-----------	---

#### 5.3.3.8 PDFImportCfg

You have the option to configure the output of PDFRenderer in terms of resolution for created images. These settings can be customized into `PDFImport.xml` file.

In order to change the default values (image resolution 300 dpi, derivatives images resolution (path, img, etc) 150 dpi) copy the `PDFImport.xml` from `***PROJECT_CFG***\Default\PDFImportCfg` into `***PROJECT_CFG***\projectname\PDFImportCfg` and modify it accordingly.

## PDFImport.xml

File contains special Arabic characters used for import (should not be modified).

Two new nodes are added and they can be altered depending on request (default resolution 300 dpi, for derivatives (path, img, etc) 150 dpi).

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<PDFImportConfig>
```

```
  <ArabicSmallCharacters>
```

```
    <ch>&#x60C;</ch>
```

```
    <ch>&#x60D;</ch>
```

```
    <ch>&#x610;</ch>
```

```
    <ch>&#x611;</ch>
```

```
    <ch>&#x612;</ch>
```

```
    <ch>&#x613;</ch>
```

```
    <ch>&#x614;</ch>
```

```
    <ch>&#x615;</ch>
```

```
    <ch>&#x616;</ch>
```

```
    <ch>&#x617;</ch>
```

```
    <ch>&#x618;</ch>
```

```
    <ch>&#x619;</ch>
```

```
    <ch>&#x61A;</ch>
```

```
    <ch>&#x61B;</ch>
```

```
    <ch>&#x61E;</ch>
```

```
    <ch>&#x640;</ch>
```

```
    <ch>&#x64B;</ch>
```

```
    <ch>&#x64C;</ch>
```

```
    <ch>&#x64D;</ch>
```

```
    <ch>&#x64E;</ch>
```

```
    <ch>&#x64F;</ch>
```

```
    <ch>&#x650;</ch>
```

```
    <ch>&#x651;</ch>
```

```
    <ch>&#x652;</ch>
```

```
    <ch>&#x653;</ch>
```

```
    <ch>&#x654;</ch>
```

```
    <ch>&#x655;</ch>
```

```
    <ch>&#x656;</ch>
```

```
    <ch>&#x657;</ch>
```

```
    <ch>&#x658;</ch>
```

```
    <ch>&#x659;</ch>
```

```
    <ch>&#x65A;</ch>
```

```
    <ch>&#x65B;</ch>
```

```

<ch>&#x65C;</ch>
<ch>&#x65D;</ch>
<ch>&#x65E;</ch>
<ch>&#x65F;</ch>
<ch>&#x670;</ch>
<ch>&#x674;</ch>
<ch>&#x6DB;</ch>
<ch>&#x6DF;</ch>
<ch>&#x6E0;</ch>
<ch>&#x6EA;</ch>
<ch>&#x6EB;</ch>
<ch>&#xfe70;</ch>
<ch>&#xfe74;</ch>
<ch>&#xfe76;</ch>
<ch>&#xfe7a;</ch>
<ch>&#xfe7b;</ch>
<ch>&#xfe7c;</ch>
<ch>&#xfe7d;</ch>
<ch>&#xfe7e;</ch>
<ch>&#xfe7f;</ch>
</ArabicSmallCharacters>
<OutputResolution>300</OutputResolution>
<OutputResolutionDerivatives>150</OutputResolutionDerivatives>
</PDFImportConfig>

```

#### 5.3.3.9 reject

After each step the document the document has to pass several validity checks. These checks and their severity are configured here.

If a document is rejected during processing an operator will assign a reject reason according to the specification. The list of rejected books could be used later on to rework those once it is possible to digitize them due to progress or changes in scanning equipment or e.g. OCR engines if foreign languages with different character sets appear.

Default-reject-VerifyPagesHierarchy.tcl	Script handles rejects in job Z-RI. Calls InOutMetadata.tcl
RejectByEnt.xml	Definition of reject conditions and job based call of defined conditions
RejectByEntEmpty.xml	Empty reject condition file. No condition is defined and called.

### 5.3.4 LISTS

default: `DATA\lists`

Contains standard lists for e.g. reject reasons, rescan, comments, ... Needed for MODS

	Examples for list entries
analysis.lst	None Page Linking Page Linking + OCR Full Structure
authorityRT.lst authorityTI.lst	Controlled vocabulary of descriptive terms
damageTypes.lst	Page fold Page tear Other
dateEncoding.lst	w3cdtf iso8601 marc
datePoint.lst	start end
dateQualifier.lst	approximate inferred questionable
dblLanguage.lst	da Danish de German en English fi Finnish fr French nl Dutch it Italian no Norwegian pt Portuguese ru Russian sv Swedish es Spanish
defpagesize.lst	A5 148.5 210.0 A4 210.0 297.0 A3 297.0 420.0 A2 420.0 594.0

IllustrationType.lst	Illustration Map Chart/Diagram Bookplate
logunitcomment.lst	New Logical Unit identified by scan operator Logical Unit not found in original
nametype.lst	personal corporate conference
OCRType.lst	Antiqua Fraktur TypeWriter Fraktur+Antiqua Typewriter+Antiqua
Orientation.lst	0 Portrait 9000 Landscape (BR) 27000 Landscape (BL)
PERIOD.lst	1800-1850 1850-1900
processing.lst	None Crop Align Double Page Verify
rejectscanreason.lst	Pages not cut Volume binding damaged Volume cannot be opened wide enough Volume too small Volume too large Text too close to the edge

rescanreason.lst	0 - ok 1 - bad OCR quality 1 - insufficient spell check 1 - bad resolution 1 - bad quality 1 - clamps over text 1 - incomplete image 1 - wrong page 1 - want color 1 - wrong page size 1 - page skewed 1 - page cut 2 - target 3 - retained 4 - missing 5 - missing in original 6 - as in original 6 - text cut off in original 6 - page skewed in original 6 - too close to binding in original 7 - not cut properly 8 - double frame on image
scancomment.lst	Missing pages detected Text close to borders Text hidden in binding Damaged before processing Damaged at scan process
titleType.lst	abbreviated alternative translated uniform
uploaddoc.lst	too many metadata errors microfilm scanning

### 5.3.5 SCANNER-INFO

default: `DATA\Scanner-info`

Contains configuration for scanner information. These information will be written to final METS and/or image files.

### 5.3.6 SCANNER\_CALIBRATION

Target scan will put images into this folder. The configuration of this folder is required if target pages are enabled.

See [docWizz ScanClient](#) manual, chapter [ScanClient Configuration](#).

### 5.3.7 TRAINING

default: [DATA\TRAINING](#)

### 5.3.8 MetadataDB

default: [DATA\MetadataDB](#)

Currently not used on client side. Will be used with the ItemTracking tool. Checks and validates IDs and does completeness checks.

### 5.3.9 DICT

default: [DATA\DICT](#)

This directory contains the user based dictionaries.

## 6 IN

default: `DATA\IN`

Folder for data transfer. Put all raw data (images) to the IN folder.

docWizz will load the files from the folder and will create a copy in POOL folder. docWizz will not change any original files which are stored in that folder.

It is possible to configure several IN folders to load data from several locations.

The location of the IN folder(s) can be configured in `XYZ-docWizz-glbl.ini`.



## 7 OUT

default: `DATA\OUT`

Folder for data transfer. OUT folder contains all the data that need to be sent to other machines.

OUT need to be configured / scripted always in detail. There it could be defined to handle different shares, if needed.

We can't provide a general solution for that. For example it is not ok to export documents from the same physical unit to different shares. Also there might be the need to export all from one project to always the same share. Finally we have sometimes re-delivery in batches.

## 8 POOL

default: `DATA\POOL`

The pool contains subfolders for each document in processing. These folders contain:

- Processing instructions and temporary results for structure analyses, metadata.
- Several versions of the image files (b/w, rotated and cropped images, preview images).
- Backup of data for each job as zip file

This folder is the main working folder of docWizz. The most read and write operations will be done here.

It is possible to create up to 256 Pool directories to handle a huge amount of data.

The location of the POOL folder can be configured in `XYZ-docWizz-glbl.ini`.

## 9 Help Menu

With the entries in the Help menu you can open the documentation files, see the error log and statistics and check the registration.

To open the menu, please press the [ALT] key.

### PDF Documentation

This function enables you to refer to the PDF documentations of docWizz.

In this documentation folder, now, only the following are available:

- docWizz User Manual
- dWScanClient User Manual
- dW List of Shortcuts

### Online documentation

When clicking on it, redirects the user to the web page "<https://extranet.content-conversion.com/dW/SitePages/Docu/Manuals.aspx>".

### Error Log

This function enables you to refer to the Error Log window that automatically lists any errors that have occurred during the current docWizz session. In this way, support staff and docWizz administrators have optimal support when looking for the cause of irregularities in the running of the program.

Error dialog shows Code and Context as additional columns. Script command reporterror has optional parameter errorcode, which will be reported in error log database.

Column "Code" will contain an unique error code to identify which error has occurred. New column "Context" contains information like document ID. This helps to select all errors related to a single document. Databases will be extended on first start automatically.

Custom filters can be created for the Error log for identifying support problems easier and faster. The same mechanism as for Document Pool custom filter is used.

### Statistics

This function enables you to refer to the Processing Statistics window, which gives you details about the total count of processed pages, users, jobs, document types or projects.

### OCR usage statistics

To check the OCR consumption, the users can use the "OCR usage statistics" dialog from Help menu. The dialog is available for Admin users only.

In this dialog the user can:

- select the time frame: either complete, or a selected range
- select a specific OCR engine or all engines (by default, the selection is set on All)


When pressing Apply button, the values will be displayed in the edit boxes at the bottom of the dialog: Total chars, Total chars without spaces, Total pages.

If for a selected time frame / OCR engine there is no data, the dialogs will display 0.

### About docWizz Client

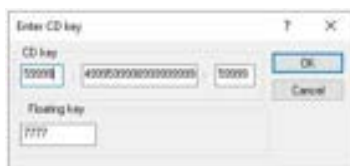
This function enables you to refer to the information window, which gives you details of the version of docWizz installed on your system.

In the view boxes you can see the name of the **User** who is currently working on the system, which **CD Key** you are using, and also the **Registration-ID** of the machine.

If you would like to extend your system with new modules or features or even additional licenses, these can be activated with a new CD Key Click the  button and enter your new CD Key number.

Enter the floating license into the field "Floating key". This key stores the possible number of parallel dW instances. You get this key from CCS.

The dialog ensures and validates, that the entered codes are valid and stored in the right ini (custom-glbl ini).



Complete the entry by clicking the [OK] button.

See additional copyright information by clicking the  button.

A selection of a component shows the additional copyright information.

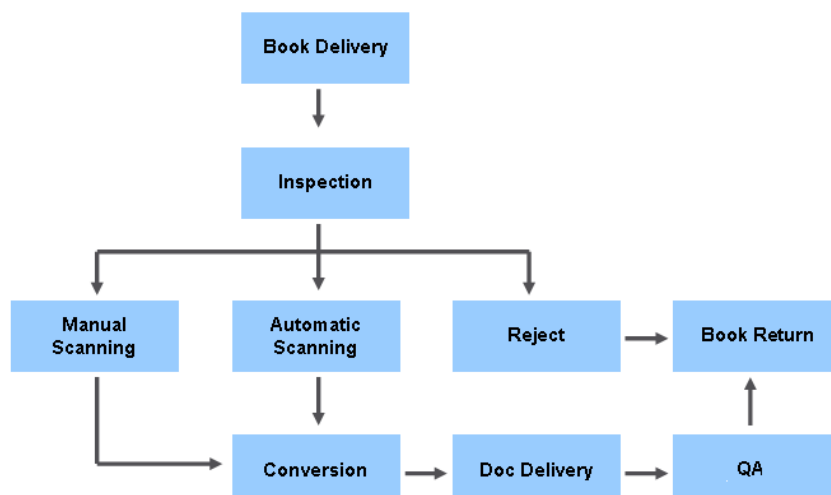
## 10 Configuration of projects

docWizz supports processing of documents using multiple configuration. Configuration includes document type and processing information as well as the configuration of the final deliverables. To each document that is loaded into docWizz, a project configuration will be assigned. By default the main folder name below the common docWizz IN-folder is taken as project name and automatically assigned to the document.

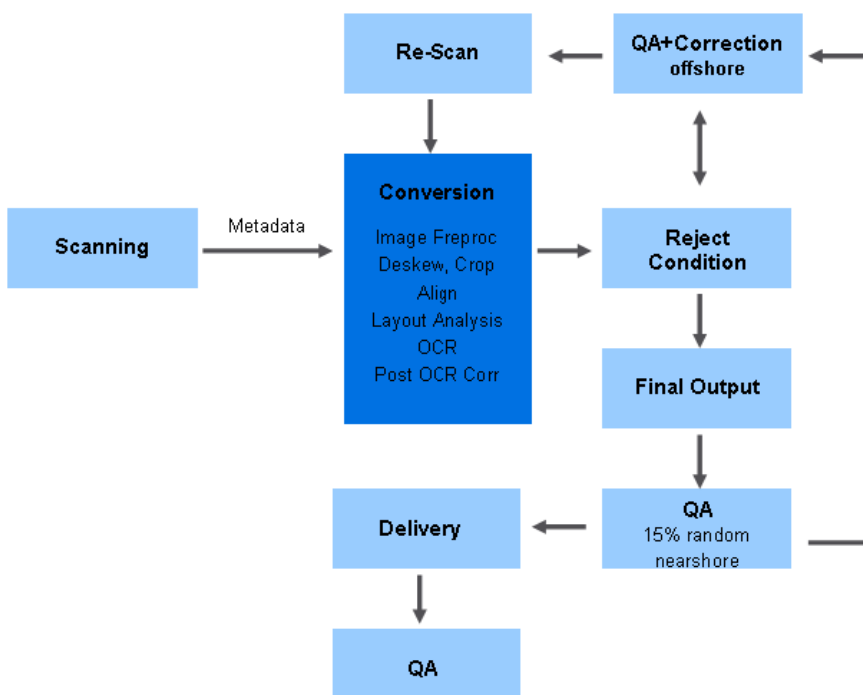
### 10.1 Workflow overview

In the following two images the workflow for an example project is shown in a brief overview.

#### Digitization Workflow



#### Conversion Workflow



## 10.2 Add a new configuration

To create a new configuration, follow the steps listed below:

- Create sub-folder in project-cfg folder, named exactly like the project in In-Folder
- Copy all files from similar project configuration folder
- Rename files (to better identify the configuration files, they must start with the project name)
- Edit the xml file(s) in RDY folder
- Edit projectname-export.xml file
- Edit projectname-export.tcl file (if special scripts shall be applied)

## 10.3 docWizz RDY files

To handle mass production with different document types and settings, docWizz supports assignment of configuration files per document.

The rdy files are validated against a schema defined at "schema\docWizz\_Import.xsd".

For each project configuration, one or more master document configuration files (workflow files) can be created. The configuration files are XML files and must be stored in the RDY subfolder of the project configuration.

Once a document has been completely scanned and moved to the docWizz IN folder, a RDY file has to be placed in the folder to notify docWizz to pick up the document and start processing. You may either copy the one of the predefined configuration files or put there just an empty file. The RDY file in the in folder must have the extension ".RDY".

Internal process of importing data and generating docWizz documents:

A docWizz background service (dWSrv) scans all subfolders below the IN folder(s) and searches for RDY files (file extension \*.rdy). Once a RDY file has been detected, a new document is created and all images from that folder are added to the document. To notify that docWizz has picked up a folder, a file "docWizz.wrk" is added in the folder.

The name of the RDY file does not matter and can be chosen free. dWSrv just parses for the rdy extension. Filename is not used.

RDY file adapted for new work modes (Advance/Basic Crop); also added `PAGESPERSCAN=1 / 2` for single/double pages.

In order to simplify the configurations and make it more meaningful the following changes are applied to rdy files

- `<IMAGE_PREPROCESSING TYPE="xxxxx" PAGESPERSCAN="y">` where `xxxxx` shall be "Align" for advanced cropping (like now) or "Crop" for simple cropping (as we have now setting for cutting by red frame). Former meaning of Crop shall not be used and configurations shall be adapted on all environments.
- `"y"` shall be "1" or "2" and tells how many frames we shall have on a scanned image
- `<PRINTSPACE UNIT="mm">`  
`<MARGIN LEFT="2.0" TOP="2.0" RIGHT="2.0" BOTTOM="2.0"/>`  
`</PRINTSPACE>`

shall be changed as follows:

```
<PRINTSPACE UNIT="mm">  
<MARGIN LEFT="2.0" TOP="2.0" RIGHT="2.0" BOTTOM="2.0"/>  
<MARGINSUP LEFT="2.0" TOP="2.0" RIGHT="2.0" BOTTOM="2.0"/>  
</PRINTSPACE>
```

This change is needed in order to handle different settings for regular and supplement pages.

## 10.4 Defining RDY files

All RDY files must be either empty or validating against the docWizz\_Import.xsd schema.

RDY files can be found in the project-cfg folder: `..\project-cfg\Default\rdy`

The root node of such an \*.xml file located in the rdy folder must have the element "DOCUMENT".

It has three child nodes:

DESCRIPTION	contains information on the document itself
PROCESSING	contains information on the processing that shall be applied
EXPORT TYPE	contains some information on deliverables

### 10.4.1 Sample RDY file

```
<?xml version="1.0" encoding="UTF-8"?>

<DOCUMENT xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://schema.ccs-
gmbh.com/docWizz/docWizz_Import.xsd">

    <DESCRIPTION TYPE="Monograph">
        <LANGUAGE Name="German"/>
        <LANGUAGE Name="Latin"/>
        <FONTTYPE>
            <FRAKTUR_ANTIQUA/>
        </FONTTYPE>
        <PROJECT>Monograph</PROJECT>
        <CONFIG>XYZ</CONFIG>
    </DESCRIPTION>

    <PROCESSING PROFILE="Default" STOPAT="VerifyPageFrames">
        <IMAGE_PREPROCESSING TYPE="Double Page">
            <DESKEW>1</DESKEW>
            <PRINTSPACE UNIT="inch">
                <MARGIN LEFT="0.2" TOP="0.2" RIGHT="0.2"
BOTTOM="0.2"/>
            </PRINTSPACE>
        </IMAGE_PREPROCESSING>
        <PAGE_LINKING/>
        <OCR/>
    </PROCESSING>

    <EXPORT TYPE="">
        <PDF/>
    </EXPORT>
</DOCUMENT>
```



## 10.4.2 DESCRIPTION element

This parameter <DESCRIPTION TYPE="xxx"> refers to the document definition used.

The description element may have the following elements:

TYPE	The basic document type which should be used for processing, like Monograph, Newspaper, Serial
SYSID	A unique identifier of the document. By default the folder name is used.
LANGUAGE	One or more languages that are the main languages in the document. The "Name" attribute defines the language name. A list of possible languages can be found below this table.
FONTTYPE	Defines the font types used in the document. Available are: <ANTIQUA/> <FRAKTUR/> <TYPEWRITER/> <FRAKTUR_ANTIQUA/> <TYPEWRITER_ANTIQUA/> You may select more than one type for the same document.
PROJECT	Defines the project the document belongs to. If this value is missing, the upmost folder name below the IN folder is taken as project name.
CONFIG	You may want to use the same configuration for multiple projects. So you can define here the name of the project where you want to take the configuration from.

### Possible values for LANGUAGE tag:

See [docWizz\\_Languages.pdf](#) for details.

### Notes for multiple language selection:

- OCR runs only once
- for the selected languages the dictionaries are loaded in addition (processing impact depending on the size of the dictionaries, but we would say in factor < 1,5 per language)
- but it is essential that the increased number of valid words in different languages increase the inaccuracy of the results i.e. due to a hit of a similar word in French and the higher character confidence (CC), the word will be set in French, but the English one would be correct.
- the recommended maximum of languages is 3, docWizz itself does not set a limit

### DESCRIPTION: SYSID

docWizz takes over the ID to the <ID.xml> as PHYSID into the document.

This ID is e.g. by default used in Export in the metadata as value for identifier in "DCMD\_PRINT".

Furthermore this allows customized adaptations e.g. to refer to and copy existing metadata to 3rd party catalogues, e.g. by Z39.50.

So the SYSID is part of the schema and due to that cause no error.

If you would like e.g. it could be customized to take over the ID into metadata of the documents. You just would need to define if it is a mandatory field (error if empty / not exists) and where to place it in metadata. Then you could do the adaption of projectCfg. field (error if empty / not exists) and where to place it in metadata. Then you could do the adaption of projectCfg.

## DESCRIPTION: FONTTYPE

docWizz is able to run Fraktur (Gothic) OCR which will be provided by third party software ABBYY Finereader XX. To use this special module, an additional license is necessary. Please contact CCS for more details.

It is also possible to run a mixture of all available font types. To enable this option, add the font types you'd like to run to <FONTTYPE> tag.

Example:

```
...
<FONTTYPE>
<FRAKTUR_ANTIQUA/>
</FONTTYPE>
...
```

Example from the ABBYY "Product & Technology Overview"(November 2010):



## 10.4.3 PROCESSING element

The processing element describes which functionality shall be applied and how. It may have the following elements:

STITCHING	enable stitching for cut pages
IMAGE_PREPROCESSING	describes any kind of image pre-processing
IMAGE_ENHANCE	defines manipulation of images (raster operations)
OCR	defines if OCR shall be applied
PAGE_LINKING	defines if image / page linking shall be applied
METADATA	defines if metadata shall be collected
STRUCTURE	defines if full structure shall be detected

SKIP_LA	skips layout analyze if only image processing and metadata is required without page linking or text
ZOOM_LEVEL	zoom factor of the image before analyze
CLIPPING	defines Clipping Profile

#### 10.4.3.1 Attributes

##### PROCESSING: PROFILE

The attribute "profile" allows you to determine special settings to the processing of your project. The values of profiles are defined at: "\config\MiscCfg\xxx-profiles.xml". Check the file for defined parameters for the related profiles. The standard profiles are defined here: "\config\MiscCfg\profiles.xml". So the file above is a subset customized on your initial requests.

You can select from the following profiles:

##### **Default**

The default profile includes all standard features of docWizz, in particular deskew, OCR on bitonal images, page linking and full structure analysis. Export of text and PDF is enabled.

##### **Despeckling**

Additionally to the default settings despeckling (or remove noise) is performed. Please have a look at chapter Despeckle to see the available parameters. Export of text is enabled.

##### **No Deskew**

Same as default profile but automated deskew is turned off. Pages can be deskewed manually in interactive steps in the GUI.

Export of text is enabled.

##### **Detect Language**

Language is detected through word confidence. Then Default profile is applied.

##### **OCR Enhance (Fraktur/Antiqua)**

Both Fraktur and Antiqua capability are applied onto the same document. Choose the primary character set/font type in the OCR type setting.

All other settings from Default profile are applied.

##### **OCR Enhance (Grayscale/BW)**

OCR is performed not only on bitonal images but also on grayscale images in order to get better results. Automatic deskew and export of text is enabled.

##### **OCR Enhance (GS/BW+Fraktur/Antiqua)**

Both the capabilities of the profiles "Enhance OCR Fraktur/Antiqua" and "Enhance OCR Grayscale/BW" are used. All other settings from Default profile are applied.

##### **OCR on Enhanced Image**

A high contrast image is created temporarily for OCR purposes only. Typically this profile is used for applying OCR on historic newspaper pages.

Enhanced Image is currently loaded per OCR call. It can be kept by using (see script files):

```
for { ... } {
  set page [...]
  set pageID [$page -num ID]
```

```

set enhancedImg [GetEnhanceImageForOCR "" $pageID ]

... do multiple OCR

$page -releasemem
delete $enhancedImg
}

```

All other settings from Default profile are applied.

### OCR on Enhanced Image & Fraktur/Antiqua

Both the capabilities of the profiles "OCR on Enhanced Image" and "Enhance OCR Fraktur/Antiqua" are used.

### Only Metadata

This profile is used if there is no interest in structure or item level metadata (chapter, article, section, preface, title section, etc.). Only metadata of issues (date, issue number, volume number) is entered in VerifyPagesHierarchy. As an effect the METS file has just basic entries in its logical structmap which typically represents the detailed structure of a digital object. VerifyHierarchy is skipped.

Automatic deskew is enabled.

### Only Metadata With Text

In addition to the above mentioned properties of the profile "Only Metadata" OCR is performed. Automatic deskew is enabled.

### Only Page Linking

Neither metadata (e.g. issue level metadata) nor structure (chapters, article, preface) is captured with this profile but just page linking is applied to verify the page sequence and detect missing pages. VerifyPagesHierarchy and VerifyHierarchy are skipped.

### Only Page Linking With Text

In addition to the above mentioned properties of the profile "Only Page Linking" OCR is performed with this profile.

### PDF Import

A special profile for the import of PDF files. If there is hidden text in imported PDF, it will be taken as OCR result, no matter what settings are in OCR configuration file (\*\*\*PROJECT\_CFG\*\*\*\ProjectName\OCRCfg.xml). Export of text and PDF is enabled.

Furthermore, you can create customized profiles that serve to your special needs:

Copy and paste the file `..\docWizz\config\MiscCfg\profiles.xml`

into the same directory and rename it to `<customer_name>-profiles.xml`, e.g. CCS-profiles.xml

Edit the new file and add a profile by copying into the file the following lines:

```

<PROFILE name="YourProfile">
    <SETTING name="Deskew" enabled="1"/>
    <SETTING name="Despeckle" enabled="0"/>
    <SETTING name="quality" enabled="1">
        <PARAM name="value" type="1" value="1"/>
    </SETTING>
</PROFILE>

```

```

        </SETTING>
        <SETTING name="OCRScripts" enabled="0">
            <PARAM name="file" type="4"
value="***DATA***\script\OCRScript.tcl"/>
            <PARAM name="Fr_An" type="1" value="0"/>
            <PARAM name="GrS_BW" type="1" value="1"/>
            <PARAM name="Lang" type="1" value="0"/>
        </SETTING>
        <SETTING name="TextExport" enabled="1"/>
        <SETTING name="PDFExport" enabled="1"/>
    </PROFILE>

```

Edit the entries according to your requirements.

By removing the lines of a profile you can delete a profile from the dropdown menu in the GUI in the Scan or PrepareImport job. You might want to do this for making usage more convenient.

#### 10.4.3.2 STOPAT

Defines the aim of the next processing you start with your choice in the window Next Job.

```

"Scan"
"VerifyPageFrames"
"VerifyPages"
"VerifyLayoutElements"
"VerifyPageNumbers"
"VerifyPagesHierarchy"
"VerifyHierarchy"
"Exported"

```

#### 10.4.3.3 ALLOWSPPLIT

Allows split of an imported document (e.g. a complete reel of newspaper pages) to decrease amount of data in a single document. If this attribute is missing, document split will be allowed.

#### 10.4.3.4 STITCHING

Enables stitching for cut pages.

#### 10.4.3.5 IMAGE\_PREPROCESSING

Defines image preprocessing.

#### **Attribute: TYPE (Align, Crop, None, ...)**

Defines the way of modifying image boundaries.

#### **Align**

The printspace will be detected and aligned on new pages. All images will have the same size. Left and right pages can be assigned to get best results for example for reprinting. Individual page sizes can be entered anyway.

#### **Crop**

Black or noisy borders will be removed, the position of the printed text will not change. Each image may have a different size.

**Double Page**

Source images contain two pages. For each of them the printspace will be detected and placed on new pages.

**Verify**

Check pages, without changes.

**None**

No page processing.

**10.4.3.6 IMAGE\_PREPROCESSING: Despeckle**

Add the element DESPECKLE with value 1 to activate despeckling. Despeckling runs on bitonal images only.

Despeckling removes small black noises and/or fills small white spots.

The sizes are configurable:

**REMOVE\_BLACK**

default=1

1 to activate removal of black noises, 0 to deactivate.

**MAX\_BLACK\_WIDTH**

default=7

Maximum width (pixels) of black noises that will be removed.

**MAX\_BLACK\_HEIGHT**

default=7

Maximum height (pixels) of black noises that will be removed.

**MAX\_BLACK\_PIXELS**

default=16

Maximum number of pixels in a black noise to be removed.

**FILL\_WHITE**

default=1

1 to activate filling of white spots, 0 to deactivate.

**MAX\_WHITE\_WIDTH**

default=3

Maximum width (pixels) of white spots that will be filled.

**MAX\_WHITE\_HEIGHT**

default=3

Maximum height (pixels) of white spots that will be filled.

**MAX\_WHITE\_PIXELS**

default=10

Maximum number of pixels in a white spot to be filled.

#### 10.4.3.7 IMAGE\_PREPROCESSING: Deskew

Add the element DESKEW with value 1 to activate deskewing.

Note: If "Align" or "Double Page" is selected as type for image processing, deskew is always done.

#### 10.4.3.8 IMAGE\_PREPROCESSING: Printspace

The element PRINTSPACE Definition of printspace, final page size and margin.

##### **UNIT**

use=optional

Defines the unit of all values given for attributes of printspace. Could be mm or Inch. If nothing is added, mm is used

##### **WIDTH**

use=optional

Defines the default width of the printspace. Can be used to ensure to have same width in different documents.

##### **WIDTH2**

use=optional

Defines the default width of the printspace of supplements. Can be used to ensure to have same width in different documents.

##### **HEIGHT**

use=optional

Defines the default height of the printspace. Can be used to ensure to have same height in different documents.

##### **HEIGHT2**

use=optional

Defines the default height of the printspace of supplements. Can be used to ensure to have same height in different documents.

##### **SIZE**

use=optional

Use "variable" to enable documents to have all pages having individual sizes. Use "fixed" to make them all the same size. Even if "fixed" is set, isolated pages can be set to have individual sizes manually. The default is "fixed".

##### **MARGIN**

Margins to be added to the printspace.

##### **LEFT**

use=optional

##### **RIGHT**

use=optional

## TOP

use=optional

## BOTTOM

use=optional

## PAD

use=optional

For cut areas, the background color can be defined as "dynamic", "white" or "original":

- In case of "original", no padding is made. From the original image, everything inside the blue frame is taken (final page frame).
- If "dynamic" is taken, the space between the red frame and the blue frame is filled using a special algorithm that simulates the paper background.
- If "white" is selected, the space remains white. The default is "dynamic".

### 10.4.3.9 IMAGE\_ENHANCE

Defines manipulation of images (raster operations).

## SCRIPT

use=optional

### 10.4.3.10 OCR

Add if OCR shall run. (If documents should be processed only with image processing, no OCR is needed and therefore no entries here.)

## SCRIPT

This element may have the attribute SCRIPT. It defines a special script to be used on OCR for optimization. OCR Scripts are always taken from the script folder.

Just the file title must be entered:

`SCRIPT="OCRImageEnh"`

The following scripts are available by default:

OCRDetectLanguage	Automated detection of languages beside those that are specified per document
OCREmpty	No OCR at all
OCRImageEnh	Create enhanced image before executing OCR
OCRImageEnhAnFrAnMerge	Create enhanced image before executing OCR, Merge Antiqua and Fraktur (Gothic) results
OCR-PDF	Take text from PDF input if available
OCRScript	Default OCR
OCRPlace	Property for layout elements. It stores information where (which computer) and when OCR has been performed. This helps for bug-tracking and allows to use this information in output (output usage not implemented in default cfg)

Other specialized OCR scripts might be added at any time.



## Use Hidden Text of Digital Born PDF instead of running OCR

docWizz application is able to use hidden text of PDF files. Hidden text exists in digital born PDFs or in some cases it has been added manually to PDFs of scanned images.

By taking existing hidden text into the docWizz workflow OCR is no longer necessary. Thus OCR errors can be avoided and processing is speeded up tremendously.

In case some zones do not contain hidden text OCR will be performed as usual, anyway. The same is true if the whole documents lacks hidden text.

To activate the feature please edit the project configuration and perform the following changes:

- Refer to the "Processing" element of the <project\_name>.rdy file.
- Edit (or add) the line <OCR> so that it quotes <OCR SCRIPT="OCR-PDF"/>

## NO\_SPELL\_CHECK

use=optional

Disable spellchecking - shall be used for born digital input or for already corrected text input.

## REPLACE

use=optional

Name of a replacement list to be used for automated OCR correction.

## OLDLANGUAGE

use=optional

Text contains old style of languages.

## OCR history

It should contain the action and the operator which caused the reset of OCR.

Action could be either manually (selecting from menu) or automatically (cut, resize etc. of zones).

<OCRHistory> is written in step.xml as string type.

A typical entry looks like:

*E PC-TEST-W7 dwcl.exe 15.10.2010 10:49 Technic Technic ruleRunOcr Manager after RunOCRonPage in doOCR.tcl*

E <computername>	E (meaning RQAManager location) or R (meaning RQALoader location).
dwcl.exe	The values here are the app name (it can be dwcl.exe, DWSrv RemoteQALoader or RemoteQAManager are present in History as well
<datetime>	date
<windows user> Technic	Why is the windows user doubled? It is doubled for ResetOcr action only (for other actions such as doOCR in dWSrv it is not. it should be DWCL for DWCL actions, and for dWSrv is dWSrv already )

ruleRunOcr Manager	mainly rules that are used when doing OCR or re-setting it (ComputeXMLText ResetOCR, etc)
after RunOCRonPage	This param is given for better insight, the ones like "E <computername>" are used in scripts, such as (in SetDocOnManagerPool.tcl -> ResestOcrDoneOnLoader)
in doOCR.tcl	The Script which is called when OCR is done, From Service or from Dwcl

The values differ from action to action, it is not a template filled with different info (different info means different action).

Manual or automatic. Generally automatic history entries are added by dwsrv processing (in DWCL no action may be considered to be automatic because you need a user to trigger it) one differentiation is when OCR is reset, you get the place and how it was reset (ruleRunOcr).

Usually you get the history cleared if you merge two zones (leaving just the OCR reset if computed).

#### 10.4.3.11 STRUCTURE

Add if document shall run to VerifyHierarchy. Full logical structure will be detected.

#### 10.4.3.12 PAGE\_LINKING

Add if document shall run to VerifyPageNumbers. It checks page sequences, missing pages or issues.

#### 10.4.3.13 METADATA

Add if document shall run to VerifyPagesHierarchy. Volume and Issue metadata can be collected there.

#### 10.4.3.14 SKIP\_LA

Skips layout analyze if only image processing and metadata is required without page linking or text.

#### 10.4.3.15 ZOOM\_LEVEL

Zoom factor of the image before analyse.

#### 10.4.3.16 CLIPPING

Add if document shall be processed with clipping articles.

##### PDF

use =optional

param='cfg'

type=string

Add if PDF shall be created.

Example:

```
<PDF cfg="***project_cfg***\Newspaper\pdf\PDFCfg-Clipping.xml"/>
```

##### TEXT

use=optional

Add if a xml text file shall be created on export.

#### 10.4.4 OCR\_CONFIG\_PATH

Define the configuration path of the OCR.

```
<OCR_CONFIG_PATH>***PROJECT_CFG***\Default\Default-  
OCRCfg.xml</OCR_CONFIG_PATH>
```

#### 10.4.5 LA\_CONFIG\_PATH

Custom Layout configuration in the path.

```
<LA_CONFIG_PATH>***DATA***\project-cfg\Default\Default-  
LACfg.xml</LA_CONFIG_PATH>
```

#### 10.4.6 EXPORT element

The export element describes some general export settings. Detailed export settings are defined in the project configuration. The export element may have the following elements:

<b>PDF</b>	Enables PDF creation
<b>TEXT</b>	Defines if simplified structure with text XML shall be created. Used for demonstration of verification purposes only
<b>EPUB</b>	Enables EPUB creation

## 10.5 dWSrv configuration per project

You can run different projects with, for instance, different OCR settings. As an example, one your project should run with Finereader 11 and another project should run with Finereader 12.

Those settings can be configured in dwsrv.ini (and custom-dwsrv.ini) in section [PROJECTSSELECTION].

Then you may define per services group which project group to be included or excluded (PROJECTSINCLUDE= ; PROJECTSEXCLUDE=).

```
[GROUP.A]
```

```
PROJECTSINCLUDE=<nameofprojgroup1>
```

```
[GROUP.B]
```

```
PROJECTSEXCLUDE=<nameofprojgroup2>
```

```
[GROUP.C]
```

```
PROJECTSINCLUDE=<nameofprojgroup1>
```

```
PROJECTSEXCLUDE=<nameofprojgroup2>
```

```
[PROJECTSSELECTION]
```

```
<nameofprojgroup1>=CUSTOMER,<nameofprojectfilter>,<more>
```

```
<nameofprojgroup2>=CUSTOMER,<nameofprojectfilter>,<more>
```

**Example:**

```
[PROJECTSSELECTION]
```

```
Mono=vaitos,vaitos2,kaunokirjallisuus,maantiede,kirjat,pelastakirja
```

```
PROJECTSINCLUDE=Mono
```

Both settings can be used per Group; exclude setting has priority.

The name which show in pool dialog in "Project" drop down list is used for project selection.

Can be used to separate e.g. OCR projects or similar workflow needs.

## 10.6 Implement substitution tables (long S)

Some characters are misinterpreted by the OCR. To correct these mistakes automatically there is a possibility to implement substitution tables. Those substitution tables can be implemented/called by a specific project. After generation they will not be standard for all projects. How to implement a project based substitution table, see point 3. of this chapter.

The following steps are necessary to implement a substitution table.

1. Open docWizz System Configuration, select Tab "Dictionaries" and add a new "Replacement Table".
2. Add the replacements by using the "New" button.

The new list is located in the DICT directory (LongS.tri or with the name you used).



3. To connect the replacement list to the project configuration, open the xml file from the RDY directory of your project configuration.

Add the red marked line in the picture below to the file and change the LongS to the name of the list you defined.

```
<?xml version="1.0" encoding="UTF-8"?>
<DOCUMENT ...>
  <DESCRIPTION TYPE="Monograph">
    <LANGUAGE Name="English"/>
    <LANGUAGE Name="German" Subtype="NewSpelling"/>
    <FONTTYPE>
      <FRAKTUR_ANTIQUA/>
    </FONTTYPE>
    <PROJECT>HUL</PROJECT>
  </DESCRIPTION>
  <PROCESSING PROFILE="Default" STOPAT="VerifyPageFrames">
    <IMAGE_PREPROCESSING TYPE="None">
```

```
<DESPECKLE>1</DESPECKLE>
<DESKEW>1</DESKEW>
</IMAGE_PREPROCESSING>
<OCR SCRIPT="OCRScript" REPLACE="LongS"/>
<PAGE_LINKING/>
<STRUCTURE/>
</PROCESSING>
<EXPORT TYPE="">
</EXPORT>
</DOCUMENT>
```

## 10.7 Disable/Enable functions

Configuration file to disable some functions per project either in RQA or in general.

Description for project specific settings:

...\Project-cfg\<projectname>\<projectname>-setting.xml

```
<PROJECT-SETTINGS>
    <!-- Parameter ENABLED has three possible values: 0 - disabled; 1 -
    enabled; 2 - disabled in RQA, enabled on manager only -->
    <EDITING>
        <FEATURE NAME="AllowDeletePage" ENABLED="2"/>
        <FEATURE NAME="AllowModifyPagesAfterVPN" ENABLED="2"/>
        <FEATURE NAME="AllowInsertPageAfterScan" ENABLED="1"/>
        <FEATURE NAME="AllowChangeIssueStart" ENABLED="2"/>
        <FEATURE NAME="AllowChangeLanguage" ENABLED="2"/>
        <FEATURE NAME="ComputeTimeConsumingRejectsInUI" ENABLED="0"/>
    </EDITING>
    <EDITINGTASK>
        <FEATURE NAME="VerifyStructure" ENABLED="0"/>
        <FEATURE NAME="ComputeMissingOCR" ENABLED="0"/>
        <FEATURE NAME="VerifyOCR" ENABLED="0"/>
        <FEATURE NAME="Typing" ENABLED="0"/>
        <FEATURE NAME="ReTyping" ENABLED="0"/>
        <FEATURE NAME="CompareKeying" ENABLED="0"/>
        <FEATURE NAME="VerifyMetadata" ENABLED="0"/>
        <FEATURE NAME="VerifyClipping" ENABLED="0"/>
        <FEATURE NAME="FinalQA" ENABLED="0"/>
        <FEATURE NAME="Frame" ENABLED="0"/>
        <FEATURE NAME="Attach" ENABLED="0"/>
    </EDITINGTASK>
    <PROCESSING>
        <FEATURE NAME="AddInImagesOnBackupAndRemove" ENABLED="1"/>
        <FEATURE NAME="FreePoolDataOnBackupAndRemove" ENABLED="1"/>
    </PROCESSING>
    <POOL>
        <FEATURE NAME="NoCompression" ENABLED="0"/>
        <FEATURE NAME="LowCompression" ENABLED="0"/>
        <FEATURE NAME="AllowCleanPoolWithoutOriginImage" ENABLED="1"/>
    </POOL>
    <ITEMTRACKING>
        <RETURNWHENREACHED>BuildHierarchy</RETURNWHENREACHED>
```

```

</ITEMTRACKING>
<DOUBLEKEYING>
    <ZONETYPE ENABLED="1" like="1">Headline Author</ZONETYPE>
    <TYPING_CONDITION ENABLED="0" THRESHOLD="500"
MINCHARS="5"></TYPING_CONDITION>
</DOUBLEKEYING>
<RQA>
    <LIST NAME="Create_RQA_BW" ENABLED="1" SELECT="b">
        <a>Threshold</a>
        <b>EnhanceWithMask</b>
        <c>CustomConversionFile</c>
        <d>FastLMSDT</d>
    </LIST>
</RQA>
</PROJECT-SETTINGS>

```

### 10.7.1 EDITING

#### **AllowDeletePage**

Sets possibility to delete pages in docWizz.

#### **AllowModifyPagesAfterVPN**

Set possibility to modify pages like using ShowOrigin in a job after Z-RPS.

Note: this causes recomputing analyze and may result in incomplete structures

#### **AllowChangeIssueStart**

Sets possibility to change checkboxes in Z-RPS for startIssue.

#### **AllowChangeLanguage**

Sets possibility to change language of document.

#### **AddInImagesOnBackupAndRemove:**

Tells, whether to link in-images into backups created in BackupAndRemove.

#### **ComputeTimeConsumingRejectsInUI**

Sets possibility to run time consuming steps (like ModifyPages, DetectLayoutElements, DetectPageNumbers) in docWizz GUI.

### 10.7.2 EDITINGTASK

The more specialized input and correction is necessary inside dedicated jobs within docWizz, the more difficult is it to handle all these tasks without forgetting something. As well there is no control about the isolated tasks. Also different tasks might be performed by different operators.

For that reason, tasks are configurable and applied to the documents.



**VerifyStructure**

Enable/Disable structural editing in tree view.

**ComputeMissingOCR**

Enable/Disable . This task has no manual interaction. Use the Process button to execute it. docWizz tries to read text which was not read.

**VerifyOCR**

Users corrects the OCR using textView in Z-RST

**Typing**

Enable/Disable first typing of text to be compared with OCR.

**ReTyping**

Enable/Disable second typing of text. Necessary for zones that do not provide OCR only.

**CompareKeying**

Enable/Disable evaluation and correction of mismatches of OCR and typing.

**VerifyMetadata**

Enable/Disable edit/QA on Metadata assigned to structural elements like articles.

**VerifyClipping**

Enable/Disable view/Edit Clipping.

**FinalQA**

Enable/Disable any kind of final verification by a different operator.

**Frame**

Enable/Disable - Specific to Re-Scan step, where in the frame of the re-scanned image can be adjusted.

**Attach**

Enable/disable - Specific to Re-Scan step. User can confirm the re-scan and sends the document forward.

### 10.7.3 PROCESSING

**AddInImagesOnBackupAndRemove**

Enable/Disable whether to link in-images into backups created in BackupAndRemove.

**FreePoolDataOnBackupAndRemove**

Enable/Disable to run "Free Pool Data" after generating backups.

You may use combination of **FreePoolDataOnBackupAndRemove** enabled and **AddInImagesOnBackupAndRemove** disabled.

Then backup can't be restored successfully without placing source images before in the correct in-folder.

## 10.7.4 POOL

### **NoCompression**

Refers to working images in pool - if they are to be kept uncompressed or compressed.

### **LowCompression**

Refers to the level of compression of the image in pool - NWIMG\_TIFF\_PACKBITS : NWIMG\_TIFF\_DEFLATE.

### **AllowCleanPoolWithoutOriginImage**

Refers to the reduce pool data task, to allow cleaning documents that have no IN images.

To enable this option can cause trouble if you'd like to restore those document backups later without storing the IN images at a separate location.

## 10.7.5 ITEMTRACKING

`<RETURNWHENREACHED>BuildHierarchy</RETURNWHENREACHED>`

Only useful if docWizz ItemTracking is in use.

Definition of job where the physical unit ID (physUnit) can return.

## 10.7.6 DOUBLEKEYING

`<ZONETYPE>`

The ZONETYPE node, refers to the zones on which the Double keying should be done. See Double Keying documentation for more details.

Example:

`<ZONETYPE ENABLED="1" like="1">Headline Author</ZONETYPE>`

## 10.7.7 LIST

Algorithm used to generate BW images on Loader side.

Note that BW images are different on Loader side, and they are generated when the document is "Set to Client Pool".

## 10.8 Define MODS or DC

Description for project specific settings:

...\Project-cfg\<projectname>\rdy\<name>.xml

If the parameter DC is inactive, MODS will be used. If the DC entry is active, Dublin Core (DC) will be used.

inactive	active
<METADATA> <!--DC/--> </METADATA>	<METADATA> <DC/> </METADATA>

Add metadata action overwrites MODS entries:

In Z-RST add partname, partnumber, subtitle and type in Hierarchy, section:titleinfo section in data view. Click "add metadata" on doc node in tree view. Some mods metadata entries are emptied, some are kept.

DELETED: partname, subtitle and type

KEPT: partnumber

Certain elements are overwritten, even when an existing entry exists. The ones that are kept are the ones that can be found in OutlineTree. (on a newspaper, you will have more info not overwritten), so it depends on the available info in tree, the other info is overwritten by the AddMetadata script.

Metadata can be called in CustomPreProcessExport procedure in project-export.tcl and it depends on its configuration.

## 10.9 Project cfg to overwrite Default cfg

For some projects it is useful that user defined project configuration overwrites the settings from Default configuration.

Description for settings:

...\config\PVSCFG\Newspaper-DW.xml

When <standalone="true" the configuration will not be merged.

Example:

In project-cfg on a frame standalone="true" only project cfgs will be available in dwcl.

```
...
<STEP name="VerifyHierarchy">
    <FRAME ID="OUTLINETREEFRAME"><standalone="true">
...

```

If "standalone" is missing configuration will be merged with default cfg.

This setting is especially useful in large environments or environments which needs customization



```
<?xml version="1.0" encoding="UTF-8" ?>
- <CONFIG>
+ <STEP name="Exported">
+ <STEP name="Re-Scan">
+ <STEP name="Scan">
- <STEP name="VerifyHierarchy">
+ <FRAME ID="ALLFRAMES">
+ <FRAME ID="DETAILIMAGEFRAME" />
+ <FRAME ID="DETAILINDEXFRAME" />
+ <FRAME ID="DETAILTREEFRAME">
+ <FRAME ID="OUTLINELISTFRAME">
+ <FRAME ID="OUTLINETEXTFRAME">
+ <FRAME ID="OUTLINETREEFRAME" <standalone="true">
</STEP>
+ <STEP name="VerifyLayoutElements">
+ <STEP name="VerifyPageFrames">
+ <STEP name="VerifyPageNumbers">
+ <STEP name="VerifyPages">
+ <STEP name="VerifyPagesHierarchy">
</CONFIG>
```

## 10.10Stitching

To enable stitching functionality, please add <STITCHING> node into <PROCESSING> node to the RDY file of your project configuration.

```
...\project-cfg\<your_project>\rdy\<name>.xml
```

Description for project specific settings:

```
...\Project-cfg\<projectname>\rdy\<name>.xml
```

Example RDY file:

```
<?xml version="1.0" encoding="UTF-8"?>
<DOCUMENT xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="docWizz_Import.xsd">
  <DESCRIPTION TYPE="Monograph">
    <LANGUAGE Name="English"/>
    <FONTTYPE>
      <FRAKTUR_ANTIQUA/>
    </FONTTYPE>
    <PROJECT>Monograph</PROJECT>
  </DESCRIPTION>
  <PROCESSING PROFILE="Default" STOPAT="VerifyPageFrames">
    <IMAGE_PREPROCESSING TYPE="none" PAGESPERSCAN="1">
      <PRINTSPACE UNIT="mm">
        <MARGIN LEFT="2.0" TOP="2.0" RIGHT="2.0" BOTTOM="2.0"
PAD="none"/>
      </PRINTSPACE>
      <DESPECKLE>0</DESPECKLE>
      <DESKEW>0</DESKEW>
    </IMAGE_PREPROCESSING>
    <PAGE_LINKING/>
    <OCR/>
    <METADATA/>
    <STRUCTURE/>
    <STITCHING/>
  </PROCESSING>
  <EXPORT TYPE="">
    <PDF/>
  </EXPORT>
</DOCUMENT>
```

Furthermore, for enabling the stitching button you need to set the STITCH property to 1 in client-docWizz-DW.ini, in [DWSCAN] section:

```
...\config\<client>-docWizz-dw.INI
```

[SCAN]

...

STITCH=1

Is important that documents will be imported and not just loaded in Scan step. Special thumbnails are created during import process - one for each corner of the images to be stitched.

See also [docWizz\\_Usermanual](#), chapter [Interactive Verify Jobs\Scan\Stitching](#).

## 10.11 PDF / OCR comparison

In order to get a more accurate text from documents imported from PDF, the PDF's embedded text and the OCR resulted with ABBYY are compared so the best version of the text can be used.

For the option to compare the text, in the OCRCfg.xml file the OCR-PDFComp.tcl script should be enabled.

```
<SET name="OCRScripts" enable="1">
<SET name="PATH" enable="1">***DATA***\script\OCR-PDFComp.tcl </SET>
</SET>
```

Description for project specific settings:

```
...\Project-cfg\<projectname>\rdy\<name>.xml
```

Sample RDY file:

```
<?xml version="1.0" encoding="UTF-8"?>
<DOCUMENT xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Metae_Import.xsd">
  <DESCRIPTION TYPE="Newspaper">
    <LANGUAGE Name="English"/>
    <FONTTYPE>
      <FRAKTUR_ANTIQUA/>
    </FONTTYPE>
  </DESCRIPTION>
  <PROCESSING PROFILE="PDF Import" STOPAT="VerifyLayoutElements">
    <!--IMAGE_PREPROCESSING TYPE="Align">
      <DESPECKLE>0</DESPECKLE>
      <DESKEW>0</DESKEW>
      <PRINTSPACE UNIT="mm">
        <MARGIN LEFT="2.0" TOP="2.0" RIGHT="2.0"
BOTTOM="2.0"/>
      </PRINTSPACE>
      <DPF_ALGORITHM>NormalFrameDetection</DPF_ALGORITHM>
    </IMAGE_PREPROCESSING-->
    <PAGE_LINKING/>
    <OCR SCRIPT="OCR-PDFComp"/>
    <METADATA>
      <!--DC/-->
    </METADATA>
    <STRUCTURE/>
  </PROCESSING>
  <!--OCR_CONFIG_PATH>***PROJECT_CFG***\Default\Default-
OCRCfg.xml</OCR_CONFIG_PATH-->
```

```
<LA_CONFIG_PATH>***PROJECT_CFG***\Default\Default-  
LACfg.xml</LA_CONFIG_PATH>  
  <EXPORT TYPE="METaE Default">  
    <PDF/>  
  </EXPORT>  
</DOCUMENT>
```



## 10.12 Keep selection in list view

To prevent long loading times, e.g. for reject list or "All items" list for very large documents, the selection in ListView can be kept when opening a new document.

It can be configured if previous list selection is kept or will be reset.

Add configuration to reset List view combo selection (default is true):

```
...\config\PVSCFG\<docType>-dw.xml
```

```
<CONFIG>
```

```
...
```

```
<STEP name="VerifyHierarchy">
```

```
...
```

```
<FRAME ID="OUTLINELISTFRAME">
```

```
<RESETCURRENTSELECTEDELEMENT>0</RESETCURRENTSELECTEDELEMENT>
```

```
</FRAME>
```

```
...
```

```
</STEP>
```

```
...
```

If line `<RESETCURRENTSELECTEDELEMENT>0</RESETCURRENTSELECTEDELEMENT>` is missing, the default value will be used.

## 10.13 Black & White algorithm usage

This is a brief description on how to choose and configure a specific BW conversion in order to have a better layout analysis. The analysis run currently on BW images in docWizz, this is why a good BW image (without noises and with all the relevant information preserved) is very important for a good zoning. Originally a simple threshold algorithm was used, but this proved not to be enough for very different material. Low contrast material became mostly a white page, dark images became full of noises and big areas of text became a big black box. Currently several different algorithms were developed. The user has the chance to choose from different BW algorithm.

### How to configure

From version 6.6 an option to configure layout analyze similar with OCR was added via a specific configuration file. A link to the configuration file can be set into rdy file, so that each individual document can behave different (assuming in general the material is ok, but due to some reasons, you could have an exception on one document which was scanned lighter). Sample of usage, inside rdy file:

```
<DOCUMENT ...>

.....

<LA_CONFIG_PATH>***DATA***\project-cfg\Default\Default-
LACfg.xml</LA_CONFIG_PATH>

.....

</DOCUMENT>
```

Inside configuration file, there is a section `<LIST name="Create_BW"...>` that defines which algorithm to use for BW conversion. This list can be extended, but is better not to add too many options, to avoid confusion and a very difficult choice. The list contains available algorithms, `ENABLE` shall be set to 1 if you want to use other algorithm than classical threshold and also `SELECT` shall point to one of the algorithms from the list. For example if we want to choose `Local Maximum Standard Deviation Thresholding`, fast version (selection "a") the setting file shall look like:

```
<LAcfg version="1.0">

  <SETTING>

    <LIST NAME="Create_BW" ENABLED="1" SELECT="a">

      <!-- new in 6.6 - BW for LA alg fast (zoomed) 201 - Fast Local
Maximum Standard Deviation Thresholding - default, best to use-->

      <a>BW::Fast_LMSDT $page</a>

      <!-- new in 6.6 - BW for LA alg 201 - Local Maximum Standard
Deviation Thresholding - slowest and a lot of memory - good for small images
as number of pixels, else Fast_LMSDT shall be used -->

      <b>BW::LMSDT $page</b>

      <!-- best available in 6.5 - BW for LA alg 106 - OTSU Adaptive
threshold - slower and a lot of memory-->

      <c>BW::OSTUAdaptive $page</c>

      <!-- used for light images in 6.4 - BW for LA old enhance alg-->

      <d>BW::OldEnhance $page</d>
```

```

        <!--best available in 6.4 version create of the BW in import that use
different parameters-->
        <e>BW::Standard64 $page</e>

        <!--don't use scripts do the threshold transformation from code same
like enabled = 0-->
        <code></code>

    </LIST>
.....
    </SETTING>
</LAcfg>

```

### How to decide best of

From our tests we recommend for best results in case material vary a lot (both dark/light images appear, variable threshold on same page, etc) to choose a or b (Local Maximum Standard Deviation Thresholding), fast version (a) in case the images are big as number of pixels, or version (b) for cases where images are not very big as number of pixels (f.e books, scanned at normal 300 dpi). For very good quality images, good contrast and when you are interested in high processing speed, then you should disable the setting (`ENABLE = "0"`) and use regular threshold. The other algorithms are older once and we recommend to be used only in case the older versions of docWizz are used, or you simply want to have same approach like before since it brings very good results and usually older algorithms uses less resources.

## 10.14 Enable/Disable certain cleaning algorithms

In order to improve the analyze results for good quality materials a new parameter was added into LACfg file. For old material is needed an aggressive noise removal approach to clean areas outside detected print space. This algorithm can be too aggressive and remove more than is needed. For high quality materials (as images originated from born digital PDFs, or very high quality scans) noises are not present and in fact this algorithm remove real content in particular cases. In order to disable this algorithm, you need to set inside LACfg.xml file following setting:

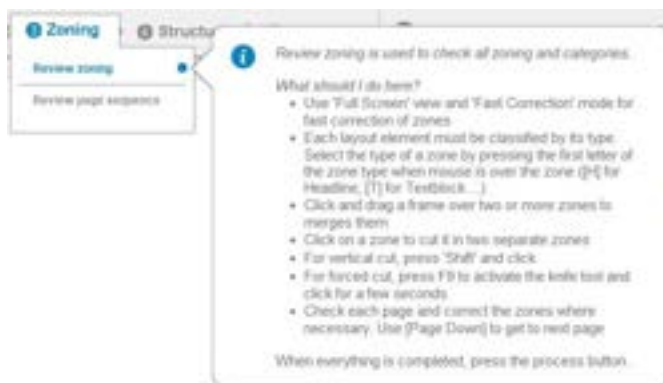
```
<LA_NEWSPAPER   ENABLED="1">  
.....  
  <NON_AGRESSIVE_MARGIN_NOISE_REMOVAL>1</NON_AGRESSIVE_MARGIN_NOISE_REMOVAL>  
.....  
</LA_NEWSPAPER>
```

By default, since dW operates more with historical material, this setting is not present and aggressive noise removal is applied.

Please enable this for very high quality scans, to improve results.

## 10.15 Edit info-boxes for tasks

To edit the tooltips for tasks go to `..\project-cfg\<projectname>\documents\InfoBox` folder and open the appropriate \*.rtf file.



You can edit \*.rtf files e.g. in Microsoft Wordpad (recommended because of smaller size than Microsoft Word files).

## 10.16 Expand tools automatically

Toolbar buttons are often grouped (see a small black triangle on the right of it). To expand use right click or stay with left mouse button for a few seconds over the tool button.



Go to `<customer>-docWizz-dW.ini` to configure if the toolbar should expand automatically.

```
[docWizz]
TOOLBARAUTOEXPAND=1
```

## 10.17 Change colors of frames for Cropping step

In the Cropping step several colors are used to identify different frame types.

As default there are red (regular pages), orange (alternative pages) and blue for individual pages.

To change colors go to `docWizz-dW.ini` and set the HEX values for the colors you like:

```
[docWizz]
IMG_BGR_COLOR=b8b8b8
DOT_FRAME_LINE=1
REG_FRAME_COLOR=0000ff
ALT_FRAME_COLOR=0099ff
```

`IMG_BGR_COLOR` is the background color of the image area (grey per default)

`DOT_FRAME_LINE` sets if the frame is dotted (=1) or plain (=0)

`REG_FRAME_COLOR` sets the color for regular frames (red per default)

`ALT_FRAME_COLOR` sets the color for alternative frames (orange per default)

Configuration of final page preview:

`FINAL_PAGE_OUTLINE_COLOR=B8B8B8` - color of the area that will be cut from page

`FINAL_PAGE_OUTLINE_ALPHA=178` - transparency of the area that will be cut from page

`FINAL_PAGE_DYNMARGIN_COLOR=e6dcbe` - color of the margins (when the margins are set "dynamic")

`FINAL_PAGE_DYNMARGIN_ALPHA=178` - transparency of the margins (when the margins are set "dynamic")

`FINAL_PAGE_BW_MARGIN_ALPHA=178` - transparency of the margins (when the margins are set "black" or "white")

## 10.18Customize Index view

There are 3 steps to add a custom index view in docWizz:

Define the elements from the custom index view

In `<client-name>-docWizz-DW.ini` file, create a new node with the title:

```
[<client-name>-<Task>.MASKE]
```

where:

`<client-name>` - is the name of the client

`<Task>` - is the task name where the custom index view will be available (!! Use old name of the tasks!!)

Example of a custom index view added in Review structure and text task (client-name=QA):

```
[QA-VerifyHierarchy.MASKE]
```

```
BTN2=K,N=<,P=-20:150:270:250,R,Y=131152
```

```
BTN1=K,N=>,P=-20:350:260:450,R,Y=131150
```

```
BTN4=K,N=<<,P=-20:550:260:650,R,Y=196688
```

```
BTN3=K,N=>>,P=-20:750:270:850,R,Y=196686
```

```
TOGGLE_TOC=B,D=0,K,N=TOC,P=-50:950:460:1050,R
```

### Define the actions of the elements

To define the actions of the elements, create a new `.dwx` file inside the `\config` folder with the desired actions for the added buttons and name it `<client-name>-<Task>.dwx`

Ex: for the node from the previous example, the name will be: `QA-VerifyHierarchy.DWX`.

### Define the name of the custom index view

To define the name of the index view, in `<client-name>-docWizz-DW.ini` file, create a new node with the title `[CLIENT_INDEX]`, with the entry `"Name=<client-name>"`.

Example:

```
[CLIENT_INDEX]
```

```
Name=QA
```

The custom index view will be displayed only in the configured task, last tab from the right side:



## 10.19 Set LA quality for very close zones

Use LACfg.xml. This parameter overwrites the default value (27 pixels) of a white vertical separator width.

Below a sample of the configuration file (see bold marker for MIN\_VWB\_WIDTH setting):

```
<?xml version="1.0" encoding="UTF-8"?>
<LAcfg version="1.0" uid="1" CRC="FBE763099D8E8B57B10AA49E463B4912">
  <SETTING>
    <LIST NAME="Create_BW" ENABLED="1" SELECT="Fast_LMSDT">
      <Default>BW::CreateSpecialBWforLA BW::Fast_LMSDT
$page</Default>
      <Fast_LMSDT>BW::Fast_LMSDT $page</Fast_LMSDT>
      <LMSDT>BW::LMSDT $page</LMSDT>
      <OTSUAdaptive>BW::OTSUAdaptive $page</OTSUAdaptive>
      <OldEnhance64>BW::OldEnhance $page</OldEnhance64>
      <StandardImport64>BW::StandardImport64
$page</StandardImport64>
    </LIST>
    <DESPECKLE REMOVE_BLACK="1" MAX_BLACK_WIDTH="7"
MAX_BLACK_HEIGHT="7" MAX_BLACK_PIXELS="16" FILL_WHITE="1" MAX_WHITE_WIDTH="3"
MAX_WHITE_HEIGHT="3" MAX_WHITE_PIXELS="10"/>
    <LA_NEWSPAPER ENABLED="1">
      <RECT_SHAPE ENABLED="1">Table Illustration
Advertisement</RECT_SHAPE>
      <MERGE_CONSECUTIVE_HEADLINES>1</MERGE_CONSECUTIVE_HEADLINES>
      <DETECT_ORIENTATION>0</DETECT_ORIENTATION>
      <SPLIT_PAGE_FOR_OCR>0</SPLIT_PAGE_FOR_OCR>
      <MIN_VWB_WIDTH>20</MIN_VWB_WIDTH>
    </LA_NEWSPAPER>
    <BIGBLACKBORDERS ENABLED="0"/>
  </SETTING>
</CFGGUI/>
</LAcfg>
```

## 10.20 Divided Document Pool

### Purpose

Depending on the operation system, the maximum capacity of volumes is physically limited at about 15 Terabyte (depending on storage type).

If a project needs more capacity there is the possibility to segment the docWizz document pool and process with several volumes with separate shares.

### Maximum amount of shares

The amount of shares is not limited in theory. In practice we recommend 5 shares at all.

### Implementation

custom-docWizz-glb.ini

Note: Configuration for pool is only read on start of program. If changes are done all programs must be restarted!

### Add the additional pools

[PATH]

```
POOL=\\name-of-PC\customer\docWizz\POOL
POOL1=\\name-of-PC\customer\docWizz\POOL1
.
.
POOLXX=\\name-of-PC\customer\docWizz\POOLXX
```

After this step the pool is activated immediately!

[POOL.SHARES]

```
0=***POOL***
1=***POOL1***
.
.
XX=***POOLXX***
```

The ID of the pool share must be unique. It is inserted into the database "Main" table in column "P\_LOCATION".

As soon as multiple pools are configured, the created documents will be balanced over the available pools by the calculated or defined ratio.

Note: There is no feature that supports automatic movements of documents from one share to another if one pool runs out of space.



## Parameters

Sample:

```
0=***POOL***,1,T
1=***POOL1***,3,T
index=path,size,unit
```

`index`

An integer. Must never be changed on add/remove new shares

`path`

The path, where the pool is located, using macro definition

`size`

Optional, size of volume to be used for this pool share.

**Note:** this value is used just for computing load balance. Real used size is not computed. If parameter is missing, size of entire volume is used. If parameter is "0", no further document shall be placed on this share.

`unit`

Unit to be used for size: M,G,T (Bytes). Just place a single character. If the unit is missing, the default value M is used.

In regular cases there is no need to add size and unit as parameters. This way it is assumed the complete share shall be used for pool data and the system guarantees automatic balancing.

### Documents are placed to balance total disk space on share divided by number of documents on share

ShareIndex (n) for new doc is where  $\max(\text{shareSize}(n)/\text{NumberOfDocsOn}(n))$ .

When adding a second share while first share exists, all new documents will be added to second share, unless ratio is lower than on already existing share.

**Note:** Remove share - Always empty the pool(s) first so that they do not contain any documents until removing the pool share! The best way is to set the size for the share to be removed to 0. Then no new document will be placed here. After completion and deletion of all documents that were located on the share, it will just contain empty folders and can be removed from configuration and disconnected or deleted.

### Example 1:

```
[POOL.SHARES]
0=***POOL***,1,T
1=***POOL1***,3,T
```

The ratio is 1 : 3 in this case. The system will automatically ensure to have the number of documents always in ratio of 1:3 on the shares.

Note: The ratio ensure this allocation on long term and usually is not respected right after adding a new empty share, because at that moment the shares are not balanced according to the provided percent. In our sample, if POOL1 is new added, this will have 0 documents, while the other (POOL) have for example 10 documents. Until the ratio from first sample in reach (1:3 - means POOL1 will have 30 documents) all new documents will be added to the new location (POOL1). Only after ratio is reached (POOL - 10 documents, POOL1 - 30 documents) the documents will be added into this ratio - one to POOL, three to POOL1. There are other situations when the pool may became unbalanced, for example when deleting documents - the system will try first to balance the pool share and then new documents will be allocated according to ratio definition.

### Example 2:

```
[POOL.SHARES]
```

```
0=***POOL***,1,T
```

```
1=***POOL1***,0
```

The ratio is 1 : 0 in this case. In this example ALL new documents go to pool 0.

Documents in Pool 1 are processed regularly till they are finished.

This can be used to move data during production process is executed.

### Adding a new share in an existing environment

Whenever you want to add a new share in an existing environment, it is recommended to limit its size in configuration to 50% of the real size. This ensures a better load balancing for the first time. After a while it may be stepwise increase to reach 100 percent. A typical time range to increase the size from 50% to 100% should be two month. This should be done because of the risk to overload the new share while the other share(s) are idle."

## 10.21 Including IgnoreTypes.dsl to a project

### The "dsl" folder

When a new project is created, a .dsd file is also created inside the "<ProjectName>\dsl" folder, named after the project and docType used for that project: <ProjectName>-<docType>.dsd. Inside this file, other configuration files needed for that document type are included.

When new zones / pages / structures are added to the project, a new file is created in the same folder: <ProjectName>-<docType>-dwCfg.dsl. Inside this file are stored all the new defined types.

The IgnoreTypes files will also be stored in the same folder.

### The IgnoreTypes.dsl file

The file is already created and used for any new project configuration. Among the configuration files that are included for the docType is also METAe-IgnoreTypes.dsl from \*\*\*PROJECT\_CFG\*\*\*\Default\dsl. It contains some classes that are ignored by default in all projects.

So, in order to apply the ignored classes only in the current project configuration and not to alter any other, the METAe-IgnoreTypes.dsl file has to be copied from \*\*\*PROJECT\_CFG\*\*\*\Default\dsl to the \dsl subfolder from current project, and renamed to "<ProjectName>-IgnoreTypes.dsl".

Inside the IgnoreTypes file, the ignored classes are grouped by docWizz tasks (the old naming!).

After the classes that need to be ignored are decided, for each one of them a new node has to be created inside each task node.

Example on how to ignore "Subheadline" zone:

```
<Class Name="Subheadline" BaseClass="" Virtual="true"/>
```

Example on how to ignore "EdgePage" page type:

```
<Class Name="EdgePage" BaseClass="" Virtual="true"/>
```

Example on how to ignore "StatementSection" structure:

```
<Class Name="StatementSection" BaseClass="" Virtual="true"/>
```

The page and zone types that need to be ignored have to be added inside each task node, but for structures, it's only necessary to be added inside the last node: <Step Name="VerifyHierarchy" BaseStep="VerifyPagesHierarchy"> (Review structure and text).

### Recommendations

Because all the changes are done manually and there are no validation rules, you need to make sure that everything that was added to the file is correct. Any mistake can lead to errors or even crashes while processing.

- Don't make any change while having Config Tool opened. In order to avoid manual changes being discarded, it's best to have Control Center closed before starting to edit the files.
- Instead of typing a new node entirely, you should copy an existing one, paste it at the end of the standard ignored classes and just modify the Name attribute.

Add comments to separate and group the new added classes:

```
<Step Name="VerifyHierarchy" BaseStep="VerifyPageHierarchy">
  <Classes>
    <!-- Ignored zones for <myProject> -->
    <Class Name="Advertisement" BaseClass="" Virtual="true"/>
    <Class Name="Author" BaseClass="" Virtual="true"/>
    <Class Name="Footnote" BaseClass="" Virtual="true"/>
    <Class Name="ItemCaption" BaseClass="" Virtual="true"/>
    <Class Name="Publishing_Stat" BaseClass="" Virtual="true"/>
    <Class Name="Subheadline" BaseClass="" Virtual="true"/>

    <!-- Ignored page types for <myProject> -->
    <Class Name="EdgePage" BaseClass="" Virtual="true"/>
    <Class Name="SpinePage" BaseClass="" Virtual="true"/>

    <!-- Ignored structures for <myProject> -->
    <Class Name="StatementSection" BaseClass="" Virtual="true"/>
    <Class Name="Contribution" BaseClass="" Virtual="true"/>
    <Class Name="List" BaseClass="" Virtual="true"/>
    <Class Name="Group" BaseClass="" Virtual="true"/>
    <Class Name="Aidax" BaseClass="" Virtual="true"/>
    <Class Name="Postface" BaseClass="" Virtual="true"/>
    <Class Name="BookReview" BaseClass="" Virtual="true"/>
    <Class Name="Debate" BaseClass="" Virtual="true"/>
    <Class Name="ProductionNote" BaseClass="" Virtual="true"/>
    <Class Name="Poem" BaseClass="" Virtual="true"/>
    <Class Name="Dedication" BaseClass="" Virtual="true"/>
    <Class Name="Corrections" BaseClass="" Virtual="true"/>
    <Class Name="ListOfPublications" BaseClass="" Virtual="true"/>
    <Class Name="BastardTitleSection" BaseClass="" Virtual="true"/>
    <Class Name="Frontispiece" BaseClass="" Virtual="true"/>
    <Class Name="CoverSection" BaseClass="" Virtual="true"/>
  </Classes>
</Step>
```

- Make a test document before using the ignoreTypes file in order to see the zones that you should add. Get the document to "Review structure and text" task and, using "Change type" or "Group to..." actions, check the available zone / page / structure types.
- Make sure you use the correct names of the classes you want to ignore.
- After adding all the classes that will be ignored, the default ones have to be deleted. They are already in Default project configuration, so they will be ignored from the start.

## Including the IgnoreTypes file

After making sure the file is correct, it needs to be included in the .dsd file of the project. Open the <ProjectName>-<docType>.dsd file and add a new node:

```
<Import File="***PROJECT_CFG***\\<ProjectName>\\dsl\\<ProjectName>-
IgnoreTypes.dsl"/>
```

**VERY IMPORTANT!!!** Make sure that the node is always the last node in the .dsd file. If new zone / page / structure types are needed for that project, first add the new types, then include the IgnoreTypes file.

## Project configuration in Config Tool

If #1 from Recommendation is respected, the project configuration will remain editable with Config Tool. But in order to keep it editable, each time a change has to be made in the IgnoreTypes file, Control Center has to be closed.

# 11 Special Tasks

A task is a special operation added to regular tasks (Review zoning, Review issues, Review structure and text or others).

A task has two functionalities:

- pre-check
- execute

Please be aware of the renamed steps and tasks and the "old" names in scripts and configuration files.

## 11.1 Scope and details of Tasks

### **IMPORTIMAGES**

Scope: Imports images

Details: Imports from IN share via autoimport

Notes: When a folder has the files "cloaked.rdy" and "cloaked.wrk", the auto-import tasks will not verify this folder and its subfolders for new documents. This can help speed up the task.

### **MNGRCOLLECTDATA**

Scope: collect documents from manager to send on Loader

Details: zip document data

### **MNGRUPDATEDATA**

Scope: Update documents on manager with loader data

Details: unzip document data/ merge data

### **UPDATEPOOLSTATUSMNGR**

Scope: Update Pool Status of RQA documents on Manager

Details: Updates the status of manager documents with the status info from Loader side. Used for pool status interchange on RQA systems

### **LDRCOLLECTDATA**

Scope: collect documents from loader side to send back on manager

Details: zip document data

### **LDRUPDATEDATA**

Scope: Update existing documents/ add new documents on loader

Details: Updates existing documents on Loader with manager information/ add new documents on loader pool

### **UPDATEPOOLSTATUSLDR**

Scope: Update Pool Status of RQA documents on Loader

Details: Updates the status of loader documents with the status info from Manager side

### **MANUALTRSFTOCLIENT**

Scope: Transfer documents offline to Loader side

Details:Used for RQA environments that do not have a direct FTP link, by creating batches of documents to be sent via HDD  
to Loader ( see CDI environment )

### **MANUALTRSFTORESQA**

Scope:Transfer documents offline back to Manager side

Details:Used for RQA environments that do not have a direct FTP link, by creating batches of documents to be sent via HDD  
to Manager (see CDI environment)

### **WAKEUPCOMPUTERS**

Scope: Turn on Computers with DWsrv's installed

Details: Turn on Computers with DWsrv's installed if the ratio is below of the configured value -  
production purpose

### **BACKUPCONFIGS**

Scope: Backup configuration files

Details: backs up configuration files

### **EXTRATASKS**

Scope: Run extratasks

Details: manually or automated generated extratasks that will be processed by DWSrv's ( e.g  
Document Validation extratasks)

### **EXTRATASKSWITHOCR**

Scope: Run Extratasks with OCR

Details: manually or automated generated extratask with ocr will be processed by DWSrv's

### **CLEANPOOL**

Scope: Free Pool Data

Details: Cleans pool data when document's status is changed to FreePoolData, and sets on  
completion the status to Reduced Pool Data.

### **RESTOREPOOL**

Scope: Restore Pool Data

Details: Restore previously reduced documents to their original pool data.

### **RESTORE**

Scope: Restore backed-up document

Details: Restores the document from it's dwb to pool document.

### **FTPPRIO**

Scope: Create the FTPPrio file

Details: To prioritize transfer of RQA documents.

### **DOCPRIO**

Scope: Create Grouped Prio document list

Details: To compute document priority based on jobs

## 11.2 ExtraTasks

A mechanism of "ExtraTasks" is available in the docWizz configuration. This should be used to handle any kind of extra processing that is not in scope of regular docWizz progress. An example is recreation of PDF files based on existing output.

Below the "work" folder there is a folder "ExtraTasks". Inside this folder any script having \*.tcl as extension can be placed.

docWizz services have configured a task called "ExtraTask", which looks into the folder and executes a script, if available and not locked by another instance (similar to remoteQA tasks). After execution, the file will be deleted. If an error occurs, the file will be rename to ".err".

To generate such TCL files, a sample script is available (Script/Samples/Sample-GeneratePDFTasks.TCL). You may let it run on an existing output folder and it will create commands for each METS file that is found below. Now docWizz Services will execute recreation. You may use priority settings in Control Center as for any other task.

Extratask allows to use a subfolder. This can be configured as separate task. (e.g. task using OCR, which shall run on a dedicated computer)

## 11.3 BackupConfig Task

- On each cfg update verify if a cfg back-up is made (look inside script to see where backups are made). Try to do this by forcing an update. (change a configuration file e.g. \*.ini;\*.tcl,\*.xml)
- Try to backup config using Backup Config Button in DWControlCenter
- In backup folder check for backup files and if they are the modified ones.
- If no backup of configurations was done, the first backup shall contain all files that were configured in docWizz-glbl.ini to be backed up.
- If back up was done, the new backup folder shall contain only the modified files, different from previous backup.

### docWizz-dw.ini:

```
[PROCESS]

TASKS=....., BACKUPCONFIGS

[BACKUPCONFIGS.TASK]
PROC=BackupConfigs
DELAY=100
TIME=10:00:00
TIMETYPE=0
VALIDATION_PROC=NoOfBackupConfigs
LOCKED=1
SCRIPT_FILE=***DATA***\SCRIPT\task\backupconfigs.tcl
```

### docWizz-glbl.ini:

```
[BACKUPCONFIG]
Folders=Config,Script
DestinationFolder=***DATA***\BkpCfg
Extensions=ini,tcl,xml
IgnoredFiles=Config\docWizz-glbl.ini
```

### ..\script\tasks:

backupconfigs.tcl

## 11.4 Task: ExtrataskWithOCR

The task shall be configured to run on first instances only.



## 11.5 Reduce-, Restore Pool data

Reduce functionality is used to free space on pool storage.

For reducing storage space in pool, temporary images could be deleted (also cropped images created after MP) and restored if necessary.

- Functionality needs manual actions, just on demand. This is not initialized by workflow dependencies.
- Only administrators can perform the actions while have high impact on pool.
- Tasks must be configured for services. (CCS additional)
- The OnProcess button is disabled if current document is in Restore Pool Data or Reduced Pool Data or Free Pool Data status.
- An image could be restored (e.g. with "Document open") if necessary "on the fly". This will last some time and the user has to wait until document is restored again.
- For safety reasons source images will not be deleted for those pages where the source IN data images are not available at initial path. Only thumbnail images are deleted in this case.
- For restoring the source images must exist in the correct folder (e. g. the IN folder).

In docWizz-dW.ini file, **RESTOREPOOL**, **CLEANPOOL** tasks must be enabled here:

```
[PROCESS]
```

```
TASKS= ...RESTOREPOOL,CLEANPOOL, ...
```

The task **CLEANPOOL** cleans pool data when document status is changed to Free Pool Data, and on completion it sets the status to Reduced Pool Data.

The task **RESTOREPOOL** restores previously reduced documents to their original pool data.

Set time in for the **CLEANPOOL** task in <customer>-docWizz-dw.ini file:

```
[CLEANPOOL.TASK]
```

```
PROC=TaskCleanPool
```

```
DELAY=300
```

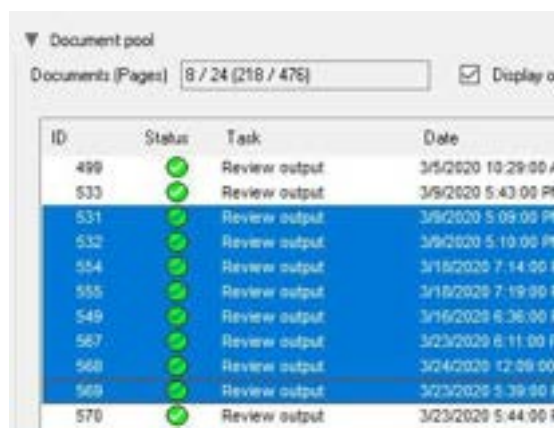
```
TIME=10:00:00
```

```
TIMETYPE=0
```

```
...
```

The manual handling is done in pool dialog and change status dialog.

Select one or more documents:



ID	Status	Task	Date
499	✓	Review output	3/5/2020 10:29:00
533	✓	Review output	3/9/2020 5:43:00 PM
531	✓	Review output	3/9/2020 5:09:00 PM
532	✓	Review output	3/9/2020 5:10:00 PM
554	✓	Review output	3/18/2020 7:14:00
555	✓	Review output	3/18/2020 7:19:00
549	✓	Review output	3/16/2020 6:36:00
567	✓	Review output	3/23/2020 6:11:00
568	✓	Review output	3/24/2020 12:09:00
569	✓	Review output	3/23/2020 3:39:00
570	✓	Review output	3/23/2020 5:44:00

Click Status/Labels button:

Project: [dropdown]  
 Task: [dropdown]  
 Subtask: [dropdown]  
 Status: [dropdown]  
 RQA Client: [dropdown]  
 Custom: [dropdown]  
 ID: [text field]  
 Label: [dropdown]  
 Buttons: Route, Delete, Status/Labels, Backup, Set Priority, Restore

Click FreePoolData entry to reduce storage space in pool:

Changes Status and/or Labels for selected documents  
 Status: [dropdown menu open with 'Free Pool Data' selected]  
 Label: [text field]  
 Priority: [text field]  
 Enter your comment: [text area]  
 Comments history: [text area]  
 Comment: [text area]  
 Buttons: OK, Cancel, Edit Labels, Add, Copy, Remove, Show all

In pool view different icons show status of documents:

150371		Exported	01/04/2011 14:29:00	Newspaper	40	86	11:30
150374		Exported	01/04/2011 14:29:00	Newspaper	20	86	11:30
150377		Exported	01/04/2011 14:30:00	Newspaper	56	86	11:30
150378		Exported	01/04/2011 14:30:00	Newspaper	56	86	11:30
150379		Exported	01/04/2011 14:28:00	Newspaper	60	86	11:30
150380		Exported	01/04/2011 14:30:00	Newspaper	52	86	11:30
150381		Exported	01/04/2011 14:28:00	Newspaper	60	86	11:30
150382		Exported	01/04/2011 14:30:00	Newspaper	60	86	11:30
150383		Exported	01/04/2011 14:28:00	Newspaper	60	86	11:30
150384		Exported	01/04/2011 14:28:00	Newspaper	60	86	11:30
150346		Exported	01/04/2011 14:28:00	Newspaper	28	86	11:30
150386		Exported	01/04/2011 14:28:00	Newspaper	28	86	11:30
150389		Exported	01/04/2011 14:28:00	Newspaper	40	86	11:30
150372		Exported	01/04/2011 14:29:00	Newspaper	48	86	11:30
150375		Exported	01/04/2011 14:29:00	Newspaper	48	86	11:30
150385		Exported	01/04/2011 14:28:00	Newspaper	60	86	11:30
150394		Exported	01/04/2011 14:28:00	Newspaper	60	86	11:30
150396		Exported	01/04/2011 14:28:00	Newspaper	52	86	11:30
150397		Exported	01/04/2011 14:28:00	Newspaper	53	86	11:30
150398		Exported	01/04/2011 14:28:00	Newspaper	56	86	11:30

- The icon shows already reduced pool data. For colored documents about 90% of data can be removed.

Following files will be removed:

All temporary files b/w images

Lowres images

Cropped/aligned images (if they are not changed manually)

RQA images (always)

<jobname>.zip (from non-interactive jobs)



Following files (all non restorable ones) will remain:

<jobname>.zip (from interactive jobs)

ID.xml

rescan images

deskewed images

- The  icon shows document to be reduced.
- The  icon shows restored pool data. Restores all temporary images data, as were existing before. Uses the task RESTOREPOOL. The document status is set back to the status that was set before the document was sent to reduce data status.

## 11.6 ComputeMissingOCR

ComputeMissingOCR is a hidden editing task, an extratask is created, that will compute the missing OCR of the document, then at the end it will be automatically put into the next editing task. Users should not open documents in ComputeMissingOCR editing task.

In ReTyping for example a headline zone (first entry in keying list) may look that there is no zoom on right hand side image view. When switching to next entry in keying list (author), zoom on zoom works. Here the zoom is present, but because of large headline (has a big width) it seems to not having any zoom.

In Compare Keying only the zones that have differences between VerifyOCR and Re-Typing, will be shown. If for example a headline is not displayed, this is because there are no differences between the detected OCR and the corrected one.

## 12 Update procedures

Updates will be installed inside a folder on the main network share ("Install"). Local copies of Binaries will be updated automatically without any interaction by administrators.

The install folders are configurable with at most 3 entries: dwBIN (as old sources), dwOCR and dwSCRIPT (last 2 folders are new and optional).

dwOCR will be used for updating local OCR folder, dwBIN for all other local folders. See "Local Binaries" chapter. dwSCRIPT stores post update scripts.

## 13 Auto-Update

docWizz has its own auto-update feature for the binaries. It allows for a centralized deployment to all connected client machines.

### What are the benefits?

Network traffic is reduced by estimated 90%.

Decreases the down time of the client environment and manual intervention by improved update logic.

No more forgotten machines; all connected machines will be forced to update.

Additionally the Post Update feature allows for installation of 3rd party software, e.g. VCREDIST, MSXML or cleanup.

Customized PostUpdate commands allow actions to be executed after an update.

Reduce manual update actions for IT admin.

### 13.1 Folder structure

At start of docWizz the update routine is executed.

In regular intervals during runtime the routine is executed.

The folder structure is mandatory in that format and syntax.

The dwShare contains the INSTALL folder. There are two subfolders:

DWBIN

DWOOCR

#### **DWOOCR**

Used to update the dwOCR folder on any connected machine.

#### **DWBIN**

Used to update all other three folders on any connected machine: docWizz, dwsrv, dwssystemservice.

The path to the mandatory folders is defined in the [Update] section within the docWizz-glbl.ini.

```
<servername>\\<cfg-share>\config\docWizz-glbl.ini
```

```
[UPDATE]
```

```
DWBIN=***DATA***\Install\DWBIN
```

```
DWOOCR=***DATA***\Install\DWOOCR
```

```
DWSCRIPT=***DATA***\Install\DWScript
```

```
EXCLUDED=global.ini,DWSrvps.dll
```

DWBIN=	used for all other local folders; contains all regular binaries and files.
DWOOCR=	will be used for updating local OCR folder; compared to DWBIN it contains additionally the OCR files.
DWSCRIPT=	stores post update scripts; the "clean.cmd" is disabled by default. vcredist_x86.exe is in the DWSCRIPT folder as it is requested by the script there.
EXCLUDED=global.ini, DWSrvps.dll	
OCRSERVICE=	

You will receive errors if DWOOCR and DWSCRIPT are not used, so if you don't have the folders, just configure them in the same path as DWBIN.

### Update Script

It allows actions to be executed AFTER an update.

### Examples:

Install files: VCREDIST, MSXML

Delete single files or file extensions. Is based on Windows batch logic, batch programming can be used.

docWizz.exe and DWSrvManager start their own post update mechanism:

docWizz\_postUpdate.cmd

DWSrvManager\_postupdate.cmd

The content of both cmd are normally the same.

For each folder logging will be done in EventViewer.

Additionally in WORK folder docWizz will keep log files.

As to post update scripts that will be configured in:

DWSCRIPT=\*\*\*DATA\*\*\*\Install\DWScript

The post update for DWSrvManager and docWizz will create a log file in a fixed folder  
\*\*\*DATA\*\*\*\WORK\RedistributableInstalls.

If docWizz installed redistributables, dwsrvmanager will not. The redistributable install is not dependant on user install so you do not need to differentiate between the user install and machine install, it will be done only on machine install.

However Clean will be performed as such:

- docWizz client post update will only clean docWizz folder
- dWSrvManager post update will clean all the other folders

Even if redistributables are already installed, clean will be always performed.

## 13.2 Routine calling customized shell scripts

In docWizz\_postUpdate.cmd the path of the update folder is needed for the redPth variable set  
RedPth=\\Swdev-qa-1\\h\\BinForUpdate2\\

and you will have to call install\_vcredist\_x86.cmd with its entire path and three parameters:

- the app that triggers the redistributable install : docWizz or DWSrvManager
- the redistributable exe name: vcredist\_x86.exe and the redpath: %RedPth%I\\Swdev-qa-1\\h\\BinForUpdate2\\install\_vcredist\_x86.cmd docWizz vcredist\_x86.exe %RedPth%

Install\_vcredist\_x86.cmd basically logs events for event viewer log, with all its actions: it will try to copy the redistributable exe to temp folder on target machine, if copy fails, it will try 10 times logging error events for each try, after 10 failed attempts it stops with error event.

If copy is successful it will try to install, if install fails, error event is logged and script stopped.

If install is ok, the exe is erased from temp path and success event is logged.

## 13.3 Main files to check and start the Update process

There are main files who check and start the update process:

docWizz.exe	Updates the main „GUI" folder and afterward starts dwcl.exe.
DWSrvManager	Updates all the „Services" folder; stops and starts registered services as needed.
(dwscanstart.exe)	(Updates the main „GUI" folder and afterward starts dwscan.exe.)

### Procedure

dWSrvManager detects an update needed, stops all registered services and calls docWizz.exe .

docWizz.exe will stop dWSrvManager and proceed with the update. (That also updates „dWSrvmanager.exe")

docWizz.exe gets checked and updated at the start of dWSrvManager.exe and dwcl.exe.

Background info: docWizz.exe is the *\*only\** file which doesn't need *\*any\** dependent files. This means it can *\*always\** run in the Windows environment, even if other DLL or runtime files are missing.

## 13.4 Prevent computer from updating

Sometimes the integrated auto-update routine needs to be suppressed or certain computers need to be excluded.

For this scenario is the `UpdateConfig.ini` in the dWShare config folder.

```
<servername>\\<cfg-share>\config\Updateconfig.ini  
[UpdateConfig]  
UpdateDisabled=0  
IgnoredComputers=sampleWS
```

UpdateDisabled has two parameters:

Disabled `0` \\ default value, only IgnoredComputers are not updated

Enabled `1` \\ no update is made on any computer

IgnoredComputers excludes certain computers from update. In CSV style, e.g.:

```
IgnoredComputers= MN-NB, WSC-04, Server01
```

## 13.5 Example how to prepare an execute a regular update

Open Windows Taskmanager.

Set „UpdateConfig.ini“ to disable updates .

Prepare folder structure with new binaries.

- Copy & overwrite into DWBIN

- Copy & overwrite into DWOCR

- Copy & overwrite into DWSCRIPT

Merge new config files.

Set „UpdateConfig.ini“ to enable updates.

DWSrvManager.exe stops all executed services.

DWSrvmanager starts „docWizz.exe“ in Task manager, who does the update in all folders, where registered services are found:

- dwSystemService

- dwSRV

- dwOCR

- docWizz

The DWSrvManager starts all services, which where executed before.

Start DWCC or DW, check version number.

Update done.



## 13.6 DWOCRREMOTE auto-update functionality

Auto-Update for bin/OCRService folder.

The functionality is only enabled on remote OCR configurations. It is designed on the standard distribution, where the OCR service is stored in a separate path.

### Used Files

\*\*\*DATA\*\*\*\WORK\OCRSERVICE\computername.DWOCRREMOTE.UPD (used as indicator, that a update is running.)

\*\*\*BIN\*\*\*\dwocrremotedirectory\UpdateOCR.cmd (batch file for update)

### DwOcrRemote Service working

Timer is looking for changes in installation folder

Creating dwocrremote.upd for signals start.

Closing all dwocrrun.exe

Creating batch file

Creating scheduler task and end service

On start, cleanup DWOCRREMOTE.UPD file and batch file

### DWCL and DWSRV working

On start of dwocrrun.exe look for \*.DWOCRREMOTE.UPD files, not older the 10 minutes. If found wait while exist

## 13.7 Background Information

On INSTALL the file „Update.csv“ is created, which lists file name, size and timestamp.

This CSV is copied to local program folder(s).

The entries of the local CSV file are checked one by one, if file name exists locally and if time and size is the same.

If a file is detected in local CSV, which is missing on local program folder, this file is transferred.

If a difference in size or timestamp is detected, it is also transferred.

**Important: Doesn't matter if timestamp is newer or older (required for downgrade)!**

To trigger update procedure after preparation, install path Update.csv **\*MUST\*** be deleted.

## 14 Manual updates

Update was optimized for network usage as follows. Update source folder has an csv text file (UpdateCSV.txt) containing last write time and size of each file.

If it's not present it'll be created by the first app that checks for update e.g.:

```
LastWriteFILETIMELow:LastWriteFILETIMEHigh, FileSizeLow:FileSizeHigh, date  
(for humans), Size (for humans), FilePath 2143835604:30035165, 622592:0,  
10/14/2009 17:49, 608 KB, e:\UpdateTest\BinForUpdate\dwOCR\ACE.dll .
```

This file is checked for different date and size and if they differ it's copied into destination folder. Local comparison between dates from this file and local files is always performed (even on equal time/size Update.csv). If dates differ, update server file is copied locally. Also on server update.csv file is checked if is outdated, and recreated if it is.

### Updates are checked:

- For GUI folder (docWizz): when starting docWizz.exe, each 30 minutes from dwcl.exe
- For services folder (dwSRV): each 1 min
- For OCR: multiple places:
- From GUI and services when OCR is needed
- For RemoteOCR, each 20 secs

Update means: not all files in source (dWBIN/dWOCR) folder are available on local disk or file date, time or size are different.

### Update cases:

- Update.csv file from source folder and from local folder are identical, if all binaries on local folder are identical ( identical will be used for now as into having the same date, file size and name ) and no files are missing from the list in Update.csv file, no update will be done. If files on local folder are missing, or are different ( as into not having the same filesize, date or name ) from the ones written in the Update.csv file, then the missing / different files are updated from source folder.
- Update.csv is different on local folder, then this file will be updated and the update procedure is initiated as for point a.
- Update CSV is missing in source folder, the file is generated, and update procedure is started according to point a. or b.

Note: if UpdateCSV file is present in dWBIN folder, and new files are added, update is not performed. For each new update UpdateCSV file shall be deleted in order to update the applications.

### 14.1 Removing Files

Removing files or obsolete files from local machine install folders, will be done via the post update shell scripts execution.

For dWScanClient the "dWScan\_Start.exe" was removed from built and added to remove file list.

### 14.2 Local Binaries

Installation of local binaries is as follows:

**docWizz** - a folder containing binaries that will be used for any application that has UI

**dWSrv** - a folder containing binaries for all regular services (DWSrvManager + DWsrvs)

**dWSystemService** - a folder containing binaries for all services that must be available to ensure operating (RQA, FTP, RemoteOCR)

**dWOOCR** - a folder containing all OCR engine specific files

Any other combination is not supported. This scenario is a little different from how it was before. Update process shall ensure to automatically modify local installations.

For now, the code doesn't move but copies to specified folders and deletes exe files.( rename to ex\_ or dl\_ for OCR engine files)

Services registration is changed to point to their new folders. dWSrvManager will always be registered if not already registered. Starting dwcl.exe from services path will not work -> as designed, should work with docWizz: starting docWizz from DWSRV folder, will update dwsrv folder content, return error and exit.

When moving files will be copied from old location to new one as follows:

All folders: **global.ini**

OCR: **soft key**

### 14.3 Updating services (dWSrv + dWSystemService)

Updating services is handled by dWSrvManager, when needed, dWSrvManager copies docWizz.exe (regardless of the fact that docWizz.exe could be latest or not ) after this copy, dWSrvManager starts docWizz.exe and exit.

docWizz.exe will restart dWSrvManager after update. If services' registration needs to change mmc consoles will be killed not to interfere with updating.

Note: Any errors from non-gui applications(dWSrvManager) are logged into EventViewer.

### 14.4 OCR

Updating of local OCR is done on OCR calls so, in order to break GUI less frequently the same timer as for GUI is used to check for OCR updates(for example if user selects from dwcl.exe to remember to update in 30 mins also OCR will be updated in 30 mins).

At RemoteOCR still need to be implemented this (RemoteOCR are still unstable at this moment): If it uses RemoteOCR, it waits at least 5 minutes. In UI a message is shown: "Waiting to update OCR". If RemoteOCR is not back, it tries to connect to a different host.

Message: OCR Update ungergoing at start of docWizz

At the start of DWCL, OCR update need is checked, each time, if it is needed, then the OCR Update is taking place, otherwise dwcl starts normally.

Timing issues are handled, if you have the DWOCRrun.exe executed for dwcl in the same folder as for DWRemoteOCR, the first one that detects OCR Update need will perform it, while the other one waits.

Dwcl checks for update when having Local OCR, while RemoteOCR will do the update for it's OCR. If OCR update is detected by dwcl, at startup, or on OCR request, the GUI is blocked until the OCR update is done, while RemoteOCR will kill all DWOCRRun.exe and update the dwocr folder.

## 14.5 GUI

Update for GUI folder is just checked from 30 min/ 4 h in function of user's last selection from dwcl.exe.

User decides when will restart dwcl.exe. Before starting dwcl.exe update is made.

## 14.6 Post Update scripts

Batch scripts are executed for each computer where docWizz or docWizz services are running. Any batch, command line scripts that can be executed on a computer.

There is no GUI for this feature, just info onto the outcome of the execution, event viewer logs, for the execution.

### Configuration

Post update scripts are configured in docWizz-GLBL.ini ( or custom glbl.ini) file, under the DWSCRIPT entry, by default the script folder is configured as such :

```
DWSCRIPT=***DATA***\UpdateScript
```

### Standard settings

docWizz client will run the docWizz\_postUpdate.cmd script after an update.

dWSrvManager will run the DWSrvManager\_postUpdate.cmd after an update.

The post update for dWSrvManager and docWizz will create a log file in a fixed folder

```
***DATA***\WORK\RedistributableInstalls
```

If docWizz installed redistributables, dwsrvmanager will not, the redistributable install is not dependant onto user install so we do not need to differentiate on the user install and machine install, it will be done only on machine install. However Clean will be performed as such:

- on docWizz post update only in docWizz folder
- on dWSrvManager post update on all other folders

Even if redistributables are already installed, clean will be always performed

As a sample, the install redistributables and clean scripts were configured to run for both docWizz and dWSrvManager, with different event reporting (docWizz postUpdate and dWSrvManager postUpdate) in event viewer.

If redistributables are already installed, an event with the type information is logged, telling the user that the redistributables are already installed from the configured path. This is done due to the first install that will log the execution on a common folder path under WORK\RedistributableInstall folder with a log of this form: "workStation-vc redistrib\_x86.exe.txt" for each computer.

The clean example script, is configurable as well, using the CleanList.txt, to configure file names, or file extensions as a list (e.g. updatemanager.cmd, \*.pdb, \*.old) the clean is executed as follows:

- docWizz post update will clean only docWizz and dWOOCR folder, logging this event in event viewer as well.
- dWSrvManager post update will clean dwsrv, dwsystemservice and dwocr folders, logging this in event viewer as well.

Note: dWSrvManager will run post update scripts after each folder update (dwsrv, dsrv or dwocr) thus three post updates will be logged. docWizz will run post update scripts after each folder update (docWizz, dWOOCR) thus two post updates will be logged.

## 14.7 Notes

Code updating docWizz.exe has moved from InitConfig. It's done by folder location:

- `.\docWizz\dwcl.exe` updates docWizz.exe file when starting
- `.\dwSrv`, `.\dwSystemService`, `.\dwOCR` dwSrvManager updates docWizz.exe file when starting

After updating a batch command is executed:

- for GUI: update path + `docWizz_postUpdate.cmd`
- for services: update path + `DWSrvManager_postUpdate.cmd`

For these batch commands the delete of obsolete or unneeded files will be used or install of redistributable software. See TFS example.

Command options for docWizz.exe:

- command prompt options for docWizz.exe:

<code>docWizz.exe /?</code>	This Help
<code>docWizz.exe</code>	Update the current folder and start dwcl.exe
<code>docWizz.exe app.exe</code>	Update the current folder and start app.exe
	<code>docWizz.exe -L \"user@domain\" \"password\" -&gt;</code> Create the logon.dll in the current folder and exit
<code>docWizz.exe -L \"user@domain\" \"password\"</code>	Create the logon.dll in the current folder and exit
<code>docWizz.exe -U [-nogui] [-startDWSrvManager] [path]</code>	Update the binaries to the specified path or the current if missing the path. Stop/Start the DWSrvManager if the parameter exist
<code>docWizz.exe -U</code>	Doesn't stop running services. This is the intended behavior, only DWSrvManager sends stop commands to other docWizz services
<code>docWizz.exe -O</code>	Update OCR binaries and exit
<code>docWizz.exe -register/unregister -dwsrv/dwsystemservice</code>	Register services/Unregister services

As a non design feature, if dwcl is started as standalone and docWizz.exe is not present in the folder, then according to a timer, dwcl will copy docWizz.exe into the current folder.

Logon.dll usage:

- If logon.dll is present in folder, docWizz runs as with user encoded to logon.dll
- If logon.dll is not present, docWizz runs as current user.
- Services are not using this file

Possible issues when updating but not handled directly by code:

- Installing redistributables - shall be installed by post update script
- No disk space ( ! )
- Wrong user name/pass in cfg - shall not happen
- Not enough rights for Global.ini(docWizz.exe starts with few rights)
- Registering services might not succeed if services.msc(windows services console) is opened

## 15 Notes for Windows 7 or Server 2008 systems

docWizz.exe needs to run with administrative rights.

It will still be needed for users logged on to Windows as non admin users, with low privileges, in order to have docWizz executed.

## 16 PDFcfg.xml

PDFcfg.xml file is stored under the `..\project-cfg\<customer>\PDF` path.

Please be aware of the renamed steps and tasks and the "old" names in scripts and configuration files.

### 16.1 Configuration file for PDF creation (PDFcfg.xml)

#### Export PDF Configuration (TCL scripting solution)

This documentation is only for using the PDF export configuration with TCL scripting (PDFCfg.xml).

The generation of PDF files with the TCL solution offers the advantage to be able for the creation of large pdf files. The appearance of the PDF files which can be spent is very simply configurable.

To activate the PDF creation with TCL solution the files specified in following table must be present into the appropriate folders. Additionally the existing files for the existing PDF creation must be removed from the appropriate PROJECT-CFG folders.

Files	Folder	Description
PDFCfg.xml	Project-cfg\[PROJECT]\pdf and Project-cfg\Default\pdf	PDF configuration file. See special parameter description below.
PDFBuilder.tcl	***DATA***\script	Functions for PDF creation
PDFBuilderFct.tcl	***DATA***\script	Functions for PDF creation
PDFBuilderSampleCall.tcl	***DATA***\script	Functions for PDF creation

The file "PDFCfg.xml" file must be located in the subfolder 'PDF' of the project configuration.

If this file exist there, those settings will be loaded by the function 'LoadExternalVariables' of the TCL script file "PDFBuilderFct.tcl".

If this file is missing, the configuration of

`***PROJECT_CFG***\default\PDF\PDFCfg.xml` will be taken.

To prevent errors by older configuration files which does not contain all possible settings, the TCL script contains the default values inside the TCL script file "PDFBuilderFct.tcl" once again for initializing. These are the final default values which will be taken if a configuration setting does not exist in the referenced external config file.

On executing the PDF generation by an manual script from the script editor, the default configuration file won't be analyzed. In that case only a manual loaded configuration (by function 'LoadExternalVariables') or the default values inside the TCL script file "PDFBuilderFct.tcl" will be used!

#### Options:

- options for configuration of topLevel bookmark groups (advertisement, tables, ...)
- option for partial images inclusion (PartialImages & PartialImageResolution)
- option for standard single page generation (previous interfaces for special single page generation still exist)
- option for hiding bookmarks which contains no sub-elements (generateIfEmpty=0) for logical structure bookmarks
- option for defining the order of bookmarks (generateGroupBookmarksFirst=1): first the structure bookmarks and afterwards group bookmarks or other way around



## 16.2 Configuration setting description

Description for current version script version 1.47.

This script will need the PDFBuilder.dll version >= 6.0.6.3.

### 16.2.1 PDF Version

**XML syntax:** `<PDFVersionNumber>1.4</PDFVersionNumber>`

**Description:** PDF Version Number

```
1.2 = Acrobat 3;
1.3 = Acrobat 4;
1.4 = Acrobat 5;  (default)
1.5 = Acrobat 6;
1.6 = Acrobat 7;
1.7 = Acrobat 8;
```

### 16.2.2 Hidden Text

**XML syntax:** `<HiddenText>1</HiddenText>`

**Description:** En-/ Disables creation of hidden text.

If hidden text is disabled, no alto files are needed.

```
0 = no;
1 = yes [ALTO files are required];          (default)
```

### 16.2.3 PartialImages

**XML syntax:** `<PartialImages>0</PartialImages>`

**Description:** Defines, if the partial images should be included into the PDF (if exist).

Option "hiddenText" must be enabled! (only for multipage / singlepage PDF please call CreatePartialImageListFromMets before creating pdf).

En-/ Disables creation of hidden text.

```
0 = no;          (default)
1 = yes [ALTO files are required];
```

### 16.2.4 GenerateBookmarks

**XML syntax:** `<GenerateBookmarks>1</GenerateBookmarks>`

**Description:** Option to switch on/off the generation of bookmarks in PDF for generating bookmarks, the logical structMap will be necessary in METS file.

(only for multipage PDF, ignored for singlepage PDF)

```
0 = no;          (default)
1 = yes;
```

This option is only regard to multipage PDF. For singlepage PDF it is ignored.

If no alto files are available no bookmarks can be created (needs the coordinates for the bookmarks).

A creation of bookmarks with linking to the pages (without detailed coordinates) is not implemented.

The inclusion of the bookmark types and their naming can be defined in the subnode of this configuration file named `<PDFConfig:PDFBookmarkTypes>`.

Each type which should be added to the bookmark list has to been inserted as subnode "`<PDFConfig:PDFBookmarkType>`" containing the type name inside the attribute "type".

If the naming inside the PDF should be different to the type name, you have to define a further attribute named like the language, f.e. 'EN'.

Then the value of this attribute will be set to the bookmark. If no languages are defined, the type itself will be inserted into bookmark.

If an language attribute for a further language (f.e. 'DE') doesn't exist, the default language 'EN' will be taken.

#### Example:

```
<PDFConfig:PDFBookmarkType type="VOLUME" EN="Volume" DE="Band"/>
<PDFConfig:PDFBookmarkType type="FRONT"/>
```

If 'DE' is specified at 'PDFBookmarkLanguage', the 'volume' element will be inserted into the bookmark structure with the naming 'Band'.

For the 'FRONT' type it will be also created an bookmark element, but it will contain 'FRONT' as label.

If this will be the only `<PDFConfig:PDFBookmarkType>` elements, no 'MAIN' or 'BACK' type will be added as bookmarks.

### 16.2.5 SinglePage

**XML syntax:** `<SinglePagePDF>0</SinglePagePDF>`

**Description:** Option to switch on/off the generation of single PDFs for each page.

The PDF file name will be extended by the image filename.

(no bookmark generation possible!)

```
0 = no;           (default)
1 = yes;
```

Note: If this option is activated, no bookmarks can be generated any more. (Bookmark references to other PDF files are impossible.) A warning will be returned, if the bookmark generation is enabled.

### 16.2.6 PageMode

**XML syntax:** `<PageMode>0</PageMode>`

**Description:** Option for Acrobat Reader, which windows/bars to display on opening document.

```
0 = none;                (default)
1 = show bookmarks;
2 = show thumbnails;
3 = fullscreen mode;
4 = show comment window;
5 = show attachment window;
```

### 16.2.7 PageLayout

**XML syntax:** `<PageLayout>0</PageLayout>`

**Description:** Option how to display the content pages itself.

```
0 = Single page;          (default)
1 = Consecutive;
2 = continuous,two pages, start left;
3 = continuous,two pages, start right;
4 = two pages, start left;
5 = two pages, start right;
```

### 16.2.8 OpenAction

**XML syntax:** `<OpenAction>0</OpenAction>`

**Description:** A value specifying a destination to be displayed or dictionary an action to be performed when the document is opened. The value is either an array defining a destination (see Section 8.2.1, "Destinations") or an action dictionary representing an action (Section 8.5, "Actions"). If this entry is absent, the document should be opened to the top of the first page at the default magnification factor.

```
0 = don't set openAction;    (default)
1 = set openAction
```

**Example:** `<OpenAction>1</OpenAction>`

**Description:** Document is opened "Fit to page". When opening the PDF file the whole page is shown.

### 16.2.9 PDFBookmarkLanguage

**.XML syntax:** `<PDFBookmarkLanguage>EN</PDFBookmarkLanguage>`

**Description:** Language of the PDF bookmarks type in the attribute name of the language defined below for the names of the bookmark types

```
EN - English;             (default)
DE - German
```

FI - Finnish  
SE - Swedish

### 16.2.10 PDFBookmarkCoords

**XML syntax:** `<PDFBookmarkCoords>0</PDFBookmarkCoords>`

**Description:** Type of the bookmark coordinates.

0 = coords of the element itself, depending on the size the PDF will be zoomed in (default);

--> Zoom factor is always different.

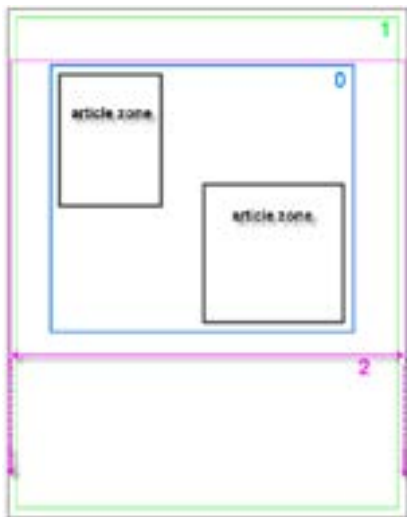
1 = coordinates of the whole page, where the referenced object is starting;

--> Zoom factor is always full page.

2 = fit bookmark's bounding rectangle to page's width and compute only it's top bound;

Depending on the screen, whole page or part of it is shown.

--> Zoom factor constantly based on page width.



### 16.2.11 LinearizedPDF

**XML syntax:** `<LinearizedPDF>1</LinearizedPDF>`

**Description:** Web optimized PDF.

0 = no

1 = yes; (default)

### 16.2.12 UserAccessPermissions

**XML syntax:** `<UserAccessPermissions>0</UserAccessPermissions>`

**Description:** User permissions for using PDF (insert decimal sum of the binary codes).

0 = all allowed (default) [ => permission property won't be set by TCL script]

If user access permission is set to '0', this property won't be set by the TCL script, so that everything will be allowed. (default)

For a more detailed documentation of other PDF user permissions have a look in the configuration file PDFCfg.xml



### 16.2.13 ImageResolution

**XML syntax:** `<ImageResolution>300</ImageResolution>`

**Description:** Target resolution of the images inside the PDF values are set in dpi (dots per inch).

default: 300

Special values:

0 - same resolution as in original

1 - same resolution like in original and the image stream is identical with the image on disk

### 16.2.14 PartialImageResolution

**XML syntax:** `<PartialImageResolution>300</PartialImageResolution>`

**Description:** Target resolution of the partial images inside the PDF values are set in dpi (dots per inch).

default: 300

### 16.2.15 ImageQualityJPG

**XML syntax:** `<ImageQualityJPG>0</ImageQualityJPG>`

**Description:** Quality option for target JPG images inside PDF .

!!! for PDFs until version 1.4

Values between 0 and 100 - 0 let docWizz decide the best quality.

In case images used for PDF are not jpg, and version smaller than 1.4, best quality is used.

In case image is bitonal, the images will be converted to Tiff G4.

default: 0

### 16.2.16 ImageQualityJP2K

**XML syntax:** `<ImageQualityJP2K type="1">7</ImageQualityJP2K>`

**Description:** Quality option for target JPG2000 images inside PDF.  
!!! for PDFs starting version 1.5

default: 0 - loseless

type: 1 = percent - between 0 and 100 (Default)  
2 = ratebytes - final image size in bytes  
3 = slope - uses the slope compression option  
4 = PSNR - uses the PSNR compression option

In case images used for PDF are not jp2, and version bigger than 1.4,  
lossless is used  
In case image is bitonal, the images will be converted to Tiff G4

### 16.2.17 UseBWPages

**XML syntax:** `<UseBWPages>0</UseBWPages>`

**Description:** Add black/white images

0 = no; (default)  
1 = yes;

If no third/fourth image reference exist in physical structMap, the normal image will be taken.

### 16.2.18 AddPageThumbnail

**XML syntax:** `<AddPageThumbnail>0</AddPageThumbnail>`

**Description:** Add thumbnails of added pages.

0 = no; (default)  
1 = yes;

### 16.2.19 AddPageLabel

**XML syntax:** `<AddPageLabel prefix="">0</AddPageLabel>`

**Description:** Add logical page number for the pages (available since PDF 1.3 [= Acrobat 4:]).  
(only proved for multipage PDF; for singlePage PDF this will be configured  
only by the parameter (is given or is not [empty string])

```
0 = no;                (default)
1 = logical page number;
2 = physical page number;
3 = physical and logical page number like this '<phys> ( <log> )'
```

If an text is inserted inside the attribute 'prefix' of the node 'AddPageLabel', this text will be added on each page.

### 16.2.20      **FontType**

**XML syntax:**    `<FontType>0</FontType>`

**Description:**    Set font type for the hidden text.

```
0 = Times;                (default)
1 = Helvetica;
2 = Courier;
```

### 16.2.21      **TextRenderingMode**

**XML syntax:**    `<TextRenderingMode>3</TextRenderingMode>`

**Description:**    Text rendering modes.

```
0 = Fill;
1 = Stroke;
2 = FillThenStroke;
3 = NeitherFillNorStroke;    (default)
4 = FillAndAddToPathForClipping;
5 = StrokeAndAddToPathForClipping;
6 = FillThenStrokeAndAddToPathForClipping;
7 = AddToPathForClipping;
8 = TextRenderingModesCount;
```

### 16.2.22      **TextObjectsAsAnsi**

**XML syntax:**    `<TextObjectsAsAnsi>0</TextObjectsAsAnsi>`

**Description:**    Defines if the text objects should be inserted as ansi coding instead of unicode.

```
0 = unicode;                (default)
1 = ansi code;
```

### 16.2.23 UseCustomizedSettings

**XML syntax:** `<UseCustomizedSettings>0</UseCustomizedSettings>`

**Description:** Defines if special customized settings has to be used.

```
0 = no customized options;           (default)
1 = special settings for NDNF [special object references];
```

### 16.2.24 TaggedPDF

**XML syntax:** `<TaggedPDF>0</TaggedPDF>`

**Description:** Enables or disables the tagged pdf export.

Suitable for mobile devices.

Overwrites TextObjectsAsAnsi to 0.

```
0 = non tagged pdf;                 (default)
1 = tagged pdf
```

### 16.2.25 PDFAType

**XML syntax:** `<PDFAType>0</PDFAType>`

**Description:** Enables PDF/A profile and overwrites LinearizedPDF to 0,

TextObjectsAsAnsi to 0,

PDFVersionNumber to 1.4/1.7,

TaggedPDF to 0 or 1 for b or a.

```
0   = disabled;                     (default)
1a  = PDF/A 1a profile, tagged archive pdf
1b  = PDF/A 1b profile, non-tagged archive pdf
2a  = PDF/A 2a profile, tagged archive pdf;
2b  = PDF/A 2b profile, non-tagged archive pdf;
3a  = PDF/A 3a profile, tagged archive pdf;
3b  = PDF/A 3b profile, non-tagged archive pdf;
```

### 16.2.26 PDFFontPath

**Description:** Use the font from the given path.

```
<PDFFontPath>
  <Unicode></Unicode>
  <ANSI></ANSI>
</PDFFontPath>
```



### 16.2.27 EmbedFont

**XML syntax:** `<EmbedFont>2</EmbedFont>`

**Description:** Embed the font from the path above or from resources.

`0 = Disable. No font embedded. (Default);`

If PDFFontPath contains a valid font path, this will not be embedded, but will be used as PDF font, and all hidden text coordinates will be computed based on this. When using the generated PDF the referenced font should be installed on the machine, else an warning will appear (font not available) when opening the PDF.

`1 = Embed the font from the given path.`

`2 = Embed font from resources. (FreeSans font is used)`

### 16.2.28 FontMinimize

**XML syntax:** `<FontMinimize>1</FontMinimize>`

**Description:** Removes glyph draw information from the font stream, by making it smaller.  
Can be used with hidden text. If used with hiddentext 0 no glyphs will be shown.

`0 = No minimize.`

`1 = Minimize font. (Default)`

### 16.2.29 FontTextShrinkPercent

**XML syntax:** `<FontTextShrinkPercent>0</FontTextShrinkPercent>`

**Description:** Shrinks the text height with the percentage value given as double.

`0 = No action taken. (Default)`

`0.0 < [value] < 1.0 = Shrink the text rectangle and centers it vertical:`

`Top = Top + Height * [value] / 2 ; Height = Height * [value]`

### 16.2.30 BiDirectionalText

**XML syntax:** `<BiDirectionalText>0</BiDirectionalText>`

**Description:** Enables processing of BiDirectional texts (arabic, hebrew).

`0 = No bidirectional text processing. (Default)`

`The characters are added the in the order provided at input.`

`1 = Process the text and put the characters in the right order,  
both LeftToRigth and RightToLeft`

### 16.2.31 EncapsulateTextSelection

**XML syntax:** `<EncapsulateTextSelection>0</EncapsulateTextSelection>`

**Description:** Computes the font size so that the text with or without selection will be drawn to fill the text rectangle.

```
0 = The text will be drawn to exactly fill the given rectangle;
    (Default)
other = The text will be drawn so the text selection will exactly fill the
        given rectangle
```

All the non alpha characters (punctuation mark) will have automatic text selection encapsulation enabled to avoid having bad selection rectangle in PDF.

Non encapsulated text in PDF:



Encapsulated text in PDF:



### 16.2.32 PDFBookmarkTypes

**XML syntax:** `<PDFConfig:PDFBookmarkTypes>`

```
<PDFConfig:PDFBookmarkType type="X" generateIfEmpty="n" EN="Y" DE="Z"/>
```

...

```
</PDFConfig:PDFBookmarkTypes>
```

**Description:** Default language is 'EN', if configured language (above) doesn't exist, the default will be taken.

If no languages are defined, the type itself will be inserted into bookmark.

Attribute 'useLabel' defines, if the label of an element (if exist) or always the translated type names should be used.

```
0 = use title of element if exist
    if not exist the translated type name will be used
1 = always the translated type name will be used (even if a label exist)
    (default)
    if no translation exist, the type name itself will be used
```

In both cases: if no translation exist, the type name itself will be used.

### 16.2.33 PDFGroupBookmarks generate

**XML syntax:** `<PDFConfig:PDFGroupBookmarks generate="1" groupBookmarksFirst="0">`

**Description:** Attribute 'generate' enables / disables the group bookmarks on top level.

```
0 = disabled;
1 = enabled;
```

**Description:** Attribute 'groupBookmarksFirst' defines, whether the logical structure bookmarks or the group bookmarks should be the first in bookmark list sequence

```
0 = disabled;      (default)
1 = enabled;
```

### 16.2.34 Additional Bookmarks

**XML syntax:** `<PDFConfig:PDFGroup name="Advertisements" generateIfEmpty="1" EN="Advertisements" DE="Werbung" FI="Ilmoitukset" SE="Annonser">`  
`<PDFConfig:PDFChild type="ADVERTISEMENT"/>`

**Description:** Each 'PDFGroup' node represent one group node in the result PDF.

The subnodes 'PDFConfig:PDFChild' contains in the attribute 'type' those export types, which should be inserted in that group.

Attribute 'generateIfEmpty' configures, if the group element should be generated, if no bookmarks were generated inside that group.

```
0 = not generated if contains no bookmarks;
1 = generated even if no bookmarks inside;
```

The number of groups and child elements is extensible.

### 16.2.35 Import PDF files as image

**XML syntax:** `<ANALYZE_AS_IMAGE>1</ANALYZE_AS_IMAGE>`

**Description:** Enables .pdf files import as images, ignoring the pdf layers.

### 16.2.36 PDF Metadata

The PDF metadata are visible in the file properties on an extra tab.

KEY	TYPE	VALUE
<b>Title</b>	text string (Optional; PDF 1.1)	The document's title.
<b>Author</b>	text string (Optional)	The name of the person who created the document.
<b>Subject</b>	text string (Optional; PDF 1.1)	The subject of the document.
<b>Keywords</b>	text string (Optional; PDF 1.1)	Keywords associated with the document.
<b>Creator</b>	text string (Optional)	If the document was converted to PDF from another format, the name of the application (for example, Adobe FrameMaker®) that created the original document from which it was converted.
<b>Producer</b>	text string (Optional)	If the document was converted to PDF from another format, the name of the application (for example, Acrobat Distiller) that converted it to PDF.
<b>CreationDate</b>	date (Optional)	The date and time the document was created, in human-readable form.
<b>ModDate</b>	date (Required if PieceInfo is present in the document catalog; otherwise optional; PDF 1.1)	The date and time the document was most recently modified, in human-readable form. (on page 140)

(source: Adobe, [PDFReference16](http://partners.adobe.com/public/developer/en/pdf/PDFReference16.pdf)  
<http://partners.adobe.com/public/developer/en/pdf/PDFReference16.pdf>)

#### Command example:

```
CreatePDF_MultiPage_metadataString2 <metsPathFilename> <resultPdfPathFilename>  
<xmpMetadataPathFileName(can be empty)> <title(can be empty)> <subject(can be empty)>  
<author(can be empty)> <creationdate(can be empty)> <modificationdate(can be empty)> <creator(can  
be empty)> <keywords(can be empty)> <producer(can be empty)>
```

```
CreatePDF_MultiPage_metadataString2 [format "%s-METS.xml" $outputname] [format  
"%s.pdf" $outputname] "" "" "" "" "" "" "" "" ""
```

#### 16.2.36.1 Detailed Explanation of Date

##### Dates

PDF defines a standard date format, which closely follows that of the international standard ASN.1 (Abstract Syntax Notation One), defined in ISO/IEC 8824 (see the Bibliography).

A date is a string of the form (D:YYYYMMDDHHmmSSOHH'mm') where:

<b>YYYY</b>	year
<b>MM</b>	month
<b>DD</b>	day (01–31)
<b>HH</b>	hour (00–23)
<b>mm</b>	minute (00–59)
<b>SS</b>	second (00–59)
<b>O</b>	relationship of local time to Universal Time (UT), denoted by one of the characters +, -, or Z (see below)
<b>HH</b>	followed by ' is the absolute value of the offset from UT in hours (00–23)
<b>mm</b>	followed by ' is the absolute value of the offset from UT in minutes (00–59)

The apostrophe character (') after HH and mm is part of the syntax. All fields after the year are optional. (The prefix D:, although also optional, is strongly recommended.)

The default values for MM and DD are both 01; all other numerical fields default to zero values. A plus sign (+) as the value of the O field signifies that local time is later than UT, a minus sign (-) signifies that local time is earlier than UT, and the letter Z signifies that local time is equal to UT. If no UT information is specified, the relationship of the specified time to UT is considered to be unknown.

Regardless of whether the time zone is known, the rest of the date should be specified in local time.

For example, December 23, 1998, at 7:52 PM, U.S. Pacific Standard Time, is represented by the string D:199812231952-08'00'.

The date fields can also be filled with normal date strings. These are however not transferred then into regional formatting while viewing (f.e. USA 12/24/2007 4:15:23pm).

#### 16.2.36.2 Single Page PDFs

##### **Create a single page PDF by getting the image and associated ALTO file**

The iteration over this function for all pages of a document has to be done outside, also the metadata files has to be generated before.

Creating special single page PDFs where each PDF (of each page) should contain different parameters / metadata!

Here no METS file is needed - each pdf is created with the given image and alto file (and optional metadata).

It is also possible to create single page PDFs using the file references of the METS file. Therefore use the Multi Page PDFs commands and enable the option single page pdf in the configuration file.

##### **Create single page PDFs with hidden text**

command syntax:

```
CreatePDF_SinglePage_widthHiddenText <imgPathFilename> <altoPathFilename>
<resultPdfPathFilename> <xmpMetadataPathFileName (can be "">
<pdfMetadataPathFileName (can be ""> <pageLabel (can be "">
```

command sample:

```
CreatePDF_SinglePage_widthHiddenText "D:\\dWExport\\Samples\\T01\\IMG\\T01-0001.tif" "D:\\dWExport\\Samples\\T01\\ALTO\\T01-0001.xml"
"D:\\dWExport\\Samples\\T01\\T01-PDF.pdf" "D:\\dWExport\\Samples\\T01\\T01-xmpMetadata.xml" "D:\\dWExport\\Samples\\T01\\T01-pdfMetadata.xml" "page 1"
```

### Create single page PDFs WITHOUT hidden text

command syntax:

```
CreatePDF_SinglePage_noHiddenText <imgPathFilename> <resultPdfPathFilename>
<xmpMetadataPathFileName (can be "")> <pdfMetadataPathFileName (can be "")>
<pageLabel (can be "")> <width in 72inch> <height in 72inch>
<sourceImgResolution>
```

command sample:

```
CreatePDF_SinglePage_noHiddenText "D:\\dWExport\\Samples\\T01\\IMG\\T01-0001.tif" "D:\\dWExport\\Samples\\T01\\T01-PDF.pdf"
"D:\\dWExport\\Samples\\T01\\T01-xmpMetadata.xml"
"D:\\dWExport\\Samples\\T01\\T01-pdfMetadata.xml" "page 1" "7.00" "15.00"
"300"
```

### 16.2.36.3 Multi Page PDFs

Create multipage PDF by getting the default mets file and the associates images and alto files (the metadata files has to been generated before).

### Creating multipage PDFs with pdfMetadata as single parameter strings

command syntax:

```
CreatePDF_MultiPage_metadataString2 <metsPathFilename>
<resultPdfPathFilename> <xmpMetadataPathFileName(can be "")> <title(can be
"")> <subject(can be "")> <author(can be "")> <creationdate(can be "")>
<modificationdate(can be "")> <creator(can be "")> <keywords(can be "")>
<producer(can be "")>
```

command sample:

```
CreatePDF_MultiPage_metadataString2 "D:\\dWExport\\Samples\\T01\\T01-METS.xml" "D:\\dWExport\\Samples\\T01\\T01-PDF.pdf"
"D:\\dWExport\\Samples\\T01\\T01-xmpMetadata.xml" "My title" "My subject"
"My author" "D:200712240000+00'00'" "D:200712240000+00'00'" "My creator"
"My keywords" "My producer"
```

### Creating multipage PDFs with pdfMetadata as single parameter strings (downgraded function interface without 'modificationdate' and 'producer')

command syntax:

```
CreatePDF_MultiPage_metadataString <metsPathFilename>
<resultPdfPathFilename> <xmpMetadataPathFileName (can be "")> <title(can be
"")> <subject(can be "")> <author(can be "")> <creationdate(can be "")>
<creator(can be "")> <keywords(can be "")>
```

command sample:

```
CreatePDF_MultiPage_metadataString "D:\\dWExport\\Samples\\T01\\T01-METS.xml" "D:\\dWExport\\Samples\\T01\\T01-PDF.pdf"
"D:\\dWExport\\Samples\\T01\\T01-xmpMetadata.xml" "My title" "My subject"
"My author" "D:200712240000+00'00'" "My creator" "My keywords"
```

## Creating multipage PDFs with pdfMetadata given inside an xmlFile

command syntax:

```
CreatePDF_MultiPage_metadataFile <metsPathFilename> <resultPdfPathFilename>  
<xmpMetadataPathFileName (can be "")> <pdfMetadataPathFileName (can be "")>
```

command sample:

```
CreatePDF_MultiPage_metadataFile "D:\\dWExport\\Samples\\T01\\T01-METS.xml"  
"D:\\dWExport\\Samples\\T01\\T01-PDF.pdf" "D:\\dWExport\\Samples\\T01\\T01-  
xmpMetadata.xml" "D:\\dWExport\\Samples\\T01\\T01-pdfMetadata.xml"
```

## 16.3 Sample PDF configuration file

```
<PDFConfig:PDFConfig xmlns:PDFConfig="PDFConfig-namespace-URI" >

<PDFConfig:PDFGenerationConfig>
  <PDFVersionNumber>1.4</PDFVersionNumber>
  <HiddenText>1</HiddenText>
  <PartialImages>0</PartialImages>
  <GenerateBookmarks>1</GenerateBookmarks>
  <SinglePagePDF>0</SinglePagePDF>
  <PageMode>0</PageMode>
  <PageLayout>0</PageLayout>
  <OpenAction>0</OpenAction>
  <PDFBookmarkLanguage>EN</PDFBookmarkLanguage>
  <PDFBookmarkCoords>0</PDFBookmarkCoords>
  <LinearizedPDF>1</LinearizedPDF>
  <UserAccessPermissions>0</UserAccessPermissions>
  <ImageResolution>300</ImageResolution>
  <PartialImageResolution>300</PartialImageResolution>
  <ImageQualityJPG>0</ImageQualityJPG>
  <ImageQualityJP2K>7</ImageQualityJP2K>
  <UseBWPages>0</UseBWPages>
  <AddPageThumbnail>0</AddPageThumbnail>
  <AddPageLabel prefix="">0</AddPageLabel>
  <FontType>0</FontType>
  <TextRenderingMode>3</TextRenderingMode>
  <TextObjectsAsAnsi>0</TextObjectsAsAnsi>
  <UseCustomizedSettings>0</UseCustomizedSettings>
</PDFConfig:PDFGenerationConfig>

  <PDFConfig:PDFBookmarkTypes>
    <!-- default language is 'EN', if configured language (above) doesn't
    exist, the default will be taken -->
    <!-- if no languages are defined, the type itself will be inserted into
    bookmark -->
    <PDFConfig:PDFBookmarkType type="METAe_Serial" generateIfEmpty="1"
    useLabel="1" EN="Serial" FR2="Priodique" />
    <PDFConfig:PDFBookmarkType type="METAe_Monograph" generateIfEmpty="1"
    useLabel="1" EN="Monograph" FR2="Monographie" />
    <PDFConfig:PDFBookmarkType type="METAe_Multivolume_Monograph"
    generateIfEmpty="1" useLabel="1" EN="Multivolume Monograph" FR2="Monographie
    plusieurs volumes" />
    <PDFConfig:PDFBookmarkType type="Newspaper" generateIfEmpty="1"
    useLabel="1" EN="Newspaper" DE="Zeitung" FR="Journal" FR2="Journal" />
  </PDFConfig:PDFBookmarkTypes>
</PDFConfig>
```



```

<PDFConfig:PDFBookmarkType type="ISSUE" generateIfEmpty="1" useLabel="1"
EN="Issue" DE="Ausgabe" FR="Édition" FR2="Édition" />

<PDFConfig:PDFBookmarkType type="VOLUME" generateIfEmpty="1" useLabel="1"
EN="Volume" DE="Band" FR="Volume" FR2="Volume" />

<PDFConfig:PDFBookmarkType type="FRONT" generateIfEmpty="1" useLabel="1" />

<PDFConfig:PDFBookmarkType type="MAIN" generateIfEmpty="1" useLabel="1" />

<PDFConfig:PDFBookmarkType type="BACK" generateIfEmpty="1" useLabel="1" />

<PDFConfig:PDFBookmarkType type="ABBREVIATIONS" generateIfEmpty="1"
useLabel="1" EN="Abbreviations" DE="Abk?rzungen" FR="Abréviations"
FR2="Abréviations" />

<PDFConfig:PDFBookmarkType type="ABSTRACT" generateIfEmpty="1" useLabel="1"
EN="Abstract" DE="Zusammenfassung" FR="Résumé" FR2="Résumé, précis" />

<PDFConfig:PDFBookmarkType type="ACKNOWLEDGEMENTS" generateIfEmpty="1"
useLabel="1" EN="Acknowledgements" DE="Danksagung" FR="Remerciements"
FR2="Remerciements" />

<PDFConfig:PDFBookmarkType type="ADVERTISEMENT" generateIfEmpty="1"
useLabel="1" EN="Advertisement" DE="Anzeige" FR="Annonce" FR2="Annonce" />

<PDFConfig:PDFBookmarkType type="ADVERTISEMENT_PAGE" generateIfEmpty="1"
useLabel="1" EN="Advertisement" DE="Anzeige" FR="Annonce" FR2="Annonce" />

<PDFConfig:PDFBookmarkType type="APPENDIX" generateIfEmpty="1" useLabel="1"
EN="Appendix" DE="Anhang" FR="Annexe" FR2="Supplément" />

<PDFConfig:PDFBookmarkType type="ARTICLE" generateIfEmpty="1" useLabel="1"
EN="Article" DE="Artikel" FR="Article" FR2="Article" />

<PDFConfig:PDFBookmarkType type="BASTARD_TITLE_PAGE" generateIfEmpty="1"
useLabel="1" EN="Bastard title page" DE="Titel" FR="Titre" FR2="Faux titre"
/>

<PDFConfig:PDFBookmarkType type="BASTARD_TITLE_SECTION" generateIfEmpty="1"
useLabel="1" EN="Bastard title section" DE="Titel" FR="Titre" FR2="Faux
titre" />

<PDFConfig:PDFBookmarkType type="BIBLIOGRAPHY" generateIfEmpty="1"
useLabel="1" EN="Bibliography" DE="Literaturverzeichnis" FR="Bibliographie"
FR2="Bibliographie" />

<PDFConfig:PDFBookmarkType type="BLANK_PAGE" generateIfEmpty="1"
useLabel="1" EN="Blank page" DE="Leerseite" FR="Page blanche" FR2="Page
blanche" />

<PDFConfig:PDFBookmarkType type="BOOK_REVIEW" generateIfEmpty="1"
useLabel="1" EN="Bookreview" DE="Buchbesprechung" FR="Compte rendu de
lecture" FR2="Compte rendu de lecture" />

<PDFConfig:PDFBookmarkType type="CHAPTER" generateIfEmpty="1" useLabel="1"
EN="Chapter" DE="Kapitel" FR="Chapitre" FR2="Chapitre" />

<PDFConfig:PDFBookmarkType type="CONTRIBUTION" generateIfEmpty="1"
useLabel="1" EN="Contribution" DE="Artikel" FR="Contribution"
FR2="Contribution" />

<PDFConfig:PDFBookmarkType type="CORRECTIONS" generateIfEmpty="1"
useLabel="1" EN="Corrections" DE="Korrekturen" FR="Corrections"
FR2="Corrections" />

<PDFConfig:PDFBookmarkType type="COVER_SECTION" generateIfEmpty="1"
useLabel="1" EN="Cover page" DE="Umschlag" FR="Jaquette" FR2="Couverture" />

<PDFConfig:PDFBookmarkType type="DEBATE" generateIfEmpty="1" useLabel="1"
EN="Debate" DE="Debatte" FR="Débat" FR2="Débat" />

<PDFConfig:PDFBookmarkType type="DEDICATION" generateIfEmpty="1"
useLabel="1" EN="Dedication" DE="Widmung" FR="Dédicace" FR2="Dédicace" />

```

```

    <PDFConfig:PDFBookmarkType type="FRONTISPIECE" generateIfEmpty="1"
useLabel="1" EN="Frontispiece" DE="Abbildung" FR="Frontispice"
FR2="Frontispice" />

    <PDFConfig:PDFBookmarkType type="GROUP" generateIfEmpty="1" useLabel="1"
EN="Group" DE="Abschnitt" FR="Section" FR2="Section" />

    <PDFConfig:PDFBookmarkType type="ILLUSTRATION" generateIfEmpty="1"
useLabel="1" EN="Illustration" DE="Abbildung" FR="Figure" FR2="Illustration"
/>

    <PDFConfig:PDFBookmarkType type="INDEX" generateIfEmpty="1" useLabel="1"
EN="Index" DE="Index" FR="Index" FR2="Index" />

    <PDFConfig:PDFBookmarkType type="INTRODUCTION" generateIfEmpty="1"
useLabel="1" EN="Introduction" DE="Einleitung" FR="Introduction"
FR2="Introduction" />

    <PDFConfig:PDFBookmarkType type="LIST" generateIfEmpty="1" useLabel="1"
EN="List" DE="Liste" FR="Liste" FR2="Relevé" />

    <PDFConfig:PDFBookmarkType type="LIST_OF_FIGURES" generateIfEmpty="1"
useLabel="1" EN="List of Figures" DE="Abbildungsverzeichnis" FR="Table des
illustrations" FR2="Relevé des illustrations" />

    <PDFConfig:PDFBookmarkType type="NECROLOGY" generateIfEmpty="1"
useLabel="1" EN="Necrology" DE="Sterbeliste" FR="Avis d'obsèques" FR2="Relevé
des illustrations" />

    <PDFConfig:PDFBookmarkType type="DEATH_NOTICE" generateIfEmpty="1"
useLabel="1" EN="Death Notice" DE="Todesanzeige" FR="avis de décès" FR2="avis
de décès" />

    <PDFConfig:PDFBookmarkType type="NOTES" generateIfEmpty="1" useLabel="1"
EN="Notes" DE="Anmerkungen" FR="Annotacions" FR2="Annotacions" />

    <PDFConfig:PDFBookmarkType type="OBITUARY" generateIfEmpty="1" useLabel="1"
EN="Obituary" DE="Nachruf" FR="Nécrologie" FR2="Nécrologie" />

    <PDFConfig:PDFBookmarkType type="PICTURE_PAGE" generateIfEmpty="1"
useLabel="1" EN="Picture page" DE="Abbildung" FR="Figures" FR2="Illustration"
/>

    <PDFConfig:PDFBookmarkType type="PICTURE_SECTION" generateIfEmpty="1"
useLabel="1" EN="Picture section" DE="Abbildung" FR="Figures"
FR2="Illustration" />

    <PDFConfig:PDFBookmarkType type="REFERENCE_LIST" generateIfEmpty="1"
useLabel="1" EN="Reference list" DE="Quellenverzeichnis" FR="Liste de
référence" FR2="Liste de référence" />

    <PDFConfig:PDFBookmarkType type="RIDAS" generateIfEmpty="1" useLabel="1"
EN="Ridas" DE="Verschiedenes" FR="Autre" FR2="Autre" />

    <PDFConfig:PDFBookmarkType type="PREFACE" generateIfEmpty="1" useLabel="1"
EN="Preface" DE="Vorwort" FR="Préface" FR2="Préface" />

    <PDFConfig:PDFBookmarkType type="POEM" generateIfEmpty="1" useLabel="1"
EN="Poem" DE="Gedicht" FR="Poème" FR2="Poème" />

    <PDFConfig:PDFBookmarkType type="POSTFACE" generateIfEmpty="1" useLabel="1"
EN="Postface" DE="Nachwort" FR="Postface" FR2="Postface" />

    <PDFConfig:PDFBookmarkType type="SECTION" generateIfEmpty="1" useLabel="1"
EN="Section" DE="Abschnitt" FR="Section" FR2="Section" />

    <PDFConfig:PDFBookmarkType type="STATEMENT_PAGE" generateIfEmpty="1"
useLabel="1" EN="Statement page" DE="Stellungnahme" FR="Prologue"
FR2="Prologue" />

    <PDFConfig:PDFBookmarkType type="STATEMENT_SECTION" generateIfEmpty="1"
useLabel="1" EN="Statement section" DE="Stellungnahme" FR="Prologue"
FR2="Prologue" />

```

```

    <PDFConfig:PDFBookmarkType type="SUPPLEMENT" generateIfEmpty="1"
useLabel="1" EN="Supplement" DE="Beilage" FR="Supplément" FR2="Supplément" />

    <PDFConfig:PDFBookmarkType type="TABLE" generateIfEmpty="1" useLabel="1"
EN="Table" DE="Tabelle" FR="Tableau" FR2="Tableau" />

    <PDFConfig:PDFBookmarkType type="TABLE_OF_CONTENTS" generateIfEmpty="1"
useLabel="1" EN="Table of content" DE="Inhalt" FR="Sommaire" FR2="Table des
matières" />

    <PDFConfig:PDFBookmarkType type="TEXTSECTION" generateIfEmpty="1"
useLabel="1" EN="Textsection" DE="Textabschnitt" FR="Section de texte"
FR2="Section de texte" />

    <PDFConfig:PDFBookmarkType type="TITLE_PAGE" generateIfEmpty="1"
useLabel="1" EN="Title page" DE="Titel" FR="Couverture" FR2="Page de titre"
/>

    <PDFConfig:PDFBookmarkType type="TITLE_SECTION" generateIfEmpty="1"
useLabel="1" EN="Title" DE="Titelsektion" FR="Titre" FR2="Titre" />

</PDFConfig:PDFBookmarkTypes>

</PDFConfig:PDFBookmarkTypes>
- <PDFConfig:PDFGroupBookmarks generate="1" groupBookmarksFirst="0">
- <!--
    attribute 'generate' enables / disables the group bookmarks on top level
        0 = disabled;
        1 = enabled;
    -->
- <!--
    attribute 'groupBookmarksFirst' defines, whether the logical struct
bookmarks or the group bookmarks should be the first in bookmark list
sequence
        0 = disabled;          (default)
        1 = enabled;
    -->
- <!--
    each 'PDFGroup' node represent one group node in the result PDF
        the subnodes 'PDFConfig:PDFChild' contains in the attribute 'type'
those export types,
        which should be inserted in that group

        attribute 'generateIfEmpty' configures, if the group element should
be generated if no
        no bookmarks were generated inside that group
        0 = not generated if contains no bookmarks;
        1 = generated even if no bookmarks inside;
    -->
- <PDFConfig:PDFGroup name="Advertisements" generateIfEmpty="0" useLabel="1"
EN="Advertisements" DE="Werbung" FR="Annonce" FI="Ilmoitus" SE="Annonser">

    <PDFConfig:PDFChild type="ADVERTISEMENT" useLabel="1" EN="Advertisement"
DE="Anzeige" FR="Annonce" />

```

```

    </PDFConfig:PDFGroup>
- <PDFConfig:PDFGroup name="Illustrations" generateIfEmpty="0" useLabel="1"
EN="Illustrations" DE="Abbildungen" FR="Figure" FI="Kuvat" SE="Bilder">
    <PDFConfig:PDFChild type="ILLUSTRATION" useLabel="1" EN="Illustration"
DE="Abbildung" FR="Figure" />
    </PDFConfig:PDFGroup>
- <PDFConfig:PDFGroup name="Tables" generateIfEmpty="0" useLabel="1"
EN="Tables" DE="Tabellen" FR="Tableau" FI="Taulukot" SE="Tabeller">
    <PDFConfig:PDFChild type="TABLE" useLabel="1" EN="Table" DE="Tabelle"
FR="Tableau" />
    </PDFConfig:PDFGroup>
- <!--
        no of groups and child elements is extensible
-->
    </PDFConfig:PDFGroupBookmarks>

</PDFConfig:PDFConfig>

```

## 16.4 Sample File metadata.xml

### metadata.xml

```

<?xml version="1.0" encoding="UTF-8" ?>
- <ROOT>
- <METADATA>
    <TITLE>title</TITLE>
    <SUBJECT>subject</SUBJECT>
    <AUTHOR>author</AUTHOR>
    <CREATIONDATE>creationdate</CREATIONDATE>
    <MODIFICATIONDATE>modificationdate</MODIFICATIONDATE>
    <CREATOR>creator</CREATOR>
    <KEYWORDS>keywords</KEYWORDS>
    <PRODUCER>producer</PRODUCER>
  </METADATA>
</ROOT>

```

## 16.5 Sample File XMP Metadata

**XMP metadata -> "0001.xml"**

```
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:ix="http://ns.adobe.com/ix/1.0/">
- <rdf:Description rdf:about="uuid:813e0307-eef2-4b52-8fd2-902bfade3ca3"
xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:format>application/pdf</dc:format>
    - <dc:title>
        - <rdf:Alt>
            <rdf:li xml:lang="en">The San Francisco Call (San Francisco)
1900-01-01 [p ]</rdf:li>
            </rdf:Alt>
        </dc:title>
    - <dc:description>
        - <rdf:Alt>
            <rdf:li xml:lang="en">Page from The San Francisco Call
(newspaper). [See LCCN: sn85066387 for catalog record.]. Prepared on behalf
of University of California, Riverside.</rdf:li>
            </rdf:Alt>
        </dc:description>
    - <dc:date>
        - <rdf:Seq>
            <rdf:li xml:lang="x-default">1900-01-01</rdf:li>
            </rdf:Seq>
        </dc:date>
    - <dc:type>
    - <rdf:Bag>
        <rdf:li xml:lang="en">text</rdf:li>
        <rdf:li xml:lang="en">newspaper</rdf:li>
    </rdf:Bag>
    </dc:type>
    - <dc:identifier>
        - <rdf:Alt>
            <rdf:li xml:lang="en">Reel number 00100478407. Sequence number
6.</rdf:li>
            </rdf:Alt>
        </dc:identifier>
</rdf:Description>
</rdf:RDF>
```

## 17 EPUB Configuration

For the publication of e-books (ePUB) a special configuration is required.

In order to enable EPUB export, a child node called <EPUB> needs to be added under the <EXPORT> node from import RDY file. See "Generating EPUB in export" subchapter.

```
project-cfg\Default\rdy\Monograph.xml
<EXPORT TYPE="">
    <EPUB/>
</EXPORT>
```

Project based epub configuration is done in `project-cfg\Default\epub\EPUBConfig.xml`.

See **EPUB configuration file** subchapter.

An original epub xml file is stored under: `script\epub\EPUBConfig.xml`

Please be aware of the renamed steps and tasks and the "old" names in scripts and configuration files.

### 17.1 EPUB Generating export

In order to enable EPUB export, a child node called <EPUB> needs to be added under the <EXPORT> node from import RDY file.

Document is corrected with the rules specified at first section (**EPUB document specific correction rules**).

The hierarchy is parsed and the EPUB generation starts according to the **EPUBConfig.xml** (defined later, in **EPUB configuration file** section) configuration file.

Issue level metadata will be transferred to the EPUB.

An EPUB file will be generated next to the mets file with the same name as it.

## 17.2 EPUB configuration file

An xml file that needs to be placed in project/epub directory (ie. project-cfg/Default/epub/) and named EPUBConfig.xml.

If not found there, the one from Default project cfg will be used.

XML file is splitted in 2 sections:

**<epub:images/>** - where the EPUP contained images compression is configured; (other valid values are: "gif" and "png")

```
<epub:images format="png" jpegQuality="70"/> <!-- format="gif" , "png" or "jpeg" -->
```

**<epub:bookedTypes>** - where structures that are going to be exported are configured.

See examples:

```
<epub:bookmark type1="BLANK_PAGE" type2="BlankPage" EN="Blank page" DE="Leerseite"/>
```

```
<epub:bookmark type1="BOOKREVIEW" type2="BookReview" EN="Bookreview" DE="Buchbesprechung"/>
```

```
<epub:bookmark type1="CHAPTER" type2="Chapter" EN="Chapter" DE="Kapitel"/>
```

```
<epub:bookmark type1="TEXTSECTION" type2="TextSection" EN="Text section" DE="Abschnitt"/>
```

```
<epub:bookmark type1="TITLE_PAGE" type2="TitlePage" EN="Title page" DE="Titel" align="center"/>
```

```
<epub:bookmark type1="TITLE_SECTION" type2="Title" EN="Title" DE="Titel" align="center"/>
```

"type" attribute contains structure's type as it is exported (in METS file)

"EN" and "DE" represents titles to be displayed in Table of Contents section, if not title found

"align" attribute represents the alignment for all the zones included in that structure (left, right, center, justify (default))

If in docWizz there is no title the `type2` element will be taken.

## 17.3 Generating EPUB in postProcess (METS, ALTO and page images)

Script that works on old exports.

METS and all ALTO files are needed.

Page images are needed only if there are any Illustrations or Tables zones.

Also, if there are Illustrations or Tables and partial images are generated, the Page Images are not needed anymore.

METS file should have been generated with Logical Struct section from which EPUB is splitted by Chapters and Table of contents is generated.

METS's Logical Structure is parsed and EPUB generation starts according to the **EPUBConfig.xml** (defined later, in **EPUB configuration file** section) configuration file.

Issue level metadata will be transferred to the EPUB.

An EPUB file will be generated next to the mets file with the same name as it.

## 17.4 EPUB document specific correction rules

If possible, first and last pages of the document (covers) should contain a full page illustration; (this way, the EPUB's first and last pages will be represented by illustrations).

Headlines and Authors OCR needs to be corrected.

Issues/Hierarchy MODS metadata should be corrected/added: Title, Author, publisher, publishing date or any other metadata info that can be retrieved from document's content.

Add GraphicalText zone type to zones that contain text but their appearance is somehow special (a signature, a handwritten text area or any other zone that interests also as OCR text and also as illustration).

Add GraphicalTable zone type when current area represents a table which's logically related to / referenced in one of the nearest text blocks from which should not be separated by moving to the end of the Chapter in a TableStruct. Chapter structure will include it in Content right after/before the text block that references it and so the logic of the document is not damaged.

GraphicalIllustration was created following the same idea as for GraphicalTable. Represents an illustration area that it is related to some closed blocks and from which should not be separated in hierarchy (moved to IllustrationStruct) to keep the logical order of the structure. Chapter structure will also include this area in Content next to the block that references it and so the logic of the document is not damaged.

TextByLines zone is used to mark zones for which the line-by-line representation it is important (i.e. a poem or an enumeration).

Poem structure should be used to group text zones representing part of a poem (Note that all those zones will be splitted line by line in the output).

Marginalia represents a few words text area on the left or on the right side of an paragraph; usually contains a short summary of the paragraph; an alternative to Marginalia zone type would be marking zone as text block and grouping it in a Paragraph before the Paragraph to which is related.

There are no other differences between normal processing and documents processed for an EPUB output.



## 17.5 EPUB on various devices

### EPUB on iPad/iPhone

You need to have an EPUB reader installed on your mobile device. There are a lot of apps available at the iTunes store. All eBook readers supporting ePUB can open the eBooks docWizz currently generates. In this manual, we're using iBooks which is free of charge available at Apple iTunes.

### EPUB on Android

This description will work on all devices which have an Android operating system installed. A list of devices with Android operating system can be found at the end of this chapter.

You need to have an EPUB reader installed on your mobile device. There are a lot of apps available at the Android Market. All of them are working. In this manual, we're using eBook which is pre-installed on your device or free of charge available at Android Market. For the connection of Android device and computer, AirDroid (also free of charge available at the Android Market) will be used.

Reference list for phones with Android operating system:

Manufacturer	Telephone
Acer	beTouch E110, beTouch E120, beTouch E130, beTouch E400, Liquid E, Liquid mt, Liquidmini, Stream
Commtiva	X1, X3, X5, Z51, Z71
Dell	Mini 3, Blaze, Thunder
Garmin	Asus A10, Nüvifone A50
Geeksphone	One, Zero
General Mobile	DSTL1 Imaginary, Fox, Fenomen, Ultimate Slim 4GB
Google	Galaxy Nexus, Nexus One, Nexus S
HTC	ChaCha, Desire, Desire HD, Desire S, Desire Z (T-Mobile G2), Dream (T-Mobile G1), Droid Incredible, EVO 3D, EVO 4G, Explorer, Gratia (USA: Aria), Hero (T-Mobile G2 Touch), Incredible S, Legend, Magic, Rezound, Rhyme, Salsa, Sensation, Sensation XE, Sensation XL, Tattoo, Tide, Wildfire, Wildfire S
Huawei	Joy, U8120 (Vodafone 845), U8150 IDEOS, U8160 (Vodafone 858 Smart), U8220 (T-Mobile Pulse), U8230, U8510 (IDEOS X3), U8650 (Sonic)
Inbrics	M1
Kyocera	Echo, Zio M6000
Lenovo	LePhone
LG	Ally, E720 Optimus Chic, GT540 Optimus, GW620, Optimus Z, Optimus Q, P350 Optimus Me, P500 Optimus One, P920 Optimus 3D, P970 Optimus Black, P990 Optimus Speed (USA: T-Mobile G2X, Italien: LG Optimus Dual, International: LG Optimus 2X)
Meizu	M9
Motorola	ATRIX, BACKFLIP, CLIQ, DEFY, DEFY+, DROID X, FIRE, FLIPOUT, Milestone (USA: DROID), Milestone XT720, Milestone 2, Pro (USA: DROID Pro), Pro+, RAZR (USA: DROID RAZR)
Panasonic	Eluga

Samsung	Galaxy Ace GT-S5830, Beam GT-I8520, Galaxy Fit GT-S5670, Galaxy Gio GT-S5660, Galaxy GT-I899, Galaxy GT-I7500, Galaxy mini GT-S5570, Galaxy Note GT-N7000, Galaxy Pro GT-B7510, Galaxy S GT-I9000, Galaxy S Plus GT-I9001, Galaxy S II GT-I9100, Galaxy GT-I9003, GT-I6500 Saturn, Galaxy Spica GT-I5700 (UK: Portal), Galaxy W GT-I8150, Galaxy Xcover GT-S5690, Galaxy Y GT-S5360, Galaxy Y Pro GT-B5510, Galaxy 3 GT-I5800 (UK: Apollo), Galaxy 5 GT-I5500 (UK: Europa), Galaxy 551 GT-I5510
Sony Ericsson	Live mit Walkman, Xperia active, Xperia arc, Xperia arc S, Xperia mini, Xperia mini pro, Xperia neo, Xperia neo V, Xperia PLAY, Xperia pro, Xperia ray, Xperia S, Xperia X8, Xperia X10, Xperia X10 Mini, Xperia X10 Mini pro
Umeox (Simvalley)	(SP-40 EDGE), A502 (SP-60 GPS), A602 (SP-80 3G), X-LAND (SPX-5 UMTS)
ZTE	Blade (BASE Lutea), Libra, Skate (BASE Lutea 2), Racer, Crescent

Reference list for tablets with Android operating system:

Acer Iconia Tab A100, A101	Pearl Touchlet X3
Acer Iconia Tab A500, A501	Pearl Touchlet X4
Amazon Kindle Fire	Pearl Touchlet X7G
Archos 28, 32, 43, 48, 70, 101[1. 1]	Pandigital Nova
Archos 70b eReader	Pandigital Novel 7??Pandigital Novel 9??Pandigital Planet
Archos 80 G9, 101 G9	Pandigital Star
Asus Eee Pad Memo	Pandigital SuperNova
Asus Eee Pad Slider	Point of View mobii
Asus Eee Pad Transformer Prime	Point of View mobii tegra
Asus Eee Pad Transformer TF101	Point of View playtab
Commtiva N700[1. 2]	Point of View playtab 2
Datawind Aakash	Point of View protab 2
Dell Streak 5	Samsung Galaxy Tab 7 GT-P1000
Dell Streak 7	Samsung Galaxy Tab 7 Plus
FUJITSU STYLISTIC M532	Samsung Galaxy Tab 7.7
General Mobile e-tab 32GB	Samsung Galaxy Tab 8.9
Hannspree Hannspad	Samsung Galaxy Tab 10.1
HTC Flyer	Samsung Galaxy Tab 10.1N
HTC Jetstream / Puccini	Samsung Galaxy Tab 10.1V
Huawei Media Pad	Sony Tablet P
Lenovo IdeaPad A1	Sony Tablet S
Lenovo IdeaPad Tablet K1	Toshiba Thrive AT-100
Lenovo ThinkPad Tablet	Toshiba Thrive AT-200
LG V900 Optimus Pad	ViewSonic ViewPad 4
Medion Lifetab P9514	ViewSonic ViewPad 7
Motorola Xoom	ViewSonic ViewPad 7x

Notion Ink Adam	ViewSonic ViewPad 10
Odys Genesis	ViewSonic ViewPad 10pro
Odys Space	ViewSonic ViewPad 10s
Odys Xtense	Yarvik Tab 210
Odys Loox	Yarvik Tab 211
Odys Xpress	Yarvik Tab 250
Odys Cosmo	Yarvik Tab 310
Packard Bell Liberty Tab [1. 3]	Yarvik Tab 410
Pearl Touchlet X2	

### **EPUB on Sony Portable Reader Device (PRS)**

This description will work on all Sony PRS devices. No additional software on mobile device or computer is needed.

## 18 Document definition

To create specific document definitions including specific zone types (e.g. birth notices, marriage notices, death notices...), page classes / feature codes (e.g. photopage, accessOKpage...) or structure types (e.g. weather, necrology...) there are several configuration files to create.

Typically the addition of zone types, feature codes or structure types is necessary for a specific project and thus should not be added to the global configuration.

So first you must create a new **Document Structure Definition**.

Then perform one or more of the steps described in the following chapters.

To change the color of how a certain zone type is displayed in the docWizz user interface, changes must be performed at global level editing the global.ini file. As we are not dealing with additionally created files, but with the global configuration file, changes might be overwritten when you update your system. So you might want to save the global.ini file for future references.

Please be aware of the renamed steps and tasks and the "old" names in scripts and configuration files.

### 18.1 Document structure definition

First of all a document structure definition is created. It always starts from a basic structure definition, e.g. Newspaper, Monograph or Serial. So, copy one of the basic Document Structure Definition files that match your documents. They are located at

```
\\***data***\doccfg\ .
```

Copy and paste it into **\*\*\*PROJECT\_CFG\*\*\*\projectname\dsl** directory and rename it into **newdocumentname.dsd** (e.g. TEST-Newspaper, CCS-Monograph, etc.).

Then special zones types, page classes or structure types can be defined in such a document structure definition file.

To do so, edit the following file according to the project for which you want to use the changes:

```
<Import File="***PROJECT_CFG***\\projectname\\dsl\\newdocumentname-  
dwcfg.dsl"/>
```

Typically the imported files link to the default configuration, but the changes outlined above show links to a project specific configuration. Therefore the according project configuration needs to be adapted so that the file

```
\\***project-cfg***\projectname\projectname-DSD.txt
```

contains the value **newdocumentname**. If the file does not exist it needs to be created. Simply create a \*.txt file and enter the document structure definition, e.g. **newdocumentname**.

Additionally, edit the following file:

```
\\***project-cfg***\projectname\RDY\projectname.xml
```

so that it contains the line

```
<DESCRIPTION TYPE="newdocumentname">
```

You may place dsd files in project/dsl folder. The system searches first in this folder and if not found then it searches in dsd folder.

In "Scan" and "PrepareImport" job, the list of available doctypes is filled as follows:

1. check for project-dsd.txt
2. if empty, check for doctypes from project/dsl
3. if empty, check for all doctypes in doccfg

## 18.2 Create custom classes file

Create a new file named `***PROJECT_CFG***\\projectname\\dsl\\newdocumentname-dwcfg.dsl` that should look like (an empty dsl file):

```
<?xml version="1.0" encoding="UTF-8"?>
<docWizz-Definition Version="4.0" xmlns:xlink="http://www.w3.org/TR/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="LogicalStruct.xsd">
</docWizz-Definition>
```

## 18.3 Create specific page classes (feature codes)

In docWizz you can specify what page classes will be available for a project. Operators verify page classes (or: Feature Codes) in VerifyPageNumbers.

Examples for page classes are 'Title Page', 'TOC Page' or 'Blank Page'. The default page classes are stored in

```
\\***project-cfg***\\default\\ds\\META-E-PageClasses (for Monographs and Serials)
\\***project-cfg***\\default\\ds\\Newspaper-PageClasses (for Newspapers).
```

To change the default set of values into a project specific one add the following into

```
***PROJECT_CFG***\\projectname\\dsl\\newdocumentname-dwcfg.dsl
```

You can add a new page classes by editing the file as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<docWizz-Definition Version="4.0" xmlns:xlink="http://www.w3.org/TR/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="LogicalStruct.xsd">
    <Step Name="VerifyLayoutElements">
        <Classes>
            <Class Name="NewPageClass" BaseClass="imagePage"/>
        </Classes>
    </Step>
</docWizz-Definition>
```

All class names need a BaseClass. Please use "imagePage" as a standard text page. A BaseClass enables docWizz to detect specific elements automatically applying certain detection rules on a page. Other available BaseClasses are the Class Names referenced in file

```
***CONFIG***\\BasicCfg\\META-E-Basic-PageClasses
and
***CONFIG***\\BasicCfg\\newspaper-Basic-PageClasses
```

## 18.4 Create specific zone types

To create specific zone types into project specific configuration add the following into

```
***PROJECT_CFG***\\projectname\\dsl\\newdocumentname-dwcfg.dsl
```

```
<?xml version="1.0" encoding="UTF-8"?>

<docWizz-Definition Version="4.0" xmlns:xlink="http://www.w3.org/TR/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="LogicalStruct.xsd">

    <Step Name="VerifyLayoutElements">
        <Classes>
            <Class Name="NewPageClass" BaseClass="imagePage"/>
            <Class Name="SpecialTextBlock" BaseClass="Textblock"/>
            <Class Name="Cartoon" BaseClass="Illustration"/>
        </Classes>
    </Step>
</docWizz-Definition>
```


BaseClass defines which sort of the zone type it is. If it is a specific text entity it would be of type "Textblock". If it is a kind of illustration (e.g. Cartoon, Photo) it would be of BaseClass "Illustration".

## 18.5 Change sequence of zone types

Sequence of zone types can only be changed, if a user is logged in having docWizz SYSTEMCFG rights.  
Default Admin user has those rights.

- Click on "Default zone type" button.



- Select the zone, from "Default type" list, and click .
- Select the zone type, from "Sequence of types" list, drag and drop it to the desired position.



## 18.6 Create specific structure or hierarchy types

To create a new structure class add the following into

```
***PROJECT_CFG***\\projectname\\dsl\\newdocumentname-dwcfg.dsl
```

```
<?xml version="1.0" encoding="UTF-8"?>

<docWizz-Definition Version="4.0" xmlns:xlink="http://www.w3.org/TR/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="LogicalStruct.xsd">

    <Step Name="VerifyLayoutElements">
        <Classes>
            <Class Name="NewPageClass" BaseClass="imagePage"/>
            <Class Name="SpecialTextBlock" BaseClass="Textblock"/>
            <Class Name="Cartoon" BaseClass="Illustration"/>
        </Classes>
    </Step>

    <Step Name="VerifyHierarchy">
        <Classes>
            <Class Name="Shipping" BaseClass="ArticleLike"/>
        </Classes>
    </Step>
</docWizz-Definition>
```



## 18.7 Defining specific colors for zone types

Defining specific colors for the use of zone types in docWizz user interface is done in

```
\\***data***\config\<docWizz-GLBL.ini
```

Edit the section [CLASSES] so that it looks like:

```
DISABLED=21,19,22,10,12,8,13
```

```
BookHeadline=ffe6cc;
```

```
Textblock=00ffff;
```

```
ParagraphBlock=00ffff;
```

```
PictureText=0000ff;
```

```
FramedTable=ffaaff;
```

```
Table=ffaaff;
```

```
Marginalia=ff8000;
```

```
Formula=ff00ff;
```

```
Motto=80ffff;
```

```
Illustration=008000;
```

```
ItemCaption=80ff80;
```

```
newzonetype=80ffff;
```

```
PageNumber=0073e6;
```



For defining zone type colors there is no need to create a new document type. The changes in this file affect the whole system and are not project specific. Thus, the choice of colors and also the addition of zone types even per project should be meaningful for the whole system. The default color for new zone types (if undefined) is red.

The color of default zones cannot be changed.

As we are not dealing with additionally created files, but with the global configuration file, changes might be overwritten when you update your system. So you might want to save the global.ini file for future references. Defining zone type colors does not work on <client>-docWizz-glbl.ini.

## 18.8 Sort zone types list

You hit "P" to change a zone to page number. The first zone on P is Portrait. If Page Number should be first, you can sort the list with pre-selected sorting (newZoneTypeDlg).

Function GetSortedTypes has a change (sort types by classes), which overwrites other changes. As well it has a mechanism to deal with translations to have correct sorting when using different UI language.

...\config\PVSCFG\Book-DW\_Types.xml and Newspaper\_DW\_Types.xml

contains all zone types.

Example:

...\config\PVSCFG\Book-DW\_Types.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<CONFIG>
```

```
    <STEP name="Re-Scan"/>
```

```
    <STEP name="VerifyHierarchy">
```

```
        <OrderedTypesArray>
```

```
            <OrderedType>Textblock</OrderedType>
```

```
            <OrderedType>Table</OrderedType>
```

```
            <OrderedType>Illustration</OrderedType>
```

```
            <OrderedType>Footnote</OrderedType>
```

```
            <OrderedType>Formula</OrderedType>
```

```
            <OrderedType>Barcode</OrderedType>
```

```
            <OrderedType>Map</OrderedType>
```

```
            <OrderedType>Motto</OrderedType>
```

```
            <OrderedType>Author</OrderedType>
```

```
            <OrderedType>Headline</OrderedType>
```

```
            <OrderedType>Caption</OrderedType>
```

```
            <OrderedType>ItemCaption</OrderedType>
```

```
            <OrderedType>PageNumber</OrderedType>
```

```
            <OrderedType>RunningTitle</OrderedType>
```

```
            <OrderedType>VerticalTextblock</OrderedType>
```

```
            <OrderedType>Marginalia</OrderedType>
```

```
            <OrderedType>Advertisement</OrderedType>
```

```
            <OrderedType>Subheadline</OrderedType>
```

```
            <OrderedType>BookBinding</OrderedType>
```

```
            <OrderedType>Chart/Diagram</OrderedType>
```

```
            <OrderedType>Bookplate</OrderedType>
```

```
            <OrderedType>CatchWord</OrderedType>
```

```
        </OrderedTypesArray>
```

```
    <NEW_ZONE>
```

```
        <HierarchyClass>PhysicalElement</HierarchyClass>
```

```
        <ScriptName>AddNewZone</ScriptName>
```

```

        <DefaultType>Textblock</DefaultType>
    </NEW_ZONE>
</STEP>
<STEP name="VerifyLayoutElements">
    <OrderedTypesArray>
        <OrderedType>Textblock</OrderedType>
        <OrderedType>Table</OrderedType>
        <OrderedType>Illustration</OrderedType>
    </OrderedTypesArray>
    <NEW_ZONE>
        <HierarchyClass>PhysicalElement</HierarchyClass>
        <ScriptName>AddNewZone</ScriptName>
        <DefaultType>Textblock</DefaultType>
    </NEW_ZONE>
</STEP>
</CONFIG>

```

Zones from "..\proj-cfg\XXX\ds\METAe-IgnoreTypes.dsl" are filtered out for list in zone change context menu in docWizz.

#### Example:

```

... \proj-cfg\Default\dsl\METAe-IgnoreTypes.dsl
<?xml version="1.0" encoding="UTF-8"?>
<docWizz-Definition Name="GenericExtensionVLE" Version="4.0"
xmlns:xlink="http://www.w3.org/TR/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="LogicalStruct.xsd">

    <!--this file need to be the last in order to make the types virtual-->
    <Step Name="VerifyLayoutElements">
        <Classes>
            <Class Name="Barcode" BaseClass="" Virtual="true"/>
            <Class Name="BirthNotice" BaseClass="" Virtual="true"/>
            <Class Name="BookBinding" BaseClass="" Virtual="true"/>
        </Classes>
    </Step>

    <Step Name="VerifyPageNumbers" BaseStep="VerifyLayoutElements">
        <Classes>
            <Class Name="Barcode" BaseClass="" Virtual="true"/>
            <Class Name="BirthNotice" BaseClass="" Virtual="true"/>
            <Class Name="BookBinding" BaseClass="" Virtual="true"/>
        </Classes>
    </Step>
</docWizz-Definition>

```

### **Book-DW\_types: New Elements and sorting order in file**

On one hand, changing types in Image views (mainly only in DImageV+FullScreen) is faster when using shortcuts. For example to change type from TextBlock -> Table one shall just type 't' letter once. But when having more types starting with t letter (TextBlock/Table/Title, etc) an order shall be clear. This is for doctype\_Types.xml

On the other hand, when changing type from OTreeV is usually more useful having related types grouped together. Common type groups that I have in mind now are: TextblockLike, TableLike (has Cells property), IllustrationLike (has Caption property). So in tree the order of types will be different that the order in image views.

1. In Image Views, right-click on zones (shall be upper steps that have zones there not Scan/C-RC/C-PC)
2. Tree views, right-click (also upper steps that have a hierarchy)

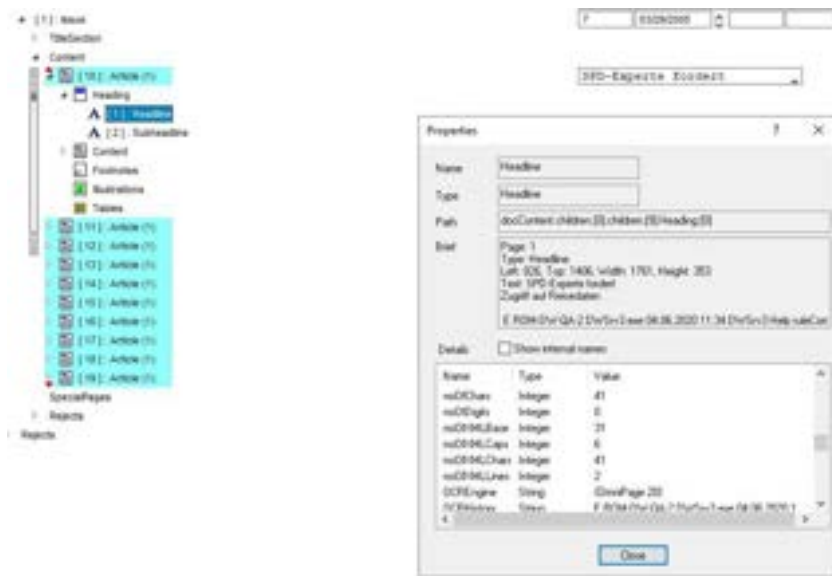
### **Configuration**

project-cfg\BLNP\PVSCFG\ProjName-DocType-DW\_Types.xml (e.g.: H:\dwcfs\generic\project-cfg\BLNP\PVSCFG\BLNP-Newspaper-DW\_Types.xml)

### **Default settings**

config\PVSCFG (different than project specific cfgs that are into project-cfg folder)

A right click to open the in context menu. Select Properties shows, among others, the OCR license used.



## 19.1 OCR configuration

**Goal and Benefits:** The OCR correction user will be able to insert in text a letter with diacritic without using the mouse or taking the hand off the keyboard.

Solution: Since the diacritic do not produce a new letter, we will group the letters under the same basic glyph that is found on the keyboard:

e é è ê ë ...

u ü ú ù û ...

On the first click the normal glyph will be inserted. If you press twice the second one, one more click and will be the third one ...

In order to help the user the list will be shown in the moment of the second press with the current rest of possibilities.

In order to optimize the letters list project base you should minimize it, not to be the same long list of possibilities that contain all the variants.

On the other hand we cannot configure it language base because the QA user will not learn the list and he will not be able to speed up (knowing in advance that 3-4 key press will produce the desired letter).

In conclusion we will configure a standard DW list in the default list `project-cfg\Default\PVSCFG\<doctype>-DW.xml` and will be possible to be overwritten in each project configuration as is considered appropriate.

The configurations will look like this:

```
<letter main='e' variants='éèêë...'>
```

```
<letter main='u' variants='üúùû...>
```

```
<letter main='ü' variants='uúûü'> (example for German keyboard)
```

### How to use in docWizz interface:

\*\*\*CONFIG\*\*\*\PVSCFG\doctype-DW.xml

[illegible]

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<CONFIG>
  <KEYBOARD>
    <KEY name="a" values="âáãäååååå?æ?" />
    <KEY name="c" values="çcccc" />
    <KEY name="d" values="dd" />
    <KEY name="e" values="èéêëèèèèèè" />
    <KEY name="f" values="f" />
    <KEY name="g" values="gggg" />
```

```

<KEY name="h" values="hh"/>
<KEY name="i" values="ìîïïïïïïï?"/>
<KEY name="j" values="j"/>
<KEY name="k" values="k?"/>
<KEY name="l" values="lll?l"/>
<KEY name="n" values="ñnnn??"/>
<KEY name="o" values="òóôõöøooooo?œ"/>
<KEY name="r" values="rrr"/>
<KEY name="s" values="sssš?ß"/>
<KEY name="t" values="ttt"/>
<KEY name="u" values="ùúûüuuuuuu"/>
<KEY name="w" values="w???"/>
<KEY name="y" values="ýÿÿ?"/>
<KEY name="z" values="zzž"/>
<KEY name="&quot;;" values="„“”»«"/>
</KEYBOARD>
<TIMEOUT>500</TIMEOUT>
</CONFIG>

```

## 19.2 OCR settings

A new OCR setting configuration is added to combine previous OCR settings with new special configuration features.

E.g. OCR settings by zone, enhanced images or dictionary support can easily be used now.

- Configurable special OCR settings and having all the OCR configurations in one place. (Previously only possible by CCS scripting adaptations).
- Better OCR detection by usage of external dictionaries.
- Possibility to set different OCR types in case of mixed content. (For example gothic OCR on zone types and on headlines and antia OCR on textblock zones.)

### Functionality

All settings regarding OCR are now defined in `project-cfg\Default\Default-OCRCfg.xml`.

These are disabled by default to keep backwards compatibility.

To activate project based this setting file has to be taken over to custom project configuration.

Note: As soon as these settings are enabled the settings from rdy file (previous solution) will be overwritten.

According to this the settings for OCR are taken in the following sequence:

1. Project configuration: New configuration file `PROJECT-CFG\<project>\<project>-OCRCfg.xml`
2. Default configuration: New configuration file `PROJECT-CFG\Default\Default-OCRCfg.xml`
3. Ready files (\*.rdy, as before. Backwards compatibility is kept).

If setting is not existing or not enabled the settings of the next level are taken.

### 19.2.1 Auto replacement not hard coded anymore

Hard coded replacements of previous releases are removed.

"spellcheck" command caused errors by replacing different characters (like: \$ and S to 5; B to 8; l to 1; detect Em and not Ein) for words that have number of digit chars greater or equal to the number of alphabetical chars or no alphabetical characters.

Clients can now individually define necessary replacements if needed. In some cases the hard coded implementation caused unexpected issues which could not be handled.

### Functionality

The auto replacement functionality is still available at the dictionaries settings (see [docWizz interface\Configuration\System\Dictionaries](#)).

If needed clients can define again auto replacements as existing before.

Note: Due to this change different results could be generated by docWizz. To keep same behavior above mentioned replacement have to be reactivated.



## 19.2.2 project-cfg\Default\Default-OCRCfg.xml

### 19.2.2.1 Font type and language settings

Similar to the previous configuration in the \*.rdy files the common OCR settings (FONTTYPE, LANGUAGE) are defined in

project-cfg\Default\Default-OCRCfg.xml.

```
<CLASS name="Import">
    <!-- ### mandatory OCR settings. Can not be disabled are set only in
    Import step### -->
    <SET name="FONTTYPE" enable="0">Antiqua</SET>
    <SET name="LANGUAGE" enable="0">German French</SET>
</CLASS>
```

#### FONTTYPE

Set the font type if a special script is configured. The merge will be done in the script.

The font type that is used in detection can be:

- Antiqua
- Fraktur
- Typewriter
- Fraktur\_Antiqua
- Automatic merge Fraktur+Antiqua
- Typewriter\_Antiqua
- Automatic merge Typewriter-Antiqua

#### LANGUAGE

Languages that are used in detection. The value is defined in other part of the configurations/code.

### 19.2.2.2 OCR related to the characters

```
<!-- ### special OCR set related to the characters ### -->
<SET name="USER_LANGUAGE" enable="0">LatvianLongS</SET>
<SET name="USE_EXTERNAL_DICTIONARY" enable="0">ExternDll.dll</SET>
<SET name="OLDLANGUAGE" enable="0" />
<SET name="CHARCONFIDENCE" enable="0" />
<SET name="DETAILED_CHARACTER_INFO" enable="0" />
<SET name="KEEP_SPACES" enable="0"/><!-- Not available in FR10/FR11 -->
```

#### USER\_LANGUAGE

In case a user language is defined in configuration, dW will use only that language for detection, ignoring the predefined languages.

Note: Do not combine user languages with generic languages since Abbyy FineReader will not perform auto-detection between a built-in language and an custom language. Instead of this, the first language defined into settings file (\*.rdy file) is taken into account. If you want to use an user language, set it as the only language into settings file.

#### USE\_EXTERNAL\_DICTIONARY

Dictionary file name for the current user language.

Note: Do not combine user languages with generic languages since Abbyy FineReader will not perform auto-detection between a built-in language and an custom language. Instead of this, the first language defined into settings file (\*.rdy file) is taken into account. If you want to use an user language, set it as the only language into settings file.

#### OLDLANGUAGE

Use old language in detection.

#### CHARCONFIDENCE

No value

#### DETAILED\_CHARACTER\_INFO

No value

#### KEEP\_SPACES

No value

#### 19.2.2.3 OCR related to the ABBYY detection using patterns

```
<!-- ### special OCR set related to the ABBYY detection using patterns ### ->
```

```
<SET name="USE_TRAINING" enable="0"/>
```

```
<SET name="USER_PATTERN" enable="0">PATTERN FILE</SET>
```

```
<SET name="USE_BUILT_IN_PATTERNS" enable="0"/>
```

```
<SET name="OCRMODE" enable="0">FASTMODE</SET>
```

#### USE\_TRAINING

No value

#### USE\_PATTERN

Use pattern file in detection

#### USE\_BUILT\_IN\_PATTERNS

No value

#### OCRMODE

Do the OCR in a faster mode.

Values are `FASTMODE` or `BALANCEMODE`.

#### 19.2.2.4 OCR for specific zone type

```
<!-- ### OCR for specific zone type ### -->
<SET name="OCRTABLE" enable="0">AutoLayoutDetection</SET>
<SET name="OCRADVERTISEMENT" enable="0" />
<SET name="CHEMICALFORMULA" enable="0" />

<SET name="NOITALIC" enable="0" />
<SET name="NOSUBSCRIPT" enable="0" />
<SET name="NOSUPERSCRIPT" enable="0" />
<SET name="ONE_LINE_ZONE" enable="0" />
<SET name="ONE_WORD_LINE" enable="0" />
```

##### OCRTABLE

The way that OCR for table is detected.

[ScriptCellsDetection](#)

Automated way by script that detects the cells for Table.

[AutoLayoutDetection](#)

Automated way by using ABBYY that detects the OCR for Table.

[LikeTextblock](#)

If empty the OCR will be made regular.

##### OCRADVERTISEMENT

The way that OCR for Advertisements is detected.

[AutoLayoutDetection](#)

Automated way by script that detects the cells for Advertisements.

##### CHEMICALFORMULA

Add chemistry language for chemical formula.

##### NOITALIC

Don't detect Italic font.

##### NOSUBSCRIPT

Don't detect sub script font.

##### NOSUPERSCRIPT

Don't detect supra script font.

##### ONE\_LINE\_ZONE

The zone has just one line.

##### ONE\_WORD\_LINE

The word has just one line.

### 19.2.2.5 Image processing settings - Job settings

```
!-- ### Image processing settings - Job settings ### -->

<SET name="SKEWCORRECTION" enable="0">BlackSquaresHorizontally
BlackSquaresVertically HorizontalLines VerticalLines HorizontalText
VerticalText</SET>

<SET name="REMOVECAMERABLUR" enable="0"/><!-- Availale only in FR10/FR11-->
<SET name="REMOVECAMERANOISE" enable="0"/><!-- Available only in FR10/FR11-->
<SET name="REMOVEGARBAGE" enable="0"/><!-- Available only in FR10/FR11-->
<SET name="REMOVEGEOMETRICDISTORTIONS" enable="0"/><!-- Available only in
FR10/FR11-->
<SET name="DISABLEORIENTATIONCORRECTION" enable="0"/><!-- Available only FR11
for Table and Advertisements-->
```

#### SKEWCORRECTION

Sets the mode skew correction is done.

#### REMOVECAMERABLUR

Removes camera blur effect on image.

#### REMOVECAMERANOISE

Removes camera noise effect on image.

#### REMOVEGARBAGE

Removes small image parts that are detected as trash on image.

#### REMOVEGEOMETRICDISTORTIONS

Removes camera distortions on image.

### 19.2.2.6 Enhanced image settings

By following parameters the usage of enhanced images can be enabled in `***DATA***\project-cfg\Default\Default-OCRCfg.xml`.

These images are specifically optimized for OCR recognition. These images are stored in addition in the pool and have to be considered in capacity load. These images are independent to output image configuration.

Special script for computing the OCR. If empty, no script will be used.

```
<!-- ### special script for computing the OCR. If empty no script will be
used ### -->

<SET name="OCRScripts" enable="0">

  <SET name="PATH"
enable="0">***DATA***\script\OCRImageEnhAnFrAnMerge.tcl</SET>

  <VALUE name="ColorImageEnhForOCRScriptParam"
enable="0">StretchContrastAndUnsharp</VALUE>

  <VALUE name="GrayImageEnhForOCRScriptParam"
enable="0">StretchContrastAndUnsharp</VALUE>
```

```
<VALUE name="BWImageEnhForOCRScriptParam" enable="0" />
</SET>
```

## OCRScripts

Possible values:

```
***DATA***\script\OCRDetectLanguage.tcl
```

Try to autodetect proper language from an extended list of languages. Is not very efficient, specially for engines with payment per page usage, since language is detected based on text statistics after running OCR multiple times for each zone, with different language settings. Might be useful when engine autodetection do not work properly or if document contains texts in more than three languages.

```
***DATA***\script\OCREmpty.tcl
```

Used when is needed to skip OCR detection on a particular step (f.e. for Fraktur languages, in order to not use too many pages, OCR is skipped for most of the jobs and only performed on export).

```
***DATA***\script\OCRImageEnh.tcl
```

Today is not used too much in practice. It performs OCR on default image and if quality is not good enough for some individual zones (based on several confidence criteria), perform OCR on enhanced images for those zones. Algorithms for enhanced images are defined in `dwShare\config\ConfigTool\ocr_values.xml`, and procedures are defined in [OCRHelpers.tcl](#).

```
***DATA***\script\OCRImageEnhForce.tcl
```

Today is not used too much in practice. Is similar with `OCRImageEnh.tcl`, but will not try first with original image, will run once on enhanced images.

```
***DATA***\script\OCRImageEnhAnFrAnMerge.tcl
```

Detects between Enhance Grayscale/BW Images. Detects Fonts Antiqua/Fraktur. Merge between Antiqua/Fraktur.

```
***DATA***\script\OCR-PDF.tcl
```

Should be used when page source is a PDF file. Tries first to get text from PDF and in case there is not text available, performs regular OCR for that particular zone.

```
***DATA***\script\OCR-PDFVote.tcl
```

Run OCR and extract text from PDF; set both texts using variant mechanism in `dw`, so that the user choose the best option from both options (OCR vs PDF).

```
***DATA***\script\OCR-PDFComp.tcl
```

Similar with `OCR-PDFVote.tcl`, but additionally compare the results and toggle the lines, so that best option is used as default one (based on different typical issues of PDF text found during the time). For PDF-Vote, always version from PDF is considered the good one

```
***DATA***\script\OCRScript.tcl
```

One of the first OCR scripts, trying different options of OCR (different font types, different source images - GS/BW, etc). Was not maintained in time and also not used in present; is used only as sample how to implement such a script.

```
***DATA***\script\OCRScriptFRorAN.tcl
```

Detects between Grayscale/BW Images. Detects Fonts Antiqua/Fraktur. Merge between Antiqua/Fraktur.

An old script used when Abbyy FR was not able to autodetect the font type. Not usable today since was calibrated on OCR confidence values from Abbyy FR 7/8 which are not used anymore.

```
***DATA***\script\OCRScriptFRorAN_Tesseract.tcl
```

Tesseract do not support, by default, type option "Antiqua + Fraktur". This script performs OCR with both settings and combine the results based on confidences (which are calibrated to work with Tesseract).

```
***DATA***\script\OCR-TessComp.tcl
```

Old script, not possible to run with last OCR implementation (using old TesseractWrapper).

```
***DATA***\script\OCRTesseract.tcl
```

Old script, not possible to run with last OCR implementation (using old TesseractWrapper).

ColorImageEnhForOCRScriptParam

Sets the script parameter used for color images.

StretchContrastAndUnsharp

Convert Image to Gray and Auto Stretch Contrast and Unsharp the image.

GrayImageEnhForOCRScriptParam

Sets the script parameter used for gray images.

StretchContrastAndUnsharp

Auto Stretch Contrast and Unsharp the image.

BWImageEnhForOCRScriptParam

Sets the script parameter used for bw images.

MakeDarker

Make the image Darker. Remove white noises.

#### 19.2.2.7 Post OCR processing

```
<!-- ### post OCR processing ### -->
```

```
<SET name="REPLACE" enable="0">
```

```
  <VALUE name="" enable="0">***DATA***\FrMorpho\LongS.tri</VALUE>
```

```
  <VALUE name="" enable="0">tri_replacement_list2</VALUE>
```

```
  <VALUE name="" enable="0">tri_replacement_list3</VALUE>
```

```
</SET>
```

```
<SET name="NO_SPELL_CHECK" enable="0" />
```

```
<SET name="PostProcessingScript" enable="0" />
```

REPLACE

Post corrections for OCR. List used to replace characters.

NO\_SPELL\_CHECK

Don't do the spell check.

PostProcessingScript

Post corrections for OCR.

#### 19.2.2.8 OCR settings

```
<!-- ### special DW settings ### -->
<SET name="NoOCR" enable="0" />
<SET name="NoPdfOcr" enable="0" />
```

NoOCR

Blocks the OCR to be done. in case that is need, an error is throwing.

NoPdfOcr

Don't use the ocr from PDF.

#### 19.2.2.9 Zone Type

For each zone type you can define the font type and language individually.

Within the <Settings> node you configure a class node for each zone type, e.g.:

```
<CLASS name="Textblock" selectiontype="Textblock">
    <SET name="FONTTYPE" enable="0">Antiqua</SET>
    <SET name="LANGUAGE" enable="0">German</SET>
</CLASS>
<CLASS name="Headline" selectiontype="Headline">
    <SET name="FONTTYPE" enable="0">Antiqua</SET>
</CLASS>
<CLASS name="Stockmarket" selectiontype="Table">
    <SET name="OCRTABLE" enable="0">AutoLayoutDetection</SET>
</CLASS>
<CLASS name="Advertisements" selectiontype="Advertisement">
    <SET name="OCRADVERTISEMENT" enable="0">AutoLayoutDetection</SET>
</CLASS>
<CLASS name="InterTitle" selectioncond="this.type == 'Textblock' & &
this.height < 140">
    <SET name="FONTTYPE" enable="0">Fraktur</SET>
</CLASS>
```

### 19.2.3 Project based / customized dictionary

Precondition: Dictionary database and settings in **docWizz-global.ini** section [\[DICTIONARY\]](#) .

Note: The dictionary database has to be configured by CCS.

[\[DICTIONARY\]](#)

SERVERNAME=

DATABASETYPE=SQLSERVER

DATABASENAME=dW\_Dict

USERNAME=

PASSWORD=

CONNECTIONSTRING=

DBTYPE=2

In the user interface are default actions to add or remove words from dictionary in text selection actions in Text view on left pane.



All projects will write into the same dictionary. Same entries can be used by multiple projects.

Select words manually or select all words.

Import existing dictionaries.

Extra task possible to configure dictionaries from zones (e.g. whatever headlines are exported will go to dictionary).



## 19.3 OCR confidence computing in docWizz

OCR confidence is computed on basis of confidence values reported by the embedded OCR engine. Each OCR engine provider uses its own methodology to compute those values and may use a different range of values. Typically, an engine reports a confidence value per character.

**Confidence level of each character (CC):** A list of numbers, one number between 0 (sure) and 9 (unsure) for each character.

It is determined by normalization of the raw confidence values, reported by the embedded OCR engine, to the ALTO standardized range of values, 0-9.

Example: CC="002500320080"

**Word confidence (WC):** A value between 1.00 (confident) and 0.00 (unsure).

Word confidence is computed based on raw character confidence values reported by the embedded OCR engine. The probability of a word to be correct increases significantly as soon as the word can be found in a dictionary. The longer the word is, the less is the probability that a word is wrong.

For that reason, in case the word is found in dictionary, we are using the formula:

$$c_{wd} = 1 - (1 - c_w) / (l_w * 3)$$

where  $c_w$  is the confidence calculated as the average of the raw confidences reported for each character and  $l_w$  is the length of the word.

In all other cases when the word is not found in the dictionary or its length is smaller then 3 characters  $Cwd=Cw$ .

In addition, CCS applies simple mathematics to use additional information such as:

- Manually corrected characters or completely typed words are treated as 100% confidence.
- Word confidence is computed when user correction takes place.

**Page confidence (PC):** A value between 1.000 (confident) and 0.000 (unsure).

Page confidence is computed based on word confidences (WC).

The weighted average of all WC's values for the words detected on the that page, weighted with the length of each word is the formula used to determine the PC value.

Accuracy is computed if manual sampling has been done: Per type of document, several text blocks are checked manually. Numbers of found errors are set into relation to computed confidence. This leads to a dependency of confidence and accuracy. This way an Accuracy value that is close to reality can be placed. As accuracy measurement requires manual sampling, this feature is optional and will be created when required only.

## 19.4 OCR accuracy statistics

Accuracy is computed if manual sampling has been done: Per type of document, several text blocks are checked manually. Numbers of found errors are set into relation to computed confidence. This leads to a dependency of confidence and accuracy. This way an Accuracy value that is close to reality can be placed. As accuracy measurement requires manual sampling, this feature is optional and will be created when required only.

Note: In docWizz confidence and accuracy calculations are carried out on character level.

### 19.4.1 Choose Random Samples

In order to get statistics about OCR accuracy a number of **random inspections** has to be carried out on text block basis. A quantity of 30 samples should be sufficient to get significant results. Text block samples are to be chosen from different pages of the document and from different areas on the page (edges/center) in order to represent varying image quality within a document or even within one page.

### 19.4.2 Attain Number of Errors

OCR Quality Checks are carried out in the step VerifyHierarchy

Switch to **ImageView** on the **right working window**. Then select the text block you want to check by marking it with left mouse click. Switch to Text View and hit the **Print Button** on the right tool bar just above the working window.

The Print Button might be hidden if the right working window and its tool bar are relatively narrow. To make the print button visible extend the right working window by moving the left edge of the working window further left.



In the printed document the OCR text can be found in the upper section and the related part of the image in the lower section. Compare the text and count the errors.

Switch to the directory **\\docWizz\\WORK** on your computer and open the document **<project\_name>-OCRStatistics.xml** using the right mouse button; then hit "Open with..." and choose the text editor or any XML editor.

The data for the text block recently printed will be added to the bottom of this file. It will look like this:

```
<SAMPLE ID="5">
  <CHARACTERS>437</CHARACTERS>
  <CONFIDENCE>989.1</CONFIDENCE>
  <SOURCE>Document ID 28, page 104</SOURCE>
  <ERRORS>not counted yet</ERRORS>
</SAMPLE>
```

**Key:**

SAMPLE ID = Unique number of printed sample.

CHARACTERS = Number of characters within the sample.

CONFIDENCE= Character confidence given by OCR engine, i.e. Characters identified per 1000 characters.

SOURCE = Additional information about the sample.

ERRORS = Number of true errors by the OCR engine (a character might have been treated as correct in confidence calculation but is actually wrong). This number has to be edited by the operator.

To edit the line `<ERRORS>Not counted yet</ERRORS>` simply type in the number of errors (e.g. 9): `<ERRORS>9</ERRORS>`. Save and close file.

Repeat the steps until you have checked a sufficient number of samples.

### 19.4.3 Evaluate Statistics

After exporting the document the OCR accuracy value for each page can be found in the corresponding XML file in the ALTO folder within the OUT folder

\\docWizz\OUT\<PROJECT\_NAME>\<DOCUMENT\_NAME>.DIR\  
<DOCUMENT\_NAME>\_ALTO\<PAGE\_ID>.xml .

Open the file using the text editor (right click the file and choose "Open with..." and Editor) or any XML editor.

In the **Layout Section** you can find information about accuracy:

```
<Layout>  
<Page ID="P4" PHYSICAL_IMG_NR="4" HEIGHT="2970" WIDTH="2099" ACCURACY="98.94">  
...
```

Beside other information about the page, accuracy information can be found at the end of the line.

**Key**

ACCURACY = OCR accuracy for selected page as number of correctly identified characters per 100 characters. The calculation is based on character confidence of OCR engine combined with an extrapolation of the errors counted manually in some text block samples.

The OCR Engine name is output in ALTO file (might be more than one engine if OCR runs on different machines, with different versions installed).

## 19.5 OCR character counting

OCR character counting for Documents with Gothic and Antiqua always both OCR types run and the better text is taken (merged).

There is no function that can check per zone which OCR type to do.

Gothic count is always done and subtracted from the total count.

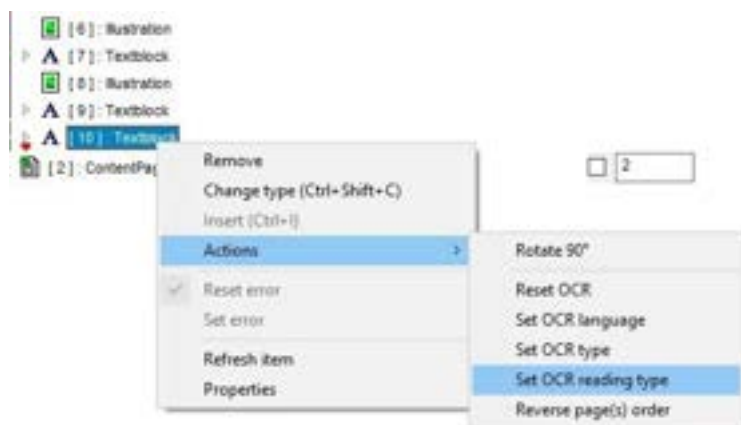
## 19.6 OCR Languages with different reading rules

Some languages have different reading rules, meaning that text is read line by line, or column by column (vertical text).



There are languages where one single option is available (f.e. european languages - horizontal, left-to-write, Hebrew - horizontal right-to-left) but as well languages where both vertical/horizontal reading is used f.e. Japanese/Chinese.

Some languages have different reading rules, meaning that text is read line by line, or column by column (vertical text).





Abbyy FR engine tries to detect automatically which is the case. Now is possible to change in interface, for a zone, or more zones the way the text is read: auto detection, horizontal stripes (by line), vertical stripes (by column).

It is possible to force this via project configuration (OCRCfg.xml) - this will apply for the entire project. Additionally is possible to change this via interface, for each particular zone (see related topic).



## 19.7 Abbyy Finreader 11/12

Finreader 12 implements new features that could be used in the future. Image processing techniques, field level OCR correction, external dictionaries for better OCR detection.

The benefit is a better OCR detection and in the case of external dictionaries it allows us to enhance the OCR detection without creating a new patch for the current binaries.

### Settings

Image processing settings (Values: boolean, except skew): SKEWCORRECTION (Values: BlackSquaresHorizontally BlackSquaresVertically HorizontalLines VerticalLines HorizontalText VerticalText), REMOVECAMERABLUR, REMOVECAMERANOISE, REMOVEGARBAGE, REMOVEGEOMETRICDISTORTIONS.

Field level OCR detection (Values: boolean): ONE\_WORD\_LINE, ONE\_LINE\_ZONE, NOITALIC, NOSUBSCRIPT, NOSUPERSCRIPT

External dictionary: USE\_EXTERNAL\_DICTIONARY (Value: Path to the dll module) and USER\_LANGUAGE (value: The user language used for the dictionary).

OCR settings file added for Import rdy file:

```
<OCR_CONFIG_PATH>***PROJECT_CFG***\Default\Default-  
OCRCfg.xml</OCR_CONFIG_PATH>
```

For the OCR correction settings in Outline View TextView the "TAB to numbers" functionality has been added, this is used for correcting numbers by operators. The view is adapted to grey out everything that is not a number, and user can tab only to numbers or words that contain numbers.

## 19.8 Abbyy Finereader license backup and recovery

There is a registration file of the Abbyy Finereader OCR service on each computer you are using. Once installed, an OCR license can only be used on this dedicated computer and cannot be switched to another one.

If you have saved a copy of the registration file and know to which computer it belongs a recovery of the license key is possible in case one of the computers crashes down.

Therefore, the following chapters will explain how to create a backup and how to use this backup for recovering the Abbyy Finereader OCR License. But even if you have not created a backup of the OCR licenses there is a way to recover the license for a single computer.

The messages "Error retrieving license data for engine xxx" and "Error Message on retrieving license data for engine xxx yyyy" are telling, that the ABBYY engine is not available at that time. If this occurs on external access (like removing license), this would be ok, in other cases check environment.

### 19.8.1 Preparing License Backup

In order to be able to run future recovery please proceed as follows:

- On each computer: Open the Finereader directory (e.g. \\ccs gmbh\OCRService\Finereaderxx).
- The file needed for the backup ends on ".frelf" (e.g. FECA-xxxx-0001-xxxx-xxxx.frelf).
- Copy this file into a backup folder on an external disk or external directory.
- Additionally, make a note (e.g. using MS Excel) to which computer each file belongs.

### 19.8.2 Recovering from your License Backup

In case you need to recover the OCR service (e.g. if a computer crashes down) please proceed as follows:

- A backup of the OCR license is needed in order to run the recovery process (see description above).
- The OCR Service folder needs to be re-established in the directory \\ccs gmbh. This can be done by copying it from any other computer (from the directory \\ccs gmbh\OCRService).
- On the computer that needs to be recovered, open the Finereader directory (\\ccs gmbh\OCRService\Finereaderxx).
- Replace the existing ".frelf" file (that belongs to the "source" computer) with the one from your OCR license backup. Make sure you use the one dedicated to this computer.
- In order to check if registration succeeded run EngineLicenseManager.exe in the Finereader directory. A pop-up window should confirm the registration.

### 19.8.3 Recovering without earlier created License Backups

If you have not backed up your OCR licenses earlier but would like to recover a OCR license on a single computer please proceed as follows:

- Firstly, re-establish the OCR Service folder within the directory \\ccs gmbh. This can be done by copying it from any other computer (from the directory \\ccs gmbh\OCRService).
- On the computer that needs to be recovered, open the Finereader directory (\\ccs gmbh\OCRService\Finereaderxx) and delete any existing file ending on ".frelf".
- Then you should create a backup of the OCR licenses still existing and used in your system environment. This is explained earlier (see above).
- From the list that matches each license with the computer on which it is used you should be able to identify the license that is left for the computer on which you would like to recover the OCR license.
- What you need now is the serial number of the license you want to recover. It can be deducted from the name of the ".frelf" file and looks like this: FECA-xxxx-0001-xxxx-xxxx. Make sure you take the serial number of the license belonging to the computer on which you want to recover.
- Open (by editing) the file "FREngine.ini" in the Finereader directory (e.g. \\ccs gmbh\OCRService\Finereaderxx).
- Check the third line "SerialNumber=". Probably, you will find there the Finereader Serial number for the computer from which you copied the directory.

- Edit this line and make sure it contains the Finereader serial number of the according computer, e.g. "SerialNumber=FECA-xxxx-0001-xxxx-xxxx".
- If you were not able to identify the serial number you must use a new license
- Save and close the FREngine.ini file
- Run the EngineLicenseManager.exe and choose activation via Internet.
- You are now able to return to work as usual.



## 19.9 Abbyy Finereader and dual core licenses

Compared to single core licenses where only one instance is configured using ABBYY FR Dual Core licenses there is a special configuration in the docWizz configuration:

- 2 instances are configured to OCR jobs, the other two instances to Non-OCR jobs.
- The Non-OCR jobs are nevertheless able to process OCR in exceptional cases.

On each server with OCR license a RemoteOCR service could be installed, too.

## 19.10 Abbyy license server

If docWizz is not starting and cannot connect to the dwocrrun.exe even if a license is installed and active, this happens if a Abbyy License Server is configured in the FREngine.ini file in the Finereader directory like this:

```
[Protection]
ServerName=127.0.0.1
ProtectionType=Server
```

But if there is no Abbyy license Server, it must look like this:

```
[Protection]
ProtectionType=File
```

## 19.11 Abbyy Fraktur (Gothic) OCR licenses

Gothic OCR licenses are character limited. Whenever the remaining number of characters gets very low this will be outlined in Control Center to make you aware about the status of each Abbyy FR license equipped with Fraktur (Gothic).

The warning status has to be configured so that there is enough time to order new Gothic OCR license until the error status is achieved.

There are two warnings implemented:

- one for each service blocking to the machine where this special type of OCR is running
- and a second one on the Remote OCR service belonging to the same machine.

### Configuration in ...-docWizz-dwsrv.ini

```
[GothicOCR]
;WarningLimit=20 means Warn on less than 20000 character for gothic license
WarningLimit=20
;ErrorLimit=5 means Error on less than 5000 character for gothic license
ErrorLimit=5
;MachineList=xxxx means machine xxxxx must have a valid gothic license, if
not show error
MachineList=
```

## 19.12 Tesseract OCR

Tesseract engine can be configured to use an alternative folder for all trained models, not the default one located into the binaries installed on each machine. In order to do this you need to set in custom-docWizz-glbl.ini following entry:

```
[DWOOCR]
REMOTEOCR=0
NOTTRYLOCAL=0
REMOTEWORKEFRESH=-1
EXCLUDEFROMREMOTEOCR=
DEBUGOCR=0
OCRCHARWARNLIMITANTIQUA=1000
OCRCHARWARNLIMITGOTHIC=1000
TESSDATADIR="path to Tesseract trained data"
```

Into Tesseract trained data (TessData into Tesseract binaries or the custom folder) there is a settings file (languages.xml) that defines a correspondence between the language/font type and the specific model (tessdata file). In case you want to add a new custom language, you need to add a new entry into this file and as well add tessdata file into models folder (either the default - but this will be overwritten on update with default language.xml - or the custom one).

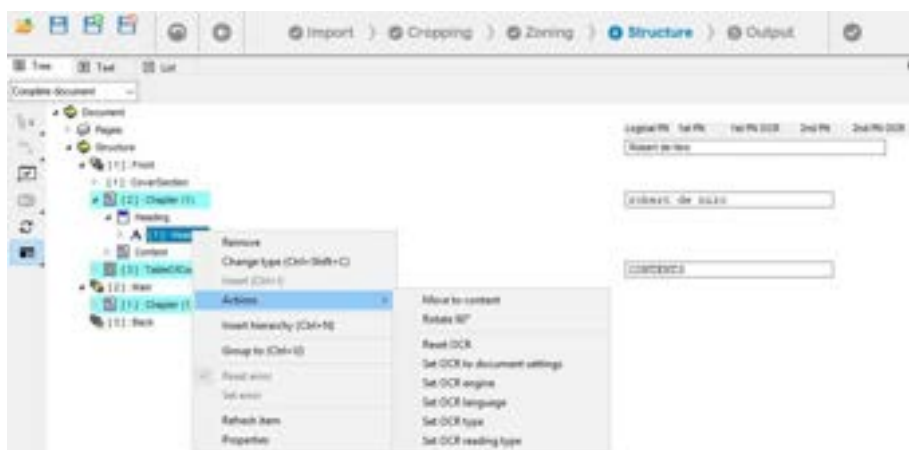
### How language.xml is used and how the correspondence is made with the correct model:

All trained data files from the TessData directory that are not mentioned in this XML file are initialized based on their name and added only if their name is a valid ISO language code that docWizz recognizes (e.g. nld, eng, ita, jpn, ... etc). These are added by default with the settings: not Gothic, not old, horizontal. If you have a trained data file that was trained with Gothic and its name is a valid ISO language, docWizz will load it as it would be with the default settings (see previous paragraph). To override the settings for that trained data, just declare it in this xml (e.g. <LANG lg="10" data="nld" fraktur="1" old="0" vert="0"/>). The same applies for any setting (Gothic font, old or vert).

Example:

The main document is in English and only some zones are in Chinese. Then you may use Tesseract to read OCR of the Chinese zones.

Or if even a few zones are in Fraktur and you do not have the Finereader Fraktur license available you may use Tesseract for this zones.



## 19.13OmniPage 20

For Omnipage 20 engine installation, the "Nuance20.2.zip" file is required. The licensing will be provided by CCS.

In order to use any other/additional license than the one that CCS provides, the following should be done:

Create a Nuance.ini file in the docWizz installation path, in the Nuance20 directory from the OCR binaries folder (e.g. "C:\Program Files\docWizz 7.1\bin\dwOCR\Nuance20").

The Nuance.ini should look like this:

```
[LICENSE]
PATH=full_path_to_license_file
SERIAL=OEM_license_code
WITHLIC=1 (values 0, 1, 2 or 3)
```

### Usage

**PATH** should have the full path to the distributed OEM license file (is a file with a lcxz, lcxp, or similar extension).

**SERIAL** is the OEM distribution code (is based on the hardware info of the computer that generated the license file set in PATH).

**WITHLIC** can be 0, 1,2 or 3 and means the following:

**0** - the default CCS license will be used, with no additional modules (PATH and SERIAL will be ignored);

**1** - the license set in PATH and SERIAL will be used (IT ONLY WORKS WITH OEM LICENSES)  
Trying to use a Volume based license or any other type, will cause the Nuance20 API to not initialize, which means that the DWOCRN20 module will not be loaded;

**2** - the default CCS license will be used, along with any other installed Nuance20 module licenses that are detected by the the Nuance Licensing Service on the machine (tested with a Volume based license for Arabic which worked ok);

**3** - no license, OmniPage will not be initialized/used.

### Important

Using any other value for WITHLIC will not initialize the engine due to bad configuration.

If the Nuance.ini file is missing, the application worked with WITHLIC 0 till version 7.1.0.53 and starting with version 7.1.0.54 is used as default WITHLIC 3 (no license by default).

Note: Omnipage20 is removed from docWizz starting with version 7.2.0.22

## 19.14OmniPage 21

For OmniPage 21 engine installation, the `Nuance21.zip` file is required. The licensing will be provided by CCS.

In order to use any other/additional license than the one that CCS provides, the following should be done:

Create a `Nuance.ini` file in the docWizz installation path, in the Omnipage21 directory from the OCR binaries folder (e.g. "C:\Program Files\docWizz 7.2\dwOCR\Omnipage21").

The `Nuance.ini` should look like this:

```
[LICENSE]
PATH=full_path_to_license_file
SERIAL=OEM_license_code
WITHLIC=1 (values 0, 1, 2 or 3)
```

### Usage:

`PATH` should have the full path to the distributed OEM license file (is a file with a `lcxz`, `lcxp`, or similar extension).

`SERIAL` is the OEM distribution code (is based on the hardware info of the computer that generated the license file set in `PATH`).

`WITHLIC` can be 0, 1, 2 or 3 and means the following:

0 - the default CCS license will be used, with no additional modules (`PATH` and `SERIAL` will be ignored);

1 - the license set in `PATH` and `SERIAL` will be used (IT ONLY WORKS WITH OEM LICENSES)  
Trying to use a Volume based license or any other type, will cause the Nuance20 API to not initialize, which means that the DWOCRN20 module will not be loaded;

2 - the default CCS license will be used, along with any other installed Nuance20 module licenses that are detected by the the Nuance Licensing Service on the machine (tested with a Volume based license for Arabic which worked ok);

3 - no license, OmniPage will not be initialized/used.

## 19.15 Google Vision

There are two possibilities to use Google Vision:

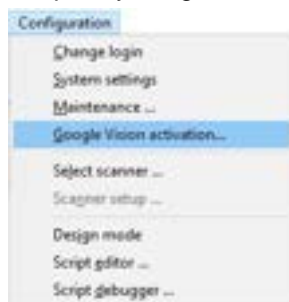
- Use CCS account license
- Create your own account for Google services and use that one

### Use CCS account license

The client needs to contact CCS Support to activate the license.

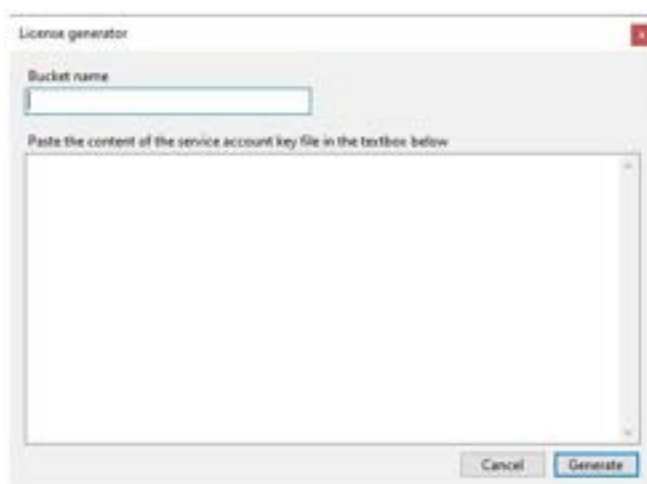
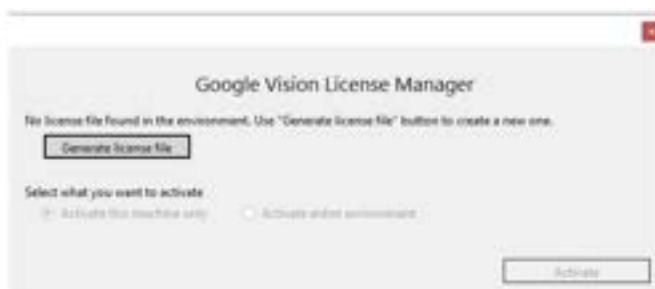
### Create your own account for Google services

After creating an account, the client also needs to create a bucket in google cloud (where the temporary images will be stored).



With the bucket name and the service account key file, the client can generate a license file using the Google Vision License Manager tool from docWizz.

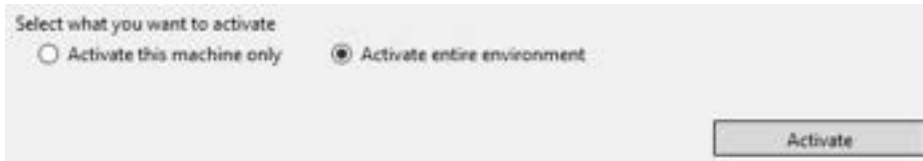
The tool generates a **gvkey.lic** file that should be placed in **dWshare\MAINTENANCE\GVLICENSE** folder.



Note: GVLM checks for the gvkey.lic in dWshare\MAINTENANCE\GVLicense folder - if the file is not found there, it cannot activate the license.

With the GV License Manager, the client can choose to activate the entire environment, in which case the **gvkey.lic** file will be copied to **dWShare\Install\dwOCR\GoogleVision** folder.

The docWizz update will distribute the license to all servers and clients from the environment.



Select what you want to activate

☐ Activate this machine only    ☒ Activate entire environment

Activate

Or he can only activate the current machine - in which case the gvkey.lic will be copied only to the docWizz installation folder from the current machine.



Select what you want to activate

☒ Activate this machine only    ☐ Activate entire environment

Activate

## 19.16Transkribus

To use Transkribus in your project, you need to make the following changes in **\*\*\*project-cfg\*\*\*\projectName\OCRConfig.xml**:

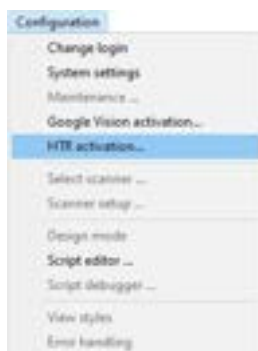
```
<OCRCfg version="1.2" uid="1" CRC="4786C68AC90303B57D5576A2C7FFF8E4">
<SETTING>
<CLASS name="Import">
<SET name="FONTTYPE" enable="0"/>
<SET name="LANGUAGE" enable="0"/>
<SET name="ENGINE" enable="1">Transkribus</SET>
<SET name="OCR_MANDATORY_ENGINE" enable="1">
</CLASS>
<CLASS name="Default">
<SET name="HTRID" enable="1">29820</SET>
<SET name="FONTTYPE" enable="0"/>
<SET name="LANGUAGE" enable="0"/>
...
```

You need to set as engine „Transkribus“, and in case you want to have that engine mandatory, set mandatory flag (OCR\_MANDATORY\_ENGINE) to 1 (for handwritten text is recommended to be like this, since for the moment is the only supported engine with this capability).

As main difference from other engine is the new parameter HTRID. For all the other engines selecting the language is enough to select the proper detection model for the engine. For Transkribus, since it deals with a lot of different writing styles, both handwritten or printed, language is less important, more important is the writing style, or maybe custom model for a particular collection. This is why you need to set also the ID of the model used.

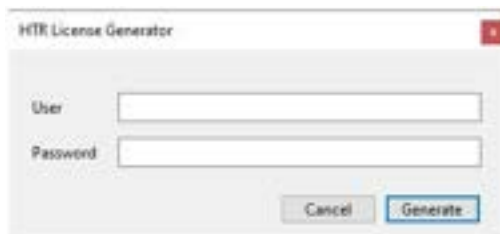
For available models and HTRID values, please have a look on <https://readcoop.eu/transkribus/public-models/>

### Transkribus activation



### Generate license file

The client can generate a license file using an user and password provided on setup.



The docWizz update will distribute the license to all servers and clients from the environment.



Or the user can only activate the license on the current machine (for both client application or processing services if available on that machine).





## 19.17 Project based auto replacements

Auto replacement of characters can now be defined per project.

Some replacements were hard coded in previous releases and could cause unexpected issues in specific situations. This has been removed now completely and is replaced by fully individual configurable solution within the project settings.

This auto replacement is additionally to the dictionaries settings (see docWizz interface: Configuration\System\Dictionaries).

Auto replacement (\*.tri files) are available as common configurable solution now (known before as "longS" solution - 'longS.tri' file used for converting f in s for longS in historical printings.)

```
<SET name="REPLACE" enable="0">
  <VALUE name="" enable="0">***DATA***\FrMorpho\LongS.tri</VALUE>
  <VALUE name="" enable="0">tri_replacement_file2</VALUE>
  <VALUE name="" enable="0">tri_replacement_file3</VALUE>
</SET>
```

## 19.18 Compare two texts and report differences (using Levenshtein distance)

For example two texts tell the number of minimum transformations needed to change the text from t1 to t2 and what are these transformations;

Allowed transformations are:

1. delete a character
2. replace a character
3. insert a character

### Example:

```
"This is text number 1"
```

```
"Ths isn't text number 2"
```

should create beside difference (as number of differences) also where these appear and what is the difference type:

```
"Th<d>i</d>s is<i>n't</i> text number <r>2</r>"
```

where:

<d> is markup for "delete"

<i> is markup for "insert"

<r> is markup for "remove"

The levenshtein distance algorithm produces the backtrace for the given strings.

The parameter ("-backtrace") uses the algorithm and the result will be the distance + a string formed with this characters:

- 'N' - Nothing changed
- 'I' - Inserted the char
- 'D' - Deleted the char
- 'S' - Substituted the char

this returned string is correlated to the second string sent as parameter.

The command is known as "levenshteinDistance \$text1 \$text2 -param", where:

- leaving the param empty will call the old version of the Levenshtein algorithm;
- using `-backtrace` will result in an additional returned value that consists of letter("N", "I", "D", "S")  
f.e 5 NNDNNNNIIINNNNNNNNNNNNS
- using `-backtraceMarkup` will result in an additional returned value that is the string with the corresponding tag for each operation done to it f.e. 5 Th<d>i</d>s is<i>n't</i> text number <s>2</s>

For a better explanation of the algorithm you can see this PDF (<https://web.stanford.edu/class/cs124/lec/med.pdf>) that explains how the distance is computed and how the backtrace is calculated using the matrix associated to the distance.

## 19.19 Hunspell dictionaries

Hunspell engine is used to support the integration of Hunspell dictionaries. Hunspell replaces the ABBYY Morphology.

### Activating the engine (docWizz-GLBL.ini):

In `[SPELLCHECK]` section include: `USEHUNSPELL=1`

```
[SPELLCHECK]
LANGUAGE =0
LEXMODE =0
USEHUNSPELL=1
DICTPATH =***DICT***
SOURCEDICTPATH =***DICT***
USERDICTPATH =***DICT***\HSPELL
SOURCEUSERDICTPATH =***DICT***\HSPELL
```

Verify that the Hunspell dictionaries (HSpellx.dict) are located in the DICT\HSPELL variable folder.

### Hunspell dictionaries

The HSpell dictionaries are a list of txt files compressed to .zip (and renamed the extension to .dict). They can be opened as normal .zip files.

### Hunspell dictionaries import tool

This tool offers two ways to import Hunspell dictionaries:

- Select one of the Mozilla or OpenOffice dictionaries installed in the machine.
- Select the compressed file with the dictionary to import.
- After source dictionary selection the user must select the language of the chosen dictionary and click in the "Create" button.
- Creating a dictionary of an existing Hunspell dictionary will replace the previous dictionary with the new version!!

### General

A set of Hunspell dictionaries are included to the standard configuration. There are different sources of Hunspell dictionaries edited by the community or organizations. If a dictionary doesn't give good results could there are different alternatives; searching for other version, language specific dictionaries or editing a personal dictionary.

### Get Source Dictionaries

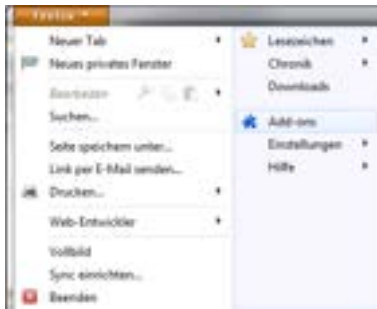
This tool offers two ways to import Hunspell dictionaries:

- Installed Mozilla Firefox Dictionaries
- OpenOffice Dictionaries

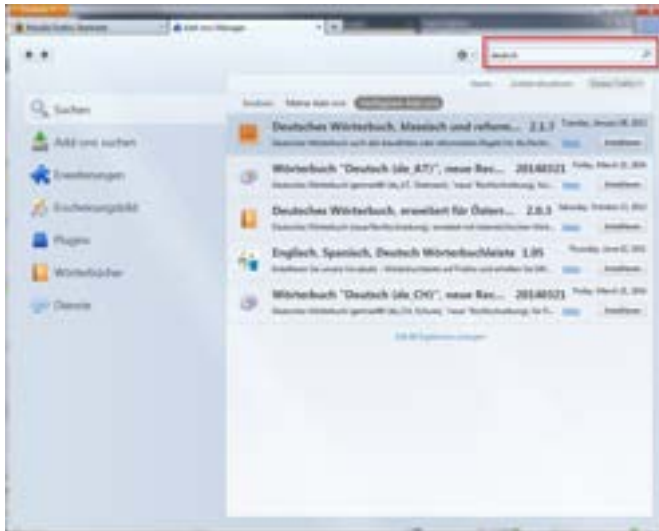
### Hunspell importer tool

Install a new dictionary in Mozilla Firefox.

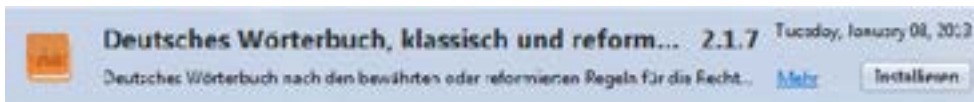
In Firefox go to Add-ons-Manager.



Go to “Add-on Search” and type the language that you want to install. You will get in the results a list of the available tools for the language including the dictionaries.



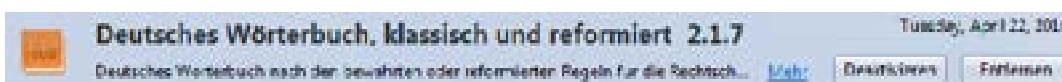
Select the dictionary and click “Install”:



A confirmation screen will appear, click in “Accept”.



The Add-on will be marked as installed.

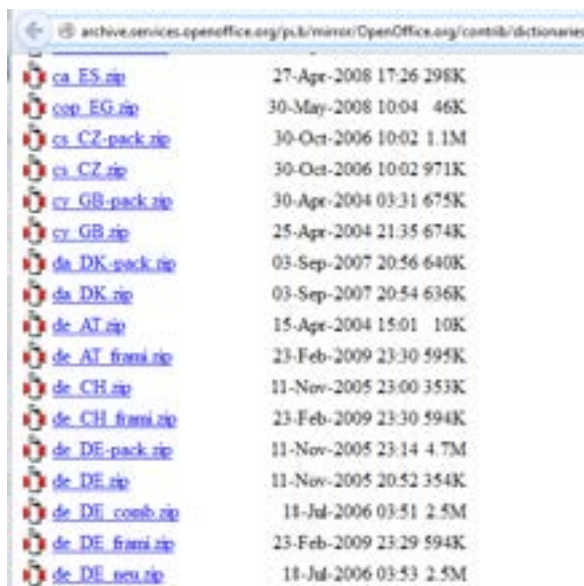


## OpenOffice dictionaries

The Open Office Dictionaries Wiki (<https://wiki.openoffice.org/wiki/Dictionaries>) links the latest URL where the Dictionaries are found.

(22/4/2014 updated link):

<http://archive.services.openoffice.org/pub/mirror/OpenOffice.org/contrib/dictionaries/>



<a href="#">ca_ES.zip</a>	27-Apr-2008 17:26 298K
<a href="#">cop_EG.zip</a>	30-May-2008 10:04 46K
<a href="#">cs_CZ-pack.zip</a>	30-Oct-2006 10:02 1.1M
<a href="#">cs_CZ.zip</a>	30-Oct-2006 10:02 971K
<a href="#">cy_GB-pack.zip</a>	30-Apr-2004 03:31 675K
<a href="#">cy_GB.zip</a>	25-Apr-2004 21:35 674K
<a href="#">da_DK-pack.zip</a>	03-Sep-2007 20:56 640K
<a href="#">da_DK.zip</a>	03-Sep-2007 20:54 636K
<a href="#">de_AT.zip</a>	15-Apr-2004 15:01 10K
<a href="#">de_AT_frani.zip</a>	23-Feb-2009 23:30 595K
<a href="#">de_CH.zip</a>	11-Nov-2005 23:00 353K
<a href="#">de_CH_frani.zip</a>	23-Feb-2009 23:30 594K
<a href="#">de_DE-pack.zip</a>	11-Nov-2005 23:14 4.7M
<a href="#">de_DE.zip</a>	11-Nov-2005 20:52 354K
<a href="#">de_DE_comb.zip</a>	18-Jul-2006 03:51 2.5M
<a href="#">de_DE_frani.zip</a>	23-Feb-2009 23:29 594K
<a href="#">de_DE_neu.zip</a>	18-Jul-2006 03:53 2.5M

## Import a new Hunspell Dictionary

Import an Installed Dictionary:



- 1.) Select Installed Dictionary
- 2.) Select Language
- 3.) Click "Create"

## Import a Dictionary File:



- 1.) Select Dictionary Compressed File (available extensions: .zip (open office) .odx (firefox) )
- 2.) Select available dictionary
- 3.) Select Language
- 4.) Click "Create"

Important: Creating a dictionary of an existing Hunspell dictionary will replace the new previous dictionary with the new version.

## Supported languages

			ABBY	Hunspell
Languages	ISO	Codepage	Included	Included
German (old spelling)	ISO8859-1	28591	X	X
English	ISO8859-1	28591	X	X
French	UTF-8	65001	X	X
Spanish	ISO8859-1	28591	X	X
Italian	ISO8859-15	28605	X	X
Swedish	UTF-8	65001	X	X
Finnish			X	X (voikko)
Danish	UTF-8	65001	X	X
Portuguese	ISO8859-1	28591	X	X
Dutch	UTF-8	65001	X	X
Russian	KOI8-R	20866	X	X
Bulgarian	microsoft-cp1251	1251	X	X
Catalan	ISO8859-1	28591	X	X
Croatian	ISO8859-2	28592	X	X

Cyech	ISO8859-2	28592	X	X
Dutch-Belgian			X	
Estonian	ISO8859-15	28605	X	X
German (New Spelling)	ISO8859-1	28591	X	X
Hungarian			X	X
Latvian	ISO8859-13	28603	X	X
Lithuanian	ISO8859-13	28603	X	X
Macedonian	microsoft-cp1251	1251	X	X
Norwegian (Bokmal)	UTF-8	28591	X	X
Norwegian (Nyorsk)	UTF-8	28591	X	X
Polish	ISO8859-2	28592	X	X
Portuguese (Brazilian)	ISO8859-1	28591	X	X
Romanian	ISO8859-2	28592	X	X
Serbian	UTF-8	48	X	X
Slovak	ISO8859-2	28592	X	X
Slovenian	ISO8859-2	28592	X	X
Tartar			X	
Turkish	UTF-8	65001	X	X
Ukrainian	UTF-8	65001	X	X
Greek	ISO8859-7	28597	X	X
Armenian (Eastern)	UTF-8	65001	X	X
Armenian (Western)	UTF-8	65001	X	X
Armenian (Grabar)			X	
Afrikaans	ISO8859-1	12		X
Albanian	ISO8859-1	13		X
Indonesian	ISO8859-1	30		X
Vietnamese	UTF-8	85		X
Thai	TIS620-2533	86		X
Maori	UTF-8	37		X
Tagalog	ISO8859-1	52		X
FAROESE	ISO8859-1	67		X

GALICIAN	ISO8859-1	70		X
MALAYSIAN	ISO8859-1	72		X
WELSH	ISO8859-14	74		X
ZULU	ISO8859-1	76		X
Esperanto	ISO8859-3	109		X
Icelandic	UTF-8	29		X
Mongol	UTF-8	148		X
Irish	UTF-8	31		X
Latin	UTF-8	33		X
Basque	ISO8859-1	14		X
Korean	UTF-8	84		X
Nyanja	ISO8859-14	154		X
Maltese	UTF-8	141		X
Malagasy	ISO8859-1	139		X
Interlingua	UTF-8	120		X
German Luxembourg	ISO8859-1	114		X
Telugu	UTF-8	82		X
Hebrew	UTF-8	81		X
Arabic	UTF-8	80		X
Somali	ISO8859-1	168		X
Quechua	UTF-8	159		X
Yiddish	UTF-8	189		X
Xhosa	UTF-8	187		X
Tswana	ISO8859-15	176		X
Swazi	ISO8859-1	171		X
Sotho	ISO-8859-1	169		X
Serbian Latin	UTF-8	165		X
Frisian	ISO8859-1	75		X
Hungarian	UTF-8	28		X
Swahili	ISO8859-1	51		X
Tongan	ISO8859-1	175		X



## 20 Export configuration

The export configuration is basically defined in two files:

```
projectname-export.xml  
and  
projectname-export.tcl
```

Please be aware of the renamed steps and tasks and the "old" names in scripts and configuration files.

### 20.1 TCL export configuration file

The file projectname-export.tcl will be loaded automatically when starting the export.

It shall contain all project specific procedures that are required during export. It should not contain any default execution except initialization of variables.

TCL handles global variables using the 'global' command. But as during export TCL procedures are called based on events, not always those global variables are available.

Instead you should use the docWizz 'globalvar' command to store variables that are used in different places.

### 20.2 Vendor Information

In ...config\...docWizz-glbl.ini a new key can be defined:

```
[INSTALLATION]  
VENDOR=Name_of_the_vendor
```

Then it will be added to METS as listed below:

```
<metsHdr          CREATEDATE="2012-02-30T00:45:55"          LASTMODDATE="2012-02-  
30T00:45:55">  
  <agent ROLE="CREATOR" TYPE="OTHER" OTHERTYPE="SOFTWARE">  
    <name>CCS docWizz/METAE Version 6.5-0</name>  
    <note>docWizz-ID: 24</note>  
  
  </agent>  
  <agent ROLE="CREATOR" TYPE="OTHER" OTHERTYPE="VENDOR">  
    <name> Name_of_the_vendor </name>  
  </agent>  
</metsHdr>
```

## 21 XML export configuration file

All XML export configuration files must validate against the docWizz\_Export.xsd schema. The schema is located in docWizz schema folder.

### 21.1 General configuration

```
<CONFIG>
```

```
<EXPORTS>
```

```
<EXPORT name="METAe Default" type="XML" num="METADEF.NUM" trace="1">
```

The configuration file must always have a root node <CONFIG>. It contains another node <EXPORTS>. Below that node one or more <EXPORT> nodes can be defined. Each <EXPORT> node has the following attributes:

<b>A name to identify the configuration.name</b>	The type of the export. By now, only "XML" is available. type
<b>num</b>	The name of a counter that might be used if each export shall get a unique number
<b>trace</b>	Set to "1" if unreferenced in logical elements shall be reported while exporting.
	XML files can be created with " and ' ; Use „<xmldoc> -save filename utf-8 uc quotes"  Documents opened in ExportXML are showing as well document hierarchy for split documents (e.g. newspaper issues): Analyze is loaded from ExportXML job.

### 21.2 Path and name configuration

```
<PATH>***EXPORT***\my_Project</PATH>
```

```
<Name>@ (TITLE) </NAME>
```

A <PATH> must be specified for the export. To handle path definitions more flexible, you should use docWizz internal path macros. Path macros are replaced by the definition during runtime. This mechanism enables you easily to change the environment or move storage locations without touching each configuration file. A macro is called by encapsulating the path name with three stars before and after.

All files that will be exported will be exported in or below the specified path.

The <NAME> tells the master name of the exported files. Files that are exported may use this master name and add special extension.

## 21.3 Image creation configuration

```
<PAGEIMAGES save="1" multipage="0" subdir="1" format="7" formatBW="0"
formatGS="0" resolution="0" name="*_####" bwextra="1" setorientation="1">
  <SCRIPT></SCRIPT>
  <PARTIALIMAGEPATH>PrintSpace</PARTIALIMAGEPATH>
  <PARTIALIMAGECLASSES>Illustration</PARTIALIMAGECLASSES>
</PAGEIMAGES>
```

<b>save</b>	Tells whether to save images or not.
<b>multipage</b>	Tells whether to create a multiple page image file or one separate file per page
<b>subdir</b>	Tells whether to locate image files in the same folder like the master output folder or in subfolders
<b>format</b>	Tells the image format to be used for each coloured image.
<b>formatBW</b>	Tells the image format to be used for each bitonal image
<b>formatGS</b>	Tells the image format to be used for each greyscale image
<b>Resolution</b>	Defines the resolution in dpi to be used when creating the images. A value of "0" indicates to use the same resolution like the source image file
<b>Name</b>	Defines the naming of the image files. A '*' represents the document export name. Each '#' represents one digit of numbering the images. If you need a more specific naming that can't be handled with that simple way of configuration, use a script for image naming
<b>BWextra</b>	Tells whether to export bitonal images in addition to the original greyscale or color image
<b>Setorientation</b>	If an image is flagged in docWizz to have a non-portrait reading orientation, the exported image might be either rotated (0) or just the orientation tag will be set (1).
<b>jpegQuality</b>	Quality to be used when creating JPEG images. The value can be in range from 1 (poor) to 100 (lossless)
<b>jpeg2000Quality</b>	Quality to be used when creating JPEG2000 images. The value can be in range from 1 (high compression) to 100 (no compression).

### 21.3.1 Partial images

docWizz provides support to create automatically partial images for special elements like tables, illustrations and more. They can be defined in page image section. The <PARTIALIMAGEPATH> defines where in docWizz internal physical structure information on partial images can be found. For normal kind of documents, it is always "PrintSpace". The element <PARTIALIMAGECLASSES> tells, for which type of zones a partial image file shall be created. It may be repeated if images shall be created for multiple element types.

```
<PARTIALIMAGEPATH>PrintSpace</PARTIALIMAGEPATH>
<PARTIALIMAGECLASSES>Illustration</PARTIALIMAGECLASSES>
```

### 21.3.2 Image creation script

```
<PAGEIMAGES ...>
  <SCRIPT></SCRIPT>
</PAGEIMAGES ...>
```

This script can be used instead of default image creation. It must return the list of names of files that have been created.

#### Available variables

<code>\$imagename</code>	the name the image should get
<code>\$page</code>	the current page object

#### Sample:

##### In **xml** file

```
<SCRIPT>ProcessMultipleImages $page $image</SCRIPT>
```

##### In **tcl** file

```
proc ProcessMultipleImages { page imagename } {
  set outfiles ""
  if { ![[$page -image lowres] -bitonal] } {
    if { 0 } { # set to 1 to export BW as well
      set bwimg [$page -image BW]
      set bwoutfile [string range $imagename 0 [expr [string last "."
$imagename] -1 ]]_BW.tif
      $bwimg -save $bwoutfile 0
      lappend outfiles $bwoutfile
    }
    Set img [$page -image]
    file delete $outfile
    set outfile [string range $imagename 0 [expr [string last "."
$imagename] -1 ]].tif
    if {[$img -save $outfile 3] != 1 } {
      error "create $outfile failed"
    }
    lappend outfiles $outfile
  } {
    set img [$page -image]
    set outfile [string range $imagename 0 [expr [string last "."
$imagename] -1 ]].tif
    $img -save $outfile 0
    lappend outfiles $outfile
  }
  return $outfiles
}
```

### 21.3.3 File format identifier

File format identifiers are used for page image parameters and for TCL <img> -save commands.

- 0 - Tiff Group 4
- 1 - Tiff Group 3
- 2 - Tiff Huffman
- 3 - Tiff uncompressed
- 4 - GIF
- 5 - BMP
- 6 - JPEG
- 7 - Tiff LZW
- 8 - JP2
- 9 - png
- 10 - Tiff deflate

### 21.3.4 JP2 settings in export.xml and export.tcl

#### **export.xml**

0 and 100 are identical (lossless, reversibel)

1-99 fixed compression factor in percent of the uncompressed image size

#### **export.tcl**

Dynamic (PSNR) variable compression, consistent quality

We recommend to use Script in most of the cases, JP2 contains Quality Layer and Tiles and they can only be accessed via Script.

Dynamic needs more CPU, that has to be considered.

The saving from a dynamic to a fixed compression cannot be determined accurately. It depends on which static compression one places opposite. A value of 30% is however reliably a reasonable size. Apart from the saving of the size however the consistent quality is the crucial advantage. There are no "bad" pictures.

## 21.4 TCL procedure calls on export

Example:

```
<SCRIPTS>

  <BEFORE_EXPORT>BeforeExport $doc</BEFORE_EXPORT>

  <AFTER_EXPORT>PostProcessExport</AFTER_EXPORT>

  <EXPORT_IMAGE_NAME>GetImageFileNameForExport $doc $page
$outputname</EXPORT_IMAGE_NAME>

  <EXPORT_ALTO_NAME>GetAltoFileNameForExport $doc $page
$outputname</EXPORT_ALTO_NAME>

  <EXPORT_IMAGE>$image -settag Artist "CCS"</EXPORT_IMAGE>

  <SPLIT_DOCUMENT>SplitDoc</SPLIT_DOCUMENT>

  <DOC_NAME>GetDocName</DOC_NAME-->

  <SUBDOC_NAME>Book_SubDocName $subDocNum $subDoc</SUBDOC_NAME>

</SCRIPTS>
```

### Before export

```
<BEFORE_EXPORT>BeforeExport $doc</BEFORE_EXPORT>
```

By default don't change the xml configuration here. Instead write TCL procedure in project-export.tcl. This procedure must be defined in the TCL file, even if it is empty.

Sample:

```
proc CustomBeforeProcessExport { doc } {
  set title [$doc -data TITLE]
  set p [string last \ $title ]
  set pmk [string range $title [expr $p+1] end]
  globalvar exportname ***EXPORT***\\CCS_University\\$pmk
}
```

### After export

```
<AFTER_EXPORT>PostProcessExport</AFTER_EXPORT>
```

By default don't change the xml configuration here. Instead write TCL procedure in project-export.tcl. This procedure must be defined in the TCL file, even if it empty.

Available variables:

\$outputname          the full output path and name

Sample:

```
proc CustomPostProcessExport { outputname } {
  source [expandpath ***project_cfg***\\my_Project\\TEXT-Transform.tcl]
  TEXT-Transform $outputname
}
```

### Export image name

```
<EXPORT_IMAGE_NAME></EXPORT_IMAGE_NAME>
```

This procedure shall return the image name to be used for the image that will be created. If an empty string is returned or no script is defined, the template from xml configuration file is used.

Available variables:

\$doc	the current document object
\$page	the current page object
\$outputname	the full output path and name

Sample:

In **xml** file

```
<EXPORT_IMAGE_NAME> MyImageName $outputname $doc $page </EXPORT_IMAGE_NAME>
```

In **tcl** file

```
proc MyImageName { outputname doc page } {  
  set title [$doc -data TITLE]  
  set p [string last \\ $title ]  
  set pmk [string range $title [expr $p+1] end]  
  set imgprefix pmk-[string range $pmk 3 end]  
  set imgname [format $imgprefix-%02d.tif [$page -num]]  
  return $imgname  
}
```

### Export image

```
<EXPORT_IMAGE></EXPORT_IMAGE>
```

This procedure may modify the image itself. So it can be used to set image tags for example.

Available variables:

\$image	the current image object
\$page	the current page object
\$outputname	the full output path and name

Sample:

In **xml** file

```
<EXPORT_IMAGE> ExportMyImageTag $image </EXPORT_IMAGE>
```

In **tcl** file

```
proc ExportMyImageTag { image } {  
  $image -settag Artist "Scanned by CCS Content Conversion Specialists"  
}
```

### Export partial image

```
<PARTIALIMAGES> </PARTIALIMAGES>
```

Condition to select just dedicated partial images in export:

```
<PARTIALIMAGES  
  CONDITION="this.type  
  &quot;Advertisement&quot;;">Advertisements</PARTIALIMAGES>
```

=

## Document name

`<DOC_NAME></DOC_NAME>`

If the document is not split into independent sub-documents, you may use this script to give a special name to the document. This script is called at the very beginning.

Available variables: none

Sample:

In **xml** file

`<DOC_NAME>MyDocName [doc]</DOC_NAME>`

In **tcl** file

```
proc MyDocName { doc } {  
    set title [$doc -data TITLE]  
    set hierarchy [$doc -getproperty ownData]  
    set metadata [$hierarchy -getproperty Metadata]  
    set date [$metadata -getproperty date]  
    set docName [format "%s-%s" $date $title]  
    return $docName  
}
```

## Path and Name Configuration

`<PATH>***EXPORT***\my_Project</PATH>`

`<Name>@ (TITLE) </NAME>`

A `<PATH>` must be specified for the export. To handle path definitions more flexible, you should use docWizz internal path macros. Path macros are replaced by the definition during runtime. This mechanism enables you easily to change the environment or move storage locations without touching each configuration file. A macro is called by encapsulating the path name with three stars before and after.

All files that will be exported will be exported in or below the specified path.

The `<NAME>` tells the master name of the exported files. Files that are exported may use this master name and add special extension.

## Split Document

`<SPLIT_DOC>SplitDoc</SPLIT_DOC>`

This procedure is used, if the document shall be split on export into multiple isolated documents. It is used especially to split newspapers in issues. Then each issue will be treated as one single document. As well used for multiple Document processing. It is a special type to process small documents as one unit in docWizz.

Sample:

In **xml** file

`<SPLIT_DOC>SplitDoc</SPLIT_DOC>`

## Subdocument Name

`<SUBDOC_NAME></SUBDOC_NAME>`

If the document was split, you may use this script to give special names for each of the documents. This script is called once per sub document.



Available variables:

<code>\$subDocNum</code>	the current document number
<code>\$subDoc</code>	the current document

Sample:

In **xml** file

```
<SUBDOC_NAME>MyIssueName $subDocNum $subDoc</SUBDOC_NAME>
```

In **tcl** file

```
proc MyIssueName { subDocNum subDoc } {
    set docContent [$subDoc -getproperty docContent]
    set ch [$docContent -getproperty children]
    set issue [$ch -getat 0]
    set metadata [$issue -getproperty Metadata]
    set date [$metadata -getproperty date]
    if { $date == "" } {
        set issuenamename "Target"
    } {
        set issuenamename [format "%s%s%s" [string range $date 6 9] [string range
$date 3 4] [string range $date 0 1]]
    }
    globalvar ExportPageCnt 0
    return $issuenamename
}
```

## 21.5 XML Output File Configuration

```
<FILES>
<FILE name="ALTO">
  <NAME>*-ALTO####</NAME>
  <ENCODING>UTF-8</ENCODING>
  <ROOTFORMAT>ALTO</ROOTFORMAT>
  <VALIDATE>0</VALIDATE>
  <CONDITION>(!ExistProp(doc.profile, "OnlyPageLinking") ||
doc.profile.OnlyPageLinking.enabled == 0) & &
(!ExistProp(doc.profile, "OnlyMetadata") ||
doc.profile.OnlyMetadata.enabled == 0) & & (GetField(this,
"ANALYSIS") != "None")</CONDITION>
</FILE>
<FILE name="METS">
  <NAME>*-METS</NAME>
  <ENCODING>UTF-8</ENCODING>
  <ROOTFORMAT>METS</ROOTFORMAT>
  <VALIDATE>0</VALIDATE>
  <CONDITION>GetField(this, "ANALYSIS") != "None"</CONDITION>
</FILE>
</FILES>
```

For each type of file a `<FILE>` element is specified. The name attribute is used for naming only.

The `<NAME>` element specifies the final file name of the created file. An asterisk is replaced by the master name of the current export. Each hash represents a digit of a counter. If the `<NAME>` is not specified for ALTO files, they will get the same name as the related image file, extension "xml".

`<ENCODING>` tells the encoding to be used.

`<ROOTFORMAT>` is predefined for METS and ALTO and must not be changed against the schema definition. Please note that this options requires internet access.

`<CONDITION>` tells, under which conditions the files shall be created. Please note that this condition is applied for the entire group, not for each page.

`<VALIDATE>` tells whether each created xml file shall automatically validated by docWizz.

## 22 Automated QA: Reject Conditions

A reject mechanism is integrated into the processing workflow of docWizz. It serves as an automated quality assurance and allows automated document processing without manual checking each document. Reject conditions are defined to automatically break processing in case of unexpected results or if quality of analysis does not match pre-defined requirements.

Documents will be set to the status "Reject" which is visible e.g. in the document pool. If a reject condition is true the document stops in the next interactive job and allows an operator to check it. Please refer to the chapter "Manual QA: The QA Workstep" of this document to learn more about status handling of documents. In the following you will read about setting, activating and defining reject conditions.

Error message contains a brief information which makes it easy to locate the suspicious item:

- Page (real number, not like PageIndex starting with 0)
- Type of element
- Position of element on page (or position of included item if it is a structural container)
- Text (or text of one of its children)
- Some may miss if not available.

Default Rejects in `project-cfg\default\reject\RejectByEnt.xml` has "Default\_" in front for the xml name..

Each reject remains persistent and an operator is requested to accept each reject. The rejects cannot be ignored on routing. Each has to be accepted inside. A warning message is displayed.

### 22.1 Explanation of rejects

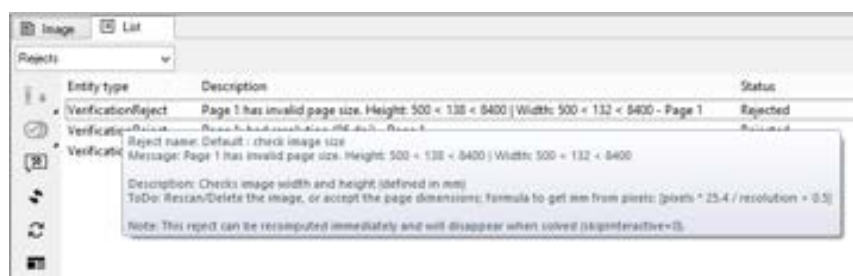
Reject conditions can be activated and their parameters can be set on project basis inside the project configuration directory. To do so, a file "RejectByEnt.xml" must be assigned to the project by placing it in a subfolder (named "reject") of the according project configuration (project-cfg\projectname\reject). It can be copied from project-cfg\Default\reject and edited afterwards.

## 22.2 Enable / disable rejects

To enable a reject condition, inside the RejectByEnt.xml file, navigate to the desired reject, and set the parameter "enabled" to "1"; for disabling a reject, set "enabled" to "0".

```
<REJECT>
  <SOURCE>
    <REJECTS>
      <JOB name="PrepareImport">
        <JOB name="Scan">
          <REJECT_CONDITION name="Default_CheckImageSize" enabled="1"/>
          <REJECT_CONDITION name="Default_CheckImageResolution" enabled="1"/>
          <REJECT_CONDITION name="Default_CheckForOriginalImage" enabled="1" type="Critical"/>
          <REJECT_CONDITION name="Default_CheckForMultiPageImage" enabled="1"/>
          <REJECT_CONDITION name="Default_CheckDocPageCount" enabled="1"/>
          <REJECT_CONDITION name="Default_Doc_EvenPages" enabled="1"/>
          <REJECT_CONDITION name="Default_CheckSetupImportPath" enabled="1" type="Critical"/>
        </JOB>
      </JOB>
    </SOURCE>
  </REJECT>
```

The reject are displayed in docWizz interface in List view -> Rejects.



## 22.3 Reject types

A reject can be set as a VerificationReject or as a CriticalReject.

### VerificationReject

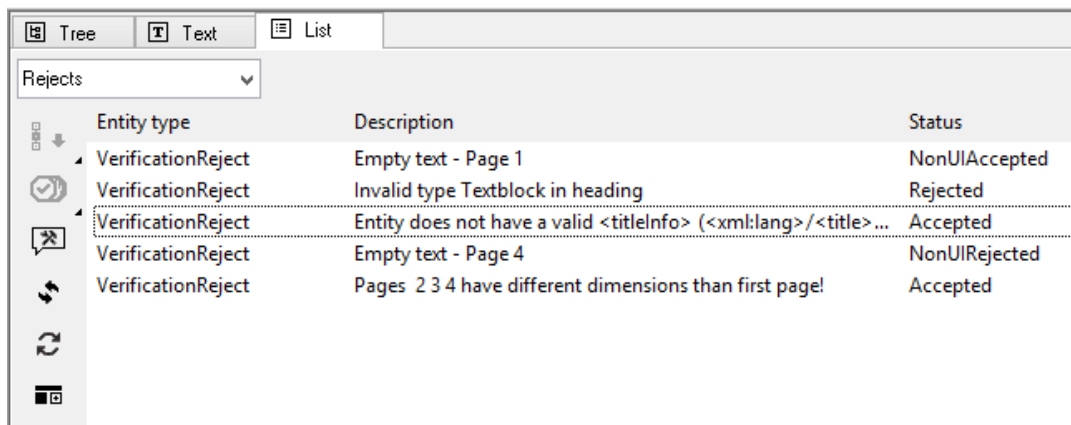
It is set as a warning, and the user has the possibility to accept the rejected condition or to fix the problem that generated the reject. In RejectByEnt.xml file, these rejects don't have the "type=Critical" attribute.

### CriticalReject

It is more of an error and the user cannot accept this type of reject. The only possibility to process the document further is to fix the problem that generated the reject. In RejectByEnt.xml file, these rejects have the "type=Critical" attribute.

## 22.4 Reject status

<b>Rejected</b>	is used for both types of rejects (Verification and Critical). As long as a reject has the status "Rejected" the document cannot be processed to the next step.
<b>Accepted</b>	is used only for VerificationReject. This status will appear when a reject condition is accepted by the user. With the status "Accepted", the reject will no longer be re-computed and the document can be processed to the next step. <i>The "accepted" rejects will still be displayed in the Rejects list even if the problem that generated the reject is fixed.</i>
<b>NonUIRejected</b>	is the same status as "Rejected", the difference being where the reject is computed. Rejects that have skipInteractive="1" in RejectByEnt.xml file are not computed in interface, usually because they are time consuming, instead they are computed in background services – this will keep the reject displayed in interface, even if the problem was fixed, until the document is re-processed. With this status for a reject, the document cannot be processed to the next step.
<b>NonUIAccepted</b>	is used when rejects with "NonUIRejected" status are accepted by the user. With this status for a reject, the document can be processed to the next step.



Entity type	Description	Status
VerificationReject	Empty text - Page 1	NonUIAccepted
VerificationReject	Invalid type Textblock in heading	Rejected
VerificationReject	Entity does not have a valid <titleInfo> (<xml:lang>/<title> ...	Accepted
VerificationReject	Empty text - Page 4	NonUIRejected
VerificationReject	Pages 2 3 4 have different dimensions than first page!	Accepted

## 22.5 Reject conditions

docWizz Reject Conditions are used to break automated processing in case of unexpected results or when the quality of analyse results does not match the pre-defined requirements.

A library of reject conditions is implemented in docWizz\_RejectLIB.tcl. It can be found in the script folder and must not be changed. Typically those pre-defined reject conditions have some parameters which might be configured.

Per project a file "reject.xml" can be placed in a sub folder (named "reject") of the project configuration. A sample looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
<REJECT>
  <JOB name="Import">
    <CONDITION name="CheckImageSizeAndResolution" enabled="1"
params="{ 300 400 600 } 500 8400 500">CheckImageSizeAndResolution</CONDITION>
  </JOB>
  <JOB name="DetectPageNumbers">
    <CONDITION name="CheckPageNumberSequence" enabled="1"
params="">CheckPNErrors</CONDITION>
    <CONDITION name="CheckOCRConfidence" enabled="1"
params="80000">CheckOCRConfidence</CONDITION>
  </JOB>
</REJECT>
```

For each job, one or more conditions can be defined:

The name attribute is just for identification and has no special meaning.

The enabled attribute allows to easily turn on and off reject conditions with removing a condition from the file.

The params attribute contains parameters to be used for the condition.

Each parameter is separated by space. Whenever a parameter itself contains spaces, the parameter must be encapsulated by { } brackets.

The condition script itself is stored as text of the CONDITION element. It will be called with the given parameters.

If no reject.xml is defined, the default one from project-cfg\default folder will be used.

For available reject conditions, see documentation inside **docWizz\_RejectLIB.tcl**. Any other additional reject condition might be defined in a script called projectname-reject-jobname.tcl (e.g. CCS-reject-DetectLayoutElements.tcl).

It must return one of:

```
succeed
failure: any description
critical: any description
```

**succeed** shall be returned if everything is ok and no reject shall be done.

CriticalError status is used when a problem is detected with a document, in transfer, while error docs is used to find documents that are lost (on loader appear as sent and on manager appear as into remoteQA)

`failure` shall be returned, if the user may decide that he wants to continue processing, even the conditions fails.

`critical` shall be returned if processing must not be continued unless the problem has been solved.

## 22.6 Configurations

The main User configuration file is project-cfg/default/RejectByEnt.xml

Format:

```
<REJECT>
  <SOURCE>
    <SRC> </SRC>
  </SOURCE>
  <REJECTS>
    <RULE name="" rejID="" applyTo="" place="">
      <initreject></initreject>
      <entprint></entprint>
      <reject></reject>
      <param name=""></param>
      <descr name="ret"></descr>
      <help></help>
      <help_qa></help_qa>
    </RULE>
  </REJECTS>
  <JOB name="">
    <AUTO_FIX name="" enabled="1"></AUTO_FIX>
    <REJECT_CONDITION name="" enabled="1"/>
  </JOB>
</REJECT>
```

SRC	Include the tcl scripts that are used to define the reject procedures. Project defined script procedures are added in here to be loaded in the system. For example: <pre>&lt;SRC&gt;***DATA***\Script\Reject\RejectByEnt_cond.tcl&lt;/SRC&gt; &lt;SRC&gt;***DATA***\Script\Reject\RejectByEnt_condOCR.tcl&lt;/SRC&gt;</pre>	
RULE	Define a reject rule. It contains:	
	name	identify the reject in the job configurations.
	rejld	contain the class of the reject.
	applyTo	hold a condition (own ccs expression in tcl) that describe the entities where the reject is defined
	place	it search the target entities on a part of the document. Values empty=all, D=doc,H=Hiereachy,P=pages,DP=DocPages
	initreject	a procedure that is run at the beginning of the reject alg to init the entities props. First on the document all the initreject procedures are called from all conditions defined
	entprint	a procedure that compute the invariant of the target entity. This procedure is used to implement the not equal operator. If the result of this procedure is different the entities are different.
	reject	this procedure actually implement the reject procedure
	param	hold the parameters that are used in the reject procedure (see implementation detail and usage for more info )



	descr	hold strings that are used to return the reject description (see implementation detail and usage for more info )
	help	short description for the "advance user" that configure the reject to know what is about. (for now is not loaded in the dw interface)
	help_qa	hint to the qa user about how the reject can be solved or the reason why appear
	For example:	<pre> &lt;RULE name="InvalidBlockType" rejID="4" applyTo="IsKindOf(\$ent, 'LayoutElement') " place="P"&gt; &lt;entprint&gt;Reject::GetLayoutElementEntPrint&lt;/entprint&gt; &lt;/entprint&gt;     &lt;reject&gt;Reject::InvalidBlockType&lt;/reject&gt;     &lt;param name="InvalidBlockType"&gt;Textblock     Headline Illustration Caption RunningTitle     Advertisement Publishing_Stmt PageNumber Author     Overline Subheadline&lt;/param&gt;     &lt;descr name="ret"&gt;Invalid block type:     %s&lt;/descr&gt;     &lt;help&gt;Determin if blocks with other types that     in param exist&lt;/help&gt;     &lt;help_qa&gt;Change type or delete the     entities&lt;/help_qa&gt; &lt;/RULE&gt; </pre>
JOB		Hold the conditions that should be evaluated in that job. name the name of the job
AUTO_FIX		keep an extra script that is executed before running all the conditions. This has the purpose to do automatic 100% sure fixes so some rejects will not fail. The name is like a short description
REJECT_CONDITION		hold a reject that will be evaluated in this step if enabled = 1. The name hold the name of the reject RULE that was configured above.

## 22.7 Setting and activating Reject conditions

Reject conditions can be activated and their parameters can be set on project basis inside the project configuration directory. To do so, a file "reject.xml" must be assigned to the project by placing it in a subfolder (named "reject") of the according project configuration (`project-cfg\projectname\reject`). It can be copied from `project-cfg\Default\reject` and edited afterwards.

To enable a reject condition set the parameter "enabled" to "1", to disable a condition set the parameter to "0". The params attribute contains parameters to be used for the condition. Please enter a value for each of the parameters that apply for the condition. Each parameter is separated by space. Whenever a parameter itself contains spaces, the parameter must be encapsulated with { } brackets. The condition script itself is stored as text of the CONDITION element. It will be called with the given parameters. o accept it manually, if only one page was existing.

In order to perform project specific settings of reject conditions edit the file `project-cfg\projectname\reject\Reject.xml`.

The following reject conditions are pre-defined:

Job	Condition
Scan	CheckImageSizeAndResolution
Re-Scan	CheckImageSizeAndResolution CheckForInvalidPages
VerifyPageFrames	CheckForInvalidPages CheckMissingFrames CheckFrameSizes
VerifyPages	CheckImageSizeAndResolution CheckForInvalidPages
VerifyLayoutElements	CheckForInvalidPages
VerifyPageNumbers	CheckPageNumberSequence CheckOCRConfidence CheckForInvalidPages
VerifyPagesHierarchy	CheckForInvalidPages
VerifyHierarchy	CheckForInvalidPages

## 22.8 How to deal with Reject conditions (example)

Here is a list containing example jobs and reject reasons and how to solve the problem:

Job	Reject Reason	Explanation	How to solve
C-PC	Document has a wrong frame on page(s) 10, 13	Frame on pages 10 and 13 are reaching outside the image.	Resize frame of pages 10 and 13 and make it smaller.
Z-RZ	Block(s) with invalid rectangle area on page(s): 8	One Block on page 8 reaches outside	Select List "Suspicious Blocks" and check the entries for page 8. Then resize the block. Most of the times it is a polygon which should be rectangular.
Z-RPS	1 page(s) have insufficient spell checking.	OCR confidence is blow a predefined value.	Double check the page! Is it upside down? Is the image corrupt? It might be that due to Advertisement with graphical elements etc OCR confidence is low. If the image is ok, operators must change page property to "6- as in original".
Z-RPS	The number sequence has errors.	There is a page number sequence error. Or an error has been corrected and the error has not been reset yet.	Double check page sequence (tree on left) for errors (red "!" sign). If page sequence is ok, change red flag of page to green.
S-RST	More than 6 text based elements have no OCR text inside.	Some text blocks (or other text based elements) have no content.	Select list "Suspicious Blocks" for "Check this block: OCR not computed/corrected (error)" to check these textblocks. For these blocks "reset OCR". OCR will then be run again on Master system
S-RST	6 Headlines have no OCR correction. (26 pages checked)	OCR result of some Headlines has not been corrected. Most probably the operator has missed them.	Select list "Suspicious Blocks" for "Check this block: OCR not computed/corrected (error)" to check and correct Headline.
S-RST	This Documents has 1 duplicate blocks	One physical item (e.g. a text block) is part of two logical containers (E.g. to articles).	Select list "Suspicious Blocks" for "duplicate Blocks" and remove from tree on left the block where it should not be located.
S-RST	Invalid block type: Author	Based on project configuration and Mark-up policy only certain zone classifications are allowed. All others are rejected.	Select list "Suspicious Blocks" for "Check this block! Invalid type (error)".

## 22.9 Defining new Reject conditions

Based on your requirements you can define new reject conditions in addition to the ones available in the file `docWizz_RejectLIB.tcl` located in the script folder.

To do so, you should be able to script TCL.

Create a script called `projectname-reject-jobname.tcl` (e.g. `CCS-reject-DetectLayoutElements.tcl`) to your project configuration folder where you might already find other project related scripts (`project-cfg\projectname`).

Note: Reject parameters are case sensitive.

Use the following schema:

```
***PROJECT_CFG***\\$projectCfg\\Reject\\$projectCfg-reject-$step.tcl
```

Add scripts to the file with the following notation:

```
proc FUNCTIONNAME { PARAMETER1 PARAMETER2 } {  
    ...  
    Code for checking  
    ...  
    return RETURNVALUE  
}
```

The procedures have to return defined return values. You can use the values for RETURNVALUE below:

```
succeed  
failure [description]  
critical [description]
```

`succeed` shall be returned if everything is ok and no reject shall be done.

`failure` shall be returned, if the user may decide that he wants to continue processing, even the conditions fails.

`critical` shall be returned if processing must not be continued unless the problem has been solved.

Finally, call the new procedure in `reject.xml` of your project configuration in the job you want to check.

```
<JOB name="VerifyPagesHierarchy">  
    <CONDITION name="FUNCTIONNAME" enabled="0"params="$PARAMETER1  
$PARAMETER2">FUNCTIONNAME</CONDITION>  
</JOB>
```

Reject allows to place "default-reject-STEP.tcl" which will be used if "PROJECT-reject-STEP.tcl" is missing.

## 22.10 Separate rejects by Monograph or Newspaper

To separate rejects by monograph or newspaper go to rejectbyent.xml

Via the doctype node per rule <doctype>Newspaper</doctype>

Example :

```
<doctype>Newspaper Serial</doctype>
```


If nothing is set it is for all.

```
<RULE          name="Default_CheckForNotAvailable"          rejID="5"
applyTo="IsKindOf($ent, 'LayoutElement') " place="P">
    <initreject></initreject>
    <entprint></entprint>
    <doctype>Newspaper Monograph Serial</doctype>
```

## 22.11 User interface

**Reject**  (only available in QA-Mode) allows to return the document to an normal user.

You can type a reason for rejecting this document - this is visible in the comment field.

If you select **QA**  you can switch QA mode on or off. You have to be authorized to use that mode. In QA mode documents are processed further only if this final check is made.

Please note: Documents are set to the status "Reject" either by the QA Operator in manual QA or if a documents is rejected through automated QA in server processing. The second is the case if certain reject conditions apply.

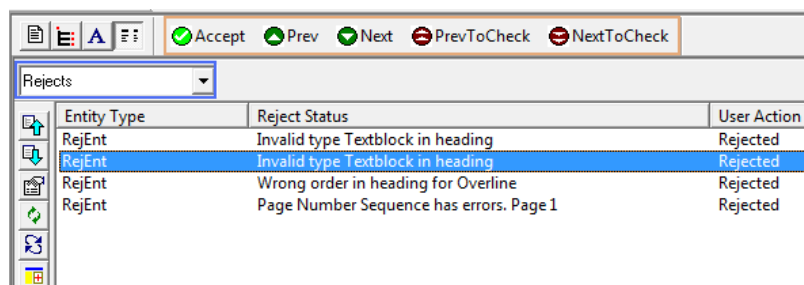
All rejects are disabled in default configuration !

### LIST Rejects

Contain the reject tool bar: Press Accept/Reject button to change the reject. Prev / Next jump to prev / next reject , PrevToCheck / NextToCheck jump to next reject that is not accepted by the user.

Whenever a user clicks "OnProcess" and a reject appears, the view is switched to OutlineListView and "Rejects" is selected.

RejectStatus contain the reason why the reject was raised. User action tells witch rejects has been accepted or not.



Entity Type	Reject Status	User Action
RejEnt	Invalid type Textblock in heading	Rejected
RejEnt	Invalid type Textblock in heading	Rejected
RejEnt	Wrong order in heading for Overline	Rejected
RejEnt	Page Number Sequence has errors. Page 1	Rejected

**Accept/Reject (Toggle):** Handle reject; only visible, when an entry is selected.

**Prev:** Go to previous entry in list.

**Next:** Go to next entry in list.

**PrevToCheck:** Go to previous reject that is not accepted.

**NextToCheck:** Go to next reject that is not accepted.

**Reject Status column:** Contains the reason why the reject was raised.

**User Action column:** Shows which rejects have been accepted or not.

Remark: If this list is empty this can happen because document has not been saved yet in Scan job, after save and re-opening the document, the rejects appear in ListView.

Functionality in C-PC: Double click in List View switches to image view instead of tree view If this is defined as start view and scrolls to page indicated by Reject or other list item. Also the rejects are displayed then on the right hand side.

## LIST StructureErrors

Contain now also all the entities that have rejects that are not accepted. The Error column holds the reject reason for that entity

StructureErrors	
Entity Type	Error
Article	This entity has rejects: 1: Invalid type Textblock in heading
Article	This entity has rejects: 1: Invalid type Textblock in heading
Article	This entity has rejects: 1: Wrong order in heading for Overline

## TREE

RejByEnt can be seen here as well, along with Status (checkbox), user and description.

The screenshot displays the RejByEnt tool interface. At the top, there is a toolbar with buttons for 'Reject' (red circle with a white 'X'), 'Prev' (green circle with a white arrow), 'Next' (green circle with a white arrow), 'PrevToCheck' (red circle with a white arrow), and 'NextToCheck' (red circle with a white arrow). Below the toolbar, a dropdown menu shows 'Default\_NP'. The main area is divided into two panes. The left pane shows a document tree structure with nodes like 'doc', 'pages', 'docContent: Hierarchy', '[1]: Issue', 'TitleSection', 'Content', '[1]: Article (1)', 'Heading', 'Content', 'Footnotes', 'Illustrations', 'Tables', and 'Rejects'. The right pane shows the details of the selected entity, 'Revolution im Stunden :'. It includes a 'Status' field with a dropdown menu set to 'Unknown', a 'Check box Accept / Reject' (checked), a 'Description' field with the text 'Invalid type Textblock in heading', and a 'User' field with the name 'Admin'. Below this, another entity is shown, 'Internationaler Intri :', with a 'Status' field set to 'RejEnt', a 'Check box Accept / Reject' (unchecked), a 'Description' field with the text 'Invalid type Textblock in heading', and a 'User' field with the name 'Admin'. A tooltip at the bottom left says 'Structural Error -> also the tool tip can be seen, as before'.

Contains the reject tool bar: Press Accept/Reject button to change the reject. Prev / Next jump to prev / next reject , PrevToCheck / NextToCheck jump to next reject that is not accepted by the user.

The check box can be used like the button Accept/Reject from tool bar. Beside this you can see the reject reason and also the last user that changed the status of the reject.

## 22.12 List of files and folders affected by the ReleaseByEnt feature

..\project-cfg\Default\Reject\	RejectByEnt.xml Book-DW.xml docWizz-VSCfg.xml	main configuration file main configuration file view style for Default NP
..\config\PVSCFG\	Newspaper-DW.xml docWizz-VSCfg.xml	tool bar configurations view style for Default NP
..\script\reject\	RejectByEnt_main.tcl (can be deleted to "disable" RejByEnt) reject_autofix_vh.tcl reject_autofix_vpf.tcl RejectByEnt_cond.tcl RejectByEnt_condOCR.tcl RejectByEnt_condOCRTextvsDict.tcl RejectByEnt_UI.tcl	main scripts
..\config\BasicCfg\	Basis_Ent.dsl	document structure library

## 22.13 Example: Merge to Advertisement

Example ..\PVSCFG\Newspaper-DW.xml in the steps: Z-RPS, Z-RZ, Z-RI

```
<ITEM_ACTION_ARRAY>
  <ACTION          name="4"          Script="MergeToAdvertisement"
  ValidationScript="ValidateIsImagePage" Description="Merges Blocks and makes
  them Advertisements"/>
</ITEM_ACTION_ARRAY>
```

In tree view in Z-RPS and Z-RI, select an "imagePage" in treeview. In context menu select "Actions" => "Merge To Advertisement".



## 23 Default rejects

Default rejects are listed in the following chapters.

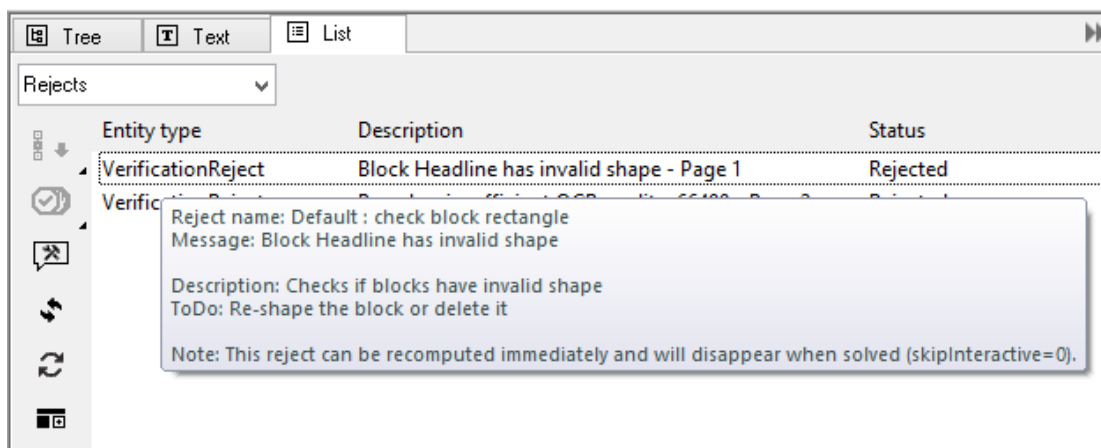
### 23.1 CheckArticleHeading

- Checks for zone types not allowed in article headings
- Has parameter for allowing multiple headlines in headings (AllowMultipleHeadlines)
- Carried out on hierarchy
- By default, it is a VerificationReject
- Used in: Review structure and text
- Processing in UI: Move the zone from Heading to Content.



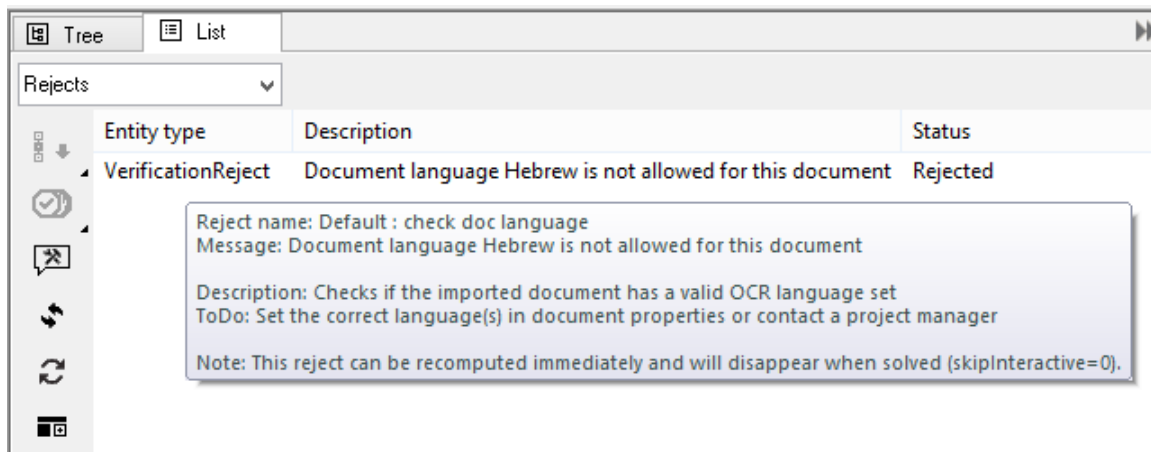
### 23.2 CheckBlockRectangle

- Checks if blocks have invalid shape
- Has parameter for checking for polygonal zones
- Carried out on pages
- By default, it is a VerificationReject
- Used in: Review zoning (disabled by default), Review page sequence, Review issues, Review structure and text
- Processing in UI: Re-shape the block or delete it.



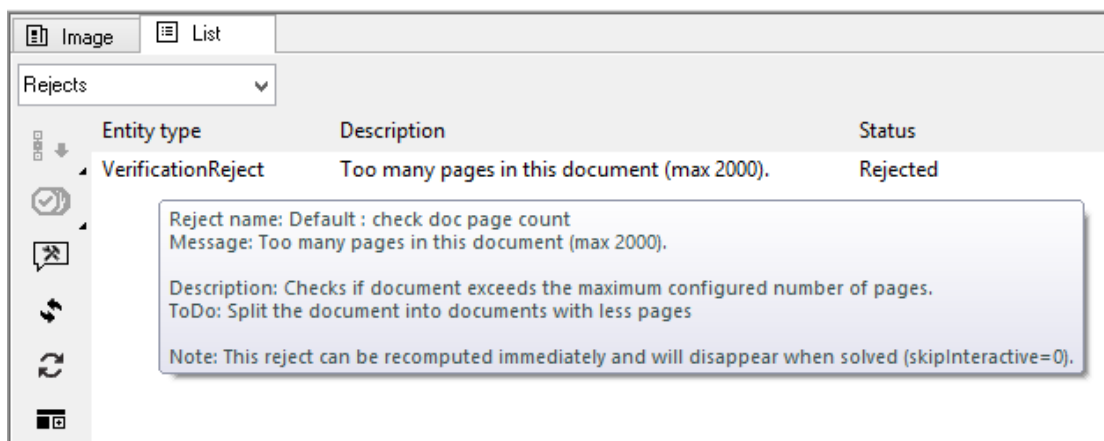
## 23.3 CheckDocLanguage

- Checks if the document has a valid OCR language set.
- Has parameter for language list (languageList)
- Is run on document
- By default, is a VerificationReject
- Used in: Review zoning, Review structure and text
- Processing in UI: Set the correct language(s) in document properties or contact a project manager



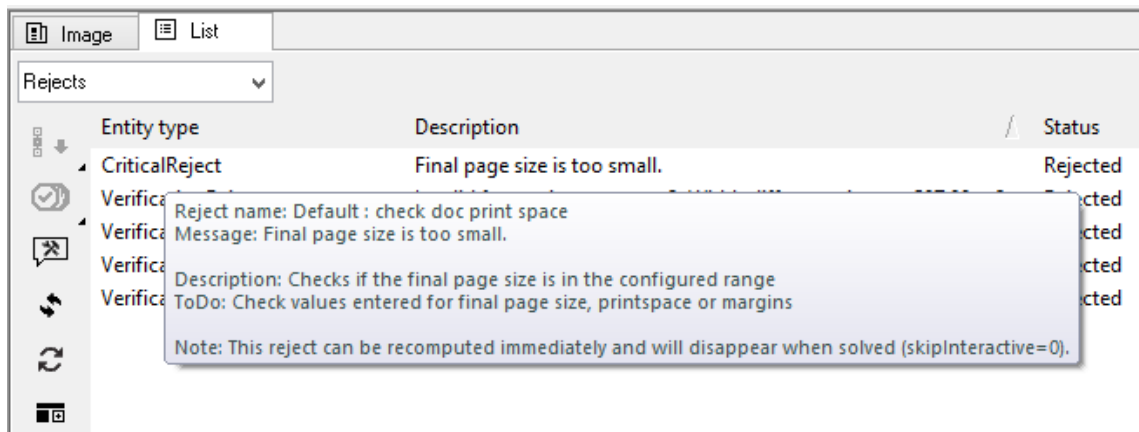
## 23.4 CheckDocPageCount

- Checks if document exceeds maximum configured number of pages
- Has parameters for minimum number of pages (minPages), maximum number of pages (maxPages) and maximum number of pages for newspapers (maxPagesNewspaper)
- Carried out on document
- By default, it is a VerificationReject
- Used in: Review import
- Processing in UI: Split the document into documents with less pages.



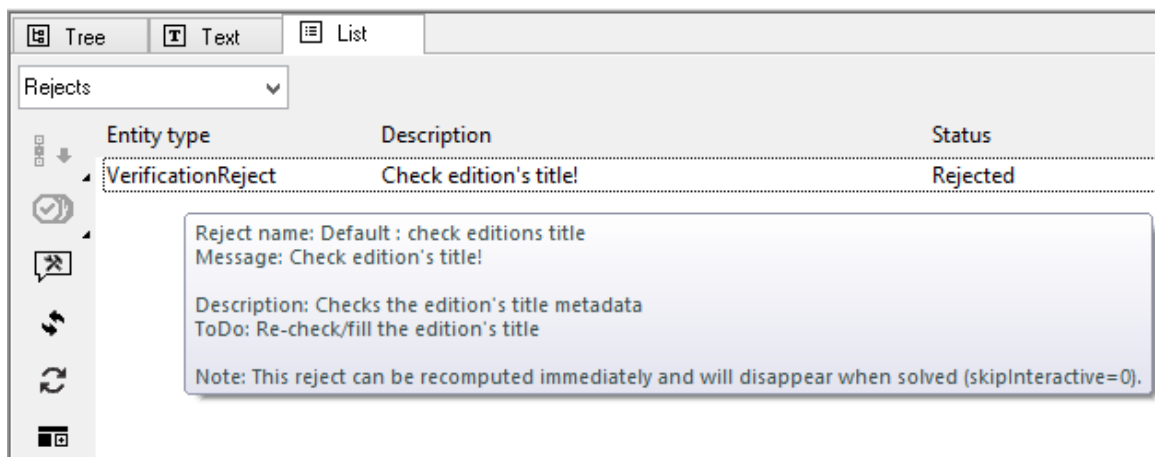
## 23.5 CheckDocPrintSpace

- Checks if the final page size is in the configured range
- Has parameters for minimum frame width (minWidth), maximum frame width (maxWidth), minimum frame height (minHeight) and maximum frame height (maxHeight)
- Carried out on document
- By default, it is a CriticalReject
- Used in: Prepare cropping
- Processing in UI: Check values entered for final page size, printspace or margins.



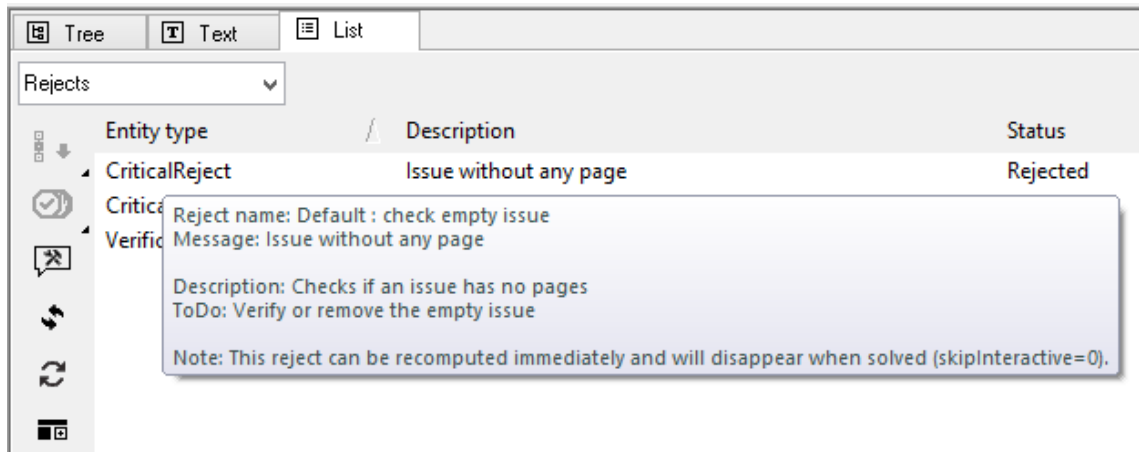
## 23.6 CheckEditionsTitle

- Checks edition's title metadata
- Has parameter for invalid titles list (InvalidTitlesList)
- Carried out on hierarchy
- By default, it is a VerificationReject
- Used in: Review structure and text
- Processing in UI: Re-check/fill the edition's title.



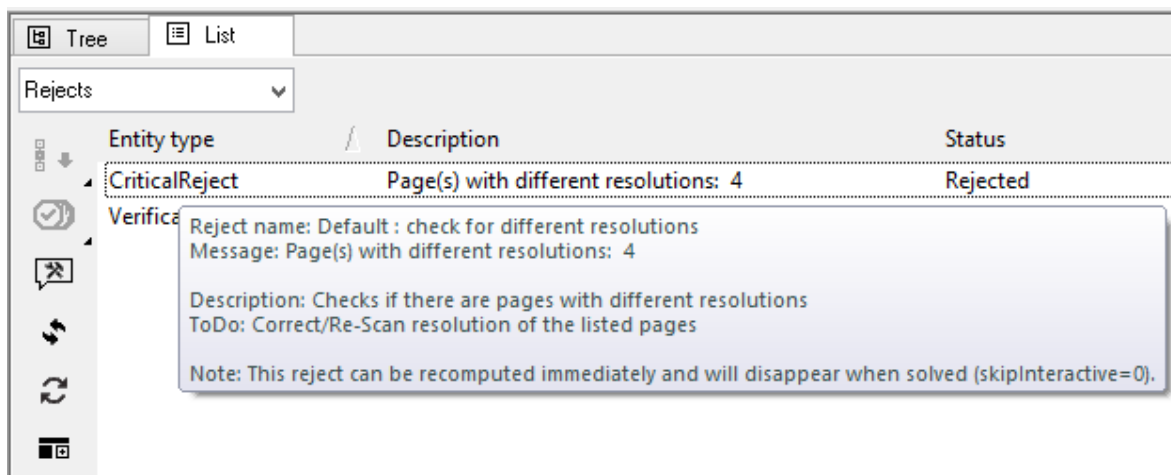
## 23.7 CheckEmptyIssue

- Checks if an issue has no pages
- Doesn't have parameters
- Carried out on hierarchy
- By default, it is a CriticalReject
- Used in: Review issues
- Processing in UI: Verify or remove the empty issue.



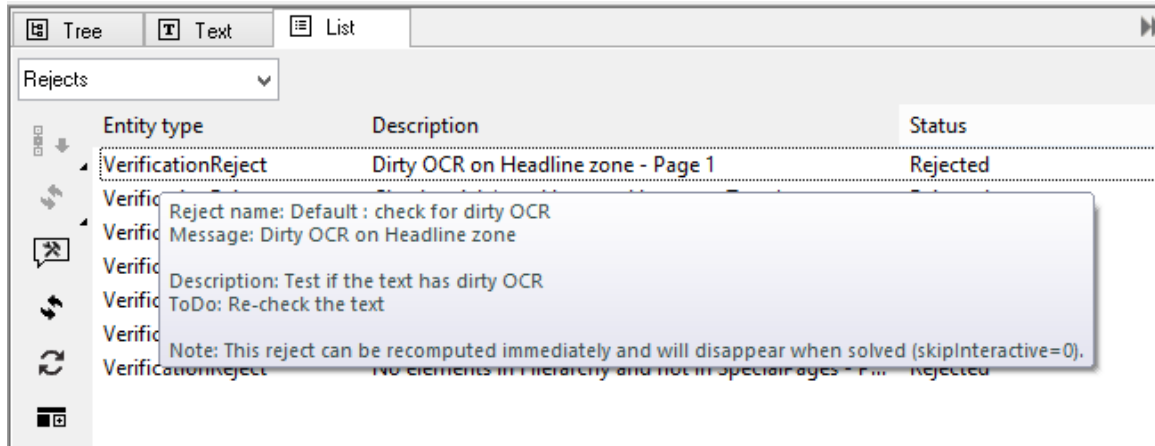
## 23.8 CheckForDifferentResolutions

- Checks if there are pages with different resolution
- Doesn't have parameters
- Carried out on document
- By default, it is a CriticalReject
- Used in: Review zoning, Review structure and text
- Processing in UI: Correct/Re-Scan resolution of the listed pages.



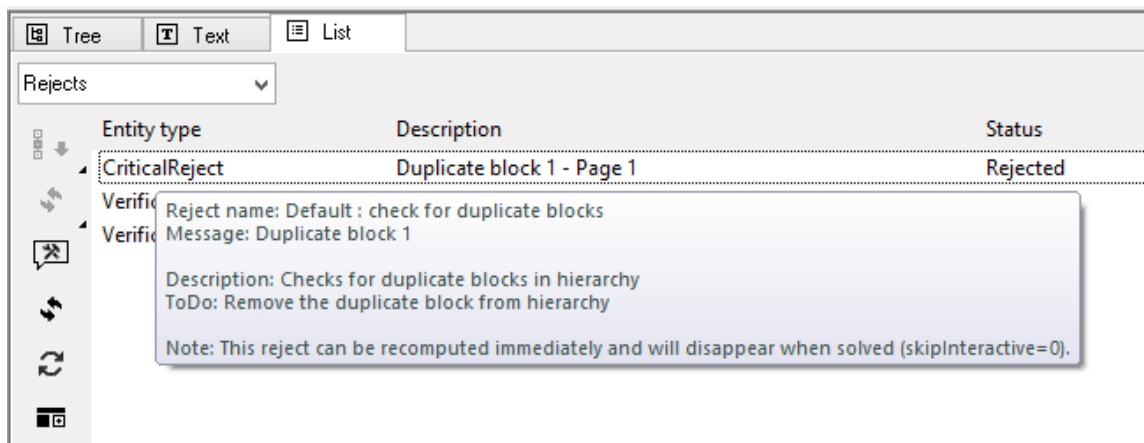
## 23.9 CheckForDirtyOCR

- Checks if the detected text has dirty OCR
- Has parameters for dirty characters
- Carried out on pages
- By default, it is a VerificationReject
- Used in: Review structure and text (disabled by default)
- Processing in UI: Re-check the text.



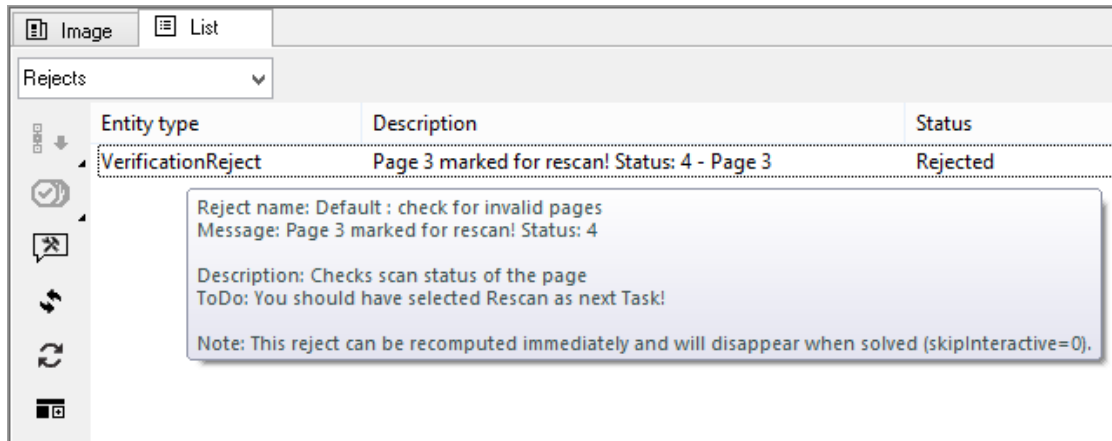
## 23.10 CheckForDuplicateBlocks

- Checks for duplicate blocks in hierarchy
- Doesn't have parameters
- Carried out on hierarchy
- By default, it is a CriticalReject
- Used in: Review structure and text
- Processing in UI: Remove the duplicate block from hierarchy.



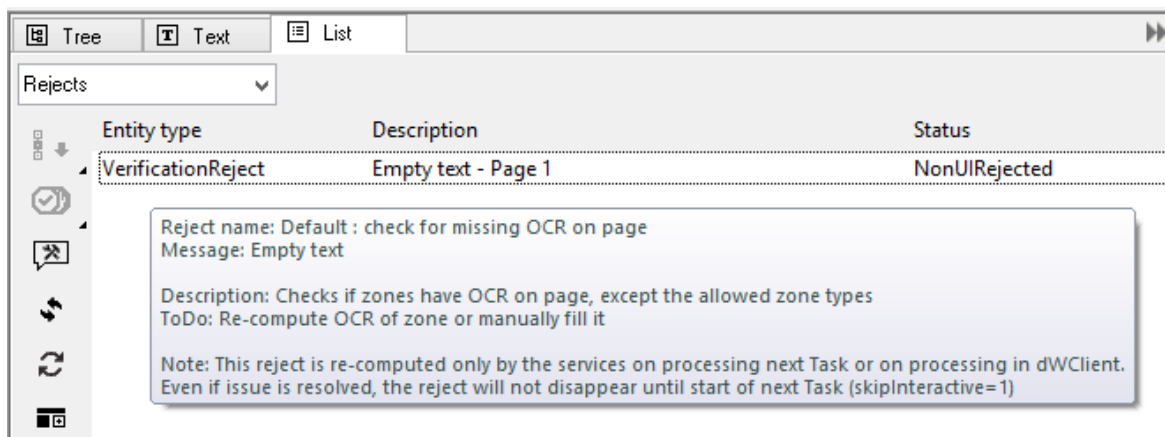
## 23.11 CheckForInvalidPages

- Checks if pages have the correct scan status
- Doesn't have parameters
- Carried out on document pages
- By default, it is a VerificationReject
- Used in: Rescan
- Processing in UI: You should have selected Re-Scan as next job!



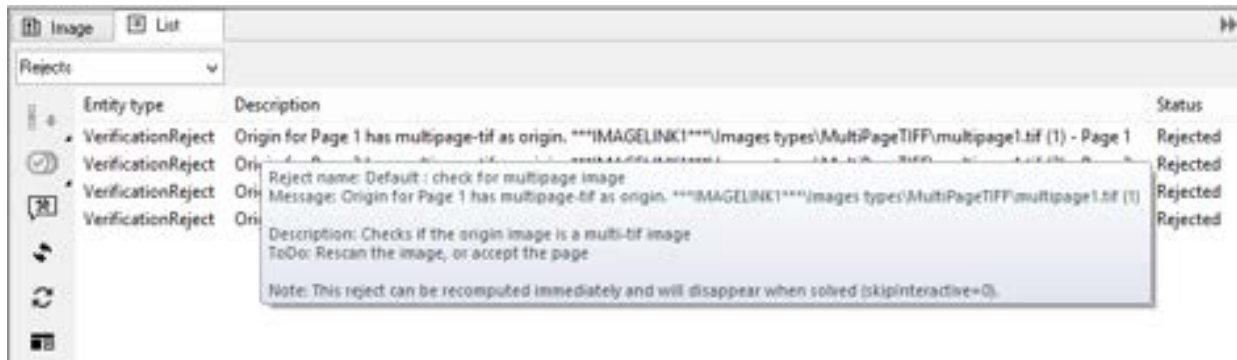
## 23.12 CheckForMissingOCROnPage

- Checks if zones have OCR.
- Has parameters for type of zones to skip, allowing empty zone types, maximum number of empty zones on page, and report on page (TypesToSkip, AllowEmptyTypes, maxEmpty, reportOnPage)
- Carried out on pages
- By default, it is a VerificationReject, nonUI (run in background)
- Used in: Review structure and text
- Processing in UI: Re-compute OCR of the zones or manually fill it.



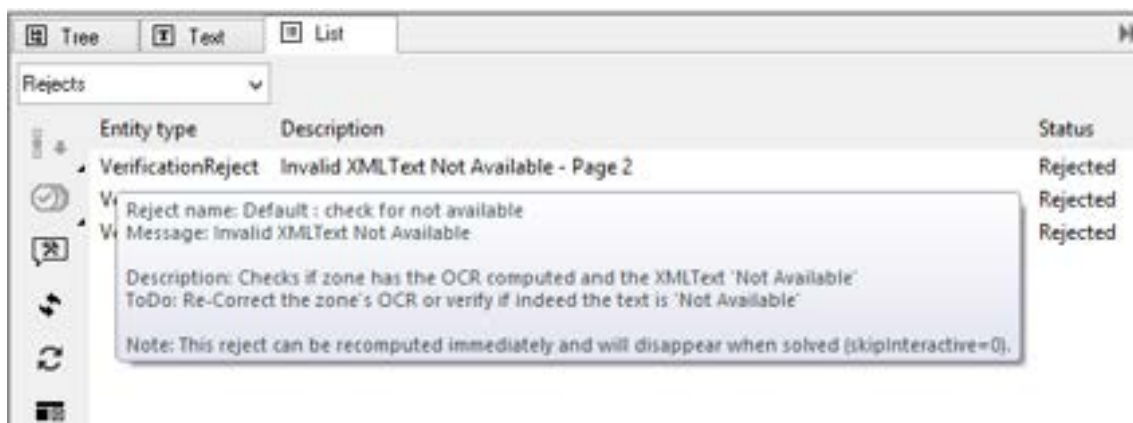
### 23.13 CheckForMultipageImage

- Checks if the image has a Multipage as origin
- Doesn't have parameters
- Carried out on document pages
- By default, it is a VerificationReject
- Used in: Review import
- Processing in UI: Rescan the image, or accept the page.



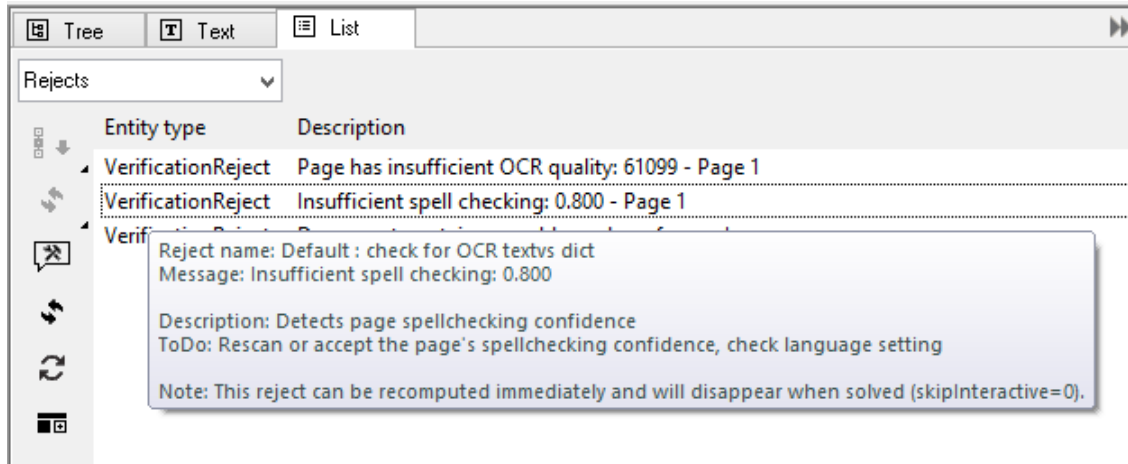
### 23.14 CheckForNotAvailable

- Checks if zone has the OCR computed and if the XMLText 'Not Available' is assigned to the zone.
- Doesn't have parameters
- Carried out on pages
- By default, it is a VerificationReject
- Used in: Review structure and text (disabled by default)
- Processing in UI: Re-Correct the zone's OCR or verify if indeed the text is 'Not Available'.



## 23.15 CheckForOCRTextvsDict

- Checks the page spell checking confidence
- Has parameter for confidence threshold (ocrInDictThreshold)
- Carried out on pages
- By default, it is a VerificationReject
- Used in: Review page sequence (disabled by default)
- Processing in UI: Rescan or accept the page's spell checking confidence, check language setting.



## 23.16 CheckForOriginImage

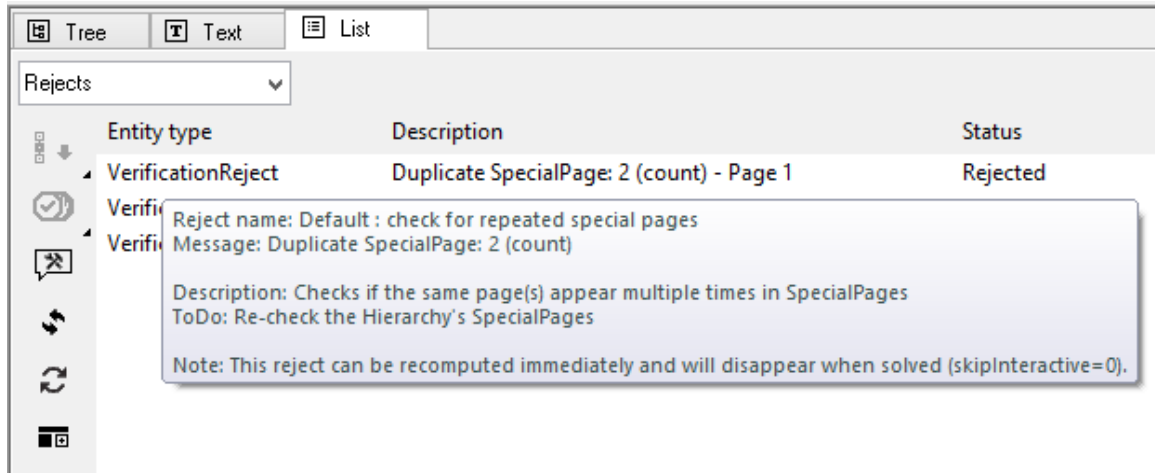
- Checks the image origin
- Doesn't have parameters
- Carried out on document pages
- By default, it is a CriticalReject
- Used in: Review import
- Processing in UI: Check origin image name/path or res-scan the image





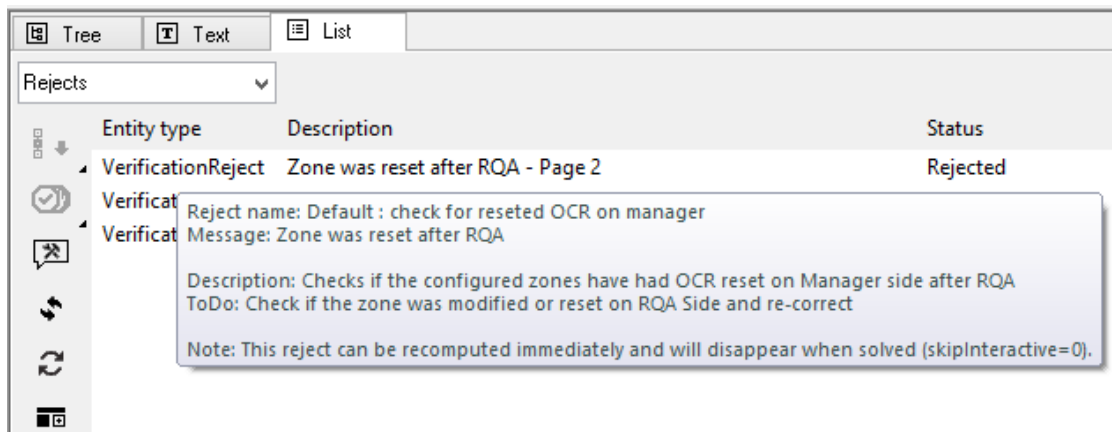
## 23.17 CheckForRepeatedSpecialPages

- Checks if the same page(s) appear multiple times in SpecialPages
- Doesn't have parameters
- Carried out on pages
- By default, it is a VerificationReject
- Used in: Review structure and text
- Processing in UI: Re-check the Hierarchy's SpecialPages



## 23.18 CheckForResetedOCROnManager

- Checks if the configured zones have had OCR reset on Manager side after RQA.
- Has parameter for zones type to be corrected (typesToCorrect) and for checking OCR place (checkOCRPlace)
- Carried out on pages
- By default, it is a VerificationReject
- Used in: Review structure and text (disabled by default)
- Processing in UI: Check if the zone was modified or reset on RQA side and re-correct.



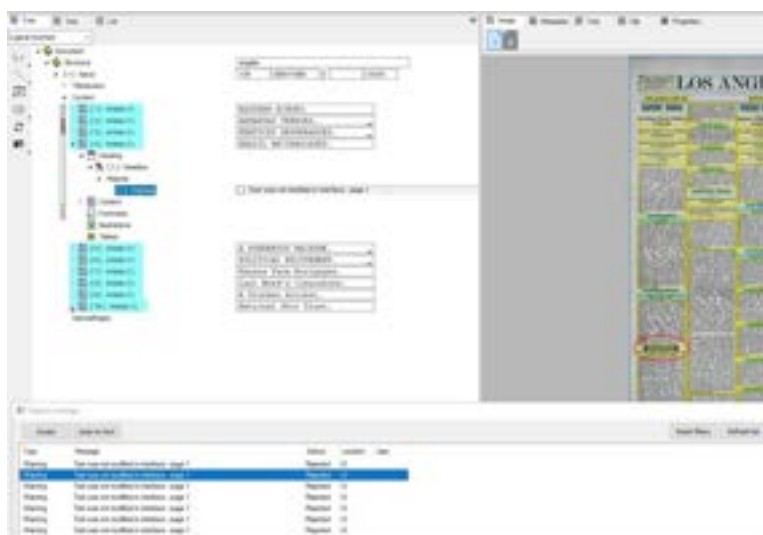
## 23.19CheckForSpecialPages

- Checks if there are any pages without zones in hierarchy and they are not in SpecialPages
- Doesn't have parameters
- Carried out on pages
- By default, it is a VerificationReject
- Used in: Review structure and text



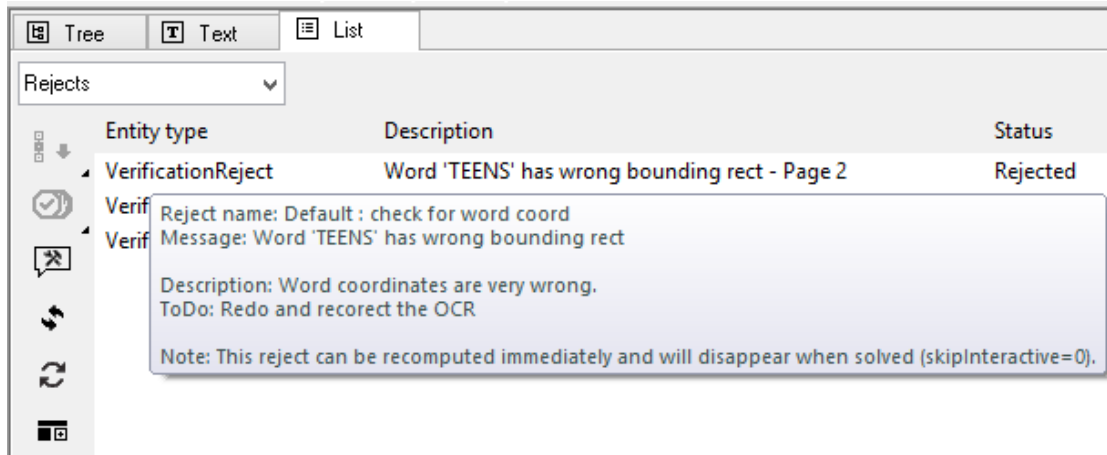
## 23.20CheckForUncorrectedOCR

- Checks if OCR has been corrected in interface on configured zones
- Has parameters for zone types to be corrected
- Carried out on pages
- By default, it is a VerificationReject
- Used in: Review structure and text
- Processing in UI: Verify if the OCR was detected correctly and accept the reject. Double check that no OCR correction is missing. Headlines get rejected as uncorrected even if they came directly from manager. Operator has to accept, even if the text is correct and he has nothing to correct manually.



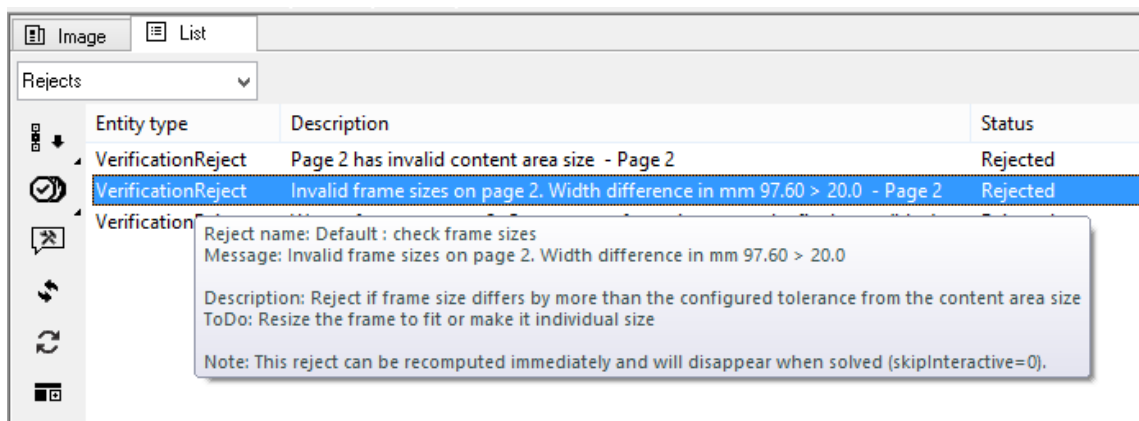
## 23.21 CheckForWordCoord

- Checks word coordinates
- Doesn't have parameters
- Carried out on pages
- By default, it is a VerificationReject
- Used in: Review structure and text
- Processing in UI: Redo and re-correct the OCR



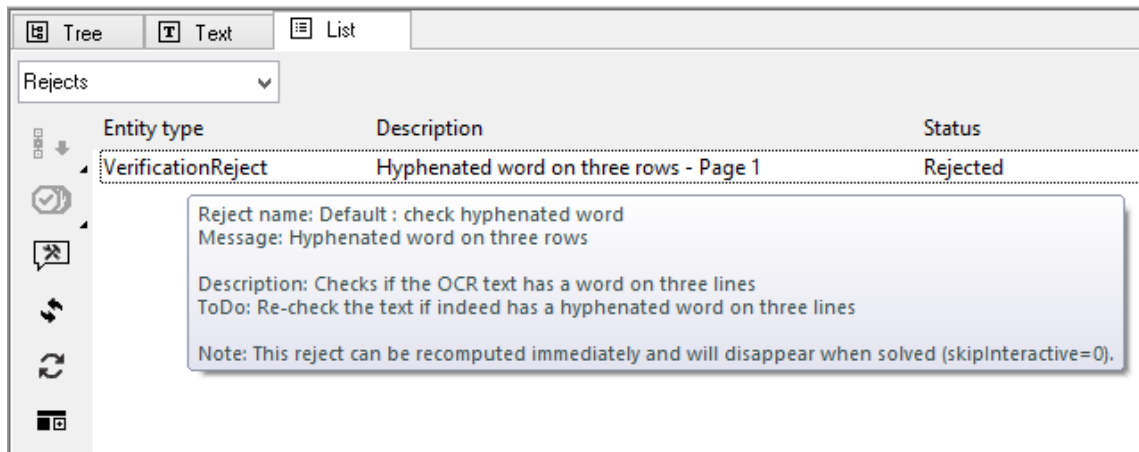
## 23.22 CheckFrameSizes

- Checks if frame size differs by more than the configured tolerance from the content area
- Has parameters for allowing frames on target pages and for maximum frame tolerance ("AllowFrameOnTarget", "maxFrameToleranceMM")
- Carried out on document pages
- By default, it is a VerificationReject
- Used in: Prepare cropping
- Processing in UI: Resize the frame to fit or make it individual size.



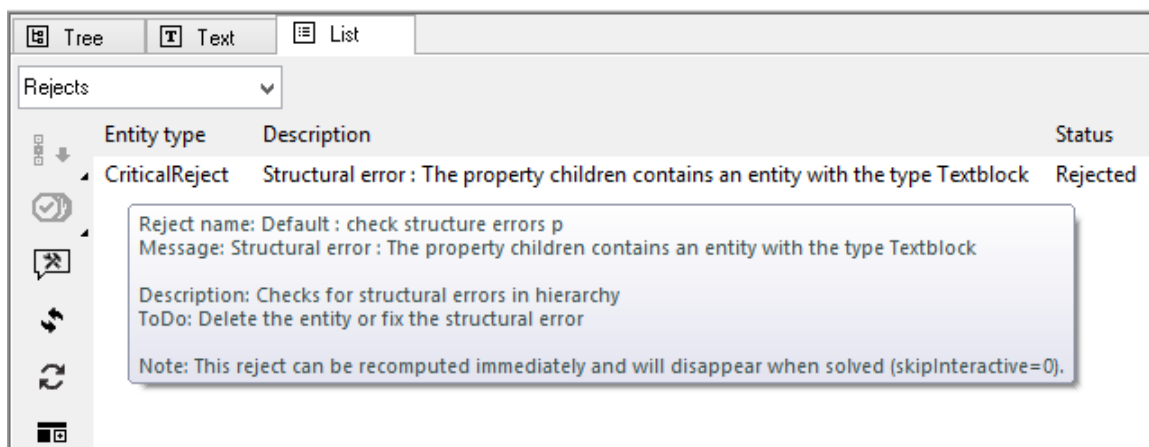
## 23.23 CheckHyphenatedWord

- Checks if OCR text has a word on three lines
- Has parameter for zone types to correct
- Carried out on pages
- By default, it is a VerificationReject
- Used in: Review structure and text (disabled by default)
- Processing in UI: Re-check the text if indeed has a hyphenated word on three lines



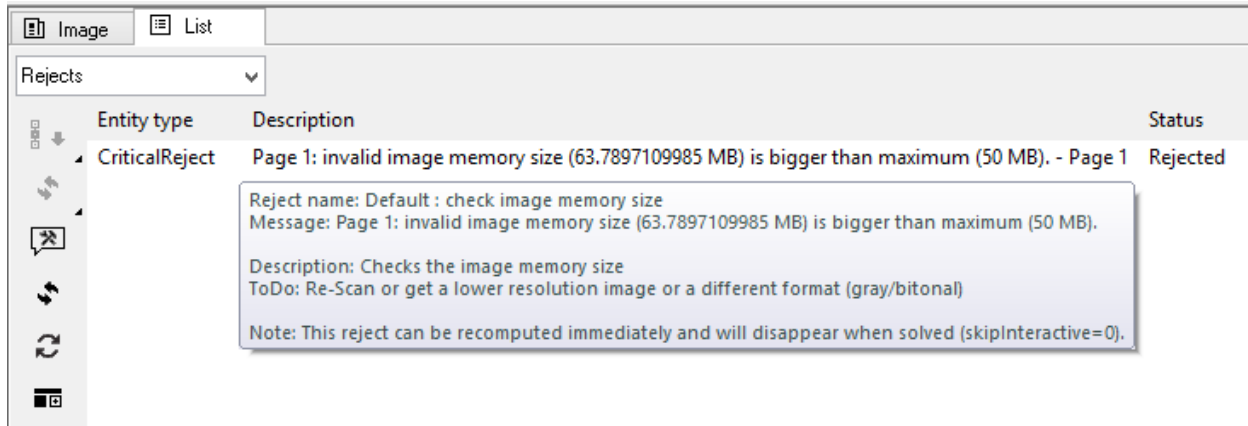
## 23.24 CheckIllustrationLikeStructs

- Checks the content of IllustrationLikeStructures if it's empty or does not contain valid zone types
- Doesn't have parameters
- Carried out on hierarchy
- By default, it is a CriticalReject
- Used in: Review structure and text
- Processing in UI: Verify the content of the IllustrationLikeStruct.



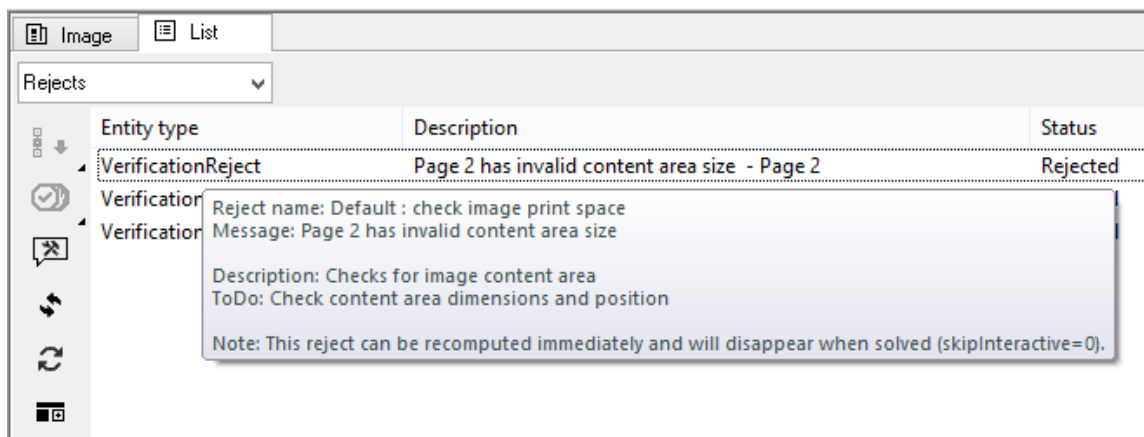
## 23.25 CheckImageMemorySize

- Checks the image memory size
- Has parameter for maximum size in Mb
- Carried out on document pages
- By default, it is a CriticalReject
- Used in: Prepare cropping
- Processing in UI: Re-Scan or get a lower resolution image or a different format (gray/bitonal).



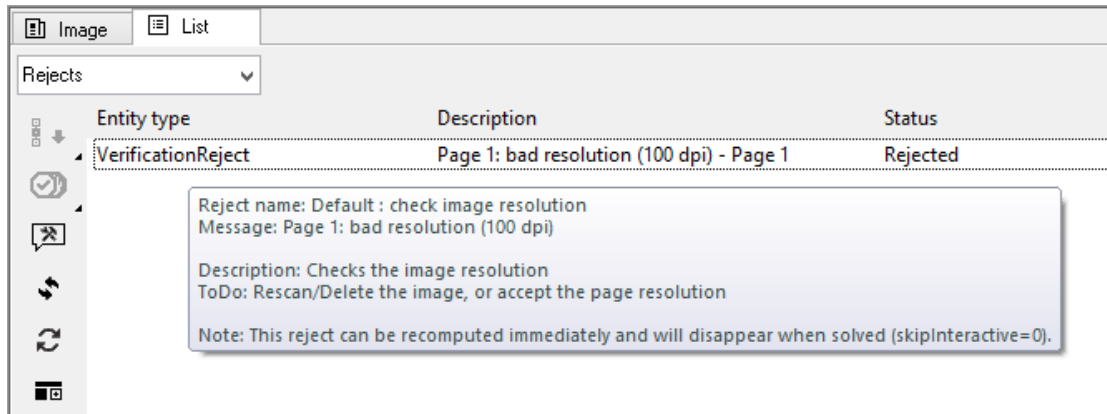
## 23.26 CheckImagePrintSpace

- Checks the image print space (content area)
- Has parameter for page types to skip (6=spine; 7=edge)
- Carried out on document pages
- By default, it is a VerificationReject
- Used in: Prepare cropping
- Processing in UI: Check the content area dimensions and position.



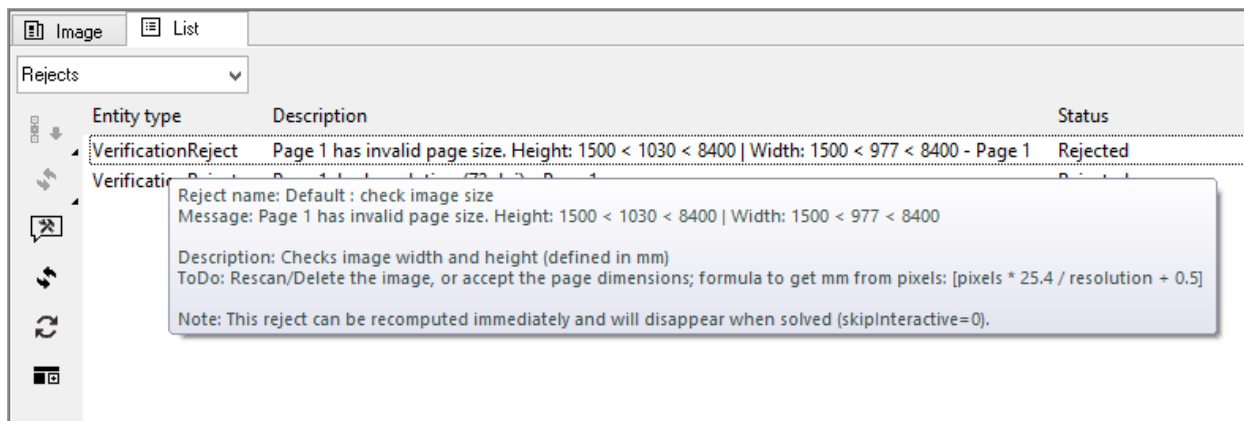
## 23.27 CheckImageResolution

- Checks the image resolution
- Has parameter(s) for accepted resolution values (ex: 300 400 600)
- Carried out on document pages
- By default, it is a VerificationReject
- Used in: Rescan, Review import, Prepare cropping, Review cropping
- Processing in UI : Rescan/Delete the image, or accept the page resolution



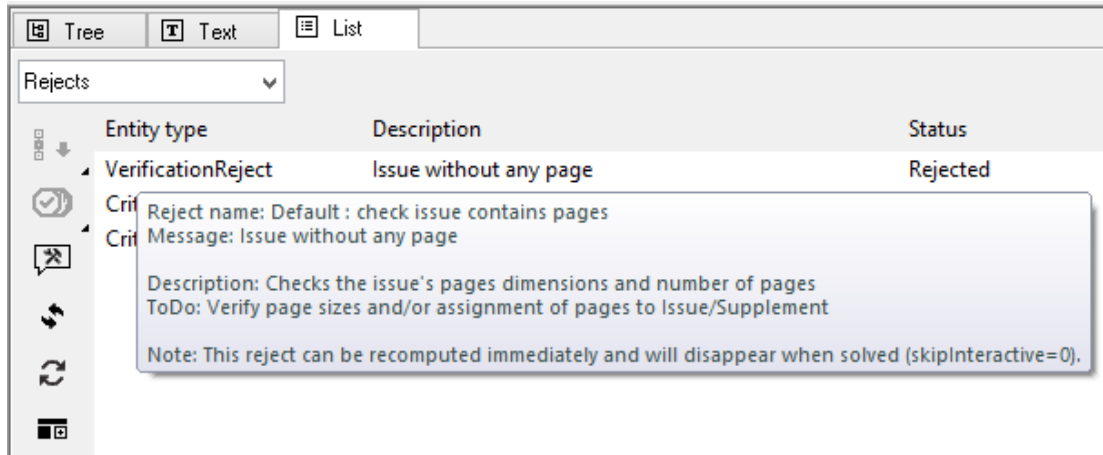
## 23.28 CheckImageSize

- Checks the image size
- Has parameters for minimum and maximum page dimensions
- Carried out on document pages
- By default, it is a VerificationReject
- Used in: Rescan, Review import, Prepare cropping, Review cropping
- Processing in UI: Rescan/delete the image, or accept the page dimensions



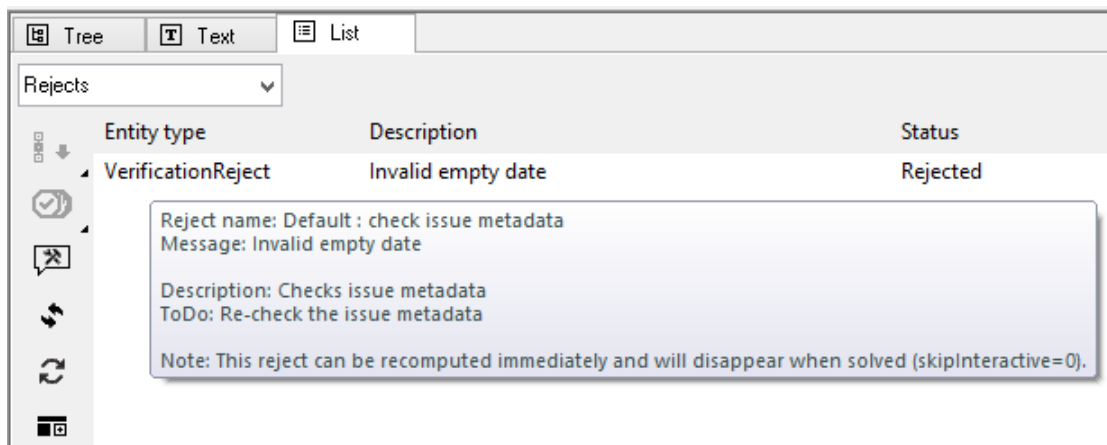
## 23.29 CheckIssueContainsPages

- Checks the issue's pages dimensions and number of pages
- Has parameter for skipping single pages (skipSingle) and skipping individual pages (skipIndividual)
- Carried out on hierarchy
- By default, it is a VerificationReject
- Used in: Review issues
- Processing in UI: Verify page sizes and/or assignment of pages to Issue/Supplement.



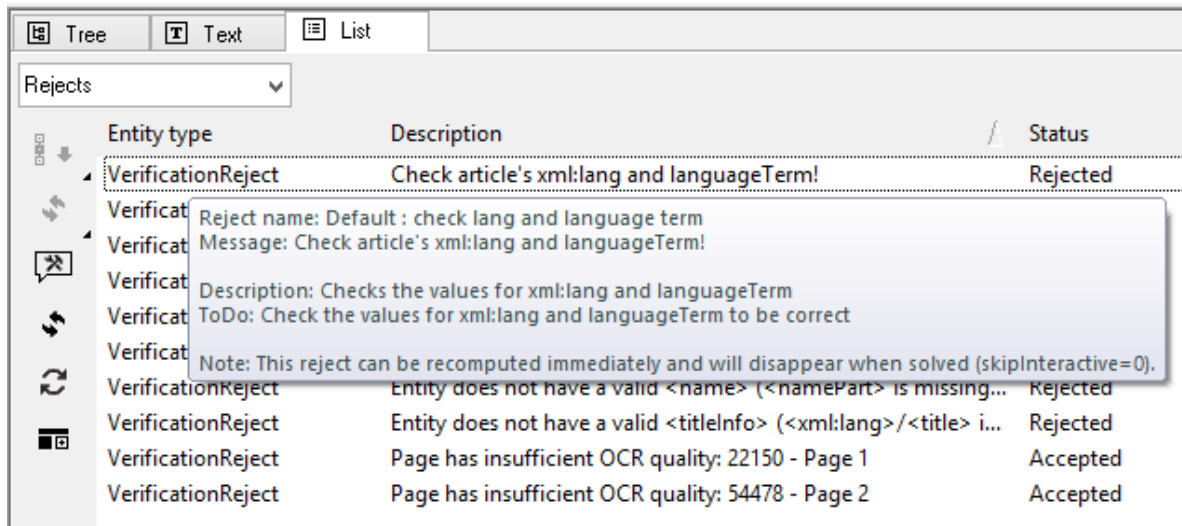
## 23.30 CheckIssueMetadata

- Checks issue metadata
- Doesn't have parameters
- Carried out on hierarchy
- By default, it is a VerificationReject
- Used in: Review issues, Review structure and text
- Processing in UI: Re-check the issue metadata



## 23.31 CheckLangAndLanguageTerm

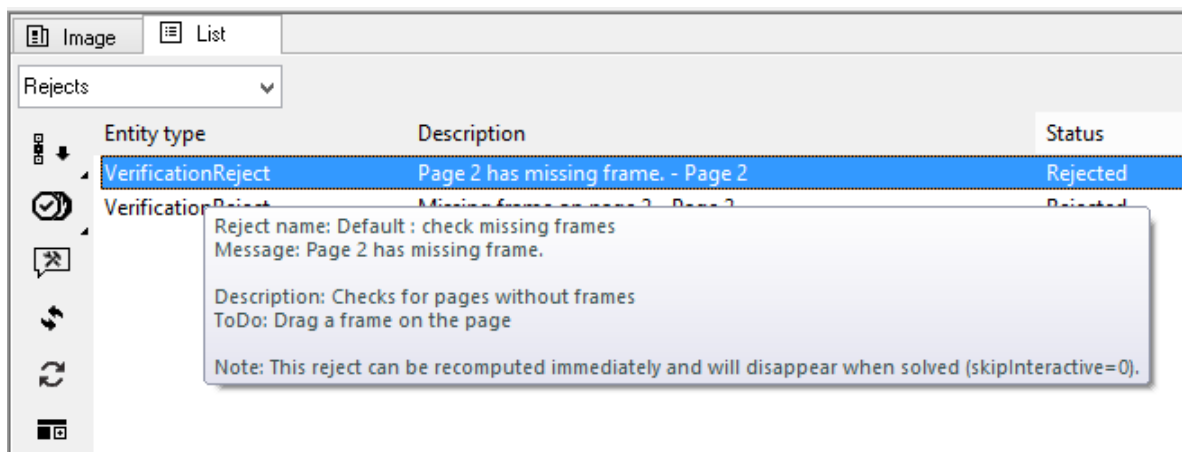
- Checks the values for xml:language and languageTerm
- Doesn't have parameters
- Carried out on hierarchy
- By default, it is a VerificationReject
- Used in: Review structure and text
- Processing in UI: Check the values for xml:lang and languageTerm metadata values.



Entity type	Description	Status
VerificationReject	Check article's xml:lang and languageTerm!	Rejected
VerificationReject	Reject name: Default : check lang and language term Message: Check article's xml:lang and languageTerm!	
VerificationReject	Description: Checks the values for xml:lang and languageTerm ToDo: Check the values for xml:lang and languageTerm to be correct	
VerificationReject	Note: This reject can be recomputed immediately and will disappear when solved (skipInteractive=0).	
VerificationReject	Entity does not have a valid <name> (<namePart> is missing...	Rejected
VerificationReject	Entity does not have a valid <titleInfo> (<xml:lang>/<title> i...	Rejected
VerificationReject	Page has insufficient OCR quality: 22150 - Page 1	Accepted
VerificationReject	Page has insufficient OCR quality: 54478 - Page 2	Accepted

## 23.32 CheckMissingFrames

- Checks if all pages have frames
- Has parameters for allowing frames on target pages, allowing double pages to have only one frame and allowing Start page and End page to have only one frame ("AllowFramesOnTarget", "AllowSingleFrameOnDoublePageDoc" and "AllowSingleFrameStartEndPage")
- Carried out on document pages
- By default, it is a VerificationReject
- Used in: Prepare cropping (disabled by default)
- Processing in UI: Drag a frame on the page

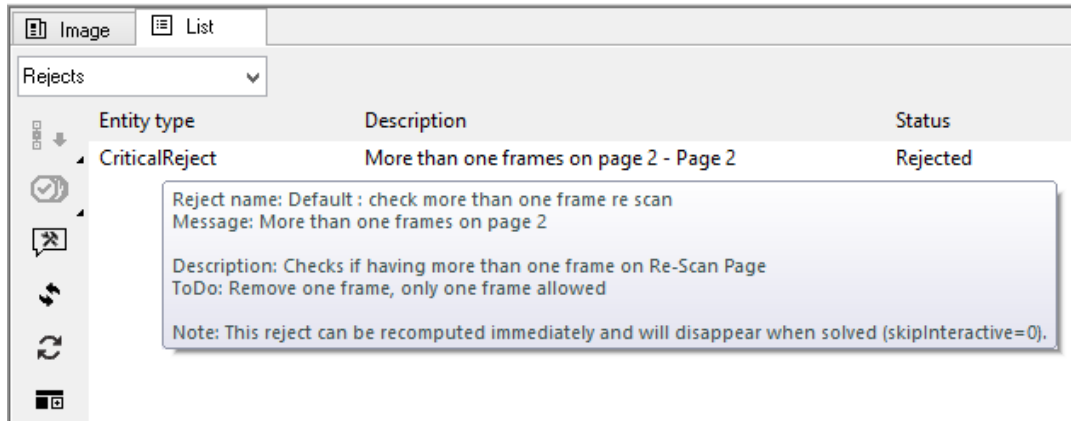


Entity type	Description	Status
VerificationReject	Page 2 has missing frame. - Page 2	Rejected
VerificationReject	Reject name: Default : check missing frames Message: Page 2 has missing frame.	
VerificationReject	Description: Checks for pages without frames ToDo: Drag a frame on the page	
VerificationReject	Note: This reject can be recomputed immediately and will disappear when solved (skipInteractive=0).	



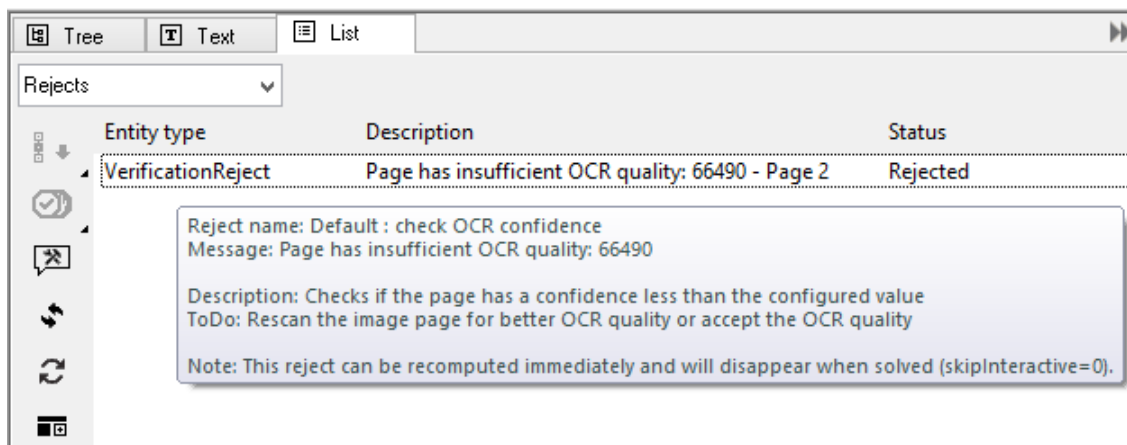
### 23.33 CheckMoreThanOneFrameReScan

- Checks if there are more than one frame on Rescan page
- Doesn't have parameters
- Carried out on document pages
- By default, it is a CriticalReject
- Used in: Rescan
- Processing in UI: Remove one frame, only one frame allowed



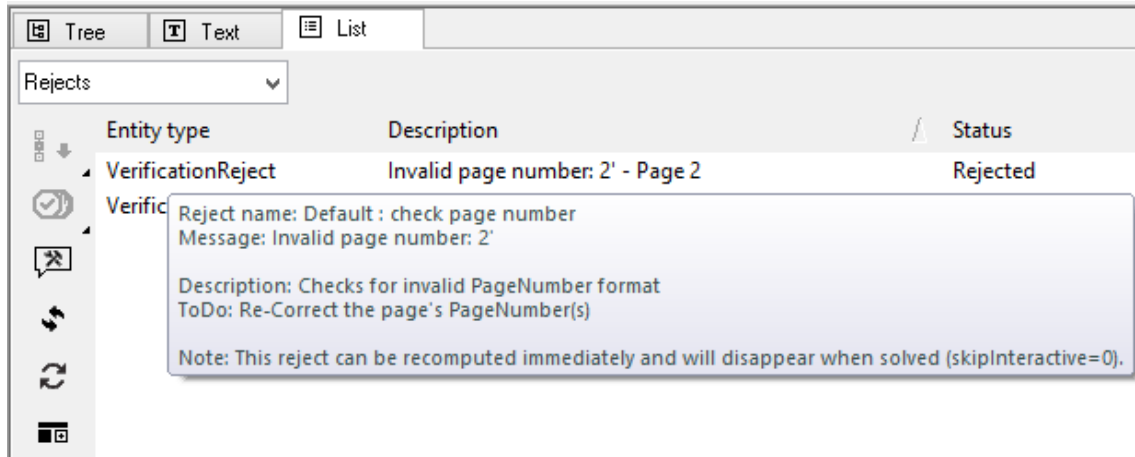
### 23.34 CheckOCRConfidence

- Checks OCR confidence of a page
- Has parameters for minimum OCR confidence on regular pages (minOCRConfidence) and minimum OCR confidence on Advertisement pages (minOCRConfidenceOnAdvPage)
- Carried out on pages
- By default, it is a VerificationReject
- Used in: Review page sequence
- Processing in UI: Rescan the image page for better OCR quality or accept the OCR quality.



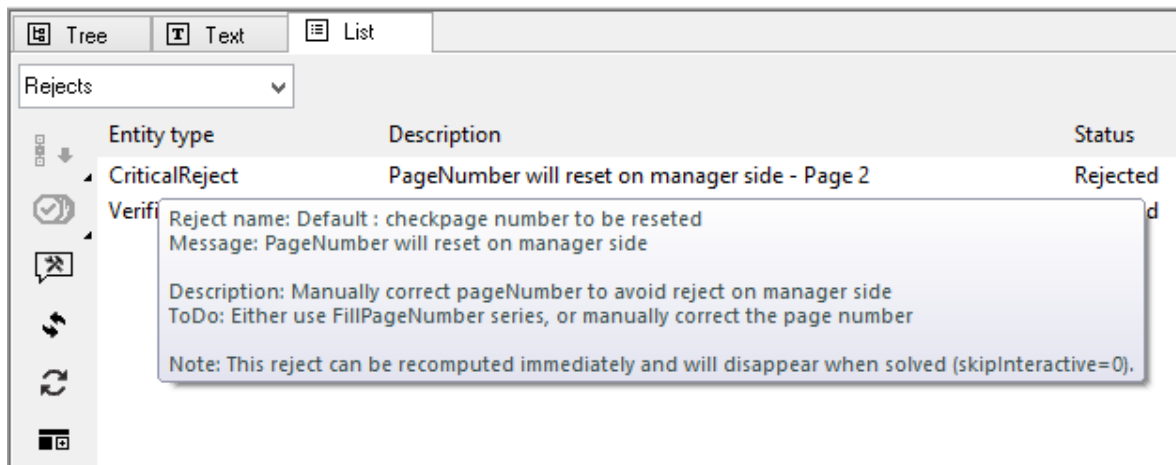
### 23.35CheckPageNumber

- Checks for invalid PageNumber format
- Doesn't have parameters
- Carried out on pages
- By default, it is a VerificationReject
- Used in: Review page sequence, Review structure and text
- Processing in UI: Re-Correct the page's PageNumber(s).



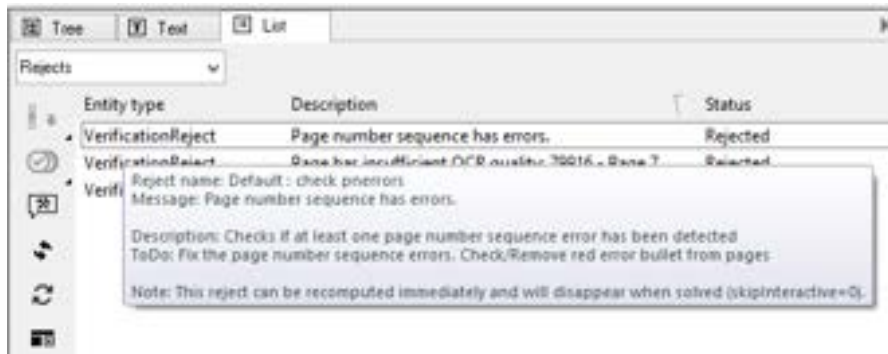
### 23.36CheckpageNumberToBeReseted

- Checks if there are pageNumbers that will be reset on manager side
- Doesn't have parameters
- Carried out on pages
- By default, it is a CriticalReject
- Used in: Review structure and text
- Processing in UI: Either use FillPageNumber series, or manually correct the page number.



## 23.37CheckPNErrors

- Checks for errors in page number sequence
- Doesn't have parameters
- Carried out on document
- By default, it is a VerificationReject
- Used in: Review page sequence, Review structure and text
- Processing in UI: Use group mode to add the zone to hierarchy or delete the zone.

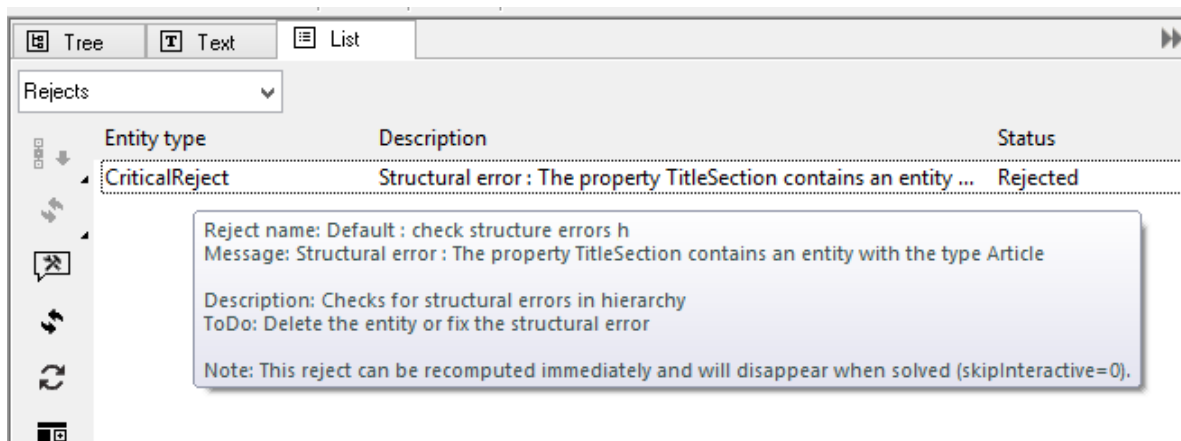


## 23.38CheckSetupImportPath

- Checks if document import path is a network path
- Doesn't have parameters
- Carried out on document
- By default, it is a CriticalReject
- Used in: Setup import, Review import
- Processing in UI: Set network import path to document

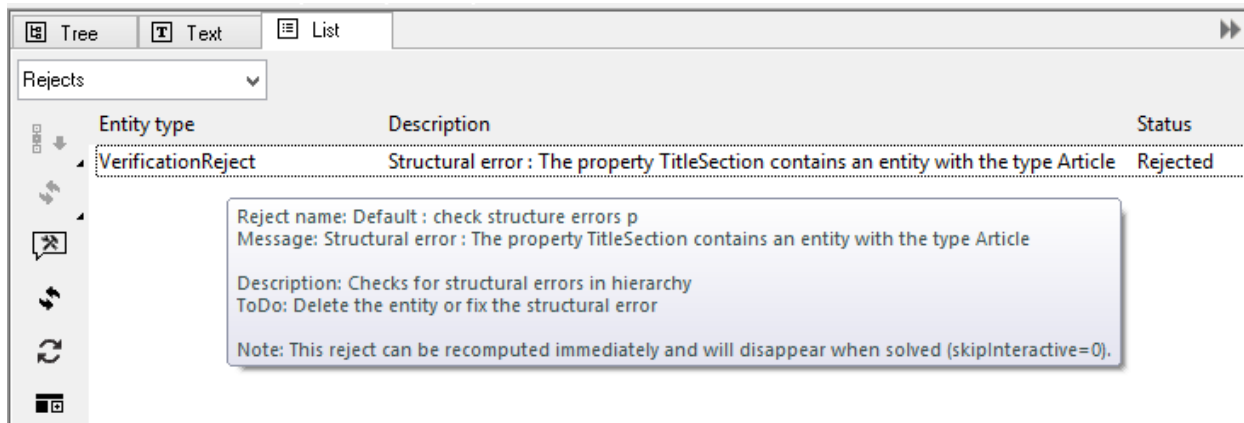
## 23.39CheckStructureErrorsH

- Checks for structural errors in hierarchy
- Doesn't have parameters
- Carried out on hierarchy
- By default, it is a CriticalReject
- Used in: Review structure and text
- Processing in UI: Fix the hierarchy structural error.



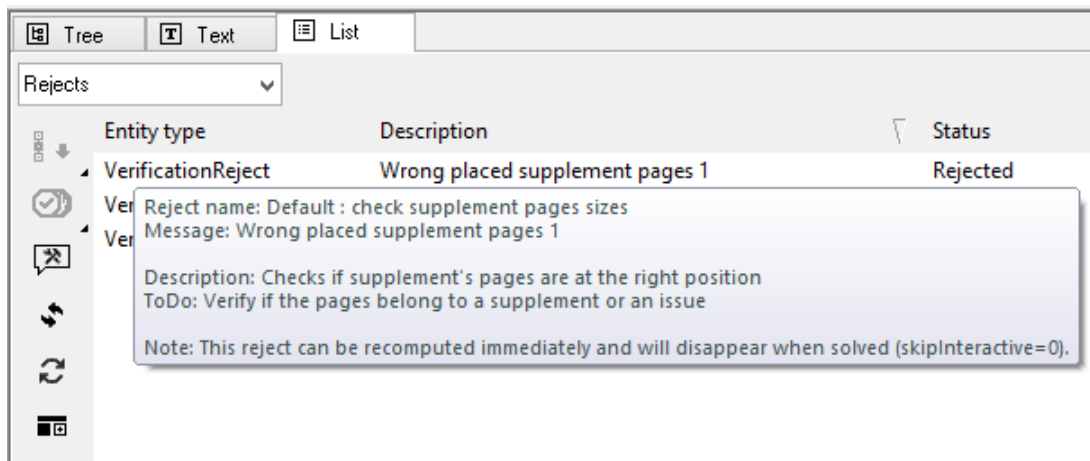
## 23.40 CheckStructureErrorsP

- Checks for structure errors in hierarchy
- Doesn't have parameters
- Carried out on pages
- By default, is VerificationReject
- Used in: Review zoning (disabled by default), Review page sequence (disabled by default), Review issues (disabled by default), Review structure and text
- Processing in UI: Delete the entity or fix the structural error.



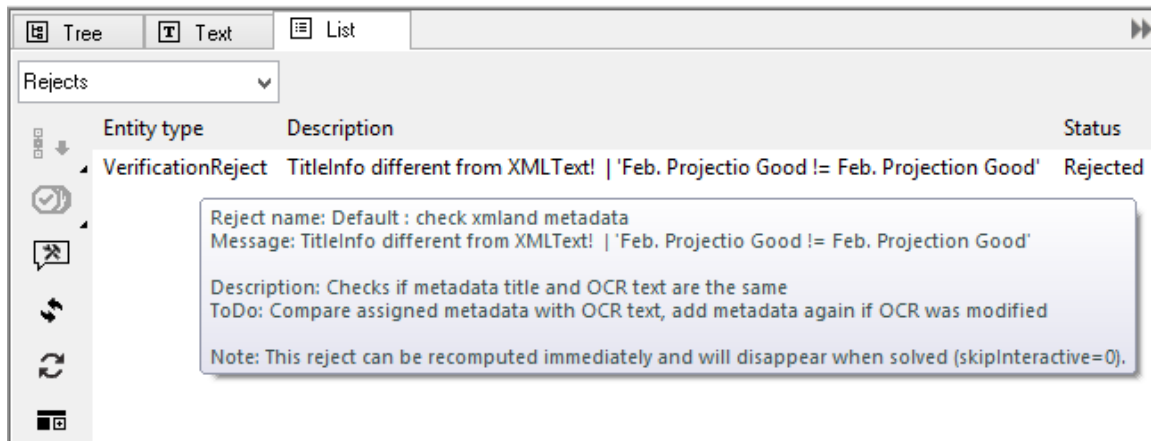
## 23.41 CheckSupplementPagesSizes

- Checks if supplement (alternative) pages are at the right position
- Has parameter for minimum percent of the average size page (minPercent)
- Carried out on document
- By default, it is a Verification reject
- Used in: Review structure and text (disabled by default)
- Processing in UI: Verify if the pages belong to a supplement or an issue. Check if pages smaller than a given percent of the average sizes are at the end of the document



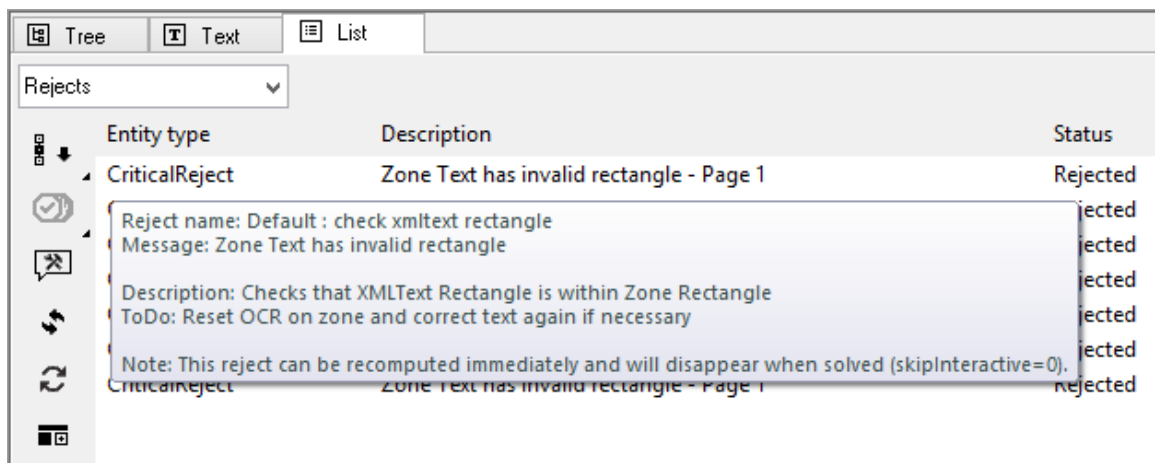
## 23.42 CheckXMLandMetadata

- Checks if metadata title and OCR text are the same
- Doesn't have parameters
- Carried out on document
- By default, it is a VerificationReject
- Used in: Review structure and text
- Processing in UI: Compare assigned metadata with OCR text, add metadata again if OCR was modified.



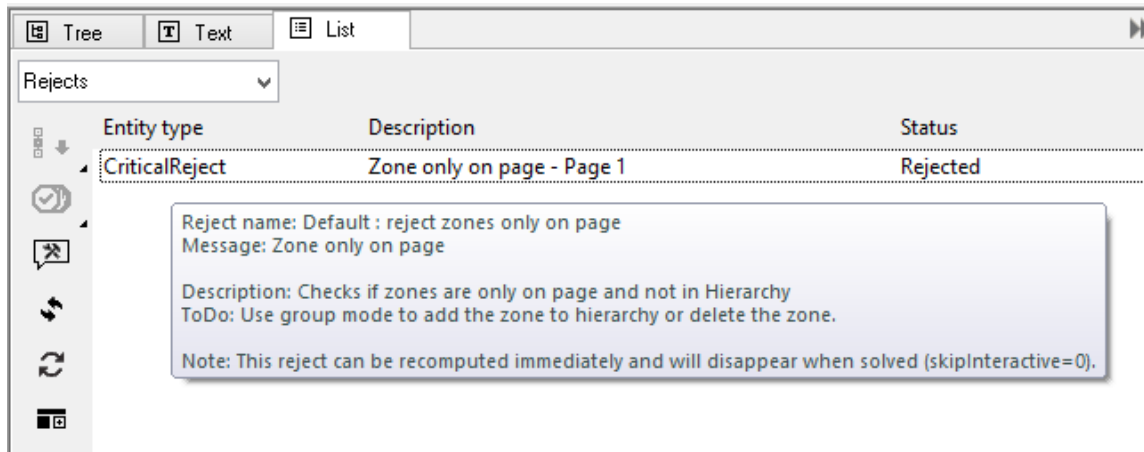
## 23.43 CheckXMLTextRectangle

- Checks that XMLText rectangle is within zone rectangle
- Has parameter for pixel tolerance (PixelTolerance)
- Carried out on pages
- By default, it is a CriticalReject
- Used in: Review structure and text (disabled by default)
- Processing in UI: Reset OCR on zone and correct text again if necessary.



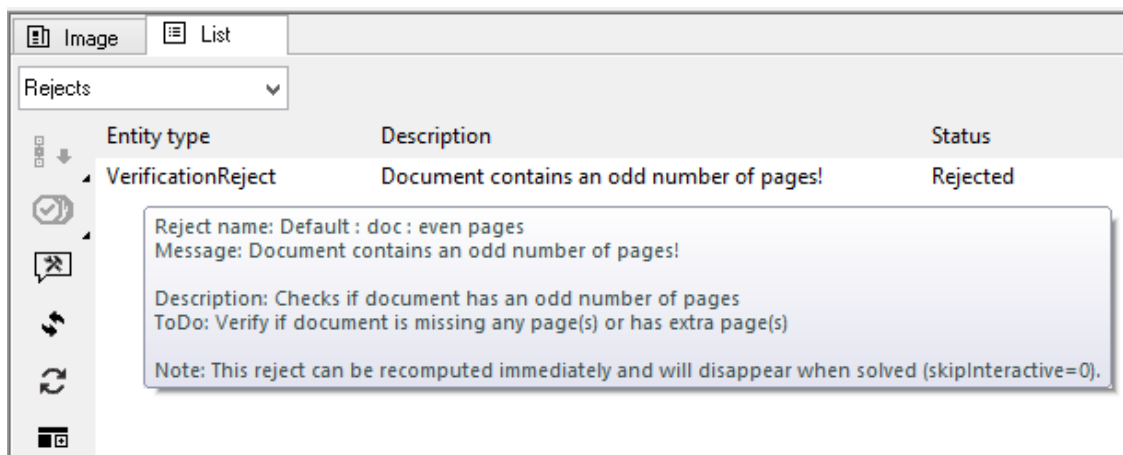
## 23.44CheckZonesOnlyOnPage

- Checks if zones are only on page and not in hierarchy
- Has parameter for allowing supplements to be part of a page (supplementPartOfPage)
- Carried out on document
- By default, it is a CriticalReject
- Used in: Review structure and text
- Processing in UI: Use group mode to add the zone to hierarchy or delete the zone.



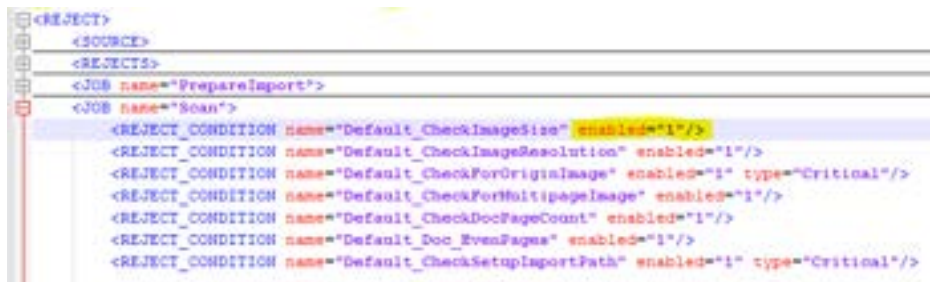
## 23.45Doc\_EvenPages

- Checks if the document has an odd number of pages
- Has parameter for skipping target pages (skipTarget)
- Carried out on document
- By default, it is a VerificationReject
- Used in: Review import
- Processing in UI: Verify if document is missing any page(s) or has extra page(s).



## 23.46 Enable / disable rejects

To enable a reject condition, inside the RejectByEnt.xml file, navigate to the desired reject, and set the parameter "enabled" to "1"; for disabling a reject, set "enabled" to "0".



The reject are displayed in docWizz interface in List view -> Rejects.

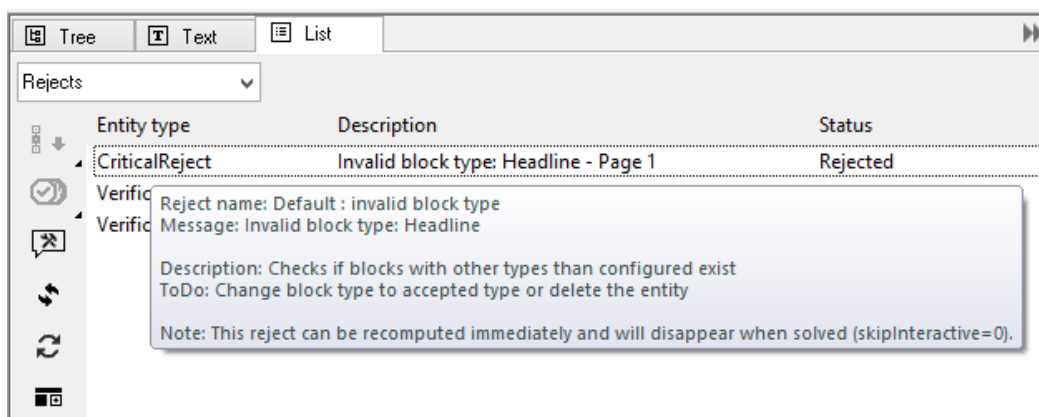


## 23.47 Explanation of rejects

Reject conditions can be activated and their parameters can be set on project basis inside the project configuration directory. To do so, a file "RejectByEnt.xml" must be assigned to the project by placing it in a subfolder (named "reject") of the according project configuration (project-cfg\projectname\reject). It can be copied from project-cfg\Default\reject and edited afterwards.

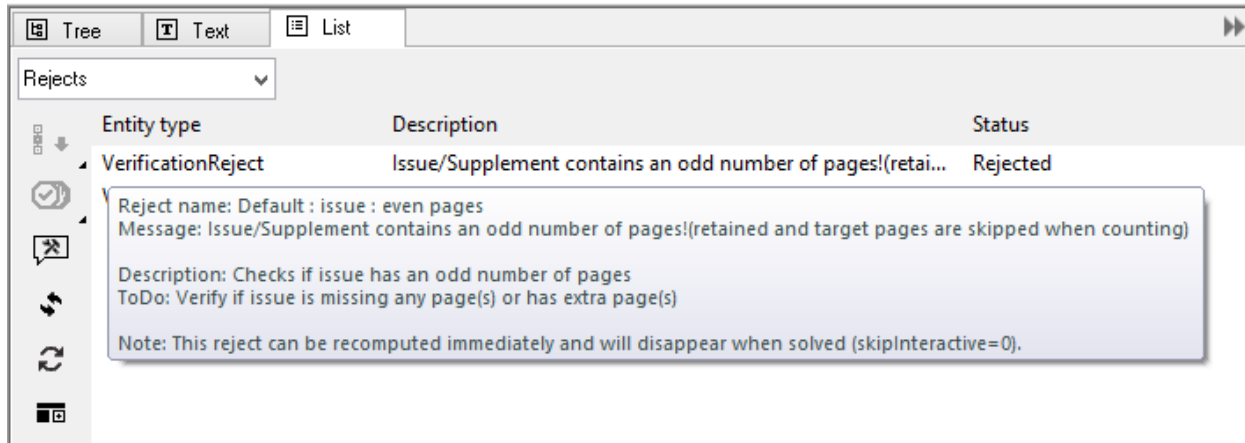
## 23.48 InvalidBlockType

- Is obsolete
- Checks if not configured types of zones appear in the document
- Has parameters for accepted zone types for newspapers and books
- Carried out on pages
- By default, it is a CriticalReject
- Used in: Review page sequence (disabled by default), Review structure and text (disabled by default)
- Processing in UI: Change block type to accepted type or delete the entity.



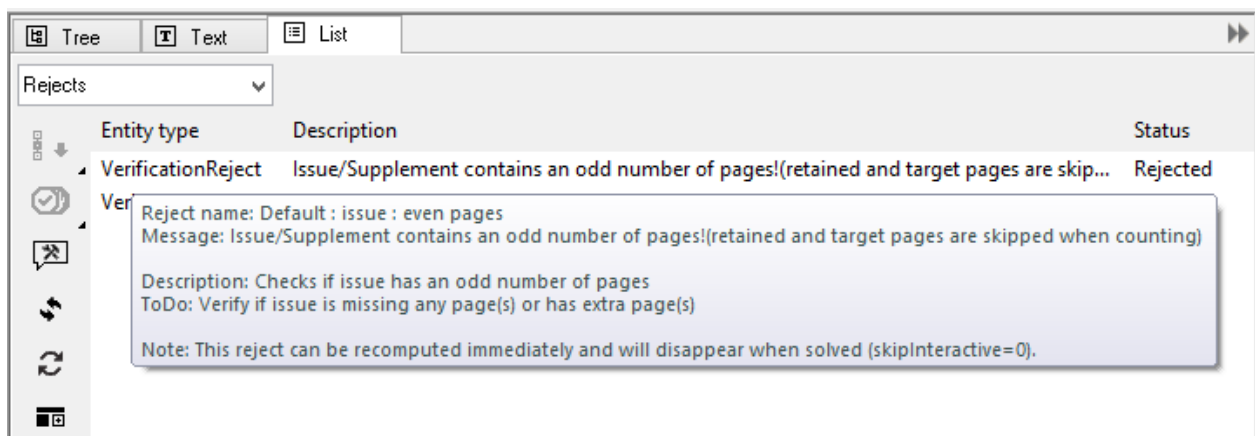
## 23.49 Issue\_ContinuousPages

- Checks for discontinuous pages in an issue
- Doesn't have parameters
- Carried out on hierarchy
- By default, it is a CriticalReject
- Used in: Review issues, Review structure and text
- Processing in UI: Verify if there are missing pages from the issue.



## 23.50 Issue\_EvenPages

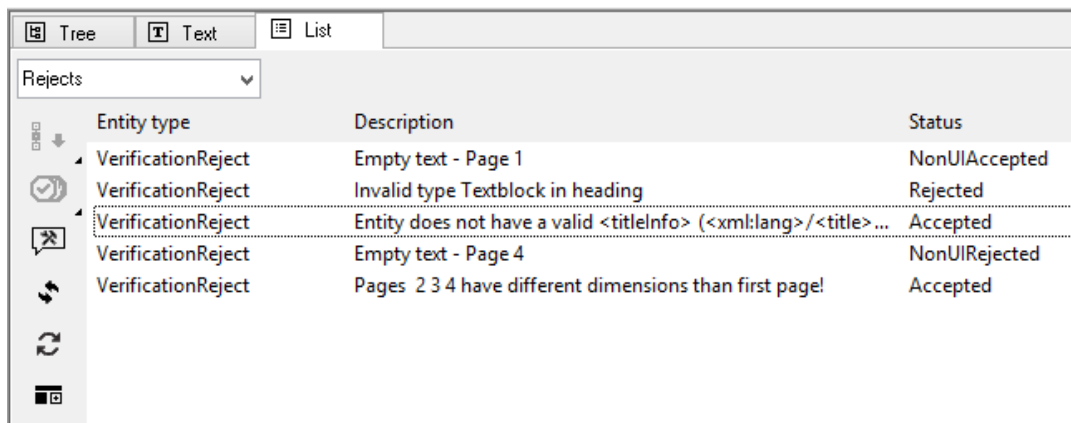
- Checks if issue has an odd number of pages
- Doesn't have parameters
- Carried out on hierarchy
- By default, it is a VerificationReject
- Used in: Review issues, Review structure and text
- Processing in UI: Verify if issue is missing any page(s) or has extra page(s).





## 23.51 Reject status

<b>Rejected</b>	is used for both types of rejects (Verification and Critical). As long as a reject has the status "Rejected" the document cannot be processed to the next step.
<b>Accepted</b>	is used only for VerificationReject. This status will appear when a reject condition is accepted by the user. With the status "Accepted", the reject will no longer be re-computed and the document can be processed to the next step. <i>The "accepted" rejects will still be displayed in the Rejects list even if the problem that generated the reject is fixed.</i>
<b>NonUIRejected</b>	is the same status as "Rejected", the difference being where the reject is computed. Rejects that have skipInteractive="1" in RejectByEnt.xml file are not computed in interface, usually because they are time consuming, instead they are computed in background services – this will keep the reject displayed in interface, even if the problem was fixed, until the document is re-processed. With this status for a reject, the document cannot be processed to the next step.
<b>NonUIAccepted</b>	is used when rejects with "NonUIRejected" status are accepted by the user. With this status for a reject, the document can be processed to the next step.



Entity type	Description	Status
VerificationReject	Empty text - Page 1	NonUIAccepted
VerificationReject	Invalid type Textblock in heading	Rejected
VerificationReject	Entity does not have a valid <titleInfo> (<xml:lang>/<title> ...	Accepted
VerificationReject	Empty text - Page 4	NonUIRejected
VerificationReject	Pages 2 3 4 have different dimensions than first page!	Accepted

## 23.52 Reject types

A reject can be set as a VerificationReject or as a CriticalReject.

### VerificationReject

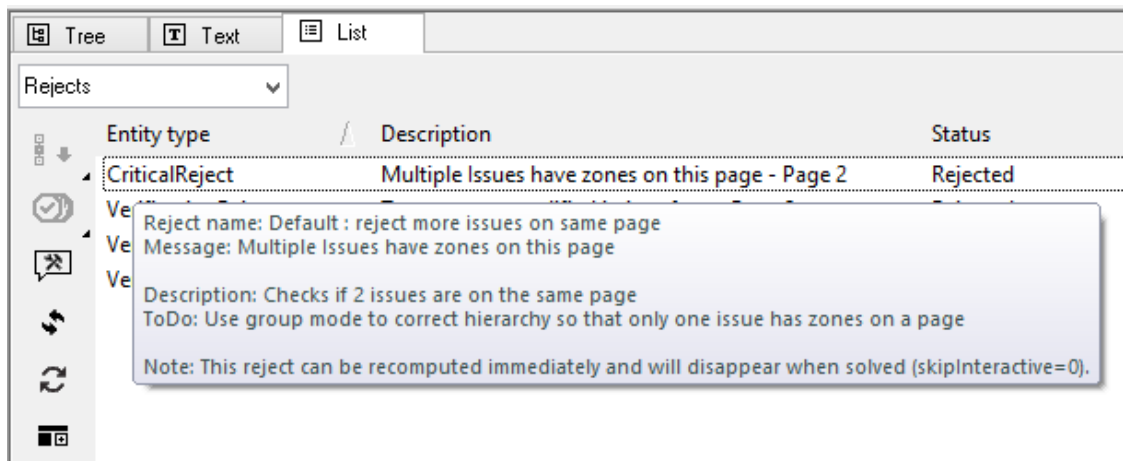
It is set as a warning, and the user has the possibility to accept the rejected condition or to fix the problem that generated the reject. In RejectByEnt.xml file, these rejects don't have the "type=Critical" attribute.

### CriticalReject

It is more of an error and the user cannot accept this type of reject. The only possibility to process the document further is to fix the problem that generated the reject. In RejectByEnt.xml file, these rejects have the "type=Critical" attribute.

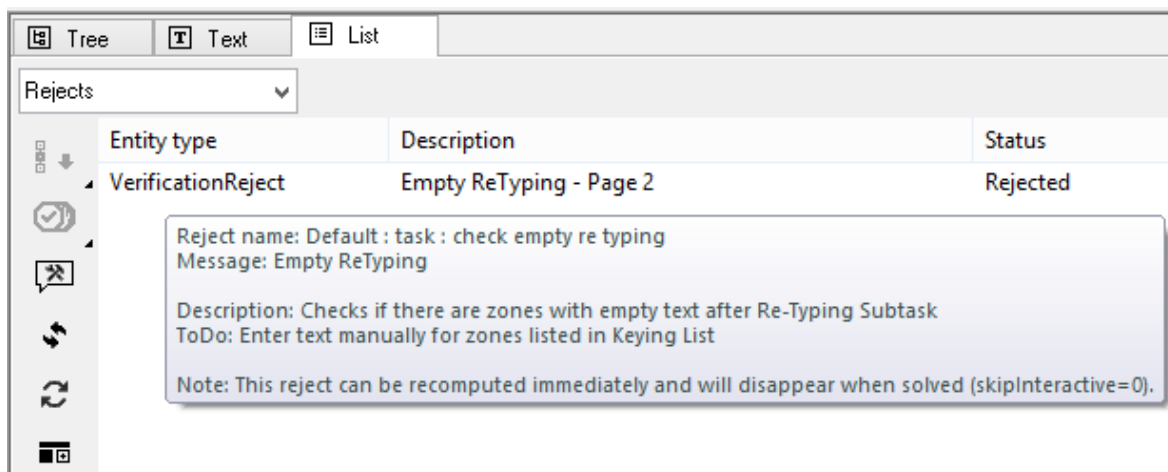
## 23.53 RejectMoreIssuesOnSamePage

- Checks if two issues are on the same page (have zones on the same page)
- Has parameter for allowing supplements to be part of a page (supplementPartOfPage)
- Carried out on document
- By default, it is a CriticalReject
- Used in: Review structure and text
- Processing in UI: Use group mode to correct hierarchy so that only one issue has zones on a page.



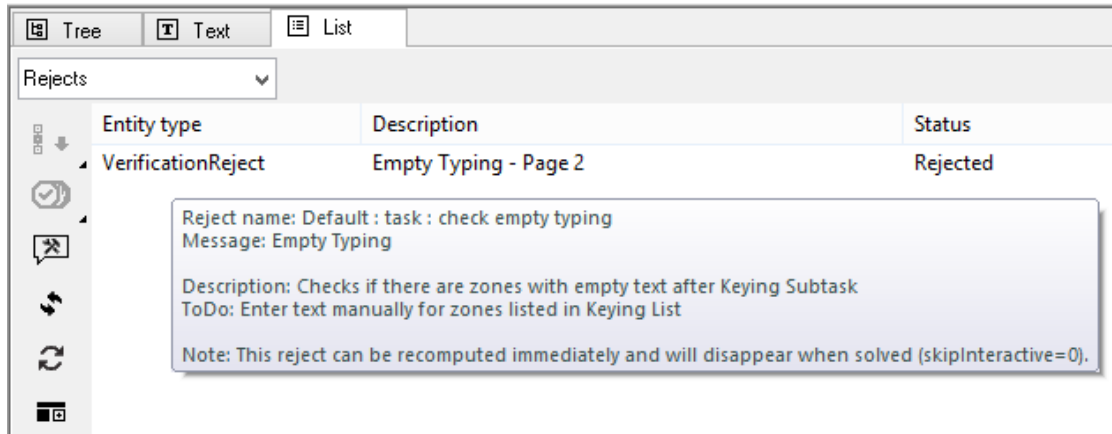
## 23.54 Task\_CheckEmptyReTyping

- Checks if there are zones with empty text after Retyping subtask
- Doesn't have parameters
- Carried out on pages
- By default, it is a VerificationReject
- Used in: Review structure and text – Retyping
- Processing in UI: Enter text manually for zones listed in Keying List



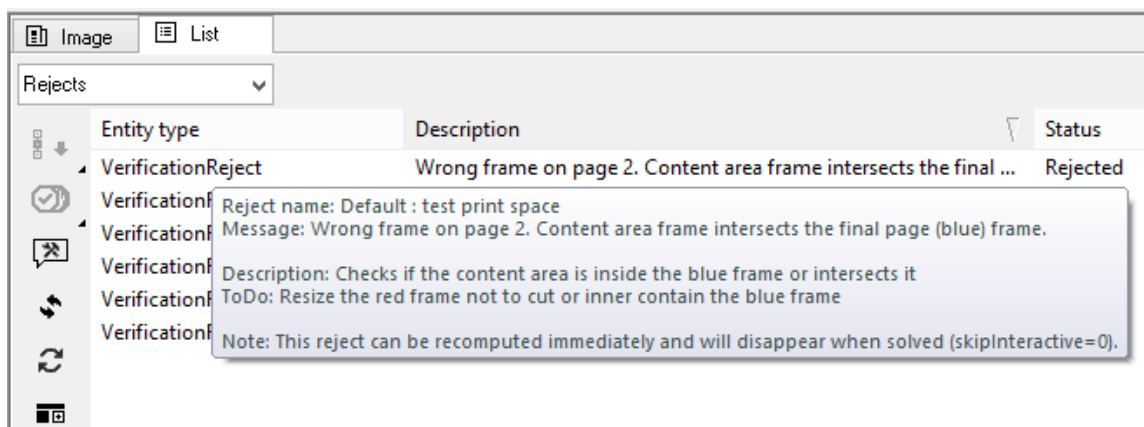
## 23.55Task\_CheckEmptyTyping

- Checks if there are zones with empty text after Keying subtask
- Doesn't have parameters
- Carried out on pages
- By default, it is a VerificationReject
- Used in: Review structure and text – Typing
- Processing in UI: Enter text manually for zones listed in Keying List.



## 23.56TestPrintSpace

- Contains also the CheckMissingFrames reject
- Checks if the Printspace (content area) is inside the final page frame or intersects it
- Has parameters for allowing frames on target pages, allowing double pages to have only one frame and allowing Start page and End page to have only one frame ("AllowFramesOnTarget", "AllowSingleFrameOnDoublePageDoc" and "AllowSingleFrameStartEndPage")
- Carried out on document pages
- By default, it is a VerificationReject
- Used in: Prepare cropping
- Processing in UI: Resize the red frame not to cut or contain the blue frame.



## 23.57 TestRedFrameOnPage

- Checks if the red frame is out of page by more than the configured percent
- Has parameter for minimum intersection percent
- Carried out on document pages
- By default, it is a VerificationReject
- Used in: Prepare cropping
- Processing in UI: Resize / move the red frame, it must not be outside page.



## 23.58 ValidateDocument

- Checks if document is invalid or corrupted
- Doesn't have parameters
- Carried out on document
- By default, it is a CriticalReject
- Used in: Review structure and text
- Processing in UI: Technical Support needed.



## 23.59 ValidateMetadata

- Checks if metadata structure is ok
- Doesn't have parameters
- Carried out on hierarchy
- By default, it is a VerificationReject
- Used in: Review structure and text
- Processing in UI: Verify and correct MODS metadata.



## 24 RejectByEnt.xml

**RejectByEnt.xml structure** (..\project-cfg\Default\Reject\)

```
<?xml version="1.0" encoding="UTF-8" ?>
- <REJECT>
- <SOURCE>
  <SRC>***DATA***\Script\Reject\RejectByEnt_cond.tcl</SRC>
  <SRC>***DATA***\Script\Reject\RejectByEnt_condX.tcl</SRC>
  <SRC>***DATA***\Script\Reject\RejectByEnt_condOCR.tcl</SRC>
  <SRC>***DATA***\Script\Reject\RejectByEnt_condOCRTextvsDict.tcl</SRC>
  <SRC>***DATA***\Script\Reject\RejectByEnt_task_cond.tcl</SRC>
</SOURCE>
- <REJECTS>
- <!-- place for optimal calcs values empty=all,
D=doc,H=Hiereachy,P=pages,DP=DocPages
-->
- <!-- initreject cfg a procedure that is run at the begining of the reject
alg to init the entities props -->
- <!-- posible doctype values: Monograph Newspaper Multivolume_Monograph
Serial doctype -->
- <!-- Run on Document Pages
-->
- <RULE name="Default_CheckImageResolution" rejID="1" applyTo="1" place="DP">
  <initreject>Reject::InitPagesStatusesResolution</initreject>
  <entprint>Reject::GetDocPagePrintResolution</entprint>
  <reject>Reject::CheckImageResolution</reject>
  <param name="resolution">300 400 600</param>
  <descr name="pageSrc">Page %d</descr>
  <descr name="ret">bad resolution (%s dpi)</descr>
  <help>Test for image resolution</help>
  <help_qa>Rescan/Delete the image, Or accept the page</help_qa>
</RULE>
- <RULE name="Default_CheckImageSize" rejID="1" applyTo="1" place="DP">
  <initreject>Reject::InitPagesStatusesSize</initreject>
  <entprint>Reject::GetDocPagePrint</entprint>
  <reject>Reject::CheckImageSize</reject>
  <param name="minWidth">500</param>
  <param name="maxWidth">8400</param>
  <param name="minHeight">500</param>
  <param name="maxHeight">8400</param>
  <descr name="pageSrc">Page %d</descr>
  <descr name="ret">has invalid %s</descr>
```

```

    <help>Test for image size</help>
    <help_qa>Rescan/Delete the image, Or accept the page</help_qa>
  </RULE>
- <RULE name="Default_CheckImageMemorySize" rejID="1" applyTo="1" place="DP"
type="Critical">
  <initreject>Reject::InitPagesStatusesSize</initreject>
  <entprint>Reject::GetDocPagePrint</entprint>
  <reject>Reject::CheckImageMemorySize</reject>
  <param name="maxSizeMB">600</param>
  <descr name="pageSrc">Page %d</descr>
  <descr name="ret">invalid %s</descr>
  <help>Test for image size</help>
  <help_qa>Choose lower resolution or different format
(gray/bitonal)</help_qa>
</RULE>
- <RULE name="Default_CheckImagePrintSpace" rejID="1" applyTo="1" place="DP">
  <entprint>Reject::GetDocPagePrint</entprint>
  <reject>Reject::CheckImagePrintSpace</reject>
  <descr name="pageSrc">Page %d</descr>
  <descr name="ret">has invalid %s</descr>
  <help>Test for image printspace</help>
</RULE>
- <RULE name="Default_CheckForOriginImage" rejID="1" applyTo="1" place="DP"
type="Critical">
  <initreject />
  <entprint />
  <reject>Reject::CheckForOriginImage</reject>
  <descr name="pageSrc">Origin for page %d does not exist.</descr>
  <descr name="ret">%s</descr>
  <help>Test if origin image exists</help>
  <help_qa>Rescan the image, Or accept the page</help_qa>
</RULE>
- <RULE name="Default_CheckForMultipageImage" rejID="1" applyTo="1"
place="DP">
  <initreject />
  <entprint />
  <reject>Reject::CheckForMultipageImage</reject>
  <descr name="pageSrc">Origin for Page %d</descr>
  <descr name="ret">has multipage-tif as origin. %s</descr>
  <help>Test if origin image is multi-tif</help>
  <help_qa>Rescan the image, or accept the page</help_qa>
</RULE>
- <RULE name="Default_CheckForInvalidPages" rejID="1" applyTo="1" place="DP">

```

```

<initreject />
<entprint />
<reject>Reject::CheckForInvalidPages</reject>
<descr name="pageSrc">Page %d</descr>
<descr name="retRescan">marked for rescan! Status: %d</descr>
<descr name="retError">marked as having errors! Status: %d</descr>
<help>Test statuses of the page</help>
<help_qa>You should have selected Re-Scan as next job!</help_qa>
</RULE>
- <RULE name="Default_CheckFrameSizes" rejID="2" applyTo="1" place="DP">
  <initreject />
  <entprint />
  <reject>Reject::CheckFrameSizes</reject>
- <!-- Set if TargetPages should have frame
  -->
  <param name="AllowFrameOnTarget">0</param>
- <!-- old param "maxFrameTolerance" still working, in case
  "maxFrameToleranceMM" is available, this will be taken!
  -->
- <!-- param name="maxFrameTolerance">200</param
  -->
  <param name="maxFrameToleranceMM">20.0</param>
  <descr name="ret">Invalid frame sizes on page %d.</descr>
  <descr name="retH">Height difference in mm %.2f</descr>
  <descr name="retW">Width difference in mm %.2f</descr>
  <help>Reject if frame size differs by more than a tolerance from the
  printspace size</help>
  <help_qa>Resize the frame or make it individual</help_qa>
</RULE>
- <RULE name="Default_CheckMissingFrames" rejID="2" applyTo="1" place="DP">
  <initreject />
  <entprint />
  <reject>Reject::CheckMissingFrames</reject>
- <!-- Set if TargetPages should have frame
  -->
  <param name="AllowFrameOnTarget">0</param>
  <param name="AllowSingleFrameOnDoublePageDoc">1</param>
  <param name="AllowSingleFrameStartEndPage">1</param>
  <descr name="ret">Page %d has missing frame.</descr>
  <descr name="retMore">Too many frames on page %d</descr>
  <help>Test for pages without frames</help>
  <help_qa>Drag a new frame</help_qa>

```

```

</RULE>
- <RULE name="Default_TestPrintSpace" rejID="2" applyTo="1" place="DP">
- <!-- contain also the CheckMissingFrames reject
-->
<initreject />
<entprint />
<reject>Reject::TestPrintSpace</reject>
- <!-- Set if TargetPages should have frame
-->
<param name="AllowFrameOnTarget">0</param>
<param name="AllowSingleFrameOnDoublePageDoc">1</param>
<param name="AllowSingleFrameStartEndPage">1</param>
<descr name="retMss">Missing frame on page %d</descr>
<descr name="retBad">Wrong frame on page %d. Print space intersects the
final page (blue) frame.</descr>
<descr name="retMore">More than 2 frames on page %d</descr>
<descr name="retMoreSingle">More than 1 frames on page %d</descr>
<help>Reject if print space is in the blue frame or intersect it</help>
<help_qa>Resize the red frame not to cut/contain the blue frame</help_qa>
</RULE>
- <RULE name="Default_TestRedFrameOnPage" rejID="3" applyTo="1" place="DP">
<initreject />
<entprint />
<reject>Reject::TestRedFrameOnPage</reject>
<param name="MinIntersectionPercent">95</param>
<descr name="retOutPercent">Too large red frame area out of page %d</descr>
<help>Reject if cut frame is out of page</help>
<help_qa>Resize the red frame: must not be outside page</help_qa>
</RULE>
- <!-- Run on pages
-->
- <RULE name="Default_InvalidBlockType" rejID="4"
applyTo="IsKindOf($ent,'LayoutElement')" place="P" type="Critical">
- <!-- obsolete please configure the file IgnoreTypes.dsl
-->
<entprint>Reject::GetLayoutElementEntPrint</entprint>
<reject>Reject::InvalidBlockType</reject>
<param name="InvalidBlockType">Textblock Headline Illustration Table
Caption RunningTitle Advertisement Publishing_Stmt FamilyNotice PageNumber
ContinuationLink ContinuationHeadline Author Overline Subheadline
Graphical_Illustration Graphical_Table Graphical_Text</param>
<param name="InvalidBlockTypeBook">Textblock Headline Illustration Table
Caption RunningTitle Advertisement Publishing_Stmt FamilyNotice PageNumber
ContinuationLink ContinuationHeadline Author Overline Subheadline Footnote

```



```

ItemCaption Marginalia Formula CatchWord Graphical_Illustration
Graphical_Table Graphical_Text</param>

  <descr name="ret">Invalid block type: %s</descr>
  <help>Determine if blocks with other types than in param exist</help>
  <help_qa>Change type or delete the entities</help_qa>
</RULE>

- <RULE name="Default_CheckBlockRectangle" rejID="4"
applyTo="IsKindOf($ent, 'LayoutElement') " place="P">
  <entprint />
  <reject>Reject::CheckBlockRectangle</reject>
  <param name="bCheckForPolygons">0</param>
  <descr name="ret">Die XML-Seite kann nicht angezeigt werden
Die XML-Eingabe kann nicht angezeigt werden, wenn Stylesheet XSL verwendet
wird. Beheben Sie den Fehler und klicken Sie dann auf Aktualisieren, oder
wiederholen Sie den Vorgang später.

Ein Name beginnt mit einem ungültigen Zeichen. Fehler beim Bearbeiten der
Ressource 'file:///C:/Dokumente und Einstellungen...

  <RULE name="Default_CheckForMissingOCROnPage" rejID="8"
applyTo="IsKindOf($ent, 'imagePage') " cleanCondition="IsKindOf($...
SPAN class="m">">Block %s has invalid shape</descr>
  <descr name="NotAllowed">Polygonal zones are not allowed</descr>
  <descr name="MissPageIndex">PageIndex is missing</descr>

```

## 24.1 Critical Condition

When [type="Critical"] flag is used, users have to correct the error before they can process the document.

```
<JOB name="Scan">
  <REJECT_CONDITION name="Default_CheckImageSize" enabled="1"/>
  <REJECT_CONDITION name="Default_CheckImageResolution" enabled="1"/>
  <REJECT_CONDITION name="Default_CheckForOriginImage" enabled="1"
type="Critical"/>
  <REJECT_CONDITION name="Default_CheckForMultipageImage"
enabled="1"/>
  <REJECT_CONDITION name="Default_CheckDocPageCount" enabled="1"/>
  <REJECT_CONDITION name="Default_Doc_EvenPages" enabled="1"/>
  <!-- Scan: add custom reject CONDITIONS here -->
</JOB>
```

### param name

This condition checks if there are invalid block zones in the document. It reads the current zones of the document and compares them with the allowed block types.

### descr name

It gives "Invalid block type" as reject status within the interface.

### help

There is a short description about the purpose of the condition in the script.

### help\_qa

There is a hint as how to correct the reason of the condition.

## 24.2 Timing Investigation

An EnableDEBUG parameter inside the \*\*\*DATA\*\*\*\script\reject\RejectByEnt\_main.tcl file was added. It's default value is set to 0. Once you enable it, an docID\_rejects.txt file will be created for each document everytime you are computing the rejects list. File will contain informations related to the rejects that were computed, the starting time of each one and duration. The file will be created inside \*\*\*DATA\*\*\*\work\RejectsDebug directory and it will be replaced everytime the rejects are recomputed for one document.

Dump file is remodeled to be a \*.csv file, with timing values on each reject (old dump file generated dump for each entity's rejects, big dump file, hard to be read, now it is totalized per levels/ rejects/ total number of rejects computed).

## 24.3 SkipInteractive

Example:

```
<AUTO_FIX name="Default_ComputeMissingOCR" enabled="1" skipInteractive="1">
<REJECT_CONDITION      name="Default_CheckForMissingOCROnPage"      enabled="1"
skipInteractive="1"/>
```

Whenever "skipInteractive" is set to "1", time consuming jobs are transferred to a service.

- If in Client workmode "Step by Step" is selected, those autofixes or rejects are executed anyway.
- If for a reject "SkipInteractive" is set to "1", it is not executed when clicking "OnProcess" and batch processing is selected. On step-by-step, it is computed.
- On reject list open, subset change or refresh button, rejects are handled as follows:

A new setting is defined in project-setting.xml: "ComputeTimeConsumingRejectsInUI". If this is enabled, the reject is computed. If this is disabled, the reject is not computed at that time. Default for already existing configurations is disabled.

- A progress dialog is shown when computing rejects. It has a "Delay" button. If "ComputeTimeConsumingRejectsInUI" is enabled and the user presses this button, any skipable reject computing is skipped immediately (if an OCR call is currently executed, the single call is completed, but no other is made).
- All rejects are executed

## 24.4 Reject per Editing Task

When rejects for tasks are enabled only the rejects for that tasks will be computed. Also at the end of the step the rejects for tasks will not be computed. QA user will be able to verify his work.

If no rejects for tasks are defined the program act like before.

To configure in RejectByEnt.xml you need to add:

```
<JOB name="VerifyHierarchy">
....rejects per step like now....
  <TASK name="Typing">
    <AUTO_FIX name="VH_SetKeyingCorrectionList"
    enabled="1">***DATA***\script\docWizz_QA.tcl</AUTO_FIX>
    <REJECT_CONDITION name="Default_Task_CheckEmptyTyping" enabled="1"/>
  </TASK>
  <TASK name="ReTyping">
....rejects per task like now....
</TASK>
</JOB>
```

And in Default-QA-Tasks.xml: the line

```
<SCRIPT_COMPLETE>source                                     [expandpath
{***DATA***\script\reject\RejectByEnt_main.tcl}}           ;           return
[IsTaskComplete Typing]</SCRIPT_COMPLETE>
```

like here:

```
<TASK NAME="Typing">
<OUTLINE VIEW="Text" SELECTION="Keying"/>
<DETAIL VIEW="Image"/>
<USERGROUP NAME="KEYING"/>
<DIFFERENT_USER>0</DIFFERENT_USER>
<SCRIPT_SELECT></SCRIPT_SELECT>
<SCRIPT_COMPLETE>source                                     [expandpath
{***DATA***\script\reject\RejectByEnt_main.tcl}}           ;           return
[IsTaskComplete Typing]</SCRIPT_COMPLETE>
<PREVTASK>VerifyOCR</PREVTASK>
<DIFFERENT_USER>1</DIFFERENT_USER>
</TASK>
~
```

## 25 Manual QA: The QA Work Step

An additional QA step has been implemented that allows administrators, team leaders/project leaders to do a final QA on documents. In this scenario, we distinguish between normal users and QA users. Normal users do correction on the documents as usual. QA users perform a final check on the whole document before it is exported. In this chapter you will learn how to work with the QA mode. If you want to learn how to define a QA user please refer to the chapter "Defining the QA mode" of the docWizz administration manual.

Please be aware of the renamed steps and tasks and the "old" names in scripts and configuration files.

### 25.1 Working with the QA Mode

If the QA mechanisms is activated, processing workflow is as follows:

- Normal operators do correction as usual.
- After hitting the process button in VerifyHierarchy documents are not exported immediately.
- Instead, their status is automatically set to "QA".
- QA users must be logged in as admin.
- The QA user can check those documents through the Document Pool (filter by status "QA")

✔ Process button behaves like in previous releases unless „Final QA“ is defined.

Note: Documents are set to the status "Reject" either by the QA Operator in manual QA or if a documents is rejected through automated QA in server processing. The second is the case if certain reject conditions apply. To learn about the use of automated QA and the use of reject conditions please refer to the chapter **Automated QA: Reject Conditions**.

### 25.2 Setting up the QA Mode

docWizz is able to deal with final QA mechanism as described above. Therefore you need define which users (operator group) shall be QA users and you need to define which processing jobs shall be controlled by the additional QA step.

For defining and authorizing users to perform the QA step, you need to call system configuration in the docWizz main menu and select the user tab. There you can assign the final QA right to any of the defined users or you may add a new user.

This is how to set up the final QA mechanism per project and job. You need to create an xml file called projectname-qa.xml. Place this file in the directory project-cfg\projectname of the according project. It shall contain a list of jobs in which final QA shall be performed:

```
<?xml version="1.0" encoding="UTF-8"?>
  <QA>
    <STEP name="VerifyPagesHierarchy"/>
    <STEP name="VerifyHierarchy"/>
  </QA>
```

In this sample, final QA will be performed in the jobs VerifyPagesHierarchy and VerifyHierarchy.

### 25.3 Report result of Double Keying

Results of Keying can be seen in BatchResults table. Reports time existing for Editing tasks. This information could be useful to track performance of operators.

The screenshot shows the 'Document Processing History' window. It contains a table with the following columns: Date, Duration, Job, Result, User, C., O., S., R., P., and Actions. The table lists 15 processing tasks, including jobs like 'VerifyHeaderArch', 'VerifyHeaderArchTyping', 'VerifyHeaderArchCopy', 'VerifyHeaderArchVerifyOCR', 'Exported - VerifyHeaderArch', 'Export04', and 'Export04L'. The results are mostly 'Task completed' or 'Exported', with some 'Rejected' or 'Edited' entries. The actions column shows various codes like 'H42', 'K149, H4', 'K315, M38', 'H4', 'H4', 'H4', 'H4', 'H4', 'H4', 'H4', 'H4', 'H4', 'H4', 'H4', 'H4', 'H4'.

Date	Duration	Job	Result	User	C.	O.	S.	R.	P.	Actions
18.01.2012 10:46:02	00:00:20	VerifyHeaderArch/CompareKeying	Task compl...	d...	S...	Vlad	W...	L...	3	H42
18.01.2012 10:48:37	00:00:24	VerifyHeaderArch/Typing	Task compl...	d...	S...	Vlad	W...	L...	3	K149, H4
18.01.2012 10:50:32	00:01:16	VerifyHeaderArch/Copy	Task compl...	d...	S...	Vlad	W...	L...	3	K315, M38
18.01.2012 10:50:32	00:00:10	VerifyHeaderArch/VerifyOCR	Task compl...	d...	S...	Vlad	W...	L...	3	H4
18.01.2012 10:50:05	00:00:17	VerifyHeaderArch/VerifyOCR	Task compl...	d...	S...	Vlad	W...	L...	3	H4
18.01.2012 10:24:04	00:00:23	Exported - VerifyHeaderArch	Rejected	d...	S...	Vlad	W...	L...	3	H42
18.01.2012 10:24:04	00:00:00	Exported - VerifyHeaderArch	Edited and ...	d...	S...	Vlad	W...	L...	3	
18.01.2012 10:24:03	00:00:00	Export04L	Export (ite...	d...	S...	Vlad	W...	L...	3	
18.01.2012 10:23:22	00:00:41	Export04L	Edited and ...	d...	S...	Vlad	W...	L...	3	
18.01.2012 10:23:22	00:00:00	Export	OK (exp-b...	d...	S...	Vlad	W...	L...	3	
18.01.2012 10:23:17	00:00:04	Export	Edited and ...	d...	S...	Vlad	W...	L...	3	
18.01.2012 10:23:15	00:00:02	VerifyHeaderArch	Edited and ...	d...	S...	Vlad	W...	L...	3	H4
18.01.2012 10:22:52	00:00:23	VerifyHeaderArch	Edited and ...	d...	S...	Vlad	W...	L...	3	H4
18.01.2012 10:21:47	00:00:22	Exported - VerifyHeaderArch	Rejected	d...	S...	Vlad	W...	L...	3	H4
18.01.2012 10:21:47	00:00:00	Exported - VerifyHeaderArch	Edited and ...	d...	S...	Vlad	W...	L...	3	
18.01.2012 10:21:46	00:00:00	Export04L	Export (ite...	d...	S...	Vlad	W...	L...	3	
18.01.2012 10:21:01	00:00:45	Export04L	Edited and ...	d...	S...	Vlad	W...	L...	3	
18.01.2012 10:21:00	00:00:00	Export	OK (exp-b...	d...	S...	Vlad	W...	L...	3	
18.01.2012 10:20:35	00:00:05	Export	Edited and ...	d...	S...	Vlad	W...	L...	3	

At the bottom of the window, there are two buttons: 'Flow Log' and 'Close'.

## 25.4 Document Pool

To manage any document in process start the **Document Pool** in the **Document** menu.

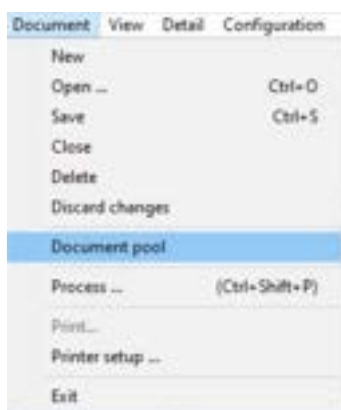
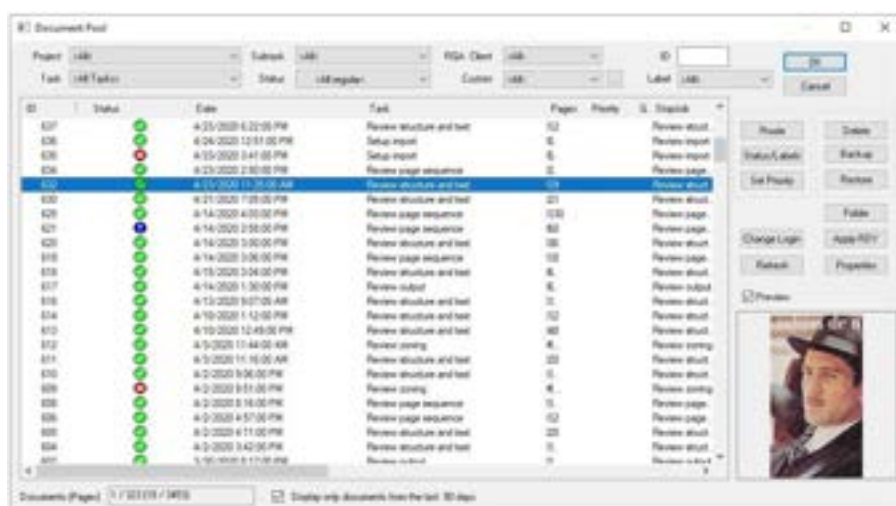


Figure 1: Document Menu: Document Pool

The document pool shows intermediate results of documents in any job:



- In order to give a better overview operators can apply **filters** to show documents in the pool. Please use the drop down menu "Project" to filter by project. Please use the drop down menu "Job" to filter by job. Please use the drop down menu "Status" to show only documents that have a certain QA status. One, two, three or none of the filters can be applied.
- Further more, operators can browse for documents within the document pool by typing in the **document ID**.
- The interface disposes of a display showing **number of selected documents** as well as **total number of documents** within the document pool.
- A button **Change Status** has been placed on the right hand side of the pool.
- Sort entries by the arrow on document's list header.
- Operators/Administrators can also enter a reason or comment.

Each document is listed along with its unique **ID**, next **Job**, **Date** of last modification, **Type** (serial or monograph or newspaper) and **Title** of the document. A lock icon indicates a document currently in use.

Whenever a job has been sent to the processing queue, the next job is an automatic process. All these are starting with **Detect...** (exception: SplitDbPages) or **Build...**, so the operator can identify prepared jobs easily.

If an entry starts with **Verify...** or just **Scan** the related document is apparently not prepared for batch processing but waiting for an operator to be opened up.

## Change Status

After selecting one or more documents in the Document Pool the status can be changed to a different status by hitting the **Status/Labels** button.

The dialog box titled "Changes Status and/or Labels for selected documents" contains the following fields and controls:

- Status:** A dropdown menu.
- Label:** A dropdown menu with an **Edit Labels** button next to it.
- Priority:** A text input field.
- Enter your comment here:** A large text area with an **Add** button to its right.
- Comments history:** A table with columns: Comment, Comment Date, Visible, and User.
- Buttons:** **OK**, **Cancel**, **Copy**, **Remove**, and a **Show all** checkbox.

You can sort the documents also by status:

This screenshot shows the same dialog box as above, but with the **Status** dropdown menu open. The menu lists the following options, each with a corresponding icon:

- Work (green circle with white checkmark)
- Error (red circle with white X)
- QA (green circle with white checkmark)
- Lead (blue circle with white checkmark)
- Reject (red circle with white X)
- Discuss (blue circle with white checkmark)
- Review (blue circle with white checkmark)
- Free Pool Data (red circle with white X)
- Send to Remote QA (purple circle with white checkmark)
- Reassign to RQA including images (purple circle with white checkmark)
- Delete from Remote QA (red circle with white X)
- Call document back (red circle with white X)



## 26 Editing/QA Tasks

The more specialized input and correction is necessary inside dedicated jobs within docWizz, the more difficult is it to handle all these tasks without forgetting something. As well there is no control about the isolated tasks. Also different tasks might be performed by different operators.

For that reason, tasks are configurable and applied to the documents. Per default, they are disabled, as they are meant to be enabled project based.

In the following chapters we describe the default editing tasks.

Additional editing tasks can be created – please contact dW support for custom editing tasks.

Please be aware of the renamed steps and tasks and the "old" names in scripts and configuration files.

Taskname	Job	Description
VerifyStructure	S-RST	Allows structural editing, mainly in tree view
ComputeMissingOCR	S-RST	This task has no manual interaction. Use the Process button to execute it.
Typing	S-RST	First typing of text to be compared with OCR.
ReTyping	S-RST	Second typing of text. Necessary for zones that do not provide OCR only.
CompareTyping	S-RST	Evaluation and correction of mismatches of OCR and typing.
VerifyMetadata	S-RST	Edit/QA on Metadata assigned to structural elements like articles
VerifyClipping	S-RST	View/Edit Clipping.
FinalQA	S-RST,C-RC,Z-RPS	Any kind of final verification by a different operator

"SCRIPT\_FILE=" can be configured per task. This makes configuration easy and side-effects are minimized.

Note: In some older configurations, script files to be used for any task were listed in section "SOURCES". This could cause troubles and was not efficient. This is still working, but should be changed.

When a non-admin user opens pool dlg:

- If document's project is empty it is not allowed to open documents with background jobs (see image BackgroundQATask.png)
- If document's project is selected, documents with pending background jobs are not visible (user cannot open them)
- It is disallowed for non-admin users to open documents if they are pending for non-gui QA Tasks (e.g. ComputeMissingOCR)

- If project is selected documents with non-gui QA Tasks are filtered out from pool selection.

The screenshot shows a Notepad++ window with the file path `H:\dwcfgs\generic\project-cfg\Default\Default-QA-Tasks.xml` in the title bar. The XML content is as follows:

```

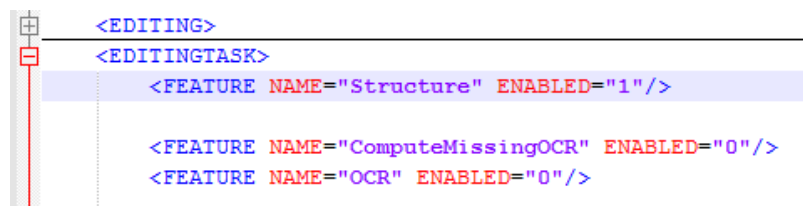
1 <TASK_DEFINITION>
2   <DOCUMENTTYPE NAME="Monograph">
3     <JOB NAME="VerifyHierarchy">
4       <TASK NAME="VerifyStructure">
11        <TASK NAME="ComputeMissingOCR">
12          <OUTLINE VIEW="List" SELECTION="" />
13          <DETAIL VIEW="Image" />
14          <USERGROUP NAME="AUTO_PROCESSING" />
15          <DIFFERENT_USER>1</DIFFERENT_USER>
16          <BACKGROUND>1</BACKGROUND>
17          <PREVTASK>VerifyStructure</PREVTASK>
18        </TASK>

```

## 26.1 QA Tasks Configuration

Editing tasks can be enabled / disabled from the <project-name>-Setting.xml file, found in the <project-name> folder.

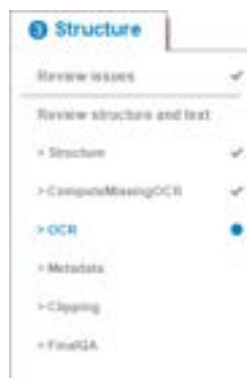
Here, navigate to the desired editing task inside <EDITINGTASK> node and change the value for "ENABLED" to "1".



To have the editing tasks available in docWizz, you need to define them in the <project-name>-QA-Tasks.xml file. In this file, advanced settings can be made: how the document is opened into a certain task, if the task is executed in background (dwServices), if scripts need to be executed.

Please contact dW support for changing advanced settings changes.

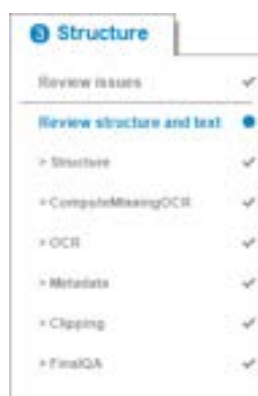
Enabled editing tasks are displayed in the workflow drop-down list, under the task:



Completed editing tasks are marked with a check after them, current editing task has a blue dot (just like normal tasks).

Processing forward / back a document between editing tasks is done the same as for normal tasks: click on the next task or the "Process" button to send the document to the next editing task, or click on a previous editing task to route back the document.

When all the editing tasks enabled for a project are completed, they will all have the check mark, and the current task will be the main task:



The order of the editing tasks cannot be changed, however, not all of them need to be enabled. A particular case is represented by the Keying editing tasks (Typing, ReTyping and CompareKeying) – see "Keying" section below.

The editing tasks are project-based, so they only apply to the project that they are enabled for.

### 26.1.1 Default-Setting.xml

For project specific configuration copy the default file to <Projectname>-Setting.xml

There is a section called <EDITINGTASK>. Here you enable (1) or disable (0) a task without creating a new configuration. Required previous tasks will be skipped automatically if not enabled. If a task is not listed it is treated as disabled. Changes take effect on existing documents when loading.

Example:

```
-<EDITINGTASK>
<FEATURE ENABLED="1" NAME="Structure"/>
<FEATURE ENABLED="0" NAME="ComputeMissingOCR"/>
<FEATURE ENABLED="0" NAME="OCR"/>
<FEATURE ENABLED="0" NAME="OCRKeying"/>
<FEATURE ENABLED="0" NAME="Typing"/>
<FEATURE ENABLED="0" NAME="ReTyping"/>
<FEATURE ENABLED="0" NAME="CompareKeying"/>
<FEATURE ENABLED="1" NAME="Metadata"/>
<FEATURE ENABLED="0" NAME="Clipping"/>
<FEATURE ENABLED="0" NAME="FinalQA"/>
<FEATURE ENABLED="0" NAME="Frame"/>
<FEATURE ENABLED="0" NAME="Attach"/>
...
</EDITINGTASK>
```

### 26.1.2 Task Definition

There is a general definition file for tasks. It is created in a hierarchic way as listed below.

The file is located in the ...\\proj-cfg\\Default folder and is called Default-QA-Tasks.xml.

The path can be changed to mirror project names: ..\\proj-cfg\\<customer>\\ and "<customer>-QA-Tasks.xml".

### 26.1.3 Default-QA-Tasks.xml

There are default \*.xml files after docWizz installation in the ...\\docWizz\\project-cfg\\Default folder.

For handling tasks you use the **Default-QA-Tasks.xml** file.

Use this file as a template for your current project.

To activate a task remove the underscore as for example <DOCUMENTTYPE NAME="\_Monograph">

## OUTLINE VIEW

Settings for the left working area in docWizz. Enter here e.g. ="Tree" and SELECTION="Chapter"

## DETAIL VIEW

Settings for the right working area in docWizz. Enter here e.g. ="Image"

## USERGROUP NAME

Enter a user and group here. This entry is projected, not working until now.

## PREVTASK

Define here which task has to be done and completed before this task.

With last task in this file all previous tasks have to be completed.

## SCRIPT\_SELECT

Scripts can be entered.

Example:

By default the functionality is disabled with an underscore in the beginning of the <DOCUMENTTYPE NAME>.

Per project it is only possible to enable a certain editing task, e.g. Monograph: ReTyping=0, Newspaper: ReTyping=1 is not possible.

```
<TASK_DEFINITION>
  <DOCUMENTTYPE NAME="Monograph">
    <JOB NAME="VerifyHierarchy">
      <TASK NAME="Structure">
        <OUTLINE VIEW="Tree" SELECTION="Chapter"/>
        <DETAIL VIEW="Image"/>
        <USERGROUP NAME="STRUCTURE"/>
        <DIFFERENT_USER>1</DIFFERENT_USER>
        <SCRIPT_COMPLETE>source                                     [expandpath
***DATA***\\SCRIPT\\task\\extraTaskDoOCREditing.tcl]           ;      ExecuteOnPocess
ComputeMissingOCR OCR</SCRIPT_COMPLETE>
      </TASK>
      <TASK NAME="ComputeMissingOCR">
        <OUTLINE VIEW="List" SELECTION=""/>
        <DETAIL VIEW="Image"/>
        <USERGROUP NAME="AUTO_PROCESSING"/>
        <DIFFERENT_USER>1</DIFFERENT_USER>
        <BACKGROUND>1</BACKGROUND>
        <PREVTASK>Structure</PREVTASK>
      </TASK>
      <TASK NAME="OCR">
        <OUTLINE VIEW="Text" SELECTION="Headlines"/>
        <DETAIL VIEW="Image"/>
        <USERGROUP NAME="OCR"/>
        <DIFFERENT_USER>0</DIFFERENT_USER>
        <PREVTASK>ComputeMissingOCR</PREVTASK>
        <PREVTASK>Structure</PREVTASK>
      </TASK>

      <!--for ReTyping enable next 3 tasks-->
      <TASK NAME="OCRKeying">
        <OUTLINE VIEW="Text" SELECTION="Keying"/>
```

```

        <DETAIL VIEW="Image"/>
        <USERGROUP NAME="OCR"/>
        <DIFFERENT_USER>0</DIFFERENT_USER>
        <PREVTASK>ComputeMissingOCR</PREVTASK>
        <PREVTASK>Structure</PREVTASK>
    </TASK>
    <TASK NAME="Typing">
        <OUTLINE VIEW="Text" SELECTION="Keying"/>
        <DETAIL VIEW="Image"/>
        <USERGROUP NAME="KEYING"/>
        <SCRIPT_SELECT></SCRIPT_SELECT>
        <SCRIPT_COMPLETE>source                                     [expandpath
{***DATA***\\script\\reject\\RejectByEnt_main.tcl}]             ;           return
[IsTaskComplete Typing]</SCRIPT_COMPLETE>
        <DIFFERENT_USER>1</DIFFERENT_USER>
        <PREVTASK>OCR</PREVTASK>
        <PREVTASK>ComputeMissingOCR</PREVTASK>
        <PREVTASK>Structure</PREVTASK>
    </TASK>
    <TASK NAME="ReTyping">
        <OUTLINE VIEW="Text" SELECTION="Keying"/>
        <DETAIL VIEW="Image"/>
        <USERGROUP NAME="KEYING"/>
        <SCRIPT_SELECT></SCRIPT_SELECT>
        <SCRIPT_COMPLETE>source                                     [expandpath
{***DATA***\\script\\reject\\RejectByEnt_main.tcl}]             ;           return
[IsTaskComplete ReTyping]</SCRIPT_COMPLETE>
        <DIFFERENT_USER>1</DIFFERENT_USER>
        <PREVTASK>Typing</PREVTASK>
        <PREVTASK>OCR</PREVTASK>
        <PREVTASK>ComputeMissingOCR</PREVTASK>
        <PREVTASK>Structure</PREVTASK>
    </TASK>
    <TASK NAME="CompareKeying">
        <OUTLINE VIEW="Text" SELECTION="Keying"/>
        <DETAIL VIEW="Image"/>
        <USERGROUP NAME="KEYING"/>
        <DIFFERENT_USER>0</DIFFERENT_USER>
        <SCRIPT_SELECT></SCRIPT_SELECT>
        <SCRIPT_COMPLETE></SCRIPT_COMPLETE>
        <PREVTASK>ReTyping</PREVTASK>
        <PREVTASK>Typing</PREVTASK>
        <PREVTASK>OCR</PREVTASK>
        <PREVTASK>ComputeMissingOCR</PREVTASK>
        <PREVTASK>Structure</PREVTASK>
    </TASK>
    ...
</JOB>
</DOCUMENTTYPE>
</TASK_DEFINITION>

```

## 26.1.4 User management

Having two users as:

Editing1=EDITINDEXLISTS,

Editing2=EDITINDEXLISTS,

### Example:

You have 4 tasks in Z-RST for a project.

Logon to docWizz as Editing1, and correct the first two editing tasks, then logon with Editing2 and correct one editing task, you can see that Editing 2 cannot reset Editing1's completed editing tasks.

The user management is related to backwards compatibility onto not letting another user ( beside the Administrator user) to modify the completed editing tasks of another user. For forward user management, each editing task can be configured to need a different user to correct the next editing task - by this we mean any other logged on user in docWizz, even the same that did the previous editing task. You cannot have it explicitly to have some users with some user rights do some editing task and other users other editing tasks.

## 26.1.5 Task storage


In the Pool database a new table "DocumentTasks" is created which contains all tasks that are assigned to a document:

Name	Type	Description
P_ARTID	Varchar	Reference to the document where it belongs to
P_NAME	Varchar(40)	Name of the task
P_STATUS	Char(1)	Current status of the task: W – wait for other task P – in progress D – done
P_GROUP	Varchar(40)	Name of the operator group which are allowed to perform this task
P_USER	Varchar(40)	Name of the operator who finished the task
T_SEQUENCE	Int	Sequence of task

## 26.1.6 Assigning tasks to documents

Whenever a document reaches a new interactive job, the configuration is checked and configured tasks are added to the database.

## 26.1.7 Workflow

When clicking on  **Process**, the current task status is set to complete.

If there is no more task to be done, regular processing starts. Else it is checked, whether there is another task which might be performed by the same user.

- If not, the document is released and another document can be opened.
- If yes, the next one is selected automatically.

User interface switches to views as configured. A message box tells which next task shall be performed now.

### 26.1.8 Dynamic Tasks

Whenever a task is completed, a script may be called to verify results. Completion will not be allowed, if the script returns false. Another use-case would be to set for example the status of DoubleKeying to done if Re-Keying is completed and no differences to OCR are found.

### 26.1.9 Log

As soon a task is completed, a log is created in BatchResults table. The job name will be extended by the task name, separated by “:”. FTP logs are saved per client.

A column "Status" contains the status a document got when creating the record. This helps to monitor status changes.

### 26.1.10 Script commands

Script commands are available for adding, changing and deleting a task. Those script commands are available for pool view and for a document itself.

### 26.1.11 XML structure

QA Editing Task - XML structure

#### Start and End tag

```
<TASK_DEFINITION> u. </TASK_DEFINITION>

<DOCUMENTTYPE NAME="XXXX">
  - All entries from \doccfg\ can be used (e.g. Monograph, Newspaper,
  etc.)
  - Per default the QA Tasks is deactivated by an underscore =>
  <DOCUMENTTYPE NAME="_Newspaper">
<JOB NAME="XXXX">
  - All interactive jobs from docWizz are valid : Z-RST, Z-RI, Z-RPS, etc.
<TASK NAME="XXXX">
```

#### Elements in <TASK>

```
* <OUTLINE VIEW="Text" SELECTION="Headlines"/>
* <DETAIL VIEW="Image"/>
* <USERGROUP NAME="KEYING"/>
* <DIFFERENT_USER>0</DIFFERENT_USER>
  <SCRIPT_SELECT>source [expandpath ***DATA***\SCRIPT\QATest.tcl] ;
  PrepareForReTyping Headline</SCRIPT_SELECT>
  <SCRIPT_COMPLETE>[doc] -qatask VerifyHierarchy Clipping
1</SCRIPT_COMPLETE>
+ <PREVTASK>VerifyStructure</PREVTASK>
```

\* = entries that should be set.

+ = with first logical QA Task (here: VerifyStructure) this task is not available



### 26.1.12 Rejects by editing tasks

Rejects can also be customized to customer's needs for the editing tasks. There are the same automatic checks available within editing tasks.

When rejects for tasks are enabled only the rejects for that tasks will be computed. Also at the end of the step the rejects for tasks will not be computed. QA user will be able to verify his work.

If no rejects for tasks are defined the program act like before.

In the following sections of the file **\project-cfg\Default\reject\RejectByEnt.xml** the rejects can be defined:

```
<JOB name="VerifyHierarchy">
....rejects per step....
  <TASK name="Typing">
    <AUTO_FIX name="VH_SetKeyingCorrectionList"
    enabled="1">***DATA***\script\docWizz_QA.tcl</AUTO_FIX>
    <REJECT_CONDITION name="Default_Task_CheckEmptyTyping" enabled="1"/>
  </TASK>
  <TASK name="ReTyping">
....rejects per task...
  </TASK>
</JOB>
```

And in Default-QA-Tasks.xml: the line

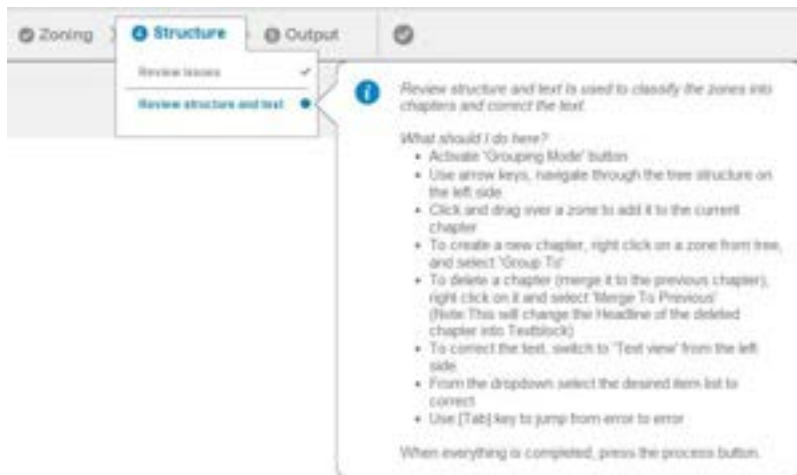
```
<SCRIPT_COMPLETE>source                                     [expandpath
{***DATA***\script\reject\RejectByEnt_main.tcl}}           ;          return
[IsTaskComplete Typing]</SCRIPT_COMPLETE>
```

like here:

```
<TASK NAME="Typing">
<OUTLINE VIEW="Text" SELECTION="Keying"/>
<DETAIL VIEW="Image"/>
<USERGROUP NAME="KEYING"/>
<DIFFERENT_USER>0</DIFFERENT_USER>
<SCRIPT_SELECT></SCRIPT_SELECT>
<SCRIPT_COMPLETE>source                                     [expandpath
{***DATA***\script\reject\RejectByEnt_main.tcl}}           ;          return
[IsTaskComplete Typing]</SCRIPT_COMPLETE>
<PREVTASK>VerifyOCR</PREVTASK>
<DIFFERENT_USER>1</DIFFERENT_USER>
</TASK>
~
```

## 26.2 Structure

Is the first editing task for **Review structure and text** task, and is used for correcting the hierarchy of the document (classify the zones into articles or different structures).



After the task is completed, the user can send the document to the next task by clicking on the next task or using "Process" button.

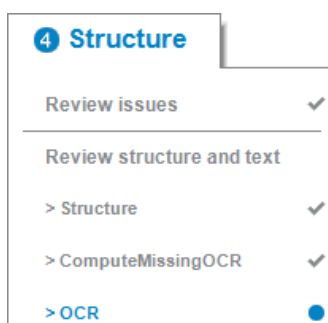
## 26.3 ComputeMissingOCR

This task is used to detect the missing text from the document. Because it's a time-consuming operation, it is processed in dW Services.



## 26.4 OCR

This task is used for correcting the text detection. The document is automatically opened on left Text view with the selection on Headlines list.



If additional zone types need to be corrected, the user can change the selection from the dropdown list. When the task is finished, the document can be sent to the next task using the "Process" button or selecting the next task from the workflow dropdown.

## 26.5 Double Keying

There are two possible workflows:

- 1.) VerifyOCR / Re-Typing / CompareKeying
- 2.) Typing / Re-Typing / CompareKeying

As you need different configurations for each workflow you can only do one option in one project.

### Verify OCR

OCR is done, but there are no results (e.g. a headline is written in handwriting) and an operator reads and types the headline manually. Input is done by an operator.

**Typing** (the one who does input 2 does simultaneously final correction)

Input 1 is done by first operator or OCR engine. 2nd operators sees the mismatches when (s)he is (re-)typing. mismatches are indicated visually by colour and/or by sound alerts (configurable). As 2nd operator is super operator, the 2nd result is the one that counts for Export.

**Re-Typing** (type again to get higher accuracy)

Input 1 is done by first operator or OCR engine. 2nd operator does not see 1st input or any mismatch when he is doing Input 2. A third operator (supervisor) checks differences between Input 1 and 2 and does correction (dropdown list in left hand working window to show mismatches, or colour indication in full text editor field). His corrected result is the one that counts for Export.

**Compare Keying.** Used to compare the OCR result with the manually typed input.

User will get both results, from the automatically detected OCR and from Re-Typing, where the OCR was manually corrected. In another case, in Compare keying, user will get the OCR from Typing and Re-typing (both manually corrected). So it is safe to say that third operator gets the OCR detected automatically and verified/corrected by the first operator and the second manually corrected OCR from the Re-Typing operator. You can easily remove one operator, and just leave the OCR as it is detected in VerifyOCR, without correcting it, and then use the first operator for re-typing, and a second one for compare keying.

**Typing** and **Re-Typing** are only input tasks. **Compare Keying** then shows the differences.

### The first workflow case

VerifyOCR editing task enables the user to have the detected OCR in the keying list to verify/correct the OCR. Re-Typing editing task, user has the empty keying list to manually correct all the OCR. In Compare keying editing task, user will have the possibility to compare the two text variants (from VerifyOCR and from Re-typing) and select a suitable one, if there are differences, and even to correct the text. The last corrected text version is kept. Headlines, that are the same (the variant from VerifyOCR and Re-Typing are identical) will not be displayed in CompareKeying editing task.

Compare Typing and Final QA in one task or separate even if it will be done by the same user?

You can drop Final QA if you'd like and leave the last one CompareKeying. If you want OCR comparison, then drop Re-typing.

### The second workflow case

User has an empty keying list in Typing editing task, where user will manually correct all the Text for the list. In Re-Typing editing task, user has an empty keying list as well, to manually correct the text. In Compare Keying, user will compare the text variants from Typing and Re-Typing, and select the good one, and even correct it where it needs. The last kept text in the list will be taken for the Text of the zone.

The behavior is dependent on the configuration. The OCR compare is done only when you have just Typing and Compare keying. When you have Typing and Re-Typing the Compare Keying is done between Typing and Re-typing.

Having just typing, it will compare the typing with the provided OCR from FineReader. Typing sets the first input to empty for the user to write it from scratch, as then to be compared, either with input 2 from Re-typing or from OCR.

It is a decision, based on OCR results if you going to use FineReader OCR as a compare method, or the manual OCR correction from two operators. For example bad images from FineReader you might use two input tasks to get good results. If you have good FineReader results you may only need one input method.

### 26.5.1 Double Keying configuration

Double keying is another expression for the tasks: VerifyOCR, Typing, ReTyping and CompareTyping.

#### Default-QA-Tasks.xml

In this file all available Double Keying tasks are stored.

#### Configure on which zones Double Keying is applied

A section in project-setting.xml is used to define zones:

```
<DOUBLEKEYING>
  <ZONETYPES>Headline Author</ZONETYPES>
</DOUBLEKEYING>
```

#### Default-Setting.xml

For project specific configuration copy the default file to <Projectname>-Setting.xml

There is a section called <EDITINGTASK>. Here you enable (1) or disable (0) a task without creating a new configuration. Required previous tasks will be skipped automatically if not enabled. If a task is not listed it is treated as disabled. Changes take effect on existing documents when loading.

Example:

```
- <EDITINGTASK>
  <FEATURE NAME="VerifyStructure" ENABLED="1" />
  <FEATURE NAME="ComputeMissingOCR" ENABLED="1" />
  <FEATURE NAME="VerifyOCR" ENABLED="1" />
  <FEATURE NAME="Typing" ENABLED="0" />
  <FEATURE NAME="ReTyping" ENABLED="1" />
  <FEATURE NAME="CompareKeying" ENABLED="0" />
  <FEATURE NAME="VerifyMetadata" ENABLED="1" />
  <FEATURE NAME="VerifyClipping" ENABLED="1" />
  <FEATURE NAME="FinalQA" ENABLED="1" />
</EDITINGTASK>
```

## 26.6 Keying – First variant – compare different text input

### How to configure

In order to enable double keying, the sub-tasks have to be activated.

In \*\*\*Project-cfg\*\*\*\<Project-Name>\<Project-Name>-Settings.xml file, navigate to <EDITINGTASK> node and set to "1": "Typing", "Retyping" and "CompareKeying":

```
<PROJECT-SETTINGS>
...
<EDITINGTASK>
...
  <FEATURE NAME="Typing" ENABLED="1"/>
  <FEATURE NAME="ReTyping" ENABLED="1"/>
  <FEATURE NAME="CompareKeying" ENABLED="1"/>
```

Additional settings can be added for double keying: what zones to be corrected and what words to be skipped.

In the same file ...\<Project-Name>-Settings.xml, navigate to <DOUBLEKEYING> node. There are two sub-nodes:

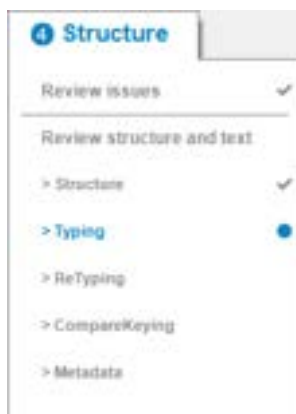
```
<PROJECT-SETTINGS>
...
  <DOUBLEKEYING>
    <ZONETYPE ENABLED="1" like="1">Headline Author</ZONETYPE>
    <TYPING_CONDITION ENABLED="0" THRESHOLD="500"
      MINCHARS="5"></TYPING_CONDITION>
  </DOUBLEKEYING>
```

Inside <ZONETYPE> node are defined the zones to be corrected. "like" can be set to "1" for correcting also the zones derived from the defined zones.

The node <TYPING\_CONDITION> can be enabled to filter the words that will be corrected. The **TRESHOLD** sets the minimum OCR confidence value and **MINCHARS** sets the minimum number of characters inside the word.

### Workflow

After activating the sub-tasks, they will appear in the dropdown from Structure group:



## Typing

Typing is the first out of the three double keying sub-tasks. On left side Text view, from the dropdown list, select "Keying". This will populate the view with the zones that need to be typed (based on the settings from ...\\<Project-Name>-Settings.xml file).

Type on each line the corresponding text – notice that the current word is underlined in Image display. Use "Enter" key to add a new line of text in the current zone and "Shift + Enter" to jump to the next zone.



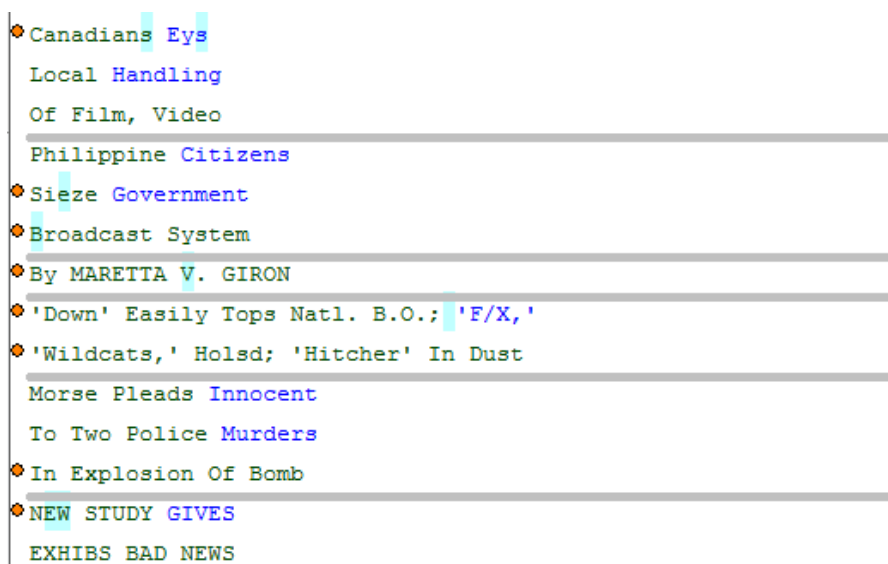
## ReTyping

ReTyping is the second sub-task for double keying. It is the same as the first sub task (Typing) and is performed by a different operator that the operator that completed the Typing sub-task.

## CompareKeying

The final sub-task for double keying is CompareKeying. Again, this is performed by a different operator.

In the same list (Keying) from Text view, only the zones where differences were detected between Typing and ReTyping are displayed. The user can use the symbol in front of the line to switch between the text from Typing and the text from ReTyping, or use "Alt + Right/Left arrow key" when the cursor is on the text line.



Differences between the texts are highlighted with light blue color.

In case neither text is correct (comparing to the original text), the user can correct it manually.

## 26.7 Keying – Second variant – compare input text with detected text

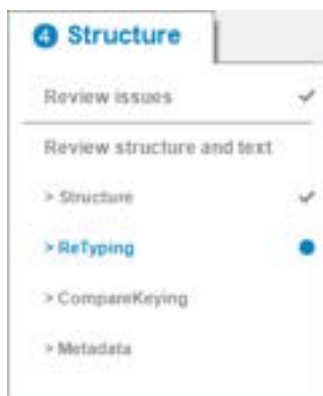
### How to configure

In order to compare the detected text with an input text, open the `***Project-cfg***\<Project-Name>\<Project-Name>-Settings.xml` file, navigate to `<EDITINGTASK>` node and set to "1": "OCRKeying", "Retyping" and "CompareKeying" (make sure that "Typing" is set to "0"):

```
<PROJECT-SETTINGS>
...
<EDITINGTASK>
...
<FEATURE NAME="OCRKeying" ENABLED="1"/>
<FEATURE NAME="Typing" ENABLED="0"/>
<FEATURE NAME="ReTyping" ENABLED="1"/>
<FEATURE NAME="CompareKeying" ENABLED="1"/>
```

### Workflow

The dropdown for Structure group should look like this:



### Sub-tasks

This variant only needs one task for typing the text, because this will be compared in CompareKeying with the detected text.

ReTyping and CompareKeying sub-tasks are the same in this variant as for the first variant.

### Tasks used together

Cases use Editing tasks together:

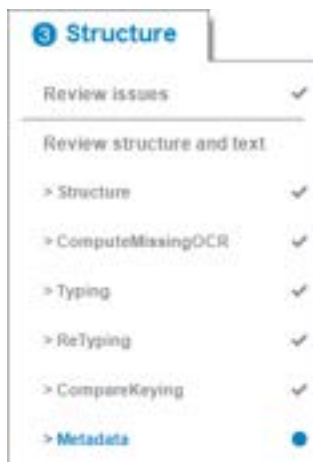
OCR	Typing	ReTyping	Compare
X	-	X	-
X	-	X	X
-	X	X	-
-	X	X	X

Default cases use Editing tasks together:

(OCR or Typing) + ReTyping + (Compare)

## 26.8 Metadata

The task is used for correcting the metadata of the document (check if the entries from the Metadata view are correct). On the left List view navigate through the elements of the METADATA list and on the right Metadata view check the entries from the elements of the tree.

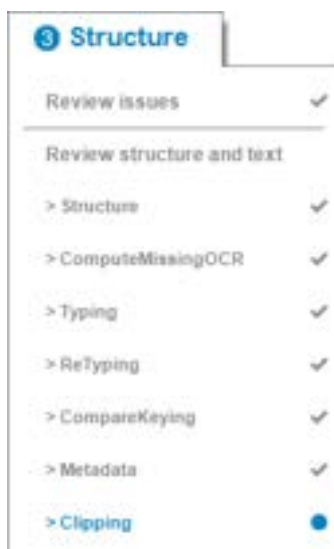


After the task is completed, the user can send the document to the next task by clicking on the next task or using "Process" button.

## 26.9 Clipping

This task is used to correct / view the clipping of articles (or other structures, if configured). It can be done using the left List view with "ToCorrectClipping" selected in the combobox and right Clip view. The list displays only the errors detected on clipping.

Clipping can be done also by using the left Tree view to navigate through articles and sections and right Clip view to inspect the clipping.



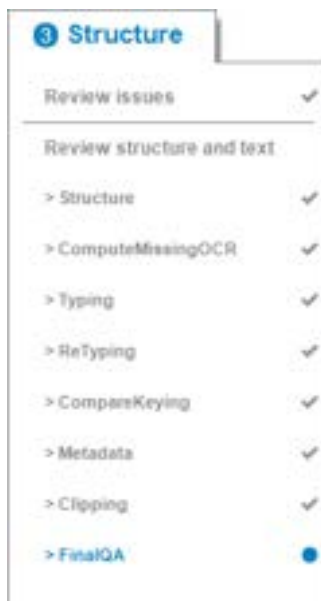
After the task is completed, the user can send the document to the next task by clicking on the next task or using "Process" button.



## 26.10 FinalQA

Is the last editing task for **Review structure and text** task, and is used for doing any kind of final verifications by a different operator.

Any problems found can be fixed in this task, before processing the document to Output.



After the task is completed, the user can send the document to Output using the process button or selecting Review output task from Output group.

## 27 Remote QA

The docWizz Remote QA system contains three components:

- FTP client (simple FTP client with priorities system, what to transfer first)
- Master system (on master machines, where processing is done)
- Slave system (on remote machine)

The communication between master and slave shall be done via command files, sent also using the FTP client.

In the next chapters we describe the implementation details.

Please be aware of the renamed steps and tasks and the "old" names in scripts and configuration files.

### 27.1 Handling Remote QA documents overview

#### Introduction

Remote verification of documents has been implemented in the docWizz processing for bigger projects. This means the process intensive automatic jobs will be carried out at the project location and the manual verification jobs will be carried out at a remote location. In simple words, the documents will be moved between two locations during the processing (compression and other processes are involved and are not discussed in this document).

#### Remote QA

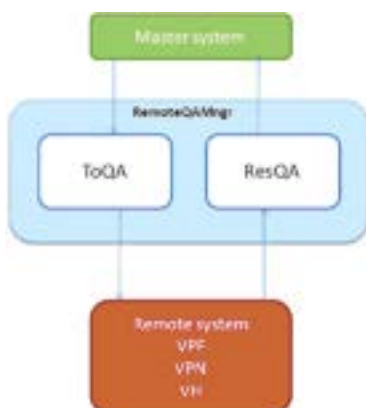
Continuous flow of documents in different "Verify" jobs is expected at the remote QA location (will be referred as 'remote system' in the rest of the document) from the project site (will be referred as 'master system' in the rest of the document). Therefore, it is necessary to monitor the transfer process to avoid the documents getting piled up at one end.

#### Priorities:

For example set "C-RC" in scan job as "stop next job". On slave it comes in "C-PC" and next job is "Z-RZ". This settings are probably taken from "RQA" project settings. So project cfg settings in files have more priority than manual setting in workflow bar. Mandatory stops are scripted and force to let the document stopped on defined step - independent to selection.

#### System overview

The service 'RemoteQAMngr' is responsible for receiving the documents from the master system and to route them to the remote site's docWizz environment and vice versa. Please refer to the below diagram.



The documents are stored in the folder "ToQA" before they go to the manual verification (at remote system pool). Likewise, the remotely processed documents are stored in the folder "ResQA" before they are sent back to the master system. These folders are located in the \docWizz\RemoteQAMngr\ (same level as the project-cfg folder).

#### **Update priority on RQA client using PoolMtnTask**

Priority should not be changed on Loader.

Manager is the one and only instance that should have control on priorities.

## 27.2 Remote QA environment

We tried to make the system as flexible as possible and we make it as a combination between hard coded parts and scripts run by DWSrv. There are scripts and one helper script in script/remoteqascripts and each of them handles a different part in this process.

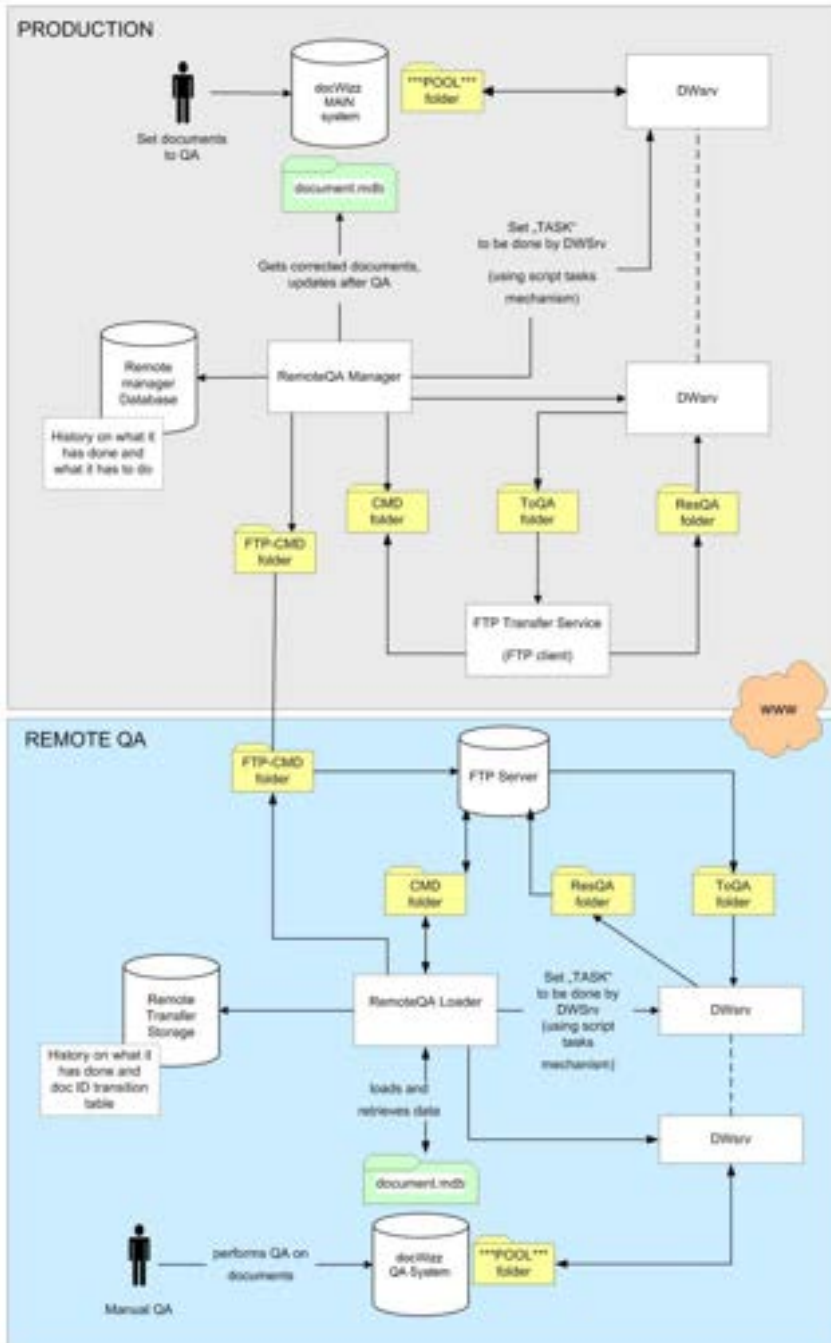
### How it works:

- some documents are set to RemoteQA status (either manually or by a postStep script)
- RemoteQAManager browse through pool and find these documents
- For each document it creates a list with "files of interest" and also keeps a copy of this to not send twice same files later
- Posts a processing order for DWSrv (a xml file containing the information generated at the step before)
- From time to time DWSrv executes the task "MNGRCOLLECTDATA" – this tasks creates a zip with data to be submitted – currently sends analyze, docID.xml and main images (high res), next step makes also jp2, does not send BW, lowRes. It might be possible to have projects where it makes sense to send all data, or only BW, does not matter, this script can be changed. After creating a zip, put a control file on disk
- RemoteQAManager finds that control file and creates a FTP command
- FTP client sends the zip and deletes it
- RemoteQAManager detects that the zip was successfully moved and generates a command for RemoteQALoader
- FTP client transfers this command
- RemoteQALoader monitor the commands folder and when it gets a command from RemoteQAManager check/updates "translation table" (new doc ID for remote document) and post a command for DWSrv to "prepare" the new received document for correction
- DWSrv executes task "LDRUPDATEDATA" - unzip, makes the ID translation, moves the files in pool perform other operation (f.e transform jp2 into tif, generates BW, low, etc). Currently no change is made on the images, next version will transform low quality jp2 into tiffs) and notify the RemoteQALoader
- RemoteQALoader updates pool database and the document is ready for correction (Remote QA in progress status)
- Correction is done and the document is then processed (press OnProcess) or routed – the document goes to another status (Remote QA correction done)
- RemoteQALoader browses through pool and finds these documents.
- For each document creates a list with "files of interest"
- Posts a processing order for DWSrv (an xml file containing the information generated at the step before)
- DWSrv executes the task "LDRCOLLECTDATA" – this creates zip to be sent back to RemoteQAManager and notifies (via a control file) the RemoteQALoader
- RemoteQALoader posts a command for the FTP-client
- FTP-client executes the command (transfer zip file and delete it)
- RemoteQALoader detects that the zip was successfully moved and generates a command for RemoteQAManager
- FTP client transfers this command
- RemoteQAManager monitor the commands folder and when it gets the command generate a processing order for DWSrv
- DWSrv executes "MNGRUPDATEDATA" – unzip, merge docID.xml (for corrected frames, printspace, moved pages and so on) and puts the updated data in pool and notifies the RemoteQAManager
- RemoteQAManager updates pool database and the document is ready for next step execution (default status and routed to the right batch job)
- Each document can have an individual priority assigned. This is considered within the services. Also visible in PoolDialog. Therefore documents can be worked on prior-ranking (redelivery, complete Binding/reel, ...). This information is also exchanged between RQA Systems.
- Multiple production environments: A remote QA location is now able to support multiple production environments. So we may send data from multiple production environments (e.g. customer, NAS) to a single destination (e.g. ROM). As well n:n relation is supported.
- When a document returns from RQA, the processing information (time, duration, user, ...) is stored into log database (batchresults) on production side.

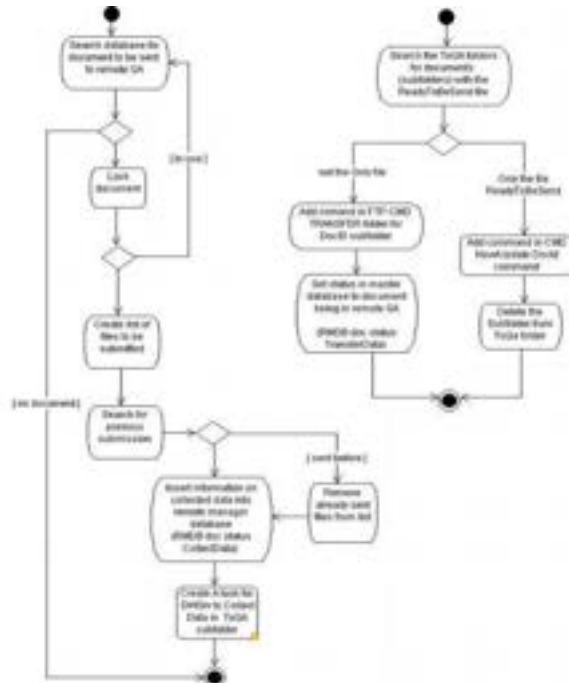
- A script can be added to be called when documents return from remote QA. This could be used to overwrite progress settings done in remote QA. For example you could add code here to apply final QA on production side. As well you could change status to hold to prevent documents being exported. For configuration add `<ScriptBeforeUpdatePool>script</ScriptBeforeUpdatePool>` into RemoteQAManagerCfg.xml on root level.

Here we tried to explain the workflow in logical steps. In real life Manager, Loader and FTP-client have each about four threads that do different things in the same time, monitoring files and check for different commands and so on.

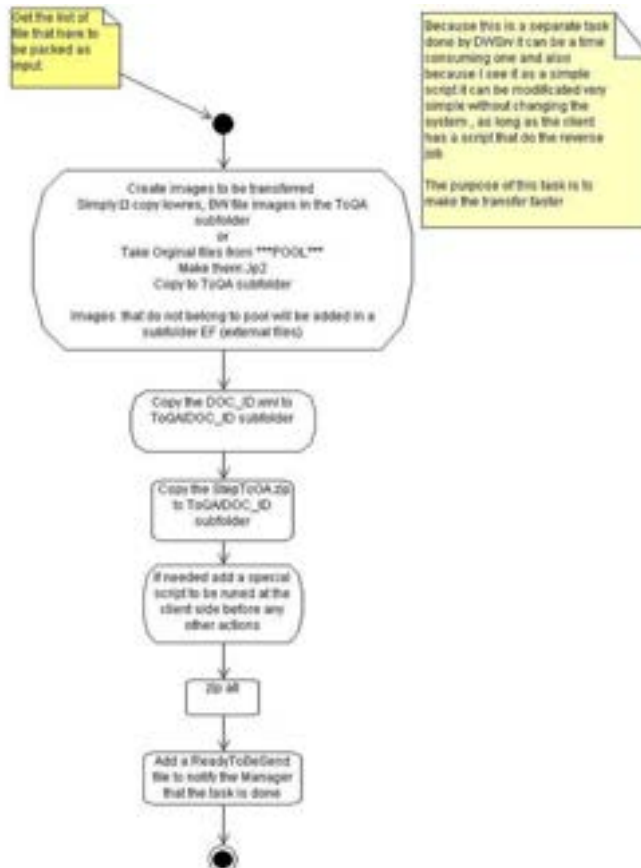
### 27.2.1 Overview



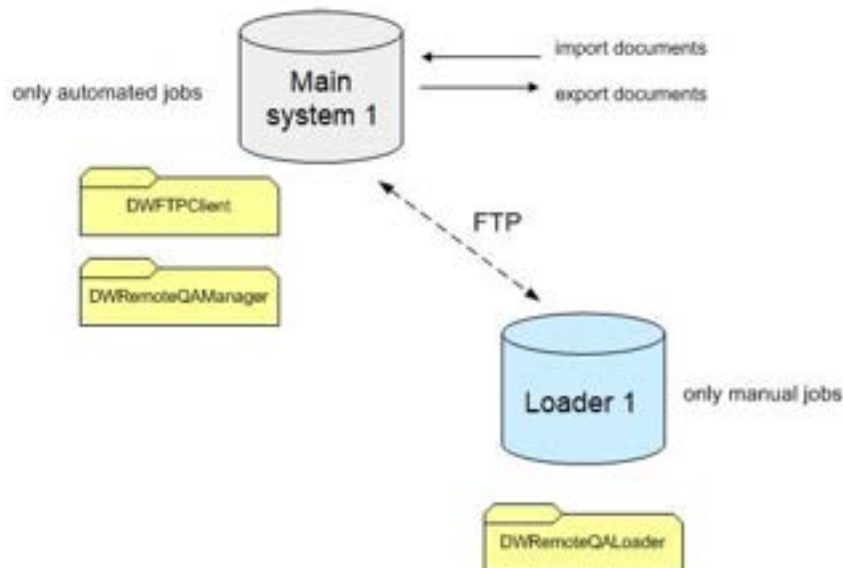
## 27.2.2 Collect and prepare data for Remote QA



### Task for dWSrv to collect data



### 27.2.3 Main system



The main system job collects the data from the pool for all documents to be corrected remote

- to manage the sent files (to avoid sending twice same information in case of multiple steps correction),
- to check if corrected documents are available and replace the analyze files with the corrected ones
- and to create best images to be sent to the loader system (this system will be continuously improved and shall be as flexible as possible).

The files RemoteQAManagerCfg.xml and RemoteQALoaderCfg.xml have to be there and registered.

The main system will permanently check on one thread if new data is available (documents in status REMOTE\_QA). When such a document is found the main system will prepare the data to send to remote QA:

- create a temporary folder
- lock the document until the remote correction is finalized
- create images to be transferred (simply copy lowres file images, create low quality images from original tiffs and so on) and copy to temporary folder. Images (their "copies" lowres, low quality, etc) - that do not belong to pool will be added in a subfolder EF (external files). A check to ID\_RQ.XML file shall be made before sending the file (if was sent before, skip).
- create/update ID\_RQ.XML file – this contains all the transferred files, names, checksum, size, but also the original file information, etc, all the necessary info to not send twice data
- move in OUT/ID folder the data to be transferred (maybe create an archive, faster transfer – archive with no compression, the data is already compressed, or should be)

If DWRemoteQAManager shows UNC macro errors and the other services run fine this can be caused by the XP share limitation for 10 accesses.

#### 27.2.3.1 IN, OUT and COMMANDS folders

While data is moved to remote machine successful this is deleted. When all data was moved from OUT/ID folder a notification file will be added (FTP client job). The main system will check in a separate thread for these files (or for empty folders? No notification files needed) and when this is found it will send a command file to the loader system (via FTP). The commands will be **xml** files named xxxxxxxx.xml

(incremental counter) to ensure that all the commands will be processed in order. Then the OUT subfolder will be deleted.

On the file server machine there are three folders for data transfer: **IN**, **OUT** and **COMMANDS**

- on **IN** the returned analysis
- in **OUT** folder all the data that need to be sent to the loader machines
- and on **COMMANDS** notifications from loader system.

Another main system job would be to monitor the **COMMANDS** folder – here a notification from the server is given, anytime new corrected data is available.

- the **IN** folder will be checked (this will have same structure like **OUT** folder – subfolders with doc ID)
- document should already be locked!
- the new corrected analysis is moved (or new **ID.XML** file in case of frames correction) into the pool
- necessary changes are made to the main pool database (f.e routing to next batch processing job, etc, depending on the loader command)
- unlock the document

#### 27.2.3.2 Commands

MAIN TO LOADER:

```
<command>
<createnewdoc ID="docID">
  <zip>zipfilename.zip</zip>
</createnewdoc>
</command>

<command>
<updatedoc ID="docID">
  <zip>zipfilename.zip</zip>
</updatedoc>
</command>

<command>
  <deletedoc ID="docID"/>
</command>
```

LOADER TO MAIN:

```
<command>
  <replace ID="docID">
    <file filename="filename"/>
    <file filename="filename"/>
  </replace>
  <routeto job="jobName">
</command>
```



### 27.2.3.3 RemoteQAManagerCfg.xml

The RemoteQAManager collects and prepares data for remote QA.

The file `***CONFIG***\QA\RemoteQAManagerCfg.xml` contains the `DWServiceCommandPath` – where the RemoteQAManager will post orders for DWSrv

```
<?xml version="1.0" encoding="UTF-8" ?>
<CONFIG>
  <DWServiceCommandPath>***DATA***\RemoteQA\cmd</DWServiceCommandPath>
  - <Clients>
    - <Client name="Name_of_Remote_Location">

      <MainClientPath>***DATA***\RemoteQA\clients\Name_of_Remote_Location</MainClientPath>
      <Projects>
        <Project name="NAME" ColorAsGray="1">
          <CompressionRate>100</CompressionRate>
          <CompressionRateVPF>200</CompressionRateVPF>
        </Project>
        <Project name="Default" ColorAsGray="1">
          <CompressionRate>100</CompressionRate>
          <CompressionRateVPF>200</CompressionRateVPF>
        </Project>
      </Projects>
      <CompressionRate Project="Name_of_project">100</CompressionRate>
      <CompressionRate Project="Default">100</CompressionRate>
      <CompressionRate>100</CompressionRate>
      <createzip>1</createzip>
      <PoolMaintenanceTimeInterval
TimeUnit="hours">1</PoolMaintenanceTimeInterval>
    - <Steps>
      - <Steps>
        - <Step name="VerifyPageFrames">
          <Other />
        </Step>
        - <Step name="VerifyPages">
          <Other />
        </Step>
        - <Step name="VerifyLayoutElements">
          <Other />
        </Step>
        - <Step name="VerifyPageNumbers">
          <Other />
        </Step>
        - <Step name="VerifyPagesHierarchy">
          <Other />
        </Step>
        - <Step name="VerifyHierarchy">
          <Other />
        </Step>
      </Steps>
    </Client>
  </Clients>
</CONFIG>
```

```

        </Steps>
        <StepToDelete>Exported</StepToDelete>

        <PoolStatusPath>***DATA***\RemoteQAMngr\clients\Name_of_Remote_Location\
PoolStatus</PoolStatusPath>
        </Client>
    </Clients>
<LogPath>***DATA***\WORK\LogRemoteQAMngr</LogPath>
</CONFIG>

```

## Clients

A manager can dispatch documents for remote QA to different locations (frames in one location, zones in another). Each client has its own folders for data transfer, commands transfer and so on.

Enter the correct name of remote location (example: ROM, DDD).

## Compression Rate

One for S-PC (just for frame setting needed), the second for all further jobs (Z-RZ, Z-RPS, Z-RI, Z-RST). Typical setting for Newspaper (original scans):

```

<CompressionRateVPF>200</CompressionRateVPF>
    <CompressionRate>100</CompressionRate>

```

100 means:  $\text{<source image size>} / 100 = \text{<RQA image size>}$

200 means:  $\text{<source image size>} / 200 = \text{<RQA image size>}$

Typical setting for Newspaper (microfilm scans):

```

<CompressionRateVPF>50</CompressionRateVPF>
    <CompressionRate>25</CompressionRate>

```

The compression is calculated based on the source image size. Source image file size is 25 MB => RQA image will be 250 KB.

This will cause issues, in case the source images are of inconsistent types and different compressions will be provided.

E.g. TIF uncompressed mixed with TIF LZW compressed.

For the uncompressed images the RQA images will be very good, for the LZW compressed images the RQA images will be in comparison very small and probably not readable.

To take instead of file size the real memory size of the images negative values can be configured in the cfg.

```

<CompressionRateVPF>-50</CompressionRateVPF>
    <CompressionRate>-25</CompressionRate>

```

This will lead all RQA images really have the same compression based on the source image.

## ColorAsGray

```

<Projects>
<Project name="MyProject" ColorAsGray="1">
<CompressionRate>100</CompressionRate>
<CompressionRateVPF>200</CompressionRateVPF>
</Project>

```

When parameter "ColorAsGray" is set to 1, any color image is converted to gray before creating jp2 rqa files.

## Projects

Enter the correct project name.

When parameter "ColorAsGray" is set to 1, any color image is converted to gray before creating jp2 rqa files. Is handled by Script and Code.

## MainClientPath

The root for the RemoteQAManager files (commands, zips and so on, including one path for FTP commands). Below this path (not configurable) are the following subpaths:

- Cmd - with subfolders MngrGet and MngrPut – here the commands are prepared for RemoteQALoader (MngrPut) and received from RemoteQALoader (MngrGet)
- ToQA – here the zips to be sent to RemoteQALoader are stored
- ResQA – here the zips from RemoteQALoader are received
- FTP\_CMD – here FTP commands are sent

## Steps

What manual steps will be corrected by this client.

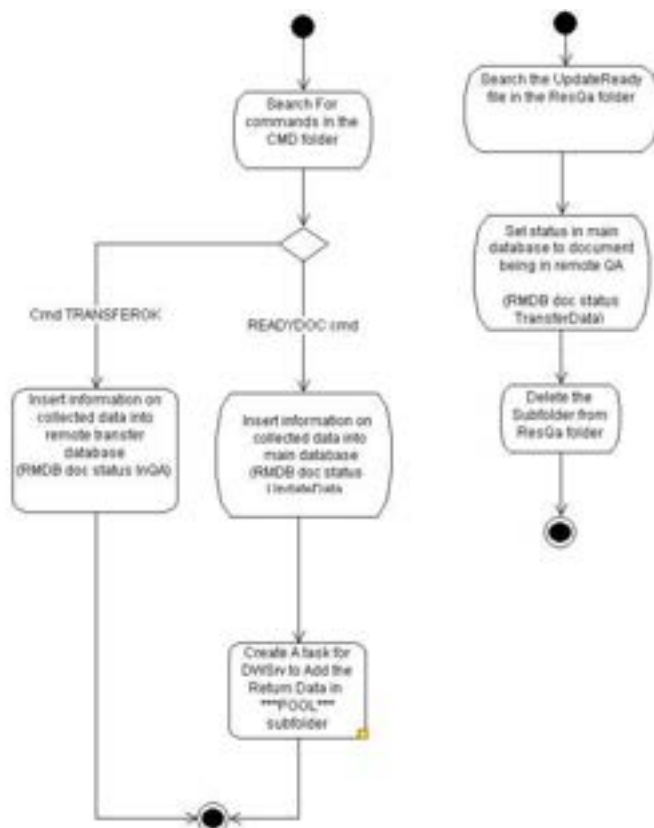
## LogPath

The RemoteQAManager will log all the activity, in 4 text files, one for each thread – we intend to make a console that will be able to monitor these files, and also to be used for CFG.

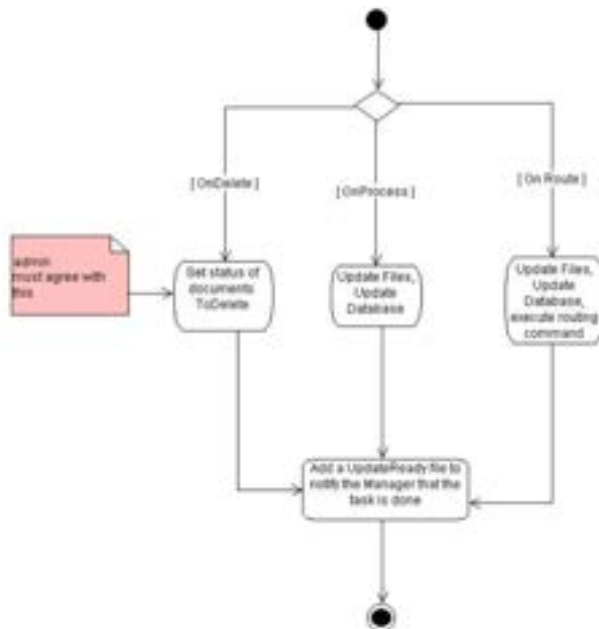
## Add more clients

Copy the Client section inside this [\\*.xml](#) file and rename the client name and main client path.

## Take results from QA and insert into Main system



## Task for dWSrv to add the Return Data in \*\*\*POOL\*\*\*



## Clarification of SENDIMAGES

File `***DATA***\config\QA\RemoteQAManagerCfg.xml`

What is the purpose of this section ?

```
<SendImageTypes>
<ImageType>SRCIMAGE</ImageType>
<ImageType>PIMAGE</ImageType>
<ImageType>P_ORIGIN</ImageType>
</SendImageTypes>
```

Each type of image that is to be sent in the <ef123456> folder (subfolder in pool) on loader. Basically in previous versions, only if you sent the document from C-PC, you would have had the <ef123456> folder on loader, otherwise sending in bigger steps would not send the images. It is configured as to send them for the RQA need (if you have projects that use C-PC step, then you will send them, or projects where you would reset OCR the same you would need these images.)

SRCIMAGE

Previous image ( the image that is the previous PIMAGE, see <PAGE> -setimage , if -modify is used, SRCIMAGE is kept as previous image)

P\_ORIGIN

Source image (the original input image from IN folder or on RQA the image in the <ef123456> folder -> Show Origin button)

PIMAGE

Hi res image, currently existing in the document

It is configurable, but it is recommended to have all of them, your encountered case is specific to sending a document for the first time in another step than C-PC ( e.g. client system corrects C-PC on main system, and Z-RZ on loader side, thus client has to know to send the needed images. This due to the fact that, when sending a document from C-PC (in C-PC, P\_ORIGIN and PIMAGE are the same, and into the next step that they are sent, P\_ORIGIN is already there, and that image is not resent on loader. Thus if they are configured, all three types, they are sent on re-send to loader.

General recommendation is to have them all enabled, and configuration to be done on specific environment (that has C-PC corrected)

What should be sent to loader?

Usually you send them all, for correction, but you can select not to send.

Are all three entries active at the same time?

If you have them selected, you will send the 3 types of images.

In the example above PrepareData is only checking for changes on source - not RQA images.

Check also the RQA images to be verified and send images back, if the RQA images are different to the initial ones use:

```
<ImageType>SRCIMAGE_rqa</ImageType>
<ImageType>PIMAGE_rqa</ImageType>
```

Detects also if the RQA images where changed from the last change.

Disable or enable P\_ORIGIN ?

There are different scenarios that need to be taken into account and indeed most efficient is not to have it set, but safest is to have it set.

For any new project, when the workflow is not finally validated, or for small size projects it is better to have P\_ORIGIN enabled.

For projects where the workflow is clear, and proved in production, this can be disabled (if C-PC correction takes place on loader too). That cfg tells what should be the images checked for availability on loader and send in case they differ.

P\_ORIGIN is usually same with SRCIMAGE when images are sent for C-PC correction, so normally this is not needed by default.

It is needed in case a document was not sent before in C-PC, but directly in Z-RZ, and then a re-cropping of the page shall be done in Z-RZ on loader. In this case the images for re-cropping are not available on loader, so there should be a way to send these images too in Z-RZ.

Other situation: it appears for example when a document is exported and deleted from the loader, then you need for some reasons to re-crop a page, but you don't want to re-correct the entire document, but save what is possible. Then you go back to Z-RPS f.e. and send document for re-correction - if P\_ORIGIN is not configured, the document will arrive fine on loader, but when you try to "Show Origin", this will not be available. These are few rare cases when that P\_ORIGIN is needed, this is why we would recommend to disable it for projects where everything is properly set and proved, but keep it for new projects/small projects. The "side effect" (not optimized situations) appear whenever you resend a documents, then P\_ORIGIN version of the images is recomputed and resent, even you are in Z-RPS and you don't need them, they are already on loader and they are correct.

## **Task Maintenance Pool**

The task Maintenance Pool compares the different Pool on local and remote QA station.

#### 27.2.3.4 FTP Client

For the RemoteQA numerous data must be transmitted into both directions. In addition CCS developed a FTP service, which automatically transfers the necessary data and supervises the status.

A simple FTP client - able to send files (move files) with a certain priority (main commands shall have priority) should be used. Also it should be able to add some notification when all data is transferred.

The home directory of the FTP server has to be configured to "RemoteDate" folder.

You can find the "default" path in the DW configuration ..\RemoteQALoader\RemoteDate.

If a loader gets documents from different RemoteQAManager there have to be several RemoteData directories (RemoteData, RemoteData1, RemoteData2 etc.).

For the appropriate FTP User configured in the RemoteQAManager (in detail: in <custom>-FTPConfig.xml under ..\docWizz\config\QA) the home directory has to be set to the belonging RemoteData directory.

##### 27.2.3.4.1 ftpconfig.xml

To setup an new RemoteQA environment copy the file \*\*\*CONFIG\*\*\*\QA\ftpconfig.xml and rename it with client name. (Example: CCS\_FTPconfig.xml).

Inside the file rename User name and service name (Example: DWFTPClient\_CCS). Check connection details and other entries in the file.

```
<?xml version="1.0" encoding="UTF-8" ?>
  - <cfg>
    <OnLoader>0</OnLoader>
  - <Connect>
    <Address>192.123.0.199</Address>
    <UserName>CCS</UserName>
    <Port>21</Port>
    <Passive>0</Passive>
    <Filter>rdy</Filter>
    <Delay>5000</Delay>
    <PassWord>Ãíâ´±@îæ</PassWord>
  </Connect>
    - <ServerEnv>
      <Ftpcmd>ftp_cmd</Ftpcmd>
      <Download>RESQA</Download>
      <Upload>TOQA</Upload>
      <Cmd_Down>cmd\LdrPut</Cmd_Down>
      <Cmd_Up>cmd\LdrGet</Cmd_Up>
    </ServerEnv>
    - <ClientEnv>

<Ftpcmd>***DATA***\RemoteQAMngr\clients\rom\FTP_CMD</Ftpcmd>

<Download>***DATA***\RemoteQAMngr\clients\rom\RESQA</Download>

<Upload>***DATA***\RemoteQAMngr\clients\rom\TOQA</Upload>

<Cmd_Down>***DATA***\RemoteQAMngr\clients\rom\CMD\MngrGet</Cmd_Down>
```

```

<Cmd_Up>***DATA***\RemoteQAMngr\clients\rom\CMD\MngrPut</Cmd_Up>
    </ClientEnv>
    <ServiceName>DWFTPClient_CCS</ServiceName>
</cfg>

```

### OnLoader

Use 0 if the FTP client is on the manager side, 1 if it is on the client side (default should be 1, the server shall be on the manager side, the client on the loader, but it works both ways)

### Connect

Connection info

### Address

Where to connect.

### UserName

User name.

### Port

Port.

### Passive

Transfer mode (0 – active, 1 passive).

### Filter

Do not transfer files with this extension (shall contain rdy – these are our control files).

### Delay

How often to check FTP command from server side (milliseconds).

### Password

Encrypted password – till we will have an gui for this, the initial config has the <UCPassWord> tag and unencrypted password – first time the client starts, check for this, encrypt the password and store it under <PassWord> tag.

Please note: <UCPassWord> and </UCPassWord> are case sensitive and have to be written like that.

In the \*-GLBL-ini in the section [SERVICE]

```
USER=
```

```
PASSWORD=
```

the user has to be set with domain (<Domain>\<user>) permissions.

If this was not done before and an error appears in Win2008 to enter this via dWCtrlDlg, you have to enter this unencrypted manually in the INI.

### ServerEnv

Paths on the server environment (relatives to home directory – usually shall be set one of the MainClientPath, depends where the FTP-client is located):

- Ftpcmd

where the client finds FTP commands on the server (will monitor permanently this folder by default)

- Download

where the client will download the zip data from (f.e. if it is on the loader side here we should have TOQA)

- Upload

where the client will upload the zip data (f.e. if it is on the loader side here we should have RESQA)

- Cmd\_Down

where the client will download commands from (f.e. if it is on the loader side here we should have cmd\MngrPut) – is about commands between RQAManger and RQALoader

- Cmd\_Up

where the client will upload the commands (f.e. if it is on the loader side here we should have cmd\MngrGet) – is about commands between RQAManger and RQALoader

### ClientEnv

- Fpcmd

where the client finds FTP commands on its side (will monitor permanently this folder by default)

- Download

where the client will download the zip data (its local path) (f.e. if it is on the loader side here we should have local MainClientPath\TOQA)

- Upload

where the data is that the client will upload it to the server (f.e. if it is on the loader side here we should have local MainClientPath\RESQA)

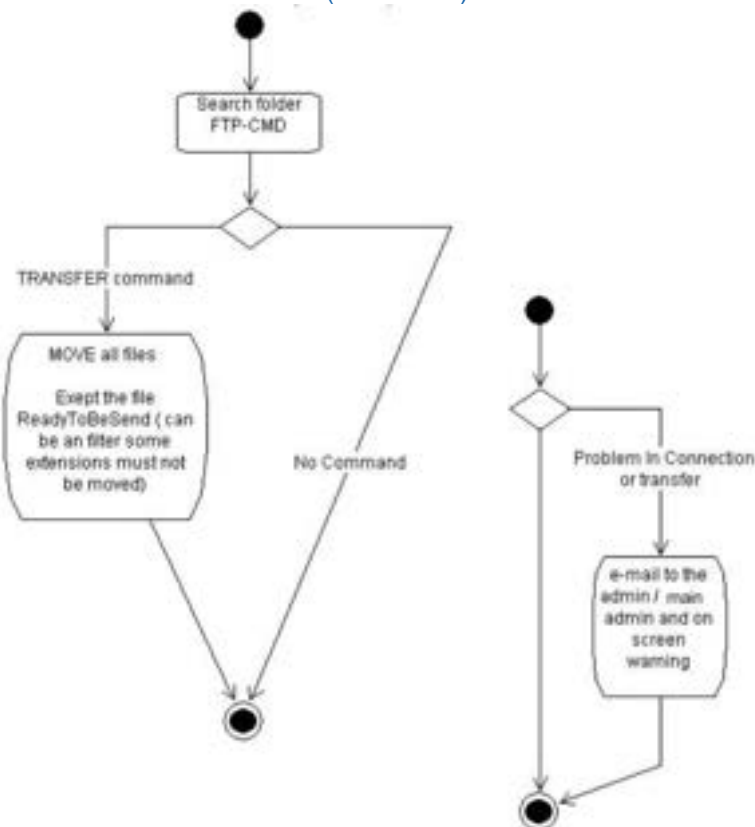
- Cmd\_Down

where the client will download the commands (f.e. if it is on the loader side here we should have local MainClientPath\CMD\LdrGet)

- Cmd\_Up

where the commands are to be uploaded (f.e. if it is on the loader side here we should have local MainClientPath\CMD\LdrPut)

#### 27.2.3.4.2 FTP Transfer Service (FTP Client)





#### 27.2.3.4.3 Features of DWFTPClient service config\qa\FTPConfig.xml

The default settings should be adapted by need or removed where not needed.

```
<DownloadUploadRatio Down="1" Up="2"/>
```

This node helps by defining a transfer ratio of 1 downloads to 2 uploads.

```
<ResetTime>36000000</ResetTime>
```

This node helps by restating the DWFTPClient if it is on idle, otherwise DWFTPClient restarts automatically at 12 hours, values for this node are set in ms (currently configured to reset after 10 hours).

```
<TransferThreadsNo>10</TransferThreadsNo>
```

Sets the number of transfer threads.

#### 27.2.3.4.4 FTPService - Multiple Remote Locations

To support multiple remote locations, FTPService will run as different instances either on the same or different computers.

To configure multiple instances you need to follow the steps listed below:

- Create a symbolic link by command line: `mklink dwftpclient-clientname.exe dwftpclient.exe`
- Register the new service by calling "sc create DWFTPClient-clientname binpath= c:\program files\ccs" gmbh\bin\dwsrv\dwftpclient-clientname.exe
- Configure LogOn information in services.msc
- Create configuration file clientname-FTPConfig.xml

For consistency, it may be useful to unregister the default ftp service and use the mechanism as shown above for each location.

NOTE: There is a space after "binpath=". Please don't ignore!

#### 27.2.3.4.5 dwftpclient.xml

If there are several slave systems are configured each **dwftpclient.xml** has to be registered.

This feature is developed for a customer's needs.

To register multiple instances of docWizz FTP client use the helper program regftp.exe. This program is developed by CCS (not a windows command).

##### Usage:

regftp.exe path clientname

##### Sample:

```
regftp.exe "C:\Program Files\docWizz\bin\docWizz\dwftpclient.exe" ROM
```

The registration program is not part of default delivery

For configuration, create customer1-FTPConfig.xml, customer2-FTPConfig.xml and customer3-FTPConfig.xml. The default DWFTPClient service should be de-installed. (delete DWFTPCLIENT).

Finally you need to set start and log-on information for those services.

#### 27.2.3.4.6 RQA-Sharing.xml for more than one RemoteQA Location

Now three locations can be set in RemoteQA. Use the file "RQA-Sharing.xml" in project directory:

Example:

```
<CLIENTS>
  <CLIENT NAME="ROM" SHARE="100"/>
  <CLIENT NAME="DDD" SHARE="0"/>
  <CLIENT NAME="PLANMAN" SHARE="0"/>
</CLIENTS>
```

Depending upon the relationship of the Shares the documents are distributed on the different Clients. A binding is regarded thereby as 1 document, i.e. all partial documents go to the same RQA station. This is automatically called with each autoimport. The up-to-date assigned RQAClient is located then in the data base Main table.

Technically said: The Client, that is queried to receive a document next by the new Script command:

```
"getnextRQAClient $project"
and assigned by
"$doc -info rqaclient $RQAClient"
```

**100** = 100 % of the data go to this remote location.

**0** = 0 % of the data go to this remote location.

#### 27.2.3.4.7 Task FTPPrio

Whenever a priority is set for a document in pool, it is be used as well for FTP transfer.

#### Usage of FTPPrio task

Configure the task in docWizz-DW.ini:

```
[PROCESS]
TASKS= FTPPRIO
```

and there the task itself:

```
[FTPPRIO.TASK]
PROC=CreateFTPPrioritiesFile
DELAY=300
TIME=10:00:00
TIMETYPE=0
VALIDATION_PROC=NoOfDocsToDoPrioFileForFtp
LOCKED=1
SCRIPT_FILE=***DATA***\SCRIPT\task\createPriorityImpFileForFtp.tcl
```

Now configure the task to be executed by services in **docWizz-dwsrv.ini**, add the task to an existing task, or add a new task under the TASKS section with FTPPRIO

Go to **..\config\QA\** folder. Open **ftp\_prio\_config.xml**.

- Now, the **ftp\_prio\_config.xml** can be configured by the desired prioritization and on the desired client .
- This list will be written to:

**\\server\docWizz\DW\_65\RemoteQAMngr\clients\<clientname>\FTP\_CMD.**

The file name is "Priorities.IMP" and contains only the DocID, one per line.

### Example file

`\\config\QA\ftp_prio_config.xml:`

```
<CONFIG Enabled="1">
  <Client name ="qa1">
    <!--path of the ftp cmd and path of the folder of the transfered folders-->
    <Ftpcmd>***DATA***\RemoteQAMngr\clients\qa1\FTP_CMD</Ftpcmd>
    <Upload>***DATA***\RemoteQAMngr\clients\qa1\TOQA</Upload>

    <!--possible values for order are:
    order="doc_prio project_cfg step"
    order="project_cfg step doc_prio"
    order="doc_prio step project_cfg"
    order="step project_cfg doc_prio"-->

    <prio order="doc_prio project_cfg step">
      <!--project cfg with prio. if empty this will be ignored-->
      <project_cfg></project_cfg>
      <!--jobs with prio. if empty this will be ignored. Hint: configure small
      zips to be transferred first-->
      <step>VerifyPageNumbers          VerifyPagesHierarchy          VerifyHierarchy
      VerifyLayoutElements VerifyPageFrames</step>
      <!--documents will be also sorted by pool prio-->
    </prio>
  </Client>
</CONFIG>
```

The prio list is not just sorted by priority, but by configuration.

## 27.2.3.5 Setting up FTP, FTPS and SFTP

### Setting up FTP

#### Server side (Filezilla):

Same as before.

#### Client side (FTPClient):

For backward compatibility, existing configurations are still valid.

For future configurations is recommended to add `<FTPType>` node under `<Connection>`; this node will contain FTP as value.

### Example of FTP connection configuration:

<Connect>

```
<FTPType>FTP</FTPType>
<Address>192.168.0.198</Address>
<Port>21</Port>
<Passive>0</Passive>
<Filter>rdy</Filter>
<Delay>5000</Delay>
<UserName>MyUser</UserName>
<UCPassWord>MyClearTextPw</UCPassWord>
```

</Connect>

For the FTP connection the other nodes in <Connect> did not change.

### Setting up FTPS

#### Server side (FileZilla):

Open Settings in FileZilla Server Interface and go to FTP over TLS settings.

Edit->Settings->FTP over TLS settings.



Enable FTP over TLS support(FTPS).

If there are requirements to disallow FTP connection then the check box **Disallow plain unencrypted FTP** should be checked.

Generate new certificate and complete the certificate form.



After generating a certificate add a key password to the certificate and hit the OK button.

#### Client side (FTPClient):

In the `<Connect>` node add a new node called **FTPTType** and set the value to FTPS:

```
<Connect>
  <FTPTType>FTPS</FTPTType>
  <Address>192.168.0.198</Address>
  <Port>21</Port>
  <Passive>0</Passive>
  <Filter>rdy</Filter>
  <Delay>5000</Delay>
  <UserName>MyUser</UserName>
  <UCPassWord>MyClearTextPw</UCPassWord>
</Connect>
```

The other nodes under `<Connect>` cheep the same meaning as for the FTP connection.

#### Setting up SFTP

##### Server side:

For the SFTP connection the setting up part is a bit more complicated and currently we do not have a SFTP Server application selected.

This part will be updated in the future.

List of possible SFTP Server:

- OpenSSH – free, has no user interface and is supported on Windows and Linux
- Solarwinds SFTP/SCP Server – has a free version but only supports files up to 4Gb
- Cerberus Professional
- Bitvise SSH Server
- CompleteFTP

##### Client side (FTPClient):

To change the connection type to SFTP, set in the `<FTPTType>` node from the FTPClient.xml the value SFTP. If the `<FTPTType>` node is missing add it under `<Connection>`.

If you are using SFTP connection you also need to add, if missing, in the `<Connection>` node, the following:

<KeyType>The ssh key type(Supported key types: OpenSsh,Putty)</KeyType>

<KeyPath>Path to the client private key</KeyPath>

The KeyType node refers to the program/format used to generate the users private key.

When generating the users private key, a password for the private key is set. This password needs to be set in the <UCPassWord> node.

Configuration Example:

```
<Connect>
  <FTPType>SFTP</FTPType>
  <Address>127.0.0.1</Address>
  <Port>22</Port>
  <Passive>0</Passive>
  <Filter>rdy</Filter>
  <Delay>5000</Delay>
  <UserName>bogdan</UserName>
  <UCPassWord>myPrivateKeyPass</UCPassWord>
  <KeyType>Putty</KeyType>
  <KeyPath>H:\clientprkey\clientprkey.ppk</KeyPath>
</Connect>
```

The meaning of the other nodes under <Connection> remains unchanged.

## 27.2.4 Loader System

There are three folders for data/commands exchange between loader and main system: **IN**, **OUT** and **COMMANDS**.

On **IN** folder the loader system will monitor the commands folder for any order from the main system. When commands are available, it will take the first one (based on file name) and execute it. The commands usually contain info regarding the **IN** data. This will contain info like doc ID, file list and main database entry and task to be done – add new document, update existing document, delete document. The commands will be processed (if update/new doc):

- The loader will search for **IN/ID** folder and will add a new document in pool if this does not exist (not found in the translation table).
- Lock the document
- Will store a translation table (an xml file) that contains a pair oldID/newID.
- The image files are copied into pool with all necessary changes. An improvement would be to convert highly compressed images (jp2 f.e.) into Tiff files for a faster load on correction. Being a time consuming operation would be a good idea to use all the computers (something similar with compression on main machine). The Loader service will coordinate this, but will share the work on multiple computers. This is needed only if the images needs long time for loading
- Set status READY\_FOR\_QA
- Unlock document

If the command will be "delete" then simply look on the translation table and delete the corresponding document

The loader will also monitor the pool on a separate thread for documents in state REMOTE\_QA\_DONE. If one document like this is found then:

- all the necessary data to be returned is collected (usually ID.XML and analyze zips),
- the necessary changes are performed (f.e. change ID.XML name, change zips names)

a new folder **OUT/oldID** is created and necessary data is added there (archive with no compression)

While data is sent back to main machine this is deleted. When all data was moved from **OUT/ID** folder a notification file will be added (FTP client job).

The loader will send back a command that contains information about uploaded data (command files will have same naming convention like for Main machine commands) to main machine and then will delete the **OUT/oldID** folder.

When OCR was done by mistake on RQA side, it is reset automatically to "On" when returning to main system.

#### 27.2.4.1 RemoteQALoaderCfg.xml

Register the file **\*\*\*CONFIG\*\*\*\QA\RemoteQALoaderCfg.xml**. In this file normally nothing has to be changed.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Config>

<DWServiceCommandPath>***DATA***\RemoteQALdr\LdrCmd\Loader</DWServiceCommandPath>

<TransitionTableFileName>***DATA***\RemoteQALdr\TransitionTable.xml</TransitionTableFileName>
    <MainClientPath>***DATA***\RemoteQALdr\RemoteData</MainClientPath>
    <LogPath>***DATA***\WORK\LogRemoteQALdr</LogPath>
- <Managers>
    <Manager name="" />
</Managers>
</Config>
```

#### DWServiceCommandPath

Where the RemoteQALoader will post orders for DWSrv.

#### TransitionTableFileName

Where the transition table will be stored (for safety reasons all the info from this transition table is duplicated in pool, so that any moment can be recovered).

#### Doc ID and Transition Table

The documents do not have the same ID at both locations. They have a different ID when they are in Remote QA side and a different ID when they are back in main system. A Transition table is maintained in \RemoteQALdr\TransitionTable\. This gives both the corresponding IDs of the documents.

#### MainClientPath

The root for the RemoteQALoader. Below this (not configurable) we have:

##### Cmd

with subfolders LdrGet and LdrPut – here the commands are prepared for RemoteQAManager (LdrPut) and received from RemoteQAManager (LdrGet)

##### TOQA

here the zips are received from RemoteQAManager

##### RESQA

here the zips to be sent to RemoteQAManager are stored

#### **FTP\_CMD**

here FTP commands are sent

#### **LogPath**

the RemoteQALoader will log all the activity, in 4 text files, one for each thread

### **Setup**

Using "modify" in setup and adding QALoader, it gets installed in new separate folder **..ldwsrv\**. It is not installed in an existing folder because all working services are installed in a separate path for better maintenance.

### **Tip**

If a RQA location gets files from different RQA Main systems, then all Managers have to be configured in the RemoteQALoaderCfg.xml:

```
<Managers>
  <Manager name="WSC-03"/>
  <Manager name="WSC-06"/>
</Managers>
```

Only one service per time can be active on one machine, if not the following error message will appear:

```
RemoteQALoader <MachineName>xxxx</MachineName> not corresponding with machine
on which loader is registered.
```

When loader is not configured on a machine, this error will be displayed in event viewer, because you might not know that machine name in configuration is not the same as the machine on which DWRRemoteQALoader is registered.

When empty, the RemoteQALoader behaves just like the field contains the correct value. Empty value of <MachineName> field is used for compatibility with old configuration versions.

### **27.2.5 Script for filtering out finalQA Documents**

There the documents are set to QA status after return from Z-RST step if they are from e.g. PLANMAN (so 100% of those ) and 1% of the other documents.

```
proc KBNL-DoFinalQA { doc } {
  set dWID [$doc -info id]
  set jobName [$doc -info job]
  message [format "DoFinalQA: id='%s'; jobName='%s'" $dWID $jobName]
  if { $jobName == "VerifyHierarchy" } {
    set rQAClientName [$doc -info "rqaclient"]
    set oldStatus [$doc -info "status"]
    set
      counterQAName [format
"***DATA***\WORK\LogRemoteQAMngr\RQA_review_%s" $rQAClientName]
    set counterQAValue [uniquenum $counterQAName]
    message [format "DoFinalQA: doc='%s'; oldStatus='%s';
RQAClient='%s'; counterQAValue='%s'" [$doc -info id] $oldStatus
$rQAClientName $counterQAValue]
    if { ($rQAClientName == "PLANMAN" || $rQAClientName == "DDDBTB" ||
$counterQAValue % 100 == 0) && $oldStatus == "W" } {
      $doc -info job "VerifyHierarchy"
      $doc -info stopatjob "VerifyHierarchy"
```



```

$doc -info status "QA"
$doc -data COMMENT [format "%s\\u000d\\u000a Selected for RandomQA
(%s)" [$doc -data COMMENT] $counterQAValue]
## write the comment field to disc
$doc -save
}
} else {
## only docs in Z-RST and which are sent to export
}
}
}

```

### 27.2.6 Error on FTP upload (Known issue)



FTPtransfer	29.09.2009 08:36:13	LastError: Failed with error 12031: (null) )
FTPtransfer	29.09.2009 08:36:13	LastError: Failed with error 12031: (null) )

The errorcode "12031" (see above in screenshot) has the meaning  
 ERROR\_INTERNET\_CONNECTION\_RESET - the connection with the server has been reset.

Problem could be caused if ISP had an upload protection for flood, and each time it passed the upload limit the internet connection stopped for some minutes.

As a solution you can limit your upload from firewall to the upload limit given by ISP.

Changes on configuration f.e. 1:1 ratio of up- and downloads.

By the short download transfers it seems to keep the NAT router opened by having activities in between. Then there should be no delays of documents any more yet.

## 27.3 Maintenance Remote System

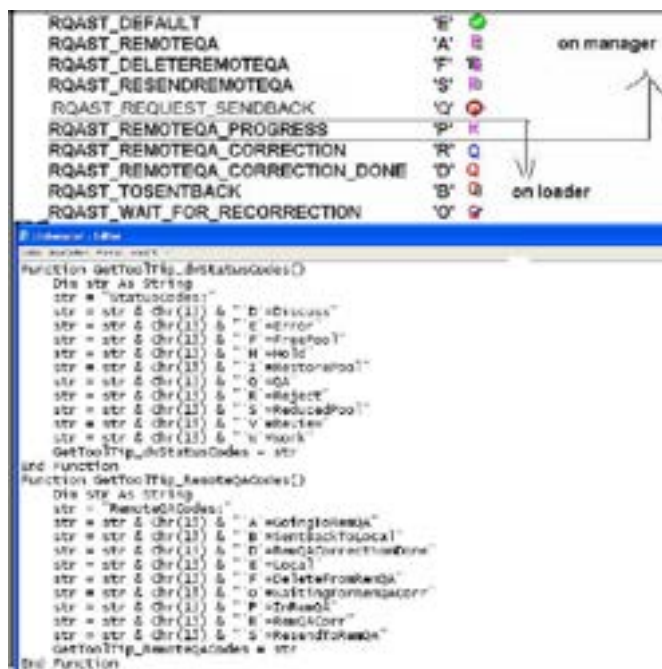
At the remote system, it is necessary to perform the below checks.

Go to docWizz application, login as admin.

Open menu File - Document Pool.

Here you can check the status of the documents in process.

See [docWizz UserManual](#) for details.



Check in the docWizz Control Center that the service 'DWRremoteQALoader' is active (green color). If not, please start the service immediately. Also check in services.msc if the loader is started.

Check the folder "ResQA". If the 'DWRremoteQALoader' is not active, then this folder will have some documents. This means those documents are not sent to the main system.

Similarly check the 'ToQA' folder. This will list the documents arrived from the main system and waiting to be loaded in the local docWizz Pool.

Frequently checking these folders (in addition to the check in docWizz Control Center) is necessary, as sometimes the status of the service is not immediately reflected in the graphical view of the docWizz Control Center.

Performing the above steps at a regular interval should be implemented as a policy at the remote QA environment, to avoid documents getting accumulated at one end.

## 27.4 Disable/Enable functions

Configuration file to disable some functions per project either in RQA or in general.

Description for project specific settings:

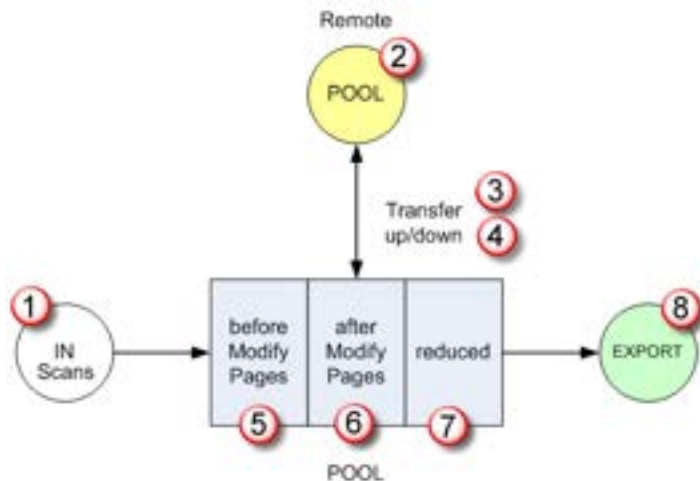
```
...\Project-cfg\<projectname>\<projectname>-setting.xml
```

```
<PROJECT-SETTINGS>
  <!-- Parameter ENABLED has three possible values: 0 - disabled; 1 -
  enabled; 2 - disabled in RQA, enabled on manager only -->
  <EDITING>
    <FEATURE NAME="AllowDeletePage" ENABLED="2"/>
    <FEATURE NAME="AllowModifyPagesAfterVPN" ENABLED="2"/>
    <FEATURE NAME="AllowInsertPageAfterScan" ENABLED="1"/>
    <FEATURE NAME="AllowChangeIssueStart" ENABLED="2"/>
    <FEATURE NAME="AllowChangeLanguage" ENABLED="2"/>
    <FEATURE NAME="ComputeTimeConsumingRejectsInUI" ENABLED="0"/>
  </EDITING>
  <EDITINGTASK>
    <FEATURE NAME="VerifyStructure" ENABLED="0"/>
    <FEATURE NAME="ComputeMissingOCR" ENABLED="0"/>
    <FEATURE NAME="VerifyOCR" ENABLED="0"/>
    <FEATURE NAME="Typing" ENABLED="0"/>
    <FEATURE NAME="ReTyping" ENABLED="0"/>
    <FEATURE NAME="CompareKeying" ENABLED="0"/>
    <FEATURE NAME="VerifyMetadata" ENABLED="0"/>
    <FEATURE NAME="VerifyClipping" ENABLED="0"/>
    <FEATURE NAME="FinalQA" ENABLED="0"/>
    <FEATURE NAME="Frame" ENABLED="0"/>
    <FEATURE NAME="Attach" ENABLED="0"/>
  </EDITINGTASK>
  <PROCESSING>
    <FEATURE NAME="AddInImagesOnBackupAndRemove" ENABLED="1"/>
    <FEATURE NAME="FreePoolDataOnBackupAndRemove" ENABLED="1"/>
  </PROCESSING>
  <POOL>
    <FEATURE NAME="NoCompression" ENABLED="0"/>
    <FEATURE NAME="LowCompression" ENABLED="0"/>
    <FEATURE NAME="AllowCleanPoolWithoutOriginImage" ENABLED="1"/>
  </POOL>
  <ITEMTRACKING>
    <RETURNWHENREACHED>BuildHierarchy</RETURNWHENREACHED>
  </ITEMTRACKING>
  <DOUBLEKEYING>
    <ZONETYPE ENABLED="1" like="1">Headline Author</ZONETYPE>
    <TYPING_CONDITION ENABLED="0" THRESHOLD="500"
MINCHARS="5"></TYPING_CONDITION>
  </DOUBLEKEYING>
  <RQA>
    <LIST NAME="Create_RQA_BW" ENABLED="1" SELECT="b">
```

```
<a>Threshold</a>
<b>EnhanceWithMask</b>
<c>CustomConversionFile</c>
<d>FastLMSDT</d>
</LIST>
</RQA>
</PROJECT-SETTINGS>
```

## 27.5 Storage and data transfer calculation

Storage and Data Transfer Calculation for docWizz Remote QA conversion projects. (Excel Sheet for calculation available).



### (1) IN (Scans)

Scanned data are in the **IN** data folder. This data has to be kept until:

- at least until successful export
- until delivery to customer
- until acceptance from customer (depends on project configuration)
- until the record retention ends (depends on project configuration)

### (2) Pool (on Remote Side)

- In Verify Page Frames images are transferred first.
- After Modify Pages the images are transferred again in Verify Layout Elements in reduced image size.

Layout Analysis files per interactive job are added. Layout Analysis files remain until the export is successfully done. They are deleted then or are again automatically transferred if required.

### (3) Upload

- Upload means: from production system to the remote QA system.
- There is no time range, only done once

High compressed images are transferred (compared to point 5 (Pool) where images are less compressed but faster in access).

### (4) Download

- After a correction step ends the appropriate Layout Analysis file is transferred (from the current job).

### (5) Before MP

- Small size, can be disregarded.

### (6) After MP

- Large size, have to be kept until successful export. This means the time range between Modify Pages and Export. It lasts so long the automatic and manual correction is processed.

## (7) Reduced

- If Export is successful all POOL data (that could be restored automatically) will be deleted. Layout analysis data remains.

## (8) Export

- As only complete batches are delivered, the export waits until all data belonging to a batch are processed.
- Project dependant record retention for exports.

### 27.5.1 Typical File Sizes per Page

Examples for typical file sizes for newspapers or books per page.

		Newspaper	Books
(1)	IN	20 - 100 MB	3 - 10 MB
(2)	Pool (Remote)	5 - 8 MB	2 MB
(3)	Upload	0,5 MB	0,2 MB
(4)	Download	0,2 MB	0,05 MB
(5)	Before MP	./.	./.
(6)	After MP	15 - 60 MB	2 - 8 MB
(7)	Reduced	1,0 MB	0,2 MB
(8)	Export	project dependant	project dependant

### 27.5.2 RQA images compression ratio

For RQA images the compression ratio is defined in **RemoteQAManager.xml**.

One for C-PC (just for frame setting needed), the second for all further jobs (Z-RZ, Z-RPS, Z-RI, Z-RST).

Typical setting for Newspaper (original scans):

```
<CompressionRateVPF>200</CompressionRateVPF>  
<CompressionRate>100</CompressionRate>
```

Typical setting for Newspaper (microfilm scans):

```
<CompressionRateVPF>50</CompressionRateVPF>  
<CompressionRate>25</CompressionRate>
```

The compression is calculated based on the source image size.

Source image file size is 25 MB => RQA image will be 250 KB

This will cause issues, in case the source images are of inconsistent types and different compressions will be provided.

E.g. TIF uncompressed mixed with TIF LZW compressed.

For the uncompressed images the RQA images will be very good, for the LZW compressed images the RQA images will be in comparison very small and probably not readable.

To take instead of file size the real memory size of the images negative values can be configured in the configuration.

```
<CompressionRateVPF>-50</CompressionRateVPF>  
<CompressionRate>-25</CompressionRate>
```

This will lead all RQA images really have the same compression based on the source image.

## 28 Final QA Definition

docWizz is able to deal with final QA mechanism. In this scenario, we decide between normal users and QA users. A normal user does correction as usual.

Once he has finished a document and selects "process" and everything is fine, he cannot access the document another time.

Document will get the status of final QA to be performed.

Then a final QA user may open the document and checks if everything is fine. If yes, he may start processing. If not, he does manually a reject and the document will be sent back to normal users.

To define a QA user, you need to call system configuration and select the user tab. There you can assign the final QA right to any of the defined users or you may add a new user.

The final QA mechanism can be defined per project and job. You need to create an **xml** file called **projectname-qa.xml**. It shall contain a list of jobs in which final QA shall be performed:

```
<?xml version="1.0" encoding="UTF-8"?>
  <QA>
    <STEP name="VerifyPagesHierarchy"/>
    <STEP name="VerifyHierarchy"/>
  </QA>
```

In this sample, final QA will be performed in the jobs VerifyPagesHierarchy and VerifyHierarchy.

Please be aware of the renamed steps and tasks and the "old" names in scripts and configuration files.



## 29 dWOCRRemote

docWizz RemoteOCR allows a centralized sharing of OCR licences for the whole docWizz customer environment.

It can be used in combination of LocalOCR. the regular way, where each client machine has its own registered OCR licence.

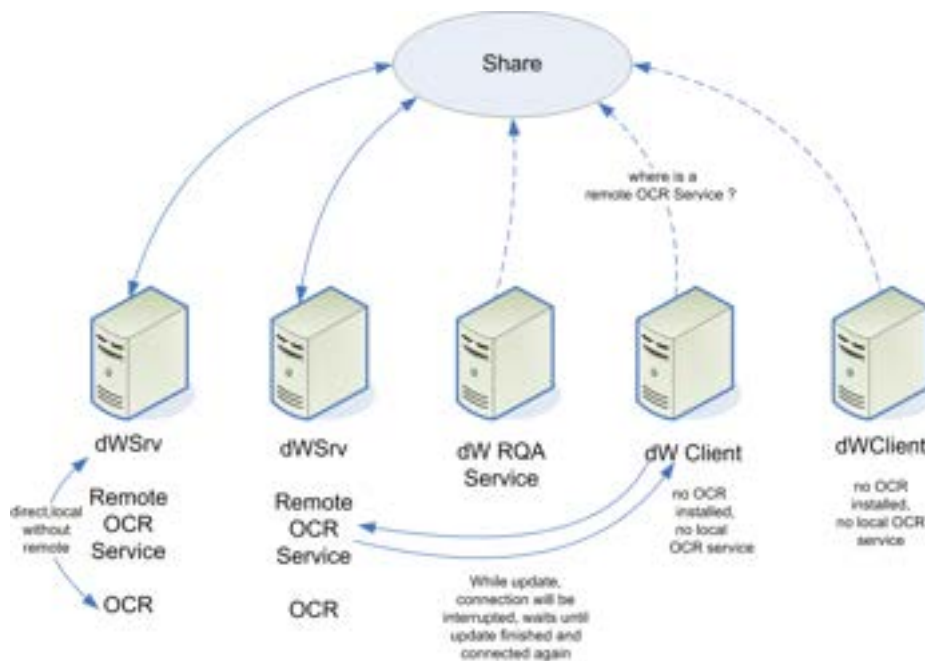
Clients don't need an OCR licence for each client machine, e.g.: 50 machines can have 2 (Remote) OCR licences, instead of 50 (Local) OCR licences.

Special OCR (e.g. Gothic), which is often more costly in purchase, can be served on demand and not kept on stock for all.

Basically, the demand on OCR licenses can be handled economically.

### 29.1 Overview

This image shows an example of a possible constellation:



Each application, which can use OCR, reads from custom-docWizz-glbl.ini, if RemoteOCR is enabled and if local OCR should be looked up primary.

It requests from RemoteOCR service an OCR instance.

RemoteOCR starts „dwocrrun.exe" on the RemoteOCR computer:

- One dwocrrun.exe is started with the service as a kind of „manager".
- Then one dwocrrun.exe per requested instance is started.

Example:

If you have 3 dwsrv and 1 DW executed, 5 dwocrrun are shown in task manager.

## 29.2 Setup a configuration for using remote OCR

There are three important parameters in the custom-docWizz-glbl.ini in the [DWOCR] section:

`REMOTEOCR = 0 or 1`

Disable or Enable the use of RemoteOCR

`EXCLUDEFROMREMOTEOCR = <computername>`

Computers to be excluded to use RemoteOCR; comma separated list

`NOTTRYLOCAL = 0 or 1` Disable or Enable .

If set to disabled, computer will look for local OCR first and secondly for RemoteOCR.

To setup a configuration for using the remote OCR you must perform following steps:

1.) change the following key in the `\\[server]\docWizz\DW_65\config\docWizz-glbl.ini`

`[DWOCR]`

`RemoteOCR=1`

2.) Register the Service DWOCRRemote on each computer, where the OCR engine should run. The Service must run under an account, on which it can access the configuration. Set the service to autostart.

To register a service you can use the Set log on functionality in the docWizz Control Center or use command line command: `<path\servicename.exe -service>`

At first start/restart of DWCC the custom-docWizz-dwsrv.ini will be extended with entry

`[SERVICES]`

`..`

`Computername:DWOCRREMOTE=`

### Functionality of the Remote OCR

The remote OCR service looks on the system for installed and licensed OCR engines.

It setup following directories on configured path

`\OCRSERVICE`

`\SERVICESTATUS\OCRRUN\cmd`

For each valid engine it generates a locked status file in the path `\OCRSERVICE\`

The client is looking in the path `\OCRSERVICE` for available engines.

They request an engine with a file in the directory `\SERVICESTATUS\OCRRUN\cmd`

After connection to the engine, the engine generates a locked file in the path `\OCRSERVICE\`

If local an engine is executed, it is preferred. On switching from Gothic (Fraktur) to Antiqua engine would not be unloaded. On starting Remote OCR engine error reporting is added.

The log files for the Remote OCR is like: `QS_NAME.224.FR9.antiqua.1984.$LF`

The first digit set is the PID of the DWCL; the second digit set is the PID of DWOCR on the remoteOCR machine. (PID: Process ID in task manager, is one of the possible fields.)

## Additional files that need to be handled/registered

Only

```
regsvr32 DWOCRMMMSG.DLL  
DWOCRREMOTE.exe /service
```

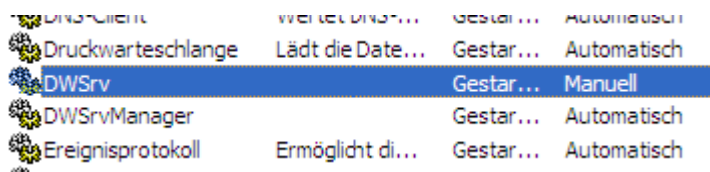
are needed. Together with the setting in the -glbl.ini and assigning service user it should run.

Note: If DPN (Detect Page Numbers) should be processed on a machine then OCR must run locally there!

## 29.3 How to register services

### Register services

- Go to „C:\Program files\ccs gmbh\bin\docWizz“
- Execute there the files reg\_DWService.bat and reg-pdf.bat with double click, so that they are registered in the system
- Change the properties of the DWSrv service in the popup window "Local services":



Right mouse click to properties, go to the log on tab

activate there "This account" and add the domain user ( e.g. docwizz@dw.miko.lib.helsinki.fi) with user password - activate - accept - OK

- Change the properties of the DWSrvManager service in the popup window "Local services":

Right mouse click to properties and select "Startup status" to Automatic

Go to the log on tab and activate there "This account" and add the domain user ( e.g. docwizz@dw.miko.lib.helsinki.fi) with user password

Go to Recovery tab and select for the 3 dropdown boxes "Restart the Service" activate - accept - OK

- This is necessary to control all services in the system with the docWizz Control Dialog console

Close the services window and the ms-dos window

## 29.4 Remote OCR configuration

On "Remote OCR Main computer" (where the actual FineReader files are and the OCR license(s) are installed)

- **custom-docWizz-glbl.ini:**

```
[DWOCR]
; Only for dev/debug use.. maybe possibility to access different computer
for ocr..
PATH=
; Parameter 0=off ; 1=on
REMOTEOCR= 1
```

**PATH:** only used for development or debugging.

The DWOCR in normal case must have its engine in the subdirectories of the installed programs. So it could be that the engine is installed multiple times. To avoid this program path `\DWOCRSERVICE` is possible. In this path a common OCR could be installed for all installed programs in the binary path.

- **custom-docWizz-dwsrv.ini**

```
[SERVICES]
; same syntax as for dWSrv, etc. Should be done before start of DWCC
computername:DWOCRREMOTE=A
```

On Remote OCR Loader computer(s) (which does not have an OCR license and need to request OCR from Remote OCR Main computer)

- Register "DWOCRRemote.exe" as a service, which has access to the configuration. Set service to "autostart".

### Other files (or folders) and Notes

- DWOCRRUN.exe in task manager of Remote OCR Main computer

When DWOCRRemote.exe is started, there will be one DWOCRRUN.exe started with it.

It is like a "manager" for the others.

Each application which is capable of requesting OCR will start a DWOCRRUN.exe in task manager of Remote OCR Main computer

#### Example:

Remote OCR Main computer has dWSrv, dWSrv2 and two Remote OCR Loader computer have a started DWCL.exe

==> 4 DWOCRRUN.exe plus 1 DWOCRRUN.exe (for "managing" the others).

- Lock files to see, if DW clients are connected or have requested OCR.

```
..\WORK\OCRSERVICE\<ocrengine>\<computername>.PID1.<ocrengine>.PID2.$lf
```

#### Example:

```
..\WORK\OCRSERVICE\FineReader 9\QS_WEIDNER.224.FR9.antiqua.1984.$LF
```

```
..\WORK\OCRSERVICE\FineReader 9 Gothic\QS_WEIDNER.224.FR9.gothic.1986.$LF
```

The first digit set is the PID of the DWCL; the second digit set is the PID of DWOCRRUN on the remoteOCR machine.

(PID: Process ID in task manager, is one of the possible fields.)

- All loaders need the same configuration file. Multiple RemoteOCR can be configured. Put them all in the custom-docWizz-dwsrv.ini.

### Functionality of the Remote OCR

The remote OCR service looks on the system for installed and licensed OCR engines.

The clients are looking in the path `<WORKDIR>\OCRSERVICE` for available engines.

They request an engine with a file in the directory:

```
<WORKDIR>\SERVICESTATUS\<computername>\OCRRUN\cmd
```

After connection to the engine, the engine generates a locked file in the path

```
<WORKDIR>\OCRSERVICE\<enginename>
```

### Remote OCR but run OCR locally

For some reason (here OCR using Z-RPS), pipe connection might be unavailable or other networking limitations may appear. In those cases you need to exclude dedicated computers from using RemoteOCR. Instead they shall use local OCR.

Configuration parameter:

```
[DWOCR]
EXCLUDEFROMREMOTEOCR=Hostname;Computername
```

Those computers listed here will not join remoteOCR but just run OCR locally.

### Use local installed OCR (NOTTRYLOCAL)

```
[DWOCR]
PATH=
REMOTEOCR=1
EXCLUDEFROMREMOTEOCR=
NOTTRYLOCAL=0
```

0 = use local installed OCR

1 = use RemoteOCR (can be on the same computer)

`nottrylocal = 0 or 1`, dwsrv/dw uses the "local" (easier) interface of installed licenses.

The dwocrrun will not be handled and tracked via the remote OCR service. Each installed license needs a remoteocr.exe service registered on the machine where the license is, which also means, that license will be available to all computers requesting remote OCR. All license types are still supported (even if a system setup should not have more than one in "real life").

### DEBUG option

```
[DWOCR]
EXCLUDEFROMREMOTEOCR=
NOTTRYLOCAL=0
REMOTEOCR=1
DEBUGOCR=1
```

In order to debug the OCR process and log more information you may set `DEBUGOCR = 1`. This is only recommended when you have any issues with OCR engines and need to understand what is happening in detail. It should remain disabled normally, due to big amount of error logs generated and should be enabled only for debug purposes.

When this flag is enable you can see logs that helps you trace the logic of engine selection and as well a set of errors that usually appear on module initialization due to licensing. The errors that appear due to invalid binaries are logged no matter if debug is enabled or not.

Here is a list of errors that are disabled normally on initialization, but can be displayed if `DEBUGOCR` is set to 1:

- if only CCS dll is available in the engine folder (f.e. `DWOCR11.dll`) - logged if `DEBUGOCR=1` ("directory did not contain engine data")
- if CCS dll cannot be loaded and the folder contains other files (like engine binaries) - logged if `DEBUGOCR=1` ("Could not load <fileName.dll>. Windows error code %d; Windows message: %s")
- if CCS dll doesn't have all necessary functions - logged if `DEBUGOCR=1` ("Did not find all the required methods to initialize the engine (<fileName.dll>).")
- if OCR engine initialization fails:
  - because of license (missing, expired, licensing service not running ...) - logged if `DEBUGOCR=1`
  - because of other reasons (wrong engine version, invalid engine files...) - logged no matter the value of `DEBUGOCR`

## Refresh rate setting

```
[DWOCR]
REMOTEOCR=0
NOTTRYLOCAL=0
REMOTEWORKREFRESH=-1
EXCLUDEFROMREMOTEOCR=
DEBUGOCR=0
OCRCHARWARNLIMITANTIQUA=1000
OCRCHARWARNLIMITGOTHIC=1000
TESSDATADIR=
```

Because some remote OCR engines can be started after the `dWSrv/docWizz` starts (or restarted at one moment), the `RemoteOCR` clients (`dWSrv/docWizz`) need to know what engines are available in the system. This is happening by refreshing the information at a certain time interval.

`REMOTEWORKREFRESH` defines this interval in seconds; -1 is used for default (300 seconds - 5 minutes). In general this setting must be adapted to environment needs, since refresh operation is a demanding one in terms of resources needs.

In case you have, in one environment, only one engine type, with same language package, is recommended to have a longer period for refresh, since a missing engine at one time does not influence the entire environment behavior.

In case you have unique particular engines (f.e one single Abbyy FineReader license including CJK package for Asia languages and many other engines without this package), then is important to have a smaller period of refresh, since a `Remote OCR` client needs to know as soon as possible such a resource is again available (in these case in recommended to have 30/60 seconds refresh time).

## 29.5 DWOCRRUN.exe

### What is the concept behind the many DWOCRRUN.exe?

When starting **docWizz** on the main workstation, maybe 1 or 2 DWOCRRUN.exe are started in task manager. On second workstation there are more DWOCRRUN.exe started. Why are there so many?

Each application that requires OCR has its own DWOCRRUN.exe. Running DWSRV,DWSRV2,DWSRV3,DWSRV4,DWCLIENT for example will open in total 5 times DWOCRRUN. On RemoteOCR Services it might be even more.

### Should related subthreads be killed, if master thread is killed?

When DWOCRRemote is executed, it starts a dwocrrun.exe for managing. When DWOCRRemote is killed in task manager, the sub task "dwocrrun.exe" is still executed. Should it not check, if parent process is available and otherwise kill itself?

If you only stop the service some dwocrrun will stay active. DWOCRRemote is only a broker between the clients, which need an OCR engine, and the OCR engines. Stopping the DWOCRRemote will only stop its own dwocrrun, but not the dwocrrun of other clients.

## 29.6 DWOCRREMOTE auto-update functionality

Auto-Update for bin/OCRService folder.

The functionality is only enabled on remote OCR configurations. It is designed on the standard distribution, where the OCR service is stored in a separate path.

### Used Files

```
***DATA***\WORK\OCRSERVICE\computername.DWOCRREMOTE.UPD (used as indicator,
that a update is running.)
***BIN***\dwocrremotedirectory\UpdateOCR.cmd (batch file for update)
```

### DwOcrRemote Service working

- Timer is looking for changes in installation folder.
- Creating dwocrremote.upd for signals start.
- Closing all dwocrrun.exe.
- Creating batch file.
- Creating scheduler task and end service.
- On start, cleanup DWOCRREMOTE.UPD file and batch file.

### DWCL and DWSRV working

On start of dwocrrun.exe look for \*.DWOCRREMOTE.UPD files, not older the 10 minutes. If found wait while exist.

## 29.7 Remote OCR for multiple environments

This is helpful especially for non-regular installations. Then different users with different systems may use the same OCR Server and license.

A special configuration path for RemoteOCR communication will be used then instead of default work sub folders.

### Configuration docWizz-glbl.ini

```
[DWOCR]
```

```
REMOTEWORk=<path>
```

On <path> you insert the "WORK"-directory of the configuration, where the DWOCRRemote is serving.

```
[DWOCR]
```

```
Path=<binpath>
```

Working only on local configuration as additional search path for the dwocrrun.exe

### Is there a Maintenance Status for DWOCRRemote?

RemoteOCR does not support maintenance at all. And it must not support maintenance. The reason is, that in maintenance mode, regular working with dwclient is still possible (should be, we can't block a complete production environment). But dwclient is using OCR remote. So OCR remote must be excluded from maintenance.

## 29.8 Stop, Kill or Shutdown commands

### STOP

Will stop as it was only the remote OCR service. So the established engines will run and serve also after this. RemoteOCR service waits with stop until last remote OCR processes have ended. It verifies if the OCR processes are in idle mode to be able to close the OCR processes then even the dW service / dW client is still kept active. Behaviour now is similar to dWSrv - the icon changes to "Stopping" until service is stopped.

### KILL

It kills also ALL engines on the OCR computer.

### SHUTDOWN

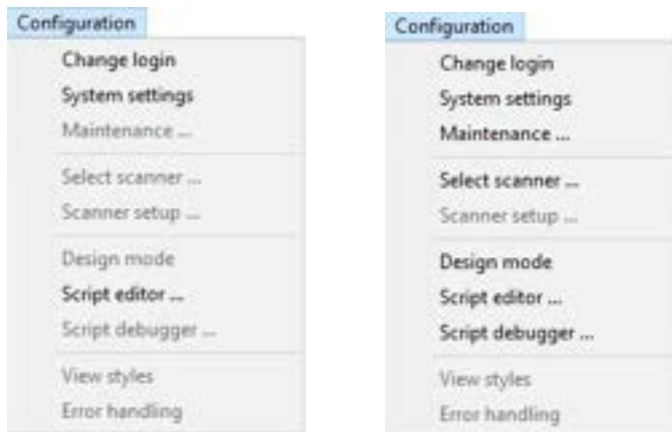
Is working like a STOP except it signals to the remote called engines (not to the local) and waits, that they are stopping after a timeout it also terminates them.

This operation immediately kills the OCR processes and then stop the RemoteOCR service. In any case the RemoteOCR service should not stop until the OCR processes are ended (except the local ones which were directly connected). If the kill operation returns a failure, an exclamation mark outlines problem on stop operation switching with "Stopping" icon.



## 30 Configuration Menu

### Configuration Menu



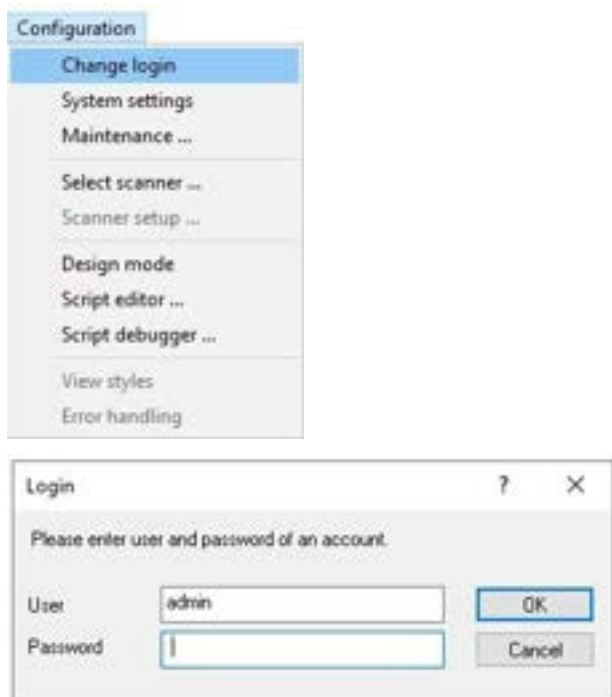
Depending on the current job the **Configuration** shows additional features.

The **Configuration** menu allows you to configure all the settings you need for the smooth operation of the program.

If you do not have administrator permissions, the **Configuration** menu offers only the **Change login** and **System** entry.

### 30.1 Change Login

**Change Login** allows another user (the administrator, f.e. to log in during a docWizz session without requiring the current user to exit the program).



**Default Login:** Uses Windows login as user name, normal access rights.

## 30.2 System configuration

If Stationname in registry does not match the current computer name (this happens if the computer name was changed after installing docWizz), the user is prompted now as "The station name in registry does not match the current computer name!\ Please contact your system administrator. An administrator may set ComputerName in registry to allow mismatch."

If you want to keep the mismatch (work on old local configuration) you need to set the value of key "ComputerName" to the current computer name in registry.

Open the system configuration dialog box of docWizz with the **System function** of the **Configuration** menu. You use this dialog box to configure the system and the individual application components. The system configuration area of the program is password-protected and can only be accessed by the system administrator.

### 30.2.1 System Card

The **System Configuration** dialog box is used to specify all the fundamental settings for the correct functioning of the program.

Using the standard Windows convention, it is organized like a file card box. Clicking a tab opens the respective card with the setting options it provides. Work through the tabs from left to right.

You specify the name of the configuration directory and the initialization file in the system tab's **Configuration** section. This is the heart of the system, and you should therefore keep a close eye on it and back it up at regular intervals.

In the **System** section, you specify the **Default Resolution for Images**. Here you can also specify the **Resolutions** for graphic zones (gray/color). Enter your settings in dpi (dots per inch).

During the screen representation of the source pages, a low-resolution reference of the image is shown. In this, the zones are then selected and clipped. Only when arranging the target page the high-resolution picture is taken from the server, which is then used on the target page according to the data. The resolution of the low-resolution reference picture (which is there only for the view) is tuned up here.

The resolution of images produced by docWizz if by other side no other resolution is given (e.g. with scanning or also with the export) usually uses a standard resolution. In case of this resolution, it should be always considered that the result is further-processable with the OCR.

For black/white-scanned images the recommended standard value, is 300 dpi. If you wish to scan grayscale or colored images, the file size of the page images will increase considerably. Therefore, we recommend reducing the graphic zone resolution. If you reduce the resolution down to 150 dpi, you will not see a quality problem on the colored or gray printings because laser printer does raster the images.

#### **Ask before exiting program**

Means when you exit the program you are prompted to respond to the query "Do you really want to exit docWizz?".

#### **Use LZW as default**

LZW is a compression procedure for color and grayscale images.

### 30.2.2 Pool Card

Settings pertaining to the storage and organization of the scanned source pages are made here:

#### Path



In the section **Path**, enter the **Main Directory** where the source pages are to be stored. Here you can only select a created "Path" from the "Paths card".

#### Database



docWizz automatically creates a database for the administration of the data relating to these source pages.

You can determine the type of database access in the field **Type** by selecting an entry from the drop down list. **Microsoft Access** and **SQL-Server:** through the OLE- DB Driver. If you select an SQL Server, you will need to enter the names of the **Server** and the **Database** in the two fields that are now activated.

#### Image Files

In the next section **Image Files**, you can activate the checkbox **Store color images as JPEG**, in which case, the scanned source page images are stored in the JPG-Format, which is considerably smaller than the usual TIFF-Format, at the cost of quality.

If grayscale and color images are processed, file sizes can become very large, depending on the size of the source page and its color resolution. The transfer of such files can affect the efficiency of the network. To avoid this, such files can be stored locally on the hard drive of the scan station.

Activate the check box **Distributed storage of big image files**, if you would like to create a folder on your local hard disk for the storage of these big files. Enter **Local path** and **Sharing** (the share name by which you will be sharing this folder in the network).

When this is done, all source page images that are in grayscale or color will automatically be stored in this folder. The database that administers these source page images will have a corresponding entry, so that other clipping stations in the network will be able to access these images. As the clipping stations will seldom need to load the entire source page, the network load will be reduced considerably.

#### Thumbnail




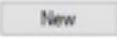
You can activate the checkbox **Save** in the section **Thumbnail**, if you would like to save thumbnails of the source pages additionally with a low resolution, which you can specify in the fields **Width** and **Height** (no. of pixels).

### 30.2.3 User Card

The various users of the system, their permissions, and passwords are administered with the help of this card. The system administrator can assign names and passwords for different users by selecting the options here:

The system administrator can also authorize or deny access rights by activating and deactivating the check boxes in front of the **Administrator, Call Configure System Dialog, Change OCR Settings, Design Index Dialogs, Do Training, Edit Indexing Lists, Maintenance, Select Scanner** and **Statistics** options.

The button  deletes the highlighted user name from the list.

The button  opens a window where a new user can be added to the existing list:



For the case, **Use Windows login name** is activated, a user under just this name will be added. Of course, this user must already exist.

Password administration is also carried out here, including that of the administrator. Passwords are not case-sensitive and the length is not restricted.

As the existence of an administrator is mandatory, the first user in the list is always the administrator, even if this user is not given administrator permissions. You can and should change the name and password of the administrator! You should also create at least one user who does not have administrator permissions.

The standard configuration comes with two users: the "Admin" with administrator permissions and the "Standard" user with restricted permissions. If no password has been entered for the user, docWizz will not prompt the user for one when the user logs in. This option is useful if one does not want to enter one's name and password every time one logs in. However, it is advisable for the administrator to use a password, so that a user does not inadvertently make changes in the system configuration.

#### Default permissions:

<b>Administrator</b>	Administrator, Edit Indexing Lists, Final QA
<b>Configuration</b>	Change OCR Settings, Delete Documents, Design Index Dialogs, Do training, Edit Indexing Lists, Final QA, Maintenance, Select Scanner, Statistics
<b>Maintenance</b>	Delete Documents, Edit Indexing Lists, Final QA, Maintenance, Statistics
<b>QA</b>	Delete Documents, Final QA
<b>Standard</b>	Edit Indexing Lists
<b>[Username]</b>	individual permissions

### 30.2.4 Document Types Card

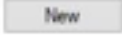
The **Document Types** let you define different types of data fields (records) for your documents:

Each document type describes a set of different data fields. Generally, only one document type is needed for a docWizz configuration.

#### Data fields

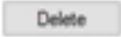
Click the button  in the **Document Types** tab to open the **Data Fields** box:



The  button lets you add a new data field.

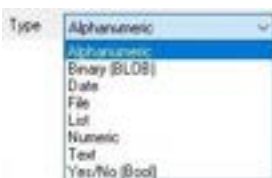


Select the name of the data field you want to edit from the list located on the left of the box. Enter or alter the name of the field in the **Name** input field.

Click the  button to delete the selected data field.

#### Type

The **Type** input field provides a pull-down list of available field types that are described below. When you open the list, you can make your choice from the following types.



##### **Alphanumeric**

Any combination of letters and numbers may be entered.

##### **Binary (BLOB)**

This field accepts only Binary Large Object Data.

##### **Date**

Only date format is allowed in this field.

##### **List**

Alphanumerical values can be saved in the field. This is for fields that allow multiple selections and not for fields that allow only one value that is selected from a selection list.

##### **Numeric**

Only numeric entries are allowed in this field.

##### **Text**

This field can contain editable text as well as references to images. For fields without image references you can also use the alphanumeric field type.

### Yes/No (Bool)

This field allows only mutually exclusive values such as On/Off, Yes/No, or 0/1.

### Length

A small rectangular input field with the text "Length" on the left and the number "80" inside.

Specifies the maximum length of a field; the entry 0 specifies that the field can be of any length.

### Min.

A small rectangular input field with the text "Min." on the left and the number "0" inside.

Specifies a field's minimum length. You can also specify whether the field must be filled in or not regardless of the minimum setting.

### Boolean


In some field types, the length of the field does not make sense, for example **Boolean**. If the field must be filled in the system will not allow further processing as long as the field remains empty.

Note: Although the field types can always be changed, it is recommended that a clear structure for data fields be designed before entering them here. It is advisable to avoid changing field names, as the changes will have to be carried out in all the places where this field is used! In addition, TCL scripts use field names.

### Formats

Clicking the  button in the **Document Types** you can configure date formats, lists, and other parameters.

The defined formats can be used for export as well as for the elements on clipping target pages.


A dialog box titled "Format" with a close button (X) in the top right. On the left is a list box containing several field types: "CATLONG", "DATEFORMAT", "FLOATINGTEXT", "FLOATINGTEXT\_P", "LISTBAR", "SEMICOLON", and "SPACEDLIST". "CATLONG" is selected. To the right of the list box are four input fields: "Name" (containing "CATLONG"), "Type" (a pull-down menu showing "Structured list"), "Format" (containing "%2" with a hint "(eg. %1 (%2)/v/n)"), "List" (a pull-down menu), and "Delimiter" (containing "/n"). At the bottom are four buttons: "New", "Delete", "OK", and "Cancel".

Select the field you want to format from the selection list. To edit a format simply select it from the list located on the left side of the box.

The name of the format is shown in the **Name** window.

The **Type** input field provides a pull-down list of available field types you open by clicking the button, and which are described below. When you select a format, a dedicated dialog mask opens.

Click the  button to add a new format.

Click the  button to delete the selected format.

### 30.2.5 Tasks Card

The **Tasks** card shows the different jobs for processing documents.

### 30.2.6 Lists Card

In the **Lists** card, you can define lists, edit and administer them. These lists are attached to **List Boxes** or **Combo Boxes** in the dialog boxes of various jobs.

The window area on the left let you choose an existing list which can be deleted, using the **Delete** button or you can edit lists, using the **Edit** button.

The list editor currently integrated in this program can edit only file lists with either 1 or 2 columns. If your list has more columns, you will need to use an external text editor.

#### Name

To define a new list, enter the name of the list.

#### Type

Select the type of list you require.



The available types are:

##### File

Here you have to define Hierarchy and File.

##### Indirect ODBC Table

Here you have to define Hierarchy, Connect and Table.

##### ODBC Table

Here you have to define Hierarchy, Connect and Table.

#### Columns

Specify how many columns the list has.



By activating the With Alias Entries check box you can access to list entries via an alias name.

Activate the Sorted check box to specify that the list should always be automatically sorted.

#### Hierarchy

You use the **Hierarchy** drop-down menu to specify the hierarchical structure that should be applied to the list. Hierarchical lists are always generated with two columns. The first column specifies the allocation of an entry, and the second column contains the entry.



The selection list offers the following options:

**without**

There is no hierarchy, the list is flat.

**by Table Column**

The hierarchical level corresponds to the numerical value of the first column of the list.

**by Length of Entry**

The length of an entry in the first column determines the hierarchical level; this is decisive for hierarchical levels because the longer the entry, the lower the hierarchical level.

**by puncture**

Dots entered in the first column indicate the place value in the hierarchical level.

**by Column with Parent ID**

This is a list with three columns. The list must be edited by means of an external editor. The columns are separated by a blank space. The first column consists of a number or "ID" which increases consecutively in each row. The second column contains the text entry. When this text consists of more than one word, the entry must be enclosed in quotation marks. The third column contains nothing if the entry belongs to the top level of hierarchy. In other cases, the third column has a number, which corresponds to the ID of its higher-level or "parent" entry.

**Examples**

**by Table Column:**

1 Politics  
2 "Domestic Politics"  
2 "External Affairs"  
1 Culture  
2 "Music of the 21st century"  
2 "Classical Music"  
1 Sport  
2 Football  
2 Handball

**by Length of Entry:**

P Politics  
P-I "Domestic Politics"  
P-A "External Affairs"  
KL Culture  
KL-M "Music of the 21st Century"  
KL-2 "Classical Music"  
S Sport  
S-1 Football  
S-2 Handball

**by Column with Parent ID:**

1 Politics  
2 "Domestic Politics" 1  
3 "External Affairs" 1  
4 Culture  
5 "Music of the 21st century" 4  
6 "Classical Music" 4  
7 Sport  
8 Football 7  
9 Handball 7

**by dots:**

1. Politics  
1.1. "Domestic Politics"  
1.2. "External Affairs"  
2. Culture  
2.1. "Music of the 21st century"  
2.2. "Classical Music"  
3. Sport  
3.1. Football  
3.2. Handball

**ODBC**

If you choose the Type "ODBC Table" or "Indirect ODBC Table" in the **Lists** window, some additional fields appear. You can select the type of **Hierarchy** from the attached list.

The **Data Source**, which is defined under ODBC Data Sources, must be entered in the **Connect** field.



The table that is to be used is entered in the **Table** field.

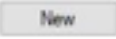
**Data Fields:** Used only for ODBC lists. The button opens a window in which you can select the fields you want to use from the database.

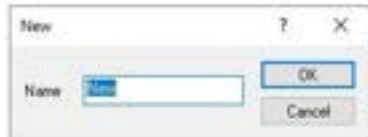
You can add or delete fields using the **Add Item**, **Add all**, **Remove Item** or **Remove all** buttons.

You can also define a **Sort by** field.

### 30.2.7 ODBC-Connection Card

Use the ODBC-Connection card to manage ODBC connections that are used for ODBC lists for example:

You can define a new connection with the  button.



In the **Name** field, you enter a name that will be used internally in the docWizz configuration.

The actual **ODBC Data Source** must be selected from a list of existing **ODBC Data Source** definitions with the button **Select**.

If this has not been defined, it can be defined now, in the **Select Data Source** window.

In the **Computer Data Source** card, you can mark an existing definition or create a new one with the **New** button. This procedure can also be carried out in the Windows Control Panel. For further information, consult the Windows System Documentation.

### 30.2.8 Languages Card

You use this tab to select the menu language of your system user interface as well as the languages you want to use for processing your documents:

#### User Interface

Displays the language available for the User Interface.

#### Config Language

Displays the language of the executed configuration. If the system is running multi-language configuration, the language of the original configuration is displayed here.

#### Available

The languages listed are the ones installed on the system, based on the installed OCR module.

#### Selected

The only languages listed are the ones supported by the OCR module you are using.

### 30.2.9 Path Card


#### Name

You choose a name and associate a path with it.

#### Paths

Simply changing the path associated with a name in the Paths list incorporate any subsequent changes in the system. This spares you the effort of having to specify the new path throughout the entire system.

#### New

Click the button  to specify a new path.

#### Delete

Click the button  to delete a selected path.

#### System default

Specify the following:

##### Work Files

Specify where you want docWizz to store the temporary files that are created during your work. Here you can only select a created "Path" from the "Paths" card.

##### Lists

Specify the default directory for the user-specific list files. In the input field, you specify the directory where you want to save the training files.

Note: It is recommended that path macros be used for these paths.

### Path Macro

In the TCL Scripts there are macros for certain directories e.g. `***PRJ_CFG***`.

Together with the `expandpath ***PRJ_CFG***/NDNP/...` command you receive then the correct path, which is defined in the `<customer>-gblb.ini`.

#### Macro examples:

`***BIN***` -directory in which docWizz was started (where the \*.exe files and the \*.dll are stored)

`***TEMP***` - temporary directory (of the operating system), usually C:\Documents and Settings\<user>\Local settings\ Temp"

These two paths cannot be defined in the Ini (as they do not appear there), but become determined and installed when docWizz runs.

In the menu **Configuration - System - Register paths** these macros cannot be edited also. They are shown for completeness and check of the paths.

Example of the path could be different on your system.

### 30.2.10 Dialog Boxes Card

The dialog box is an integral part of the user interface. This area can be configured to suit the individual requirements of the user.

You use the **Dialog Boxes** card to define dialog boxes:


Dialog boxes are used to prepare fields for adding structured information to scanned documents. If this information is available in the document itself, it can be read into a field using the OCR process.

The name of the selected dialog box is shown in the **Name** input field. When you create a new dialog box, you must enter its name here.

In the **Title** input field, you can enter the text of the title that should be displayed in the dialog box. This may be the same as the name of the dialog box.

In the **Doc. Type** field, you can enter a reference to the document types.

#### Properties

 Open dialog box with an index tab system for defining dialog box properties.

#### 30.2.10.1 New button



For creating a new dialog box.

#### 30.2.10.2 Delete button



For deleting the dialog box that is highlighted in the list.

#### 30.2.10.3 Edit button



To open the Input Controls dialog box and an index tab system for editing dialog boxes.

#### Input Controls

This is where you define input and control elements for each dialog box:





For adding new control elements:



**Data fields** opens a selection box **New Dialog Control** from which you can configure a new control element. A list of available data fields is displayed for selection.

Select the desired category with the mouse or use the buttons **Button**, **Static Text**, **Image**, or **Graphics**.

Only those fields, which have been defined in the Data fields list, but have not already been positioned in the dialog box, are displayed in the list for selection.



Delete a highlighted field from the list. The dialog box is represented as a file card system with four to five tabs.

The register card **List Display** is only available for control objects that are defined as type "list".

The register card **Tab-Control** is only available for control objects that are defined as type "Tab-Control".

You can highlight any of these input controls in the list and assign attributes to them:



#### 30.2.10.3.1 General



You can use the **General** card to define fundamental input object characteristics.

##### Type

Specify the basic characteristic of the input control field. Only those options supported by the respective field type are displayed for selection.

##### Background

Specify the background color for an object.

##### Shortcut key

Define shortcuts to jump at objects. Applied on buttons the according script is executed. You can specify whether the shortcut is selected using the [Shift], [Ctrl] or [Alt] key. Note that you can only use key combinations that are not currently in use elsewhere in the program. A drop down list displays the available keys.

### 30.2.10.3.2Position

Use the **Position Card** to specify the position and dimensions of the input object in the dialog box:



All values of position intervals are specified in lines (the vertical distance between lines, one line being approximately equivalent to the height of a character) or characters (the horizontal width of a number other than "1"; "0" and "2" to "9" have approximately the same width). This permits a uniform definition independent of the screen resolution and font being currently used.

Left

**Left:** Specifies the distance of field from left margin (without label).

Top

**Top:** Specifies the distance of field from top margin (without label).

Width

**Width:** Specifies the width of the field.

Height

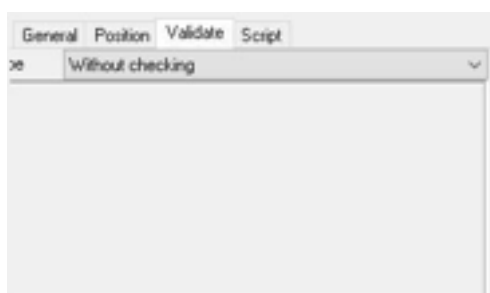
**Height:** Specifies the height of the field; note that in the case of drop-down lists, this is the height of the list.

Title  
☒ Left  
☐ Top

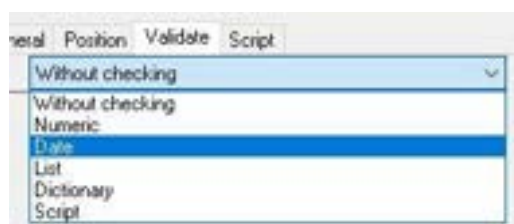
**Title:** Could be set **Left** or **Top**.

### 30.2.10.3.3Validate

You use the **Validate** card to specify how entries in the highlighted field should be verified:

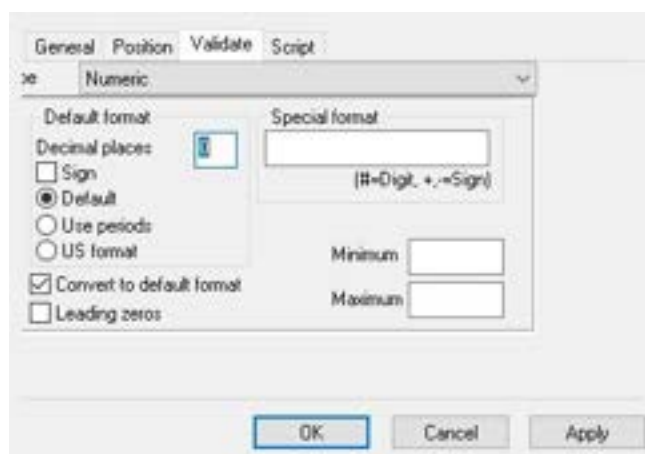


Use the **Type** drop-down list to assign the verification method you want. Selecting a list entry opens the respective dialog box:



**Without checking:** Contents are not checked.

**Numeric:** Specifies the number formats you want to use:



#### Decimal Places

Specifies the number of decimal places.

#### Sign

Allows negative numbers to be used.

#### Default

Displays numbers in the default format, for example, 12345,67

#### Use periods

Places a period at every 1,000th position.

#### US-Format

Uses a decimal period instead of a comma.

#### Convert to default format

Converts numbers to the default number format automatically.

#### Leading Zeroes

Allows the use of leading zeros, such as 01 instead of 1. Missing places in smaller numbers are filled with leading zeros (for example 01; 001; 0001).

#### Special Format

Describes number formatting using the placeholder "#" for number, "+" for positive signs, and "-" for negative signs

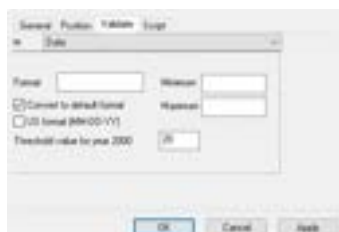
#### Minimum and Maximum

Determines the permitted minimum or maximum for an entered value.

## Date



You can specify the formats for the **Date** fields in this section:



### Format

Specifies the desired **Date** format here.

### Convert to default format

Converts the entered or recognized date to the specified format. If the option is not selected, the date is verified but left in the original format.

### US-Format

Formats the date according to the US convention of month-day- year.

### Threshold value for

Specifies the threshold for converting the year format from two places to 4 places. If the threshold is set to 20, year entries of 1-20 are converted to 20XX, and all entries above 20 are converted to 19XX. For example, 1.5.06 is interpreted as 1.5.2006, and 1.3.34 as 1.3.1934.

### Minimum

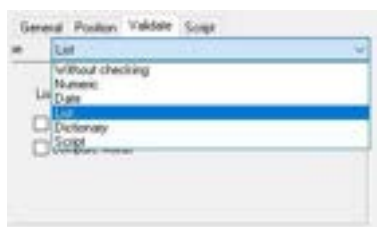
Specifies the earliest allowed date.

### Maximum

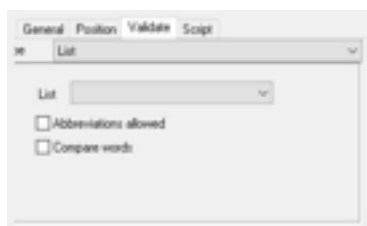
Specifies the latest allowed date.

In both cases, it is also possible to enter relative values like -7 or +0 for example. This example means that the maximum entry is the current date and the minimum is a date from last week.

## List



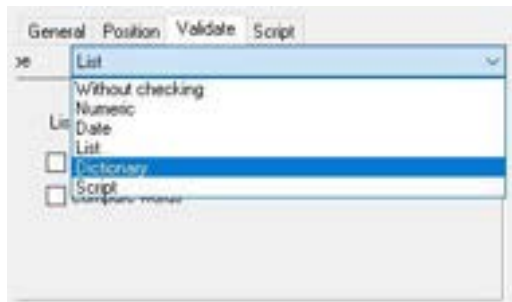
List entries are compared to a specified list. You can specify whether abbreviations should be permitted (**Abbreviations Allowed**) or a word-based comparison (**Compare Words**) should take place.



If **Compare Words** is selected, when the comparison process starts the system will not try to find the whole text in the list, but instead individual words. Every time a match is made the corresponding entry is selected.

In this way names, for instance, can be filtered from a text field.

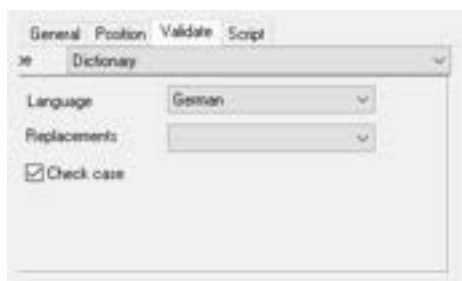
## Dictionary



**Dictionaries Field** contents are checked against dictionaries.

Select the desired **Language** and if want to make a distinction between upper and lower case letters activate the **Check Case** check box. If no language is selected, the language selected for the current document is used.

You can also select the list for automatic string **Replacements**, if this feature is activated.



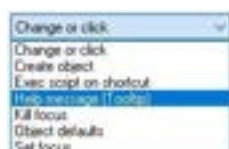
The specified script must return one of the following values: 0 = Error; 1 = OK, 2 = Value was corrected

### 30.2.10.3.4Script

Using the **Script** card, you can link an event to a specific script, which describes and controls the process to be executed:



In the **Type** input field, select the event for which you want to enter a script. Scripts can be executed in the following events:





### Change or click

Called when the contents of a control field are modified or a button is clicked, for example after every entry in the field.

### Create object

Called when a control field is created for the first time.

### Help message (Tool tip)

When you enter a script here the return value of the last script command (not the return value of the script itself, of course) appears in a floating message window highlighted in yellow when the mouse pointer is left on the control element for a few seconds. For example, you can enter the following in the **Script** card of a field: set msg "Edit List". When you move the mouse pointer on this field, you will see the tool tip reading "Edit List".

### Kill focus

Called when you leave a control field.

### Object defaults

Called when a control field should be set to its default settings. Choose an event from the drop-down list in the **Type** field and enter the script for this action in the **Script** window.

### Set focus

Called when a control field is activated.

You can Run the script using the following buttons:

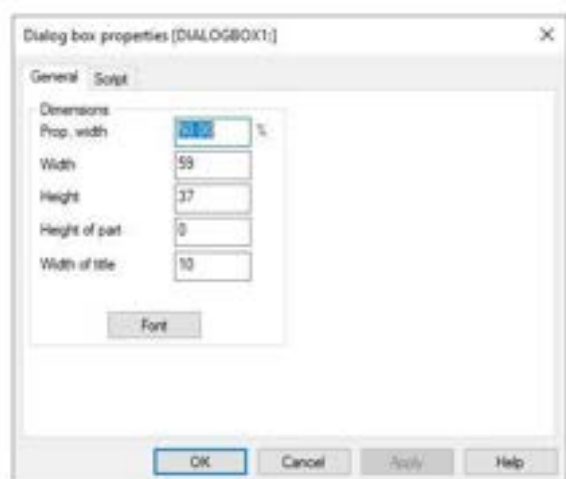
**All** - Executes the entire script for manual verification.

**Line** - Executes only the selected line or lines of the script marked as a block.

? (Help) - Displays an overview of the TCL commands.

#### 30.2.10.4 Properties of the Dialog Boxes

The **Properties** button in the **Dialog Boxes** card opens the dialog box **Properties** card with four tabs: **General** and **Script**.



## General

Use the **Dimensions** area to specify the dimensions of the dialog box:

You use the **Prop. width** input field to specify how wide the dialog box should be in proportion to the screen. Entering 50 here, for example, sets the width of the capture box to 50% of the screen width.

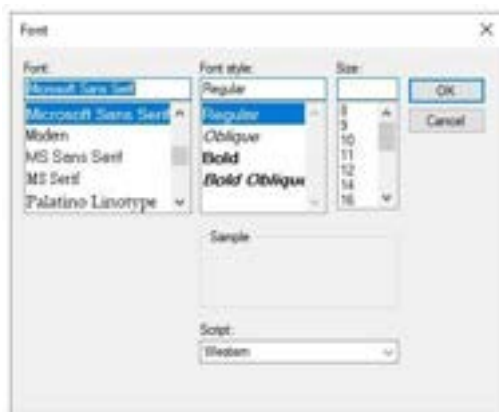
Use the **Width** input field to specify the width of the dialog box in characters. Entering 49 here, for example, means that the capture box should be 49 characters wide.

Use the **Height** input field to specify the height of the dialog box in lines.

You use the **Height of Part** input field to specify whether the dialog box should be capable of being divided, enlarged or reduced. The value 0 deactivates this function. Any other whole number specifies the height of the top (not collapsible) part of the dialog box in lines. The dialog box can be restored to its full size by clicking of buttons or with the function key.

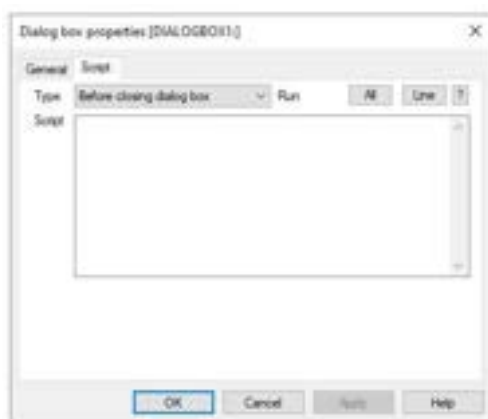
The input field **Width of Title** specifies the width for the label in characters. By default the label that appears to the left of the field. The entered value is only used for those elements that appear on the left edge of the box.

To select the **Font** type, size, and attributes for the legends in the dialog boxes, use the **Font** button to open the font selection window:



## Script

This script card is used to define the TCL-scripts that are associated with this particular job, to be executed at various stages of processing:



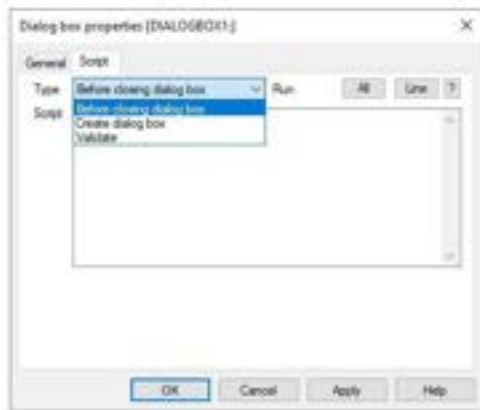
You can Run the script using the following buttons:

All Executes the entire script for manual verification.

Line Executes only the selected line or lines of the script marked as a block.

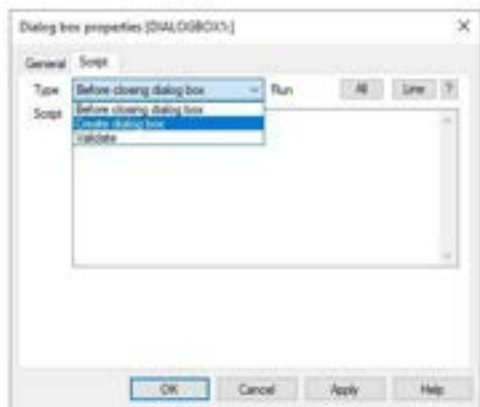
? (Help) Displays an overview of the TCL commands.

The events listed in the **Type** drop-down list specify the moment when the script entered in the window should be run:



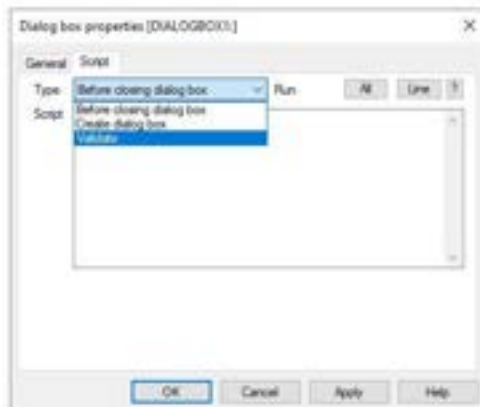
### Before closing dialog box

Script will be executed before closing of the dialog box.



### Create dialog box

When a new dialog box is created and input fields (particularly lists) can be initialized.



### Validate

When all entries are checked before processing begins. If the script returns a 0, the document cannot be processed.

### 30.2.11 Zone Classes Card

The various types of zones that are used during the process can be defined here:

A list of zone types, predefined for the standard configuration, is displayed on the right side in the Available field. It is possible to move classes to the **Disabled** field by the double arrows in the center of the card and vice versa. The disabled entries are not shown in docWizz selection.

With the help of the **Custom Class** dialog box that opens up when the button **Add Custom Class** is pressed, you can add your own zone classes:



You enter the **Name**, the **Color** and the **Key**, which you want to use to allocate this zone class to a marked zone. A list of available hot-keys is provided.

In the "Zone Classes" window, you can **Disable** certain zone classes by marking them in the **Available** list and moving them with the double-arrow keys to the left side of the window. These will not be available during the clipping process. They can be made available again by marking them and moving them back to the **Available** list.

### 30.2.12 Dictionaries Card

#### Path

Path to the dictionary. Here **\*\*\*DICT\*\*\*** where the \*.tri files will be stored.

#### Replacement Tables

Create new or edit existing replacement table.



Create, edit, copy or delete replacements.

### Spell checking

Spell checking option: **Replace before**  
Replacement generally done before dictionary check.

Spell checking option: **Replace on error**  
Replacement if not found by dictionary check.  
If then found in dictionary, the word will be replaced. If not found in dictionary - no replacement will be done.

### 30.2.13 Script Card

A script is a file that contains a number of commands written in the script language TCL (= "Tool Command Language"). A script gives the system instructions on how to perform certain operations in a particular order to achieve specific results. Scripts can be executed at various moments during processing, and can be associated with various jobs, fields etc.

#### Type: System script

In the **Script** card, you can enter system scripts that are executed automatically when the docWizz is started.

You can **Run** the script using the following buttons:

**All** Executes the entire script for manual verification.

**Line** Executes only the selected line or lines of the script marked as a block.

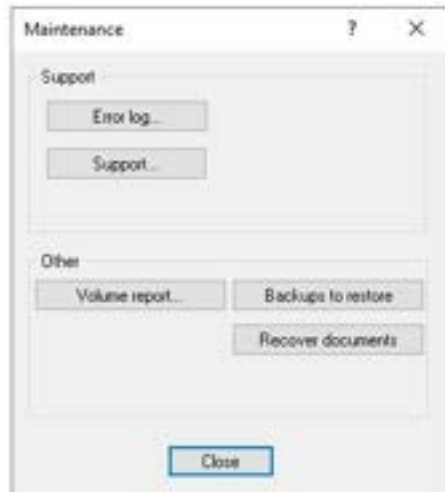
**? (Help)** Displays an overview of the TCL commands.

The script in the illustration consists of the command:

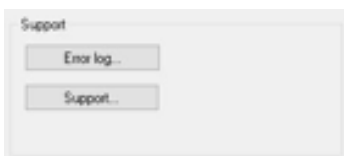
```
source [expandpath ***DATA***\script\docWizz_main.tcl]
```

This command does not execute any script, but compiles all the scripts in the script-file **docWizz\_main.tcl** and prepares them for execution at any point during the processing.

## 30.3 Maintenance



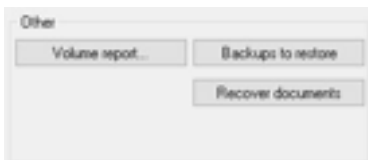
### Support



Click on **Error log...** to get the log details.

Click on **Support...** to switch to the Maintenance Client. See separate documentation "MTNClient".

### Other



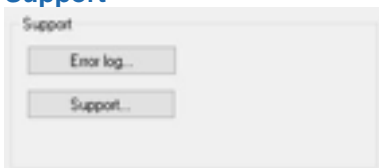
The **Volume Report** is an additional function in dW that creates a pdf file containing the total page counter and the number of pages of the selected month. A new dialog is opened where you can select the desired period of which you want to create the **Volume Report**. For more information please refer to "Volume Report" subchapter.

**Backup to restore** - see a list of documents in the restore queue.

**Recover documents** - recover documents and get the number of restored documents.

### 30.3.1 Support and Error log

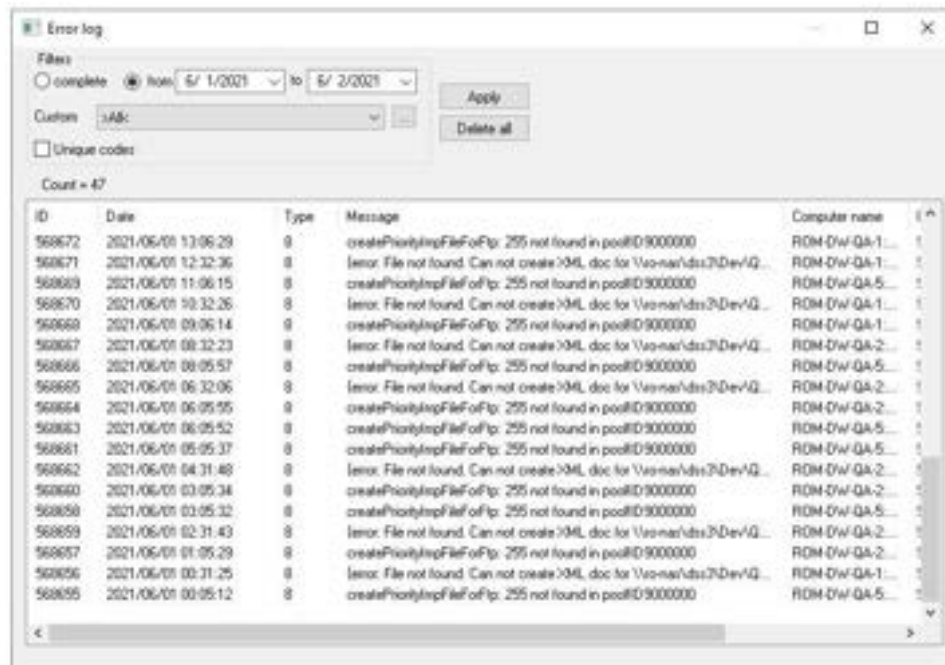
#### Support



#### Error Log

This function enables you to refer to the **Error Log** window that automatically lists any errors that have occurred during the current session. In this way, support staff and docWizz administrators have optimal support when looking for the cause of irregularities in the running of the program.

The **Support** button opens the **Maintenance Client** which is a module used for automatic problem reporting and customer support and problem solving. See description in the following chapters.



### 30.3.2 Volume Report

#### What's the VolumeReport?

The VolumeReport is an additional function in docWizz that creates an pdf file containing the total page counter and the number of pages of the selected month.

#### How to create the VolumeReport?

The VolumeReport is positioned at Configuration -> Maintenance. Inside this dialog you can choose "Volume Report...".

A new dialog is opened where you can select the desired period of which you want to create the VolumeReport.

It is selected the minimum date of the BatchResult table and generated all month until last month of current date.



After creating the report by click on the button, the location of the stored PDF-file will be shown in the text area below. The default location of the VolumeReports is **\*\*\*MAINTENANCE\*\*\***. You can change the location in the system configuration in the register "paths". The path name is "MAINTENANCE".

#### The content of VolumeReport

The VolumeReport contains the following informations:

- short name of the customer

- total page counter
- date of the selected period
- number of processed pages in the selected period
- select completed Pages from BatchResult where date=actualPeriod and JobName='ExportXML'
- two validation codes
- total page counter and number of pages in this period encoded by Base64Encoder
- list of processed pages for each job

### Sending the VolumeReport to CCS

At the moment there exists no automatic transfer of these information to CCS in any way. The generated PDF-file has to be send by mail to [metae-support@ccs-gmbh.de](mailto:metae-support@ccs-gmbh.de).

### How to prove the validation of the VolumeReport

The properties of the PDF file contains the following information

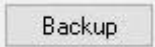
- title VolumeReport
- creator CCS docWizz
- theme <customerName>
- created at <actual system time>
- application CCS docWizz
- creator: CCS Content Conversion Specialists GmbH, Hamburg
- created by docWizz <FVersion> (f.e. docWizz 6.5.1.7)

### Behavior in case of errors

The VolumeReport dialog contains an own filter for errors. If an unexpected error appears the dialog would be closed without a message. Perhaps in the log database an error will be logged. The user can continue his work with docWizz without any problems. In case of expected errors (f.e. error in database requests) the error will be shown and logged in the log database.

### 30.3.3 Backups to Restore

The buttons **Backups to Restore** and **Recover documents** offer same functionality that you already know from the document pool (see **docWizz User's Manual**).

To create a backup manually, hit the  button. Choose the destination where to save the backup and decide whether you want to **include linked files** or not. This means, whether source images are included or not.



Of course, the backup file is much bigger if those files are included (especially if source images are grayscale or color) but for certain purposes it is better to do so. Finally you are not depending on the files on the file share with a backup including linked files.

To create a backup via UI, the document will be locked. If a document is already locked, no backup is possible.

Documents which are in the step RQA could be backup-ed, too.



There is a routine called Free Poolspace. Used before creating Backup files in order to make size of Backups smaller.

Note: you may use combination of "FreePoolDataOnBackupAndRemove" enabled and "AddInImagesOnBackupAndRemove" disabled. Then backup can't be restored successfully without placing source images before in the correct in-folder!

See setting "FreePoolDataOnBackupAndRemove" in ..\project-cfg\default\default-setting.xml (Default = 1, on).

```
<PROJECT-SETTINGS>
...
<PROCESSING>
  <FEATURE NAME="AddInImagesOnBackupAndRemove" ENABLED="1" />
  <FEATURE NAME="FreePoolDataOnBackupAndRemove" ENABLED="1" />
</PROCESSING>
...
```


If enabled, all recoverable data is deleted before creating a backup.

Note: You may use combination of "FreePoolDataOnBackupAndRemove" enabled and "AddInImagesOnBackupAndRemove" disabled. Then backup can't be restored successfully without placing source images before in the correct in-folder.

Log files for "Backups to Restore" entries are located in the restore table for documents, the Groups table "Groups ( GroupID int NOT NULL AUTO\_INCREMENT, Description varchar (255) , Project varchar (255) , Status tinyint, Priority tinyint, PRIMARY KEY (GroupID))"

### 30.3.4 Recover Documents

The buttons **Backups to Restore** and **Recover documents** offer same functionality that you already know from the document pool (see **docWizz User's Manual**).

To restore a document from a backup file, push the  button. Go to the directory where the backup files can be found and double click on the backup you want to restore. Choose whether you want to perform the restore job immediately on this workstation or you want the processing servers to perform the restore job.

Note: Restore might take some time, therefore we recommend to perform restore jobs on the processing servers.

If you add a couple of backups to the restore queue you can access this queue using the menu 'Configuration' - 'Maintenance'. Here, go to 'Backups to restore'. You are able to remove jobs here.

Note: This feature is restricted to administrators.

### 30.3.5 Auto-save

In case of system errors the auto-save functionality is very useful to save already done work.

- Auto-save is done:
  - every 10 minutes of inactivity (Idle status)
  - every 30 minutes when working (Active status), a short message will be shown



- Auto-save files are stored in Pool folder additionally to the document files
- If docWizz crashes and will be restarted you can select if you want to go back to the auto-saved status or not.
- Auto-save file will be deleted if the document is the next job
- Auto-save works in all jobs except Exported job
- On regular close of documents or docWizz, auto-saves are deleted
- <docID>AS<timestamp>.xml is created by rename after <docID>AS<timestamp>.zip is successfully stored.
- In ScanClient all page based data is stored immediately on disk. So no auto-save needed.
- A message will be shown when opening a document that was auto-saved.

## 30.4 Select scanner

You need administrator permission (login and password) to enter this area.



If a scanner is connected to your system, you can select it here.

The scanner setup can be controlled over this button, if a scanner is connected to your system. You need administrator permission (login and password) to enter this area.

## 30.5 Attach MS SQL-Server Express

### Attach Microsoft SQL-Server Express 2008 to docWizz

If the document-pool and log data base of docWizz shall be connected to a SQL-Server Express database the following steps are required:

Edit the file `***DATA***\config\<customername>-docWizz-glbl.ini`. If not already entered, add the section `DOCUMENT.POOL` and enter the following entries:

```
[DOCUMENT.POOL]
OLEDB=1
DATABASENAME="C:\Program Files\Microsoft SQL Server\MSSQL\Data\Pool"
SERVERNAME=<servername>
DATABASETYPE=SQLSERVER
DBTYPE=2
```

#### DATABASENAME

Is the local path where the databases of the SQL-Server are installed. The above example shows the default settings when the SQL-Server is installed on the local drive C:. The last string `...\Pool` of the pathname is the beginning of the database name which will be created by docWizz.

#### SERVERNAME

Is the name of the computer hosting the SQL-Server.

The log data base hosts processing data and the error log. To use the log data base with the SQL-Server the section `[ERRORLOGDB]` needs to be changed this way:

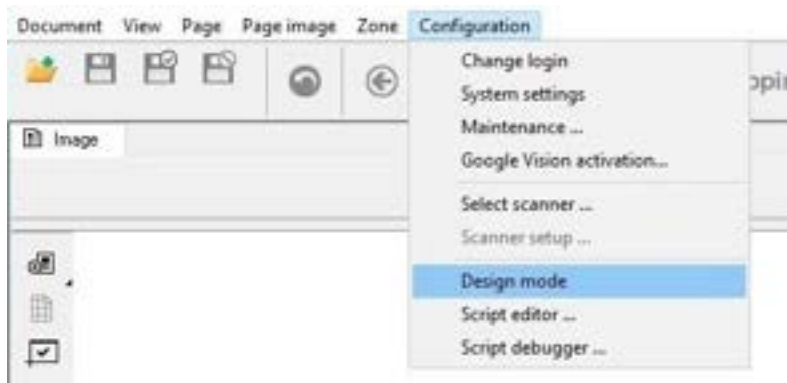
```
[ERRORLOGDB]
OLEDB=1
DATABASENAME="C:\Program Files\Microsoft SQL Server\MSSQL\Data\Log"
SERVERNAME=<servername>
DATABASETYPE=SQLSERVER
USERNAME=user
PASSWORD=password
```

When docWizz is started the first time with these configuration entries the databases will be created automatically.

**Note:** Be sure to restart any docWizz client or service once you created the database.

## 30.6 Design mode

You need administrator permission (login and password) to enter this area.



By pressing the right mouse button on right hand side working window on somewhere in the background area or on a specific button or field you will open a context menu.

Here you can edit elements (e.g. Field, Button, Checkbox, Text, Graphics, Image or System Control) and their functionality. This context menu provides the following menu functions:



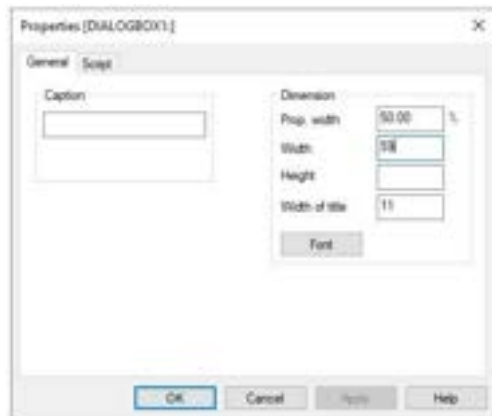
### Properties




The **Properties** function opens the **Properties** dialog that provides a variety of tabs for making specific settings.

You use the Graphics Element button to enter graphics elements you want - backgrounds or frames - in the dialog box. These elements are for appearance only, and have no function. Click the button and place the mouse pointer where you want the graphics element to appear in the mask. Draw a frame by holding down the left mouse button and then click the frame with the right mouse button. The context menu appears.

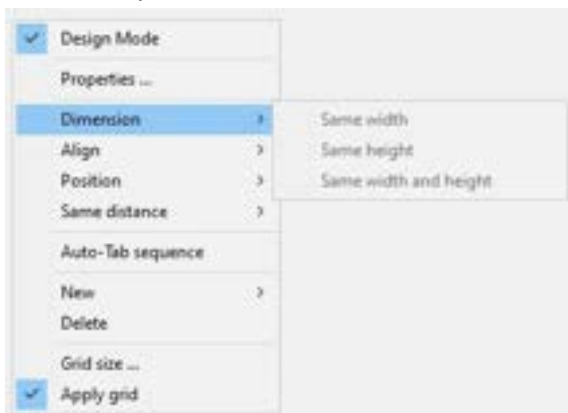
Clicking the **Properties** function opens the **Properties** dialog box for the graphic element for example GRF1.



Here you can specify the type and label of the graphics element, as well as other attributes. Confirm and exit by pressing .

### Dimension

If you have selected multiple elements in the dialog box with the Shift key and the mouse, the **Dimension** function allows you to standardize the size of all the elements at the same time. Placing the mouse pointer on the **Dimension** function, another selection menu offers three commands:



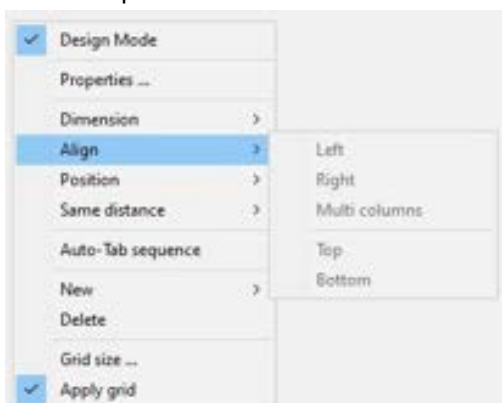
With the function **Same Width**, you can scale all the selected elements to the same width.

With the function **Same Height**, you can scale all the selected elements to the same height.

With the function **Same Width and Height**, you can scale all the selected elements to the same width and height.

### Align

If you have selected multiple elements in the dialog box with the Shift key and the mouse, the **Align** function allows you to align all the elements you have selected. Placing the mouse pointer on the **Align** function opens a selection menu beside the arrow that offers 5 commands:



The **Left** command aligns the marked elements to the left.

The **Right** aligns the marked elements to the right.

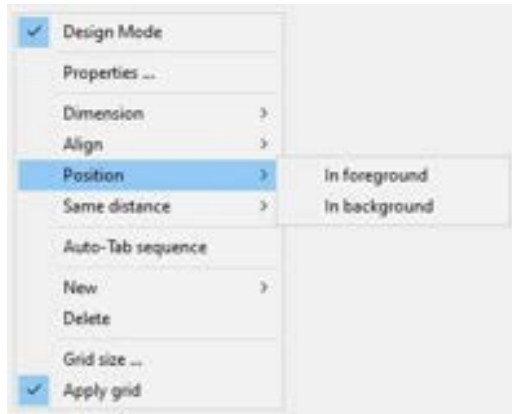
The **Multi Columns** command aligns the marked elements in multiple columns.

The **Top** command aligns the marked elements along the top.

The **Bottom** command aligns the marked elements along the bottom.

## Position

Use the **Position** function to place the element you select in the foreground or background. Placing the mouse pointer on the **Position** command opens a selection menu with two commands:



## Same distance

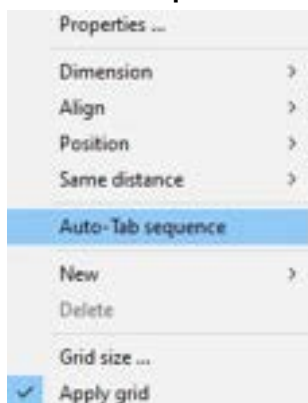
With the **Same Distance** function, you can specify whether the vertical separation between the selected elements should be uniform.

Placing the mouse pointer on the **Same Distance** function opens a selection menu offering the following command:



**Vertical** means same distance in vertical dimension.

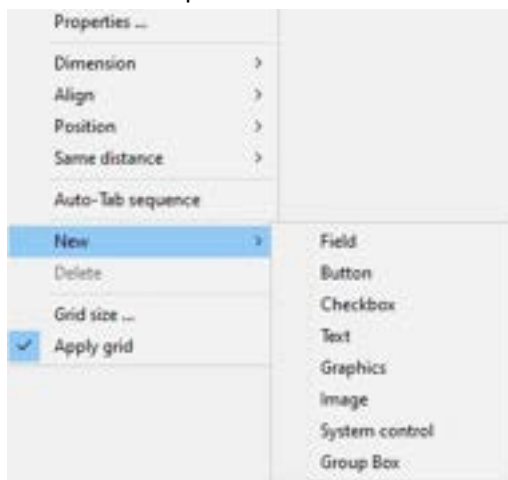
## Auto Tab Sequence



You use the **Auto Tab Sequence** function to have automatically set the jump sequence for addressing the control elements when the **Tab** key is pressed. There is also the possibility to determine the order manually.

## New

You can add a new element to the dialog box with the **New** function. Placing the mouse pointer on the **New** function opens a selection menu beside the arrow that offers 6 functions:



With these functions, you are able to create different elements like **Field**, **Button**, **Checkbox**, **Text**, **Graphics**, **Image** and **System Control** buttons.

## Delete



You can delete the selected element with the **Delete** function.

## Grid Size

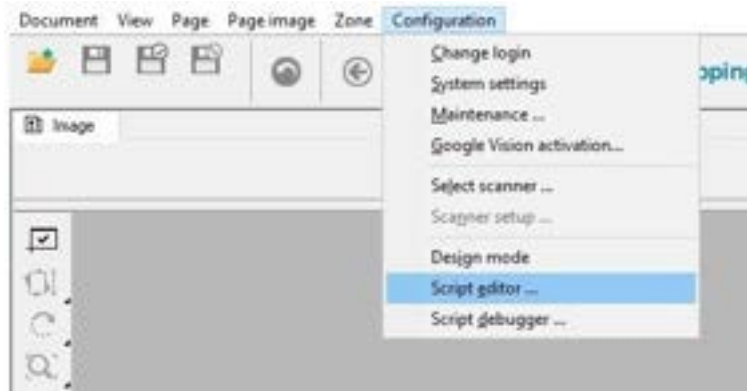


You set the size of the grid the system uses for orientation purposes with the **Grid Size...** function. Clicking on this function opens an input mask in which you specify the desired horizontal and vertical spacing between the grid lines. Make your settings in millimeters:





## 30.7 Script Editor

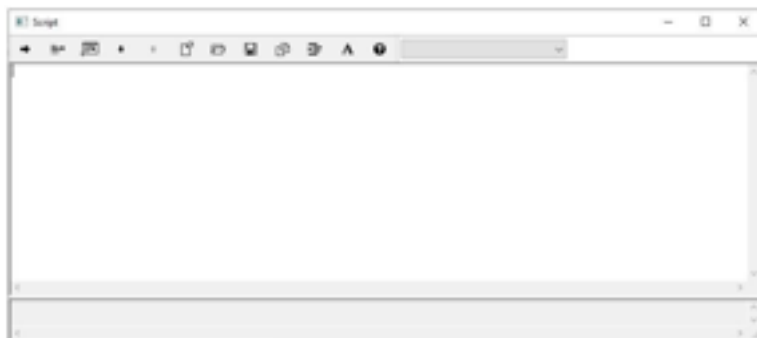


You need administrator permission (login and password) to enter this area.

The execution of the program functions can be influenced with the help of the script editor, where scripts can be written, edited and test-run.

You can restructure the graphic interface to suit your special needs, write your own applications, and edit existing ones.

We use the scripting language TCL (Tool Command Language), which has its own syntax. It is not within the scope of this manual to describe this syntax. For a detailed description of TCL, we recommend that you refer to technical literature on this subject e.g. Ousterhout, John K: Tc and the Tk Toolkit (Addison-Wesley Professional Computing).



A small collection of scripts is included in the installation CD, an extract of which is shown in the illustration above. To write in the script editor, click the mouse in the editor window. A cursor will appear in the first line.

You can operate the script editor with the help of the icons in the toolbar at the top of the editor window.



**Run**

With this icon, you can execute the written (visible) script; it will be run in the script editor. Results and error messages will be displayed in the status bar below. In this manner, you can check out the script in the script editor to see if the desired results are being produced.

The executed script is saved as a sheet in the editor. If changes are made to this script, this is saved as a separate sheet. In this manner, one can watch the progress of the script, and can leave through the various sheets, cut and paste from them without having to rewrite whole scripts or deleting them (History-Function). Thus, it is also possible to execute the individual components of the scripts and change them if necessary.



**Run line**

With this icon, you can execute only that line of the script where the cursor is positioned. It is also possible to mark one or more lines and execute only the marked lines. You can mark parts of the text in

the usual manner by holding down the left mouse key and drawing the mouse to highlight the desired lines. A new sheet is saved also after carrying out this function.



#### **Trace**

The **Script-Debugger** is opened with this icon. This program is used to check out scripts. You can run sections of the script, and check out the contents of the variables and set breakpoints. See Script Debugger.



#### **Next Script**

Displays the next sheet of the script. When several sheets have been saved, you can scroll through the pile of sheets with the help of this icon and the next one.



#### **Previous Script**

Displays the previous sheet and helps you leaf through the pile of sheets along with the previous icon.



#### **New Script**

A fresh sheet is opened for the creation of a new script. Previously created sheets are not deleted. You can move forward and backwards among the script sheets with the help of the icons **Next Script** and **Previous Script**.



#### **Load Script**

Opens the Windows file system dialog box to help you navigate to the script-file that you want to load.



#### **Save Script**

Opens the Windows file system dialog box to help you navigate to the folder where you want to save the script as a script-file.

The sheet that is visible on the screen whilst carrying out this function, i.e. the final version of the script is the sheet that is saved.

Scripts, regarding a Task are stored in sub-folder "Task".



#### **Search a string in the script**

You can search the script for a word or string, and specify with the help of the check boxes whether the search should be case sensitive, whether to search for the full word, and to specify the direction of the search in the script: upwards or downwards.



#### **Format the script (indent) automatically**

Reformats the command lines of the TCL script automatically, for example the text between brackets are indented. This function locates errors very easily.



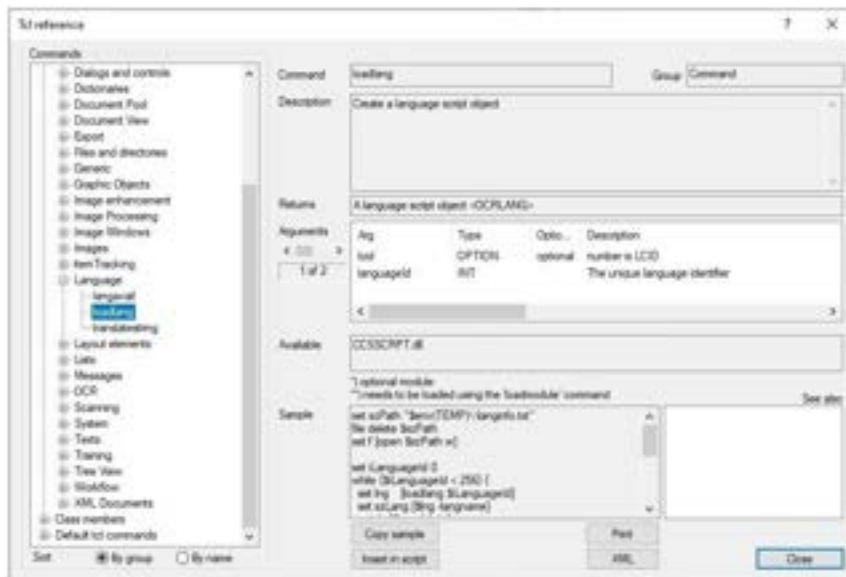
#### **Select font**

You can specify the font type and font attributes for the editor with this icon.



#### **Help**

With this icon you open the TCL Reference window. The complete list of all the script commands and related information is available here for reference:



The **Commands window** on the left lists all the TCL commands sorted alphabetically by name or according to groups: **Commands**, **Class Members** or **Default Commands**. **Highlighting** a command on the left will display all the relevant information in various fields on the right.

The **Command** and its **Group** are displayed at the top, followed by a **Description** of the function it serves. The next field lists the possible **Returns** from the script, e.g. "1" if successful, "0" otherwise. The **Arguments** that can be used with this command are listed next, along with information about the type, the options, and the description respectively.

Further, information is provided about the modules support this command in the **Available** window. Related information sources are listed in the **See Also** window.

After perusing the information sheet, you can **Insert** the command in the script you are currently working on, or you can **Print** out the information. Choose **sort by name** and **next Print** to print a complete list of all TCL commands.

Pressing the **Close** button will exit the TCL reference window. You can close and exit the Script Editor with the help of the exit button on the top right of the screen. The scripts that have been edited are not deleted. If there are unsaved scripts in the **Script Editor** when you close main program, you will be prompted with a question about whether you would like to save them or not.

TCL History (Script-Editor) is not in the config directory (docWizz.dwx), but in the Work directory of each workstation.

### Script editor short keys

Key	Action Performed	Comments
[CTRL] + [F]	Find	
[CTRL] + [O]	Open TCL (script) file	
[CTRL] + [R]	Run Script	
[CTRL] + [S]	Save	
[F9]	Next Step	in Script debugger

## 30.8 Script debugger

You need administrator permission (login and password) to enter this area.

The **Script Debugger** window can be opened from the **Script Editor** as described above:



With the help of the **Debugger**, you can run a TCL-Script one line at a time and check the contents of the variables at every step. Break points can also be defined in this window.

### Stop Tracing

When the **Debugger** window is open, and the **Run** button in the **Script Editor** is pressed, the script stops after executing the first command and you have the choice of executing the script command by command. Toggling this button will execute the script without breaking until the end. The defined breakpoints will not be taken into consideration. [F5]

### Start Tracing

The **Debug Mode** is switched on again.








### Watch Variables

A window is opened, in which the name of the variables and their contents can be displayed in two columns.

Clicking the mouse in the column **Variable** will allow you to enter the name of a variable or to edit it. You need not proceed the variable name with a "\$"-sign as in the script.

### Show Break Condition

In this window, which opens up, you can enter the conditions, which will cause the script to make a break. (See also the **Break** execution after every command button below):

-  Add a Break Point
-  Edit the selected Break Point
-  Activate the selected Break Point
-  Disable the selected Break Point
-  Activate/Disable all Break Points
-  Delete the selected Break Point
-  Delete all Break Points

Write the following script in the **Script Editor**:

```
for {set i 0} {$i < 10} {incr i} {
  set a [expr $a +$i]
}
```

Enter the following condition for a break point:

```
$i == 5
```

In this window, the name of the variable must be entered with the \$-sign as in the script. When the script is started with the **Break Points Mode**, it runs until the value of the variable i reaches 5.



#### **Continue execution**

The script executes the next command. [F10]



#### **Break execution**

After every command: When this toggle button is pressed, the execution of the script halts after every command when the script is started with the **Run** button from the script editor or when continued with the button **Continue execution**. If the button is not pressed, the script stops only at the defined break points. [F9]



#### **Clear Trace Window**

All the entries in the trace window are deleted.



#### **Exit execution**

The executed script is stopped and the program is exited.

#### **Script debugger short keys**

**[F5]** - Stop tracing - continue script execution\nStop tracing

**[F9]** - Break execution after every command\nOne step execution

**[F10]** - Continue and interrupt execution in trace mode\nContinue and interrupt execution

## 30.9 Encrypted passwords can be used

docWizz supports storing encrypted passwords in configuration. This is helpful for example to store passwords for database connections. Please note that docWizz treat such passwords strictly confidential. Never decrypted passwords will be visible in user interface. As well, there is no scripting command or error message that could help to read a password.

### Storing of encrypted passwords

Encrypted passwords are stored in docWizz-glbl.ini in section "pwd". To store a password, you are able to execute a script command: "sys enterpwd name".

A dialog shows up and the user is requested to enter a password. After selecting OK, the password is encrypted and saved using the given "name".

### Accessing encrypted passwords

To access an encrypted password in script commands, use the placeholder ">>>name<<<" instead of the password. When the command will be executed, the placeholder is internally replaced by the encrypted password.

### Sample usage

```
sys enterpwd DATABASE ; # this call should be executed just once to enter
and store the password

# the next two lines show how to use it when accessing a database
set szOpenStatement "Driver={Sql Server}
;Provider=SQLOLEDB;PWD=>>>DATABASE<<<;Trusted_Connection=no;UID=DBUser;DATA
BASE=MyDB;
SERVER=OurServer;Network Library=dbmssocn"
set db [dbconnect $szOpenStatement]
```

## 30.10 SuspiciousBlock

Configuration added **SuspiciousBlocks** to list view to identify and correct or delete untypical elements like noise blocks.

## 30.11 Black & White conversion configuration


Due to the fact that black & white images sometimes have poor quality in several places in production the core KDocPage::GetImageSW() function is configurable. This script is customizable per project. As well a default for that script is available.

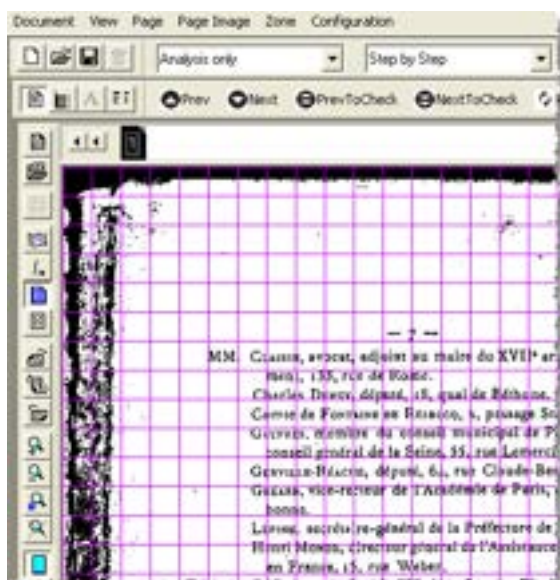
### Script call in KImage\* KDocPage::GetImageSW()

```
#if this script return empty the script is doing nothing old code is runned
#if true is returned means that $page -createbwimage was called so the code
will not do anything
#if <IMAGE> is returned that image will be assigned as BW image on page
```

By default the old code is enabled (so the script return empty). The scripts/code now keep the back compatibility with all projects.

## 30.12 Grid configuration

The  **Show page grid** button shows a grid on the whole page. This helps to adjust the image.



- The grid can be configured for any view in any step in `<custom>-docWizz-dw.ini`. Outline image view (not in full screen, or detail image view - since here it overlaps with the other correction way).
- The grid button by default is to have in outline image view a blue grid with 10X10 mm grid squares. The appearance of the grid can be changed in color and size of the grid squares.
- Add the following parameter in docWizz-dw.ini under the [docWizz] group. This will change the grid appearance to a magenta color and 15 mm squares and magenta color:

`PAGEGRIDSTEPSIZE=15`

size in mm for the grid step

`PAGEGRIDCOLOR=0xfc00cd`

hex value starting with 0x representing a color

### 30.13 Re-Scan as well from Scan job

For some projects it is required to stop documents for doubled image QA in Scan step. In case there a Re-Scan request is detected, there was no possibility to use the Re-Scan job with all the functionalities on the scanner machine.

Now adapted to set the right `stopatjob` when routing from Scan to Re-Scan.

In case a document has problems in Scan job, these documents can be sent to Re-Scan straight away to prevent manual workaround or required re-import of the document.

Use the Re-Scan job with all functionalities as well in the Scan job; intermediate solution, till ApplyRescan will be adapted.

In config/misccfg/job\_workflow.xml you need to replace:

```
<JOB name="Scan">
  <NEXTJOB>DetectPageFrames</NEXTJOB>
  <ALTERNATEJOB>PrepareImport</ALTERNATEJOB>
</JOB>
```

with

```
<JOB name="Scan">
  <NEXTJOB>DetectPageFrames</NEXTJOB>
  <ALTERNATEJOB>PrepareImport</ALTERNATEJOB>
  <ALTERNATEJOB>Re-Scan</ALTERNATEJOB>
</JOB>
```

and

```
<JOB name="Re-Scan">
  <NEXTJOB>ApplyRescan</NEXTJOB>
</JOB>
```

with

```
<JOB name="Re-Scan">
  <NEXTJOB>ApplyRescan</NEXTJOB>
  <ALTERNATEJOB>Scan</ALTERNATEJOB>
</JOB>
```

and you need to choose Scan to return back to Re-Scan.



## 30.14 Local processing folder configuration

In xxx-docWizz-glbl.ini a local processing folder can be defined. By that it can easily be implemented that exports first are written locally and after successful completion the complete data set is moved to OUT share

```
[LOCALEXPORLOCATION]
1=c:\DW_TEMP1,2,G
2=***DATA***\DW_TEMP2,1,G
```

Explanation:

"c:\DW\_TEMP1" is the path for the local export location.

"2" is the minimum limit of required free space;

"G" is the unit the required free space is calculated in.

In case the first location is out of space ("1="), the second location ("2=") is used.

Temporary local processing folder usage is helpful for SetDocOnLoader task as well as for exporting documents.

By the "localprocessing" command it can easily be implemented that exports first are written locally and after successful completion the complete data set is moved to OUT share.

Uses script command "localprocessing" where options could be:

- "get networkpath estimatedsizerequired" this function returns an unique path (depending on exe name) on the first local drive from list that has sufficient memory free. If none, the networkpath itself is returned
- "clean" this function deletes all files from local unique path
- "move networkpath" this function moves all files from local unique path to network folder.

The local unique path must always add "localprocessing" to the path that is configured.

## 30.15 Set up a Test Configuration

### Create Configuration Files and Folders

Open folder **\docWizz\project-cfg** and create folder named **Test**.

Copy the following data from your Project Configuration Folder (**\docWizz\project-cfg\<PROJECT\_NAME>**) into this folder:

Document Structure Data file (<PROJECT\_NAME>-DSD.txt) and re-name it Test-DSD.txt.

Export XML file (<PROJECT\_NAME>-export.xml) and re-name it Test-export.xml.

- **PDF** folder and content.

**rdy** folder and content. Re-name the file <PROJECT\_NAME>-.xml into Test.xml.

Open IN folder **docWizz\IN** and create folder named **Test**.

Copy one or more documents on which you would like to run the test into the Test folder on the IN directory.

Run docWizz, open Document Pool, select the document. You can either press OK to start at the first step and proceed as usual. Or you can hit Route and select the next step according to your needs; the Job field in Document Pool will change, you can click it, hit ok and the document will be processed in the selected step.

Having set up a test environment the tests can be carried out.

## 31 Add or edit new "NonSortWords"

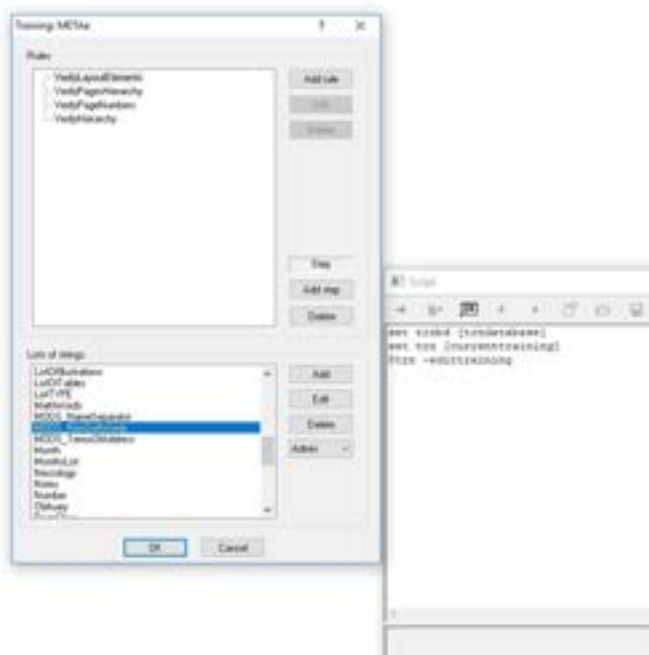
To add or edit new "NonSortWords" this procedure should be followed:

- Open the "Script editor" tool from Configuration menu. You need administrator permission (login and password) to enter this area.
- Place this TCL script lines inside the editor window.

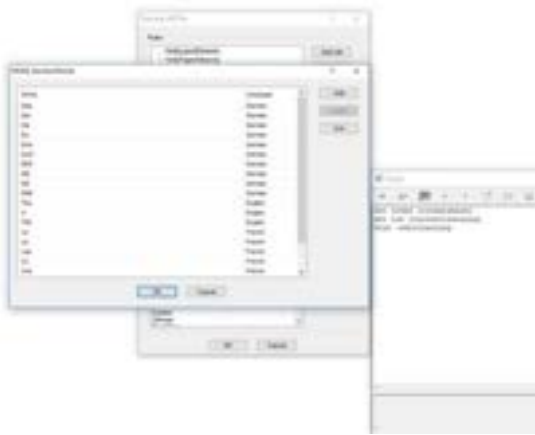
```
set trnbd [trndatabase]  
set trn [currenttraining]  
$trn -edittraining
```



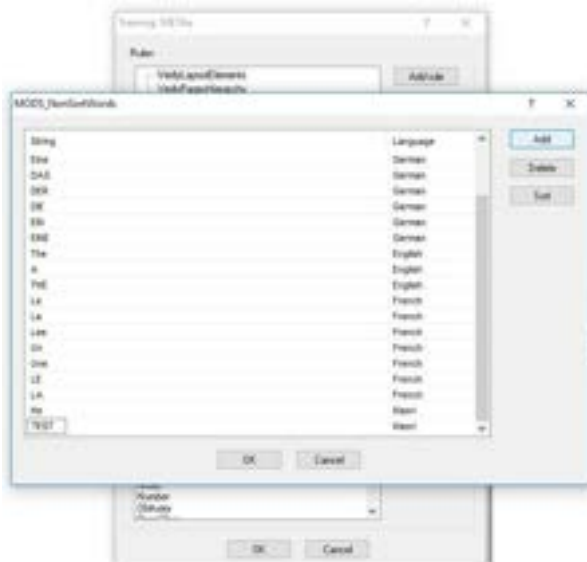
- Execute the written (visible) script. For this press "Run" button from "Script editor" interface.
- A new interface window "Training: METaE" appear.



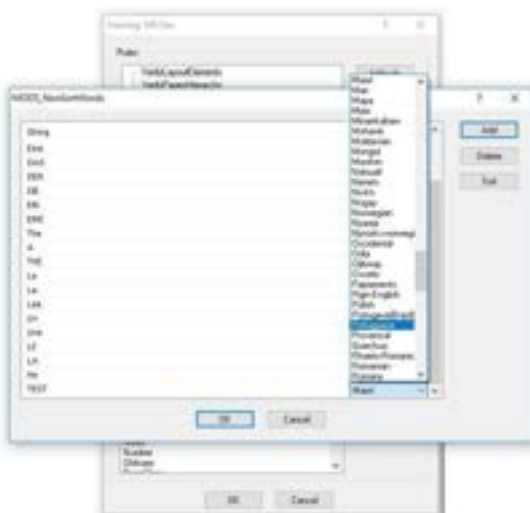
- In "List of string" select the "MODS\_NonSortWords" and press **Edit**.
- A new interface window - "MODS\_NonSortWords" appear.



- To add a new "NonSortWord" press **"Add"**. Then insert the content in the specific **"String"** field (e.g. TEST) generated at the bottom of the list.



- Select the right language for the new added "NonSortWord" and press **"OK"** to save.



- To edit a record please double-click on the **"String"** field for the specific "NonSortWord" record and modify the content or click on **"Language"** field and select a different language.
- To delete a "NonSortWord" record just select and press **"Delete"** button.

## 32 docWizz ControlCenter

The dWControlCenter is a cockpit for managing the production workflow and system environment.

Here you monitor the docWizz services. Steps and tasks can be prioritized and different administration tools are available.

In contrast to a patchwork of different tools as often used in digitization projects, dWControlCenter integrates all necessary components. It enables multi-project management and statistical analysis and covers support- and error-handling.

It's highly beneficial for project managers, team leaders and IT administrators.

Log in to dWControlCenter with the same user login mechanism like in docWizz.


### 32.1 License key

Go to the **Help** menu and open the **About dWControlCenter** entry.

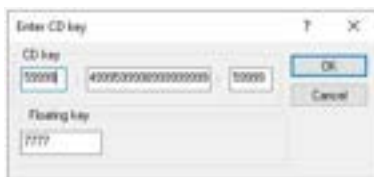
This function enables you to refer to the information window, which gives you details of the version of docWizz Control Center installed on your system.

In the view boxes you can see the name of the **User** who is currently working on the system, which **CD Key** you are using, and also your **Registration-ID**.


The extent to which you can access different system components depends on the CD Key you have received from CCS. You can retrieve information about your module access rights from a system list.

If you have purchased new modules or features or even additional licenses, these can be activated with a new CD Key. Click the  button and enter your new CD Key number.

There is a additional field to be able to enter the license codes in the CD key dialog. The dialog ensures and validates, that the entered codes are valid and stored in the right ini (custom-glbl ini).



Complete the entry by clicking the OK button.

See additional copyright information by clicking the  button.

## 32.2 Services status

Here you manage all the services on root-, group- and services-level. You can start, stop, kill, shutdown, cancel, and restart services. Another functionality is to check and edit the configuration of services.

The main dialog of the Services status is as follows.

If the buttons on the right appear inactive please login as administrator using the **Change login** button.

When in left tree view a group element is selected, on the right all services are shown as list on the right. By double click (in case you want to apply actions on it) you can jump directly to the service instance on the left tree view.

### 32.2.1 docWizz service levels

The tree view shows different levels:

The services groups nodes automatically expand in the tree when the numberOfChildren is higher or equal to MaxNoOfEntriesToAutomaticallyOpenTheGroup of [DWControlCenter] section from docWizz-dwsrv.ini















By extending the tree on the left hand side below docWizz Services all machines part of the production environment are shown.



















This includes workstations and servers where docWizz service(s) is/are executed.

docWizz services can be categorized into the following:

- regular services such as: dWSrv, dWSrv2, dWSrv3, and dWSrv4
- special services such as: dWRemoteOCR, dWFTPClient, dWRemoteQAManager, dWRemoteQALoader

For each machine an icon indicates its current status which can be one of the following:

dW 6.8	dW 6.5	Description
		service is executed and currently processing
		service is executed but currently idle. It will pick up documents as soon as there are documents to be processed
		service is stopped and needs to be started in order to pick up documents for processing
		service is about to start
		service is about to stop
		service is about to shut down
		service is about to restart

		service is not available. It either does not exist or has been disconnected. Check local event log/computer is running, but "dWSrvManager" is not running.
		only available on RQA manager. Service is executed on a configuration other than the one of the system (Error Message). Any issues which should be investigated, temporary issues. - In PoolStatus folder (under clients folder) one of the *.csv files (csv.mtn / xml.mtn) has an error into it. On non document error the type is shown as button text: - Error documents on RQA transfer - Missing documents on RQA transfer - Files are too old
		service is performing tasks (e.g. auto import, extra Tasks, ...)
		service is still running, but has not reported back any progress. Is indication that the service might not be operational.
		environment stopped for "maintenance" - service is kept stopped
		environment stopped for "maintenance" - service kept running
		machine is set to night mode
		OCR license expired (RemoteOCR service, Gothic OCR)
		only on Group element - the services within the group have different status. When all have same status, group icon is the same as the state of the services.

By clicking on the according machine one gets a **detailed view** on the selected computer, including current document ID, current job, action performed etc. Here, the according machine can be started, stopped or shut down. Return to docWizz Control Center by hitting docWizz Services on top of the tree again.

A tool tip is displayed showing which limit cause the icon - also for the green ones to show that the green is not due to second condition on same job (f.e. POOL and EXPORT), than for POOL1 and POOL2.

Note: On one computer up to 4 instances docWizz can work parallel. Thus we reach a very efficient use of the hardware and the support of multi-processor computer and Dual- and/or Quad-Core processors.

The Order of the groups at the left tree view is alphabetically.

There are four subtasks (CollectData, CommandFTP, UpdatePool, UpdateReady), how is the handling, when more than one reports a state, which causes the icon to change?

CollectData and UpdatePool both have a different warning/error, which would change the icon on RQAManager in DWCC.

It is shown just the first error is coming up.

Usually all subtasks show the same status like: start, stop, process, maintenance.

In the case of start first worker updates icon.

In the case of stopping the service the logic is reverse (last child update icon as stop).

In the case of processing if any subprocess status working icon will show RQAManager as working. Subtasks are independent so if anyone is working that icon shows as working.

For Dwsrv, DwOCRRemote, DWRemoteQAManager and DWRemoteQALoader, there are several labels displayed in DWControlCenter in the corresponding service view (right hand side), for example, user can see the version of the service, command line of the service, time difference and configuration file.

Label sets the client name according to the client name configured in GLOBAL.ini file.

The user's benefit is that he can now correctly see the configuration on which the corresponding service runs. In DwControlCenter, for each of the mentioned services that use the same Client cfg, the Config file: label will display the correct configuration file.

### 32.2.2 Monitor progress

Processing logs (like ID, Task name, Filter, Start time, ...) can be sorted by clicking on header column.

For investigation of systematic issues it is now possible to verify just by one click, if each document in that job failed on that machine. Also possible to filter out f.e. the tasks like AutoDelivery to find just the history of this task and verify the scheduled execution times are working fine.

With right mouse button you open context menus in tree view to start/stop/shutdown/restart/kill... services.

#### Context menu

Start All Services is also available for none-admin users.

#### Context menu per service

##### **Check now**

Starts the service.

##### **Stop**

Stops the service; task in progress will be finished.

##### **Restart**

Restarts the service. Combines a „stop" and „start" command

##### **Shutdown**

Task in progress will be interrupted and stops the service. Graceful. Ends (as in waits for a proper moment to interrupt the task, in order to not affect the document) the current task and stops the service. The document is returned to Prepare cropping and is not locked.

##### **Kill**

Interrupts the task (no matter what) and stops the service. Ungraceful. It will take some time until the dWCC realizes that the service has been killed. So it might take a couple of seconds until the task is displayed in stopped-state. The document is locked in Modify Pages.

##### **Cancel**




Ends the current process; like pressing "break" when processing a document in step-by-step and most importantly - the service is not stopped. The document is returned to Prepare cropping and is not locked.

**dWSrv** is frequently updating the status file (is executed) if the time difference between current time and file time is bigger than 2 minutes. If so, Control Center shows a warning for services having that time difference.



Note: You must not test the time difference when status file is not updated, because then the service could be stopped etc..

There are three colored icons for time difference:

- |   |        |   |  |
|---|--------|---|--|
|  | GREEN  | = | No time difference (or less than 2 minutes)  |
|  | RED    | = | Time difference (greater than 2minutes)      |
|  | YELLOW | = | Status undefined (can't be read from remote) |

Next to the box is the time difference in minutes, with a + or - suffix; + means more accurate time, DWSRV has "late" time.

### 32.2.3 Monitor progress for RemoteQA/Remote OCR

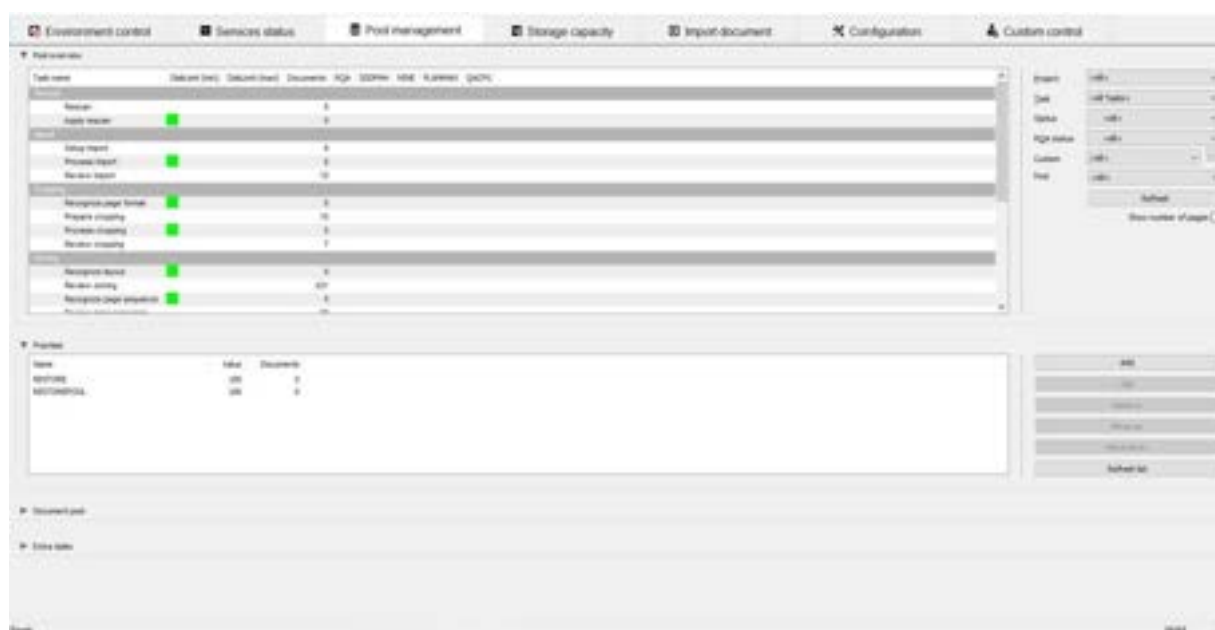
Because the Gothic OCR licenses are character limited whenever the remaining number of characters gets very low this needs to be outlined in Control Center properly to make you aware about the status of each Abbyy FR license equipped with Fraktur (Gothic). See "Faktur (Gothic) OCR licenses" chapter.

There are two warnings implemented:

- one for each service blocking to the machine where this special type of OCR is running
- and a second one on the Remote OCR service belonging to the same machine.

### 32.3 Pool management

Check progress: number of documents/pages per project or task.



Tool tip is displayed showing which limit cause the icon - also for the green ones to show that the green is not due to second condition on same job (f.e. POOL and EXPORT), than for POOL1 and POOL2.

### 32.3.1 Pool overview

In the area located in the center of the interface the number of documents for each job is shown. By hitting the Refresh button the view is updated.



Control Center shows number of pages in a different color to make visible, if the user has selected pages option.

### 32.3.2 Project, Job and Status



You can apply various **filters** in order to reduce complexity of the view. Certain projects, jobs and/or status can be selected from the drop down menus.

The **Project**, **Job** and **Status** combo boxes will behave the same like in Pool open documents dialog. In the list control below, you will see all jobs and the number of documents per job that are matching the selection from combo boxes. If a single job is selected, the list box will show projects and the number of documents. If a single job and a single project is selected, the list box will show each available status and the number of documents.

With the check box **Show number of pages** you can switch between number of pages instead of number of documents.

The Controls will be refreshed from time to time to reduce network/SQL traffic. If you click on **Refresh**, all items are updated immediately.

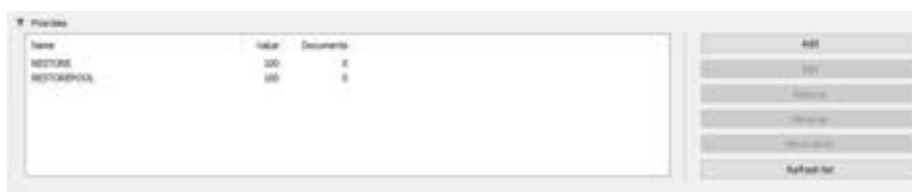
**RQA Client** for RemoteQA services.

Control Center displays number of documents per RQA location for automatic jobs in light gray. This helps to see, how many documents will appear soon in which RQA location.

Each process in RemoteQA services reports its processID in the status file. The processID is displayed after the status, like "Idle (PID:5677)" for each RemoteQA service.

### 32.3.3 Priorities

The list control shows all defined priorities. docWizz will process the documents as specified by the priority settings.



Name	Value	Documents
RESTORE	100	18
RESTOREPOOL	100	0

The list has the following columns:

- Name (Job**Name**/Project/DocumentID/Title/SQL Query)
- Priority **Value**
- No. of **Documents** matching the priority

No of documents that will be processed before the next document of this priority will be processed.

#### Priorities

Priorities are handled from top to bottom.

The priority value specifies, how often documents of a lower priority are processed.

A value of 100 defines that first all documents that are matching the priority condition will be processed.

A value of 80 identifies, that 20% of document processing is used for documents that have a lower priority.

A value of 0 identifies, that those documents are processed if no other documents are available for processing.

The single services can be moved up or down or removed from priority list. Existing ones can be edited.

Priorities will be stored in the document pool database.

**Move up/Move down** tells the sequence.

### Add new Priority

When clicking on **Add**, a dialog opens:



The user may add a **Priority Value** in range of 0-100 (default is 60).

### There are different ways to define the priority

Task/Project based priority checked – the user selects a job/project that shall run with priority (based on job/project priority) doc priority edit box and SQL query edit box is disabled.

Doc ID based priority checked - instead of using docWizz, we are also be able to add a doc with high priority from this console. The SQL query edit box and job/project lists are disabled.


Complex SQL query based priority - and SQL query shall be input. A Validate button is available to be sure a correct query is added. Doc ID edit and job/project lists are disabled.

### Document ID based priority



Beside the **Jobname/Project priority**, as well a **specific document (ID)** can be added. Therefore in pool open dialog in docWizz, the user can perform a right-click on a document and check from context menu "High Priority". Then the document is added at top to the priority queue. Also using Set Priority dialog box this can be done.

## Complex SQL Query based priority



Complex SQL query based priority

SELECT \* FROM [MAIN] (add the rest of the statement below)

Validate query

Priority name:

**Complex SQL query** based priority - and SQL query shall be input. A Validate button is available to be sure a correct query is added. Doc ID edit and job/project lists are disabled.

## Work task



Work Task

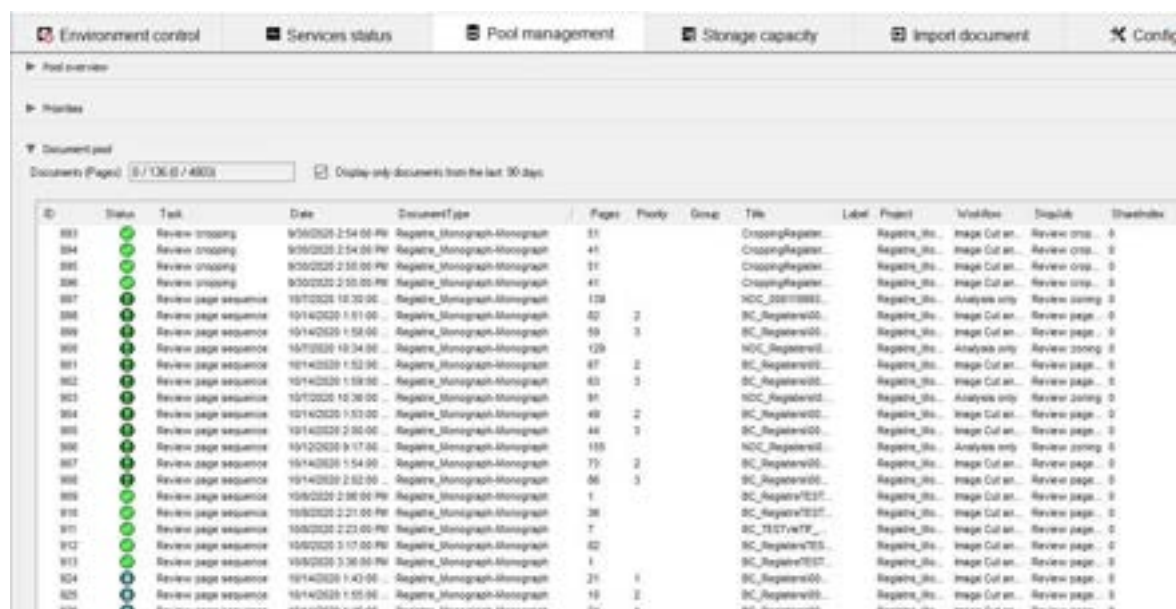
EXTRATASKSWITHOCR  
FTPFRID  
IMPORTIMAGES  
LDRCOLLECTDATA  
LDRUPDATEDATA  
MNGRCOLLECTDATA

## Handling of locking priorities:

"Couldn't lock priorities.xml" appears just in case of a real problem.

## 32.3.4 Document pool

The document pool shows intermediate results of documents in any step.



Environment control Services status Pool management Storage capacity Import document Config

Pool overview

Priorities

Document pool

Documents (Pages): 5 / 106.6 / 4803 Display only documents from the last 30 days

ID	Status	Task	Date	Document type	Pages	Priority	Group	Title	Label	Project	Workflow	Exclude	ShareIndex
883	✓	Review cropping	9/5/2020 2:54:00 PM	Registre_Monograph-Monograph	31			CroppingRegiste...	Registe_ite	image Cut en	Review crop...	0	
884	✓	Review cropping	9/5/2020 2:54:00 PM	Registe_Monograph-Monograph	41			CroppingRegiste...	Registe_ite	image Cut en	Review crop...	0	
885	✓	Review cropping	9/5/2020 2:55:00 PM	Registe_Monograph-Monograph	31			CroppingRegiste...	Registe_ite	image Cut en	Review crop...	0	
886	✓	Review cropping	9/5/2020 2:55:00 PM	Registe_Monograph-Monograph	41			CroppingRegiste...	Registe_ite	image Cut en	Review crop...	0	
887	✓	Review page sequence	10/1/2020 10:30:00	Registe_Monograph-Monograph	128			NDC_200118881	Registe_ite	Analysis only	Review sorting	0	
888	✓	Review page sequence	10/14/2020 1:51:00	Registe_Monograph-Monograph	42	2		BC_Registe000	Registe_ite	image Cut en	Review page...	0	
889	✓	Review page sequence	10/14/2020 1:58:00	Registe_Monograph-Monograph	59	3		BC_Registe000	Registe_ite	image Cut en	Review page...	0	
890	✓	Review page sequence	10/1/2020 10:34:00	Registe_Monograph-Monograph	129			NDC_Registe00	Registe_ite	Analysis only	Review sorting	0	
891	✓	Review page sequence	10/14/2020 1:52:00	Registe_Monograph-Monograph	47	2		BC_Registe000	Registe_ite	image Cut en	Review page...	0	
892	✓	Review page sequence	10/14/2020 1:59:00	Registe_Monograph-Monograph	43	3		BC_Registe000	Registe_ite	image Cut en	Review page...	0	
893	✓	Review page sequence	10/14/2020 1:38:00	Registe_Monograph-Monograph	91			NDC_Registe00	Registe_ite	Analysis only	Review sorting	0	
894	✓	Review page sequence	10/14/2020 1:53:00	Registe_Monograph-Monograph	49	2		BC_Registe000	Registe_ite	image Cut en	Review page...	0	
895	✓	Review page sequence	10/14/2020 2:00:00	Registe_Monograph-Monograph	44	3		BC_Registe000	Registe_ite	image Cut en	Review page...	0	
896	✓	Review page sequence	10/12/2020 9:17:00	Registe_Monograph-Monograph	105			NDC_Registe00	Registe_ite	Analysis only	Review sorting	0	
897	✓	Review page sequence	10/14/2020 1:54:00	Registe_Monograph-Monograph	75	2		BC_Registe000	Registe_ite	image Cut en	Review page...	0	
898	✓	Review page sequence	10/14/2020 2:02:00	Registe_Monograph-Monograph	56	3		BC_Registe000	Registe_ite	image Cut en	Review page...	0	
899	✓	Review page sequence	10/6/2020 2:08:00 PM	Registe_Monograph-Monograph	1			BC_Registe000	Registe_ite	image Cut en	Review page...	0	
910	✓	Review page sequence	10/6/2020 2:21:00 PM	Registe_Monograph-Monograph	38			BC_Registe000	Registe_ite	image Cut en	Review page...	0	
911	✓	Review page sequence	10/6/2020 2:23:00 PM	Registe_Monograph-Monograph	7			BC_Registe000	Registe_ite	image Cut en	Review page...	0	
912	✓	Review page sequence	10/6/2020 3:17:00 PM	Registe_Monograph-Monograph	42			BC_Registe000	Registe_ite	image Cut en	Review page...	0	
913	✓	Review page sequence	10/6/2020 3:36:00 PM	Registe_Monograph-Monograph	1			BC_Registe000	Registe_ite	image Cut en	Review page...	0	
924	✓	Review page sequence	10/14/2020 1:43:00	Registe_Monograph-Monograph	21	0		BC_Registe000	Registe_ite	image Cut en	Review page...	0	
925	✓	Review page sequence	10/14/2020 1:55:00	Registe_Monograph-Monograph	19	2		BC_Registe000	Registe_ite	image Cut en	Review page...	0	
926	✓	Review page sequence	10/14/2020 1:45:00	Registe_Monograph-Monograph	24	1		BC_Registe000	Registe_ite	image Cut en	Review page...	0	

In order to give a better overview operators can apply **filters** to show documents in the pool. Please use the drop down menu "Project" to filter by project. Please use the drop down menu "task" to filter by task. Please use the drop down menu "Status" to show only documents that have a certain QA status. One, two, three or none of the filters can be applied.

Further more, operators can browse for documents within the document pool by typing in the **document ID**.

The interface disposes of a display showing **number of selected documents** as well as **total number of documents** within the document pool.

A button **Change Status** has been placed on the right hand side of the pool.

Sort entries by the arrow on document's list header.

Operators/Administrators can also enter a reason or comment.

Display only documents from the last ... days. The amount of days can be modified from `***DATA***\config\docWizz-dw.ini` "RECENTDOCSDAYS=90" line. This will reduce database loading time. If set to RECENTDOCSDAYS=0 all documents are shown, no matter how old they are.

Each document is listed along with its unique **ID**, next **Task**, **Date** of last modification, **Type** (serial or monograph or newspaper) and **Title** of the document. A lock icon indicates a document currently in use.

Whenever a task has been sent to the processing queue, the next step is an automatic process. All these are starting with **Detect...** (exception: SplitDblPages) or Build..., so the operator can identify prepared tasks easily.

If an entry starts with **Verify...** or just **Scan** the related document is apparently not prepared for batch processing but waiting for an operator to be opened up.

### Status/Labels



After selecting one or more documents in the **Document Pool** the status can be changed to a different status by hitting the button.

**Note:** the status *Correcting on Remote*, *Remote QA done*, *Prepared to be sent*, *Wait for correction* and *>in use<* are only visible on the remote system.

#### 32.3.4.1 Reduce-, Restore Pool data

Reduce functionality is used to free space on pool storage.

For reducing storage space in pool, temporary images could be deleted (also cropped images created after MP) and restored if necessary.

- Functionality needs manual actions, just on demand. This is not initialized by workflow dependencies.
- Only administrators can perform the actions while have high impact on pool.

- Tasks must be configured for services. (CCS additional)
- The OnProcess button is disabled if current document is in Restore Pool Data or Reduced Pool Data or Free Pool Data status.
- An image could be restored (e.g. with "Document open") if necessary "on the fly". This will last some time and the user has to wait until document is restored again.
- For safety reasons source images will not be deleted for those pages where the source IN data images are not available at initial path. Only thumbnail images are deleted in this case.
- For restoring the source images must exist in the correct folder (e. g. the IN folder).

In docWizz-dW.ini file, **RESTOREPOOL**, **CLEANPOOL** tasks must be enabled here:

```
[PROCESS]
TASKS= ...RESTOREPOOL,CLEANPOOL, ...
```

The task **CLEANPOOL** cleans pool data when document status is changed to Free Pool Data, and on completion it sets the status to Reduced Pool Data.

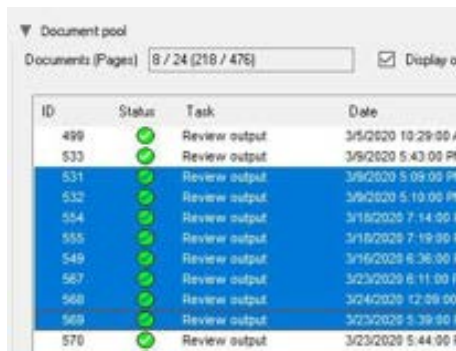
The task **RESTOREPOOL** restores previously reduced documents to their original pool data.

Set time in for the **CLEANPOOL** task in **<customer>-docWizz-dw.ini** file:

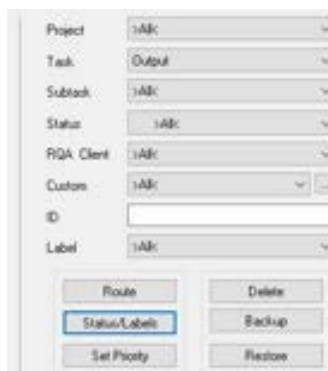
```
[CLEANPOOL.TASK]
PROC=TaskCleanPool
DELAY=300
TIME=10:00:00
TIMETYPE=0
...
```

The manual handling is done in pool dialog and change status dialog.

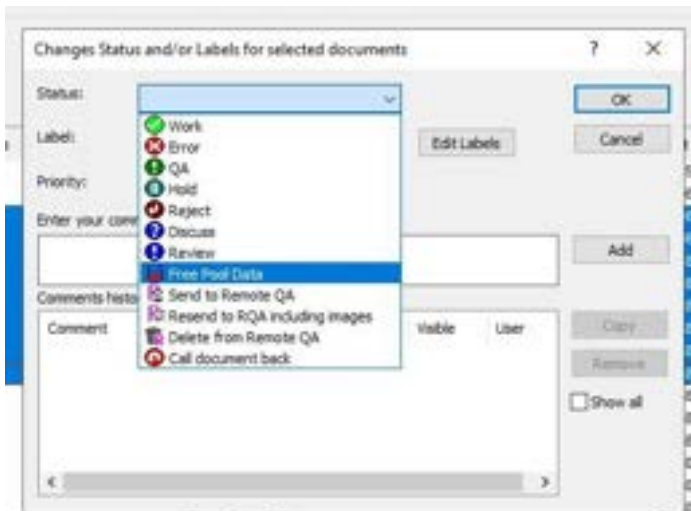
Select one or more documents:



Click Status/Labels button:



Click FreePoolData entry to reduce storage space in pool:



In pool view different icons show status of documents:

150371		Exported	01/04/2011 14:28:00	Newspaper	40	86	11:30
150374		Exported	01/04/2011 14:28:00	Newspaper	20	86	11:30
150377		Exported	01/04/2011 14:30:00	Newspaper	56	86	11:30
150378		Exported	01/04/2011 14:30:00	Newspaper	56	86	11:30
150379		Exported	01/04/2011 14:28:00	Newspaper	60	86	11:30
150380		Exported	01/04/2011 14:30:00	Newspaper	52	86	11:30
150381		Exported	01/04/2011 14:28:00	Newspaper	60	86	11:30
150382		Exported	01/04/2011 14:30:00	Newspaper	60	86	11:30
150383		Exported	01/04/2011 14:28:00	Newspaper	60	86	11:30
150384		Exported	01/04/2011 14:28:00	Newspaper	60	86	11:30
150346		Exported	01/04/2011 14:28:00	Newspaper	28	86	11:30
150366		Exported	01/04/2011 14:28:00	Newspaper	28	86	11:32
150369		Exported	01/04/2011 14:28:00	Newspaper	40	86	11:32
150372		Exported	01/04/2011 14:29:00	Newspaper	48	86	11:32
150375		Exported	01/04/2011 14:29:00	Newspaper	48	86	11:32
150385		Exported	01/04/2011 14:28:00	Newspaper	60	86	11:32
150394		Exported	01/04/2011 14:28:00	Newspaper	60	86	11:32
150396		Exported	01/04/2011 14:28:00	Newspaper	52	86	11:32
150397		Exported	01/04/2011 14:28:00	Newspaper	53	86	11:32
150398		Exported	01/04/2011 14:28:00	Newspaper	56	86	11:32

- The icon shows already reduced pool data. For colored documents about 90% of data can be removed.

Following files will be removed:

All temporary files b/w images

Lowres images

Cropped/aligned images (if they are not changed manually)

RQA images (always)

<jobname>.zip (from non-interactive jobs)

Following files (all non restorable ones) will remain:

<jobname>.zip (from interactive jobs)


ID.xml

rescan images

deskewed images

- The icon shows document to be reduced.



- The  icon shows restored pool data. Restores all temporary images data, as were existing before. Uses the task RESTOREPOOL. The document status is set back to the status that was set before the document was sent to reduce data status.

#### 32.3.4.2 Properties - History

The **History Dialog** shows interactions done with the current selected document plus time stamp and operator.

If you click on the menu entry **View History** the **Document Processing History** window opens:



The column "pages" helps to analyse changes of document content over the processing flow.

Select a row and click with right mouse button. You can now copy a single cell or a row to insert the content in an external text editor (e.g. MS Word, E-Mail application).

History can be sorted by clicking on the column header (Date, Duration, etc.)

The Action column shows how many clicks are done by keyboard (K ) or how many mouse clicks on keyboard are used (M ). All clicks are counted here no matter if they are useful or not.

Actions and Pages: is helpful in big productions to supervise the changes in a document and for operator statistics and training.

Notes like "here is a missing entry that says that document stopped because of an error" will occur. This is happening because document's history is stored in two places: HISTORY node from documentID.xml (from where you've extracted the xml sample) and also in log database (BatchResults table, configured in config\docWizz-glbl.ini file, [ERRORLOGDB] section) which can be accessed from Document -> View History menu for the current document or by pressing History button in Document Pool Window.

This log database contains more informations than documentID.xml and it is recommended you want a full/complete processing history.

#### 32.3.4.3 Custom filters

To have a more specific view on pool, custom filters for Pool dialog are configurable. They contain a pair of displayed name and a fragment of a WHERE expression from SQL select statement. Administrator users can define new filters within UI. Filters will be selectable in a combo box and can be combined with any other filter.



The button with the three dots opens a separate window where you can define custom filters.



Use the  button to create a new filter.



Each document is listed along with its unique **ID**, next **Job**, **Date** of last modification, **Type** (serial or monograph or newspaper) and **Title** of the document. A lock icon indicates a document currently in use.

Whenever a job has been sent to the processing queue, the next job is an automatic process. All these are starting with **Detect...** (exception: SplitDbIPages) or Build..., so the operator can identify prepared jobs easily.

If an entry starts with **Verify...** or just **Scan** the related document is apparently not prepared for batch processing but waiting for an operator to be opened up.

The pool folder structure can be extended to two levels of folders to improve performance on mass digitization projects.

When changing a filter manually it is checked whether new document type is available.

Here you define disk space limits for different tasks and locations and set "critical disk space" values for "low space" warnings.

It is not the case that the limit set is the space that we guarantee it will remain free. If the space limit is reached during one document processing then that document processing will be finalized and will fill the limited free space.

We suggest to set a limit considering the number of services (e.g limit = number of services \* 50Gb).

A feature for local export can be customized, if no TEMPFREESPACE node is present in LowDisk.xml, then the default temp space value is considered (2 GB by default).

It is available in configuration as a comment, and if needed it can be decommented and adapted by need. This function will not interfere with the disk space check the Control Center does in its UI for export and pool folders.

## 32.5 Environment control

Here you check and edit notes people should pay attention to, check and manage error log or create reports and detailed statistics.



### 32.5.1 Statistics



#### Computers

The **Computers** statistic view shows efficiency of labor. You can sort by subgroups like name of pc or **Jobs**. Under this selection is splitted to **automatic** or **interactive**. There you can choose between a single work procedure or a specific document.

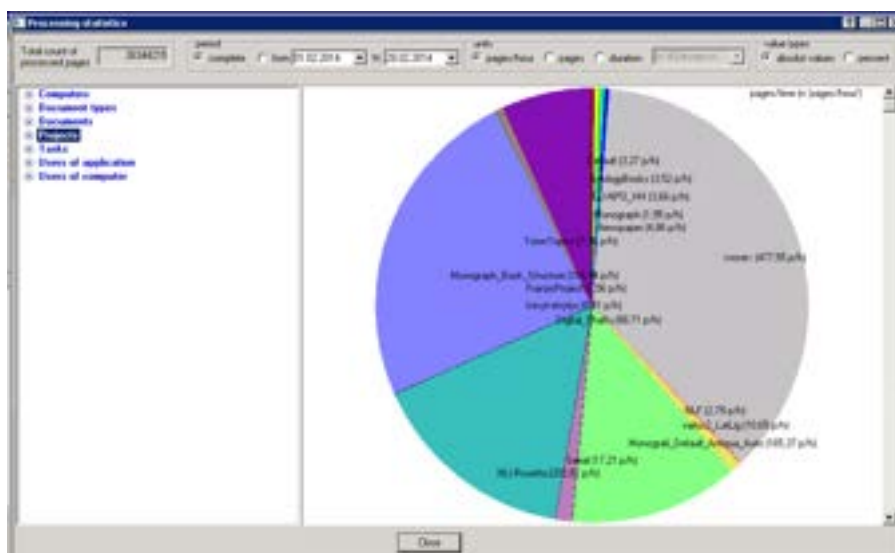
#### Document Types

Document types can be **Monograph**, **Multivolume Monograph** or **Serial**.

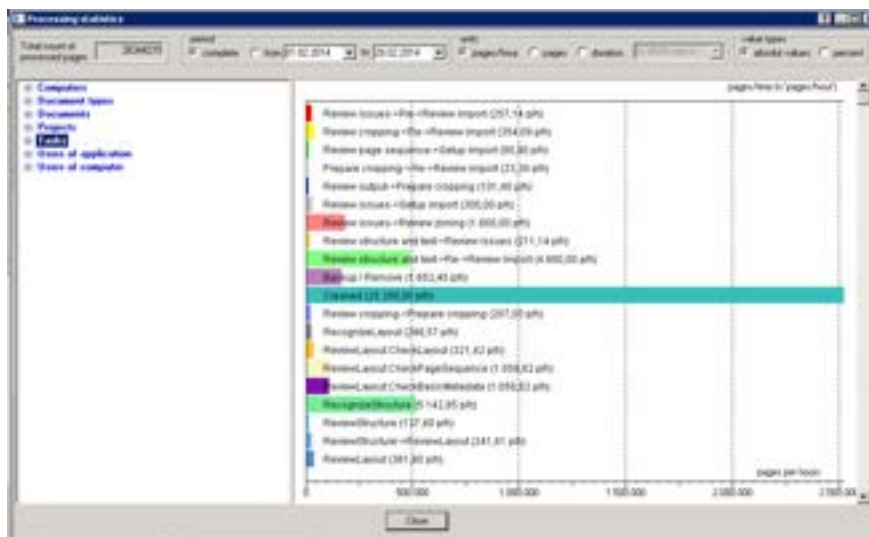
#### Documents

You can sort **Documents** statistic view by name of the processed documents. There you can split results by **Computers**, **Jobs**, **Projects**, **Users of application** or **computer**.

#### Projects



## Tasks



## Users of application

## Users of computer

### 32.5.2 Health status



## Notes

Click first to **Edit** to enable the notes entry area.

Click **Save** or **Cancel**.

### 32.5.3 Error log



Enables you to refer to the **Error log** window that automatically lists any errors that have occurred during the current session. In this way, support staff and docWizz administrators have optimal support when looking for the cause of irregularities in the program.

Sometimes, processing fails due to depleted memory. In many of this cases, restarting DWSrv will solve the problem.

If this error occurs, the document will not get error status but remains in the current job to be performed. DWSrv will restart and the document will run again through this job. As soon the document has 5 or more failures, it will be set to error status anyway.

### 32.5.4 Environmental functions



#### Restore documents



See a list of documents in the restore queue and restore documents and batches.

#### Clear log files



Clears the log files.

#### Set services logon



Set user and password.



## Volume report



By hitting the Volume Report button, a PDF file containing total page counter and number of pages processed in the selected month is created. For more information please refer to the **Volume report** chapter.

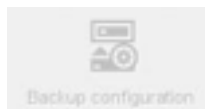


## Statistics



With the Statistics tool you get statistical records of the docWizz system. There are different ways to analyze work procedures, jobs or documents. The statistic of docWizz visualizes the logged data about the processed documents and used time, inform about the behavior of the machines and users and so on. See **Statistics** chapter.

## Backup configuration



Use Backup Configuration to make a backup of the current configuration. This feature is still under development and will be available in further releases of the Control Center.

### 32.5.5 Volume report

#### What's the Volume report?



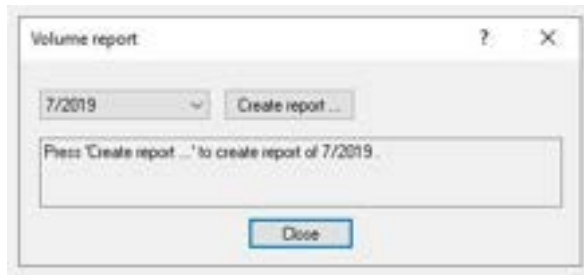
The VolumeReport is an additional function in docWizz that creates an pdf file containing the total page counter and the number of pages of the selected month.

#### How to create the Volume report?

The **Volume report** is positioned at Configuration -> Maintenance. Inside this dialog you can choose **Volume Report**.

A new dialog is opened where you can select the desired period of which you want to create the VolumeReport.

It is selected the minimum date of the BatchResult table and generated all month until last month of current date.



After creating the report by click on the button, the location of the stored PDF-file will be shown in the text area below. The default location of the VolumeReports is **\*\*\*MAINTENANCE\*\*\***. You can change the location in the system configuration in the register "paths". The path name is "MAINTENANCE".

### The content of VolumeReport

The VolumeReport contains the following informations:

- short name of the customer
- total page counter
- date of the selected period
- number of processed pages in the selected period
- select completed Pages from BatchResult where date=actualPeriod and JobName='ExportXML'
- two validation codes
- total page counter and number of pages in this period encoded by Base64Encoder
- list of processed pages for each job

### Sending the VolumeReport to CCS

At the moment there exists no automatic transfer of these information to CCS in any way. The generated PDF-file has to be send by mail to [metae-support@ccs-gmbh.de](mailto:metae-support@ccs-gmbh.de).

### How to prove the validation of the VolumeReport

The propertied of the PDF file contains the following information

- title VolumeReport
- creator CCS newsWORKS/docWizz
- theme <customerName>
- created at <actual system time>
- application CCS newsWORKS/docWizz
- creator CCS Content Conversion Specialists GmbH, Hamburg
- created by docWizz <FVersion>

### Behavior in case of errors

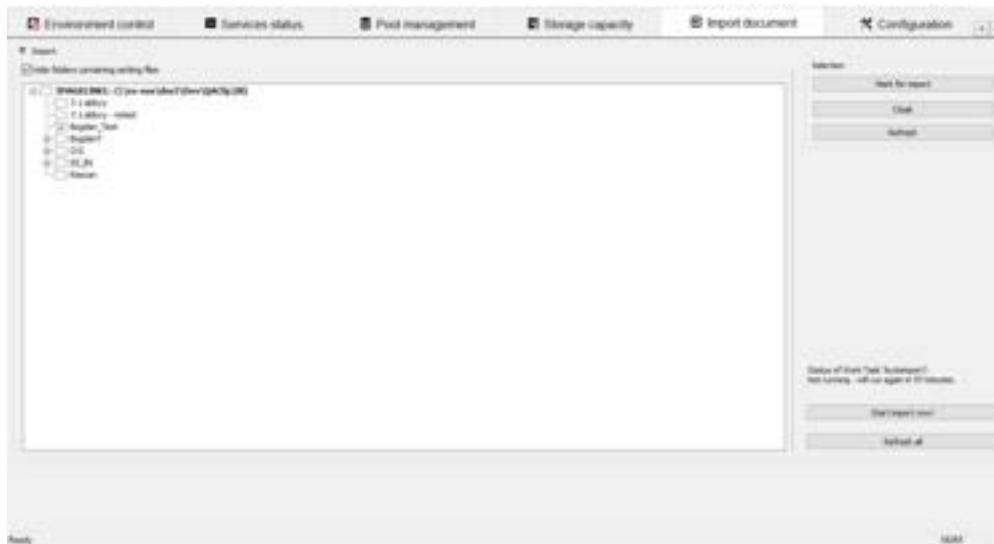
The VolumeReport dialog contains an own filter for errors. If an unexpected error appears the dialog would be closed without a message. Perhaps in the log database an error will be logged. The user can continue his work with docWizz without any problems. In case of expected errors (f.e. error in database requests) the error will be shown and logged in the log database.



## 32.6 Import document

Here you prepare folders for import, trigger import of documents and check the import status.

This method to import documents into docWizz is recommended for importing larger amount of documents in batch mode.

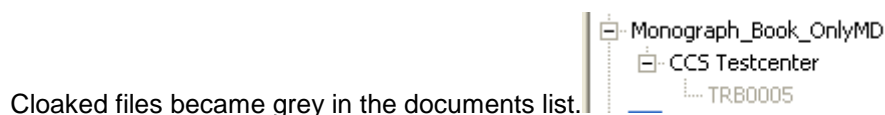


This window shows a tree structure. The different projects are shown on top level, what is scanned on second level and the status on third level.

File formats supported: \*.tif; \*.tiff; \*.jpg; \*.jpeg; \*.jp2; \*.pdf; \*.cr2; \*.png; \*.bmp; \*.gif to set ready. The same extensions are supported in the import script.

Select a document and click on Mark for import button to import the documents into docWizz. If you select a project (top level) all documents on lower levels are also set ready with one click.

There is also a button which is called Cloak to create cloaked files to block parsing of current and all subfolders to improve import task performance. Press button to block parsing of current and all sub folders. When a folder has the files "cloaked.rdy" and "cloaked.wrk", the auto-import tasks will not verify this folder and its subfolders for new documents. This can help speed up the task.



Cloaked files became grey in the documents list.

Use the Refresh button for refreshing the list of new documents that are ready to be imported..

Use the Set import now! button to trigger the import task in background. This forces the import tasks and user has not wait for the standard two hours to perform the task. It does not start the import at once.

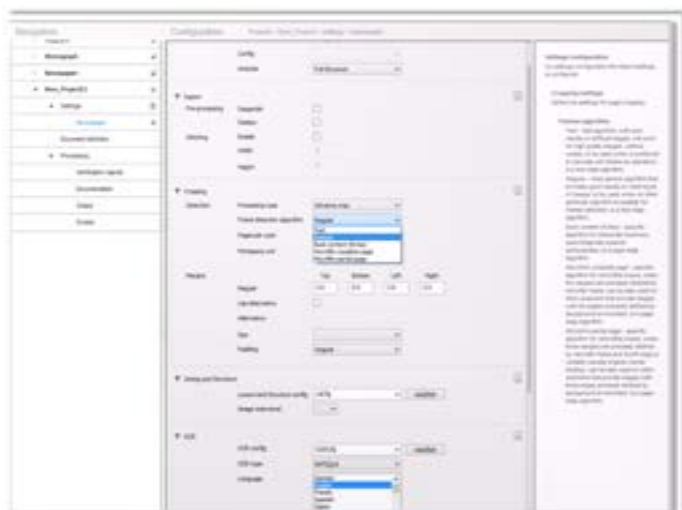
Use the Refresh all button for refreshing without restarting the tool.

It is also possible to store in the IN directory a special Ready file (\*.rdy) in which you can make settings to your needs. docWizz checks this file and processes the files automatically. So you can for example define that the Review import step is skipped.

## 32.7 Configuration

The configuration tool shows an user friendly interface and gives a full overview of your project configurations.

The left side is used for navigation inside / between the projects, center view is used to display the settings and right side will display the help for hovered controls.



### 32.7.1 Projects

All projects available in **PROJECT-CFG** folder are displayed in the list.

#### Create new project



To add a new project, press the button . This will create a standard project configuration.

Create a standard project configuration.

Add the desired name, project template (Monograph, Newspaper, Serial, SimpleMonograph, SimpleNewspaper or SimpleSerial) and document type (Monograph, Newspaper or Serial).

#### **Monograph**

Document type used for regular books processing.

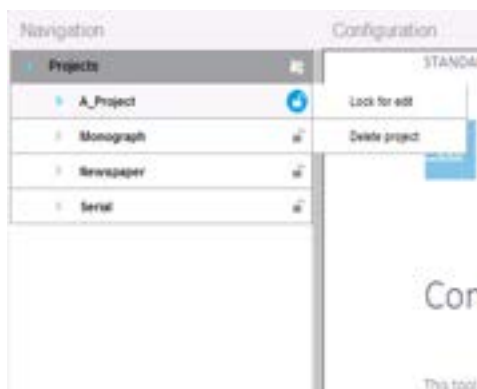
#### **Serial**


Document type used for serials processing.

#### **Newspaper**

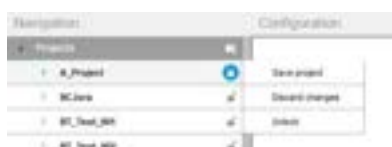
Document type used for newspapers processing.


## Current project




To start editing a project, hover over the "lock" icon  for the desired project and select "Lock for edit". This will set the project in editing mode; at the same time, no other user is allowed to modify the project.

To delete a project, hover over the "lock" icon and select "Delete project". This option is only available if the project is not in editing mode.



To make changes to a project, hover the "lock" icon  and select "Save project" or "Discard changes". The actions will still keep the project in edition mode.

In order to unlock the project, select "Unlock" option from the same menu.

If a project has the icon , it means it's currently being edited by someone else. Hovering over the icon will display who is editing the project. This project cannot be unlocked by you.

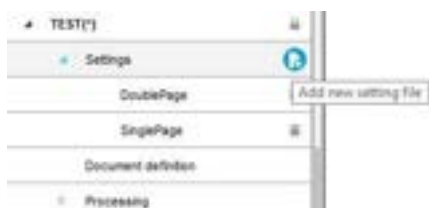
## 32.7.2 Settings files



Once the project is in editing mode, the button



becomes available. This is used to add a new setting file in the selected project.



The setting file will have the standard values and can be adapted for the project.

To remove the existing settings files, press the button



. It is used to delete the settings file.



## Settings configuration



### General

Define general document settings.

### Import

Define the settings for image import.

### Cropping

Define the settings for page cropping.

### Zoning and Structure

Define the settings for layout analysis.

### OCR

Define the settings for OCR detection.

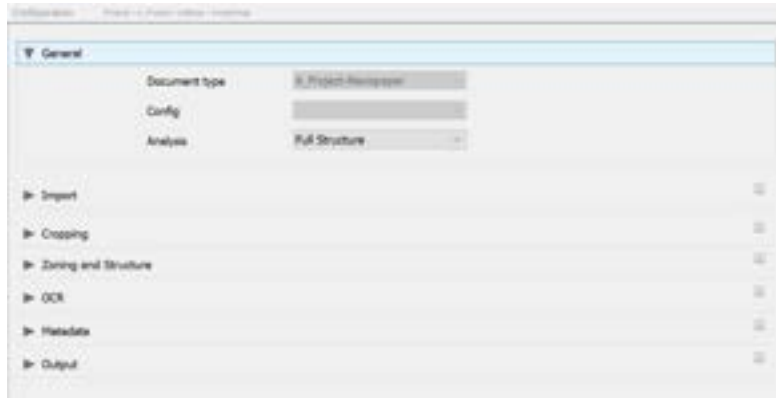
### Metadata

Define metadata standard used into output.

### Output

Define the settings for output.

### 32.7.2.1 General



#### Document type

Select the document type. There are available three basic document types: Monograph, Serial or Newspaper.

#### Config

Field read-only, currently obsolete, used for backward compatibility only.

#### Analysis

Select a specific analysis type to be used for document processing.



#### None

No analysis is done. After Cropping, the documents are exported. As output you only receive here the defined images with reference in the METS (no physical or logical structure generated).

#### Page Linking

Zoning, OCR and Metadata are disabled. After Cropping, the documents stop in Review page sequence to perform the page linking. As output you only receive here the defined images referred in METS with physical structure outlining the page numbers.

#### Page Linking + OCR

Disables Metadata group. Cropping, Zoning and OCR are done before exporting the document.

#### Metadata

Disables Zoning and OCR. Cropping and Metadata are done before exporting the document.

#### Metadata + OCR

Cropping, Zoning, OCR and Metadata are done before exporting the document.

#### Full Structure

All groups are available. In addition, the hierarchy correction needs to be done.

### 32.7.2.2 Import

Define the settings for image import.

The screenshot shows the 'Import' settings panel. It includes a 'Pre-processing' section with 'Despeckle' and 'Deskew' checkboxes, and a 'Stitching' section with an 'Enable' checkbox, 'Width' and 'Height' input fields (both set to 1). Below this are expandable sections for 'Cropping', 'Zoning and Structure', 'OCR', 'Metadata', and 'Output', each with a small icon to its right.

#### Pre-processing despeckle

Remove small defects due to dust or scratches; used for bitonal images.

#### Pre-processing deskew

Adjust the angle, in case of skewed images.

#### Stitching

Enable stitching two halves of a page into a single page.

### 32.7.2.3 Cropping

Define the settings for page cropping:

The screenshot shows the 'Cropping' settings panel. It includes a 'Detection' section with 'Processing type' (Basic crop), 'Frame detection algorithm' (Regular), 'Pages per scan' (Single page), and 'Printspace unit' (millimeter). Below this is a 'Margins' section with 'Regular' (Top: 2.0, Bottom: 2.0, Left: 2.0, Right: 2.0), 'Use alternative' (checkbox), 'Alternative' (input fields), and 'Padding' (Original). Below this are expandable sections for 'Zoning and Structure', 'OCR', 'Metadata', and 'Output', each with a small icon to its right.

#### Detection

- Processing type

The screenshot shows the 'Processing type' dropdown menu. The options are: None, Basic crop, Advanced crop, Clean page borders, and Verify. The 'Basic crop' option is currently selected.

### None

No cropping is done.

### Basic crop

Simple cropping; all frames are set as individual.

### Advance crop

Complex cropping; all frames are available, also content area is used. Not available for Starter version.

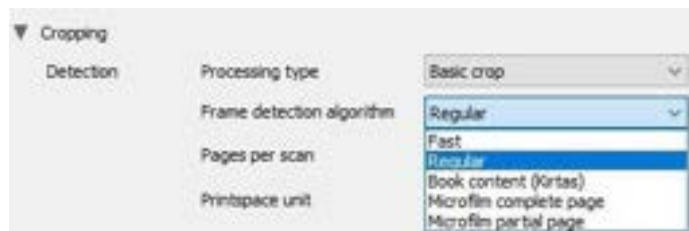
### Clean page borders

No cropping is done; the frames are used to clean the page borders. Not available for Starter version.

### Verify

No cropping is done; images are only checked. Not available for Starter version.

- **Frame detection algorithm**



### Fast

Fast algorithm, with poor results on difficult images; will work for high quality images, without noises; to be used when is preferred to manually set frames by operators; is a text edge algorithm.

### Regular

Most generic algorithm that provides good results on most types of images; to be used when no other particular algorithm is suitable for frames detection; is a text edge algorithm.

### Book content (Kirtas)

Specific algorithm for Kirtas-like scanners; uses Kirtas-like scanner particularities; is a page edge algorithm.

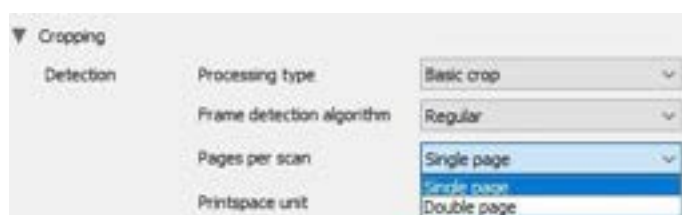
### Microfilm complete page

Specific algorithm for microfilms scans, when the margins are precisely defined by microfilm frame; can be used, also, on other scanners that provide images with the edges precisely defined by background environment; is a page edge algorithm.

### Microfilm partial page

Specific algorithm for microfilms scans, when three margins are precisely defined by microfilm frame and fourth edge is variable (usually original volume binding); can be also used on other scanners that provide images with three edges precisely defined by background environment; is a page edge algorithm.

- **Pages per scan**



### Single page

Only one frame will be detected on images.

### Double page

Two frames will be detected on images.

- **Printspace unit**



The screenshot shows the 'Cropping' section of a settings interface. It includes a 'Detection' sub-section with the following settings:

- Processing type: Basic crop
- Frame detection algorithm: Regular
- Pages per scan: Single page
- Printspace unit: A dropdown menu with 'millimeter' selected and 'inch' as an alternative option.

Measurements units used for cropping details interface (milimeter/inch).

## Margins



The screenshot shows the 'Margins' section of the settings interface. It includes a 'Regular' section with the following settings:

- Top: 2.0
- Bottom: 2.0
- Left: 2.0
- Right: 2.0

Below the 'Regular' section, there is a 'Use alternative' checkbox which is currently unchecked. Below that, there is an 'Alternative' section with four empty input fields for Top, Bottom, Left, and Right margins. At the bottom, there is a 'Padding' dropdown menu set to 'Original'.

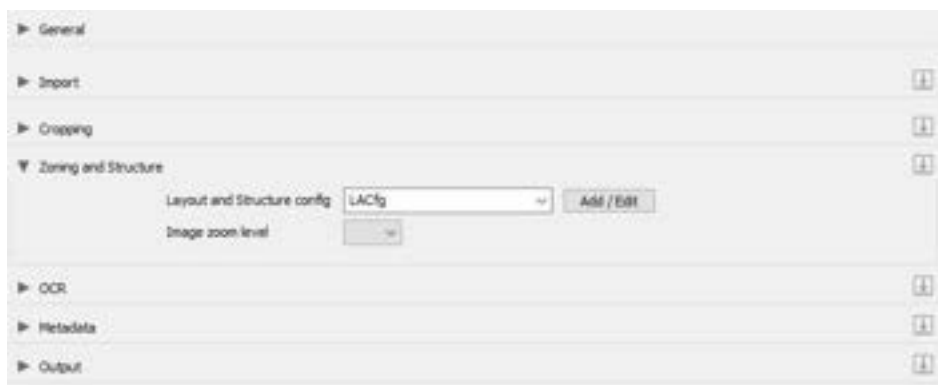
Enable alternative content area margins.

Used to set different margins for alternative frames than for regular frames.

If it's not enabled, the alternative frames will have the same margins as regular frames.

### 32.7.2.4 Zoning and Structure

Define the settings for layout analysis:



The screenshot shows the 'Zoning and Structure' section of the settings interface. It includes the following settings:

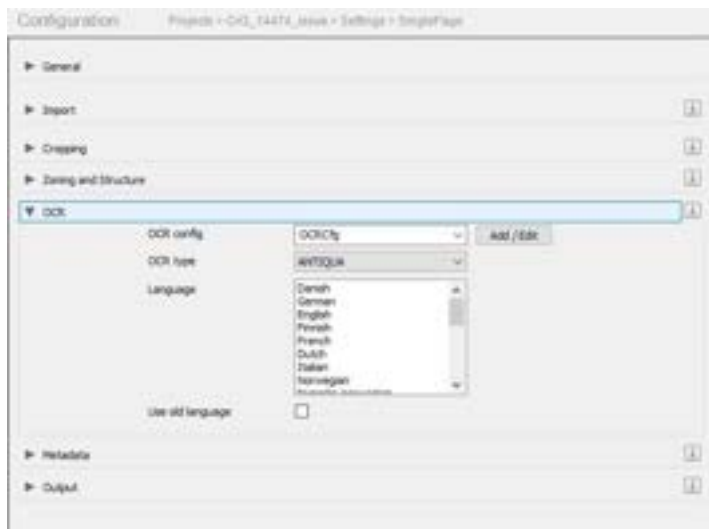
- Layout and Structure config: LAClg
- Image zoom level: A dropdown menu.

There is an 'Add / Edit' button next to the 'Layout and Structure config' dropdown. The interface also shows a list of other settings sections on the left: General, Import, Cropping, Zoning and Structure, OCR, Metadata, and Output.



### 32.7.2.5 OCR settings

Define the settings for OCR detection:



#### **OCR config**

Set the name of OCR configuration file.

#### **OCR type**

Specify the OCR font types.

#### **Language**

Select language(s) for OCR detection.

#### **Use old language**

Specify if text contains old style of languages.

#### 32.7.2.5.1 OCR configuration file

Set the name of OCR configuration file:



For more OCR settings press the button

Add / Edit

Define the "OCR engine", "Post OCR operations", "OCR parameters (Finereader specific)", "Zone based settings" and click the button

Done

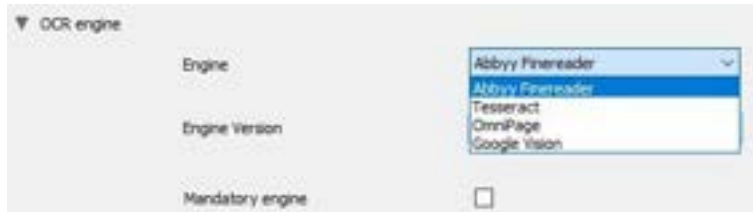


- **OCR engine**

Define OCR engine type, OCR engine version and OCR mandatory engine.

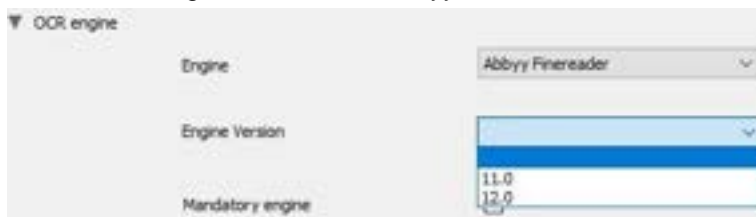
### Engine

Select OCR engine type Abbyy Finereader / Tesseract / Omnipage / Google Vision:



### Engine version

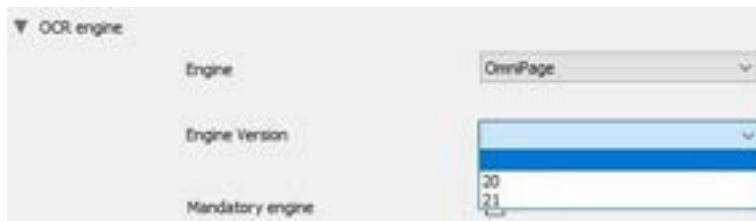
Select OCR engine version; for Abbyy Finereader, the available versions are 11.0 and 12.0.



Select OCR engine version; for Tesseract, the available version is 4.1.




Select OCR engine version; for Omnipage, the available versions are 20 and 21.



Note: Omnipage20 is removed from docWizz starting with version 7.2.0.22

### Mandatory engine

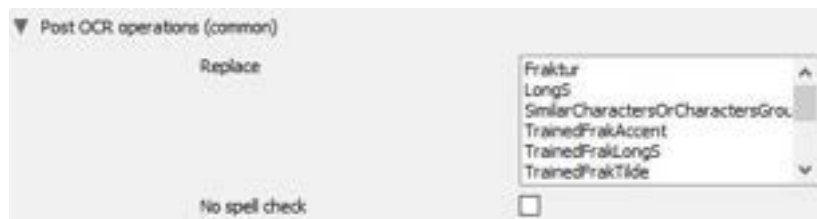


If this option is enabled, the engine is mandatory to be used for processing.

If this option is disabled, the engine selected may not be used for processing and a better one can be selected.

- **Post OCR operations (common)**

Define OCR common settings.

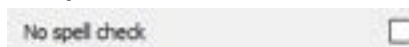


### Replace



Select one or more docWizz user defined replacement tables used in post OCR processing (Fraktur, LongS, SimilarCharacterOrCharacterGroup, TrainedFrakAccent, TrainedFrakLongS, TrainedFrakTilde, Default\_REPLACE, LatvianCharWithoutSpellcheck, LatvianCharWithSpellcheck, LDD\_REPLACE, LVGothic, etc).

### No spell check

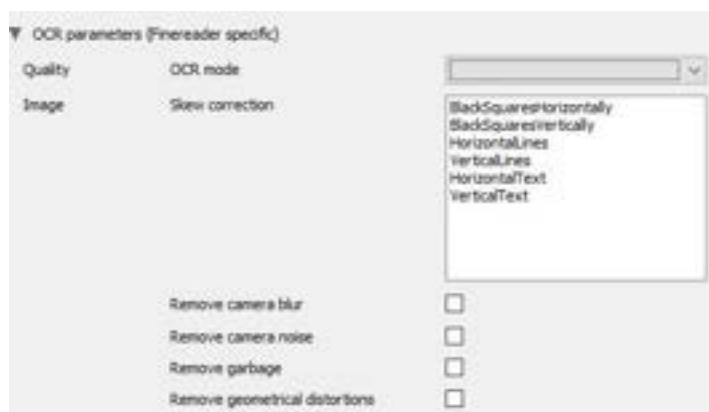


If this option is enabled, post OCR processing spell check is not done.

If this option is disabled, spell check correction is done and spell check results are visible on OCR correction interface, in "Review structure and text" step.

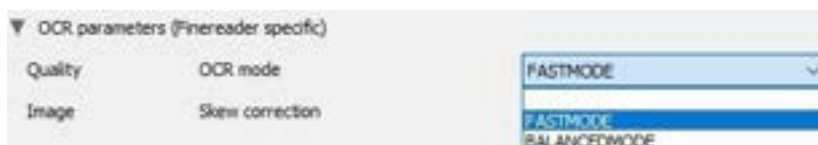
- **OCR parameters (Finereader specific)**

Available only if selected engine is Abbyy Finereader.



### OCR mode

OCR can be done in 3 modes:



#### Full mode

This mode provides full recognition, slower recognition speed than fast mode and better quality.

#### FASTMODE

This mode provides 2-2.5 times faster recognition speed at the cost of a moderately increased error rate (1.5-2 times more errors).

#### BALANCEDMODE

In this mode the recognition will run in balanced mode. The balanced mode is an intermediate mode between full and fast modes.

## Skew correction



Image pre-processing; corrections applied to image:

### **Black squares horizontally**

The image skew angle is corrected based on so-called "black squares" (the skew angle is calculated based on the horizontal pairs of squares). Black squares are often placed on forms. We recommend that you use this constant only when working with images of forms, otherwise you may obtain incorrect results.

### **Black squares vertically**

The image skew angle is corrected based on so-called "black squares" (the skew angle is calculated based on the vertical pairs of squares). Black squares are often placed on forms. We recommend that you use this constant only when working with images of forms, otherwise you may obtain incorrect results.

### **Horizontal lines**

The image skew angle is corrected based on horizontal lines. We recommend that you use this constant only when working with images, which contain horizontal lines (e.g. invoices, price lists, or other documents, which contain tables with visible borders), otherwise you may obtain incorrect results

### **Vertical lines**

The image skew angle is corrected based on vertical lines. We recommend that you use this constant only when working with images, which contain vertical lines (e.g. invoices, price lists, or other documents, which contain tables with visible borders), otherwise you may obtain incorrect results.

### **Horizontal text**

The image skew angle is corrected based on horizontal text lines.

### **Vertical text**

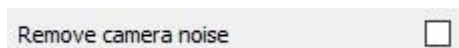
The image skew angle is corrected based on vertical text lines. The constant may be useful when working with documents in Chinese, Japanese, or Korean language, or if page orientation is incorrect.

## Remove camera blur



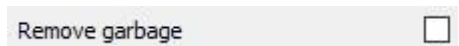
If enabled this setting removes motion blur from the specified region of the image.

## Remove camera noise



If enabled this setting removes ISO noise from the specified region of the image.

## Remove garbage



If enabled this setting removes garbage (excess dots that are smaller than a certain size) from the image.

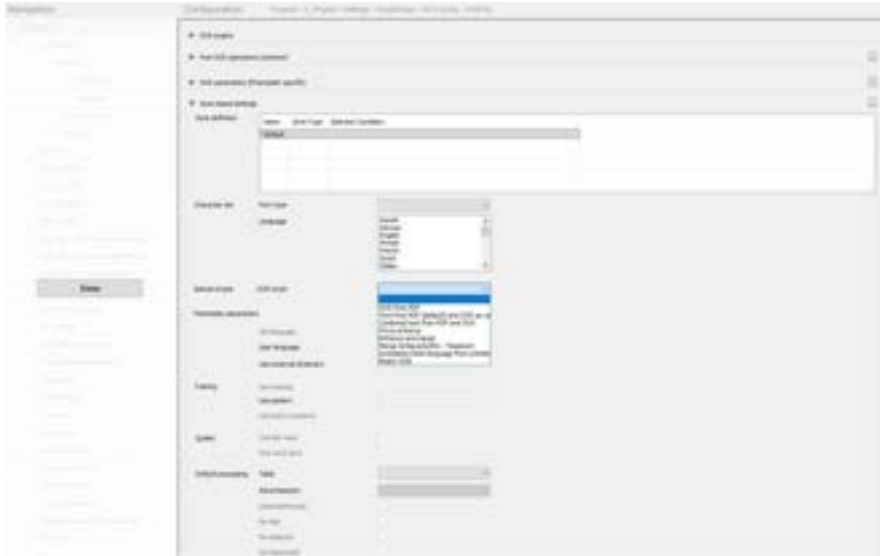
## Remove geometrical distortions

Remove geometrical distortions ☐

If enabled this setting straightens out distorted lines on an image. Distorted lines may occur close to the binding when scanning/photographing thick books.

## Zone based settings

OCR zone settings:



## OCR scripts

Enable this to run OCR with specialized script from the dropdown list:

### OCR from PDF

Get the text from the PDF file, if no result from PDF then normal OCR is done on the image.

### Text from PDF (default) and OCR as variant

Get the text from PDF and run OCR. Set the text from PDF as main text and set as variant the OCR result.

### Combined text from PDF and OCR

Get the text from PDF and run OCR. Set as main text for each line as best between text from PDF and OCR (based on typical PDF text issues). The alternative text is set as variant.

### Force enhance

Performs OCR on the enhanced image.

### Enhance and merge

Perform OCR on the enhanced image and merges it with the normal image result based on confidences.

### Merge Antiqua/Gothic

Tesseract - Performs OCR on the images with Tesseract engine, and chooses best confidence between Antiqua and Gothic (needed since Tesseract do not perform font autodetection).

### Autodetect best language from a limited list

Use it whenever you have a document with more than two languages inside (is configured for most used European languages; list can be adapted directly into TCL script).

### Empty OCR

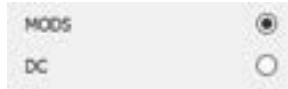
Returns always empty OCR result.

### 32.7.2.6 Metadata

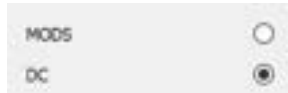
Defines metadata standard used into output.



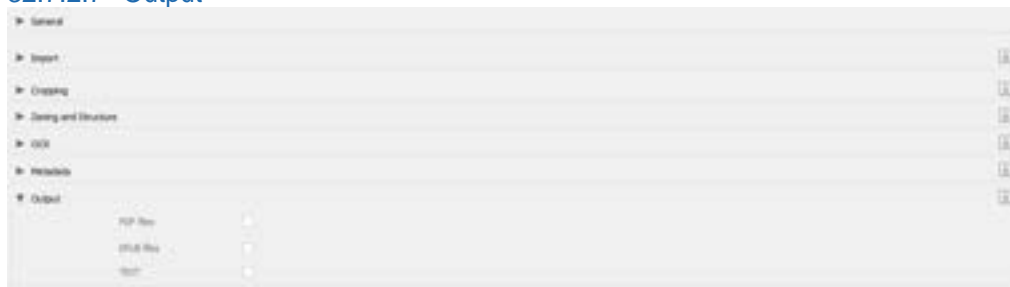
**MODS** - MODS standard is used for metadata records.



**DC** - Dublin Core standard is used for metadata records.



### 32.7.2.7 Output



This settings is kept into interface to view former project configuration settings.

Output files can be enabled / disabled into Output panel.



#### **PDF files**

Export PDF files

#### **EPUB files**

Export EPUB files

#### **TEXT**

Export TEXT files

### 32.7.3 Document definition

Define additional page, zone and structure types.

▼ Document definition

Document type: TEXT Monograph

Page types

Page name	Base type	

Zone types

Zone name	Base type	

Define zones colors

Structure types

Structure name	Base type	

- TEST[""]
  - Settings
  - Document definition
  - Processing
  - Verification rejects
  - Documentation
  - Output
  - Scripts

## Verification rejects

Import Settings

Test Case ID:  Filter:

Test Case ID	Test Case Name	Test Case Description	Test Case Status
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6
7	7	7	7
8	8	8	8
9	9	9	9
10	10	10	10
11	11	11	11
12	12	12	12
13	13	13	13
14	14	14	14
15	15	15	15
16	16	16	16
17	17	17	17
18	18	18	18
19	19	19	19
20	20	20	20
21	21	21	21
22	22	22	22
23	23	23	23
24	24	24	24
25	25	25	25
26	26	26	26
27	27	27	27
28	28	28	28
29	29	29	29
30	30	30	30
31	31	31	31
32	32	32	32
33	33	33	33
34	34	34	34
35	35	35	35
36	36	36	36
37	37	37	37
38	38	38	38
39	39	39	39
40	40	40	40
41	41	41	41
42	42	42	42
43	43	43	43
44	44	44	44
45	45	45	45
46	46	46	46
47	47	47	47
48	48	48	48
49	49	49	49
50	50	50	50
51	51	51	51
52	52	52	52
53	53	53	53
54	54	54	54
55	55	55	55
56	56	56	56
57	57	57	57
58	58	58	58
59	59	59	59
60	60	60	60
61	61	61	61
62	62	62	62
63	63	63	63
64	64	64	64
65	65	65	65
66	66	66	66
67	67	67	67
68	68	68	68
69	69	69	69
70	70	70	70
71			

**Report settings**

Task: Report missing Editing task: Report missing

**Autofill list**

Autofill name	Enabled	Step instruction
Default_TestReport	<input type="checkbox"/>	<input type="checkbox"/>

**Report list**

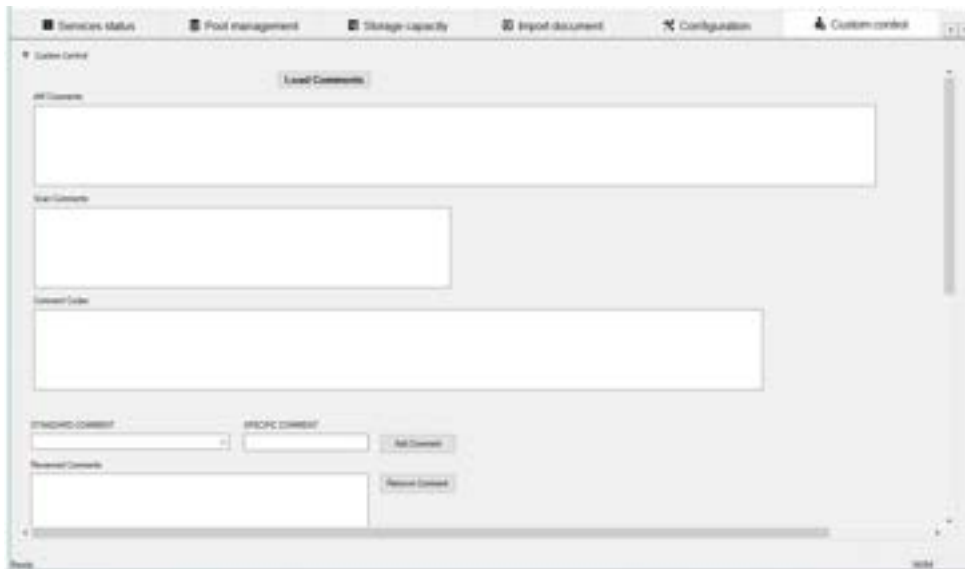
Report name	Enabled	Critical
Default_CheckImageSize	<input type="checkbox"/>	<input type="checkbox"/>
Default_CheckImageResolution	<input type="checkbox"/>	<input type="checkbox"/>
Default_CheckImageFrameSize	<input type="checkbox"/>	<input type="checkbox"/>
Default_CheckImageFrame	<input type="checkbox"/>	<input type="checkbox"/>
Default_CheckImageFrameSize	<input type="checkbox"/>	<input type="checkbox"/>
Default_TestImageSize	<input type="checkbox"/>	<input type="checkbox"/>
Default_TestImageResolution	<input type="checkbox"/>	<input type="checkbox"/>
Default_TestImageFrameSize	<input type="checkbox"/>	<input type="checkbox"/>
Default_TestImageFrame	<input type="checkbox"/>	<input type="checkbox"/>

**Parameters list**

400



### 32.7.5 Custom control



Extra custom tab for defining/scripting special features like e.g.:

- start ingest in external repository
- delete data older than 100 days
- etc.

Right click into the background to activate the **Design Mode**.




Right click to get the context menu to create check boxes, buttons, system controls and others:



### Properties

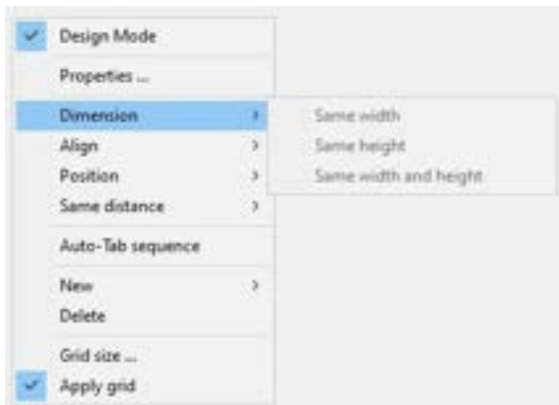
The **Properties** dialog provides a variety of tabs for making specific settings.



Here you can specify the type and label of the graphics element, as well as other attributes. Confirm and exit by pressing .

## Dimension

If you have selected multiple elements in the dialog box with the Shift key and the mouse, the **Dimension** function allows you to standardize the size of all the elements at the same time. Placing the mouse pointer on the **Dimension** function, another selection menu offers three commands:



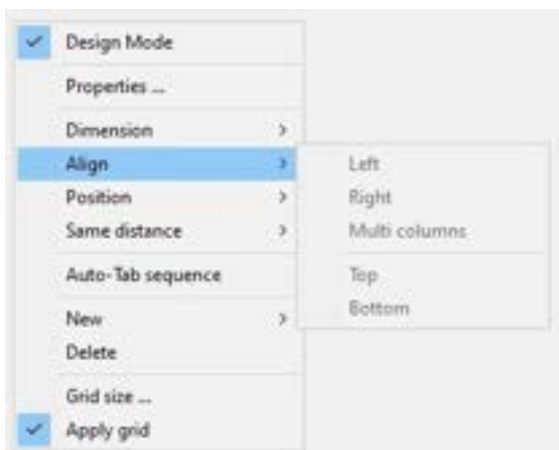
With the function **Same Width**, you can scale all the selected elements to the same width.

With the function **Same Height**, you can scale all the selected elements to the same height.

With the function **Same Width and Height**, you can scale all the selected elements to the same width and height.

## Align

If you have selected multiple elements in the dialog box with the Shift key and the mouse, the **Align** function allows you to align all the elements you have selected. Placing the mouse pointer on the **Align** function opens a selection menu beside the arrow that offers 5 commands:



The **Left** command aligns the marked elements to the left.

The **Right** aligns the marked elements to the right.

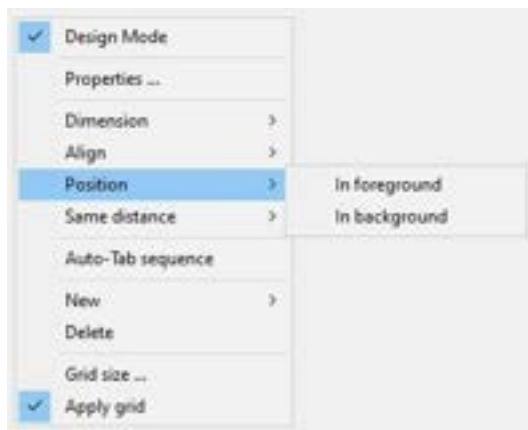
The **Multi Columns** command aligns the marked elements in multiple columns.

The **Top** command aligns the marked elements along the top.

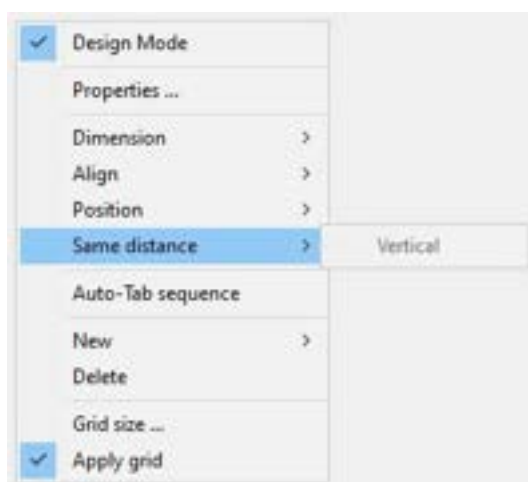
The **Bottom** command aligns the marked elements along the bottom.

## Position

Use the **Position** function to place the element you select in the foreground or background. Placing the mouse pointer on the **Position** command opens a selection menu with two commands:



## Same distance

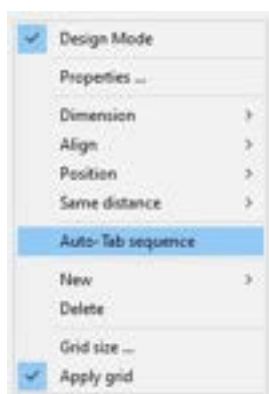


With the **Same Distance** function, you can specify whether the vertical separation between the selected elements should be uniform.

Placing the mouse pointer on the **Same Distance** function opens a selection menu offering the following command:

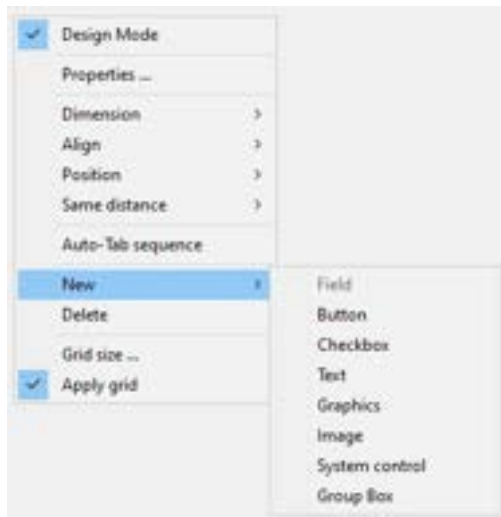
**Vertical**, means same distance in vertical dimension.

## Auto Tab Sequence



You use the **Auto Tab Sequence** function to have automatically set the jump sequence for addressing the control elements when the **Tab** key is pressed. There is also the possibility to determine the order manually.

## New



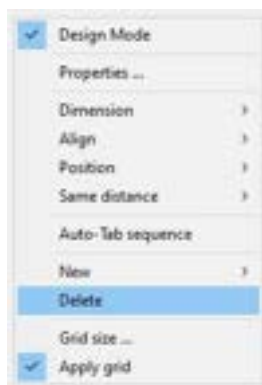
You can add a new element to the dialog box.

With these functions, you are able to create different elements like **Field**, **Button**, **Checkbox**, **Text**, **Graphics**, **Image** and **System Control** buttons.

Example: You use the **Graphics** button to enter graphics elements you want - backgrounds or frames - in the dialog box. These elements are for appearance only, and have no function. Click the button and place the mouse pointer where you want the graphics element to appear in the mask. Draw a frame by holding down the left mouse button and then click the frame with the right mouse button. The context menu appears.

Clicking the **Properties** function opens the **Properties** dialog box for the graphic element for example.

## Delete



You can delete the selected element with the **Delete** function.

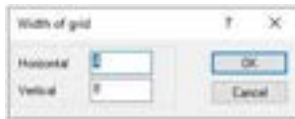
## Grid Size



You set the size of the grid the system uses for orientation purposes with the **Grid Size...** function.

Clicking on this function opens an input mask in which you specify the desired horizontal and vertical spacing between the grid lines.

Make your settings in millimeters:



Use the **Apply Grid** to turn the grid on and off. The check mark beside the menu item indicates its status.

### 32.7.6 Maintenance

The buttons **Backups to Restore** and **Recover documents** offer same functionality that you already know from the document pool (see **docWizz User's Manual**).

Additionally, for system maintenance purposes it can start/stop or shut down all docWizz services at once by hitting the appropriate button. When **stopping all services** the current job will be finished and services stop thereafter.

By hitting **Shut Down All** all services are stopped at once without finishing their current jobs; documents will bounce back to job before. After maintenance has been performed, all services can be started again by hitting **Start all Services**.

In both cases "MaintenanceUnderway" will be set to "1", so no service shall start automatically. Only when manually start a service via MMC, it shall run.

Here the possibility is given to announce yourself as administrator. The **Change Login** function lets another user (the administrator, for example) log in during a docWizz session without requiring the current user to exit the program.

With **Clear All Logs** button you clear all log files. When logs grow in size, they slow DWControlCenter, so when they are not necessary anymore is better to delete them.

**Maintenance** is shown as status for services if they are in maintenance mode.

There is an icon to show services that process documents but will go to maintenance mode when finished.

The file MAINTENANCEUNDERWAY.wrk is only temporarily available in the \WORK directory, if Maintenance mode is switched on. There is no \*.ini or configuration entry for that.

#### 32.7.6.1 Maintenance Mode and its Interaction with the docWizz Services



## What is the purpose of the maintenance mode?

The maintenance mode can only be used in docWizz Control Center (DWCC) by an user with administrative docWizz permissions.

Setting services into maintenance mode can be useful for updating docWizz in live production environments.

When Maintenance is enabled, all executed services will finish their current tasks, but no new tasks will be started.

The update procedure will check if all services are in maintenance mode and no task is still in process. Then it starts the update. After update is successfully finished, the maintenance mode can be removed.

All services will go back into idle status and tasks will be processed again.

## Technical information

### Maintenance mode is enabled

A file "MAINTENANCEUNDERWAY.wrk" is written into the configured \*\*\*WORK\*\*\* directory. This file is deleted, when maintenance is disabled in DWCC.

This is the file which triggers the blocking of new tasks to be processed.

### Maintenance mode is enabled

When all executed tasks are finished, the status information of the services is changed:

The status information can be found (default settings) in:

\*\*\*WORK\*\*\* \ServicesStatus\<computername>\<service>\Log

There are two relevant entries: "ReportedState" and "Status". The latter is shown in the GUI and the first is a value defined in the code.

Example in file:

```
## Running service ##
ReportedState = 8
Status=idle
## Running service, Maintenance flag is set in docWizz Control Center ##
ReportedState = 11
Status=maintenance
```

Maintenance mode is disabled

All (previously) running services go to "idle" status.

### 32.7.6.2 Memory Handling for dWCtrlDlg

Needed to clean up its cached data.

Image Name	User Name	CPU	Private Usage	Page Faults	PF Docks	VM Size
lsass.exe	Root	00	200 K	1,796	0	836 K
csrss.exe	SYSTEM	00	576 K	2,159	0	2,848 K
csrss.exe	SYSTEM	00	540 K	900	0	1,320 K
csrss.exe	SYSTEM	00	2,964 K	1,001,669	0	2,320 K
csrss.exe	SYSTEM	00	200 K	1,051	0	896 K
csrss.exe	SYSTEM	00	2,052 K	78,411	0	1,156 K
csrss.exe	Root	00	492 K	2,495	0	464 K
dWCtrlDlg.exe	Root	00	314,905,772	236	0	694,311 K
dwcc.exe	dWCtrl	00	25,660 K	7,879	0	19,772 K
dwcc.exe	dWCtrl	00	25,660 K	7,879	0	19,772 K
dwcc.exe	dWCtrl	00	25,660 K	7,879	0	19,772 K
dwcc.exe	dWCtrl	25	111,932 K	734,211	3,757	306,680 K
dwcc.exe	dWCtrl	00	108,500 K	171,973	0	258,936 K
dwcc.exe	dWCtrl	00	118,424 K	1,162,735	0	222,472 K
dwcc.exe	dWCtrl	00	69,252 K	29,957,092	0	174,812 K
dwcc.exe	dWCtrl	00	69,876 K	31,872,157	0	172,356 K
dwcc.exe	dWCtrl	05	1,760 K	2,442	0	3,936 K
explorer.exe	Root	00	9,724 K	4,122,320	13	8,240 K
SMSS.exe	SYSTEM	00	228 K	707	0	344 K

Found huge memory leaks in case filter were set.

Use the checkbox "recent", which is checked by default. If it is checked, it shows just latest 1000 entries of current selection. It helps to reduce used memory significantly.

### 32.7.6.3 Information for Updating Process

A useful information to keep in mind for updating in production:

**Old behaviour:** Set MTN as admin, wait till update is detected and processed. Services are stopped and new files from ..INSTALL are copied.

After update is finish, start DWCC and the services were automatically restarted (still with MTN icon). Deactivate MTN and proceed with production.

**NEW:** As before, but after Update is finished, MTN flag needs to be removed, only then services are restarted.

## 33 Auto-Power-Off

Enable "NightMode" when computers that idle from CCS point of view are shutdown.

The mechanism is useful, because it saves a lot of energy. In addition to shutdown it handles as well starting computers if needed using "Wake-Up-On-Lan" feature.

dWSrvManager has control on shutting down.

If a command for shut down has been found and the following conditions are fulfilled:

- All services are stopped
- No docWizz Client is running (test for lock-files)
- The computer is configured to allow shut-down
- No RQA services are running on this computer

While the power-off command is available, dWSrvManager will not start any service. Before shutting down, dWSrvManager reports MAC-Address for Wake-Up on file share and deletes the command.

To shutdown/wake computers, they must be first configured in docWizz-dwsrv.ini file as follows:

```
[NIGHTMODE]
Groups=Group1,Group2
LongIdleDuration=120; seconds
CriticalApps=dwcl.exe
NightModeTimer=20000; milliseconds
TimeToWaitForCriticalAppRestart=500; seconds
```

### Groups

Stands for the configured shutdown groups of workstations.

### LongIdleDuration

Stands for the time needed for the idle state of the service to be considered as long idle state.

### CriticalApps

Stands for the applications that found running leads to shutdown not taking place. The workstation will not be shutdown. ( in this case if DW is running, and the ws is configured to be shutdown from it's group, it will not be shutdown, because DW is configured to be a critical Application.)

### NightModeTimer

The service is verified periodically for shutdown , the time interval for this check is given by NighModeTimer.

### TimeToWaitForCriticalAppRestart

This timer sets the time in which the nightmode shutdown waits for the last critical application to be restarted, if the application is restarted the shutdown is cancelled.

```
[Group1]
Enabled=0
StartDaytime=08:00:00
```



```
EndDayTime=21:00:00
Computers=ws1,ws2,ws3,ws4
WakeableComputers=ws1,ws2,ws3
Days=1,2,3,4,5
```

### **Enabled**

0 – shutdown is disabled.

1 – shutdown is enabled. `Computers=<name>` in that group will be sent a nightmode shutdown command after 21:00 hr, if they do not have any of the CriticalApps started= dwcl.exe and they are not processing anything for over 120 seconds.

### **StartDaytime**

The beginning of the working day, in which the nightmode shutdown will not shutdown computers

### **EndDaytime**

The end of the working day, when shutdown starts to shutdown computers when they meet the criteria for shutdown.

### **Computers**

The workstations to be taken in consideration for shutdown, any other computers on the same configuration will not be included.

### **WakeableComputers**

The computers to be taken in consideration for wake-on LAN, any other computers on the same configuration will not be included.

### **Days**

The numbers 1-7 stand for the days of the week. 1= Monday, 2= Tuesday, 3 = Wednesday, 4= Thursday, 5=Friday , 6= Saturday, 7=Sunday

Computers that have in docWizz-dwsrv.ini configured DWFTPCLient, DWRemoteQALoader or DWREmoteQAManager, Shutdown is disabled, even if the workstation is configured for shutdown, these services exclude it from shutdown.

## 33.1 Shutting down

dWSrvManager has control on shutting down.

If a command for shut down has been found and following conditions are fulfilled:

- All services are stopped
- No docWizz Client is executed (test for lock-files)
- The computer is configured to allow shut-down
- No RQA services are executed on this computer

the computer will shut down.

While the power-off command is available, dwsrvmanager will not start any service. Before shutting down, dwsrvmanager reports MAC-Address for Wake-Up on file share and deletes the command.

To shutdown/wake computers, they must be first configured in docWizz-dwsrv.ini file as follows:

```
[NIGHTMODE]
Groups=Group1,Group2
LongIdleDuration=1200; seconds
CriticalApps=dwcl.exe
NightModeTimer=20000; milliseconds
TimeToWaitForCriticalAppRestart=500; seconds
```

### Groups

Stands for the configured shutdown groups of workstations.

### LongIdleDuration

Stands for the time needed for the idle state of the service to be considered as long idle state.

### CriticalApps

Stands for the applications that found executed leads to shutdown not taking place. The workstation will not be shutdown.( in this case if DW is running, and the ws is configured to be shutdown from it's group, it will not be shutdown, because DW is configured to be a critical Application.)

### NightModeTimer

The service is verified periodically for shutdown , the time interval for this check is given by NighModeTimer.

### TimeToWaitForCriticalAppRestart

This timer sets the time in which the nightmode shutdown waits for the last critical application to be restarted, if the application is restarted the shutdown is cancelled.

```
[Group1]
Enabled=0
StartDaytime=08:00:00
EndDayTime=21:00:00
Computers=ws1,ws2,ws3,ws4
WakeableComputers=ws1,ws2,ws3
Days=1,2,3,4,5
```

**Enabled**

If configured on 0, shutdown is disabled, if configured on 1 , shutdown is enabled.

**StartDaytime**

The beginning of the working day, in which the nightmode shutdown will not shutdown computers

**EndDaytime**

The end of the working day, when shutdown starts to shutdown computers when they meet the criteria for shutdown.

**Computers**

The workstations to be taken in consideration for shutdown, any other computers on the same configuration will not be included.

**WakeableComputers**

The computers to be taken in consideration for wake-on LAN, any other computers on the same configuration will not be included.

**Days** - the numbers 1-7 stand for the days of the week. 1= Monday, 2= Tuesday, 3 = Wednesday, 4= Thursday, 5=Friday , 6= Saturday, 7=Sunday

For computers to become wakeable , they must be first configured for shutdown, their reported state will be for shutdown, another important thing that it is used for wake-on LAN is the mac.ini file found in the WORK folder, this file contains the mac addresses used by wake-on LAN to wake a workstation. If a computer has never been shutdown with nightmode shutdown, then wake-on LAN will not wake that computer, because it can't find the mac address where to send the command for wake.

Computers that have in docWizz-dwsrv.ini configured DWFTPCLient, DWRemoteQALoader or DWREmoteQAManager, Shutdown is disabled, even if the workstation is configured for shutdown, these services exclude it from shutdown.

## 33.2 Service behavior

As soon power-off command is available for the current computer, all docWizz Services behave like receiving a stop command: Current action is completed, afterwards the service stops.

## 33.3 Wake-Up

A task checks for Wake-Up commands. If a wake-up command exists, it executes code for sending Wake-Up-On-Lan command to network card of off-powered computer. Then it deletes the command.

Nigthmode is configured in docWizz-dwsrv.ini file:


```
[NIGHTMODE]
Groups=NMGroup1,NMGroup2
LongIdleDuration=120; seconds
CriticalApps=dwcl.exe, DocumentValidator.exe
NightModeTimer=20000;milliseconds
TimeToWaitForCriticalAppRestart=120;seconds

[NMGroup1]
Enabled=0
StartDaytime=07:00:00
EndDaytime=22:00:00
Computers=
WakeableComputers=
Days=1,2,3,4,5
```

Wakeable computers if they are configured, and a MAC address is stored in mac.ini file under the ServiceStatus folder or WORK folder, they will be woken up if there is the need to process jobs/ tasks, according to a specific ratio constructed on the number of available computers to process jobs/tasks and the number of jobs/ tasks to be processed. They will be sent a wakeup command, one computer per wakeup task at a time, until the ratio is overrun, thus considering that the right amount of computers were awoken for processing, then entering again the long idle status, they will be powered off.

The mac.ini file found in the WORK folder contains the mac addresses used by wake-on LAN to wake a workstation. If a computer has never been shutdown with nightmode shutdown, then wake-on LAN will not wake that computer, because it can't find the mac address where to send the command for wake.

### Appearance

DWControlCenter shows a special icon (sleeping) , if wake-up is available. This is the case, if a MAC-Address for wake-up is stored. It is displayed when a DWSrv gets the nightmode shutdown command, afterwards, the icon is changed to stopped icon, so it is displayed just until the computer enters the sleep state from when it received the shutdown command.

## 33.4 Manual actions

You can manually wake-up or create power-off commands by context menu in ControlCenter.

## 33.5 Configuration

In dwsrv.ini, power-off groups can be configured. Each Computer or a group of computers can be assigned to a power-off group. Each Power-Off group has following settings:

```
[Group1]
Enabled=0/1
StartDaytime=07:00:00
EndDaytime=22:00:00
Computers=
WakeableComputers=
Days=1,2,3,4,5 (Mo-Fr)
```

This is needed to protect computers to shut down that may be used by humans. E.g. pure servers will be configured in a different group with different settings.

## 33.6 Long Idle

If DWSrvManager detects that all services are idle for more than a configured amount of time (minutes), and conditions of Shutting down are fulfilled, it creates power-off command automatically.

## 33.7 Load Balancing

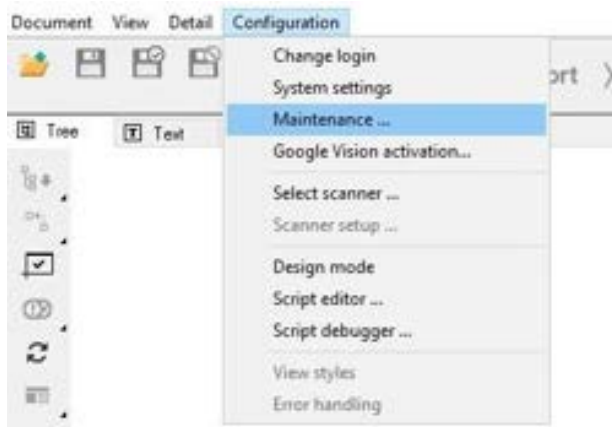
It is important to ensure documents/tasks to be processed with higher priority on computers that will not shut down. For that reason, a priority list can be generated (order all computers in that list by their priority). From this list and from configuration of jobs/tasks, for each job/task a priority list of computers can be created.

Now, when a service on a computer that may shut down checks for a job/task, it searches the job/task list whether a service with higher priority is in idle state. If so, it does not execute the job/task.

This way a service on a computer that may shut down executes jobs, if all higher prioritized services that may execute this job/task are busy, stopped or unavailable.

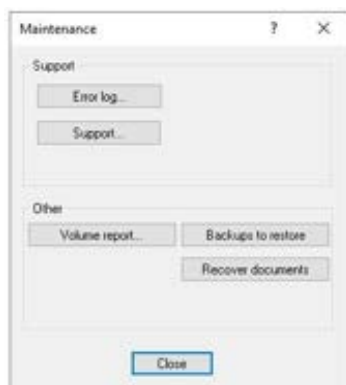
## 34 Maintenance

Go to Configuration menu - open Maintenance. You need to be logged in as administrator.

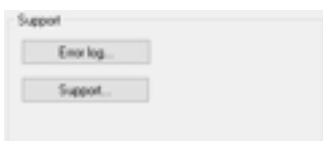


This directory contains maintenance files for support. E.g. VolumeReport.

C:\Program Files\docWizz\docWizz\Maintenance\...



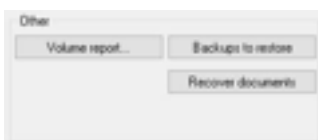
### Support



Click on **Error log...** to get the log details.

Click on **Support...** switch top the Maintenance Client.

### Other



The **Volume Report** is an additional function in docWizz that creates a PDF file containing the total page counter and the number of pages of the selected month. A new dialog is opened where you can select the desired period of which you want to create the **Volume Report**.

Click on **Backup to restore** to see a list of documents in the restore queue.

Opens a separate window where you specify different statistic views. For more information please refer to **Processing Statistics**.

Recover documents and get the number of restored documents.

## 34.1 Maintenance Client (MTN)

The MAINTENANCE (MTN) Client is an automatic problem reporting and customer supporting package for most efficient problem solving.

To start the MTNClient from inside the docWizz Control Center. The MTNClient creates a zip package containing all selected information to diagnose and solve the problems.

### Behavior in case of errors

All handled problems will be logged in the central protocol file.

The **MTNClient** dialog as submodule of docWizz is included with an error filter.

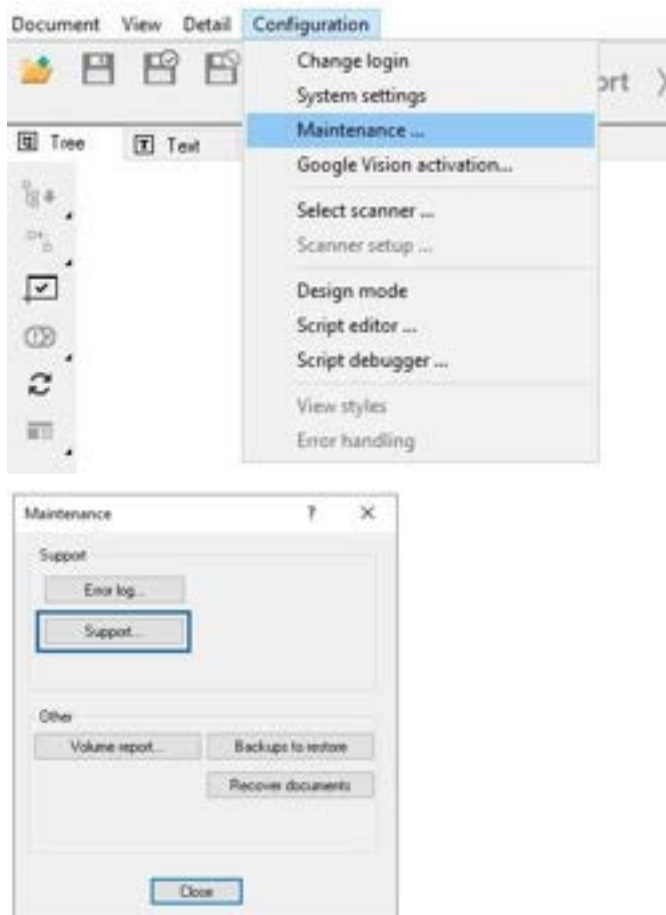
If an unexpected error appears the dialog would be closed without a message. In the log database an error will be logged. The user can continue his work with docWizz without any problems.

In case of expected errors the error will be shown and logged in the log database.

## 34.2 How to use the MTNClient

### 34.2.1 Starting the MTNClient

Start from the docWizz Configuration menu by the Maintenance entry and click on **Support** button.



### 34.2.2 Surface of the MTNClient

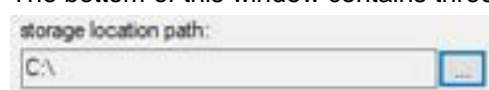
The MTNClient has the following structure:

The control box (left side) will be extended by each module which is installed on your computer.

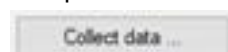
The description box (top right) contains details for the current selected option in the control box.

The option box (logo side) will always display the selectable options and dialog interfaces for the current selected module option.

The bottom of this window contains three buttons:



The path selection to change the location where the package will be stored



Starts storing all information in a new package.



Close this window.



On changing the storage location, following dialog appears:



**yes:** location will be stored

**no:** location change will be only used for current session

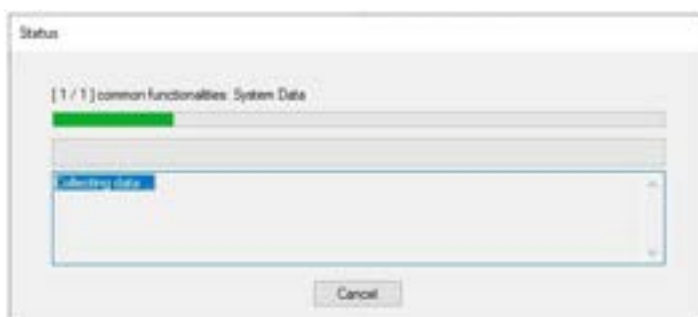
### 34.2.3 MTN Package Generation

Click **Collect data** button to generate a MTN package.

The package generation can take some more time - depending on the selected modules and suboptions.

When a package is written all data is in there. The packages can be deleted manually, while the folder is defined by client.

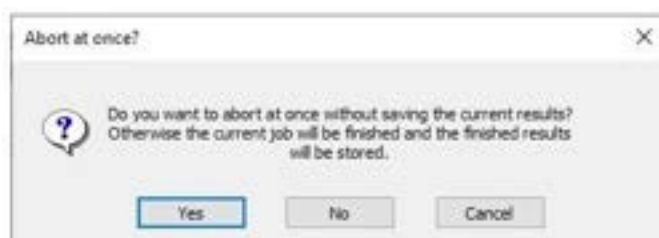
Probably it would be good to have in a dialog the view on the existing items to delete.



This process can be abort this processing by pressing

Cancel

Then the following message will be shown:



Yes

abort at once - no results will be available

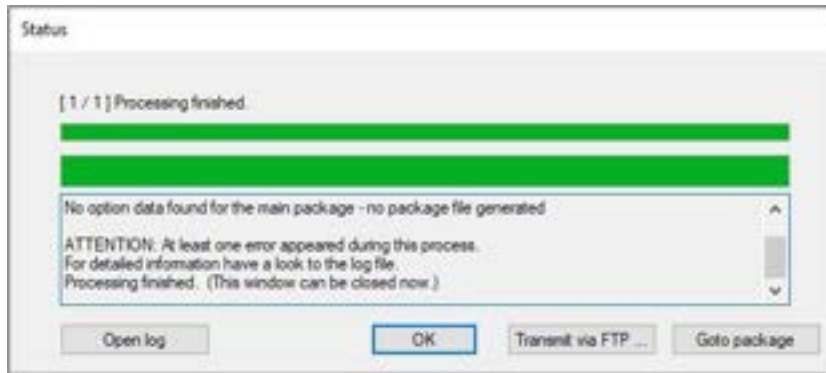
No

current step will be finished - that results will be stored

Cancel

continue without any changes

If the processing will be finished during the message dialog is shown, the dialog will disappear without any interaction.



### **Transmit via FTP**

When the package has been created, the package can be transmitted. The detailed handling is described in the chapter "Sending the data package to CCS".

### **Goto package**

This opens an Explorer window in the result directory.

### **Open log**

In case of errors the protocol file can be reviewed in an separated sub dialog by pressing this button. This protocol file is included in the package automatically.

## **34.2.4 Sending the data package to CCS**

At present there are following possibilities to send the package to CCS:

- FTP transfer (a customer account will be created on FTP@CCS)
- Postal services ( CD / DVD )

Note: Please refrain from sending attachments via E-Mail.

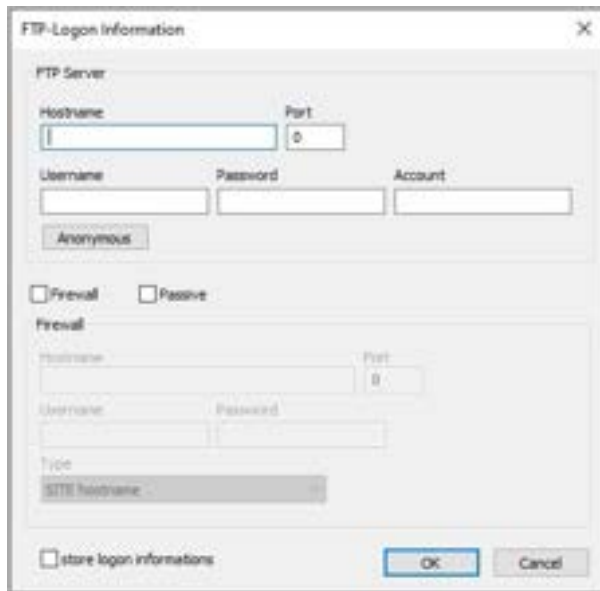
### **Standard FTP transfer**

After the successful generation of an MTNPackage a button "Transmit via FTP ..." is shown in the progress dialog.

For transmission of the MTN package via FTP you will need the access data from CCS for your ftp account.

If you don't have an access, please contact the CCS support team.

After selecting this transmission button a dialog for inserting the logon details will be displayed.



The image shows a dialog box titled "FTP-Logon Information". It contains two main sections: "FTP Server" and "Firewall". In the "FTP Server" section, there are input fields for "Hostname", "Port" (with a default value of 0), "Username", "Password", and "Account". Below these fields is a button labeled "Anonymous". In the "Firewall" section, there are checkboxes for "Preval" and "Passive". Below these is another set of input fields for "Hostname", "Port" (with a default value of 0), "Username", and "Password". There is also a "Type" dropdown menu with "SITE hostname" selected. At the bottom of the dialog, there is a checkbox labeled "store logon informations" and two buttons: "OK" and "Cancel".

**Hostname**

By default "ftp.ccs-gmbh.net"

**Username**

Account name you got from CCS

**Password**

Password you got from CCS

**Account**

Currently not needed

If you are using a firewall with authentication, activate.

## 34.3 MTN Modules

MTNClient uses separated unique defined modules which can be included into the program appending to starting manner (from inside an CCS product or external by MTNClient.exe) and requirements.

MTNClient uses one static module "common functionalities", which will always be loaded and contains certain basic functions.

All other modules are loaded dynamically.

The whole session of the MTNClient is written in a log file. This file is always stored in the user temporary folder, f.e. *"C:\Documents and Settings\<UserName>\Local Settings\Temp"*.

The log file is overwritten, if the last modification date is of the day before, otherwise it is continued!

The log file is also copied into the package containing all information until it is included itself into the package. Later on generated errors (f.e. at closing the MTNClient) only will be included in a second package generation or has to be sent manually.

The modules which should be loaded are registered in the following registry path:

for multibyte module DLLs:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\CCS GmbH\MTNClient\ProductDLLs]
```

for unicode module DLLs:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\CCS GmbH\MTNClient\ProductDLLsU]
```

Each module has to be inserted as own key element in the following way:

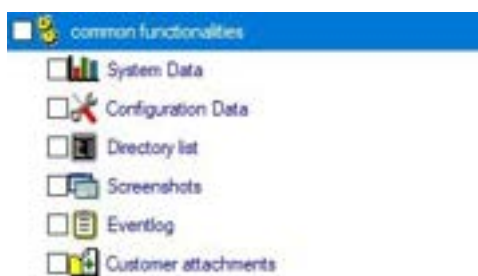
```
"<name of the module>" REG_SZ "<full path and filename>"
```

f.e.

```
"myModule" REG_SZ "C:\Temp\myModule.dll"
```

## 34.4 MTN Module "common functionalities"

The module "common functionalities" contains the following options:



### 34.4.1 System Data

The system data option is for collecting global environment information of the computer and the executed operation system to be able to detect and to exclude typical faults causing errors with docWizz.

The possible selections are:

- System environment: creates an PDF report containing:
  - Operating system version
  - User & computer name
  - Display resolution
  - System directories
  - Environment variables
  - CPU information
  - Memory status
  - Disk drives and capacity
  - All drive letters and their mappings
  - Network information (adapter details, IP addresses, DHCP information)
- Hardware profiles
- Desktop symbols
- Start menu symbols
- CCS specific registry entries
- List of registered DLLs



### 34.4.2 Configuration Data

### 34.4.3 Directory List

The directory list option is for creating an list of desired folders including all their subfolders. The list is generated in XML format.

To view the results of this list in a clear structure there exist a xsd-stylesheet.

Maximum 30 items possible in this list.

*This feature helps to view, if error appears in case of missing or wrong positioned files or folders.*



### 34.4.4 Screenshots

The screenshot option is to select the created screenshots for adding to the package. Here it is possible to manage the created screenshots, to insert or adapt the comments to the actual selected screenshot and to select which screenshots should be included into the package.

After selecting one screenshot in the list you can write a comment into the lower text area. For a closer look click on the small image thumbnail. A new preview window will be opened. In this preview dialog it is possible to mark areas on the screenshot with a red pen.

*This feature should help to explain easily the customer requests or questions.*

While the MTNClient is executed, it is possible to create easily screenshots of the customers surface.

To create a screenshot of the current active dialog press the keys [Ctrl] + [Alt] + [Print].

To create an screenshot of the whole desktop, press [Ctrl] + [Alt] + [Shift] + [Print].

The screenshots can be copied to other programs or locations by drag & drop (not via copy & paste).

After the creation of an screenshot a dialog with the current created screenshot will be opened.

### 34.4.5 EventLog

The eventlog option is to collect easily the logged events of the system eventlog and copy it into the package.

*Several CCS products are executed as service which reports their status into the event log. With this option the support is able to detect reasons of services and global system problems.*



### 34.4.6 Customer Attachments

This option is to collect further attachments, the support needs to reconstruct or analyse the problem.

The two buttons allows selecting a single file or an complete folder.

The selected data will be copied into a compressed zip file.



### 34.4.7 Shortcuts

Shortcuts are available and are mentioned directly in the MTN application.



## 34.5 MTN Module docWizz

### 34.5.1 Backup documents

Add the desired backup files of docWizz.

### 34.5.2 Exported documents

Select the desired exported documents of docWizz. Maximum 30 items possible in the list.

More than 30 items selected would cause a huge package size. If more documents would need to be selected a second package could be created.

### 34.5.3 Pool documents

Select the desired pool documents of docWizz. Maximum 30 items possible in the list.



## 34.6 MTNClient - Results

Explanation to the results generated by the MTN Modules.

### 34.6.1 Results of the MTN Module "common functionalities"

For each option of this module an own zip package is generated named like the option name, f.e.

```
"<optionNo>_<optionName>.zip"
```

and also for all entries of suboptions, named in the way see below:

```
"<listNo>_<path>_<subpath>.zip" or  
"<listNo>_<path>_<subpath>_<filename.ext>.zip"
```

All slashes ['/'] and backslashes ['\'] are replaced by ['\_'].

### 34.6.2 Result: System Data

The registry data are stored in the following xml structure:

```
<?xml-stylesheet type="text/xsl" href="http://schema.ccs-gmbh.com/supportManager/regTree.xsl"?>  
<registry version="1.2">  
  <k n="HKEY_LOCAL_MACHINE">  
    <k n="Software">  
      <k n="CCS GmbH" writeAccess="true">  
        <v n="LastDate" v="2012.10.25.12.07.59" dt="REG_SZ"/>  
        <v n="StationName" v="" dt="REG_SZ"/>  
        <v n="Serial-Num" v="" dt="REG_SZ"/>  
        <v n="LicenseCode" v="" dt="REG_SZ"/>  
        <v n="VALID" v="05.10.2012" dt="REG_SZ"/>  
        <v n="PageCount" v="17" dt="REG_SZ"/>  
        <k n="MTNClient_TestKey_{87FD2801-14FD-4310-8F41-A9093C070723}"/>  
      </k>  
    </k>  
  </k>  
</registry>
```

For reviewing this content an XSL stylesheet is assigned to the xml content.

On opening the xml file with an XSL processor the data will be transformed in the following way:



### 34.6.3 Result: Directory List

For reviewing this content an XSL stylesheet is assigned to the xml content.

On opening the xml file with an XSL processor (f.e. Internet Explorer > 9.0), the data will be transformed in the following way:

[illegible]

#### 34.6.4 Result: Screenshots, Eventlog, Customer attachments

## Screenshots

Each appended screenshot will be included into the package as png image. If an comment was inserted into the dialog item, a text file will also exist with the same filename.

## Eventlog

The eventlog is stored in the following xml structure.

For reviewing this content an XSL stylesheet is assigned to the xml content.

## Customer attachments

This zip contains only the files as selected by the customer.

The eventlog data are stored in the following xml structure:

For reviewing this content an XSL stylesheet is assigned to the xml content.



CCS  
Certified Conversion Specialists GmbH

© Copyright CCS-Certified Conversion Specialists GmbH, Hamburg

## EventLog table

Info Version: 6.4.4.10  
XSL Version: 1.5

log category: Application  
 period: 01.10.2011 01:00:00 - 31.10.2011 13:57:30  
 sorted by: none

[0] highlighting

event source type, src	event ID	time generated	time written	src ID	source name	computer name	SID Name	SID domain	SID user name	SID user type	de
 94017	0	11.10.2011 13:57:30	11.10.2013 13:57:41	72	Update	SR-PC	0-4	0-4	0-4		
 94016	0	11.10.2011 13:26:00	10.10.2012 13:26:00	72	update	SR-PC	0-4	0-4	0-4		
 94015	0	11.10.2011 13:26:00	10.10.2012 13:26:00	72	update	SR-PC	0-4	0-4	0-4		
 94016	0	11.10.2011 13:26:00	10.10.2012 13:26:41	72	Update	SR-PC	0-4	0-4	0-4		
 94013	0	11.10.2011 13:57:41	11.10.2013 13:57:41	72	Update	SR-PC	0-4	0-4	0-4		

To get the full text of the entries, the actual line is highlighting where the mouse cursor is over. This function can be en-/disabled by a left mouse click.



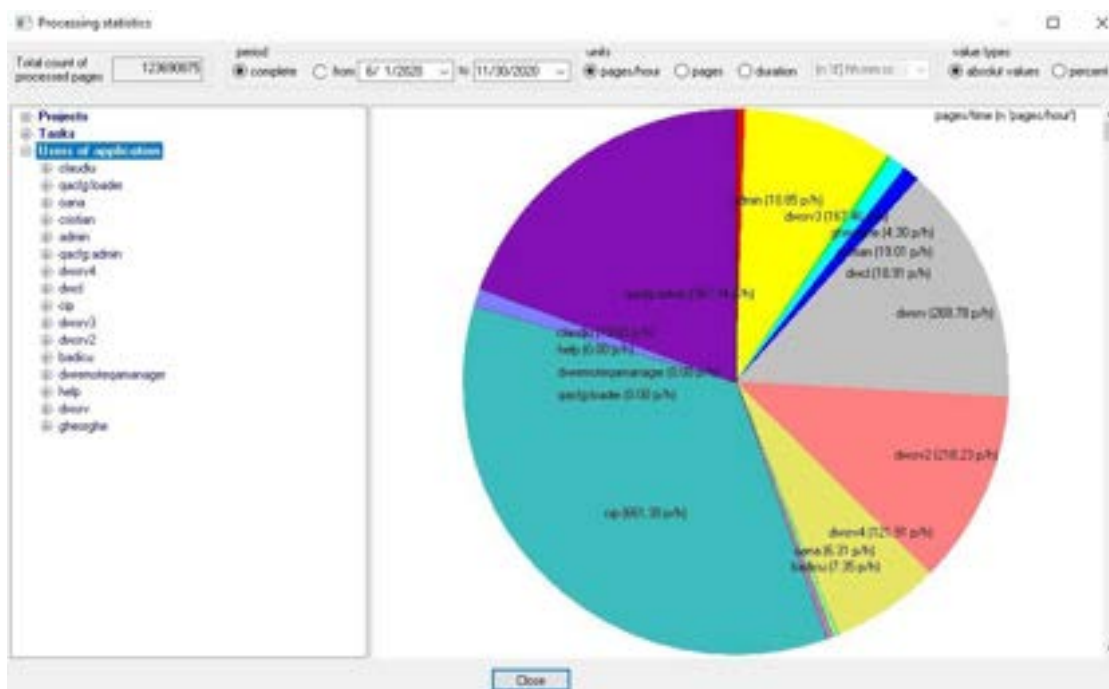
On positioning the mouse cursor over this information button the following help is shown to sort the eventlog entries by a special column.

For sorting the eventLog change the attribute "sort" in the root element "eventLog" of your current xmldata file.  
Following values are accepted (in case of wrong value no sorting will be done):

- "name" - sort by the 'type' (name of the xml nodes)
- "rn" - sort by the 'record number'
- "eid" - sort by the 'event ID'
- "tg" - sort by the 'time generated'
- "tw" - sort by the 'time written'
- "uSIDo" - sort by the 'user SID offset'
- "sn" - sort by the 'source name'
- "cn" - sort by the 'computer name'
- "SIDn" - sort by the 'SID name'
- "SIDdn" - sort by the 'SID domain name'
- "SIDun" - sort by the 'SID user name'
- "SIDut" - sort by the 'SID user type'
- "data" - sort by the 'data'
- "text" - sort by the message texts (content of the xml nodes)

If the sorting is done correct, the sorting row is displayed in a blue frame below the period dates.

## 34.7 Processing Statistics



## 35 EDS error list

As return values, EDS SDK APIs return error codes defined as follows.

For each API, the return values mainly used are identified based on API characteristics. However, the principal factors that actually caused the problems are specified as error codes. Thus, all error codes may be specified in return values.

### 35.1 General errors

EDS_ERR_UNIMPLEMENTED	Not implemented
EDS_ERR_INTERNAL_ERROR	Internal error
EDS_ERR_MEM_ALLOC_FAILED	Memory allocation error
EDS_ERR_MEM_FREE_FAILED	Memory release error
EDS_ERR_OPERATION_CANCELLED	Operation canceled
EDS_ERR_INCOMPATIBLE_VERSION	Version error
EDS_ERR_NOT_SUPPORTED	Not supported
EDS_ERR_UNEXPECTED_EXCEPTION	Unexpected exception
EDS_ERR_PROTECTION_VIOLATION	Protection violation
EDS_ERR_MISSING_SUBCOMPONENT	Missing subcomponent
EDS_ERR_SELECTION_UNAVAILABLE	Selection unavailable

### 35.2 File access errors

EDS_ERR_FILE_IO_ERROR	IO error
EDS_ERR_FILE_TOO_MANY_OPEN	Too many files open
EDS_ERR_FILE_NOT_FOUND	File does not exist
EDS_ERR_FILE_OPEN_ERROR	Open error
EDS_ERR_FILE_CLOSE_ERROR	Close error
EDS_ERR_FILE_SEEK_ERROR	Seek error
EDS_ERR_FILE_TELL_ERROR	Tell error
EDS_ERR_FILE_READ_ERROR	Read error
EDS_ERR_FILE_WRITE_ERROR	Write error
EDS_ERR_FILE_PERMISSION_ERROR	Permission error
EDS_ERR_FILE_DISK_FULL_ERROR	Disk full
EDS_ERR_FILE_ALREADY_EXISTS	File already exists
EDS_ERR_FILE_FORMAT_UNRECOGNIZED	Format error
EDS_ERR_FILE_DATA_CORRUPT	Invalid data

EDS_ERR_FILE_NAMING_NA	File naming error
------------------------	-------------------

### 35.3 Directory errors

EDS_ERR_DIR_NOT_FOUND	Directory does not exist
EDS_ERR_DIR_IO_ERROR	I/O error
EDS_ERR_DIR_ENTRY_NOT_FOUND	No file in directory
EDS_ERR_DIR_ENTRY_EXISTS	File in directory
EDS_ERR_DIR_NOT_EMPTY	Directory full

### 35.4 Property errors

EDS_ERR_PROPERTIES_UNAVAILABLE	Property (and additional property information) unavailable
EDS_ERR_PROPERTIES_MISMATCH	Property mismatch
EDS_ERR_PROPERTIES_NOT_LOADED	Property not loaded

### 35.5 Function parameter errors

EDS_ERR_INVALID_PARAMETER	Invalid function parameter
EDS_ERR_INVALID_HANDLE	Handle error
EDS_ERR_INVALID_POINTER	Pointer error
EDS_ERR_INVALID_INDEX	Index error
EDS_ERR_INVALID_LENGTH	Length error
EDS_ERR_INVALID_FN_POINTER	FN pointer error
EDS_ERR_INVALID_SORT_FN	Sort FN error

### 35.6 Device errors

EDS_ERR_DEVICE_NOT_FOUND	Device not found
EDS_ERR_DEVICE_BUSY	Device busy
	Note: If a device busy error occurs, reissue the command after a while. The camera will become unstable.
EDS_ERR_DEVICE_INVALID	Device error
EDS_ERR_DEVICE_EMERGENCY	Device emergency
EDS_ERR_DEVICE_MEMORY_FULL	Device memory full

EDS_ERR_DEVICE_INTERNAL_ERROR	Internal device error
EDS_ERR_DEVICE_INVALID_PARAMETER	Device parameter invalid
EDS_ERR_DEVICE_NO_DISK	No disk
EDS_ERR_DEVICE_DISK_ERROR	Disk error
EDS_ERR_DEVICE_CF_GATE_CHANGED	The CF gate has been changed
EDS_ERR_DEVICE_DIAL_CHANGED	The dial has been changed
EDS_ERR_DEVICE_NOT_INSTALLED	Device not installed
EDS_ERR_DEVICE_STAY_AWAKE	Device connected in awake mode
EDS_ERR_DEVICE_NOT_RELEASED	Device not released

### 35.7 Stream errors

EDS_ERR_STREAM_IO_ERROR	Stream I/O error
EDS_ERR_STREAM_NOT_OPEN	Stream open error
EDS_ERR_STREAM_ALREADY_OPEN	Stream already open
EDS_ERR_STREAM_OPEN_ERROR	Failed to open stream
EDS_ERR_STREAM_CLOSE_ERROR	Failed to close stream
EDS_ERR_STREAM_SEEK_ERROR	Stream seek error
EDS_ERR_STREAM_TELL_ERROR	Stream tell error
EDS_ERR_STREAM_READ_ERROR	Failed to read stream
EDS_ERR_STREAM_WRITE_ERROR	Failed to write stream
EDS_ERR_STREAM_PERMISSION_ERROR	Permission error
EDS_ERR_STREAM_COULDNT_BEGIN_THREAD	Could not start reading thumbnail
EDS_ERR_STREAM_BAD_OPTIONS	Invalid stream option
EDS_ERR_STREAM_END_OF_STREAM	Invalid stream termination

### 35.8 Communication errors

EDS_ERR_COMM_PORT_IS_IN_USE	Port in use
EDS_ERR_COMM_DISCONNECTED	Port disconnected
EDS_ERR_COMM_DEVICE_INCOMPATIBLE	Incompatible device
EDS_ERR_COMM_BUFFER_FULL	Buffer full
EDS_ERR_COMM_USB_BUS_ERR	USB bus error



## 35.9 Camera UI lock/unlock errors

EDS_ERR_USB_DEVICE_LOCK_ERROR	Failed to lock the UI
EDS_ERR_USB_DEVICE_UNLOCK_ERROR	Failed to unlock the UI

## 35.10 STI/WIA errors

EDS_ERR_STI_UNKNOWN_ERROR	Unknown STI
EDS_ERR_STI_INTERNAL_ERROR	Internal STI error
EDS_ERR_STI_DEVICE_CREATE_ERROR	Device creation error
EDS_ERR_STI_DEVICE_RELEASE_ERROR	Device release error
EDS_ERR_DEVICE_NOT_LAUNCHED	Device startup failed

## 35.11 Other general errors

EDS_ERR_ENUM_NA	Enumeration terminated (there was no suitable enumeration item)
EDS_ERR_INVALID_FN_CALL	Called in a mode when the function could not be used
EDS_ERR_HANDLE_NOT_FOUND	Handle not found
EDS_ERR_INVALID_ID	Invalid ID
EDS_ERR_WAIT_TIMEOUT_ERROR	Timeout
EDS_ERR_LAST_GENERIC_ERROR_PLUS_ONE	Not used

## 35.12 PTP errors

EDS_ERR_SESSION_NOT_OPEN	Session open error
EDS_ERR_INVALID_TRANSACTIONID	Invalid transaction ID
EDS_ERR_INCOMPLETE_TRANSFER	Transfer problem
EDS_ERR_INVALID_STRAGEID	Storage error
EDS_ERR_DEVICEPROP_NOT_SUPPORTED	Unsupported device property
EDS_ERR_INVALID_OBJECTFORMATCODE	Invalid object format code
EDS_ERR_SELF_TEST_FAILED	Failed self-diagnosis
EDS_ERR_PARTIAL_DELETION	Failed in partial deletion
EDS_ERR_SPECIFICATION_BY_FORMAT_UNSUPPORTED	Unsupported format specification
EDS_ERR_NO_VALID_OBJECTINFO	Invalid object information

EDS_ERR_INVALID_CODE_FORMAT	Invalid code format
EDS_ERR_UNKNOWN_VENDOR_CODE	Unknown vendor code
EDS_ERR_CAPTURE_ALREADY_TERMINATED	Capture already terminated
EDS_ERR_INVALID_PARENTOBJECT	Invalid parent object
EDS_ERR_INVALID_DEVICEPROP_FORMAT	Invalid property format
EDS_ERR_INVALID_DEVICEPROP_VALUE	Invalid property value
EDS_ERR_SESSION_ALREADY_OPEN	Session already open
EDS_ERR_TRANSACTION_CANCELLED	Transaction canceled
EDS_ERR_SPECIFICATION_OF_DESTINATION_UNSUPPORTED	Unsupported destination specification
EDS_ERR_UNKNOWN_COMMAND	Unknown command
EDS_ERR_OPERATION_REFUSED	Operation refused
EDS_ERR_LENS_COVER_CLOSE	Lens cover closed
EDS_ERR_OBJECT_NOTREADY	Image data set not ready for live view

### 35.13 TakePicture errors

EDS_ERR_TAKE_PICTURE_AF_NG	Focus failed
EDS_ERR_TAKE_PICTURE_RESERVED	Reserved
EDS_ERR_TAKE_PICTURE_MIRROR_UP_NG	Currently configuring mirror up
EDS_ERR_TAKE_PICTURE_SENSOR_CLEANSING_NG	Currently cleaning sensor
EDS_ERR_TAKE_PICTURE_SILENCE_NG	Currently performing silent operations
EDS_ERR_TAKE_PICTURE_NO_CARD_NG	Card not installed
EDS_ERR_TAKE_PICTURE_CARD_NG	Error writing to card
EDS_ERR_TAKE_PICTURE_CARD_PROTECT_NG	Card write protected

### 35.14 ED-SDK Generic Error IDs

```

/*-----
    ED-SDK Function Success Code
    -----*/
#define EDS_ERR_OK                                0x00000000L

/*-----
    ED-SDK Generic Error IDs
    -----*/

```

```

/* Miscellaneous errors */
#define EDS_ERR_UNIMPLEMENTED 0x00000001L
#define EDS_ERR_INTERNAL_ERROR 0x00000002L
#define EDS_ERR_MEM_ALLOC_FAILED 0x00000003L
#define EDS_ERR_MEM_FREE_FAILED 0x00000004L
#define EDS_ERR_OPERATION_CANCELLED 0x00000005L
#define EDS_ERR_INCOMPATIBLE_VERSION 0x00000006L
#define EDS_ERR_NOT_SUPPORTED 0x00000007L
#define EDS_ERR_UNEXPECTED_EXCEPTION 0x00000008L
#define EDS_ERR_PROTECTION_VIOLATION 0x00000009L
#define EDS_ERR_MISSING_SUBCOMPONENT 0x0000000AL
#define EDS_ERR_SELECTION_UNAVAILABLE 0x0000000BL

/* File errors */
#define EDS_ERR_FILE_IO_ERROR 0x00000020L
#define EDS_ERR_FILE_TOO_MANY_OPEN 0x00000021L
#define EDS_ERR_FILE_NOT_FOUND 0x00000022L
#define EDS_ERR_FILE_OPEN_ERROR 0x00000023L
#define EDS_ERR_FILE_CLOSE_ERROR 0x00000024L
#define EDS_ERR_FILE_SEEK_ERROR 0x00000025L
#define EDS_ERR_FILE_TELL_ERROR 0x00000026L
#define EDS_ERR_FILE_READ_ERROR 0x00000027L
#define EDS_ERR_FILE_WRITE_ERROR 0x00000028L
#define EDS_ERR_FILE_PERMISSION_ERROR 0x00000029L
#define EDS_ERR_FILE_DISK_FULL_ERROR 0x0000002AL
#define EDS_ERR_FILE_ALREADY_EXISTS 0x0000002BL
#define EDS_ERR_FILE_FORMAT_UNRECOGNIZED 0x0000002CL
#define EDS_ERR_FILE_DATA_CORRUPT 0x0000002DL
#define EDS_ERR_FILE_NAMING_NA 0x0000002EL

/* Directory errors */
#define EDS_ERR_DIR_NOT_FOUND 0x00000040L
#define EDS_ERR_DIR_IO_ERROR 0x00000041L
#define EDS_ERR_DIR_ENTRY_NOT_FOUND 0x00000042L
#define EDS_ERR_DIR_ENTRY_EXISTS 0x00000043L
#define EDS_ERR_DIR_NOT_EMPTY 0x00000044L

/* Property errors */
#define EDS_ERR_PROPERTIES_UNAVAILABLE 0x00000050L
#define EDS_ERR_PROPERTIES_MISMATCH 0x00000051L
#define EDS_ERR_PROPERTIES_NOT_LOADED 0x00000053L

```

```

/* Function Parameter errors */
#define EDS_ERR_INVALID_PARAMETER 0x00000060L
#define EDS_ERR_INVALID_HANDLE 0x00000061L
#define EDS_ERR_INVALID_POINTER 0x00000062L
#define EDS_ERR_INVALID_INDEX 0x00000063L
#define EDS_ERR_INVALID_LENGTH 0x00000064L
#define EDS_ERR_INVALID_FN_POINTER 0x00000065L
#define EDS_ERR_INVALID_SORT_FN 0x00000066L

/* Device errors */
#define EDS_ERR_DEVICE_NOT_FOUND 0x00000080L
#define EDS_ERR_DEVICE_BUSY 0x00000081L
#define EDS_ERR_DEVICE_INVALID 0x00000082L
#define EDS_ERR_DEVICE_EMERGENCY 0x00000083L
#define EDS_ERR_DEVICE_MEMORY_FULL 0x00000084L
#define EDS_ERR_DEVICE_INTERNAL_ERROR 0x00000085L
#define EDS_ERR_DEVICE_INVALID_PARAMETER 0x00000086L
#define EDS_ERR_DEVICE_NO_DISK 0x00000087L
#define EDS_ERR_DEVICE_DISK_ERROR 0x00000088L
#define EDS_ERR_DEVICE_CF_GATE_CHANGED 0x00000089L
#define EDS_ERR_DEVICE_DIAL_CHANGED 0x0000008AL
#define EDS_ERR_DEVICE_NOT_INSTALLED 0x0000008BL
#define EDS_ERR_DEVICE_STAY_AWAKE 0x0000008CL
#define EDS_ERR_DEVICE_NOT_RELEASED 0x0000008DL

/* Stream errors */
#define EDS_ERR_STREAM_IO_ERROR 0x000000A0L
#define EDS_ERR_STREAM_NOT_OPEN 0x000000A1L
#define EDS_ERR_STREAM_ALREADY_OPEN 0x000000A2L
#define EDS_ERR_STREAM_OPEN_ERROR 0x000000A3L
#define EDS_ERR_STREAM_CLOSE_ERROR 0x000000A4L
#define EDS_ERR_STREAM_SEEK_ERROR 0x000000A5L
#define EDS_ERR_STREAM_TELL_ERROR 0x000000A6L
#define EDS_ERR_STREAM_READ_ERROR 0x000000A7L
#define EDS_ERR_STREAM_WRITE_ERROR 0x000000A8L
#define EDS_ERR_STREAM_PERMISSION_ERROR 0x000000A9L
#define EDS_ERR_STREAM_COULDNT_BEGIN_THREAD 0x000000AAL
#define EDS_ERR_STREAM_BAD_OPTIONS 0x000000ABL
#define EDS_ERR_STREAM_END_OF_STREAM 0x000000ACL

```

```

/* Communications errors */
#define EDS_ERR_COMM_PORT_IS_IN_USE 0x000000C0L
#define EDS_ERR_COMM_DISCONNECTED 0x000000C1L
#define EDS_ERR_COMM_DEVICE_INCOMPATIBLE 0x000000C2L
#define EDS_ERR_COMM_BUFFER_FULL 0x000000C3L
#define EDS_ERR_COMM_USB_BUS_ERR 0x000000C4L

/* Lock/Unlock */
#define EDS_ERR_USB_DEVICE_LOCK_ERROR 0x000000D0L
#define EDS_ERR_USB_DEVICE_UNLOCK_ERROR 0x000000D1L

/* STI/WIA */
#define EDS_ERR_STI_UNKNOWN_ERROR 0x000000E0L
#define EDS_ERR_STI_INTERNAL_ERROR 0x000000E1L
#define EDS_ERR_STI_DEVICE_CREATE_ERROR 0x000000E2L
#define EDS_ERR_STI_DEVICE_RELEASE_ERROR 0x000000E3L
#define EDS_ERR_DEVICE_NOT_LAUNCHED 0x000000E4L

#define EDS_ERR_ENUM_NA 0x000000F0L
#define EDS_ERR_INVALID_FN_CALL 0x000000F1L
#define EDS_ERR_HANDLE_NOT_FOUND 0x000000F2L
#define EDS_ERR_INVALID_ID 0x000000F3L
#define EDS_ERR_WAIT_TIMEOUT_ERROR 0x000000F4L

/* PTP */
#define EDS_ERR_SESSION_NOT_OPEN 0x00002003
#define EDS_ERR_INVALID_TRANSACTIONID 0x00002004
#define EDS_ERR_INCOMPLETE_TRANSFER 0x00002007
#define EDS_ERR_INVALID_STRAGEID 0x00002008
#define EDS_ERR_DEVICEPROP_NOT_SUPPORTED 0x0000200A
#define EDS_ERR_INVALID_OBJECTFORMATCODE 0x0000200B
#define EDS_ERR_SELF_TEST_FAILED 0x00002011
#define EDS_ERR_PARTIAL_DELETION 0x00002012
#define EDS_ERR_SPECIFICATION_BY_FORMAT_UNSUPPORTED 0x00002014
#define EDS_ERR_NO_VALID_OBJECTINFO 0x00002015
#define EDS_ERR_INVALID_CODE_FORMAT 0x00002016
#define EDS_ERR_UNKNOWN_VENDOR_CODE 0x00002017
#define EDS_ERR_CAPTURE_ALREADY_TERMINATED 0x00002018
#define EDS_ERR_INVALID_PARENTOBJECT 0x0000201A
#define EDS_ERR_INVALID_DEVICEPROP_FORMAT 0x0000201B
#define EDS_ERR_INVALID_DEVICEPROP_VALUE 0x0000201C

```

```

#define EDS_ERR_SESSION_ALREADY_OPEN 0x0000201E
#define EDS_ERR_TRANSACTION_CANCELLED 0x0000201F
#define EDS_ERR_SPECIFICATION_OF_DESTINATION_UNSUPPORTED 0x00002020

/* PTP Vendor */
#define EDS_ERR_UNKNOWN_COMMAND 0x0000A001
#define EDS_ERR_OPERATION_REFUSED 0x0000A005
#define EDS_ERR_LENS_COVER_CLOSE 0x0000A006
#define EDS_ERR_LOW_BATTERY 0x0000A101
#define EDS_ERR_OBJECT_NOTREADY 0x0000A102

#define EDS_ERR_TAKE_PICTURE_AF_NG 0x00008D01L
#define EDS_ERR_TAKE_PICTURE_RESERVED 0x00008D02L
#define EDS_ERR_TAKE_PICTURE_MIRROR_UP_NG 0x00008D03L
#define EDS_ERR_TAKE_PICTURE_SENSOR_CLEANING_NG 0x00008D04L
#define EDS_ERR_TAKE_PICTURE_SILENCE_NG 0x00008D05L
#define EDS_ERR_TAKE_PICTURE_NO_CARD_NG 0x00008D06L
#define EDS_ERR_TAKE_PICTURE_CARD_NG 0x00008D07L
#define EDS_ERR_TAKE_PICTURE_CARD_PROTECT_NG 0x00008D08L

#define EDS_ERR_LAST_GENERIC_ERROR_PLUS_ONE 0x000000F5L

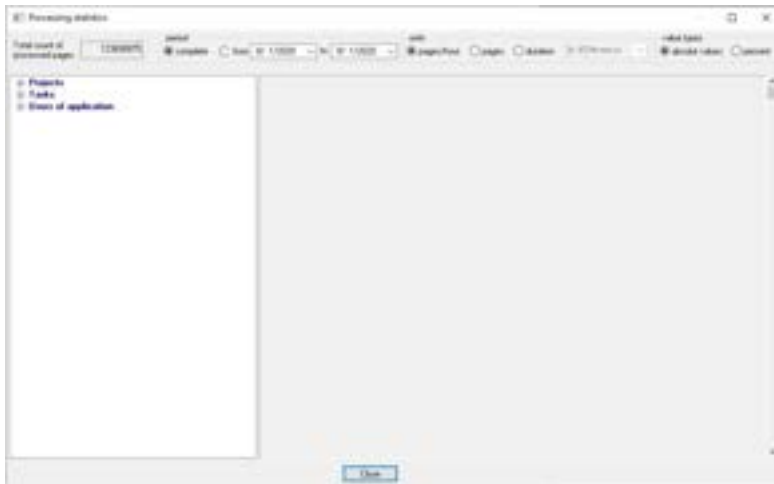
```

## 36 Statistics

With the **Statistics** tool in **Help** menu you get statistical records of the docWizz system. There are different ways to analyze work procedures, jobs or documents.

The statistic of docWizz visualizes the logged data about the processed documents and used time, inform about the behavior of the machines and users and so on.

Note: You need administrator login and password to use this functionality!



The new opened dialog contains three frames: on the top the dynamical properties for the statistic results, on the left the tree containing the information elements and on the right the draw area.

### Top bar



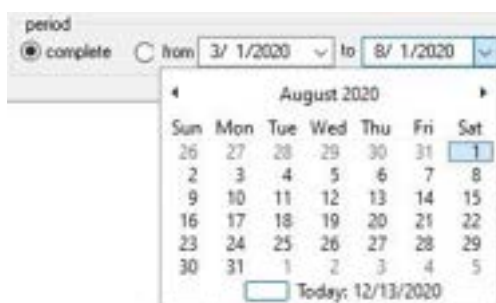
On the top bar you get the result of the total page counter. Beside this value you can choose between the following options:

### Total count of pages



This entry shows the number of processed pages and cannot be changed manually.

### Period

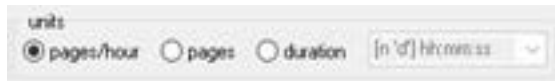


Here you can choose each period of time, from one day until the complete period dW was executed in your environment. You have the choice between complete, over a specific date or fixed period.

**complete** Means the whole period, system is executed.

**from .... to** Here you specify a specific date or fixed period.

### Units (of the diagrams)



**pages/hour** The statistical analysis average of completed pages per hour.

**pages** The number of completed pages.

**duration** Time, which was needed. You can select **[n 'd'] hh:mm:ss** or **sec**

### Value types



**absolute values** The calculated value in his unit. Output is displayed as time, page, hour or pages per hour.

**percent** Percentage of each element in the shown result.

### Tree frame

The tree frame shows a list of selectable elements. The tree is built by dynamic and static elements. The static element groups are:

- + Computers
- + Document types
- + Documents
- + Jobs
- + Projects
- + Users of application
- + Users of computer

Computers	contains computer name on which the job was executed
Document types	contains doc types, e.g. Monograph, Serial
Documents	contains documents (Name + ID)
Jobs	contains Jobs, e.g. DetectLayoutElements, Export
Projects	contains dW Projects
Users of application	contains user of dW
Users of computer	contains user of the operating system

The dynamic elements are the values of the parent element group, e.g. computers contains the list of all computers which processed pages in the defined period. On the right the result of the analyzed data and on the left the child elements, where you can select one of them to analyze in detail, why this computer was so quick or need more time than the others. With the closer look, you can determine which documents were processed, which job needed the most time on that computer or which user processed these documents on that computer, and so on.

It is possible to configure the visible static element groups and the deep of their sub elements.

The configuration of the tree is stored in MiscCfg/Statistics.xml

### Right - draw area

On the right the diagram is drawn for the selected element in the tree. The results are shown in pie and beam diagrams. The beam diagrams are now displayed horizontal.



Note: It is possible to configure the visible static element groups and the deep of their sub elements by a XML structure in the META-E-DW.ini.

## 36.1 Configuration of the Tree

The static group elements can be configured by a XML structure in the docWizz-DW.ini.

The parent element for this section is called STATISTICS.

Generally this section has the following behaviour.

- Is no element defined for an element level, all possible elements will be shown. So also for the root element it means, that if no subelement is insert in the <STATISTICS>-tag, all elements will be shown.
- Is one element at least defined, only the defined elements will be insert in the tree.
- Contains the element an attribute "final", no subelements will be insert in the tree.

```
<STATISTICS>
  <COMPUTERS>
    <JOBS>
      <DOCUMENTS>
        <USERS_APPL final="1"/>
        <JOBS/>
        <USERS_COMP/>
      </DOCUMENTS>
    </JOBS>
  </COMPUTERS/>
</COMPUTERS>
<DOCTYPES final="1">
  <USERS_APPL>
    <DOCUMENTS/>
    <COMPUTERS final="1"/>
  </USERS_APPL>
</DOCTYPES>
<DOCUMENTS/>
<PROJECTS/>
<JOBS/>
<USERS_APPL>
  <DOCUMENTS/>
  <COMPUTERS/>
</USERS_APPL>
<USERS_COMP admin="1"/>
</STATISTICS>
```



Like in the configuration defined, the tree Computers -> Jobs -> Documents -> Users of application is restricted and no "Users of computer" is shown.

## 36.2 Further functionalities

At the moment there exists no further functionalities. A print would be able by copying the dialog into the clipboard ( Alt + Print ) and insert it into an other program.

No data will be stored or changed by this additional function.

## 36.3 Behavior in case of errors

The docWizz Statistic dialog contains an own filter for errors.

If an unexpected error appears the dialog would be closed without a message. Perhaps in the log database an error will be logged. The user can continue his work with docWizz without any problems.

In case of expected errors (f.e. error in database requests) the error will be shown and logged in the log database.

## 37 Document processing Log (BatchResults table)

Whenever a document is loaded into dW, after releasing the document or moving to another job, a record is added to the log database which stores information on the document, the processing time and the result of processing. As well the record is added into the documents private history in its main XML file.

### Example:

Filter2	Result	ComputerName
Monografi_Default_Antiqua_Auto/digbok_200805260003	1	OCR-SERVER7
Monografi_Default_Antiqua_Auto/digbok_200805260003	10	OCR-SERVER3
Monografi_Default_Antiqua_Auto/digbok_200805260003	10	OCR-SERVER1
Monografi_Default_Antiqua_Auto/digbok_200805260003	10	OCR-SERVER1
Monografi_Default_Antiqua_Auto/digbok_200805260003	10	OCR-SERVER8
Monografi_Default_Antiqua_Auto/digbok_200805260003	10	OCR-SERVER8
Monografi_Default_Antiqua_Auto/digbok_200805260003	10	OCR-SERVER8
Monografi_Default_Antiqua_Auto/digbok_200805260003	10	OCR-SERVER8
Monografi_Default_Antiqua_Auto/digbok_200805260003	31	NBR-OCR1

Data fields in the log database:

#### ID

An unique identifier for each log entry.

#### JobName

Name of the job the document was in.

#### DocumentID

The unique id of the document.

#### CompletedPages

The number of pages in the document.

#### StartTime

The time when the document has been loaded or processing has started.

#### Duration

The duration of processing in seconds. Please note that in interactive jobs, idle time (more than 20 seconds no mouse or keyboard input) is ignored.

#### Filter1

The content of filter1 of the document, by default the document type.

#### Filter2

The content of filter2 of the document, by default the document title.

#### Result

A status, how processing was terminated:

-1	An error occurred during batch processing
1	Document successfully created by AutoImport
10	Batch processing terminated successfully
11	Batch processing terminated successfully, document has finally been deleted
12	Batch rejected

20	User has edited and saved the document
21	User has edited the document and started processing
22	User has routed the document to a different job
23	User has discarded his changes
24	An error has occurred during StepByStep processing
25	StepByStep processing terminated successfully
26	Export in StepByStep processing terminated successfully
27	Delete in StepByStep processing terminated successfully, document has finally been deleted
28	Processing canceled by user
29	Editing task completed
30	Task was resetted
31	Document manually deleted by user
32	Restore document
40	Interactive send to RQA
41	Interactive resend to RQA
42	Interactive reduce pool data
43	Interactive change status
44	Route to job
45	RQA Manager send to LDR
46	RQA Manager receive from LDR
47	RQA LDR receive from manager
48	RQA LDR send to manager

### **ComputerName**

The computer name, where the document was processed on.

### **ComputerUserName**

The name of the user that was logged on to windows.

### **ApplicationUser**

The name of the user that was logged on to dW. By default a user logs on to dW with the same name like the user is logged on to windows.

### **WasBatch**

0 if the processing was performed in dW client application in StepByStep mode or in an interactive job.

### **Filter3**

The content of filter3 of the document, by default the project name.

### **P\_LEVEL**

Processing level of the document. A bit flag value which tells the processing features applied to the document.

### **Status**

C	critical error
D	discuss

E	error
F	free pool
H	hold
I	restore pool
Q	QA
R	reject
S	reduced pool
V	review
W	default

#### **RQAStatus**

A	remote QA
B	to sent back
D	remote QA corrction done
E	default
F	delete remote QA
O	wait for recorection
P	remote QA progress
Q	request send back
R	remote QA correction
S	resend remote QA

#### **Actions**

M#K#

#	represents an number
M#	number of mouse operations
K#	number of keyboard operations

With Route the job name is: "fromJob->toJob"

With Task the job name is: "Job:Task"

### **37.1 Active / Idle status**

Active is counted always if the application is no longer in idle state than 20 secs. Whenever idle state is longer than 20 secs, time is counted as idle, including the 20 secs waiting time (active counter will decrease by 20, idle will start with 20). All times are summed unless the document remains in the same editing state. When the editing state changes, a report is generated in batchresults including active time and timers are reset.

## 38 Abbreviations

Step	Task (manually / background)	Abbreviation
------	------------------------------	--------------

RESCAN	Rescan	R-RS
	Apply rescan	R-ARS

IMPORT	Setup import	I-SI
	Process import	I-Procl
	Review import	I-RI

CROPPING	Recognize page format	C-RecPF
	Prepare cropping	C-PC
	Process cropping	C-ProcC
	Review cropping	C-RC

ZONING	Recognize layout	Z-RecL
	Review zoning	Z-RZ
	Recognize page sequence	Z-RecPS
	Review page sequence	Z-RPS

STRUCTURE	Recognize issues	S-RecI
	Review issues	S-RI
	Recognize structure and text	S-RecST
	Review structure and text	S-RST

OUTPUT	Prepare output	O-PrepO
	Process output	O-ProcO
	Review output	O-RO

CLEANUP	Process backup	CL-ProcB
	Process cleanup	CL-ProcC

Copyright © 2022 CCS Content Conversion Specialists GmbH. All rights reserved.

No part of this publication may be reproduced, stored in databases, or transferred in any form (electronically, photo-mechanically, chemically, manually, or otherwise) without the express written permission of CCS Content Conversion Specialists GmbH. The software described in this manual is licensed software that may be used only in compliance with the licensing terms and conditions. CCS GmbH reserves the right to make changes to the content of this manual without notice. CCS GmbH makes no guarantee regarding the accuracy of the information provided in this manual. Microsoft, and Windows are registered trademarks of the Microsoft Corporation.

Product or company names that are mentioned may be trademarks or registered trademarks of the respective company. CCS GmbH uses these names and trademarks in the following manual merely for explanatory purposes and for the benefit of the respective user, and such use does not imply trademark infringement.

Under this software license, you are only permitted to reproduce materials that are not protected by copyright laws. This excludes only materials where you hold the copyright and/or legal permission to reproduce copyrighted materials. If you are uncertain about the copyright status of certain materials then please seek legal counsel. CCS GmbH holds no liability over copyright violations resulting from the use of this software.

Last updated: 01.03.2022

**CCS Content Conversion Specialists GmbH**

Weidestraße 134

22083 Hamburg, Germany

Phone: +49-(0)40-228582990

E-Mail: [info@content-conversion.com](mailto:info@content-conversion.com)

Website: [www.content-conversion.com](http://www.content-conversion.com)