SavaPage User Manual

Version 1.7.0

Rijk Ravestein

SavaPage User Manual : Version 1.7.0

SavaPage User Manual © 2011-2026 by Rijk Ravestein is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License[1].

SavaPage Open Print Portal Software © 2011-2026 by Datraverse B.V.[2] is OSI Certified Open Source Software[3], licensed under the terms of the GNU Affero General Public License (AGPL)[4] version 3, or any later version, in compliance with Third Party Software Licenses[5]. “OSI Certified” is a certification mark of the Open Source Initiative[6].

[1] https://creativecommons.org/licenses/by-sa/4.0/
[2] https://www.datraverse.com
[3] https://opensource.org/
[4] https://www.gnu.org/licenses/agpl.html
[5] https://www.savapage.org/docs/licenses/
[6] https://opensource.org/

Table of Contents

Preface xxiv
1. About this Manual xxiv
2. Expectations and Prerequisites xxiv
3. Conventions used in this Document xxiv
3.1. Typographical Conventions xxiv
3.2. Feature Preview xxv
3.3. Notes xxv
4. Screenshots xxvi
5. Notice xxvi
6. Your Feedback xxvi
1. Introduction 1
1.1. What is SavaPage? 1
1.1.1. Open Source Software 1
1.1.2. Benefits 1
1.1.3. Key Features 2
1.2. System Requirements 3
1.2.1. Server 3
1.2.2. Clients 6
1.3. How does SavaPage work? 6
1.3.1. Key Concepts 6
1.3.2. The SavaPage Work Flow 8
1.3.3. Architecture Overview 9
2. Server Installation 11
2.1. Step 1 - System Requirements 11
2.2. Step 2 - Create System Account 11
2.3. Step 3 - Configure CUPS and Samba 12
2.3.1. CUPS Remote Printer Browsing 13
2.3.2. CUPS Job History 13
2.3.3. CUPS Job ID 13
2.3.4. CUPS Job Privacy 14
2.3.5. CUPS Web Interface 15
2.3.6. CUPS security fix 15
2.3.7. Test CUPS 16
2.4. Step 4 - Optional System Settings 16
2.4.1. Set Default Paper Size 16
2.5. Step 5 - Check Firewall Settings 16
2.6. Step 6 - Download and Install 16
2.7. Step 7 - Save Encryption Keys 17
2.8. Step 8 - Configure 17
2.8.1. Step 1 - Login 17
2.8.2. Step 2 - Change Admin Password 18
2.8.3. Step 3 - Set Locale 18
2.8.4. Step 4 - Set Currency Code 18
2.8.5. Step 5 - Set User Source 18
2.8.6. Step 6 - User Synchronization 19
2.8.7. Step 7 - Set Mail Options 19
2.8.8. Step 8 - Set Driverless File Transfer Print 19
2.9. Step 9 - Share SavaPage Client Files 19
2.10. Step 10 - Testing 20
2.11. What's next? 20
3. User Web App 22

3.1. Login ... 22
3.1.1. About ... 23
3.1.2. Select Language ... 24
3.1.3. Self Registration ... 24
3.1.4. Single Web App Session ... 26
3.1.5. Card Self Association Dialog ... 26
3.1.6. 2-Step Verification ... 27
3.2. Login Alternatives ... 28
3.3. SafePages ... 29
3.3.1. Document Expiration ... 31
3.3.2. Footer ... 32
3.3.3. Document Details ... 34
3.3.4. Browser ... 37
3.4. PDF ... 40
3.4.1. PDF Filters ... 42
3.4.2. Links ... 42
3.4.3. Document Scope ... 42
3.4.4. Description ... 43
3.4.5. Security ... 43
3.4.6. Passwords ... 45
3.4.7. Letterhead ... 46
3.4.8. Download ... 46
3.4.9. Send ... 46
3.5. Print ... 47
3.5.1. Printer Selection ... 47
3.5.2. Printer Settings ... 48
3.5.3. Selected Printer ... 52
3.5.4. Delegated Print ... 53
3.5.5. Print Job Settings ... 53
3.5.6. Direct Print Release ... 57
3.5.7. Full Print Scope and Jobs ... 57
3.5.8. Delegated Print Edit ... 57
3.5.9. Job Ticket Print ... 64
3.6. Letterheads ... 68
3.7. Delete ... 70
3.8. Log ... 71
3.8.1. Documents ... 71
3.8.2. Transactions ... 73
3.9. Sort ... 77
3.10. User Details ... 78
3.10.1. Internet Printer ... 79
3.10.2. Telegram Configuration ... 80
3.10.3. Authentication ... 81
3.10.4. Pagometers ... 82
3.10.5. Financial ... 83
3.10.6. Redeem Voucher ... 84
3.10.7. Transfer Credit ... 84
3.10.8. Transfer Money ... 85
3.11. Upload ... 85
3.11.1. Upload Dialog ... 86
3.12. Upload Drop Zone ... 87
3.13. GDPR Dialog ... 88
4. Admin Web App ... 89
4.1. Login ... 89

4.2. Menu 89
4.3. Dashboard 91
4.3.1. Status 92
4.3.2. Services 95
4.3.3. News 95
4.3.4. Pagometers 96
4.3.5. Environmental Impact 96
4.3.6. Financial Summary 97
4.3.7. Filtered Requests 97
4.3.8. Real-time Activity 98
4.4. Users 98
4.4.1. User List 98
4.4.2. Download Personal Data 101
4.4.3. Erased Users 101
4.4.4. Registration Requests 102
4.4.5. Edit User 102
4.4.6. Create Internal User 108
4.4.7. Deleted Users 109
4.4.8. Administrator Role 109
4.5. Groups 109
4.5.1. Built-in Groups 110
4.5.2. Group List 110
4.5.3. Add & Remove Groups 112
4.5.4. Edit Group 112
4.6. Accounts 117
4.6.1. Account List 117
4.6.2. Edit Account 119
4.7. Queues 120
4.7.1. Queue List 120
4.7.2. Edit Queue 123
4.7.3. IPP Routing 124
4.7.4. Reserved Queues 126
4.8. Proxy Printers 127
4.8.1. Proxy Printer List 127
4.8.2. Edit Proxy Printer 132
4.8.3. Printer Groups 137
4.8.4. Raw Print 138
4.8.5. Rename Proxy Printer 139
4.9. Devices 139
4.9.1. Network Card Reader 141
4.9.2. Terminal 146
4.10. Reports 149
4.11. Options 149
4.11.1. User Source 150
4.11.2. Custom 154
4.11.3. User Creation 155
4.11.4. User Authentication 157
4.11.5. Mail 162
4.11.6. Mail Print 164
4.11.7. Web Print 166
4.11.8. Internet Print 167
4.11.9. Proxy Print 168
4.11.10. Eco Print 170
4.11.11. Financial 172

4.11.12. Integration 175
4.11.13. Backups 177
4.11.14. Advanced 178
4.11.15. Config Editor 189
4.12. Documents 190
4.12.1. Document List 191
4.12.2. Proxy Print Reversal 196
4.12.3. Proxy Print Status 196
4.13. Log 197
4.14. About 199
4.14.1. Version 199
4.14.2. License 200
4.14.3. Privacy 200
4.14.4. Community 201
4.14.5. Organization 201
4.14.6. Support 201
4.14.7. Java 202
4.14.8. JavaScript 203
4.14.9. Environment 203
4.14.10. Host System 204
4.14.11. Host Packages 204
4.15. Vouchers 206
4.15.1. Voucher Actions 207
4.15.2. Create Vouchers 209
4.15.3. Voucher Usage 209
5. Job Tickets Web App 211
5.1. Login 211
5.2. Open Tickets 211
5.2.1. Select and Sort 211
5.2.2. Ticket List 213
5.2.3. Job Ticket Bulk Actions 214
5.2.4. Job Ticket Transactions 215
5.2.5. Job Ticket Edit 215
5.2.6. Job Ticket Print 217
5.2.7. Job Ticket Settle 219
5.3. Closed Tickets 220
5.3.1. Job Ticket Refund 220
5.3.2. Job Ticket Reversal 221
5.3.3. Job Ticket Reopen 221
5.4. Job Ticket Configuration Properties 222
6. Mail Tickets Web App 224
6.1. Login 224
6.2. Mail Ticket Selection 224
6.3. Mail Ticket Configuration Properties 225
7. Point-of-Sale Web App 226
7.1. Login 226
7.2. Sales 226
7.3. Deposit 227
7.4. Receipts 229
7.5. Configuration Properties 230
8. Payment Web App 232
8.1. Login 232
8.2. Payment Page 232
8.2.1. Payment Dialog 233

9. Print Site Web App 234
9.1. Login 234
9.2. PaperCut Integration 235
9.2.1. Users 235
9.2.2. Financial 235
9.2.3. Printing 236
10. PDF/PGP Verification 237
10.1. PDF/PGP in a Nutshell 237
10.2. PDF/PGP Signature 237
10.3. PDF/PGP Verification Web App 238
11. User Client 240
11.1. User Client Options 241
11.2. User Client Deployment 242
11.2.1. Deployment on Windows 242
11.2.2. Deployment on macOS 242
11.2.3. Deployment on GNU/Linux 242
12. SavaPage Financial 243
13. SavaPage on GNU/Linux 244
13.1. The Installation Process 244
13.1.1. Manual extraction 244
13.1.2. The install process 244
13.2. Logs 246
13.3. OpenPGP Settings 247
13.4. IP Address Restriction 248
13.4.1. CIDR Notation 248
13.4.2. CIDR Set 248
13.5. Advanced Configuration 248
13.5.1. server.properties 248
13.5.2. log4j.properties 249
13.5.3. Alternative TCP/IP Settings 249
13.5.4. HTTP/2 Settings 251
13.5.5. HTTP XFF Header Settings 251
13.5.6. Database Connection Settings 252
13.5.7. CUPS Settings 253
13.5.8. Server-side Download 254
13.5.9. RESTful Client 255
13.5.10. User Home Clean 255
13.5.11. Database Locking 256
13.5.12. pdf2cairo Image Strategy 256
13.5.13. Alternative File Locations 256
13.5.14. Miscellaneous Settings 257
13.5.15. System Environment 257
13.6. Upgrading SavaPage 258
13.7. Removing SavaPage from a GNU/Linux server 258
14. SavaPage as Printer 260
14.1. IPP Everywhere and AirPrint 260
14.1.1. Step 1: Check Avahi configuration 260
14.1.2. Step 2: Create AirPrint Queue 261
14.1.3. Step 3: Create Avahi Service File 261
14.2. PostScript Driver Print 262
14.2.1. IPP Options from PostScript 262
14.3. Printing from GNU/Linux 263
14.4. Printing from macOS 264
14.5. Printing from Chrome OS 264

14.6. Printing from Android ... 264
14.6.1. SavaPage Android Print App ... 265
14.7. Printing from iOS and iPadOS ... 267
14.8. Printing from Windows ... 267
14.8.1. IPP Printer ... 267
14.8.2. Raw Printer ... 268
14.9. Driverless File Transfer Print ... 269
14.9.1. Processing Driverless PDF Transfer ... 269
14.10. PostScript Print Processing ... 271
14.10.1. Licensed Fonts ... 271
14.10.2. Printing Encrypted PDF ... 272
14.10.3. PostScript Configuration Properties ... 272
14.11. IP Restricted Printing ... 273
14.12. Printer Availability ... 273
15. Authenticated Printing ... 274
15.1. Key Concepts ... 274
15.1.1. User ... 274
15.1.2. Person ... 274
15.1.3. Domain User ... 274
15.1.4. Synchronized User ... 274
15.1.5. Synchronized Person ... 274
15.1.6. Internal Person ... 274
15.1.7. Authenticated User ... 274
15.1.8. Authenticated Abstract User ... 274
15.1.9. Authenticated Person ... 275
15.1.10. Trusted SavaPage Queue ... 275
15.1.11. Public SavaPage Queue ... 275
15.1.12. Standard IP Based Authentication ... 275
15.1.13. Basic Authentication ... 275
15.1.14. Mail Print Authentication ... 276
15.1.15. Local User ... 276
15.1.16. Local Abstract User ... 276
15.1.17. Local Person ... 276
15.1.18. User Alias ... 276
15.2. Abstract User ... 276
15.3. Single Sign-On Domains ... 276
15.3.1. Authentication Loopholes ... 277
15.3.2. Unauthenticated Users ... 278
15.4. Peer to Peer Networks ... 279
15.5. User Name Aliases ... 280
16. Printing Impact ... 282
16.1. Financial Impact ... 282
16.2. Environmental Impact ... 282
16.2.1. Printed Sheet Units ... 282
16.2.2. Trees ... 283
16.2.3. Energy ... 283
16.2.4. Carbon ... 283
17. Security ... 284
17.1. HTTP Service Filters ... 284
17.1.1. Denial of Service Filter ... 284
17.1.2. Inet Access Filter ... 285
17.1.3. HTTP/2 Rate Control ... 286
17.2. User Authentication ... 286
17.2.1. Login Passwords ... 286

17.2.2. PIN Codes ... 287
17.2.3. Authentication Tokens ... 287
17.2.4. One-time Authentication Tokens ... 287
17.2.5. User Dialog ... 287
17.3. Access over Internet ... 287
17.3.1. security.txt ... 288
17.4. Web Sessions ... 289
17.4.1. Web Session Timeout ... 289
17.4.2. Web Session Cookies ... 289
17.4.3. Web App WatchDog ... 290
17.5. Rate Limiting ... 290
17.6. SSL Certificate and Password ... 291
17.7. Secured JMX Connection ... 291
17.8. Encrypted Secrets ... 292
17.9. Document Signature ... 292
17.10. User Client ... 293
17.11. Server Commands ... 293
17.12. Web Services ... 293
17.13. Log Files ... 293
17.14. Network Card Reader ... 293
17.15. Internal Services ... 294
17.16. External Services ... 294
17.17. Vouchers ... 294
18. Freedom and Privacy ... 295
18.1. Open Source ... 295
18.1.1. Libre JavaScript ... 295
18.2. General Data Protection Regulation ... 295
18.2.1. Data Portability ... 296
18.2.2. Data Erasure ... 296
18.3. Secure Print Release ... 296
18.4. CUPS Privacy ... 296
19. Internationalization ... 297
19.1. Localization ... 297
19.1.1. Notes for Translators ... 297
19.2. Internal Fonts ... 297
19.2.1. Default Font ... 298
19.2.2. CJK Font ... 298
19.2.3. Unifont ... 298
20. Customization ... 300
20.1. Custom Web App ... 300
20.1.1. Web App Look-and-feel ... 300
20.2. Email Templates ... 304
20.2.1. Email Template Syntax ... 305
20.2.2. Email Stationary Template ... 306
20.2.3. Email Message Template ... 306
20.2.4. Email Placeholders Objects ... 307
20.2.5. Email Stationary Types ... 308
20.2.6. Email Message Types ... 309
20.2.7. Custom Template Locations ... 309
21. Using an External Database ... 311
21.1. Supported Databases ... 311
21.2. Migrating to an External Database ... 311
21.2.1. Step 1 - Stop SavaPage ... 312
21.2.2. Step 2 - Create a Backup ... 312

21.2.3. Step 3 - Create new Database in External DBMS ... 312
21.2.4. Step 4 - Change SavaPage Connection Parameters ... 312
21.2.5. Step 5 - Initialize new Database ... 313
21.2.6. Step 6 - Restore Backup into new Database ... 313
21.2.7. Step 7 - Restart SavaPage ... 313
22. Tuning ... 314
22.1. Linux Kernel Parameters ... 314
22.1.1. IP Ports ... 314
22.1.2. TCP Buffer Sizes ... 314
22.1.3. Queue Sizes ... 315
22.1.4. Congestion Control ... 315
22.1.5. Setting Linux kernel parameters with sysctl ... 315
22.2. Linux User Limits ... 316
22.2.1. SysVinit User Limits ... 316
22.2.2. Systemd User Limits ... 317
22.3. JVM Tuning ... 318
22.3.1. JVM Memory Allocation ... 318
22.3.2. JVM Garbage Collection ... 318
22.3.3. JVM Temporary Files ... 319
22.4. Server Thread Pooling ... 319
22.5. Database Connection Pooling ... 319
23. SavaPage Community ... 320
23.1. Importing the Service Card ... 320
A. Proxy Print Scenarios ... 321
A.1. Proxy Print Costs ... 321
A.2. Personal Print Scenarios ... 321
A.2.1. Personal Print - Non-Secure Scenarios ... 321
A.2.2. Personal Print - Secure Scenarios ... 321
A.2.3. Personal Print - PaperCut Scenario ... 321
A.3. Delegated Print Scenarios ... 322
A.3.1. Delegated Print - (Non) Secure & Job Ticket Scenarios ... 323
A.3.2. Delegated Print - Job Ticket - PaperCut - Scenario ... 324
A.3.3. Delegated Print - PaperCut Scenario ... 325
B. NFC Authentication ... 326
B.1. Card Number Format ... 326
B.2. Local Card Reader ... 326
B.3. Network Card Reader Service ... 327
C. Tools ... 328
C.1. Server Commands ... 328
C.1.1. Common Options ... 329
C.1.2. addInternalUser ... 331
C.1.3. addUserGroup ... 332
C.1.4. changeBaseCurrency ... 332
C.1.5. deleteUser ... 333
C.1.6. deleteUserGroup ... 333
C.1.7. deleteUserGroupAccount ... 333
C.1.8. eraseUser ... 334
C.1.9. getConfigProperty ... 334
C.1.10. listUsers ... 335
C.1.11. listUserGroups ... 335
C.1.12. listUserGroupMembers ... 335
C.1.13. listUserGroupMemberships ... 336
C.1.14. listUserSourceGroups ... 336
C.1.15. listUserSourceGroupMembers ... 336

C.1.16. listUserSourceGroupNesting ... 337
C.1.17. printerAccessControl ... 337
C.1.18. printerSnmp ... 338
C.1.19. setConfigProperty ... 338
C.1.20. setUserProperties ... 338
C.1.21. setUserGroupProperties ... 340
C.1.22. syncUserGroup ... 340
C.1.23. syncUsersAndGroups ... 341
C.1.24. systemStatus ... 341
C.2. Web Services ... 341
C.2.1. XML-RPC ... 342
C.2.2. JSON-RPC ... 343
C.2.3. RESTful API ... 346
C.3. Atom Feed Service ... 352
C.4. Matrix Messaging Service ... 352
C.5. Database Commands ... 353
C.5.1. db-check ... 354
C.5.2. db-check-fix ... 354
C.5.3. db-config-get ... 354
C.5.4. db-config-set ... 354
C.5.5. db-delete-logs ... 354
C.5.6. db-export and db-export-to ... 355
C.5.7. db-import ... 355
C.5.8. db-init ... 356
C.6. Stopping and Starting the Server ... 356
C.7. SSL Key Generation ... 356
C.7.1. Re-Create the Self-Signed Certificate ... 357
C.7.2. Importing an Existing SSL Certificate ... 358
C.7.3. Installing the Keystore ... 359
D. Capacity Planning ... 360
D.1. Database Sizing and Growth ... 360
D.2. SafePages Sizing and Growth ... 361
D.3. Network Bandwidth Planning ... 361
E. URL Cheat Sheet ... 362
F. File Locations ... 365
G. Printable File Types ... 369
G.1. Standard File Types ... 369
G.1.1. XPS to PDF Installation Instructions ... 370
G.2. Advanced File Types ... 370
G.3. PDF Font Substitution ... 371
H. Upgrading from a Previous Version ... 373
H.1. Upgrading the Server ... 373
H.2. Upgrading Client Printer Drivers ... 373
H.3. Testing the Upgrade ... 373
I. Migrating to a New Server ... 374
I.1. Upgrade Old Server ... 374
I.2. Install New Server ... 374
I.3. Freeze Old Server ... 374
I.4. Migrate Data to New Server ... 374
I.5. Rename Printers ... 375
I.6. Update SavaPage Printers ... 375
J. Advanced LDAP Configuration ... 376
J.1. LDAP Server Default Configuration ... 377
J.1.1. OpenLDAP ... 377

J.1.2. FreeIPA ... 377
J.1.3. Apple Open Directory ... 377
J.1.4. Novell eDirectory Defaults ... 378
J.1.5. Microsoft Active Directory Defaults ... 378
J.1.6. Google Cloud Directory ... 380
K. PPD Extensions ... 382
K.1. PPD to IPP Mappings ... 382
K.1.1. Mapping PPD to IPP ... 383
K.1.2. Mapping PPD to IPP Extensions ... 385
K.1.3. Restricting Standard Options ... 385
K.2. PPD Rules ... 386
K.2.1. Generic PPD Rules ... 386
K.2.2. Custom PPD Rules ... 387
K.3. IPP Rules ... 390
K.3.1. SPConstraint ... 390
K.4. Job Extensions ... 391
K.4.1. Job Copy Options ... 391
K.5. Job Ticket Extensions ... 391
K.5.1. Job Ticket Media Options ... 392
K.5.2. Job Ticket Sheet Options ... 393
K.5.3. Job Ticket Copy Options ... 394
K.5.4. Job Ticket Set Options ... 395
K.6. Tips and Tricks ... 396
K.6.1. Fast Print A4 and Letter to Single Tray ... 396
K.7. proxy-print.log ... 396
L. IPP Extensions ... 397
L.1. Internal IPP Extensions ... 397
L.1.1. Internal IPP - PPD Mapping Extensions ... 397
L.1.2. Internal IPP Job Extensions ... 400
L.1.3. Internal IPP Job Ticket Extensions ... 401
L.1.4. Internal IPP Marker ... 403
L.2. External IPP Extensions ... 404
L.3. IPP Localization ... 404
M. SavaPage Plug-ins ... 405
M.1. Web API Callback Plug-in ... 405
M.1.1. Payment Gateway Plug-in ... 405
M.2. OAuth Client Plug-in ... 406
M.3. Mail Print IMAP OAuth Plug-in ... 407
M.4. Notification Plug-in ... 407
M.5. IPP Routing Plug-in ... 407
M.6. User Source Plug-ins ... 410
M.6.1. User Synchronization Plug-in ... 410
M.6.2. User Authentication Plug-in ... 411
N. PaperCut Integration ... 412
N.1. Delegated Print to PaperCut ... 412
N.1.1. PaperCut Configuration ... 412
N.1.2. PaperCut Delegated Print Processing ... 413
N.1.3. PaperCut Delegated Print Accounting ... 414
N.1.4. PaperCut Queries and Reports ... 415
N.2. Delegated Print to PaperCut Front-end ... 416
N.3. Personal Print to PaperCut ... 416
N.4. Advanced Print Configuration ... 417
N.4.1. PaperCut Print Log Monitoring ... 417
N.5. User Source Solutions ... 417

N.5.1. SavaPage as Sync Source in PaperCut 417
N.5.2. PaperCut as User Source in SavaPage 418
N.6. PaperCut Personal User Account 418
N.7. Integration Pitfalls 419
O. Job Scheduling 420
O.1. Cron Trigger Format 420
O.2. Hourly Jobs 420
O.3. Daily Jobs 420
O.4. Daily Maintenance 421
O.5. Weekly Jobs 421
O.6. Monthly Jobs 421
O.7. Atom Feed Job 421
O.8. Matrix Summary Job 421
P. GNU Affero General Public License (AGPL) 422

List of Figures

1.1. SavaPage High-Level Architecture 9
2.1. CUPS Job Privacy 14
3.1. Web App: Login Dialog 22
3.2. Web App: Select Language Dialog 24
3.3. Web App: Register Option 24
3.4. Web App: Register Dialog 25
3.5. Self Registration Verification Email 25
3.6. Self Registration Completed 26
3.7. Same type Web App session detected 26
3.8. Web App type change detected 26
3.9. Web App: Login Dialog - Card Self Association 27
3.10. Web App: Login Dialog - 2-Step Verification 27
3.11. Web App: Login Dialog - 2-Step Verification (recovery) 27
3.12. Web App: Login Dialog - ID Number 28
3.13. Web App: Login Dialog - Local NFC Card 28
3.14. Web App: Login Dialog - Network NFC Card 29
3.15. User Web App: Main View 29
3.16. User Web App: SafePages 30
3.17. User Web App: SafePages - Aggregated 31
3.18. User Web App: Footer Base 32
3.19. User Web App: Hold Print Jobs 33
3.20. User Web App: Hold Copy Job 34
3.21. User Web App: Hold Job Transactions 34
3.22. User Web App: Document Details 35
3.23. User Web App: Document Original Details 35
3.24. User Web App: Landscape Job 36
3.25. User Web App: Rotated Pages 37
3.26. User Web App: SafePage Browser (8 pages) 37
3.27. User Web App: SafePage Browser - Page Options 38
3.28. User Web App: SavaPage Draw 39
3.29. User Web App: PDF - Overview 41
3.30. User Web App: PDF - Links 42
3.31. User Web App: PDF - Document Scope 42
3.32. User Web App: PDF - Description 43
3.33. User Web App: PDF - Security 44
3.34. User Web App: PDF - Passwords 45
3.35. User Web App: PDF - Letterhead 46
3.36. User Web App: PDF - Send 46
3.37. User Web App: Print - Select Printer 47
3.38. User Web App: Printer - Settings 48
3.39. User Web App: Printer - Settings - Finishings 49
3.40. User Web App: Printer - Settings - Job Ticket 49
3.41. User Web App: Print - Page Scaling (Fit) 50
3.42. User Web App: Print - Page Scaling (None) 50
3.43. User Web App: Print - Preview (4-up, landscape, staple, punch) 51
3.44. User Web App: Printer Options - Custom Icons 51
3.45. User Web App: Print - Selected Printer 52
3.46. User Web App: Print - Job Settings 53
3.47. User Web App: Print - Delete Pages 54
3.48. User Web App: Print Confirmation 55
3.49. User Web App: Print Archive Option 56

3.50. User Web App: Printer - Direct Print Release 57
3.51. User Web App: Print Delegation Copies 58
3.52. User Web App: Delegated Print - Group Invoicing 59
3.53. User Web App: Delegated Print - Group copies 60
3.54. User Web App: Delegated Print - User Invoicing (Groups) 60
3.55. User Web App: Delegated Print - Personal Invoicing (Users) 61
3.56. User Web App: Delegated Print - Shared Account Invoicing (Groups) 61
3.57. User Web App: Print - Delegated Print - Shared Account Invoicing (Extra) 62
3.58. User Web App: Print - Select Job Ticket Printer 64
3.59. User Web App: Print - Job Ticket Settings - Print 64
3.60. User Web App: Print - Job Ticket Settings - Labels 65
3.61. User Web App: Print - Job Ticket Settings - Copy 65
3.62. User Web App: Print - Job Ticket Settings 66
3.63. User Web App: Print - Job Ticket - Sent 66
3.64. User Web App: Letterheads 68
3.65. User Web App: Letterhead - New 69
3.66. User Web App: Letterhead - Detail 70
3.67. User Web App: Delete SafePages 70
3.68. User Web App: Log - Documents 72
3.69. User Web App: Log - Document Transactions 72
3.70. User Web App: Log - Queue Journal 73
3.71. User Web App: Log - Transactions 73
3.72. User Web App: Log - SavaPage User Account Transaction 74
3.73. User Web App: Log - Proxy Print PaperCut Transaction 74
3.74. User Web App: Log - IPP Routing Transaction 74
3.75. User Web App: Log - Voucher Transaction 74
3.76. User Web App: Log - Payment Gateway Transactions 75
3.77. User Web App: Log - User Transfer Transactions 75
3.78. User Web App: Log - Transactions 76
3.79. User Web App: Sort 77
3.80. User Web App: User Details - Internet Printer 80
3.81. User Web App: User Details - Telegram Configuration 81
3.82. User Web App: User Details - Authentication 82
3.83. User Web App: User Details - pagometer 83
3.84. User Web App: User Details - Environmental Impact 83
3.85. User Web App: User Details - Financial 84
3.86. User Web App: Redeem Voucher 84
3.87. User Web App: Transfer Credit 85
3.88. User Web App: Transfer Money from Credit Card 85
3.89. Web Print: Upload File 86
3.90. Web Print: Drop Zone - Upload Dialog 87
3.91. Web Print: Drop Zone - Main 87
3.92. User Web App - GDPR Dialog 88
4.1. Admin Web App: Menu 90
4.2. Admin Web App: Action Pop-up Menu 91
4.3. Admin Web App: Dashboard - Membership 92
4.4. Admin Web App: Dashboard - System Mode 92
4.5. Admin Web App: Dashboard - Setup is needed 92
4.6. Admin Web App: Dashboard - Environment 93
4.7. Admin Web App: Dashboard - Users 93
4.8. Admin Web App: Dashboard - User Home Clean 93
4.9. Admin Web App: Dashboard - Sessions and Messages 93
4.10. Admin Web App: Dashboard - OpenPGP 94
4.11. Admin Web App: Dashboard - Tickets 94

4.12. Admin Web App: Dashboard - Technical Information 94
4.13. Admin Web App: Dashboard - Deadlocked Threads (intentionally caused as example) 94
4.14. Admin Web App: Dashboard - Services 95
4.15. Admin Web App: Dashboard - Pagometer 96
4.16. Admin Web App: Dashboard - Pagometer Trend 96
4.17. Admin Web App: Dashboard - Pagometer Print-in PDF 96
4.18. Admin Web App: Dashboard - Environmental Impact 97
4.19. Admin Web App: Dashboard - Financial Summary 97
4.20. Admin Web App: Dashboard - Filtered Requests 98
4.21. Admin Web App: Dashboard - Real-time Activity 98
4.22. Admin Web App: User - List 99
4.23. Admin Web App: User - Select and Sort 100
4.24. Admin Web App: User - Select Registration Requests 100
4.25. Admin Web App: User - List Item 101
4.26. Admin Web App: User - List Item - PGP Key 101
4.27. Admin Web App: User - PaperCut Account Balance 101
4.28. Admin Web App: User Data Portability 101
4.29. Admin Web App: Erased User 102
4.30. Admin Web App: Edit External User - Identity 102
4.31. Admin Web App: Edit User - Roles 103
4.32. Admin Web App: Edit User - Email 104
4.33. Admin Web App: Edit User - Card, Yubikey 104
4.34. Admin Web App: Edit User - ID number 105
4.35. Admin Web App: Edit User - UUID 105
4.36. Admin Web App: Edit User - OpenPGP 106
4.37. Admin Web App: Edit User - Financial 106
4.38. Admin Web App: Internal User - PIN Actions 106
4.39. Admin Web App: Internal User - PIN Reset 107
4.40. Admin Web App: Internal User - Password Actions 107
4.41. Admin Web App: Internal User - Password Reset 108
4.42. Admin Web App: Edit User - Delete 108
4.43. Admin Web App: User Group - List 110
4.44. Admin Web App: Group - Select and Sort 111
4.45. Admin Web App: Group - List Item 111
4.46. Admin Web App: User Groups - Add & Remove 112
4.47. Admin Web App: User Group - Edit - Roles 113
4.48. Admin Web App: User Group - Edit - User Privileges 114
4.49. Admin Web App: User Group - Edit - Admin Privileges 116
4.50. Admin Web App: User Group - Edit - New User Settings 116
4.51. Admin Web App: Account - List 117
4.52. Admin Web App: Account - List - Sub Accounts 117
4.53. Admin Web App: Account - SavaPage Account Balance 118
4.54. Admin Web App: Account - List - Select and Sort 118
4.55. Admin Web App: Account - Edit 119
4.56. Admin Web App: Queue - List 120
4.57. Admin Web App: Queue - Select and Sort 122
4.58. Admin Web App: Queue - Edit 123
4.59. Admin Web App: Queue - IPP Routing Terminal Options 124
4.60. Admin Web App: Queue - IPP Routing Printer Options 124
4.61. Admin Web App: Queue - List - IPP Routing: Terminal Queue 125
4.62. Admin Web App: Queue - List - IPP Routing: Printer Queue 125
4.63. Admin Web App: Queue - List - IPP Routing: Printer Hold Queue 125
4.64. Admin Web App: Queue - List - IPP Routing: Job Ticket Print Queue 125
4.65. Admin Web App: Proxy Printer - List Header 127

4.66. Admin Web App: Proxy Printer - List Items 128
4.67. Admin Web App: Proxy Printer - Select and Sort 131
4.68. Admin Web App: Proxy Printer - Edit - Identity 132
4.69. Admin Web App: Proxy Printer - PaperCut front-end 133
4.70. Admin Web App: Proxy Printer - Edit - Raw Print 133
4.71. Admin Web App: Proxy Printer - Edit -Job Ticket Printer 133
4.72. Admin Web App: Proxy Printer - Edit - Media Source 135
4.73. Admin Web App: Proxy Printer - Edit - Job Sheet Media Sources 136
4.74. Admin Web App: Proxy Printer - Edit - Manual Media Source 136
4.75. Admin Web App: Proxy Printer - Edit - Manual Media Size (Simple) 136
4.76. Admin Web App: Proxy Printer - Edit - Manual Media Size (Advanced) 137
4.77. Admin Web App: Proxy Printer - Rename 139
4.78. Admin Web App: Device - List 140
4.79. Admin Web App: Device - Select and Sort 141
4.80. Admin Web App: Devices - Network Card Reader - Custom User Login 142
4.81. Admin Web App: Devices - Network Card Reader - Proxy Print Authentication 143
4.82. Admin Web App: Devices - Terminal - Custom Proxy Print 146
4.83. Admin Web App: Devices - Terminal - Custom Proxy Print 147
4.84. Admin Web App: Devices - Terminal - Custom User Login 148
4.85. Admin Web App: Devices - Terminal - Custom User Login - Default 148
4.86. Admin Web App: Print Invoicing Report 149
4.87. Admin Web App: Options - User Source 150
4.88. Admin Web App: Options - User Source - LDAP 151
4.89. Admin Web App: Options - User Source - LDAP 152
4.90. Admin Web App: Options - Internal Users 153
4.91. Admin Web App: Options - User Source - Custom 154
4.92. Admin Web App: Options - User Creation - Import 155
4.93. Admin Web App: Options - User Creation - From Group 155
4.94. Admin Web App: Options - User Creation - Synchronize 156
4.95. Admin Web App: Options - User Creation - On Demand 157
4.96. Admin Web App: Options - User Authentication 157
4.97. Admin Web App: Options - User Authentication - Login Methods 158
4.98. Admin Web App: Options - User Authentication - Username Login 159
4.99. Admin Web App: Options - User Authentication - Email Login 159
4.100. Admin Web App: Options - User Authentication - ID Number Login 159
4.101. Admin Web App: Options - User Authentication - Local NFC Card Login 160
4.102. Admin Web App: Options - User Authentication - YubiKey Login 160
4.103. Admin Web App: Options - User Authentication - Default Login 161
4.104. Admin Web App: Options - User Authentication - 2-Step Verification 161
4.105. Admin Web App: Options - Mail - SMTP 162
4.106. Admin Web App: Options - Mail - Messages 162
4.107. Admin Web App: Options - Mail - Test 163
4.108. Admin Web App: Options - Mail Print (IMAP) 164
4.109. Admin Web App: Options - Mail Print (Attachments) 165
4.110. Admin Web App: Options - Web Print 166
4.111. Admin Web App: Options - Internet Print 167
4.112. Admin Web App: Options - Proxy Print General 168
4.113. Admin Web App: Options - Proxy Print Modes 169
4.114. Admin Web App: Options - Proxy Print Delegation 170
4.115. Admin Web App: Options - Proxy Print Preprocessing 170
4.116. Admin Web App: Options - Eco Print 172
4.117. Admin Web App: Options - Financial - Currency 172
4.118. Admin Web App: Options - Financial - General 173
4.119. Admin Web App: Options - Financial - POS 173

4.120. Admin Web App: Options - Financial - Vouchers 174
4.121. Admin Web App: Options - Financial - Transfer funds 174
4.122. Admin Web App: Options - PaperCut Integration 175
4.123. Admin Web App: Options - PaperCut Server 175
4.124. Admin Web App: Options - PaperCut Database 176
4.125. Admin Web App: Options - PaperCut Delegated Print Integration 176
4.126. Admin Web App: Options - PaperCut Personal Print Integration 177
4.127. Admin Web App: Options - Backups 177
4.128. Admin Web App: Options - Automatic Backups 178
4.129. Admin Web App: Options - Advanced - User Client 179
4.130. Admin Web App: Options - Advanced - User Name Aliases 179
4.131. Admin Web App: Options - Advanced - Reset Admin Password 180
4.132. Admin Web App: Options - Advanced - JMX Agent 180
4.133. Add JMX Connection with Java VisualVM 181
4.134. Connecting to Remote Process with JConsole 181
4.135. Admin Web App: Options - Advanced - Locale 182
4.136. Admin Web App: Options - Default Paper Size 182
4.137. Admin Web App: Options - Default Paper Size 183
4.138. Admin Web App: Options - Converters 183
4.139. Admin Web App: Options - Advanced - Proxy Printing 184
4.140. Admin Web App: Options - Advanced - Document Store 185
4.141. Admin Web App: Options - Advanced - Telegram Messaging 187
4.142. Admin Web App: Options - Advanced - Atom Feed 188
4.143. Admin Web App: Options - Advanced - Matrix Messaging 188
4.144. Admin Web App: Options - Advanced - Pagometers 188
4.145. Admin Web App: Configuration Editor - List 189
4.146. Admin Web App: Configuration Property - Edit 190
4.147. Admin Web App: Configuration Property - Edit Secret 190
4.148. Admin Web App: Document - List 191
4.149. Admin Web App: Document - IPP Routing 191
4.150. Admin Web App: Document - Proxy Print PaperCut Costs 191
4.151. Admin Web App: Documents - Print-in - PDF out 192
4.152. Admin Web App: Documents - Select and Sort - All 194
4.153. Admin Web App: Documents - Select and Sort - In 194
4.154. Admin Web App: Documents - Select and Sort - Out 194
4.155. Admin Web App: Documents - Select and Sort - PDF 195
4.156. Admin Web App: Documents - Select and Sort - Print 195
4.157. Admin Web App: Documents - Select and Sort - Ticket 196
4.158. Admin Web App: Log - List 197
4.159. Admin Web App: Log - Select and Sort 198
4.160. Admin Web App: Log - Select Date 198
4.161. Admin Web App: About 199
4.162. Admin Web App: About - Version 200
4.163. Admin Web App: About - License 200
4.164. Admin Web App: About - Privacy 200
4.165. Admin Web App: About - Community 201
4.166. Admin Web App: About - Organization 201
4.167. Admin Web App: About - Support 201
4.168. Admin Web App: About - Import Service Card 202
4.169. Admin Web App: About - Support - Service Provider 202
4.170. Admin Web App: About - Java 203
4.171. Admin Web App: About - JavaScript 203
4.172. Admin Web App: About - Environment 203
4.173. Admin Web App: About - Host Packages - 1/3 204

4.174. Admin Web App: About - PDF Standard Font Substitutes - 2/3 205
4.175. Admin Web App: About - Host Packages - 3/3 205
4.176. Admin Web App: Voucher List 206
4.177. Admin Web App: Vouchers - Select and Sort 207
4.178. Admin Web App: Voucher Actions 208
4.179. Admin Web App: Create Vouchers 209
5.1. Job Tickets: Open Tickets - Select and Sort 211
5.2. Job Tickets: Open Tickets - Group 212
5.3. Job Tickets: Open Tickets - List 213
5.4. Job Tickets: Cancel All 214
5.5. Job Tickets: Print All 214
5.6. Job Tickets: Close All 214
5.7. Job Tickets: Transactions 215
5.8. Job Tickets: Edit Ticket 216
5.9. Job Tickets: Print Ticket 217
5.10. Job Tickets: Print Pending 218
5.11. Job Tickets: Print Pending (external) 218
5.12. Job Tickets: Print Canceled 218
5.13. Job Tickets: Print Completed 219
5.14. Job Tickets: Print Completed (external) 219
5.15. Job Tickets: Settle Ticket 219
5.16. Job Tickets: Closed Ticket List 220
5.17. Job Ticket: Refund 220
5.18. Job Ticket: Refund Transactions 220
5.19. Job Ticket: Reopen Print Ticket 221
5.20. Job Ticket: Reopen Copy Ticket 221
5.21. Job Ticket: Reopened Print Ticket 222
5.22. Job Ticket: Reopened Ticket Printed 222
6.1. Mail Tickets Web App: Ticket selection 224
6.2. Mail Tickets Web App: Print Confirmation 225
7.1. Point-of-Sale: Sales 227
7.2. Point-of-Sale: Deposit Start 228
7.3. Point-of-Sale: Deposit Completed 229
7.4. Point-of-Sale: Receipts 229
8.1. Payment Web App: Main Page 232
8.2. Payment Web App: Payment 233
9.1. Print Site Web App: Menu 235
10.1. PDF/PGP Verify Link 238
10.2. PDF/PGP Verification - Upload 238
10.3. PDF/PGP Verification - Good signature 239
10.4. PDF/PGP Verification - Public key not found 239
10.5. PDF/PGP Verification - Signature not found 239
13.1. Rejected request in Firefox browser 250
14.1. SavaPage Android Print App - Printer List 265
14.2. SavaPage Android Print App - Menu 266
14.3. SavaPage Android Print Service 266
15.1. SavaPage in a Single Sign-On Domain 277
15.2. IP Based Authentication for Abstract User 278
15.3. Authentication for Unauthenticated User 279
15.4. IP Based Authentication in Peer-to-Peer Network 280
18.1. Web App - About Dialog - Software & Privacy Section 295
18.2. Web App - JavaScript Info (first 2 of 20+ sources) 295
20.1. User Web App: Custom CSS - Sample #1 302
20.2. User Web App: Custom CSS - Sample #2 302

C.1. JSON-RPC : Basic Error 344
C.2. JSON-RPC : systemStatus (request) 344
C.3. JSON-RPC : systemStatus (response) 345
C.4. JSON-RPC : authUserSource (request) 345
C.5. JSON-RPC : authUserSource (response) 346
N.1. Admin Web App: Options - PaperCut User Source 418

List of Tables

1. Typographical conventions xxiv
3.1. SavaPage Draw - Buttons (top to bottom) 39
3.2. PDF Rasterize Filter: Configuration Properties 42
3.3. Page Scaling Configuration Properties 50
3.4. Basic IPP Printer Attributes 52
3.5. Print Job Settings Configuration Properties 56
3.6. Print Job Archive Configuration Properties 56
3.7. Delegated Print Configuration Properties 63
3.8. Job Ticket Print Configuration Properties 66
3.9. Document Type Select: Configuration Property 71
3.10. User Web App Environmental Impact Configuration Properties 83
3.11. Configuration Properties for Web Print Document Types 86
3.12. Configuration Properties for User Web App GDPR 88
4.1. Admin Web App Environmental Impact Configuration Properties 97
4.2. User Roles 103
4.3. Configuration Properties for User ID number 105
4.4. Queue Icon 121
4.5. Configuration Properties for IPP Routing 125
4.6. Primary Printer Icons 129
4.7. Secondary Printer Icons 129
4.8. Configuration Properties for Printer SNMP 130
4.9. Self Registration: Configuration Properties 153
4.10. YubiKey Configuration Properties 160
4.11. 2-Step Verification Issuer Properties 161
4.12. Mail Configuration Properties 163
4.13. Mail Print Configuration Properties 166
4.14. Proxy Print Preprocessing: Configuration Properties 170
4.15. LibreOffice Configuration Properties 183
4.16. Document Store Configuration Properties 186
4.17. Telegram Messaging Configuration Properties 187
5.1. Job Ticket Print Configuration Properties 222
6.1. Mail Ticket Print Configuration Properties 225
7.1. Point-of-Sale Configuration Properties 230
10.1. PDF/PGP Signing Configuration Properties 238
10.2. PDF/PGP Verification Web App Configuration Properties 238
11.1. User Client Access Configuration Properties 240
11.2. User Client Options Configuration Properties 241
13.1. Secured Application Areas 245
13.2. Log Files 246
13.3. OpenPGP server.properties 247
13.4. OpenPGP Configuration Properties 247
13.5. Alternative TCP/IP Ports : server.properties 249
13.6. ThreadPool Settings: server.properties 250
13.7. Session Settings: server.properties 251
13.8. HTTP/2 Settings: server.properties 251
13.9. HTTP/2 Configuration Properties 251
13.10. HTTP Configuration Properties 251
13.11. Database Connection: server.properties 252
13.12. CUPS Configuration Properties 254
13.13. Server-side Download Configuration Properties 254
13.14. RESTful Client Configuration Properties 255

13.15. User Home Clean Configuration Properties 255
13.16. Database Locking Configuration Properties 256
13.17. pdf2cairo Image Strategy Configuration Property 256
13.18. Alternative File Locations: server.properties 256
13.19. Miscellaneous Settings: server.properties 257
14.1. IPP Everywhere and AirPrint 260
14.2. PostScript elements mapped to IPP options 263
14.3. Printing from GNU/Linux 263
14.4. Printing from macOS 264
14.5. Printing from Chrome OS 264
14.6. Printing from Android 264
14.7. Printing from iOS and iPadOS 267
14.8. Printing from Windows 267
14.9. Driverless PDF Transfer Processing Configuration Properties 270
14.10. PostScript Print Configuration Properties 272
17.1. Denial of Service Filter : Configuration Properties 284
17.2. Inet Access Filter : Configuration Properties 285
17.3. Web App Internet Access Configuration Properties 287
17.4. Configuration Properties for security.txt 288
17.5. Default Web Session Timeout Values 289
17.6. Web Session Timeout Configuration Properties 289
17.7. Web App WatchDog Configuration Properties 290
17.8. Rate Limiting Configuration Properties 290
20.1. Stock Image Identifiers 305
20.2. Placeholder: Stationary 307
20.3. Placeholder: Application 307
20.4. Placeholder: Ticket 308
20.5. Placeholder: User 308
20.6. Placeholder: User Registration 308
20.7. Email Stationary Types 308
20.8. Placeholder Objects: JobTicketCanceled 309
20.9. Placeholder Objects: JobTicketCompleted 309
20.10. Placeholder Objects: UserRegVerification 309
20.11. Configuration Properties for Custom Template Locations 310
A.1. Personal Print - PaperCut Scenario 322
A.2. Delegated Print - (Non) Secure & Job Ticket Scenarios 323
A.3. Delegated Print - Job Ticket - PaperCut Scenario 324
A.4. Delegated Print - PaperCut Scenario 325
C.1. XML-RPC method: onetime-auth.createToken 342
C.2. JSON-RPC Configuration Properties 343
C.3. RESTful Admin Services Configuration Properties 347
C.4. Atom Feed Configuration Properties 352
C.5. Matrix Messaging Configuration Properties 353
D.1. Database size increase metrics per document flow. 360
E.1. SavaPage URL Cheat Sheet 362
F.1. SavaPage File Locations 365
G.1. Standard Printable File Types 369
G.2. Advanced Printable File Types 371
J.1. LDAP Configuration Properties 376
J.2. OpenLDAP Default Settings 377
J.3. Apple Open Directory Default Settings 378
J.4. Novell eDirectory Default Settings 378
J.5. Microsoft Active Directory Default Settings 379
J.6. Microsoft Active Directory Custom Settings 379

L.1. Internal IPP Attribute: org.savapage-finishings-staple 397
L.2. Internal IPP Attribute: org.savapage-finishings-punch 398
L.3. Internal IPP Attribute: org.savapage-finishings-fold 398
L.4. Internal IPP Attribute: org.savapage-finishings-booklet 399
L.5. Internal IPP Attribute: org.savapage-finishings-jog-offset 399
L.6. Internal IPP Attribute: org.savapage-cover-front-type 400
L.7. Internal IPP Attribute: org.savapage-cover-back-type 400
L.8. Internal IPP Attribute: org.savapage-finishings-ext 402
L.9. Internal IPP Attribute: org.savapage-cover-type 402
L.10. Internal IPP Attribute: org.savapage-job-sheets 403
L.11. Internal IPP Marker Attribute: org.savapage.int-page-rotate180 403
M.1. Web API Callback Configuration Property 405
M.2. Payment Gateway Configuration Property 405
M.3. IPP Routing Plug-in: .properties keys 408
M.4. User Synchronization Plug-in CLI 410
M.5. User Authentication Plug-in CLI 411
N.1. SavaPage as Sync Source in PaperCut: Configuration Properties 417
N.2. PaperCut as User Source in SavaPage: Configuration Properties 418
N.3. PaperCut Personal User Account Configuration Settings 418
O.1. Cron Trigger Format - simplified 420

Preface

1. About this Manual

The SavaPage User Manual covers the setup, management and configuration of *SavaPage Open Print Portal*. Please take a few moments prior to installing the application to read the key sections of this manual. The latest version of this manual in HTML, PDF and EPUB format are available from the SavaPage website[1].

2. Expectations and Prerequisites

SavaPage is a network based application. Experience with basic network concepts, such as server administration and network connectivity is expected. Prior to installing or evaluating SavaPage you should be familiar with the concepts of:

- Basic GNU/Linux Systems Administration.
- Client/Server computing.
- Internet Printing Protocol (IPP).
- IP Printing (JetDirect/RAW)[2] .
- Common Unix Printing System (CUPS).
- Lightweight Directory Access Protocol (LDAP).

3. Conventions used in this Document

3.1. Typographical Conventions

This is a list with examples of the typographical conventions used in this manual.

Convention	Example
Executable program with options.	Enter **ls** `--reverse` to get a directory listing in reverse order.
A character or string indicating the start of an input field.	Enter you name after the `Username:` prompt.
User input.	John entered **`john`** as his login name.
A button.	Press the Cancel button.
A text prompt.	Enter your full name after the **Name** prompt.
Content that may or must be replaced by the user.	Please enter **`filename`** to save the content to.

[1] https://www.savapage.org/
[2] JetDirect is the most common Socket API, and a *de facto* standard, introduced by Hewlett-Packard. It allows a TCP/IP connection via port 9100 to a printer attached to a Local Area Network similar to a connection to its serial or parallel port. Windows supports this protocol as *Standard TCP/IP port* monitor for print server attached print devices.

Convention	Example
Filename.	Please open the file `server.properties` in your favorite editor.
Directory.	`/opt/savapage`
A question-and-answer set.	**Q:** To be, or not to be? **A:** That is the question.
Key on a keyboard	Press **F1** for help.
A combination of input actions.	Press **CTRL**+**C** to abort the program.
A selection or series of selections from a menu.	Select Print → Settings to open the dialog.
An inline code fragment.	`Text in this format` is used to show code examples, the contents of files, and console output, as well as the names of variables, commands, and other code excerpts in the text.

Code (listing).	`Line A Inline annotation on A`
	`Line B ❶ Inline annotation on B`
	`Line C ❷`
	
	❶ Comment for Line B.
	❷ Comment for Line C.

Link (external).	A link to an URI is formatted like this: https://wiki.savapage.org and mailto:support@savapage.org Or as an alternative: Visit our website[3] or write an email[4]
Link (internal).	See Chapter 2, *Server Installation* [11] of this manual.
Name of a variable.	In Perl, `@ARGV` contains the command line parameters used when the script was run.
Inline text that is some literal value.	When `debug` is activated more detailed logging is produced.

Table 1. Typographical conventions

3.2. Feature Preview

The icon above marks a section where a “feature preview” is discussed. This feature is operational yet impermanent, and available for preview to provoke feedback based on real world use. This may lead to it becoming permanent in a future release.

3.3. Notes

You should pay special attention to notes set apart from the text with the following icons:

Important

Important notes are marked like this.

[3] https://wiki.savapage.org
[4] mailto:support@savapage.org

Note

Notes provide extra background information.

Tip

Tips provide useful advice to make your life easier.

Caution

Indicate situations where you have to be careful what you are doing.

Warning

Where extreme care has to be taken.

Deprecated

Notes about a deprecated feature.

Preview

Additional notes to a Feature Preview.

4. Screenshots

To focus attention on the content, all screenshots of SavaPage Web Apps in this manual have been created without the default background image (green hills with a blue sky). See Section 20.1.1, “Web App Look-and-feel” [300].

5. Notice

While every effort has been taken to ensure the accuracy and usefulness of this manual, we cannot be held responsible for occasional inaccuracy or typographical errors. If you find an inaccuracy or error, please let us know.

Information in this document is subject to change without notice. The names of companies, products, people, characters, and data mentioned in the examples are fictitious and are in no way intended to represent any real individual, company, product, or event, unless otherwise noted.

6. Your Feedback

This manual isn't “done”. In fact, this manual will probably never be completely “done”. The subject it covers is expected to change and expand over time, and we consider this work a reflection of our ongoing conversation with the SavaPage Community[5]. Publication of this manual highlights the openness of the product, and that you, as a user, can play a pivotal role in helping to maintain and improve the product. If you see anything in this manual that can be improved: spelling, examples, explanations, then you should tell us, and send us an email[6]. Also, if you have ideas about improving the product in general, please let us know. All feedback will be rewarded with a gracious response.

[5] https://www.savapage.org/
[6] mailto:support@savapage.org

Chapter 1. Introduction

1.1. What is SavaPage?

SavaPage is an Open Print Portal that uses Open Standards and Commodity Hardware for Secure Pull-Printing, Pay-Per-Print, Delegated Print, Job Ticketing, Auditing and PDF Creation.

SavaPage is implemented as Print Server on GNU/Linux. Any workstation or device that supports Internet Printing Protocol (IPP), IP Printing (JetDirect), or AirPrint® [1]can print to SavaPage. This includes workstations running GNU/Linux, macOS or Windows, as well as devices running Chrome OS, Android, iOS and iPadOS. Moreover, a Driverless File Transfer Print alternative is available via Web Print and Mail Print.

Printed pages are shown in the SavaPage Web Application. Print jobs are accumulated in a single personal preview, where they can be manipulated and pruned, before storing or routing them as PDF document.

In the Web App, documents are routed to a “real” printer, optionally via intermediate Job Tickets, with Common Printing Dialogs of server-side installed printers (*proxy printers*). This makes SavaPage the Central Print Portal where documents to be printed are acquired and routed.

Pay-per-Print functions charge printing costs to individuals, groups, or shared accounts. With Delegated Print authorized users can print on behalf of other users.

SavaPage Web App is optimized for desktop as well as mobile browsers. This opens up many useful scenario's. For instance, a user can walk up to a printer and send a print job on the spot, by pushing a button on their smartphone. With a special Web App, administrators on the go can easily monitor the system on their tablet.

SavaPage turns printing into a unified experience, abstracted from platform specifics. It is the logical stopover where users are guided through sustainable print scenario's that help reduce overall printing costs.

1.1.1. Open Source Software

SavaPage is *OSI Certified Open Source Software* [2], licensed under the GNU Affero General Public License (AGPL)[3] as published by the Free Software Foundation[4], either version 3 of the License, or (at your option) any later version. All JavaScript code run in SavaPage Web Apps is GNU LibreJS compliant.

1.1.2. Benefits

The key benefits of SavaPage are:

- *Less administration*. SavaPage is the one printer you need to print to any printer in your organization.
- *Zeroconf printing*. All you need is AirPrint or a simple file upload to print from any workstation or device, and a Web Browser for Print Preview.
- *Zero-install WebTop*. Printers in your organization are sandboxed in a Web Desktop and available for advanced Print Scenarios on previewed SavaPage print jobs.

[1] *AirPrint* is a registered trademark of Apple Inc.
[2] OSI Certified is a certification mark of the Open Source Initiative [https://opensource.org/].
[3] https://www.gnu.org/licenses/agpl.html
[4] https://www.fsf.org/

- *Easy follow-me printing*. Several hold-release scenarios, optionally with NFC cards, are supported. Users can even use their own mobile device as print release terminal.
- *Think before you print*. SavaPage Print Preview makes you think twice. Do you really need to print all these pages?
- *Eco-friendly*. Create environmental awareness by drawing end-user attention to the cost of printing, and save precious paper, trees and money along the way.
- *Reduction of printing costs*. Remove unnecessary pages and graphics. Save as PDF, or route to a "real" printer with n-up, gray-scale and duplex to reduce printing costs.
- *No pre-printed paper needed*. Eliminate the cost of pre-printed paper. Create virtual letterheads and apply them to any print job.
- *Advanced Print Services* with Job Ticketing and Delegated Print.
- *Open Source Software* and *Open Standards* above Proprietary Software.
- *Commodity Hardware* above expensive Proprietary Devices.
- *Peer-To-Peer Cooperation* above Centralized Corporation. The SavaPage Community is there to help.

1.1.3. Key Features

The key features of SavaPage are:

- One SavaPage Printer
  - IPP Everywhere and AirPrint support on Local Network.
  - Secure Internet Print and Android Print Service.
  - Driverless File Transfer Print via Web Print and Mail Print.
  - Advanced IPP Queues for various trust models and IPP Routing.
- Follow-me Printing
  - Release Terminals
  - NFC Authentication
- Web Apps for Desktops *and* Mobile Devices
  - Easy authentication
    - Username/Password, ID/PIN and NFC Card authentication.
    - LDAP (Active Directory) Integration.
    - Raspberry Pi Network Card Reader.
    - 2-Step Verification (2FA)
  - User Web App
    - Real-time print preview with Browse, Sort and Delete options.
    - Server-side Proxy Printing (no local drivers needed).
    - PDF Download or Email of accumulated print jobs.
    - Multi-page Letterheads.
    - Option to remove graphics from PDF and proxy print output.
    - Innovative Eco Print to reduce ink and toner cost.
    - Delegated Print for delegates to proxy print for other users and groups.
    - Job Ticket Print for voluminous proxy print jobs.
  - Admin Web App
    - Comprehensive Web App to configure the SavaPage environment.
  - Multi-language support.
  - Customizable Web Interface.
  - SSL Encryption.
- SavaPage Financial

  - Pay-per-Print
  - Vouchers
  - Point-of-Sale Web App
  - Online Payments (credit cards, bank accounts).
- Tooling and Tuning
  - Command-Line Interface to server methods.
  - Web Services API.
  - Third party Database support.
- Third Party Integration
  - LDAP (OpenLDAP, FreeIPA, Microsoft Active Directory, Google Cloud Directory ...)
  - Single-Sign-On: OAuth, Moodle[5].
  - Third Party Print Management Systems (PaperCut).
- Comprehensive Documentation
  - User Manual in PDF, EPUB and HTML format.
- GDPR Compliant.

1.2. System Requirements

1.2.1. Server

SavaPage Open Print Portal can be installed on any modern GNU/Linux system that supports **systemd** service manager like distributions based on Debian and Red Hat Enterprise Linux (RHEL), and openSUSE. Debian based distributions that use **SysV** init scripts are also supported.

Note

Throughout this manual GNU/Linux command and file examples are given for Debian based systems. Commands and files might differ for other distributions, but not in function. For example, the Debian **apt** command has a RHEL **yum** and openSUSE **yast** equivalent. It is trusted that system administrators can translate the examples to their own environment. If applicable, functional differences between distributions will be explained.

Installed host package versions are shown in the Admin Web App.

1.2.1.1. Java

SavaPage is a Java program and must have JDK 11 or higher installed. Check the installation by executing both the **java** and **javac** commands: they should echo usage information.

On Debian based systems you can install the JDK package with this command:

```
sudo apt install default-jdk-headless
```

1.2.1.2. CUPS

SavaPage uses local CUPS printer queues for Proxy Printing. CUPS 1.4 or higher must be installed [6].

On Debian based systems you can install CUPS with these commands:

[5] https://moodle.org/plugins/block_savapage
[6] CUPS 1.4 or higher provides the Job Template attribute “fit-to-page” that is used by SavaPage proxy printing to scale documents to fit the size of selected media. See https://www.cups.org/doc/options.html. Also see Section K.1.1.7, “print-scaling” [384].

```
sudo apt install cups       ❶
sudo apt install cups-bsd   ❷
```

❶ This package provides the parts of CUPS which are needed for using printer drivers.
❷ This package provides the BSD commands for interacting with CUPS, like **lpr**.

Important

SavaPage will automatically add any local CUPS printer as proxy printer. So, for proxy printing to work, first add each proxy printer as CUPS printer.

Tip

Modern GNU/Linux distributions have everything prepared for using most printers. For USB printers it is often enough to simply plug them in. For network printers you simply start the distribution's printer setup tool out of the system administration menu or out of a system administration application, click on a button for adding a new printer and then follow the screen instructions. If this does not work, usually there is no suitable driver installed on your system. Verify in the OpenPrinting database[7] whether your printer is supposed to work and whether there is a driver or PPD file available. See the OpenPrinting CUPS Quick Start[8] for more details.

1.2.1.3. SMTP

An SMTP host must be available for sending email messages. See Section 4.11.5, “Mail” [162].

1.2.1.4. Database

SavaPage is packaged with an internal database that offers you the opportunity to evaluate the product on a small scale right away. However, when promoting SavaPage to a production environment with multiple users, we strongly advise you to use PostgreSQL[9] as external database server. See Chapter 21, *Using an External Database* [311].

Warning

Using the internal database in situations with concurrent use will inevitably lead to locking, deadlock and out-of-memory errors, which can make the system totally unresponsive. See our notes on Database Locking.

1.2.1.5. Poppler

SavaPage needs the PDF utilities based on Poppler to function properly.

Poppler[10] is a PDF rendering library based on Xpdf PDF viewer. The command line utilities are used to get information of PDF documents, convert them to other formats, or manipulate them. SavaPage uses **pdftocairo** to convert PDF pages to images and to Repair PDF documents.

Check if the package is installed by entering the following command:

```
pdftocairo -v
```

On Debian based systems you can install the package with this command:

[7] https://www.openprinting.org/printers
[8] https://www.linuxfoundation.org/collaborate/workgroups/openprinting/database/cupsdocumentation
[9] https://www.postgresql.org/
[10] https://poppler.freedesktop.org/

```
sudo apt install poppler-utils
```

> **Note**
>
> See Section 13.5.12, “pdf2cairo Image Strategy” [256] if errors are encountered when creating SafePage images.

1.2.1.6. QPDF

QPDF[11] is a command-line program that does structural, content-preserving transformations on PDF files. The package is used to decrypt PDF from Web Print and Mail Print when Encrypted PDF for Proxy Printing is allowed. See Section 4.11.14.9, “SafePages” [184].

Check if the package is installed by entering the following command:

```
qpdf --version
```

On Debian based systems you can install the package with this command:

```
sudo apt install qpdf
```

1.2.1.7. rsvg-convert

rsvg-convert[12] is a command-line program that converts SVG images into PNG raster images, PDF, PS, or SVG vector images, or dumps of Cairo XML or recording surfaces. The program is used by SavaPage Draw and needed to handle Standard File Type SVG.

Check if the package is installed by entering the following command:

```
rsvg-convert --version
```

On Debian based systems you can install the package with this command:

```
sudo apt install librsvg2-bin
```

1.2.1.8. heif-convert

heif-convert is a command-line program that converts HEIF images to JPEG. Installation is *optional*.

Check if the package is installed by entering the following command:

```
heif-convert --help
```

On Debian based systems you can install the package with this command:

```
sudo apt install libheif-examples
```

1.2.1.9. ImageMagick

SavaPage needs the **convert** command of the ImageMagick software suite to manipulate images.

ImageMagick is a software suite to create, edit, compose and convert bitmap images in a variety of formats (over 100) .

Check by entering the following command:

[11] https://sourceforge.net/projects/qpdf/
[12] https://www.systutorials.com/docs/linux/man/1-rsvg-convert/

```
convert --version
```

On Debian based systems you can install the package with this command:

```
sudo apt install imagemagick
```

1.2.1.10. Avahi

Avahi is needed if you want to use IPP Everywhere and AirPrint to print to SavaPage. It is also needed to retrieve the hostname of CUPS printers whose URI refers to a DNS-SD instance (`dnssd://` scheme) in order to establish an SNMP connection. See Section 4.8.1.1, “Printer SNMP” [130].

Avahi[13] is a system which facilitates service discovery on a local network via the mDNS/DNS-SD [14] protocol suite. Any modern GNU/Linux system has Avahi installed, but to be sure you can check by entering the following command:

```
avahi-browse --version
```

On Debian based systems you can install the package with this command:

```
sudo apt install avahi-utils avahi-discover libnss-mdns
```

1.2.1.11. Hardware

The SavaPage server process requires a minimum of 2 CPU cores, 2GB of RAM and 1 GB of free disk space.

Note

Depending on the expected print quota you should reserve extra disk-space for each SavaPage user. See Appendix D, *Capacity Planning* [360].

1.2.2. Clients

On *desktop and mobile* clients you need an HTML5 compatible browser and support for Internet Printing Protocol (IPP), IP Printing (JetDirect) or AirPrint® .

1.3. How does SavaPage work?

To explain how SavaPage works we first introduce the key concepts for the usage scenarios. After that we will describe a typical work flow, and end with a high-level overview of the application's architecture.

1.3.1. Key Concepts

Each concept is described in an abstract definition with its SavaPage implementation.

1.3.1.1. Print Server

A print server is a system responsible for hosting print queues and sharing printer resources to client workstations. Client users submit print jobs to a print server rather then directly to the printer itself. A print server may be a dedicated server. However, on many networks this server may also perform other tasks such as file serving.

[13] https://avahi.org/

[14] mDNS/DNS-SD enables you to plug your laptop or computer into a network and instantly be able to view other people who you can chat with, find printers to print to or find files being shared. Compatible technology is found in Apple macOS (branded Bonjour [https://www.apple.com/support/bonjour/] and sometimes Zeroconf).

SavaPage is a regular Print Server in the technical sense, but is special in the sense that it shares multiple print queues of the one SavaPage virtual printer. The GNU/Linux host where SavaPage is deployed on may offer file services on its own account.

1.3.1.2. Print Queue

A print queue is first-in-first-out queue holding all jobs pending on a given printer.

SavaPage virtual queues redirect print jobs to the originating user's personal queue called SafePages. The SavaPage Web App is the viewport on these SafePages.

1.3.1.3. User ID/Username

In a multi-user environment, users login to a network or computer using a username and password. Often these are managed by services like Active Directory or LDAP. The username is known as the user's identity.

SavaPage uses this identity for authentication and auditing purposes.

Note

User authentication is a topic of its own. Please see Chapter 15, *Authenticated Printing* [274] for more elaboration on the *User* concept.

1.3.1.4. Client/Server Model

Client software is a small application that runs on each workstation and communicates with a central server. The printing process on most networks works according to a client/server model with clients (workstations) submitting jobs to a server.

SavaPage utilizes the client/server model with standard components on the workstation, i.e. an IPP or JetDirect printing client and a Web browser.

1.3.1.5. Application Server

An application server is a server program responsible for centrally processing “business logic” and providing services to end-users.

SavaPage is an application server since it provides “business logic” for showing, editing and routing printed documents.

1.3.1.6. Information Provider

A provider is a software component or program responsible for providing information to an Application Server.

SavaPage uses an integrated IPP and JetDirect Server to capture Driver Print jobs from client workstations and devices. It communicates with IMAP to capture Mail Print jobs and uses HTTP upload to capture Web Print jobs. The generic information provider for capturing print jobs is called the “Print Provider”. Other important providers are “User Directory Provider”, “Authentication Provider” and “CUPS Information Provider”.

1.3.1.7. Web Application Interface

A Web Application, or Web App for short, is a software program that interacts with end-users via a web browser. A Web App gives flexibility because it allows access from any location on the network and avoid the need for installation of separate software.

SavaPage provides a web-based interface for end-users and system administrators. Since it is optimized for desktops and mobile devices an even greater flexibility is achieved.

1.3.1.8. SafePages

SafePages is the SavaPage term for the personal user space with accumulated jobs from SavaPage printer queues. See Section 1.3.1.2, “Print Queue” [7].

1.3.1.9. Proxy Printer

Proxy Printer, or *ProxyPrinter*, is the SavaPage term for a printer that is available in the SavaPage Web App for printing selected SafePages.

Important

It is important to understand that using a Proxy Printer does *not* require its printer driver on the client workstation. Proxy Printer queues are CUPS queues located on the GNU/Linux SavaPage host and are *not* shared on the local network, hence *not* visible for client workstations. Proxy Printer queues can only be selected and used in the SavaPage Web App sandbox for pass-through printing.

1.3.2. The SavaPage Work Flow

To illustrate what SavaPage is about and how it works we'll start with a simple use case.

Note

Advanced user scenario's are described in Appendix A, *Proxy Print Scenarios* [321].

1.3.2.1. End-user perspective

1. John opens a web browser, clicks on the SavaPage bookmark and logs into SavaPage with his regular Active Directory credentials.
2. John prints a document from his favourite editor to his *SavaPage Network Printer*.
3. John sees the printed pages appear as thumbnails in his web browser.
4. John browses through the thumbnails and zooms in on page 15 and 16 to see more detail.
5. Things look good, apart from two void pages at the end. So, John deletes these pages using the Delete dialog.
6. After selecting the company letterhead as standard background, John selects the Brand-X Multi-functional Proxy Printer located down the hall, checks the settings (duplex and grayscale), and presses the Print button.
7. Since John also wants to save a PDF document of the result, he sets the PDF properties (title, author, subject, keywords, encryption) and presses the Download button.

Note

John could also have opened a web browser on his smartphone and do exactly the same things.

1.3.2.2. Technical perspective

This is what happens behind the scenes.

1. When John prints to SavaPage from his editor, his workstation transfers the print job to the SavaPage Print Server.
2. The SavaPage Print Provider handles the print job, analyzes the information and retrieves:
   a. The identity of the *user* who printed the document.
   b. The identity of the *queue* the job was printed to.
3. The Print Provider submits the information about the job to the Application Server to process the business logic.
4. The Application Server approves the print request, transfers the job to the user's SafePages, and signals John's browser session that a new job has arrived.
5. The Web Application in John's browser picks up the signal, handles the information and displays the newly printed pages.
6. The Web Application transfers each of John's editing actions (delete, letterhead) to the Application Server where the *state* of the SafePages is saved.
7. When Johns selects the Brand-X Multi-functional Proxy Printer, the Web App asks the Application Server for the printer options, so it can display the Printer Settings dialog for this specific printer.
8. When Johns presses the Print button, the print action plus the selected printer options are passed to the Application Server. The server composes the print job (applying editing actions and selected letterhead) and sends the result to the Proxy Printer, using the printer options John selected.
9. John's download request is fulfilled by the Application Server with a PDF document holding the edited SafePages, including the letterhead, and the chosen PDF settings.

1.3.3. Architecture Overview

Figure 1.1, “SavaPage High-Level Architecture” [9] shows a high level view of the components and communication involved.

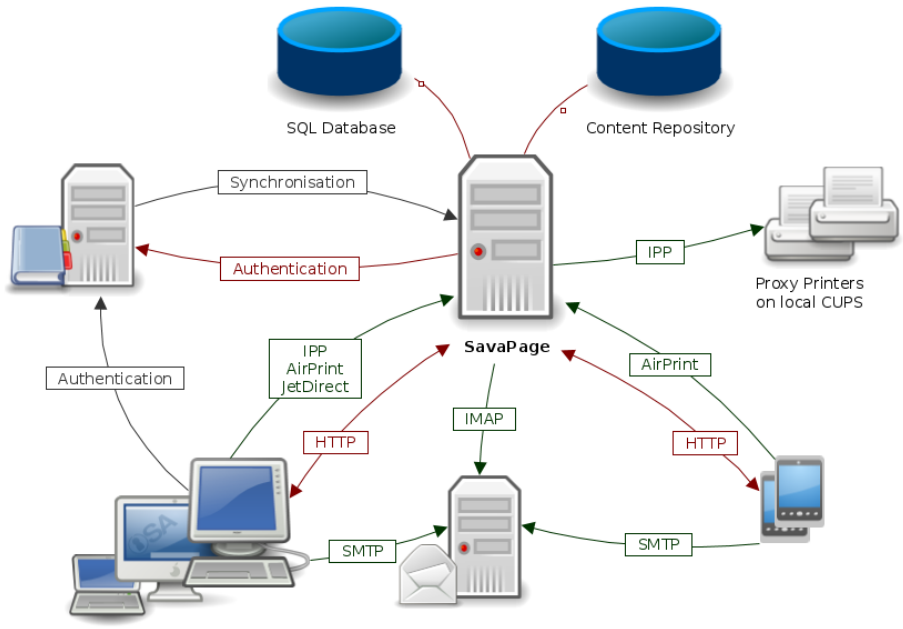


Figure 1.1. SavaPage High-Level Architecture

- The SavaPage Print Server synchronizes users from the LDAP/NIS source to its own SQL Database.

- The Client Web Application on desktops, laptops and mobile devices communicates with the Application Server using HTTP.
- Desktop and laptops users can be forced by their OS to login and be authenticated at the LDAP/NIS source.
- SavaPage Web App users on desktops, laptops and mobile devices are authenticated by the SavaPage Print Server at the LDAP/NIS source.
- Desktop and laptop users print to SavaPage with the SavaPage printer driver using IPP or JetDirect protocol.
- macOS and iOS users can print to SavaPage with AirPrint®.
- Every user can use SMTP to Mail Print to SavaPage.
- SavaPage uses IMAP to monitor the Web Print Inbox.
- Every user can use HTTP to Web Print to SavaPage.
- The Content Repository holds letterhead documents.
- A print command in the Web App to a Proxy Printer is executed by SavaPage with an IPP operation to local `CUPS`.

Chapter 2. Server Installation

This chapter covers the *initial* installation and configuration of SavaPage in your network environment.

- If you are installing a new version over an existing installation please consult Appendix H, *Upgrading from a Previous Version* [373].
- If this installation is part of a migration from an old server please consult Appendix I, *Migrating to a New Server* [374] before going on.

Initial installation takes only a few minutes on a prepared server. This guide will walk you through installation and configuration step-by-step. The process is summarized below:

1. System requirements check.
2. Downloading and installing SavaPage.
3. Completing the configuration.
4. Testing the software.

Tip

If you would like to know the technical details behind the SavaPage installer, take a look at Section 13.1, "The Installation Process" [244].

Important

By installing the program, you are accepting and agreeing to the terms of the GNU Affero General Public License (AGPL). Please review Appendix P, *GNU Affero General Public License (AGPL)* [422] before continuing.

2.1. Step 1 - System Requirements

Before proceeding with the installation you should take a few moments to verify system requirements. Is the operating system version supported and are patches up-to-date? Take a few minutes to verify the system is current and supported (see Section 1.2, "System Requirements" [3]).

The SavaPage installation program needs the commands **which**, **strings**, **gunzip** and **perl**. So, make sure the **binutils**, **debianutils** (for Debian based systems), **perl** and **gzip** packages are installed.

2.2. Step 2 - Create System Account

SavaPage runs and installs under a system user account called `savapage`. This account is fixed, you cannot choose another name. You are free though to pick a location for the application. However, GNU/Linux Filesystem Hierarchy Standard (FHS) dictates that the application is installed in the `/opt/savapage` directory.

Create the system user account at the command prompt by entering:

```
sudo useradd -r savapage
```

The syntax for `useradd` may differ slightly on different versions of GNU/Linux. It may also be called `adduser`.

Next, create the install directory and set the ownership to the `savapage` account:

```
sudo mkdir -p /opt/savapage
```

... and set the ownership to the savapage account. For Debian and RHEL based systems:

```
sudo chown savapage:savapage /opt/savapage
```

... and for openSUSE:

```
sudo chown savapage:users /opt/savapage
```

For convenience, set the login shell and login directory:

```
sudo usermod -s /bin/bash savapage
sudo usermod -d /opt/savapage savapage
```

Some GNU/Linux distributions impose strict resource usage limits on user accounts (`ulimit`). The `savapage` account is a dedicated account used for hosting the SavaPage application and hence should be granted sufficient resource limits such as the ability to open many files. Please consult Section 22.2, “Linux User Limits” [316] on how to change these limits.

2.3. Step 3 - Configure CUPS and Samba

Make sure to *not* publish shared printers in CUPS *and* Samba. Publishing shared printers creates a loophole by which users can access a printer directly from their workstation and print outside the control of SavaPage.

For Samba, just edit the `/etc/samba/smb.conf` file and disable the `[printers]` share definition.

In CUPS, do *not* enable the “Publish shared printers connected to this system” option as offered in the Print Server Settings dialog. When no such dialog is available you can make the adaption in the CUPS Administrator Web interface (“Share printers connected to this system”), or manually in the `cupsd.conf` [1]file.

Important

Before editing `cupsd.conf` first stop CUPS by entering this command:

```
sudo systemctl stop cups.service
```

Check if `cupsd.conf` contains the snippets below that restrict CUPS to `localhost` only. If not, change the configuration file accordingly and leave all other content as it is.

```
# Only listen for connections from the local machine.
Listen localhost:631

# Disable printer sharing.
Browsing Off

# Restrict access from localhost only
<Location />
  Order allow,deny
</Location>
```

[1] On Debian, RHEL and openSUSE systems `cupsd.conf` is located in the `/etc/cups/` directory.

> **Important**
>
> Each individual proxy candidate CUPS printer must be *shared* locally so the `savapage` system account can access it. Enabling the shared option can be done in a printer GUI dialog, in the CUPS Administrator Web interface, or directly in the `printers.conf` [2] file by setting the `Shared Yes` option for a printer.

2.3.1. CUPS Remote Printer Browsing

To prevent remote printers that DNSSD broadcast themselves, to be discovered by CUPS and synchronized into SavaPage, edit `cups-browsed.conf` [3]file, and change it as follows:

```
BrowseRemoteProtocols none
```

2.3.2. CUPS Job History

An active SavaPage server captures print job statuses real-time, but when the server is restarted it needs CUPS job history to catch up with the latest statuses. To avoid lost job statuses, CUPS must be told to “Preserve job history”.

You can set the Job History option in the Print Server Settings dialog (“Preserve job history but not files”, or optionally “Preserve job history (allow reprinting)”), in the CUPS Administrator Web Interface (Advanced settings, “Retain Metadata : Yes”, and optionally “Retain Documents : Yes”) , or manually by changing the `cupsd.conf` file as follows:

```
MaxJobs 500                ❶
PreserveJobHistory Yes     ❷
PreserveJobFiles No        ❸
```

❶ `MaxJobs` Controls the maximum number of jobs that are kept in memory. Once the number of jobs reaches the limit, the oldest completed job is automatically purged from the system to make room for the new one. If all of the known jobs are still pending or active then the new job will be rejected. The default setting is 500 (in this example this default is explicitly set). Set to `0` to allow an unlimited number of jobs.

❷ `PreserveJobHistory` specifies whether *metadata* is preserved after a job is printed. A value of `Yes` will preserve history, a value of `No` will not. If a numeric value is specified, history is preserved for the indicated number of seconds after printing. Set to **`Yes`**.

❸ `PreserveJobFiles` specifies whether job *files* (documents) are preserved after printing. A value of `Yes` will preserve files, a value of `No` will not. If a numeric value is specified, files are preserved for the indicated number of seconds after printing. Set this option as you wish, but remember that (spool) files can get big. If you run SavaPage on a host system with limited storage (for instance on a virtual machine) you better set this value to **`No`**.

The number of preserved history jobs can be checked with this command:

```
sudo find /var/spool/cups/ -maxdepth 1 -type f -name 'c*' | wc -l
```

The number of preserved job files can be checked with:

```
sudo find /var/spool/cups/ -maxdepth 1 -type f -name 'p*' | wc -l
```

2.3.3. CUPS Job ID

This section gives you some background information about the ID that CUPS assigns to a print job[4]. There is nothing you need to configure for this, and you don't really need to consider this info at a first time SavaPage installation. However, do read carefully when Migrating to a New Server, or if you plan to meddle with CUPS cache.

[2] On Debian, RHEL and openSUSE systems `printers.conf` is located in the `/etc/cups/` directory.
[3] On Debian, RHEL and openSUSE systems `cups-browsed.conf` is located in the `/etc/cups/` directory.
[4] CUPS `NextJobId` is held in `/var/cache/cups/job.cache`

The CUPS Job ID is a global integer that is incremented and assigned to each new print job. In this way, a Job ID refers to a single job within the CUPS cache. When a document is printed from SavaPage, its Job ID is persisted in the SavaPage database and used to retrieve CUPS job status information, or to link incoming CUPS status notifications to the printed document.

When you delete the content of the CUPS cache directory `/var/cache/cups/`, the Job ID offset is reset to zero. If SavaPage depends on this CUPS instance, the cache will be out-of-sync with the CUPS Job ID range as persisted in the SavaPage database. This will most likely result in newly issued job ID's that are already present in the SavaPage database, and so introduce a non-unique relation between printed document and CUPS job ID. Also, when SavaPage is moved to another server an out-of-sync is very likely.

How does SavaPage deal with the possibility of an out-of-sync situation?

1. For incoming CUPS notifications, the most recent document that matches the provided Job ID, and has not yet reached final state (completed, cancelled, aborted), is updated with the provided status.
2. When retrieval of Job ID status information for a document with non-final state from CUPS fails, then print status is set to “unknown” and time completed to system time.

2.3.3.1. CUPS Job ID Duplicates

During intensive proxy printing, a `MaxJobs` value that is too low (even the default value of `500`) may be reached during peak times. This can happen in a Job Ticket scenario where different departments send tickets simultaneously, each for a specific Job Ticket Group. In this case, we have found that sometimes the same Job ID was issued for different print jobs.

Although these jobs with duplicate IDs reach the printer and are printed successfully, the completed status is only updated in SavaPage for the most recent document that matches the specified Job ID. As explained above, older documents with the same Job ID are not updated and retain their **Active** (“pending”) or **Unknown** status. See Figure 4.156, “Admin Web App: Documents - Select and Sort - Print” [195]

Important

If you observe this behavior, first set `MaxJobs` to a much higher value to prevent future occurrences.

Second, in the Admin Web App, as administrator with permission to edit accounts, you can manually simulate a Cancel or Complete for Printed documents with duplicate Job IDs that have the status **Active** or **Unknown**. For this you must set Configuration property `proxy-print.job-state.edit.enable = Y`. Set the property back to `N` after you have made the corrections. See Section 4.12.3, “Proxy Print Status” [196].

2.3.4. CUPS Job Privacy

CUPS makes “job-name”, “job-originating-host-name” and “job-originating-user-name” private by default. This means that personal data are anonymized in the CUPS Web interface, as shown below.

▼ ID ▼	Name	User	Size	Pages	State
Xerox-D125-8586	Unknown	Withheld	1k	1	completed at Mon 30 May 2016 10:47:27 AM CEST

Figure 2.1. CUPS Job Privacy

You can restore the default by changing the `JobPrivateValues` directive in the `cupsd.conf` file as follows:

```
<Policy default>
  JobPrivateValues default
```

2.3.5. CUPS Web Interface

If you want to use the CUPS Web Interface for administration from all machines on the local network you should adapt `cupsd.conf` as follows:

```
# Allow remote access
Port 631

# Disable printer sharing.
Browsing Off

WebInterface Yes

<Location />
  # Allow shared printing...
  Order allow,deny
  Allow @LOCAL
</Location>

<Location /admin>
  Order allow,deny
  Allow @LOCAL
</Location>

<Location /admin/conf>
  AuthType Default
  Require user @SYSTEM
  Order allow,deny
  Allow @LOCAL
</Location>
```

Note

The `@LOCAL` name will allow access from all *local* interfaces. This can be confusing since the host system running CUPS often belongs to a different set of subnets from its clients. In that case you can specify multiple `Allow` directives, one for each allowed subnet. See the cupsd.conf man page[5].

2.3.6. CUPS security fix

CUPS versions before 2.4.3 contain a security issue that allows users to retrieve documents (pending or printed) without authentication [6]. If this CUPS version is installed, modify `cupsd.conf` as shown below.

```
...
<Policy default>
  ...
  # Remove the CUPS-Get-Document operation from <Limit> tag ...
  <Limit Send-Document ... CUPS-Get-Document >
    Require user @OWNER @SYSTEM
    Order deny,allow
  </Limit>
  # ... and insert <Limit> section below
  <Limit CUPS-Get-Document>
    AuthType Default
    Require user @OWNER @SYSTEM
    Order deny,allow
  </Limit>
```

[5] https://www.cups.org/doc/man-cupsd.conf.html

[6] CVE-2023-32360 [https://nvd.nist.gov/vuln/detail/CVE-2023-32360] [An unauthenticated user may be able to access recently printed documents] was reported by Gerhard Muth and explained in personal communication with the author of this manual. Also see About the security content of macOS Ventura 13.4 [https://support.apple.com/en-us/HT213758] and RHSA-2023:4771 - Security Advisory [https://access.redhat.com/errata/RHSA-2023:4771].

```
  ...
</Policy>
...
```

2.3.7. Test CUPS

When CUPS was stopped earlier you need to start CUPS again with this command:

```
sudo systemctl start cups.service
```

Now you can test if the CUPS print queues to be used as Proxy Printer work as expected.

2.4. Step 4 - Optional System Settings

2.4.1. Set Default Paper Size

You can optionally set the default system paper size in the file `/etc/papersize`. This default is used by SavaPage, but can again be overridden in the Admin Web App. See Section 4.11.14.6, “Default Paper Size” [182].

The format of the `/etc/papersize` file is very simple: whitespace and anything starting with “#” is ignored, and the name of the paper is the first string found; the case in the name of the paper is irrelevant. Commonly valid paper size values are: `a3`, `a4`, `a5`, `b5`, `letter`, `legal`, `executive`, `note` and `11x17`.

For example, use this command to set the default to `a4`:

```
sudo su -c 'echo a4 > /etc/papersize'
```

2.5. Step 5 - Check Firewall Settings

SavaPage uses TCP/IP port `8631` (for HTTP), port `8632` and `8633` (for HTTPS/SSL) and port `9100` (for JetDirect/RAW printing) by default.

> **Note**
>
> You can change the TCP/IP interface and port defaults in the `server.properties` file after installation. See Section 13.5.3.1, “Alternative TCP/IP Interface and Ports” [249].

For Proxy Printer access the standard IPP port `631` of local CUPS is used.

For IPP Everywhere and AirPrint UDP port `5353` is used.

Depending on the Mail settings common SMTP ports are `25`, `465` and `587`.

Common IMAP ports used for Mail Print are `143` and `993`.

The Secure JMX Connection uses port `8639`.

Many GNU/Linux distributions have strict default firewall policies, so take some time now to ensure that these ports are open. Consult your distribution documentation for details on how to open firewall TCP and UDP ports.

2.6. Step 6 - Download and Install

SavaPage is supplied as a self-extracting and self-installing archive for 64-bit (`x64` , `x86_64`, `amd64`) systems. The installation must be performed as the newly created `savapage` user in the `/opt/savapage` directory. Some parts of the installation need to be executed as `root`. When you choose to do so during the main install, please have the

`root` password or `sudo` password handy. You can also choose to execute the `root` tasks *after* the main install. In that case you can simply `sudo` execute a post-install script. For more detail about the install process please see Chapter 13, *SavaPage on GNU/Linux* [244].

Change to the newly created `savapage` user, download and execute the installer in `/opt/savapage`, and follow the instructions.

```
sudo su savapage
cd /opt/savapage
rm ./savapage-setup-*-linux-*.bin  ❶
wget URL  ❷
sh ./savapage-setup-*-linux-*.bin  ❸
```

❶ Remove old downloads, if any, first, to prevent that a new download gets `.bin.1` suffix.
❷ Use the *`URL`* from the SavaPage Wiki[7] download section.
❸ For example `savapage-setup-1.7.0-linux-x64.bin`

The installation process will take between one and two minutes depending on the speed of the system. A system restart is not required. When installing on a *live* production system, administrators are advised to choose a period of low activity - for example, not during backup operations or other administration activities.

2.7. Step 7 - Save Encryption Keys

SavaPage creates the `/opt/savapage/server/data/encryption.properties` file at first installation. The encryption keys held in this file are used to store Encrypted Secrets and Document Signatures in the database.

Make a backup of this file now, and store it at a secure place, so you can restore it when you need to migrate to a new server.

Warning

The `encryption.properties` file is *crucial* for decrypting secret data in the database and verifying the authenticity of document signatures. When you lose this file you won't be able to use any database copy which was based on its encryption keys **ever**.

2.8. Step 8 - Configure

After installation, you will be prompted to open a web browser at `https://savapage:8632/admin` to complete the configuration. The configuration steps are explained below.

2.8.1. Step 1 - Login

Login with username `admin`. This is the built-in administration account. Enter `admin` as password. This is the standard password as set by the installer.

After login the Dashboard is shown where you will notice the system status “Setup is needed”. The next steps guide you in configuring the system so that the status will change to “Ready to use”.

Note

As long as system setup is needed login attempts at the User Web App are blocked with a message saying “Application setup is required”.

[7] https://wiki.savapage.org

2.8.2. Step 2 - Change Admin Password

As a first security measure change the master password for the built-in `admin` account. This account is independent and not related to the operating system or domain. The password needs to meet minimum strength requirements, and must contain at least six characters. Select Options → Advanced → Reset internal admin password, enter and confirm the new password and press the Apply button.

Caution

Make sure that this password is kept at a secure place since it is the key to your system.

More information about the admin password can be found in Section 17.2.1.3, “Internal Admin Password” [286].

2.8.3. Step 3 - Set Locale

Set the system's locale; ensure that these are correct before proceeding. Select Options → Advanced → Locale, and enter the locale string. Some examples are: `en`, `en-GB`, `en-US`, `nl`, `nl-NL`, `nl-BE`. You can leave the locale empty to accept the system default. The locale is applied to all system messages which are logged in the system log or send by email. See Section 19.1, “Localization” [297].

2.8.4. Step 4 - Set Currency Code

Set the system's currency code; ensure that these are correct before proceeding. Select Options → Financial, and enter the ISO 4217 Currency code. Some examples are: `USD`, `EUR`, `GBP`. The currency *symbol* is determined in the context of the user or system locale. See Section 19.1, “Localization” [297].

2.8.5. Step 5 - Set User Source

SavaPage optionally imports user information from a **Unix** (PAM, NIS, etc.) or **LDAP** source.

Select Options → User Source.

Select **Unix** if the user accounts are setup and defined on the local system as standard Unix accounts or mapped into the system from a central directory service such as LDAP via `nsswitch.conf` and PAM. Most large established networks will use this option.

Note

For administrators wishing to customize the PAM authentication method at the application level, SavaPage reports itself as “savapage”.

The **LDAP** option is appropriate for large networks with existing LDAP domains. This includes networks running OpenLDAP, FreeIPA, Apple Open Directory, Novell eDirectory, Microsoft Active Directory and Google Cloud Directory.

More information on LDAP is available in Section 4.11.1.2, “LDAP” [150].

After selecting the source, enter the necessary parameters (LDAP only) and press the Apply button.

2.8.6. Step 6 - User Synchronization

Skip this step if you did not set an external User Source in the previous step. Otherwise, select Options → User Creation → Synchronization → Synchronize now to import users.

Important

An option exists to import a subset of users from the source by selecting a *group*. This option is relevant if only a subset of users will ever use SavaPage. Select Options → User Creation → Change Group to select the group.

Tip

Test the import first by pushing the Test button. A simulated import will start, with each step echoed below the button, so you can verify the effect of your action.

2.8.7. Step 7 - Set Mail Options

Select Options → Mail, enter the SMTP and Message options and press the Apply button. Data from the Messages section is used for system generated mail messages.

You can send a test mail message to a recipient of your choice by pressing the Test button after you applied the changes.

2.8.8. Step 8 - Set Driverless File Transfer Print

Mail Print and Web Print are disabled by default. You can enable and configure these options at Options → Mail Print and Options → Web Print.

If you enabled one of these Driverless File Transfer Print options, decide which PDF converters you want to enable at Options → Advanced → Converters. Beware that you might need to install the converter software on the SavaPage host.

2.9. Step 9 - Share SavaPage Client Files

SavaPage client files are located in directory `/opt/savapage/client`. This includes the User Client, SavaPage Printer Driver and JMX related files. It is useful to share this directory over the network so users can use, copy or install the files they need on their workstation. Common sharing methods include:

- *Samba* - used to share files to Windows based workstations. GUI tools are available on GNU/Linux to help you with sharing the client directory via Samba. However, some system administrators may be more comfortable creating the share by hand-editing the `/etc/samba/smb.conf` file. The following configuration will share the directory in read-only form:

```
[savapage-client]
path = /opt/savapage/client
comment = SavaPage Client
public = yes
only guest = yes
read only = yes
```

- *NFS* - a popular sharing method used for GNU/Linux and Unix based workstations.

> **Note**
>
> The `/opt/savapage/client` directory is standard shared via the `client/` URL. See Appendix E, *URL Cheat Sheet* [362].

2.10. Step 10 - Testing

Now the installation is complete, it is time to do a basic test to check if the system is ready-to-use.

- Pick a workstation and login as a user that is part of the user source as configured in Section 2.8.5, "Step 5 - Set User Source" [18].
- Install the SavaPage Printer Driver. See the instructions at Section 14.2, "PostScript Driver Print" [262].
- Open a Web Browser and go to the User Web App at `https://savapage:8632/`
- Login to the Web App with the same credentials as used in the workstation login.
- Print a test document such as a web page or basic document to the SavaPage printer.
- Thumbnail images of the printed pages should appear in the Web App.

2.11. What's next?

Congratulations! At this point you have a ready-to-use SavaPage system. This concludes the Install Guide.

If you like, take some time to further explore the features of SavaPage in a more extensive free-format test drive.

Or, continue reading about the user interface details at Chapter 3, *User Web App* [22], Chapter 4, *Admin Web App* [89], Chapter 5, *Job Tickets Web App* [211] and Chapter 7, *Point-of-Sale Web App* [226].

At this point you can also proceed with the configuration of IPP Everywhere and AirPrint, Mail Print and Web Print.

Chapter 9, *Print Site Web App* [234] is a Feature Preview for setting up a location with self-service printers and copy machines.

Chapter 10, *PDF/PGP Verification* [237] is a Feature Preview of an alternative PKI method to verify authenticity and integrity of PDF documents.

Chapter 11, *User Client* [240] explains how to use a *system tray* notifier of SavaPage print events for desktops and notebooks.

Chapter 12, *SavaPage Financial* [243] introduces the main pay-per-print concepts with references to more detailed parts of the manual.

Chapter 13, *SavaPage on GNU/Linux* [244] offers an in-depth explanation of the GNU/Linux installation process, the directory layout and tools involved.

Chapter 14, *SavaPage as Printer* [260] explains how for print from different platforms.

Chapter 15, *Authenticated Printing* [274] describes how SavaPage determines the digital identity of users in different settings like Single Sign-On (SSO) Domains and Peer to Peer Networks.

Chapter 16, *Printing Impact* [282] explains the metrics used when giving users feedback about the costs and environmental impact of their printing habits.

Chapter 17, *Security* [284] discussed security issues and precautions.

Chapter 18, *Freedom and Privacy* [295] explains how digital freedom and privacy is secured in the SavaPage domain.

Chapter 19, *Internationalization* [297] explains how SavaPage is adapted to various languages and regions.

Chapter 20, *Customization* [300] explains how SavaPage can be customized to match your organizational identity.

Chapter 21, *Using an External Database* [311] explains how to use an alternative *external* relational database.

Chapter 22, *Tuning* [314] discusses performance optimization and parameter tuning.

Chapter 23, *SavaPage Community* [320] describes the SavaPage Community and explains how to use the Service Card.

Appendix A, *Proxy Print Scenarios* [321] summarizes several Proxy Print scenarios in a shorthand catalogue.

Appendix B, *NFC Authentication* [326] explains how SavaPage supports RFID as authentication method.

Appendix C, *Tools* [328] explains the command-line interface for calling server methods, manipulate the database, stop and start the server, and for applying SSL certificates for secure HHTP connections.

Appendix D, *Capacity Planning* [360] discusses how SavaPage uses disk space and network resources.

Appendix E, *URL Cheat Sheet* [362] offers a Quick Reference Card of the available Web Interface URLs.

Appendix F, *File Locations* [365] is an outline of locations and files in `/opt/savapage/`.

Appendix G, *Printable File Types* [369] gives a summary of the file formats supported by Driverless File Transfer Print.

Appendix H, *Upgrading from a Previous Version* [373] describes the procedure to install a new version.

Appendix I, *Migrating to a New Server* [374] describes the procedure to move your current SavaPage installation to a new server.

Appendix J, *Advanced LDAP Configuration* [376] gives an in depth explanation of the LDAP configuration options.

Appendix K, *PPD Extensions* [382] explains how to map vendor specific PPD keywords to IPP attributes.

Appendix L, *IPP Extensions* [397] gives a summary of IPP attributes and values as extensions to the IANA registrated ones.

Appendix M, *SavaPage Plug-ins* [405] explains how to deploy software components that add specific features to SavaPage.

Appendix N, *PaperCut Integration* [412] explains how functions not present in PaperCut can be implemented with SavaPage as pre-processor and integrator.

Appendix O, *Job Scheduling* [420] describes SavaPage background job scheduling.

Appendix P, *GNU Affero General Public License (AGPL)* [422] contains the full text of the AGPL.

Chapter 3. User Web App

The User Web App can be reached at `https://savapage:8632/`. For all URL options see Appendix E, *URL Cheat Sheet* [362].

> **Important**
>
> When using the User Web App concurrently with the User Client and Proxy Print Authentication you are strongly advised to use an external database like PostgreSQL. See Chapter 21, *Using an External Database* [311].

3.1. Login

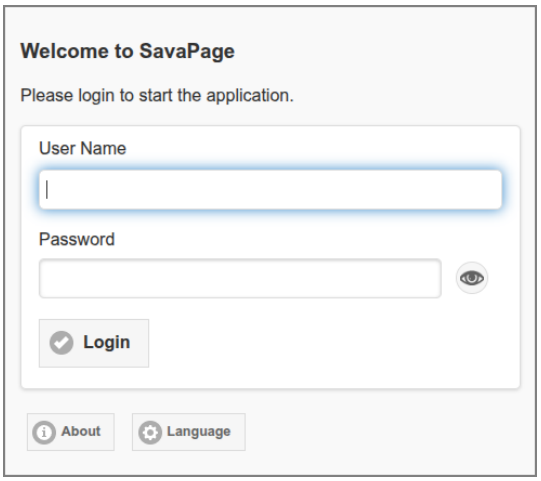


Figure 3.1. Web App: Login Dialog

> **Note**
>
> When a user opens the Web App the login dialog is skipped when an Authentication Token is present in *local storage* of the browser. The login dialog is also skipped when the Web App is opened from a trusted and authenticated User Client or with a One-Time Authentication Token.
>
> For a description of the global user authentication defaults see Section 4.11.4, "User Authentication" [157].

- The language of the dialog defaults to the language setting of the browser.
- You can overrule the default language and country or preselect a user by using the URL parameters. See Appendix E, *URL Cheat Sheet* [362]
- Version and copyright information is shown when you press the About button. See Section 3.1.1, "About" [23].
- You can choose an alternative language by pressing the Language button. See Section 3.1.2, "Select Language" [24].

- The top of the Login dialog can be customized: see Section 20.1.1.3, “Custom HTML” [302].
- Use the Eye button on the right side of the **Password** entry field to show/hide the password.
- Login Alternatives appear at the bottom of the dialog.
- Rate Limiting is applied to login failures.

Some invariants:

- Only Persons can login.
- Disabled users are not allowed to log in.
- The internal "admin" user is not allowed to log in as user.
- As long as system setup is needed user login attempts are blocked with a message saying “Application setup is required”.
- When the system is in Maintenance Mode, access is restricted to users with Administrator role. Regular users are shown a message explaining the situation at this Login dialog, or after login in the authorized session.

Tip

You can use an *alias* as **User Name**. See Section 15.5, “User Name Aliases” [280].

3.1.1. About

The About dialog shows version and copyright information. The top of the dialog can be customized: see Section 20.1.1.3, “Custom HTML” [302].

Note

The dialog contains a Printer Driver section with a download link for the CUPS SAVAPAGE.ppd file. This section can be enabled or disabled by setting the configuration key `webapp.about.driver-download.enable` to `Y` or `N`. See Section 4.11.15, “Config Editor” [189] on how to change this value.

3.1.2. Select Language

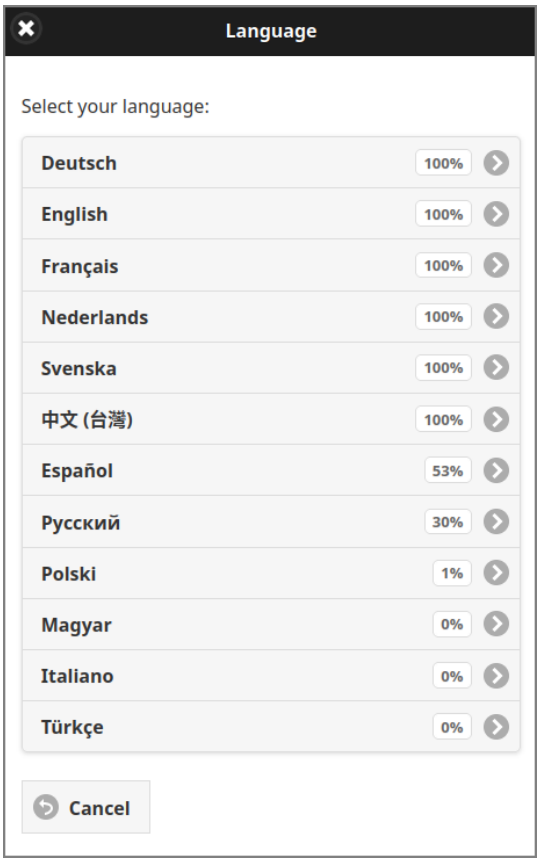


Figure 3.2. Web App: Select Language Dialog

- At the moment English, French, German, Swedish, Chinese (Traditional, Taiwan) and Dutch are fully supported. Press the language of your choice. This will reload the login dialog in the newly selected language.
- Press Cancel to return to the login dialog.

Note

You can use the `webapp.language.available` configuration property to enter a comma separated list of selectable ISO 639 language codes. For instance: `de,en,fr,es,ru,nl`. See Section 4.11.15, "Config Editor" [189] on how to enter this value.

When the list is restricted to a single language, the Language button in is not shown in the Login dialog, and the Web App Locale is set to the single available language.

3.1.3. Self Registration

If User Self Registration is enabled a Register button is displayed.

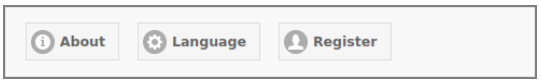


Figure 3.3. Web App: Register Option

If Register is pressed, registration data can be entered.

Note

Rate Limiting is applied before the Registration dialog is shown. This could result in a message that registration is currently unavailable. In that case, try again later.

Press Send for verification to the specified **Email** adress. This address must match one of the approved domains. See Section 4.11.1.3, "Internal Users" [153]. For email customization see Section 20.2.6.3, "UserRegVerification" [309].

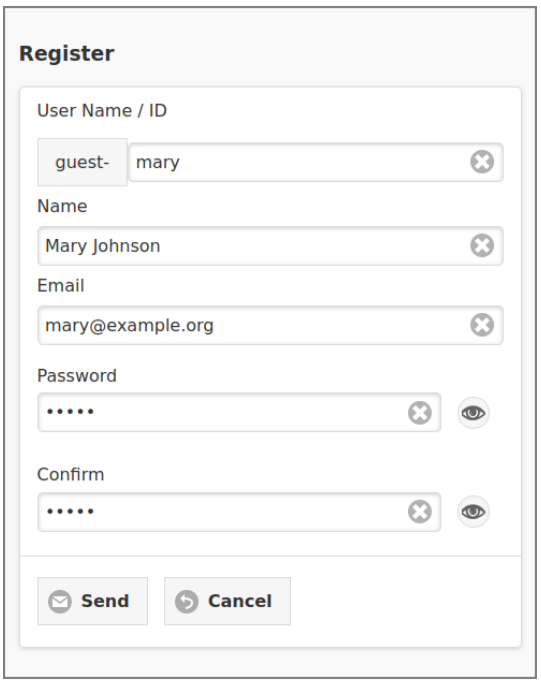


Figure 3.4. Web App: Register Dialog

A prefix (in this case **guest-**) is applied to the **User Name / ID** to distinguish registered Internal Users from users synced from a User Source. The prefix and other parameters are held as configuration property, as explained in Section 4.4.6, "Create Internal User" [108].

By pressing Send a verification link is send to the specified **Email** adress. The address must match one of the approved domains. See Section 4.11.1.3, "Internal Users" [153]. For email customization see Section 20.2.6.3, "UserRegVerification" [309].

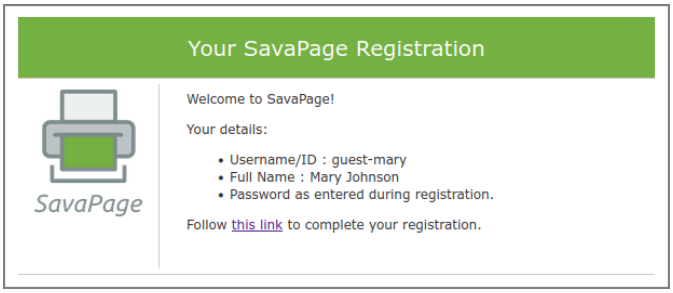


Figure 3.5. Self Registration Verification Email

The status of the verification (Pending or Expired) is shown in the User List. See Section 4.4.4, "Registration Requests" [102].

After the user follows the link in the verification email, the registration is completed.

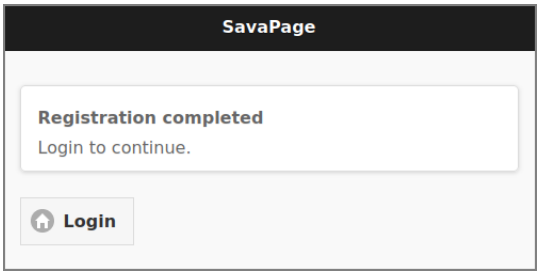


Figure 3.6. Self Registration Completed

3.1.4. Single Web App Session

A warning message is shown when a *desktop* [1] user tries to open the *same* Web App type a second time in the same browser instance. In rare occasions, for instance, due to network connectivity issues, this statement might be false. In that case a Login will bring you back on track.

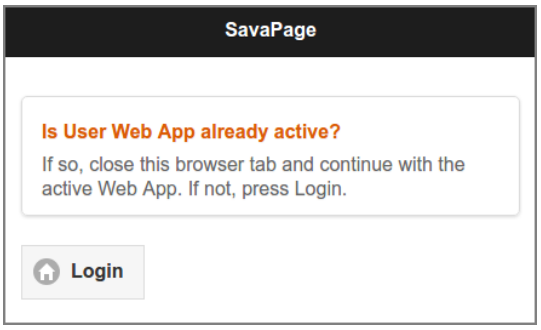


Figure 3.7. Same type Web App session detected

When a user opens a second Web App session of *another* type this message is shown:

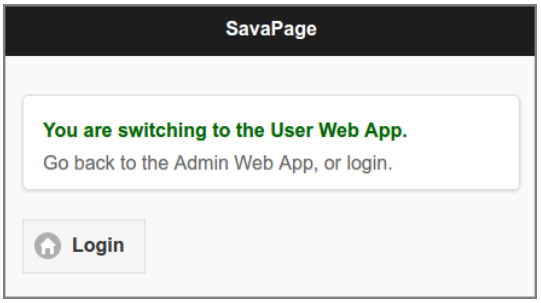


Figure 3.8. Web App type change detected

In both situations, either go back to the active Web App session or press Login to login to the intended Web App type. This will invalidate any other SavaPage session in the same browser instance.

3.1.5. Card Self Association Dialog

If an unknown card is swiped, and NFC Card Self Association is enabled, the user is presented this dialog to associate the new card.

[1] The Single Web App Session check is solely done for certain *desktop* browser sessions. Sessions on macOS and mobile devices are *not* checked.

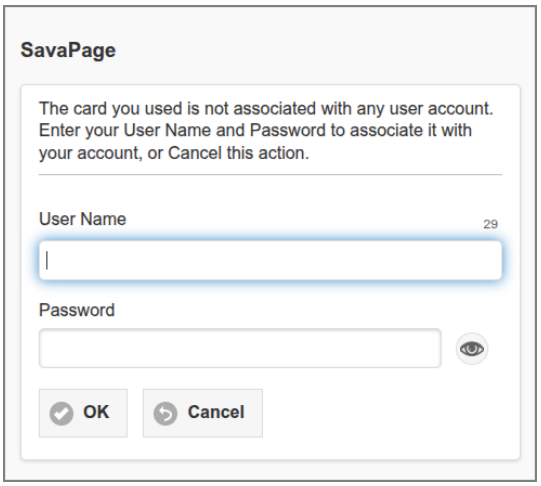


Figure 3.9. Web App: Login Dialog - Card Self Association

There is a time limit to enter the Username and Password. The remaining seconds are shown and when counted down to zero the dialog is automatically closed. The time limit (seconds) is contained in configuration key `webapp.card-assoc.dialog-max-secs`. See Section 4.11.15, "Config Editor" [189] on how to change this value.

3.1.6. 2-Step Verification

If a user enabled 2-Step Verification, they are prompted to enter their verification code, after a successful first login step.

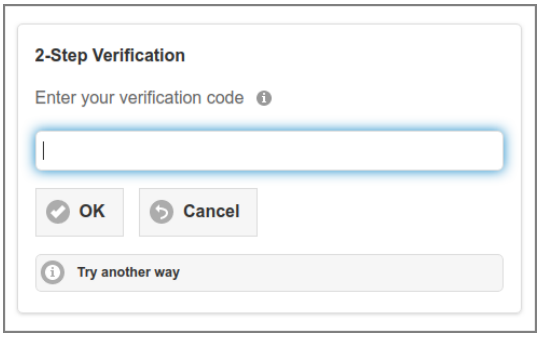


Figure 3.10. Web App: Login Dialog - 2-Step Verification

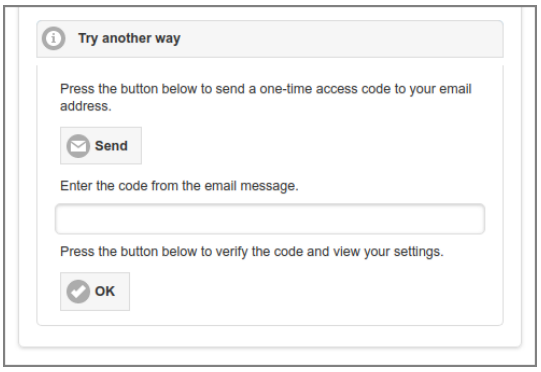


Figure 3.11. Web App: Login Dialog - 2-Step Verification (recovery)

3.2. Login Alternatives

The appearance of the Login dialog on a device depends on the following settings:

- The globally activated Login Methods. See Section 4.11.4, “User Authentication” [157].
- The Terminal settings for the device. See Section 4.9.2.2, “Custom User Login” [147].
- The `login` URL parameter with the preferred login method. See Appendix E, *URL Cheat Sheet* [362].
- Active OAuth Client Plug-ins.

Terminal settings overrule global settings, and the URL parameter overrules the defined default. When available, alternative login methods can be selected by tapping the method button at the bottom of the dialog. Some sample Login dialogs are shown below.

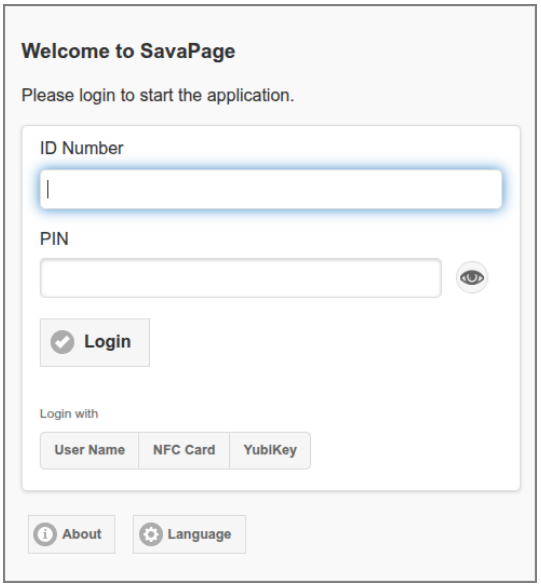


Figure 3.12. Web App: Login Dialog - ID Number

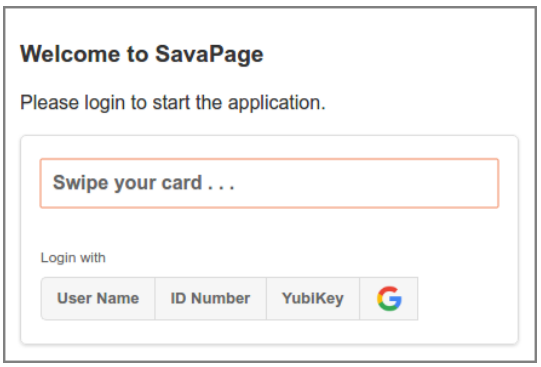


Figure 3.13. Web App: Login Dialog - Local NFC Card

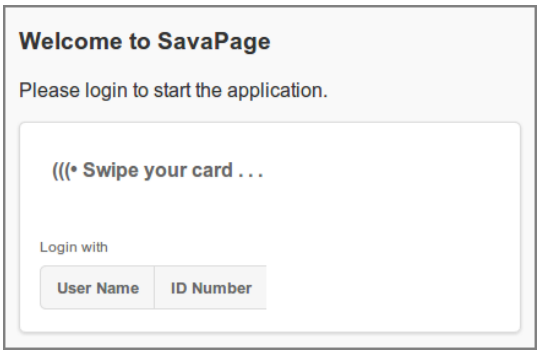


Figure 3.14. Web App: Login Dialog - Network NFC Card

> **Tip**
>
> A special URL path is available to use OAuth as SSO for User Web App. This URL can be presented on the site of an OAuth provider, to seamlessly link to SavaPage. See Appendix E, *URL Cheat Sheet* [362].

3.3. SafePages

This is the main view with the acquired SafePages since the last login. This view is also known as “inbox”.

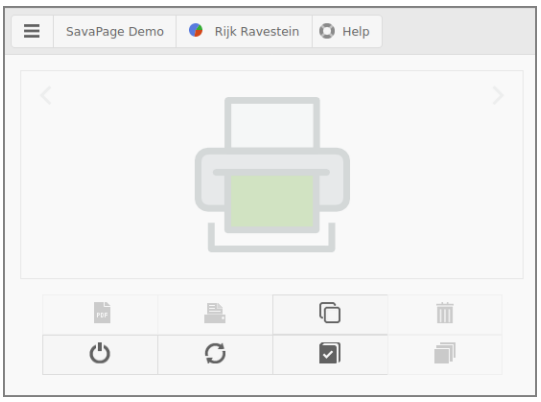


Figure 3.15. User Web App: Main View

The top bar holds the following items (from left to right):

- An action button that opens a pop-up menu. For now, the SavaPage logo with the GDPR and About button is available. See Section 3.13, “GDPR Dialog” [88].
- The name of your Organization, in our case "SavaPage Demo".
- A button with the inline pagometer Pie-Chart followed by the id of the logged in user. The *blue* color in the chart represents the number of pages the user printed to SavaPage. The *green* color represents the number of pages exported to PDF. The *red* color depicts the pages printed to Proxy Printers. The button opens a dialog with User Details, including pagometer details.
- The Help button opens a custom URL in a new tab. The URL is set with configuration property "`webapp.user.help.url`" and is enabled with "`webapp.user.help.url.enable`" default value `Y` (disable with value `N`).

The two-line center bar holds buttons for the main work-flow. When no SafePages are present, the SavaPage logo is shown, and irrelevant buttons are disabled (these buttons are described at Figure 3.16, “User Web App: SafePages” [30]).

By default, the center bar shows button text on mobile devices only: on desktops a hover text is shown. You can change this behavior by setting configuration property `webapp.user.main.nav-button-text` to value `ON` (button text is always shown), `OFF` (button text is never shown) or `AUTO` (button text is shown on mobile devices only).

See Section 4.11.15, “Config Editor” [189] on how to change configuration properties.

Important

Depending on User Privileges and User Roles some buttons might not be shown, replaced (relocated).

The Letterhead button brings you to a dialog where you can create, browse and select letterheads. See Section 3.6, “Letterheads” [68].

Press the Log button to get a list of the See Section 3.8, “Log” [71].

The Logout button brings you back to the login screen.

Press the Refresh button when, due to whatever reason, the automatic detection of SavaPage changes fails. This will update the view with the latest state.

Note

Each print to SavaPage is logged as Document. SafePages that do not match a logged Document are removed. This can happen when a database is restored, or when old documents are deleted after a Database Backup, or Database Command.

Figure 3.16. User Web App: SafePages

This screen shows the result of a user who printed 3 pages to the SavaPage printer. The following actions on thumbnails are defined:

- You can scroll through the thumbnails by dragging them horizontally.
- A *tap* on the transparent area "<" or ">" in the top corner of the thumbnail view port scrolls the view a single page to the left or right. A *taphold* brings the start or the end of the page range in view.
- A *tap* (click) opens up a detailed view of the page in the Page Browser: Figure 3.26, “User Web App: SafePage Browser (8 pages)” [37].
- A *tap* on the page number underneath the thumbnail or a *taphold* on the thumbnail itself, opens the Document Details dialog with rotate, delete and undo actions: Section 3.3.3, “Document Details” [34].

The following sections describe the actions for the newly enabled buttons:

Section 3.4, “PDF” [40].

When the user is a Job Ticket Creator and *not* a Print Job Creator, or Job Ticket Printers are the only available printers, a Ticket button is shown instead of a Printer button. Section 3.5, “Print” [47].

Section 3.7, “Delete” [70].

Section 3.9, “Sort” [77].

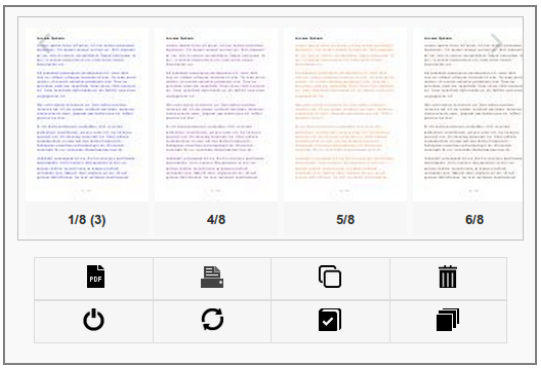


Figure 3.17. User Web App: SafePages - Aggregated

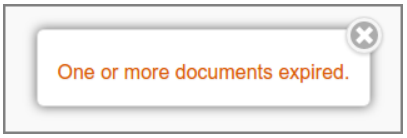
This screen shows the result of a user printing 8 pages to the SavaPage printer, and illustrates *thumbnail aggregation*.

- Note that only 6 thumbnails are displayed, and that the first thumbnail tells by its numbering that it is the first of a (3) page aggregation.
- A *tap* on the first thumbnail will bring the (3) aggregated thumbnails in view. As a side-effect an aggregate will appear at another location in the thumbnail sequence.
- Thumbnail aggregation is a protection against information (and resource) overload. Imagine what would happen if you printed a 500 page document to the SavaPage printer and ended up with 500 thumbnails. Aggregation gives you the high-level means to easily zoom in and out.
- A mark after the page number tells that the page was edited with SavaPage Draw. In case of an aggregated thumbnail, a mark is shown prefixed with the underlying number of edited pages.
- As always, a *tap* on a single thumbnail will bring you to the Page Browser, where you can navigate to any page, sequentially or directly. See Figure 3.26, “User Web App: SafePage Browser (8 pages)” [37].

3.3.1. Document Expiration

When the document expiration time signal is set and expiration for a document is due, every thumbnail of the document is marked with a clock icon in a colored (orange) footer.

When the document is auto-deleted after expiration a notification message us shown.



See Section 4.11.14.9, “SafePages” [184].

3.3.2. Footer

The footer is positioned at the bottom of the main user panel. The base items are depicted in the figure below.

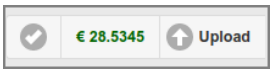


Figure 3.18. User Web App: Footer Base

From left to right:

- A *check-mark* icon as indication that the Web App is connected to the SavaPage server. Other icons are shown when the connection is being (re) established or lost. When the server is not down, this usually is a temporary condition due to a network hiccup. Don't worry, SavaPage will automatically restore the connection when the network permits.
- The account balance of the logged in user. When you press the button a dialog with User Details is shown, including pagometer details. This button might not show depending on Financial User Privileges.
- The Upload button is shown if Web Print is enabled. A tap on the button shows the Upload File dialog. This button may have moved to the center button bar, depending on User Privileges.

> **Note**
>
> Depending on User Privileges some buttons might changed, not shown or moved to the main button bar.

3.3.2.1. Paper Size Indicator

If SafePages are present the unique paper sizes of the jobs are depicted in the footer. The text color indicates if a paper size is supported by the selected printer or not. The examples below illustrate how this works.

A4 and A3 jobs are present: a printer is selected that supports both paper sizes.

A4 and A3 jobs are present: a printer is selected that supports solely the A4 paper size. The A3 page is *cropped* to A4.

A4 and A3 jobs are present: a printer is selected that supports solely the A4 paper size. The A3 page is *shrinked* to A4.

A4 and A3 jobs are present: a printer is selected that supports none of the paper sizes (or no printer is selected yet).

3.3.2.2. SavaPage Draw Indicator

The number of pages edited with SavaPage Draw are shown in the footer. The example below shows that 2 pages were edited.

3.3.2.3. Hold Print Jobs

A summary of Hold Print Mode jobs and Job Tickets as a result of Proxy Printing is shown in the footer. The example below explains the layout.

From left to right:

- The shortest remaining job release time.
- The total number of hold jobs.
- The total number of pages to be printed.
- The total cost charged for printing.

A tap on the summary show a dialog where all Hold Jobs are shown in detail. Each job has three button, from left to right:

- Show invoicing information.
- Preview the PDF document to be printed.
- Cancel the job.

And for all jobs:

- Cancel (n) all n jobs.
- Extend the expiry time of Hold Print Mode jobs.
- Refresh dialog content

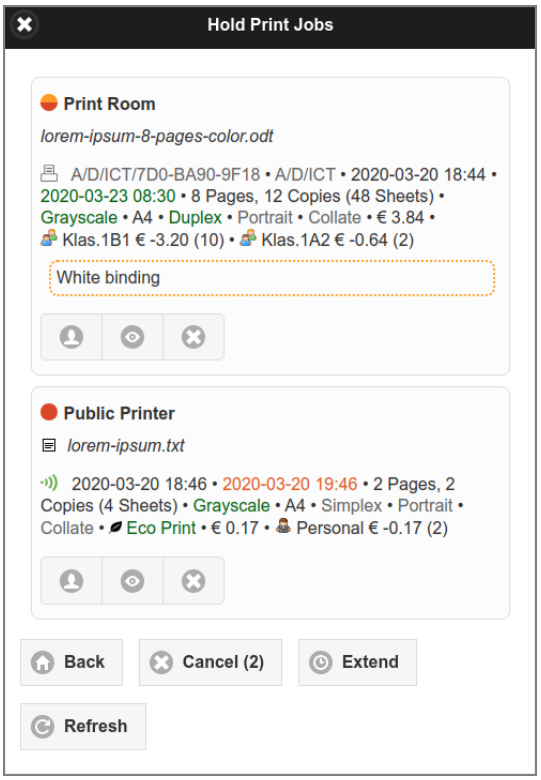


Figure 3.19. User Web App: Hold Print Jobs

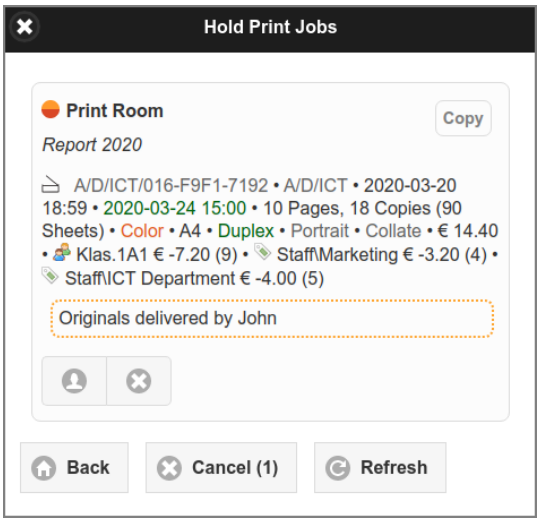


Figure 3.20. User Web App: Hold Copy Job

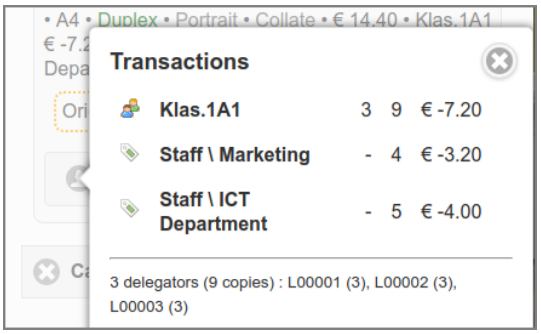


Figure 3.21. User Web App: Hold Job Transactions

More information can be found at:

- Section 3.5.5, “Print Job Settings” [53].
- Section 3.5.9, “Job Ticket Print” [64].

3.3.3. Document Details

A *Tap* on the page numbering below the image pops up the Document Details dialog.

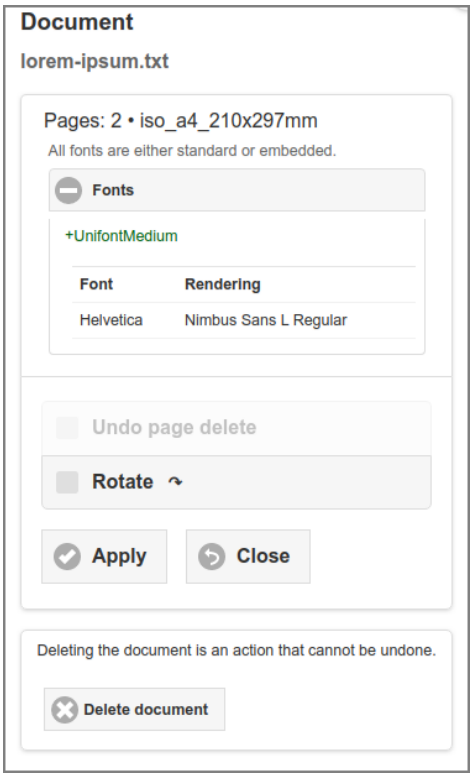


Figure 3.22. User Web App: Document Details

In this example "+UnifontMedium" is an embedded font and "Helvetica", as one of the Standard PDF Fonts, is rendered as "Nimbus Sans L Regular". How fonts are resolved in SafePages Documents, depends on Driverless PDF Transfer Processing and PostScript Print Processing.

If the PDF document is transferred via HTTP (Upload), the details of the Original PDF are shown is a separate section. The same details are available in the corresponding item of the Document List.

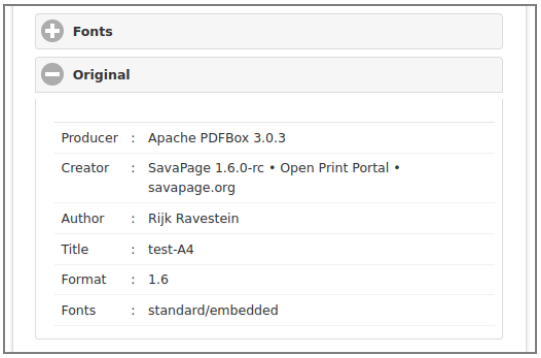


Figure 3.23. User Web App: Document Original Details

If document expiration is set the expiration time is shown. See Section 4.11.14.9, "SafePages" [184].

> **Note**
>
> If Inherit IPP options from print-in document is enabled for Fast Print Mode the IPP Options from PostScript Driver Print-in are displayed.

3.3.3.1. Delete and Undo

Press the Delete document button in this dialog to *delete* all pages of a SafePages document, or check and apply Undo page delete to restore the full job in case document pages were deleted. See Section 3.9, “Sort” [77].

3.3.3.2. Rotation

When a user prints to a printer and selects landscape orientation, the print manager of the originating application will translate and rotate the printed content to fit the dimensions of the hard copy target page.

When doing so, it makes assumptions about the (0,0) origin of the logical space on this page. The SavaPage printer driver provides a hint to the print manager about the origin, so it can rotate and translate the pages in a way that is compatible with the SavaPage printer.

Contrary to real printers, where hard copies can easily be rotated by hand, pages produced by the virtual SavaPage printer need special attention, since it is possible that landscape oriented prints rotate the content to portrait oriented images and PDF pages. Probably this is not what you want, so you can ad-hoc rotate job pages in SavaPage to landscape display orientation.

Figure 3.24. User Web App: Landscape Job

- This what you might see when you print a job in landscape orientation.
- Select the *Rotate* option and press the Apply button to rotate the page and all sibling pages belonging to the same job.
- The result after rotation is shown in Figure 3.25, “User Web App: Rotated Pages” [37].

> **Note**
>
> Although the Rotate dialog is triggered from a *single* SafePage, the rotation affects *all* SafePages within the same print job.

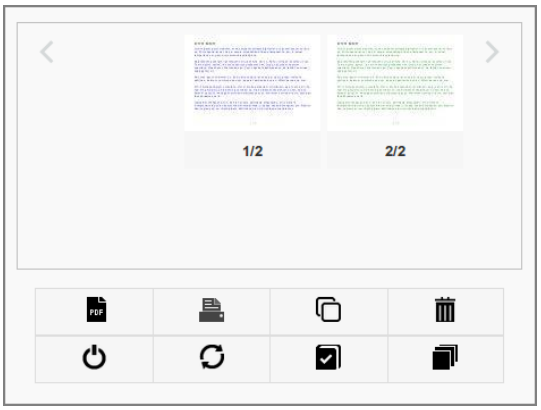


Figure 3.25. User Web App: Rotated Pages

- The SafePages after rotation.

> **Warning**
>
> When two WebApps are opened for the same user, the result of a page rotation performed in one Web App will not automatically be shown in the other. The user should do a manual refresh to update the SafePages preview.

3.3.4. Browser

A *tap* on a non-aggregated SafePage thumbnail image will show the page detail in the Browser.

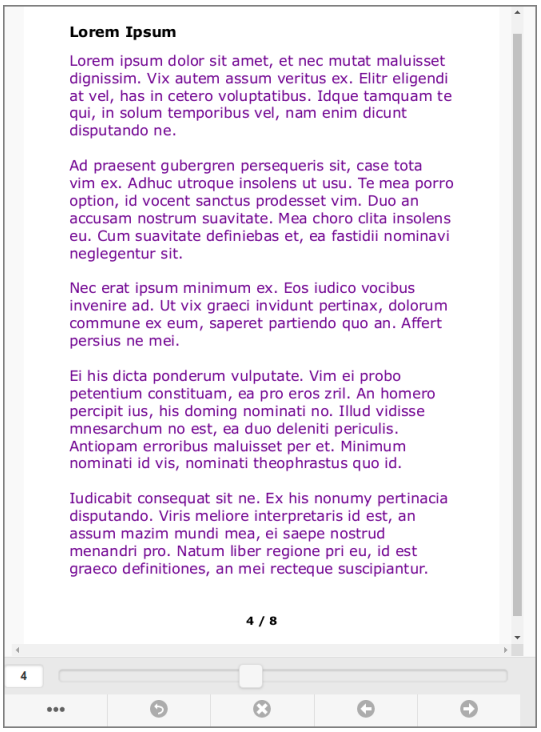


Figure 3.26. User Web App: SafePage Browser (8 pages)

Use the *slider* control to directly jump to the page of your choice. Extra actions are available by buttons at the bottom. From left to right:

- The `...` button opens a menu where you can zoom-in, zoom-out and restore the zoom default. See Figure 3.27, “User Web App: SafePage Browser - Page Options” [38].
  - Depending on SavaPage configuration and user permission you will see an `Edit` button to toggle the SavaPage Draw panel.
- The *Return* button brings you back to the main SafePages screen: Section 3.3, “SafePages” [29].
- The *X* button deletes the page in view.
- The *arrow-left* button selects the previous page.
- The *arrow-right* button selects the next page.

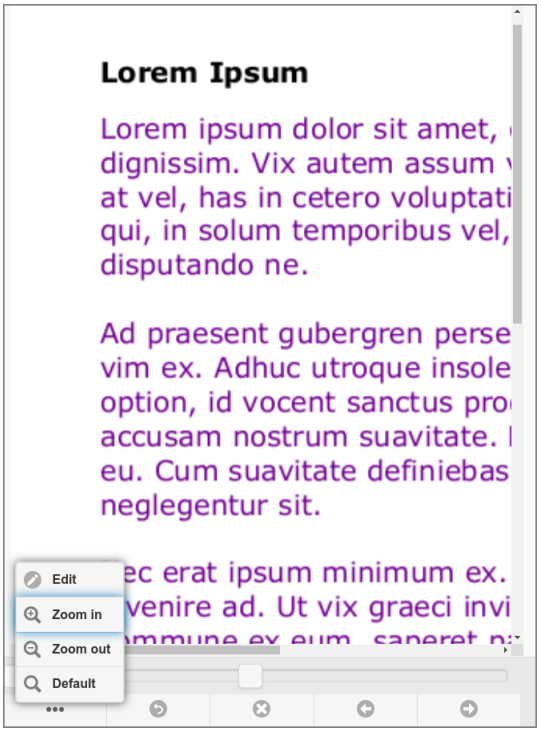


Figure 3.27. User Web App: SafePage Browser - Page Options

- This screen shows the Menu options with Zoom-in applied. Use the standard page scrolling of your browser to scroll the image up/down and left/right.
- The optional `Edit` button toggles the SavaPage Draw panel.

3.3.4.1. SavaPage Draw

If rsvg-convert is installed and a user has SafePages Editor privileges they can draw lines, shapes and text on a SafePage as shown in the example below.

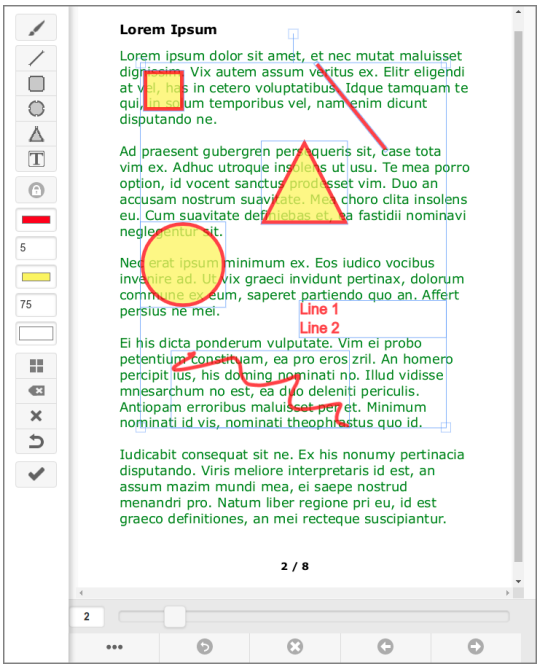


Figure 3.28. User Web App: SavaPage Draw

Buttons at the left panel can be used to select properties and draw lines, shapes and text, as explained in the table below.

Button	Action
Brush	Toggles Brush (ON) or Shape (OFF) mode. Brush mode draws a trail with Color, Size and Opacity under the moving mouse pointer while pressing the left mouse button. Shape mode enables the Shape and Text buttons. A drawn Brush trail can be edited by selecting it in Shape mode.
Line, Rectangle Circle, Triangle, Text	Draws a standard size of the respective Shape or Text at the top left of the page with Border, Fill and Opacity. A Text object is initialized with two sample text lines. The drawn object is selected by default so it can immediately be edited, resized and moved. Note: buttons are disabled when Brush mode is ON.
Lock	Toggles Lock mode for Border and Fill Color. When Lock mode is ON, pressing a Color button does not open the color selection dialog, but just applies the color to the selected drawn objects.
Brush | Border Color	Selects the Brush or Shape Color (depending on Brush or Shape mode). The selection is applied to any selected object. If Lock mode is ON, the color selection dialog does not open, and the locked color is applied to any selected object.
Size | Border	Selects the Brush Size or Shape Border (depending on Brush or Shape mode). The selection is applied to any selected object.
Fill Color	Selects the Fill Color of a Shape. The selection is applied to any selected object. If Lock mode is ON, the color selection dialog does not open, and the locked color is applied to any selected object.
Opacity	Select the Opacity of a Brush or Shape object. The selection is applied to any selected object.
Transparent	Makes the Fill of any selected object Transparent.
Select all	Selects all drawn objects (Shape and Brush trails).
Clear selection	Clears (deletes) all selected objects.
Clear all	Clears (deletes) all drawn objects.
Restore, Save	Saves the drawing or restores the earlier saved version.

Table 3.1. SavaPage Draw - Buttons (top to bottom)

Note

The number of edited pages are shown in the SafePages footer. SavePages thumbnails of edited pages are marked with a small ✎ icon.

Note

SavaPage Draw can be globally disabled by setting configuration property

`webapp.user.page-browser.draw.enable` = `N`

See Section 4.11.15, “Config Editor” [189] on how to set this property.

Caution

The alignment of the drawing does not adjust to a change in Document Rotation.

Warning

Do not use SavaPage Draw as a tool to redact content in PDF exports. Redaction applies only to the visible surface: the underlying elements remain in the PDF and can be inspected with special tooling.

3.4. PDF

A *tap* on the PDF button in the main SafePages view shows a dialog with PDF properties and export actions. See Section 3.3, “SafePages” [29].

Note

All PDF properties (except Passwords) are preserved when the dialog is closed or a PDF is generated, and are re-used when needed in current or future sessions.

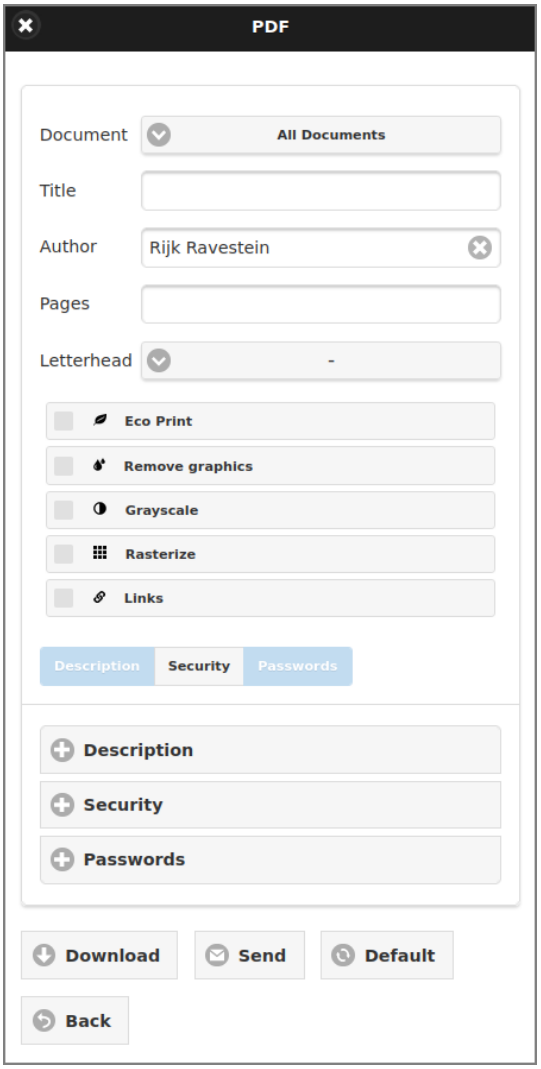


Figure 3.29. User Web App: PDF - Overview

- This screen shows the full PDF dialog.
- The *Author* defaults to the name of the authenticated user at login, and can be edited.
- Details are discussed at:
  - Section 3.4.1, “PDF Filters” [42]
  - Section 3.4.2, “Links” [42]
  - Section 3.4.3, “Document Scope” [42].
  - Section 3.4.4, “Description” [43].
  - Section 3.4.5, “Security” [43].
  - Section 3.4.6, “Passwords” [45].
  - Section 3.4.7, “Letterhead” [46].
  - Section 3.4.8, “Download” [46].
  - Section 3.4.9, “Send” [46].
- A selection of SafePages to incorporate into the PDF output can be entered as a range of **Pages**. For example: `1-4,6,8-10`. The value can be a single page, a range of pages, or a collection of page numbers and ranges separated by commas. The pages will always be exported in ascending order, regardless of the order of the pages

in the page-ranges option. The page range is automatically emptied after printing. Be aware that the page ordinals are related to the Document Scope.

- Check one of the PDF Filters Eco Print or Remove graphics. Additionally check Grayscale and/or Rasterize.
- You can activate a Description, Security and Passwords setting by toggling the corresponding button in the so-called **Apply** section. A toggle button is disabled when no setting is specified.

> **Note**
>
> SavaPage tries to translate URL formatted text like “www.example.com” and “info@example.com” to PDF links. Implicit URLs in the source document, such as those contained in text like “click here”, are not sent to the SavaPage printer, and therefore not preserved in the PDF document.

3.4.1. PDF Filters

Activate *either* the Eco Print or Remove graphics filter option. The last option removes all graphic images from the PDF. When choosing Eco Print the selected SafePages will be ad-hoc converted in the background. While conversion is busy a message box will tell you to wait a while and retry later. Take about 3 seconds per page waiting time into consideration. Automatic filtering may help to diminish waiting times: see Section 4.11.10.2, “Eco Print Settings” [172]. Grayscale and Rasterize filters can be selected as extra. The Rasterize filter creates an image of each PDF page. The DPI can be set with the following configuration properties:

Configuration property	Description
pdf-out.convert.rasterize.dpi.download	DPI of rasterized PDF for Download. Enum values: `DPI_100`, `DPI_150`, `DPI_200`, `DPI_300` (default), `DPI_600`.
pdf-out.convert.rasterize.dpi.email	DPI of rasterized PDF Send by email. Enum values: see above.

Table 3.2. PDF Rasterize Filter: Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to change these properties.

3.4.2. Links

If the Links option is checked, PDF text recognized as URI is annotated as PDF link and optionally underlined with a thickness and color of your choice. Select “Invisible” to hide underlining.

Figure 3.30. User Web App: PDF - Links

3.4.3. Document Scope

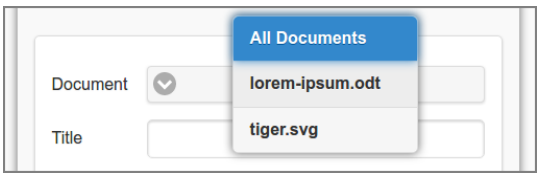


Figure 3.31. User Web App: PDF - Document Scope

- When pressing the **Document** button you get a selection pop-up with titles of acquired SavaPage print jobs. *Tap* a job title to restrict the scope of the PDF output to that job. Select the top item **All Documents** to activate *full* scope.
- When *full* scope is selected the **Pages** ordinals are related to the SafePages total. When a job is selected as scope, the page ordinals are related to the page total of the job.
- Selecting a scope initializes the **Title** text with the job name, or clears it when you choose full scope. You can edit the **Title** if you wish.
- If there is just a *single* acquired SavaPage print job, this job is shown as only option.

Note

When a user rearranged or deleted any SafePages the scope is confined to *full* scope.

3.4.4. Description

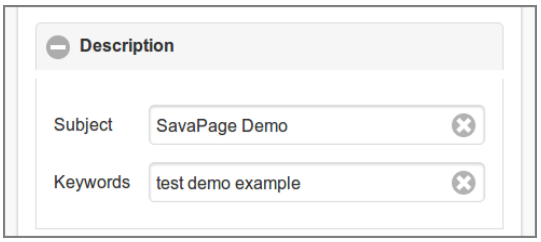


Figure 3.32. User Web App: PDF - Description

- Press the Description button to expand the section.
- Enter the **Subject** and (space separated) **Keywords** of the PDF document.

Note

When a Subject or Keywords are entered, the Description toggle in the **Apply** section will be enabled. Use this toggle to apply (or deny) the Description to the generated PDF.

3.4.5. Security

Press the Security button to expand the section.

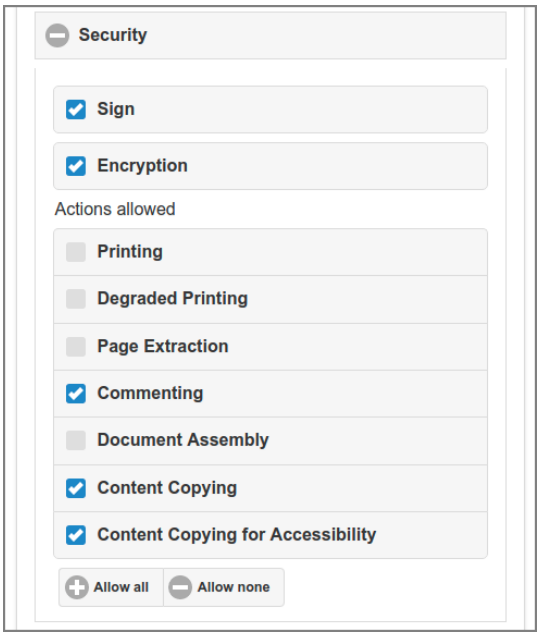


Figure 3.33. User Web App: PDF - Security

When the OpenPGP Key Pair is present on the server, users who have the Privilege to Sign will see the `Sign` option, When selected, an OpenPGP Signature is applied to the generated PDF.

When the `Encryption` option is selected the **Actions allowed** for the generated PDF are shown. Use the `Allow all` or `Allow none` buttons to select the actions in one go. Or select each allowed action separately. This is a list with the allowed actions, each with a short description:

- *Printing*: Printing the document.
- *Degraded Printing*: same as *Printing*, but with a lower quality.
- *Page Extraction*: Modifying the contents. For example, changing the contents of a page, or inserting or removing a page.
- *Commenting*: Adding or modifying text annotations.
- *Document Assembly*: Inserting, removing, rotating and bookmarking pages. The content can't be changed, unless *Page Extraction* is also selected.
- *Content Copying*: Copying or otherwise extracting text and graphics from the document, This also applies for screen readers or other accessibility devices.
- *Content Copying for Accessibility*: Extracting text and graphics for use by accessibility devices.

Note

SavaPage uses 128-bit PDF encryption.

When `Sign` or `Encryption` is selected, the `Security` toggle in the **Apply** section will be enabled. Use this toggle to apply (or deny) the security settings to the generated PDF.

3.4.6. Passwords

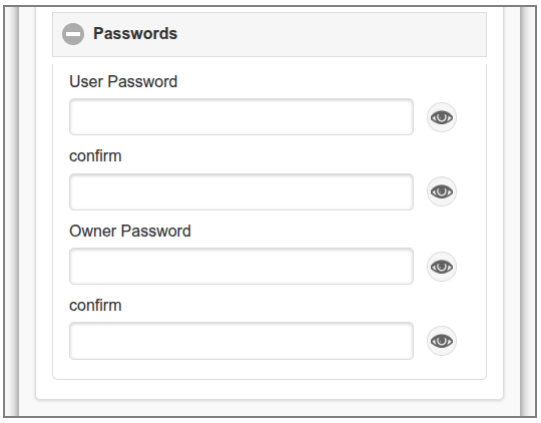


Figure 3.34. User Web App: PDF - Passwords

- The *User* password (also known as the *open password*) locks the PDF file for anyone who doesn't know the password.
- The *Owner* password (also known as the *permissions password*) is needed to read the PDF file in order to change the permissions.
- The maximum password length is 32 characters.
- If you don't enter a user password, all users will be able to open the PDF document without being prompted for a password. However, the security settings will remain in place.
- When both PDF user and owner password are entered they must be different.

Important

When a User password is set or Security settings are active, and the Owner password is *not* set, SavaPage will replace it by a random string.

Warning

Security settings *without* a User password aren't really secure, since the encryption key is derived from the User password. When the User password is omitted, the content is encrypted as described in the public PDF reference, so decryption is also known in this case (although illegal to practice).

Note

When a User or Owner password is entered, the `Passwords` toggle in the **Apply** section will be enabled. Use this toggle to apply (or deny) the passwords to the generated PDF.

Note

Passwords are **not** preserved when the dialog is closed or a PDF is generated, so they can **not** be re-used in current or future sessions.

3.4.7. Letterhead

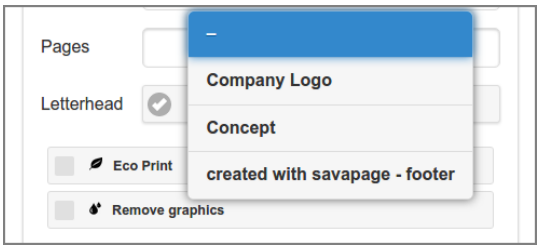


Figure 3.35. User Web App: PDF - Letterhead

- Press the button in the Letterhead section to get a selection pop-up with available Letterheads.
- *Tap* on a letterhead title to select it, or select the top item *dash* to deselect a letterhead.

> **Note**
>
> Depending on User Privileges the Letterhead section might not be shown.

> **Note**
>
> Letterheads are not subjected to PDF Filters but are applied to the filtered result.

3.4.8. Download

- Press the Download button to download the SafePages as PDF file, with the properties set in this dialog.
- Your browser will present a Save dialog so you can save the PDF file in the location of your choice.
- The default PDF file name will be identical to the **Title** you entered as PDF property.

> **Note**
>
> Depending on User Privileges the Download button might not be shown.

3.4.9. Send

A *tap* on the Send button in the PDF dialog, shows this Send dialog. See Section 3.4, “PDF” [40].

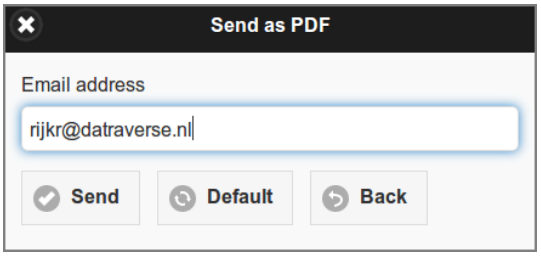


Figure 3.36. User Web App: PDF - Send

- Enter the Email address of the recipient.
- The last used Email address is shown. Press the Default button to reset the address to the one that belongs to the logged in user.
- Press the Send button to generate the PDF document and send it as attachment to the recipient.

> **Note**
>
> Depending on User Privileges the Send button might not be shown.

3.5. Print

A *tap* on the Printer or Ticket button in the main SafePages view shows the Print dialog. See Section 3.3, “SafePages” [29].

If just one printer is available, it is automatically selected, and its Printer Settings dialog is opened.

> **Note**
>
> The Print dialog enables users to set custom printer and job options. When a single copy with default printer options is required, users can apply Fast Print Mode (when this mode is configured for a printer).

3.5.1. Printer Selection

When a printer was not yet selected a Select Printer dialog is displayed.

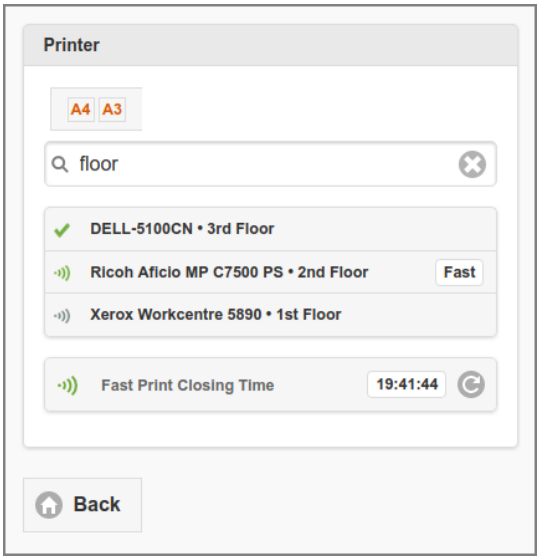


Figure 3.37. User Web App: Print - Select Printer

- A list with a maximum of 5 accessible printers is shown in alphabetic order. Access is according to User Roles “Print Job Creator” and “Job Ticket Creator”.
  - The number of printers shown can be set in the `webapp.user.printers.quick-search.max` config item. See Section 4.11.15, “Config Editor” [189] on how to change this value.
  - Page navigation buttons are added if more printers are available than shown.
- Printers marked with an icon are secured with Hold Print Mode Release. An icon means the printer is secured with Direct Print Mode Release. Printers marked with a **Fast** label are (additionally) enabled for Fast Print Mode Release.
- Job Ticket Printer instances are marked with an icon.
- Printers can be selected by entering part of the printer name or location. In the figure above the text “floor” was entered, resulting in 3 hits.

- You select a printer from the list by tapping (clicking) it. This brings the Printer Settings dialog of the selected printer into view.
- The Fast Print Closing Time button shows the expiration time of Fast Print Release. The time is reset when the Print Dialog opens or the button is pushed. This button is shown when the user is granted access to at least one proxy printer with Fast Print Release enabled.

Note

When User Role permits access to Job Ticket Printers only, the dialog will show “Ticket” as title.

3.5.2. Printer Settings

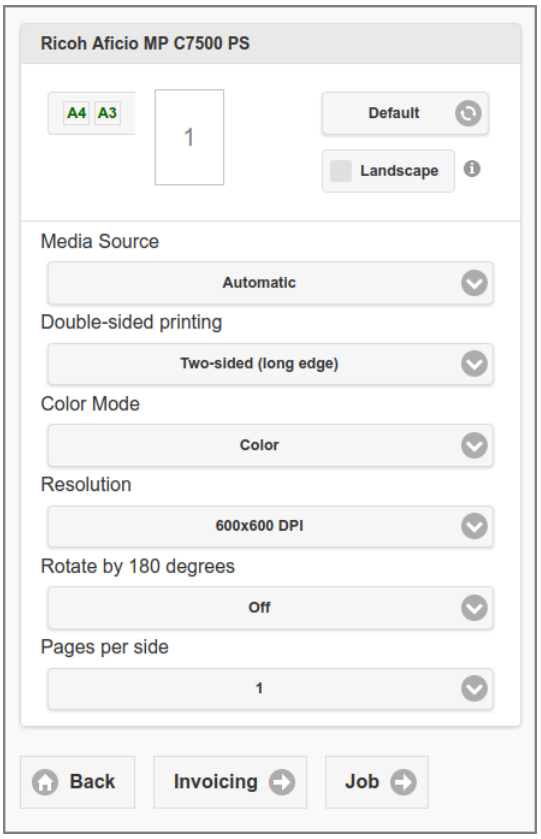


Figure 3.38. User Web App: Printer - Settings

Set one or more printing options by pushing the pick-list buttons. The options are initialized with the CUPS printer defaults at the start of a user session. Changes made in this dialog are held during a user session, unless they are cleared after a proxy print: see Table 3.5, “Print Job Settings Configuration Properties” [56].

Options are printer specific, and are automatically identified by SavaPage. See Section 3.5.2.4, “Printer Setting Options” [51].

A preview of the Finished Page is shown at the top. A checkbox shows the orientation (Portrait, Landscape) of the first page of the selected document or page range. See Section 3.5.2.2, “Print Preview” [51] for more information.

The Invoicing button is visible when Proxy Print Delegation is enabled and the user has Print Job Delegate role. Pressing it opens the Print Delegation Dialog.

Pressing the Job button opens the Print Job Settings.

> **Note**
>
> Options are validated with SPConstraint rules, when leaving the Printer Settings dialog with Invoicing or Job button.

If options present in the CUPS PPD file are missing you can add them, as explained in Appendix K, *PPD Extensions* [382]. In this way you can add finishing options:

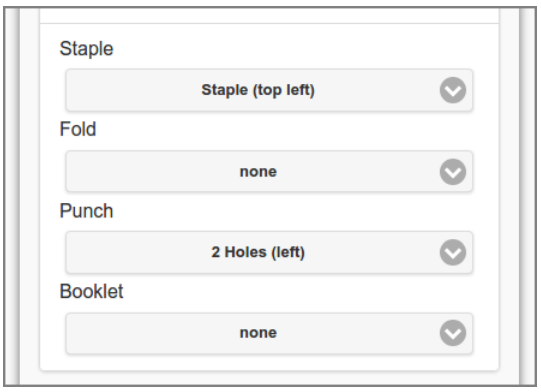


Figure 3.39. User Web App: Printer - Settings - Finishings

Or Job Ticket Options:

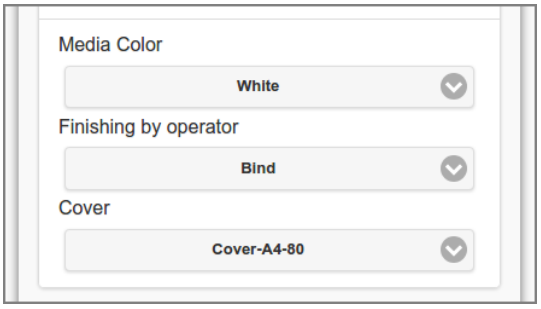


Figure 3.40. User Web App: Printer - Settings - Job Ticket

When all paper sizes of SafePages jobs are supported by the printer, each paper size is indicated in green at the top of the dialog, and the **Media Source** defaults to Automatic . In this example A4 and A3 are supported. See Section 4.8.2.3, "Media Sources" [134] on how media size is assigned to a printer's media sources.

- When a user selects Media Source Automatic, and the target printer supports automatic media source selection, SavaPage will use IPP `media-source` attribute value `auto` when sending the job to the printer. In this way, even when the job is redirected to a another (compatible) printer, the right media source will be selected automatically, based on media size.

> **Tip**
>
> If you always want to force a print job to automatic media source selection, irrespective from the **Media Source** choice, you can use a PPD Extension to map all `media-source` values to the same PPD `auto` value.

3.5.2.1. Page Scaling

When you choose a specific Media Source (holding a specific paper size) that does *not* match all paper sizes of SafePages jobs, a **Page Scaling** option appears, with an extra page size indicator (with light orange background) at

the top of the dialog. Also, a mnemonic text is displayed at the bottom of the Print Preview, with the media size of the chosen Media Source, optionally appended with the “fit” option (if Fit is selected).

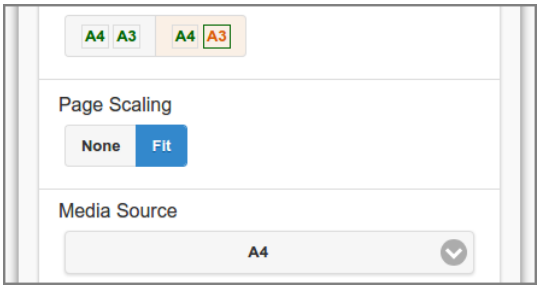


Figure 3.41. User Web App: Print - Page Scaling (Fit)

In this example, as the orange A3 indicator shows, the selection of the A4 media source does not fit available A3 page sizes.

The default option Fit (indicated as solid green square) will scale the deviant pages according to the print-scaling option “`fit`”.

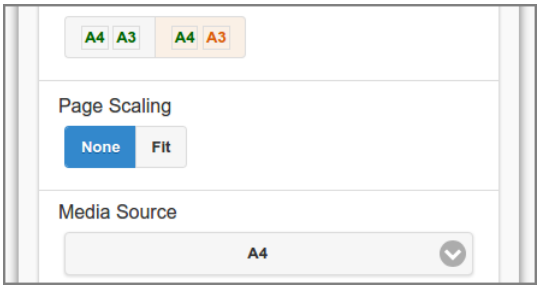


Figure 3.42. User Web App: Print - Page Scaling (None)

The None option (indicated as solid white square) will scale the deviant pages according to the print-scaling option “`none`”.

When printed media of document(s) *do* match the assigned media of the (implicitly) selected Media Source, the Page Scaling choice is not shown, and option “None” is applied.

Note

If the `print-scaling` option is restricted to `none`, the Fit option is not shown. See Section K.1.3.2, “print-scaling” [385].

Scaling defaults can be overruled with the configuration items shown in the table below:

Configuration property	Description
webapp.user.proxy-print.scaling.media-match.show	Set to `Y` or `N` (default) to show/hide Page Scaling choice when media sizes match assigned media of selected media source.
webapp.user.proxy-print.scaling.media-match.default	Default Page Scaling value when media sizes match assigned media of selected media source. Values: `AUTO` (default), `FIT` or `NONE`. Value `AUTO` applies a 'fit', 'fill' or 'none' method to get an optimized imposition of the document onto the printable area of the requested media size: this is the best method to scale documents with mixed portrait/landscape pages.

Configuration property	Description
webapp.user.proxy-print.scaling.media-clash.show	Set to `Y` (default) or `N`, to show/hide Page Scaling choice when media sizes do *not* match assigned media of selected media source.
webapp.user.proxy-print.scaling.media-clash.default	Default Page Scaling value when media sizes do *not* match assigned media of selected media source. Values: `NONE` or `FIT` (default).

Table 3.3. Page Scaling Configuration Properties

See Section 4.11.15, "Config Editor" [189] on how to set these items.

3.5.2.2. Print Preview

Print Preview helps the user to envision the effect of **Pages per side**, **Rotate by 180 degrees**, **Staple** and **Punch** on the Finished-Page.

A checkbox at the top right shows the orientation (Portrait, Landscape) of the first page of the selected document or page range. This is the right default in most cases, but the value can be toggled if the user knows for sure that the first page has a different orientation. For example, because he selects a specific page range from SafePages document(s) that have mixed page orientation. Whatever the choice is, SavaPage always checks if the assumption was right. When the actual print orientation turns out different as the one selected in the preview, and orientation dependent options, like staple and punch, were chosen, the user will be prompted to adapt the print settings and try again.

Figure 3.43. User Web App: Print - Preview (4-up, landscape, staple, punch)

> **Note**
>
> Print Preview can be enabled/disabled by setting `webapp.number-up-preview.enable` to `Y` (default) or `N`.

3.5.2.3. Custom Text and Icons

Option text can be customized, and choices can be preceded by a 16x16 pixel icon. See Section L.3, "IPP Localization" [404]. An example is shown below.

Figure 3.44. User Web App: Printer Options - Custom Icons

3.5.2.4. Printer Setting Options

The options presented in the Printer Settings dialog are a collection from:

- Basic IPP Printer Attributes as shown in the table below.
- org.savapage.int-page-rotate180
- Internal IPP - PPD Mapping Extensions.
- Internal IPP Job Ticket Extensions.

Attribute	Value source
media	CUPS (a subset of common media sizes is used).
media-source	CUPS or PPD Extension.
media-type	PPD Extension.
number-up	CUPS (values 1, 2, 4, 6, 9).
output-bin	CUPS or PPD Extension.
print-color-mode	CUPS or PPD Extension.
printer-resolution	CUPS.
print-scaling	Section K.1.1.7, “print-scaling” [384]
sides	CUPS or PPD Extension.

Table 3.4. Basic IPP Printer Attributes

Important

User input is validated with SPConstraint rules.

3.5.3. Selected Printer

The selected **Printer** is shown at the top section of the dialog.

Figure 3.45. User Web App: Print - Selected Printer

- The selected printer is shown as button at the bottom of the **Printer** section. Settings of the selected printer can be changed by pushing the button.
- The Page Size Indicator is shown at the top with symbols for the main printer options:
  - : color printing.
  - : black and white printing.
  - : duplex printing.
  - : two or more pages on one sheet.
- Another printer can be selected by reentering the search text (you can clear the quick search first my pushing the “cross” button at the right). *If just one printer is available no search option is offered.*

3.5.4. Delegated Print

The Delegated Print Invoicing option is offered in the Printer Settings and Print Job Settings dialog when the following conditions are met:

- Proxy Print Delegation is enabled.
- The User has Print Job Delegate role.

See Section 3.5.8, “Delegated Print Edit” [57].

3.5.5. Print Job Settings

Print job settings can be entered in the **Job** section.

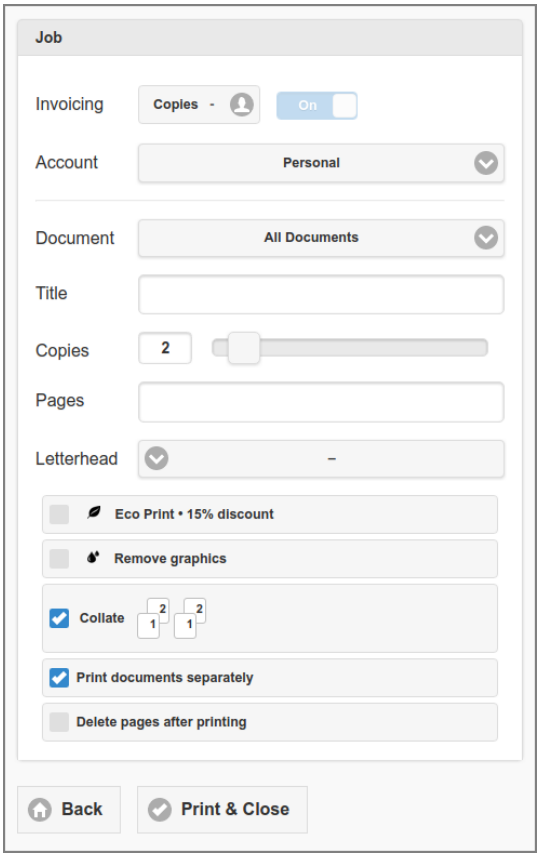


Figure 3.46. User Web App: Print - Job Settings

Select an **Account** to which the costs of the job is charged. The account can either be a Personal (default) or a Shared Account the user has access to. See Section 4.6.2, “Edit Account” [119].

- **Account** selection is *hidden* when Copies, collected by Delegated Print **Invoicing**, is switched `On`. Availability of this switch is dependent on configuration property webapp.user.proxy-print.delegate-copies-apply-switch.
- Account Personal may not be available due restricted Personal Print Privileges.
- When a Shared Account is selected the job will be handled as Delegated Print.

Note

When Job Ticket Labels are enabled and activated for this printer, additional labels like **Domain**, **Application** and **Tag** can be selected, just as in the Job Ticket Print dialog. Also see:

- Section 3.5.9.1, “Configuration Properties” [66]
- Section 4.8.2.1, “Proxy Printer Identity” [132].

Selecting a **Document** scope and **Title** for your print job is identical as in Section 3.4.3, “Document Scope” [42].

The number of **Copies** can be entered via the slider control. The “maximum number of copies per job” is set as Proxy Print Option. The number of copies is automatically reset to one (1) after printing.

- **Copies** selection is *hidden* when Delegated Print **Invoicing**, is switched `On`.
- When Delegated Print **Invoicing** is switched `Off`, and a Job Ticket Printer is selected, the number of **Copies** is not restricted and can thus be entered as text.

A selection of SafePages to print can be entered as a range of **Pages**. For example: `1-4,6,8-10,15-`. The value can be a single page, a range of pages, or a collection of page numbers and ranges separated by commas. The pages will always be printed in ascending order, regardless of the order of the pages in the page-ranges option. The page range is automatically emptied after printing. Be aware that the page ordinals are related to the **Scope**.

Selecting a **Letterhead** for your print job is identical as in Figure 3.35, “User Web App: PDF - Letterhead” [46].

- Depending on User Privileges the Letterhead section might not be shown.

Check one of the PDF Filters Eco Print or Remove graphics.

- The Eco Print, including discount percentage, must be enabled in Section 4.11.10.2, “Eco Print Settings” [172].
- See the Configuration Properties table below on how to disable the Remove graphics filter.

The Collate option is shown if multiple copies are selected and describes how printed material will be organized. For example, if you have a five page document and print multiple copies with collate checked, SavaPage prints pages 1,2,3,4 and 5 in that order and then repeats for each copy. However, if collate is unchecked and you print three copies of those same five pages, SavaPage prints pages in this order: 111, 222, 333, 444, 555. The icons in the checkbox are a mnemonic of the output when the collate option is (un)checked.

- If configuration property `webapp.user.proxy-print.collate.show` is `N`, the Collate checkbox is hidden and SavaPage acts as if it was checked.
- See Section K.1.1.9, “sheet-collate” [384] on how to specify the collate option as PPD Extension.

When the Print documents separately option is checked, each acquired document is printed as a separate job. In this way finishings (like stapling) can be applied per printed document.

- This option is *not* shown when the acquired documents are edited (pages have been deleted or rearranged) , or the **Document** scope is restricted to a single document .
- When this option is *not* checked, one (1) CUPS job is created. To separate documents on paper output, in case print is in duplex, blank filler pages are inserted in between, so the first page of a next document is on the front page of a new sheet. The same strategy is applied for single-sided n-up print jobs. However, for booklet printing this strategy is not applied.

When the option Delete pages after printing is checked, input documents or pages are deleted after the printing command is issued, and an extra option is displayed to select the deletion scope. The option is automatically reset after printing.

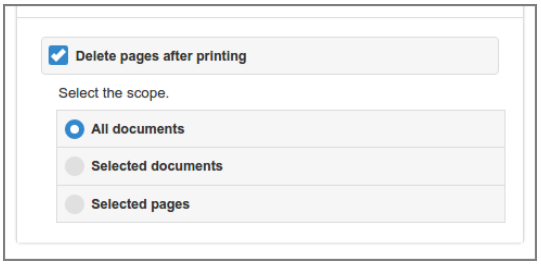


Figure 3.47. User Web App: Print - Delete Pages

- **All documents**: all input documents are deleted.
- **Selected documents**: documents for which pages were printed are deleted.
- **Selected pages**: all pages selected for printing are deleted.

The "Delete pages after printing" option can be preset and disabled for editing in Section 4.11.9, "Proxy Print" [168]. In that case a fixed text, denoting the action, is (optionally) displayed instead of the checkbox.

Press the Print & Close button to validate and print the job, and close the dialog.

Tip

If `webapp.user.proxy-print.confirm` configuration property is `Y` the Print ... button is shown to get confirmation about the costs.

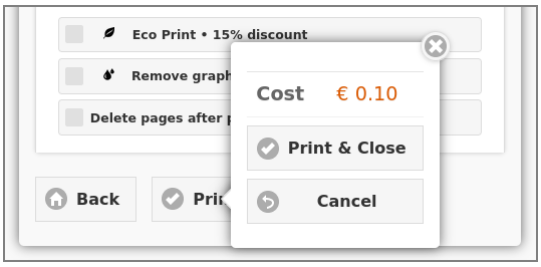


Figure 3.48. User Web App: Print Confirmation

When the job is valid, it is printed right away, *unless* it is a Job Ticket or when NFC Authentication is configured for the printer. In the later case the user must authenticate with an NFC card swipe to release the print job.

Important

When finishing options (staple, punch, booklet) are selected, and the orientation selected in the Print Preview differs from one of the documents to be printed, a warning message is shown telling the user to adjust print settings and try again. The message contains information about the selected finishings, number of deviating documents, and a tailored advise. If just one document is present (selected) for printing the advise is: *"Adjust your print settings and try again"*. For multiple documents: *"Clear the selection of finishing options, or print the documents separately"*.

Note

In rare situations, when the host system is under heavy load, and connecting to CUPS fails, a warning message is displayed saying *"The print service is currently unavailable or too busy. Please try again later."*

In the Hold Release scenario the job is held so it can be released by the user at a later time without using the User Web App. In the Direct Print Release scenario described below the user is prompted to authenticate immediately.

Pending Job Tickets and Hold Print Jobs can be inspected and removed when needed: see Section 3.3.2.3, "Hold Print Jobs" [33].

Warning

Job Tickets allow unrestricted printing, but printing to "regular" printers may be denied when the number of job pages exceeds a maximum. See the Proxy Print Options.

The following configuration properties apply:

Configuration property	Description
proxy-print.remove-graphics.enable	Set to Y or N (default), to show/hide the Remove graphics option.
webapp.user.proxy-print.clear-printer	If Y, the selected printer is cleared after each proxy print. If N (default), the selected printer and options are preserved, and will be used for the next print job.
webapp.user.proxy-print.clear-delegate	If Y, Delegated Print data is cleared after each proxy print. If N (default), this data is preserved, and will be used for the next print job.
webapp.user.proxy-print.collate.show	Set to Y (default) or N, to show/hide the Collate option. If hidden SavaPage acts as if the option was checked.
webapp.user.proxy-print.confirm	If N (default), the Print & Close button is displayed to send the print job immediately and close the dialog box. If Y, the Print ... button is displayed that opens a confirmation dialog box with printing costs and Print & Close and Cancel buttons.
webapp.user.proxy-print.separate	The default Print documents separately option value, when All Documents are selected. If Y, a separate print job is created for each vanilla inbox document. If N (default), one (1) print job is printed of the complete vanilla inbox content.
webapp.user.proxy-print.separate.enable	Enable the Print documents separately option. If Y, the option is enabled (shown), if N (default) it is not.

Table 3.5. Print Job Settings Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these items.

3.5.5.1. Print Archive

An extra option is offered when the Document Store for Print Archive is enabled, and the user is privileged to select the Archive. See Section 4.5.4.2, “User Privileges” [114].

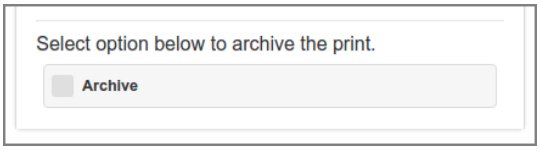


Figure 3.49. User Web App: Print Archive Option

When the user *has* Print Archive permission, but *no* permission to (de)select it, just a text is shown saying the “Print will be archived”. When explaining text is configured to be hidden, as described in the table below, in this case the printed document will silently be archived.

Configuration property	Description
webapp.user.doc.store.archive.out.print.prompt	Set to Y (default) or N, to show/hide a prompt explaining the Archive function.

Table 3.6. Print Job Archive Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these properties.

3.5.5.2. Print Journal

When Print Journal is enabled and applicable for Printer an User, the Print Job is silently journalled in the Document Store.

3.5.6. Direct Print Release

When a print job is issued for a printer secured with Direct Print Release, a dialog is shown prompting the user to swipe their card to release the print job.

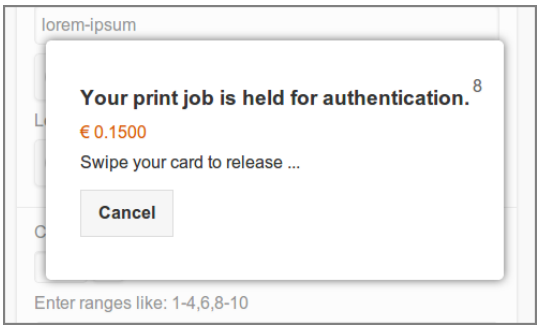


Figure 3.50. User Web App: Printer - Direct Print Release

- The cost of the print job is shown in orange.
- A countdown of the remaining seconds for the card swipe is shown in the top right corner of the pop-up. The time limit (seconds) is contained in configuration key `proxy-print.direct-expiry-secs`. See Section 4.11.15, “Config Editor” [189] on how to change this value.

> **Note**
>
> Since the card reader will be mounted near the printer this implements a *secure pull-print* scenario.

3.5.7. Full Print Scope and Jobs

When a user did *not* rearrange or delete any SafePages and *full* scope is selected without a range of **Pages**, each input job is printed as a separate job when *duplex* printing is selected. As a result the first page of new input job will always start on a new sheet. When the **Title** is left blank, the titles of the print jobs will correspond to the titles of the input jobs. When a **Title** is specified it will be used for *all* print jobs.

When a user *did* rearrange or delete any SafePages the scope is confined to *full* scope and SafePages will always be printed as a single job. When the **Title** is left blank the print job title will be generated according to the timestamp format `SavaPage-CCYY-MM-DDTHH:MM:SS`.

3.5.8. Delegated Print Edit

This dialog is part of the Invoicing track, and is started from Printer Settings or Print Job Settings. It is used to edit Delegated Print.

In a Delegated Print scenario the user prints as *delegate* on behalf of other users, known as *delegators*. The result is a single job with multiple copies of a document. The total cost of the job is *pro rata* charged to the account of each delegator.

*The **Copies** section at the bottom of the dialog summarizes the selected delegators from the **Invoicing** section at the top.*

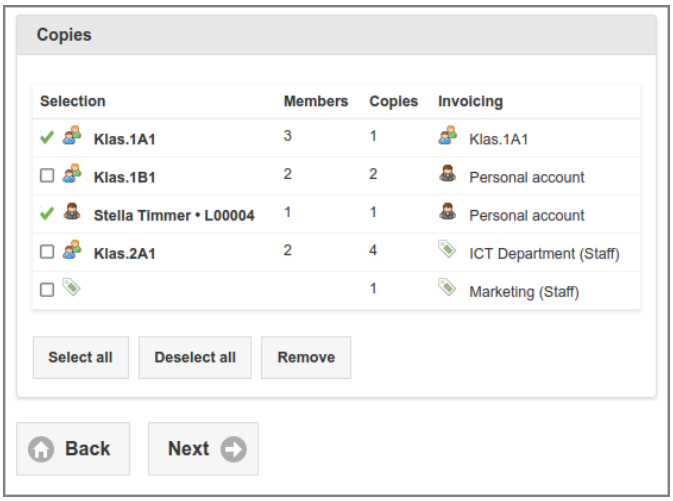


Figure 3.51. User Web App: Print Delegation Copies

- **Selection** type is marked with one of these icons:
  - : User Group.
  - : Individual User.
  - : Shared Account.
- Each selection line shows the number of **Members** and **Copies**. The latter is the multiplication of members and the number of copies entered in the **Invoicing** section.
  - All *User Group* members are included, *whatever their role*.
  - *Individual Users* of course have just one member and have Print Job Delegator role.
  - *Shared Account* counts unspecified users. Just the number of copies is shown.
- The **Account** column shows where printing costs are charged to:
  - : Group Account.
  - : Personal Account of Group **Members**.
  - : Shared Account.

*The screenshots from the **Invoicing** section, presented in the sections below, explain how the lines in the **Copies** section were created.*

Remember that for items to appear on the various search list:

- **Groups** must have Print Job Delegator role *enabled*.
- **Users** must have Print Job Delegator role *enabled*.
- **Accounts** must be accessible by the Delegate as member of one of the User Groups set in Account Access Control.
  - *When a Delegate has access to a single Shared Account only, the Account search list is **not** displayed, and the single account is implicitly selected.*

Availability of the option to enter multiple number of copies is dependent on configuration property proxy-print.delegate.multiple-member-copies.enable.

Tip

Delegated Print can be integrated with PaperCut. See Section N.1, “Delegated Print to PaperCut” [412].

Caution

After printing, information about copies by group/user/account is kept in the financial data if the cost turns out to be greater than zero. See Section 4.12.1, “Document List” [191]. If the printing cost is zero, all this information is lost after printing.

3.5.8.1. Group Invoicing

Select `Group` *to invoice copies for* Groups *on Group Account.*

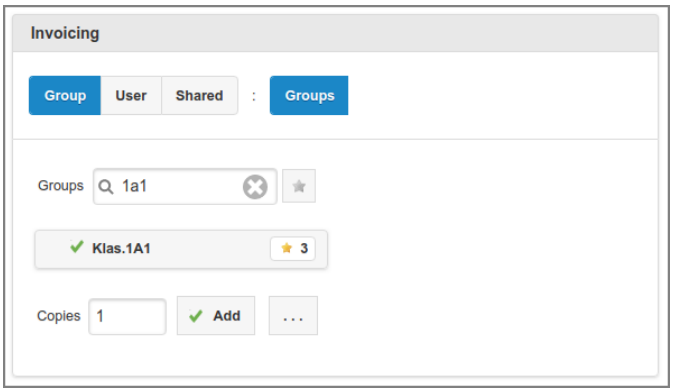


Figure 3.52. User Web App: Delegated Print - Group Invoicing

Availability of this option is dependent on configuration property proxy-print.delegate.account.group.enable.

- Enter a part of the group name in the **Groups** quick search field and the first chunk with matching groups will show below. Paging buttons are shown when more chunks are available.
  - The list shows the Group Name with the ID appended (when it is not already part of the Name).
  - Group quick search is on both Group Name (sorted) and ID, unless configuration item webap-p.user.proxy-print.delegator-group.detail has value `NAME`, so display and search is on Group Name (sorted) only , or has value `ID` , so display and search is on Group ID (sorted) only.
- Note the star button at the left of the quick search field, and at the right of each list entry. Press the star button in a list entry to bookmark a group as “preferred”. Press the star button besides the quick search field to search for “preferred” groups only. This feature is enabled when configuration item proxy-print.delegate.groups.preferred.enable is Y.
- Select one or more **Groups** from the list by tapping them (a green check mark sign will show at the front).
- Enter the number of **Copies** to be printed for each group member, or press the `. . .` button to enter the total number of Group copies.
- Press the `Add` button to add the groups as delegator.
- *The Group Account will be invoiced.*

When configuration item proxy-print.delegate.group-copies.enable is Y, the `. . .` button is visible. When pressed, a pop-up shows where the number of copies for the selected group(s) can be entered. This feature can for instance be used when a group of 20 need only 10 copies, because group members work in pairs.

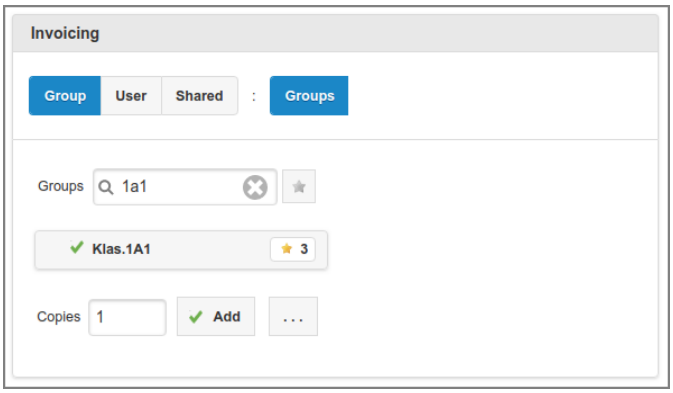


Figure 3.53. User Web App: Delegated Print - Group copies

3.5.8.2. User Invoicing for Groups

Select User *to invoice copies for* Groups *on User Account.*

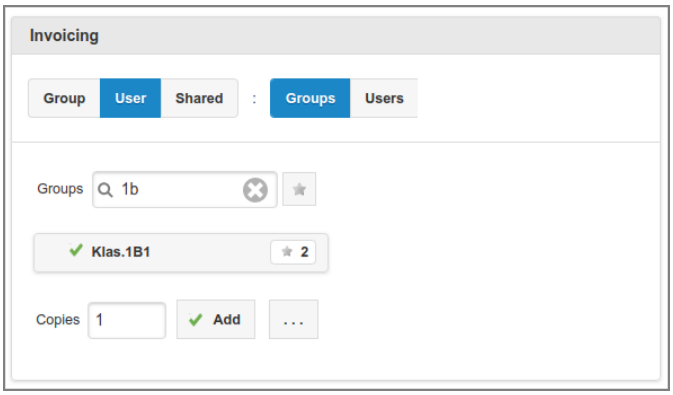


Figure 3.54. User Web App: Delegated Print - User Invoicing (Groups)

- Select one or more **Groups** from the list.
  - See remarks about Group selection and display in previous section.
- Enter the number of **Copies** to be printed for each group member.
- Press the Add button to add the group members as delegator.
- *The Personal Account of each individual group member will be invoiced.*

3.5.8.3. User Invoicing for Users

Select User *to invoice copies for Individual* Users *on User Account.*

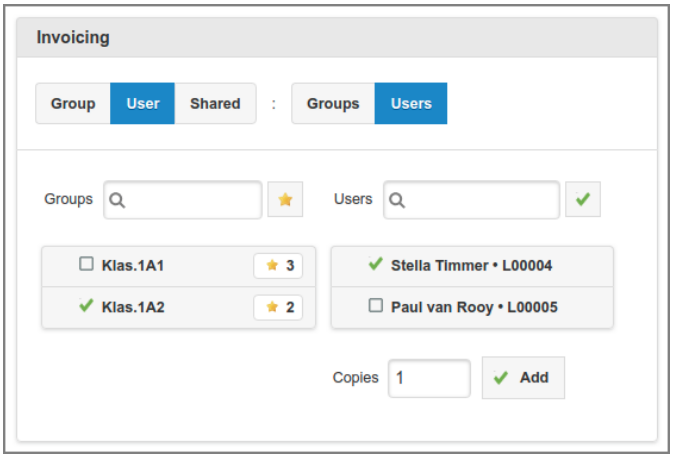


Figure 3.55. User Web App: Delegated Print - Personal Invoicing (Users)

Availability of this option is dependent on configuration property proxy-print.delegate.account.user.enable.

- Select one of the **Groups** to narrow down the search scope. When *no* group is selected the All Users built-in group is implied.
  - See remarks about Group selection and display in previous section.
- Search and select one or more **Users** from the list.
  - The list shows the User Name with the ID appended (when it is not already part of the Name).
  - User quick search is on both User Name (sorted) and ID, unless configuration item webapp.user.proxy-print.delegator-user.detail has value `NAME`, so display and search is on User Name (sorted) only , or has value `ID` , so display and search is on User ID (sorted) only.
  - Users can automatically be (un)selected by using the “Select All” toggle at the right-hand side of the search field.
- Enter the number of **Copies** to be printed for the selected users, and press the Add button to add them as delegator.
- *The Personal Account of the selected users will be invoiced.*

> **Note**
>
> The session user is added in the User List, even when this user does not have Print Delegator role. The session user's Name and ID is highlighted with an orange color.

3.5.8.4. Shared Invoicing for Groups

Select Shared to invoice copies for Groups on Shared Account.

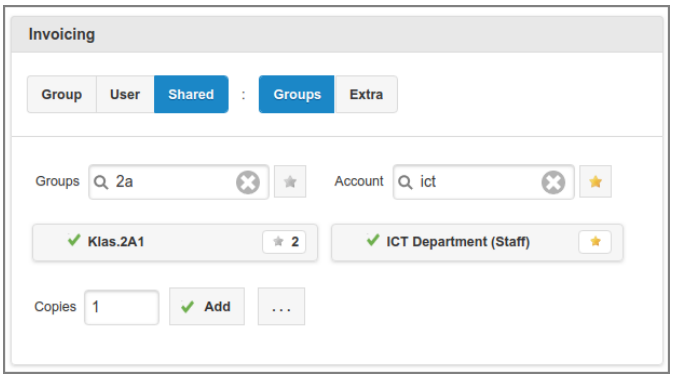


Figure 3.56. User Web App: Delegated Print - Shared Account Invoicing (Groups)

Availability of the `Shared` button is dependent on configuration property proxy-print.delegate.account.shared.enable and on Shared Account Access Control. When the Delegate user does not have access to any shared account by User Group membership, the button will not show.

Availability of the `Groups` option is dependent on proxy-print.delegate.account.shared.group.enable.

- Select one or more **Groups** from the list.
  - See remarks about Group selection and display in previous section.
- Select a Shared **Account**.
  - *Shared Account selection is hidden if just a single account is available.*
  - Note the star button at the left of the quick search field, and at the right of each list entry. Press the star button in a list entry to bookmark a group as “preferred”. Press the star button besides the quick search field to search for “preferred” accounts only. This feature is enabled when configuration item proxy-print.delegate.accounts.preferred.enable is Y.
- Enter the number of **Copies** to be printed, and press the `Add` button.
- *The Shared Account will be invoiced.*

3.5.8.5. Shared Invoicing for Extra Copies

Select `Shared` to invoice Extra copies on Shared Account.

Availability of the `Shared` button is explained in the previous section.

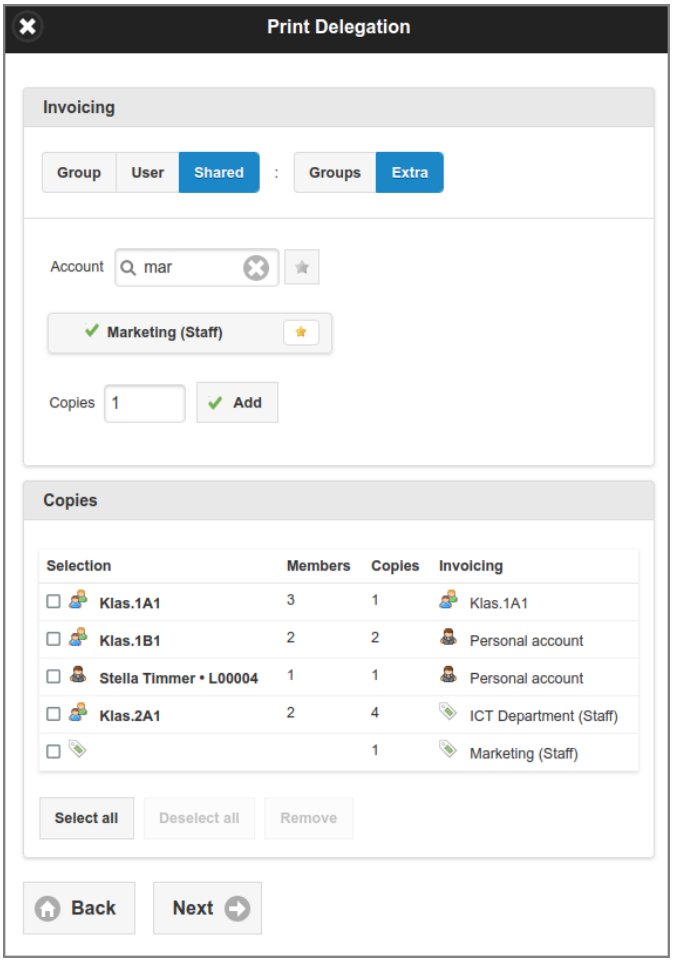


Figure 3.57. User Web App: Print - Delegated Print - Shared Account Invoicing (Extra)

Availability of this option is dependent on configuration property proxy-print.delegate.account.shared.enable.

- Select a Shared **Account**.
  - See remarks in previous section.
- Enter the number of **Copies** to be printed, and press the Add button.
- *The Shared Account will be invoiced.*

As this is the last Invoicing Selection of our example, the complete Delegated Print dialog is shown in the screenshot.

Note

Once Users and Groups are selected as Delegate they are disabled in the quick search list so they can't be selected a second time.

3.5.8.6. Configuration Properties

The following Delegated Print configuration properties apply:

Configuration property	Description
proxy-print.delegate.account.group.enable	Set to Y (default) or N, to enable/disable invoicing with Group Account.
proxy-print.delegate.account.shared.enable	Set to Y (default) or N, to enable/disable invoicing with Shared Account.
proxy-print.delegate.account.shared.group.enable	Set to Y (default) or N, to enable/disable invoicing with Shared Account for Groups copies.
proxy-print.delegate.account.user.enable	Set to Y (default) or N, to enable/disable invoicing with User Account.
proxy-print.delegate.multiple-member-copies.enable	Set to Y (default) or N, to enable/disable multiple copies for Groups and Users. When disabled, the number of copies is 1.
proxy-print.delegate.group-copies.enable	Set to Y or N (default), to enable/disable option to assign (overrule) the number of copies for a group. When enabled, an extra ... button is displayed in the Delegated Print Dialog, at the right of the Add button, that leads to a pop-up where the number of copies for the selected groups can be entered.
proxy-print.delegate.groups.preferred.enable	Set to Y (default) or N, to enable/disable option to bookmark delegate groups as preferred. When enabled, an extra “star” button is displayed at the left of the quick group search field, and at the right of each list entry. When the star button in a list entry is pressed, a group is bookmarked as “preferred”. When the star button besides the quick search field is pressed, only “preferred” groups are searched.
proxy-print.delegate.accounts.preferred.enable	Set to Y (default) or N, to enable/disable option to bookmark shared accounts as preferred. When enabled, an extra “star” button is displayed at the left of the quick account search field, and at the right of each list entry. When the star button in a list entry is pressed, an account is bookmarked as “preferred”. When the star button besides the quick search field is pressed, “only preferred” accounts are searched.
webapp.user.proxy-print.delegate-copies-apply-switch	Set to Y (default) or N, to show/hide an On/Off switch to apply Delegated Print Copies in the User Web App Print Dialog. When N, Delegated Print print Invoicing is active when the sum of collected delegator **Copies** is greater than zero.

Configuration property	Description
webapp.user.proxy-print.delegator-user.detail	Set to `FULL` (default), `ID` or `NAME`, to show FULL details of User in Delegated Print Dialog, or just its ID or NAME.
webapp.user.proxy-print.delegator-group.detail	Set to `FULL` (default), `ID` or `NAME`, to show FULL details of User Group in Delegated Print Dialog, or just its ID or NAME.

Table 3.7. Delegated Print Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these items.

3.5.9. Job Ticket Print

When Job Ticket Printer instances are present, users with role Job Ticket Creator can select the marked printers.

Figure 3.58. User Web App: Print - Select Job Ticket Printer

By default, Job Tickets are of Type `Print`.

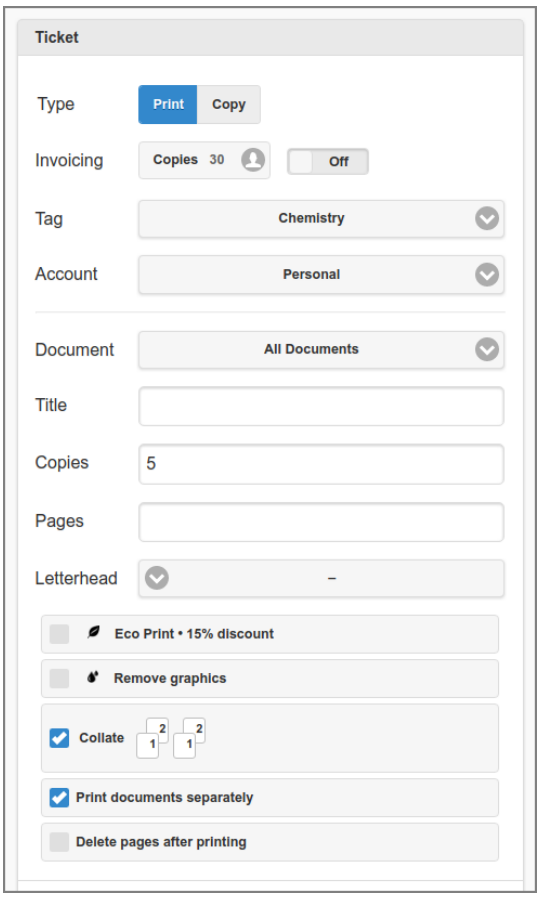


Figure 3.59. User Web App: Print - Job Ticket Settings - Print

All options for a regular print job apply.

The **Tag** item is a label for Job Tickets. Additional **Domain** and **Application** (use) labels can be configured, as shown in screenshot below. See the next section on how to configure these items.

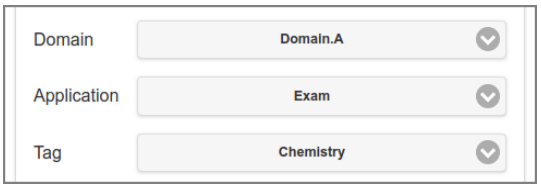


Figure 3.60. User Web App: Print - Job Ticket Settings - Labels

Note

The last selected **Domain** label is saved, and used as default in future user session. You can disable this behavior by setting configuration property `jobticket.domains.retain` to `N`. See Section 3.5.9.1, "Configuration Properties" [66].

Note

When the **Title** of the Ticket is *not* specified, SavaPage will compose one, based on the **Document** title(s). In this way extra identifying data will be available.

When a single document is selected the **Title** will default to the name of this document. When the **Title** is left empty, SavaPage will compose one, based on the **Document** name(s). In this way identifying data will be available.

When configured as such, Job Tickets can also be used for Copy requests. All printer settings of a regular proxy print apply, as well as some job options. However, SafePages are not needed, and no PDF document will be attached to the ticket.

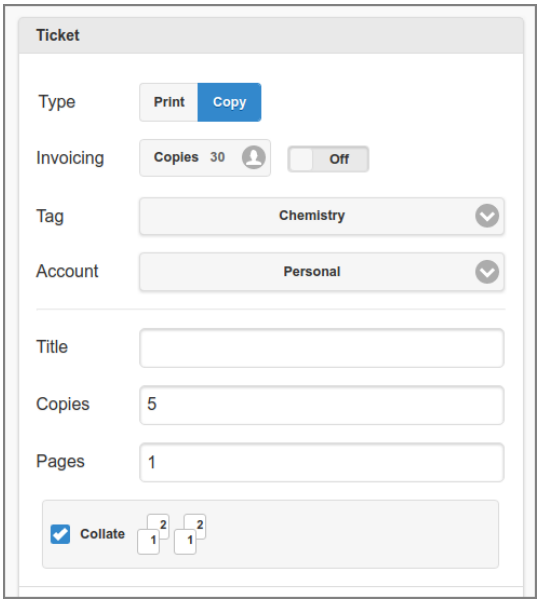


Figure 3.61. User Web App: Print - Job Ticket Settings - Copy

Users can enter a **Time of delivery** and **Remarks**. Valid delivery days are displayed at the right. Press the reset icon on the left to reset the time to the default.

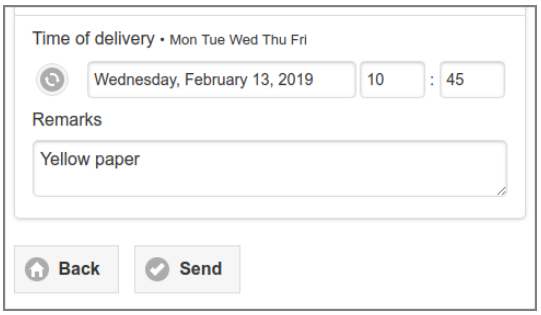


Figure 3.62. User Web App: Print - Job Ticket Settings

Note

The default **Time of delivery** and its reset button are *not* displayed if configuration property jobticket.delivery-days is 0 (zero).

After the ticket is sent, the issued Ticket Number is displayed, and the user can view the Job Ticket on the Hold Print Jobs list.

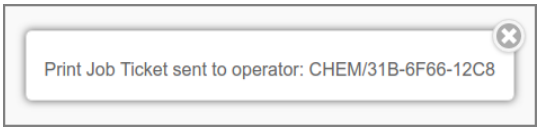


Figure 3.63. User Web App: Print - Job Ticket - Sent

In case of a Copy Job, the user will add a note with the Ticket Number to the hard copy original, before handing it to the Job Ticket operator, who will use it for off-the-glass copying according to the specs in the job ticket.

A Job Ticket is printed on a central queue and is handled and released by users with role Print Job Operator in a special Web App. See Chapter 5, *Job Tickets Web App* [211]. When the Job Ticket is printed or canceled, the user is optionally notified by email.

The following configuration properties apply:

3.5.9.1. Configuration Properties

Configuration property	Description
Delivery	
jobticket.delivery-datetime.enable	Set to `Y` (default) or `N`, to show/hide the **Time of delivery** option. When date/time is hidden its value is set by the system to the date/time of the submit.
jobticket.delivery-time.enable	Set to `Y` (default) or `N`, to show/hide the time part of **Time of delivery**. This option is valid when "jobticket.delivery-datetime.enable" is `Y`.
jobticket.delivery-days	Default value of the date part of **Time of delivery** as days-of-week count after ticket creation date (default = 1). If value is 0 (zero), the date part default and its reset button are not displayed.
jobticket.delivery-days-min	Minimal value of the date part of **Time of delivery** as days-of-week count after ticket creation (default = 1).
jobticket.delivery-days-of-week	CRON expression with valid delivery dates as days-of-week range (default: MON-FRI).

Configuration property	Description
jobticket.delivery-day-minutes	Default value of the time part of **Time of delivery** on delivery days as minutes after midnight (default: 510). For instance: 8h30m = 8*60+30 = 510.
Copy Job	
jobticket.copier.enable	Set to `Y` or `N` (default), to enable/disable the creation of a Copy Job Ticket.
Email notification	
jobticket.notify-email.completed.enable	See Section 5.4, “Job Ticket Configuration Properties” [222].
jobticket.notify-email.canceled.enable	See Section 5.4, “Job Ticket Configuration Properties” [222].
Domain labels	
jobticket.domains	A comma separated list of Job Ticket domains to be applied as job ticket number prefix. Each tag on the list is formatted as "`DOM/word`", where "`DOM`" is a unique N-letter upper-case mnemonic, "`/`" is a fixed separator, and "`word`" is a case-sensitive single word used in UI context. E.g. "`"A/Domain.A,B/Domain.B,C/Domain.C`". When "`A`" domain is applied, a generated ticket number looks like "`A/EE1-FA3E-6596`".
jobticket.domains.enable	Set to `Y` or `N` (default) to enable `jobticket.domains`. When *enabled*, Job Ticket domains can optionally also be used for regular printers, i.e. printers that do not have Job Ticket Printer role. See Section 4.8.2.1, “Proxy Printer Identity” [132].
jobticket.domains.required	Set to `Y` (default) or `N`, to make usage of a Job Ticket Domain compulsory or optional.
jobticket.domains.retain	Set to `Y` (default) or `N`, to retain the last selected Domain label and use it as default in a next Job Ticket (this also works for a new User Web App session).
Application labels	
jobticket.uses	A comma separated list of Job Ticket applications (uses) to be applied as job ticket number prefix. Each tag on the list is formatted as "`USE/word`", where "`USE`" is a unique N-letter upper-case mnemonic, "`/`" is a fixed separator, and "`word`" is a case-sensitive single word used in UI context. E.g. "`E/Exam,T/Test`". When "`E`" use is applied in domain "`A`", a generated ticket number looks like "`A/E/EE1-FA3E-6596`". A use item can be restricted for use with one or more domains by appending the domain mnemonics. E.g. "`E/Exam/A/C`" restricts the "`E/Exam`" use to domains "`A`" and "`C`".
jobticket.uses.enable	Set to `Y` or `N` (default) to enable `jobticket.uses`. When *enabled*, Job Ticket uses can optionally also be used for regular printers, i.e. printers that do not have Job Ticket Printer role. See Section 4.8.2.1, “Proxy Printer Identity” [132].
jobticket.uses.required	Set to `Y` (default) or `N`, to make usage of a Job Ticket Use compulsory or optional.
Tag labels	
jobticket.tags	A comma separated list of Job Ticket tags to be applied as job ticket number prefix. Each tag on the list is formatted as "`TAG/word`", where "`TAG`" is a unique N-letter upper-case mnemonic, "`/`" is a fixed separator, and "`word`" is a case-sensitive single word used in UI context. E.g. "`MATH/Maths,PHYS/Physics,CHEM/Chemistry`".

Configuration property	Description
	When "CHEM" tag is applied for use "E" in domain "A", a generated ticket number looks like "A/E/CHEM/EE1-FA3E-6596". A tag item can be restricted for use with one or more domains by appending the domain mnemonics. E.g. "PHYS/Physics/A/C" restricts the "PHYS/Physics" tag to domains "A" and "C".
jobticket.tags.1	An extension of configuration property `jobticket.tags`
jobticket.tags.enable	Set to Y or N (default) to enable `jobticket.tags`. When *enabled*, Job Ticket tags can optionally also be used for regular printers, i.e. printers that do not have Job Ticket Printer role. See Section 4.8.2.1, "Proxy Printer Identity" [132].
jobticket.tags.required	Set to Y (default) or N, to make usage of a Job Ticket Tag compulsory or optional.

Table 3.8. Job Ticket Print Configuration Properties

See Section 4.11.15, "Config Editor" [189] on how to set these items.

3.6. Letterheads

A *tap* on the Letterhead button in the main SafePages view shows the Letterhead dialog. See Section 3.3, "SafePages" [29].

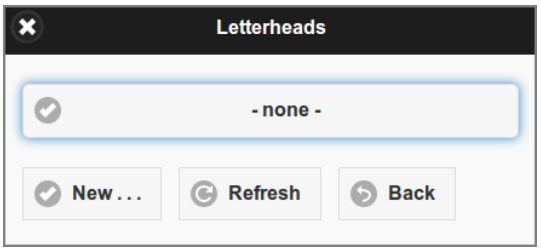


Figure 3.64. User Web App: Letterheads

- Press the top button to get a pop-up list of letterheads. When a letterhead is selected from the list it can be previewed and edited. The selected letterhead acts as default in the Print and PDF dialog: see Figure 3.35, "User Web App: PDF - Letterhead" [46] and Figure 3.37, "User Web App: Print - Select Printer" [47].
- Press the Refresh button to rebuild the list if you suspect public letterheads were added or removed.
- Press the New button to create a new letterhead from the current SafePages. All the SafePages are used for the letterhead. Figure 3.65, "User Web App: Letterhead - New" [69] is an example of a fresh created letterhead.

Note

Depending on User Privileges buttons and sections for creating and editing letterheads might not be shown.

Note

If the SafePages contain DRM-restricted content the New button is not visible.

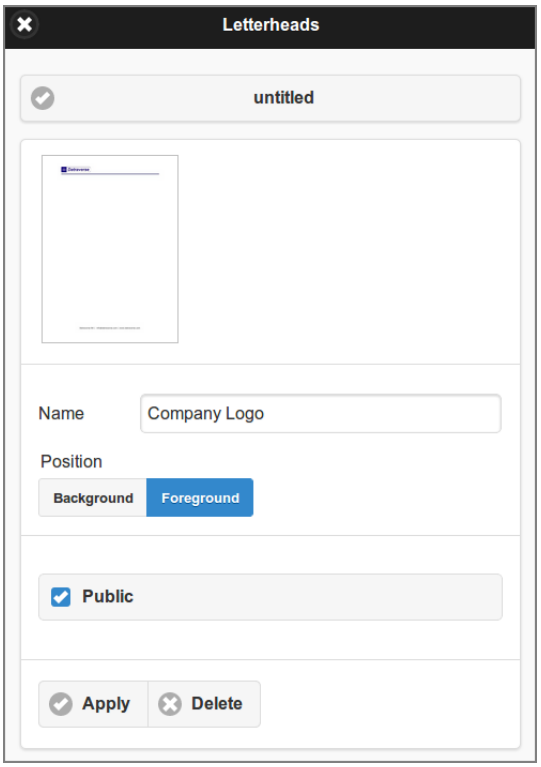


Figure 3.65. User Web App: Letterhead - New

- If the letterhead document consists of one page, this page is applied to every page of the document. The letterhead page is scaled and rotated as needed to fit the input page.
- If the letterhead document has more than one page, each page of the letterhead is applied to the corresponding page of the output document. If the output document has more pages than the letterhead, then the final letterhead page is repeated across these remaining pages of the output document.
- Letterheads can be applied as foreground or background.
- Users who are administrator can mark letterheads as *public* by ticking the "Public" checkbox. Public letterheads are available for all users, but can be edited and deleted by administrators only.
- The title of a private letterhead can be edited. The title of a public letterhead can be edited by an administrator only.
- Tap on a letterhead thumbnail to get a detailed pop-up image. See Figure 3.66, “User Web App: Letterhead - Detail” [70].
- Press the Apply button to commit changes or push the Delete button to remove the letterhead.

Note

For a background letterhead to be effective, SafePages should be transparent. HTML pages printed from browsers like Internet Explorer and Firefox might pose a problem here. The white background on HTML pages might act as a solid background.

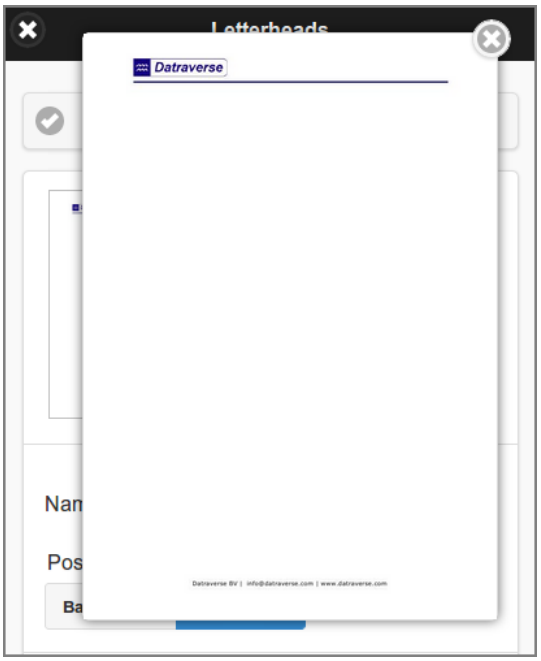


Figure 3.66. User Web App: Letterhead - Detail

- Press the x button in the upper right corner, or *tap* outside the pop-up, to close the pop-up image.

3.7. Delete

A *tap* on the Delete button in the main SafePages view shows the Delete dialog. See Section 3.3, “SafePages” [29].

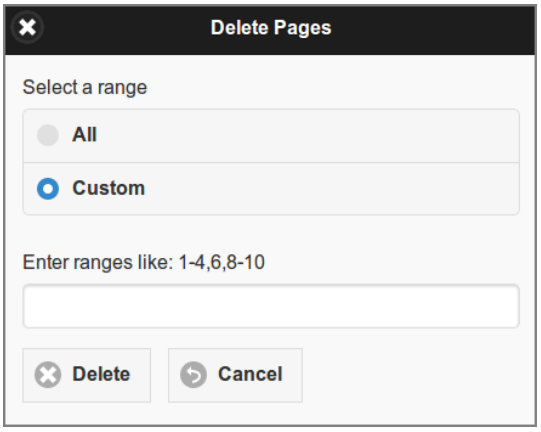


Figure 3.67. User Web App: Delete SafePages

- Select a range of SafePages to delete. Press the All button to select all pages. Or, push the Custom button to enter a custom range of SafePages: the value can be a single page, a range of pages, or a collection of page numbers and ranges separated by commas.
- Tap the Delete button to perform the delete action.

Caution

After SafePages are deleted any Fast Print will fail and all SafePages will be cleared as a result. Please use a Hold Print instead.

Tip

You can delete all SafePages of a SavaPage print job in the Document Details dialog, as described in Section 3.3.3.1, “Delete and Undo” [36].

3.8. Log

A *tap* on the Log button in the main SafePages view shows the Document Log. See Section 3.3, “SafePages” [29].

3.8.1. Documents

The Document Log shows all documents the user printed to SavaPage, and which were subsequently exported or Proxy-Printed. For a detailed description of this screen see Section 4.12, “Documents” [190] in the Admin Web App chapter.

Some Document Type selections are *hidden*, depending on user role, privileges and resource (Queue, Printer) availability.

- Type PDF is hidden when user has no privileges to Send and Download SafePages.
- Type Ticket is hidden when user is not a Job Ticket Creator.
- Type Print is hidden when user is not a Print Job Creator, or has no access to a printer.

The following configuration property applies:

Configuration property	Description
webapp.user.doclog.select.type.default-order	A comma separated priority list of enum values `ALL`, `IN`, `OUT`, `PDF`, `PRINT`, `TICKET`, that correspond to Document Type selection options. When this list is empty, `ALL` is assumed. The default Type selection is the first value from the list that corresponds to a visible select option. To comply to EU GDPR ("Right for a data subject to receive personal data concerning them"), `IN` and `OUT` types are always available, so full history can be queried regardless of current status.

Table 3.9. Document Type Select: Configuration Property

See Section 4.11.15, “Config Editor” [189] on how to set these items.

The Letterhead selection is hidden when user has no Letterhead privileges. The Queue selection does not hold disabled or inactive (reserved) queues (/mailprint, /webprint). The Printer selection holds accessible Printers only (Job Ticket "Printers" are not shown).

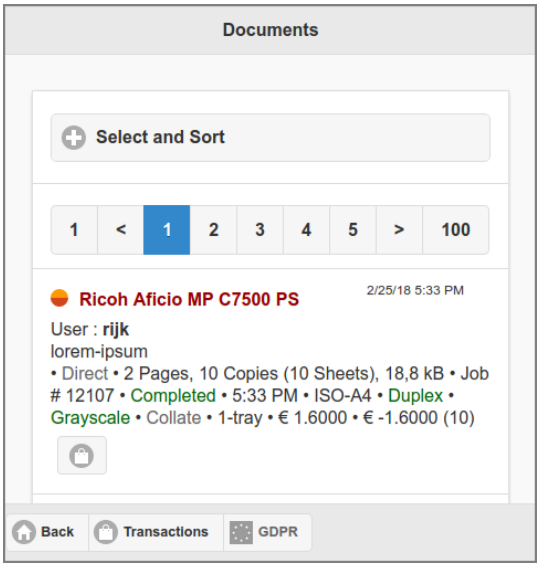


Figure 3.68. User Web App: Log - Documents

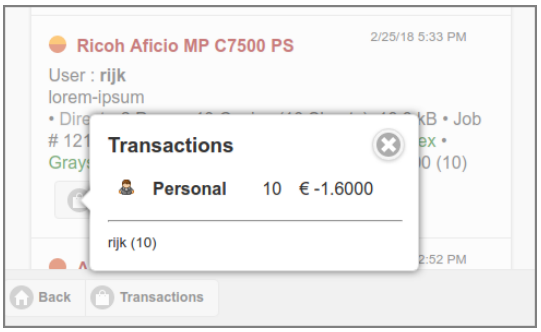


Figure 3.69. User Web App: Log - Document Transactions

Press the transaction button in the document item to see invoicing details. Press the Transactions button in the footer bar to view the Transaction List, and the GDPR button to view the GDPR Dialog.

Note

Depending on User Privileges the Transactions button might not be present.

3.8.1.1. Queue Journal

If the **Queue Journal** function is enabled a print-in document is journaled, as indicated by a small icon at the left of the document title. See Section 4.11.14.10, “Document Store” [185] and Section 4.7.2, “Edit Queue” [123].

Figure 3.70. User Web App: Log - Queue Journal

Depending on User Privileges buttons are displayed to Download, Select (add to or replace the SafePages) and Delete the journal.

Note

When viewed in the Admin Web App the journal can *not* be Selected.

3.8.2. Transactions

The Transaction Log shows the financial transactions on the user's account.

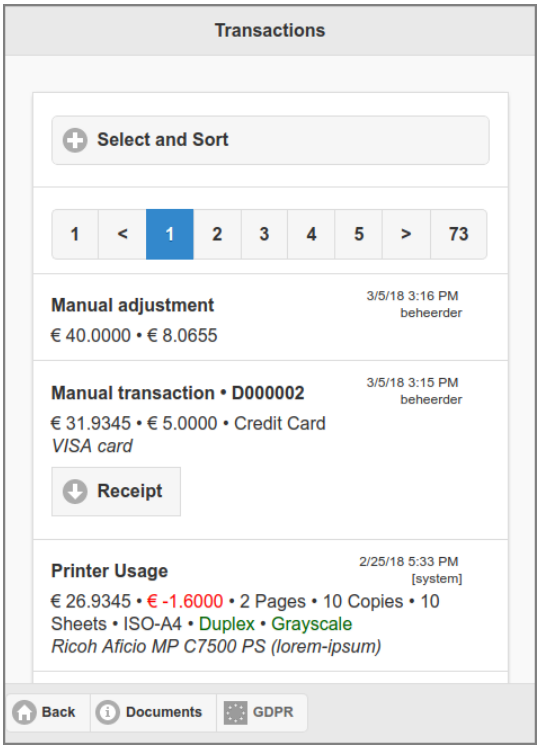


Figure 3.71. User Web App: Log - Transactions

Each entry in the list has the following lines.

- A header with the transaction type.

- The resulting account balance with the transaction amount.
- Extra information is added depending on the transaction type.
- A Manual Transaction denotes the Receipt Number in the header and has a Receipt button to download the receipt as PDF.

Some samples of other transactions:

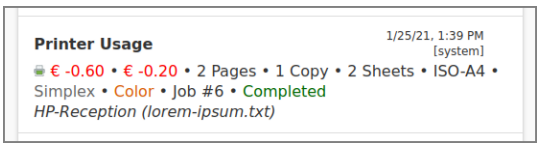


Figure 3.72. User Web App: Log - SavaPage User Account Transaction

- If PaperCut Personal User Account is enabled, the amount is prefixed with a SavaPage icon.

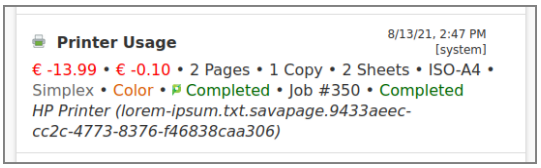


Figure 3.73. User Web App: Log - Proxy Print PaperCut Transaction

- A Proxy Print to a PaperCut managed printer where PaperCut Printer Costs were leading.
- Note the SavaPage (print supplier) and PaperCut (third-party print manager) icons.
- For a sample screenshot of the document log, see Figure 4.150, “Admin Web App: Document - Proxy Print PaperCut Costs” [191].

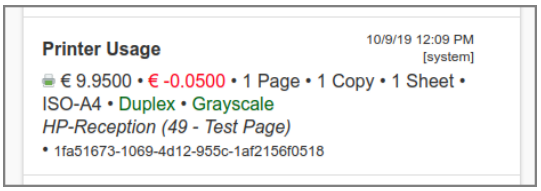


Figure 3.74. User Web App: Log - IPP Routing Transaction

A Printer Usage transaction due to IPP Routing, with QR code “1fa51673-...” delivered by an IPP Routing Plug-in.

- Note that in this case PaperCut Personal User Account is enabled, since the amount is prefixed with a SavaPage icon.

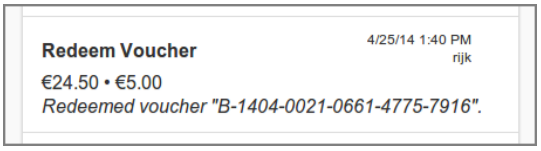


Figure 3.75. User Web App: Log - Voucher Transaction

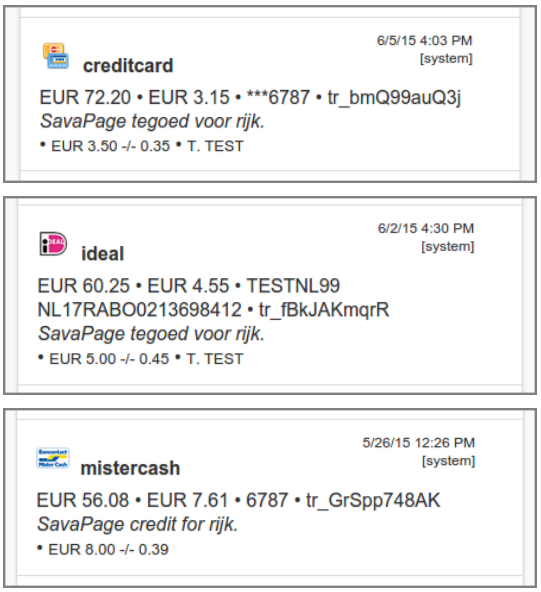


Figure 3.76. User Web App: Log - Payment Gateway Transactions

Figure 3.77. User Web App: Log - User Transfer Transactions

Push the Documents button in the footer bar to view the Document List, and the GDPR button to view the GDPR Dialog.

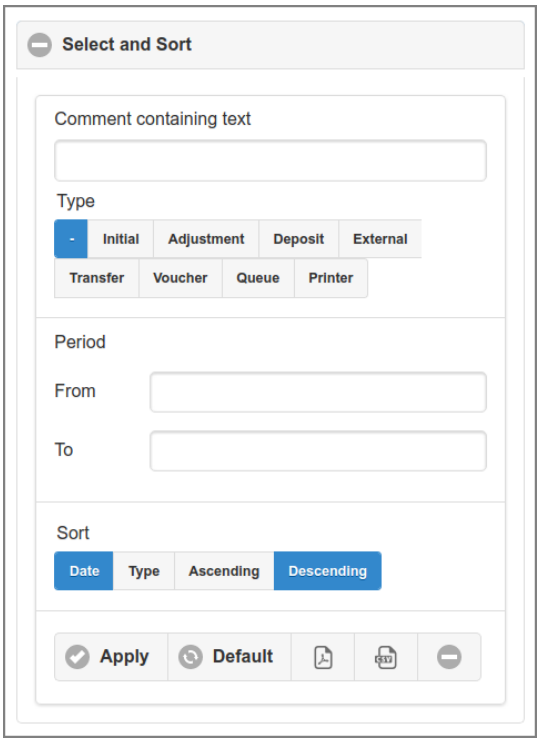


Figure 3.78. User Web App: Log - Transactions

Transactions can be filtered and sorted as follows:

- **Comment containing text**: selects transactions with comments containing a text snippet.
- **Type**: selects **–** (all) or a single one of the transaction types...
  - **Initial**: Balance allocated when account was created.
  - **Adjustment**: Manual adjustment by an administrator. See Section 4.4.5.8, “Financial” [106].
  - **Deposit**: Adjustment of balance at a Point-of-Sale.
  - **External**: Increment of balance by transferring funds from an external account. See Section 3.10.8, “Transfer Money” [85].
  - **Transfer**: Increment or decrement of balance by transferring credit to another user. See Section 3.10.7, “Transfer Credit” [84].
  - **Voucher**: Increment of balance by redeeming a voucher.
  - **Queue**: A delegator transaction as part of a Delegated Print where the document and delegate info is retrieved from an *External Supplier* and printed to a Proxy Printer managed by a *Third Party Print Management System* (TPPMS). The transaction appears as **Queue Usage** item in the list. Print status and cost are retrieved from the TPPMS. The cost will be 0.00 when the status is “Pending (external)”, “Expired” or “Canceled”. When status is “Completed” the cost will be known. Currently External Supplier TPPMS PaperCut is supported.
  - **Printer**: A transaction for proxy printing. When the Proxy Printer is solely managed by SavaPage the costs are according to the specified Printer Costs. When the Proxy Printer is additionally managed by a *Third Party Print Management System*, status and cost are retrieved from that system and displayed just like the previously discussed **Queue Usage** item.
- Select a creation **Period** by entering a `From` and `To` date. Tap the `x` button after a date to clear it. See this example Data Selection Dialog.
- Transactions can be sorted `Ascending` or `Descending` by creation `Date` or `Type`.
- The list is refreshed, and the selection applied, after you push the `Apply` button.
- The `Default` button resets the selection items to their default values.

- The PDF and CSV buttons downloads the Transaction List in their respective formats, using the selection item values.
  - See Section C.2.3.2.9, “POST /reports/accounttrx” [350] on how to download these lists with the RESTful Admin Service.
- The minus icon collapses the Select and Sort section.

3.9. Sort

A *tap* on the Sort button in the main SafePages view switches to Sort mode. See Section 3.3, “SafePages” [29].

In sort mode pages can be rearranged and deleted.

Figure 3.79. User Web App: Sort

- This screen shows the result after some editing.
- One *cut* page is shown with a red border.
- Notice the mini scissor icon at the bottom of the screen, showing the page number of the cut page.
- One *selected* page is shown with an orange border.

- The footer bar shows a scissor icon with a page number of the cut, and a pin icon with the page number of the selected page.

These are the actions that can be performed on page images:

- A *tap* on a *single* page will (un)select it.
- A *tap* on an *aggregated* page will expand it. Aggregated pages are described at Figure 3.17, “User Web App: SafePages - Aggregated” [31].

Here is a summary of the buttons and their function:

More actions : shows a pop-up with more actions (see below).

More actions pop-up : pop-up with more actions. Selected pages are transferred to the PDF and Print dialog.

Unselect all : unselects all selected pages. Selected pages that are in view are marked with a orange border. The mini pin icon at the bottom of the screen shows all selected pages.

Cut : cuts the selected pages into the clipboard. Cut pages that are in view are marked with a red border. The mini scissor icon at the bottom of the screen shows the cut page ranges in the clipboard.

Undo : reverts all cut actions and empties the clipboard.

Left Paste : pastes the cut pages *before* the first selected page.

Right Paste : pastes the cut pages *after* the first selected page.

Delete : deletes the selected pages.

Inbox : returns to the Main view.

Note

By default a fixed button text is shown on mobile devices only: on desktops a hover text is shown. See Section 3.3, “SafePages” [29] on how to change this behavior.

Caution

After SafePages are sorted any Fast Print will fail and all SafePages will be cleared as a result. Please use a Hold Print instead.

3.10. User Details

This dialog shows the Pagometers and Financial status of the user, and is shown after a tap on the User button in the footer of the main panel.

Note

Depending on User Privileges the Financial section might not be shown.

- For an Internal User a Password button is shown, when a password has been set (i.e. not erased): see Section 4.4.5.10, “Password” [107]. A tap on the button will show the Password Reset Dialog.
- When users are allowed to change their PIN a PIN button is shown. A tap on the button will show the PIN Reset Dialog. Also see Section 4.11.4, “User Authentication” [157].
- When an Internet Print `protocol://authority` is present the Internet Printer button is shown. A tap on the button will show the Internet Printer Device URI.
- When Telegram Messaging is enabled the Telegram button is shown. It leads you to the Telegram Configuration dialog.

> **Deprecated**
>
> Telegram support is **deprecated** and will be removed in a next release.

- If 2-step Verification is enabled the Authentication button is shown. It brings you to the Authentication Dialog.

3.10.1. Internet Printer

Users can copy/paste their private Internet Print Device URI for driver print from this dialog. The `SAVAPAGE.ppd` file can be downloaded.

This dialog also shows the URL and credentials for macOS, Chrome OS and the SavaPage Android Print App.

The UUID can be replaced with a new one by pressing the UUID button.

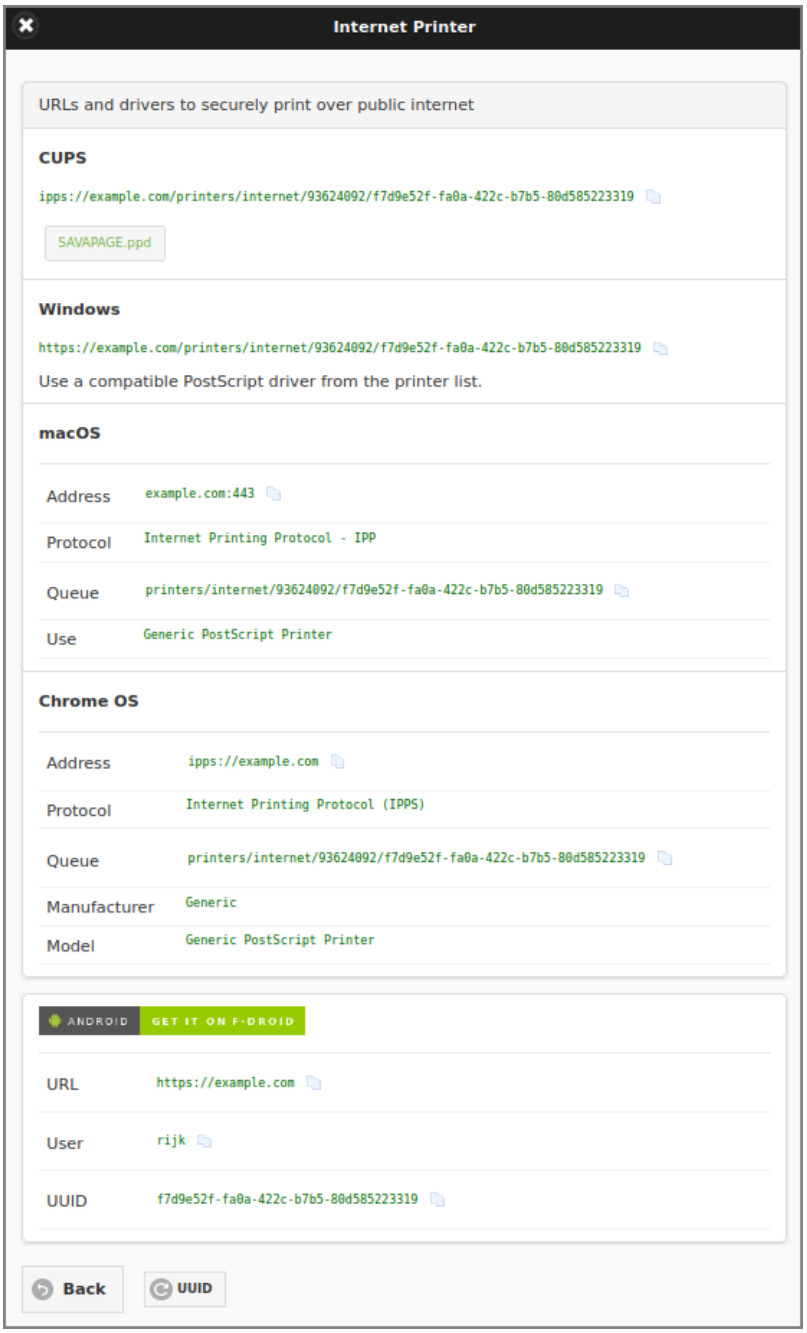


Figure 3.80. User Web App: User Details - Internet Printer

3.10.2. Telegram Configuration

Deprecated

Telegram support is **deprecated** and will be removed in a next release.

This dialog contains a step by step procedure to configure Telegram Messaging for a Telegram ID owned by a user.

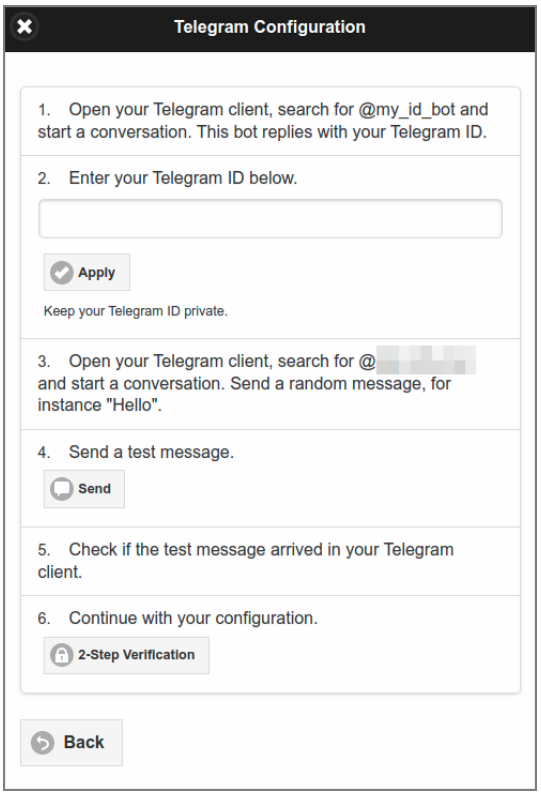


Figure 3.81. User Web App: User Details - Telegram Configuration

3.10.3. Authentication

In this dialog 2-Step Verification can be Enabled, and the user can Replace their TOTP code. To ensure that the TOTP code is scanned/copied correctly, the user must enter a verification code before pressing the Apply button.

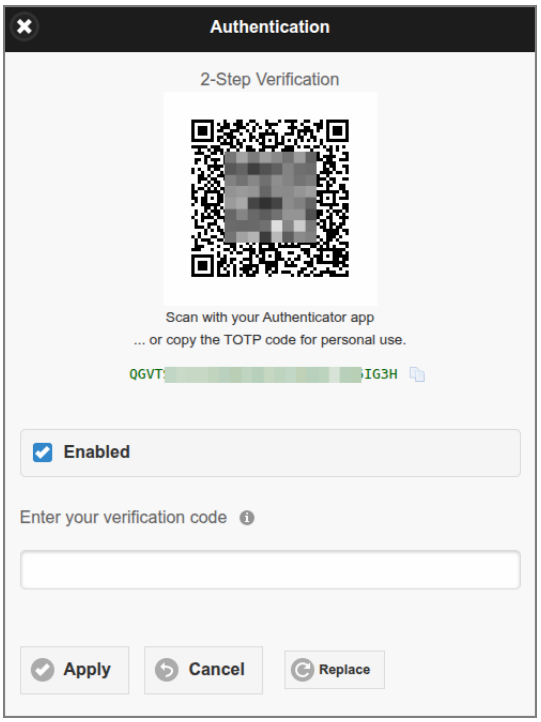


Figure 3.82. User Web App: User Details - Authentication

Note

If 2-Step Verification is `Enabled` and User Telegram Configuration is applied, the `Send code to Telegram` option is visible: if checked, no Authenticator app is needed and verification codes are send to the user's Telegram ID. However, at this time, you must enter a verification code generated by an Authenticator app.

Deprecated

Telegram support is **deprecated** and will be removed in a next release.

3.10.4. Pagometers

This section shows the *personal* Pagometers of the user, and are analogous to the ones in the Admin Dashboard as shown in Figure 4.15, “Admin Web App: Dashboard - Pagometer” [96] and Figure 4.18, “Admin Web App: Dashboard - Environmental Impact” [97].

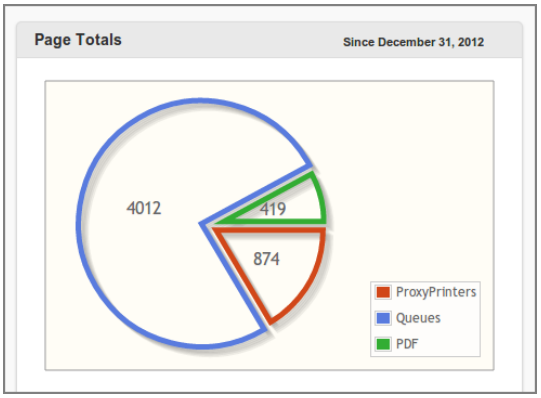


Figure 3.83. User Web App: User Details - pagometer

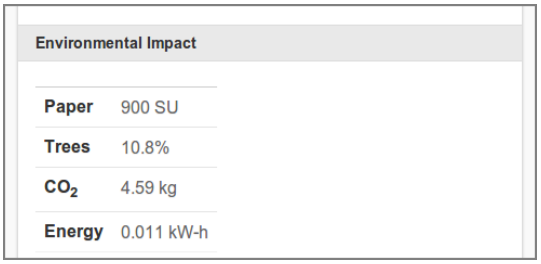


Figure 3.84. User Web App: User Details - Environmental Impact

The **Environmental Impact** section can be hidden with configuration property in table below.

Configuration property	Description
webapp.user.show-env-info	Set to `Y` (default) or `N`, to show/hide the **Environmental Impact** section.

Table 3.10. User Web App Environmental Impact Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these items.

3.10.5. Financial

This section shows the financial status of the user account and ways to increment the account balance from an external account.

Note

Depending on User Privileges the section for account transactions might not be shown.

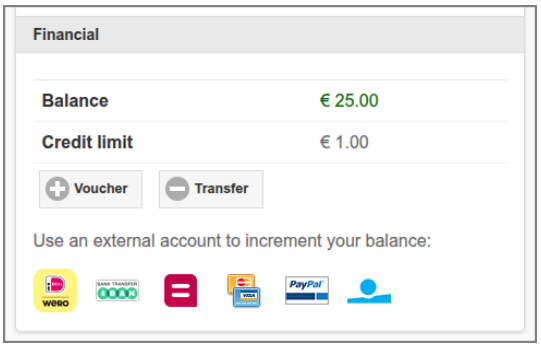


Figure 3.85. User Web App: User Details - Financial

- **Balance**: the user's account balance.
- **Credit limit**: see Section 4.4.5.8, “Financial” [106].
- If PaperCut Personal User Account is enabled, the **Balance** label is prefixed with the PaperCut icon and **Credit limit** is not shown.
- A push on the Voucher button opens the Redeem Voucher dialog. The visibility of this button is dependent on an application setting.
- A push on the Transfer button opens the Transfer Credit dialog. The visibility of this button is dependent on an application setting and a **Balance** amount greater than zero.

When a Generic Payment Gateway Plug-in is enabled, an icon is shown for each active payment method. Pushing (clicking) the payment method icon will pop-up the dialog to Transfer Money.

Note

The dedicated Payment Web App can be used if a Generic Payment Plug-in is enabled.

3.10.6. Redeem Voucher

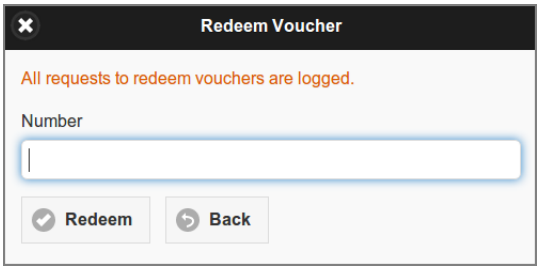


Figure 3.86. User Web App: Redeem Voucher

- Enter the voucher **Number** in the dialog box and press Redeem. Make sure to enter the number exactly as listed on the voucher including any dashes (-).
- If you entered the number correctly, the value as shown on the voucher will be transferred to your account and a new entry will list in your transaction log.

3.10.7. Transfer Credit

This dialog is used to transfer funds to another user.

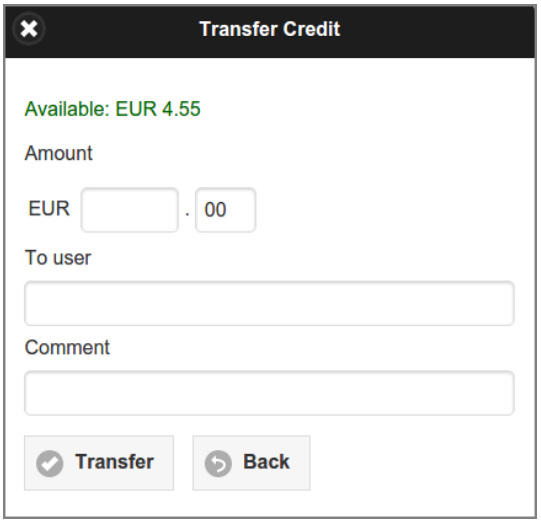


Figure 3.87. User Web App: Transfer Credit

- Enter the **Amount** in currency units and cents. The available amount is shown in green at the top.
- Enter the **To user** id and an optional **Comment**. A Quick Search for a User ID can be configured for this dialog.
- When you press the Transfer button the amount will be transferred from your account to the account of the target user. New entries will list in your transaction log and the log of the target user.

To configure this dialog see Section 4.11.11.5, “Transfer Funds” [174].

3.10.8. Transfer Money

This dialog is used to transfer money from an external account. The figure below shows a dialog in preparation for a credit card transaction. Other payment methods are available as defined by the active Generic Payment Gateway Plug-in. See for example Section M.1.1.1.1, “Mollie Payment Plug-in” [406].

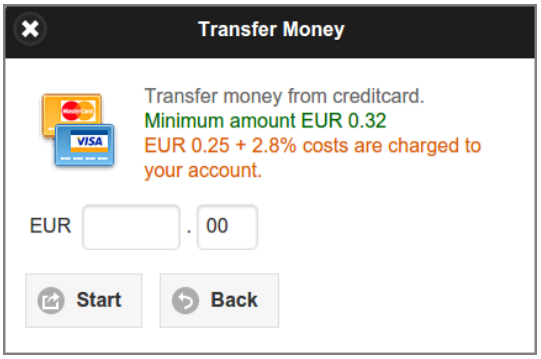


Figure 3.88. User Web App: Transfer Money from Credit Card

- Enter the **Amount** in currency units and cents.
- Press the Start button to start the payment transaction.

3.11. Upload

3.11.1. Upload Dialog

This dialog implement the SavaPage Web Print function, and is shown after a tap on the Upload button in the footer of the main panel.

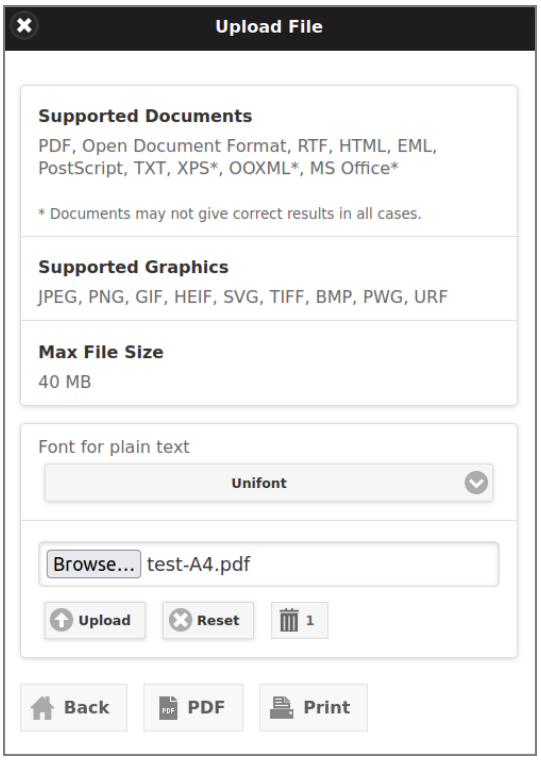


Figure 3.89. Web Print: Upload File

- **Font for plain text**: change the font when you upload a plain text file (TXT) that contains characters not supported by the Default font, like CJK. Use Unifont when the source text has a real broad Unicode coverage.
- Choose Files to be uploaded. Beware that the actual file selection button differs from browser to browser.
- Press the Upload button to start the upload.
- The status of the upload will be displayed below the selected file name.
- The Reset button clears the status messages and selected file.
- The "garbage bin" button shows the number of (uploaded) documents currently present in the SafePages inbox. When you press this button, all documents will be deleted, and the button hidden (until the inbox is filled again).
- After the upload, use the Back, Print or PDF buttons to navigate to the next step.

Supported Document and Graphics can be restricted with the following configuration properties:

Configuration property	Description
web-print.graphics.enable	Enable graphics files for Web Print: `Y` (default) | `N`
web-print.file-ext.exclude	A comma or space separated list of file extensions (without leading point), that are excluded for Web Print. For example: `rtf,html,ps,txt`. When TXT file type is excluded, the **Font for plain text** selection in Upload File dialog is removed.

Table 3.11. Configuration Properties for Web Print Document Types

See Section 4.11.15, "Config Editor" [189] on how to set these items.

> **Note**
>
> For uploaded file types that do not have a page size defined (HTML, TXT) the default paper size is used. For image files the graphic involved is a best fit on the default paper size.

3.12. Upload Drop Zone

The Upload Dialog and thumbnail area of the main view can act as Web Print *Drop Zone*. You can (multiple) select files in any desktop application and drag & drop them into the zone, after which they are immediately uploaded. When files are dragged into the zone, it lights up with a green border.

A Drag & Drop of an URL of a Web Page or Document has the same effect. In that case Web Page or Document is server-side downloaded and rendered as PDF. See Server-side Download properties.

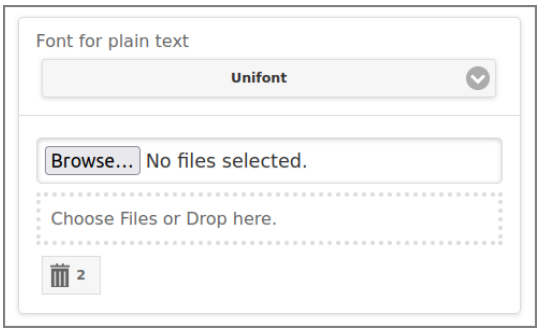


Figure 3.90. Web Print: Drop Zone - Upload Dialog

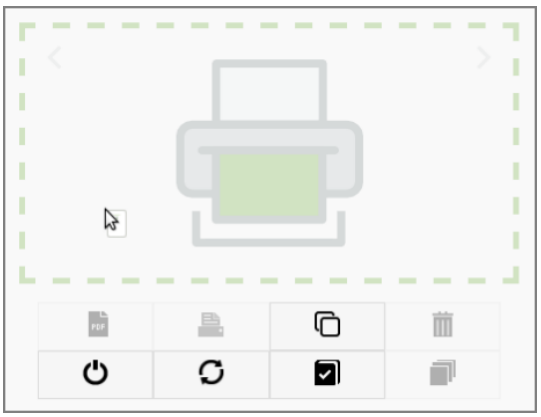


Figure 3.91. Web Print: Drop Zone - Main

The Drop Zone can be enabled at Section 4.11.7, “Web Print” [166].

> **Note**
>
> For plain text files dropped in the Main Drop Zone, the selected font in the Upload Dialog is used.

> **Note**
>
> A Web Page can be rendered by a Drag & Drop of its URL, but the result might not be accurate.

3.13. GDPR Dialog

This dialog is shown when the GDPR button is pressed from the SafePages top bar action menu, the Transaction List or the Document List.

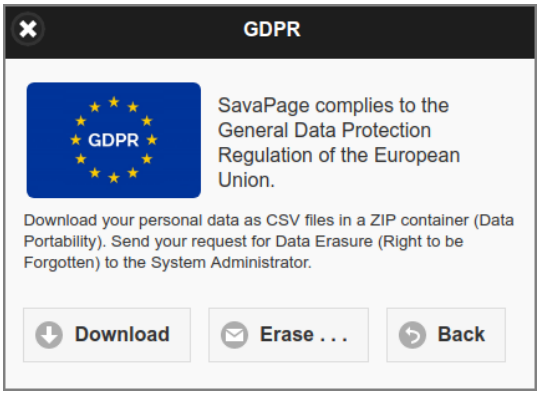


Figure 3.92. User Web App - GDPR Dialog

See Section 18.2.1, “Data Portability” [296] and Section 18.2.2, “Data Erasure” [296].

The following configuration properties apply:

Configuration property	Description
webapp.user.gdpr.enable	Enable the User Web App GDPR Dialog: `Y` (default) or `N`.
webapp.user.gdpr.contact.email	Optional email address for GDPR Data Erasure requests.

Table 3.12. Configuration Properties for User Web App GDPR

See Section 4.11.15, “Config Editor” [189] on how to set these items.

Chapter 4. Admin Web App

The Admin Web App can be reached at `https://savapage:8632/admin`. See Appendix E, *URL Cheat Sheet* [362].

4.1. Login

The Admin Login screen is a variant of the User Login screen described at Section 3.1, "Login" [22], with the following exception:

- The internal `admin` user and Persons with admin rights are allowed to log in. See Section 4.4.5, "Edit User" [102] how to assign admin rights to users.
- After a successful login Figure 4.1, "Admin Web App: Menu" [90] is shown.

Note

Initially, just after installation, only the internal `admin` user can login. See Section 4.4.8, "Administrator Role" [109].

Important

If configuration property `webapp.internet.admin.auth-mode.enable` = `Y`, user `admin` is denied access over the internet. See Section 17.3, "Access over Internet" [287].

4.2. Menu

After a successful login this main Admin screen is shown. If this is a first time login, a message will show, telling you that SavaPage needs to be set up and is not ready to use yet. The message will prompt you to go to the Options section and to check the settings. A long as setup is not completed this message will keep appearing after login. When setup is completed, a similar message will appear when the password of the internal `admin` account still has the default value.

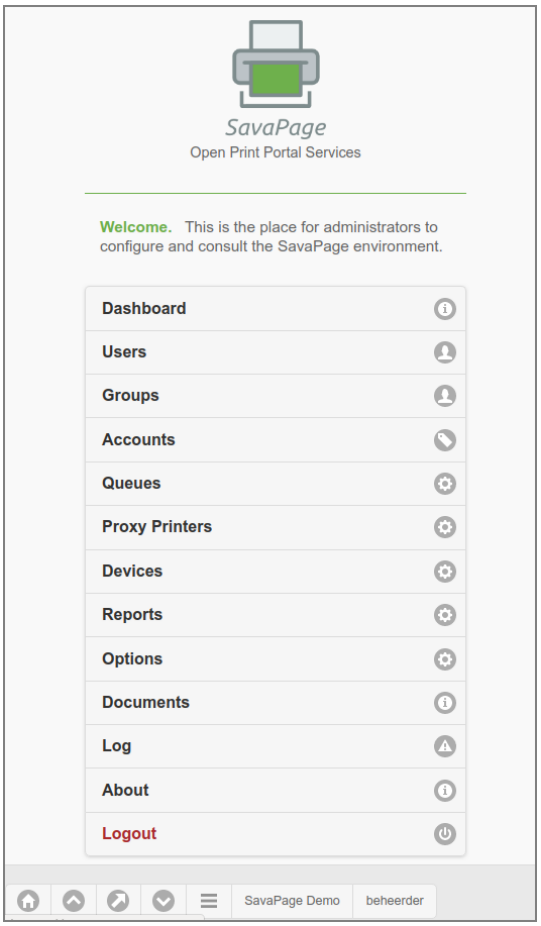


Figure 4.1. Admin Web App: Menu

A *tap* on a the Logout button brings you back to the Login screen. A *tap* on any other menu option brings a detailed screen into view. Please see the sections below for a description of each menu option:

- Section 4.3, “Dashboard” [91].
- Section 4.4, “Users” [98].
- Section 4.5, “Groups” [109].
- Section 4.6, “Accounts” [117].
- Section 4.7, “Queues” [120].
- Section 4.8, “Proxy Printers” [127].
- Section 4.9, “Devices” [139].
- Section 4.10, “Reports” [149].
- Section 4.11, “Options” [149].
- Section 4.12, “Documents” [190].
- Section 4.13, “Log” [197].
- Section 4.14, “About” [199].

These are the buttons in the footer and their function:

Navigate to the top of the page.

Navigate to the top of the menu.

Navigate to the top of the detail panel.

Navigate to the bottom of the page.

Show pop-up menu with additional actions as shown in the figure below.

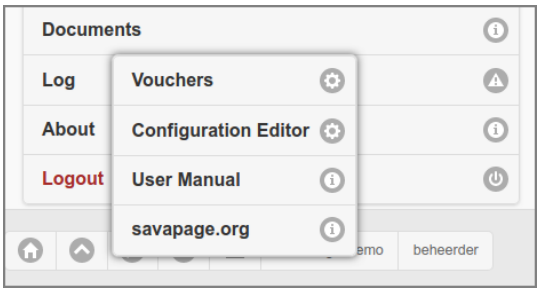


Figure 4.2. Admin Web App: Action Pop-up Menu

Please see the sections below for a description of menu options:

- Section 4.15, “Vouchers” [206].
- Section 4.11.15, “Config Editor” [189].

Note

Due to Admin Privileges certain menu options might *not* be visible.

The User Manual and savapage.org menu items open in a new browser tab.

At the leftmost of the footer is a button with the Community member name. When pushed it opens the About dialog. A label with the logged-on user id closes the ranks.

4.3. Dashboard

After a *tap* on the Dashboard button in the main menu this panel is shown. See Section 4.2, “Menu” [89].

Note

The Dashboard section is automatically refreshed every 60 seconds.

4.3.1. Status

The head section of the Dashboard displays indicators about Community Membership status and application runtime:

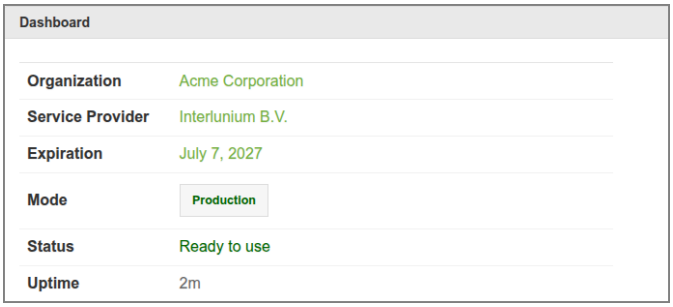


Figure 4.3. Admin Web App: Dashboard - Membership

- **Organization** : The name of your organization. It must be entered in the Organization section of the About page.
- **Service Provider** : The name of the support organization in the uploaded Service Card, or **none** if no Service Card is uploaded. See Section 4.14.6, “Support” [201].
- **Expiration** : expiration date of Service Card.
- **Mode** : the system mode. Press the button to change the mode to Maintenance or Production.

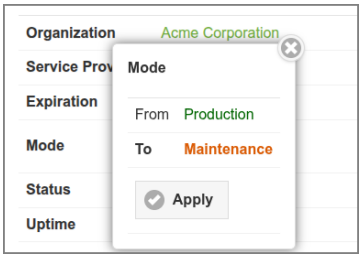


Figure 4.4. Admin Web App: Dashboard - System Mode

When system is in **Maintenance** Mode:

- Access to all Web Apps is restricted to users with Administrator role.
- Regular users are shown a message explaining the situation, at the Login dialog or in the authorized session.

- **Status**
  - **Ready to use** : SavaPage can be used without impediments.
  - **Setup is needed** : There are one or more options that need to be set up. Press the Currency , Mail and Organization button to do so. See Section 4.11.11.1, “Currency Code” [172], Section 4.11.5, “Mail” [162] and Section 4.14.5, “Organization” [201].

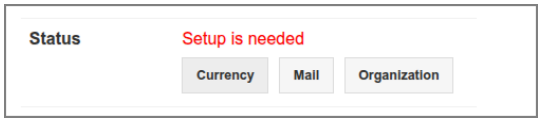


Figure 4.5. Admin Web App: Dashboard - Setup is needed

Access to the User Web App is denied till setup is finished. In the Admin Web App, editing of user details and adding internal users, user groups and shared accounts will not be available.

  - **Not available** : Database access is blocked because a batch job is in progress (database backup, user synchronization, database cleanup). To prevent that user actions requiring database access will block, resulting in long waits,

access to the User Web App is temporarily denied. Users are informed after a login attempt or, when already logged in, are warned when actions requiring database access are not permitted.

- **Environment** : shown if SavaPage runs in Docker. See Section 4.14.9, “Environment” [203].

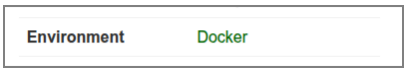


Figure 4.6. Admin Web App: Dashboard - Environment

- **Uptime** : the time the application has been working and available.

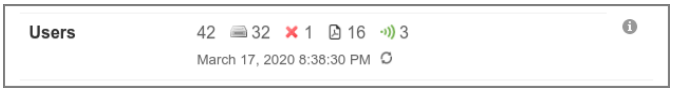


Figure 4.7. Admin Web App: Dashboard - Users

- **Users** : the number of users in the database (excluding Deleted Users) appended with the result of the latest User Home Clean.

  Cleanup is done nightly, but can be ad-hoc started by pressing the refresh button at the right of the cleanup timestamp.

  In this example there are `42` users in the database. At the time of the cleanup `32` users have used the User Web App and thereby have a User Home safepages [367] location on disk, and in *all* User Homes, `1` file is unidentified, `16` documents are stored because of using SavaPage as Printer, and `3` Hold Print jobs are pending. The number of bytes of each total is displayed when moving the cursor over it, or by pressing the information button at the right.

  The number of cleaned files is communicated with a separate icon as shown below:

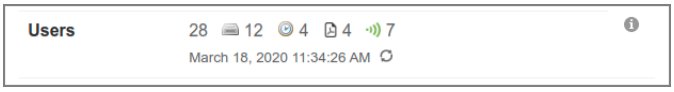


Figure 4.8. Admin Web App: Dashboard - User Home Clean

  - In this example a total of `4` documents and/or jobs are cleaned. Press the information button at the right to see details like “Cleaned: 2 Users. 1 Document: 36 KB. 3 Jobs: 121 KB”.
  - When Cleanup Test Mode [257] is set, cleanable files are signaled without actually being deleted. In that case the clean icon will show a red colored marker.

Figure 4.9. Admin Web App: Dashboard - Sessions and Messages

- **Client sessions** : the number of active User Web App and User Client sessions.
- **Recent errors** : the number of errors that occurred in the Application Log during the last hour.
- **Recent warnings** : the number of warnings that occurred in the Application Log during the last hour.
- **SSL valid till** : when the SSL Certificate of SavaPage expires within a year, its expiration date is shown . When expiration is due within 30 days it is shown in orange. Note: SSL expiration of any reverse proxy server used by SavaPage clients is not signaled.

- **OpenPGP** information is shown when server keys are configured: see Section 13.3, “OpenPGP Settings” [247]. Press the information button at the right to see the KeyID and Fingerprint.

Figure 4.10. Admin Web App: Dashboard - OpenPGP

- **Tickets** : Print and Copy Job totals just as in Job Tickets Web App, or zero ("0") if ticket queue is empty.

Figure 4.11. Admin Web App: Dashboard - Tickets

4.3.1.1. Technical Info

Technical information about the server process can be added to, or removed from, the Dashboard by setting the value of configuration key `webapp.admin.dashboard.show-tech-info` to `Y` or `N`. See Section 4.11.15, “Config Editor” [189] on how to change this value. When added, the following extra information appears:

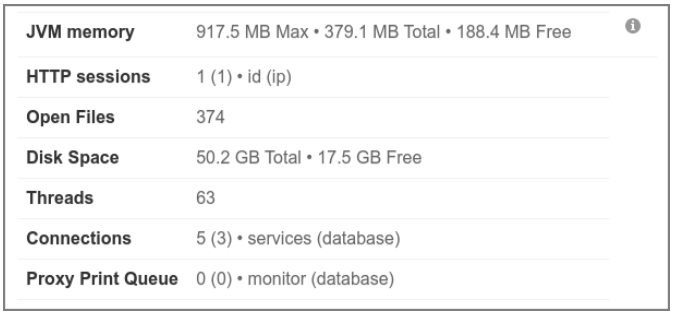


Figure 4.12. Admin Web App: Dashboard - Technical Information

- **JVM memory** : “Max” is the maximum memory the JVM will attempt to use. “Total” is the total memory in the JVM (varies over time). “Free” is the free memory in the JVM (increases after a garbage collect).
- **HTTP sessions** : the number of non-expired Web App Session id's and their client ip addresses. Due to DHCP, the number of ip's may be greater than the number of id's.
- **Open Files** : The number of open file descriptors.
- **Disk Space** : Disk capacity and free space.
- **Threads** : An estimate of the number of active process threads.
  - The number of “active (deadlocked)” threads (if any) are shown in red color. See Section 4.3.1.2, “System Monitor” [94].

Figure 4.13. Admin Web App: Dashboard - Deadlocked Threads (intentionally caused as example)

- **Connections** : The number of active connections to internal services and the database.
- **Proxy Print Queue** : The number of pending Proxy Print jobs in memory cache and the database.

4.3.1.2. System Monitor

A System Monitor process checks SavaPage server health status every 120 seconds. This frequency is set in configuration property `system.monitor.heartbeat-sec`. The following situation is checked:

- *Deadlocked threads* : JVM threads that are deadlocked waiting for “object monitors”[1] or “ownable synchronizers” [2]. The number of deadlocked threads is shown as Technical Info on the Dashboard. Detailed information can be found in the `server.log` : see Section 13.2, “Logs” [246].

> **Important**
>
> Deadlocked threads should not happen. When they occur, contact your SavaPage Service Provider.

4.3.2. Services

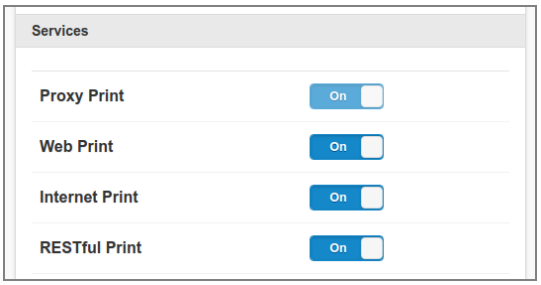


Figure 4.14. Admin Web App: Dashboard - Services

This section lists the status of services.

- The Proxy Print indicator is read-only, and shows the CUPS connection status. When connectivity is broken, the switch shows “Off”, with the reason displayed at the right.
- Core services like Mail Print must be *enabled* to be on the list.
- Web Print, Internet Print and RESTful Print services are a fixed part of the list.
- Plug-in services like Mollie Payment Gateways are part of the list if they are enabled in their property file.

With Dashboard Editor Privileges, you can turn a service **On** or **Off** by flipping the status switch. *With Reader Privileges service switching is disabled.*

When the SavaPage server restarts enabled *core* services are turned **On** by default. The initial state of enabled *plug-in* services is governed by the `online` setting in their property file. The on/off state of Internet Print translates to the enabled/disabled state of the reserved `/internet` Queue.

4.3.3. News

The News section shows the currently installed versus the latest published SavaPage version. A push on the button brings you to the Downloads and Release Notes Internet page.

> **Note**
>
> The latest published SavaPage version number is cached on the server and retrieved from the Internet every 12 hours.

[1] An “object monitor” is a mechanism that ensures that only one thread can be executing a given section (or sections) of code at any given time. In Java, there are two kinds of monitor regions: synchronized statements and synchronized methods.
[2] In Java, an ownable synchronizer [https://devdocs.io/openjdk~11/java.management/java/lang/management/lockinfo] is a synchronizer that may be exclusively owned by a thread and uses AbstractOwnableSynchronizer (or its subclass) to implement its synchronization property. ReentrantLock and ReentrantReadWriteLock are two examples of ownable synchronizers provided by the platform.

4.3.4. Pagometers

The *Pagometers*[3] counting the pages printed-out with Proxy Printers, printed-in from SavaPage Queues, and exported as PDF are displayed in a Pie-Chart. Pagometers are explained at Section 4.11.14.14, “Pagometers” [188].

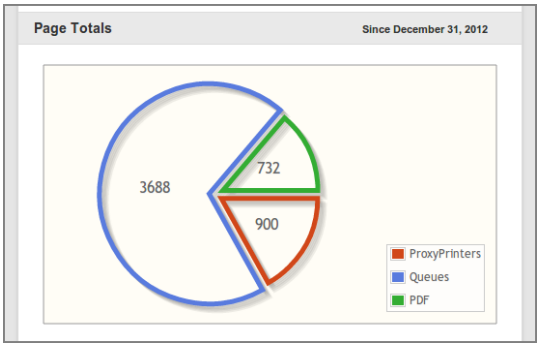


Figure 4.15. Admin Web App: Dashboard - Pagometer

A Line-Graph shows the day pagometers for the three sources over the last 30 days.

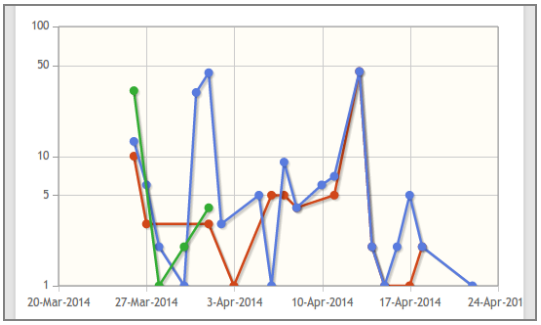
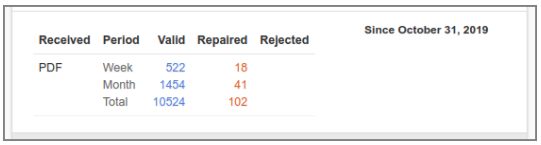


Figure 4.16. Admin Web App: Dashboard - Pagometer Trend

A table shows the running Total of Received PDF documents by Driverless File Transfer Print that are Valid, Repaired or Rejected, including totals for last Week and Month. See Section 14.9.1, “Processing Driverless PDF Transfer” [269].



Since October 31, 2019

Received	Period	Valid	Repaired	Rejected
PDF	Week	522	18	
	Month	1454	41	
	Total	10524	102	

Figure 4.17. Admin Web App: Dashboard - Pagometer Print-in PDF

4.3.5. Environmental Impact

The Environmental Impact for the Proxy Printer pagometer are displayed in a separate section. The metrics and units used are discussed at Section 16.2, “Environmental Impact” [282].

[3] In analogy with the term *Odometer*, the term *Pagometer* is introduced as an instrument to count the number of processed pages.

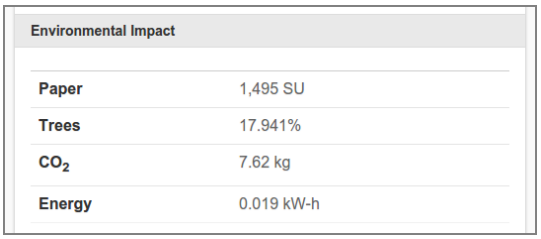


Figure 4.18. Admin Web App: Dashboard - Environmental Impact

The **Environmental Impact** section can be hidden with configuration property in table below.

Configuration property	Description
webapp.admin.dashboard.show-env-info	Set to `Y` (default) or `N`, to show/hide the **Environmental Impact** section.

Table 4.1. Admin Web App Environmental Impact Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these items.

4.3.6. Financial Summary

A Financial Summary of User Accounts, Point-of-Sale and Payment Gateway amounts are displayed in a separate section.

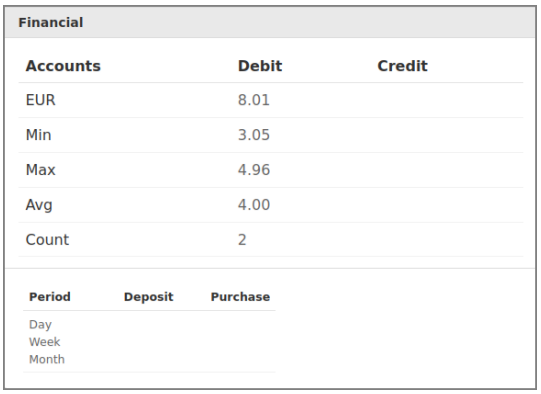


Figure 4.19. Admin Web App: Dashboard - Financial Summary

The User **Accounts** total and statistics like **Min**, **Max** and **Avg** are shown as **Debit** or **Credit** amount over **Count** number of accounts. Also, **Deposit** and **Purchase** amounts are shown from the Point-of-Sale Web App. The summary is updated as the dashboard is (auto) refreshed.

Note

Payments via a Payment Gateway Plug-in are shown as **External** besides **Deposit** and **Purchase** amounts.

4.3.7. Filtered Requests

This section shows the operation of all possible HTTP Service Filters when at least one of them is enabled.

Figure 4.20. Admin Web App: Dashboard - Filtered Requests

Request that exceed Denial of Service Filter limits, are **Denied** by the Inet Access Filter or are **Closed** due to HTTP/2 Rate Control are totaled over the last **Hour** and current **Day**. The filtering subjects (remote IP addresses, or other descriptors) acquired during the day are shown at the bottom of the table. Data is updated as the dashboard is (auto) refreshed. The column headers of inactive filter actions are grayed out.

See Section 17.1.1, “Denial of Service Filter” [284], Section 17.1.2, “Inet Access Filter” [285] and Section 17.1.3, “HTTP/2 Rate Control” [286] for configuration options.

Note

Statistics are collected during the server process. After a server restart, collection starts again from scratch.

4.3.8. Real-time Activity

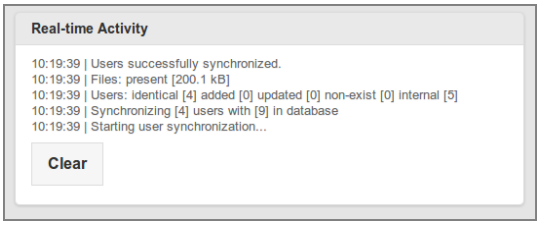


Figure 4.21. Admin Web App: Dashboard - Real-time Activity

Relevant system events are real-time displayed in this section. A maximum of 20 event messages remain in view, with the most recent one at the top.

System events are persisted in the rotating log file:

```
/opt/savapage/server/logs/adminpublisher.log
```

This file has a tab separated value (TSV) format for easy import and manipulation into spreadsheet programs. See `log4j.properties` for more information.

4.4. Users

4.4.1. User List

After a *tap* on the Users button in the main menu this panel is shown. See Section 4.2, “Menu” [89].

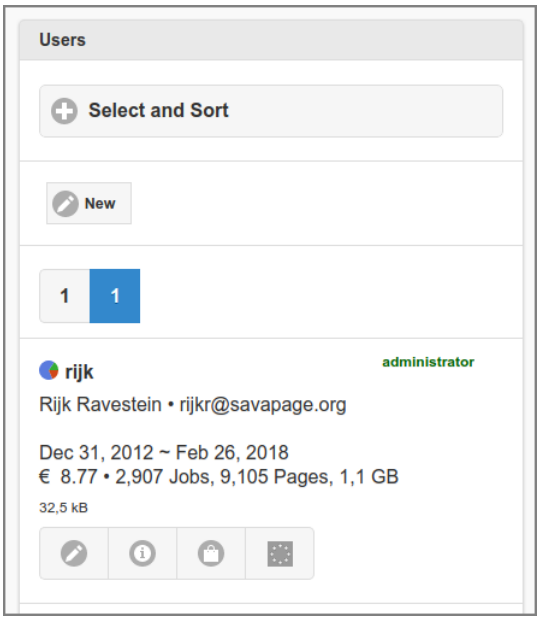


Figure 4.22. Admin Web App: User - List

- All non-deleted users are listed alphabetically by default. A different selection and sorting can be entered: see Figure 4.23, "Admin Web App: User - Select and Sort" [100].
- Press the New button to create and edit a new Internal User. Note: the button is *not* visible if System Status says **Setup is needed**.
- The list can be traversed by tapping one of the buttons at the *pager* at the top or bottom of the page.
- An entry is displayed for each user, with identifying data and some usage statistics. From top to bottom:
  - The user's role or status (at the top right corner).
  - An inline pagometer Pie-Chart followed by the user id. The *blue* color in the chart represents the number of pages printed to SavaPage. The *green* color represents the number of pages exported to PDF. The *red* color depicts the pages printed to Proxy Printers.
  - The user id of an Internal User is shown with an *orange* color.
  - The full name and email address. A key icon is shown if a PGP Public Key ID is present. See Section 4.4.1.1, "User List Item" [101].
  - If applicable, the User Name Aliases the user is known by.
  - The period in which user activity was accumulated on the pagometer.
  - The account balance and the pagometer including the number of jobs and bytes printed to any SavaPage Queue.
  - The size of the user's SafePages home.

  More user details are discussed in Section 4.4.1.1, "User List Item" [101].

- Tap the Edit button to change or delete the user. See Section 4.4.5, "Edit User" [102].

> **Note**
>
> Deleted Users cannot be edited.

- The Documents button brings you to the list of documents the user processed. See Figure 4.148, "Admin Web App: Document - List" [191]
- The Transactions button brings you to the list of financial transactions on the user's account. For a detailed description of this list see Section 3.8.2, "Transactions" [73] in the User Web App chapter.

- The rightmost GDPR button opens a pop-up where personal user data can be downloaded. See Section 4.4.2, “Download Personal Data” [101].

Note

Buttons are *not* visible if System Status says **Setup is needed**. If setup is not needed, certain buttons might *not* be visible due to Admin Privileges.

Tip

The pagometers of all users can be reset at Options → Advanced → Reset Pagometers

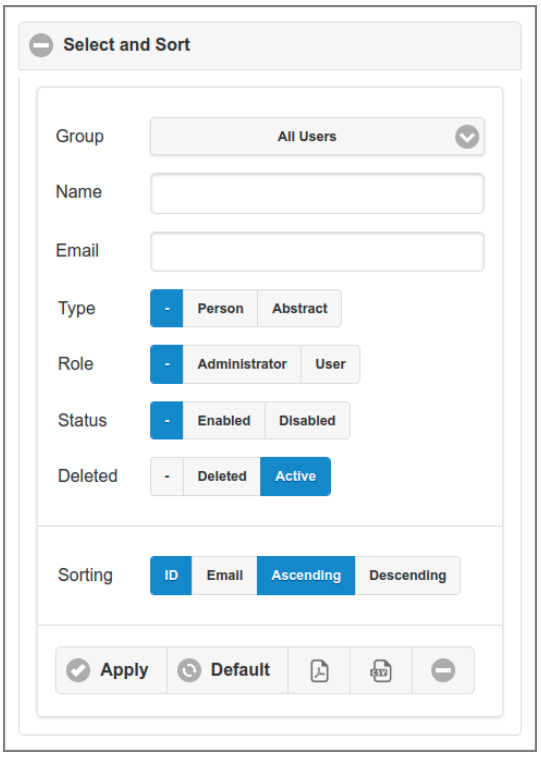


Figure 4.23. Admin Web App: User - Select and Sort

- Users can be selected by Group and by entering a part (fragment) of their (display) **Name** or **Email**. So entering "son" as ID will select both "jason" and "sonja".
- Select the **Type**, **Role** and (**Deleted**) **Status**. The - button will select both option values.
- If Self Registration is enabled, Registration Requests can be selected. The status of the verification (Pending or Expired) is shown in the User List. See Section 4.4.4, “Registration Requests” [102].

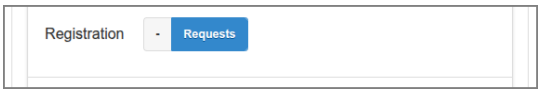


Figure 4.24. Admin Web App: User - Select Registration Requests

- The list can be sorted Ascending or Descending on Name ID or Email.
- *Tap* the Apply button to (re)display the list.
- A *tap* on the Default button resets the selection and sort fields to their default values.
- The PDF and CSV buttons downloads the User List in their respective formats, using the selection item values.
- The minus icon collapses the Select and Sort section.

4.4.1.1. User List Item

Each Users List Item displays the resolved number of enabled and disabled User Roles, Group Roles, User Privileges and Admin Privileges objects with their respective icon. User Roles are displayed next to the User ID. Objects that are resolved by User Group membership are shown below. A green color means that the objects are enabled, a red color means they are disabled. A description of the objects is displayed as a tooltip when pointed to with the mouse. An example item is shown below.

Figure 4.25. Admin Web App: User - List Item

Figure 4.26. Admin Web App: User - List Item - PGP Key

A key icon is shown at the email address if a PGP Public Key ID is present.

Figure 4.27. Admin Web App: User - PaperCut Account Balance

If PaperCut Personal User Account is enabled, the account balance of both PaperCut and SavaPage is displayed with their respective icons, as shown in the screenshot below:

4.4.2. Download Personal Data

This pop-up opens when the rightmost GDPR button is pressed in an entry from the User List.

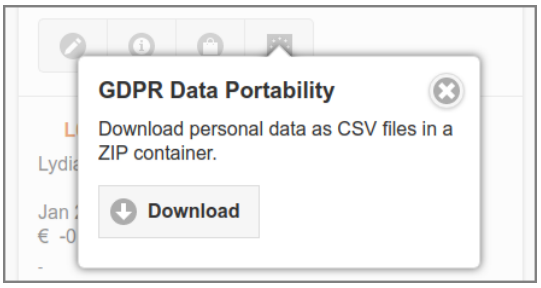


Figure 4.28. Admin Web App: User Data Portability

See Section 18.2.1, “Data Portability” [296].

4.4.3. Erased Users

Erased Users have their personal data cleared. Since their ID is empty, an “anonymous” placeholder with date/time of erasure is used for display.

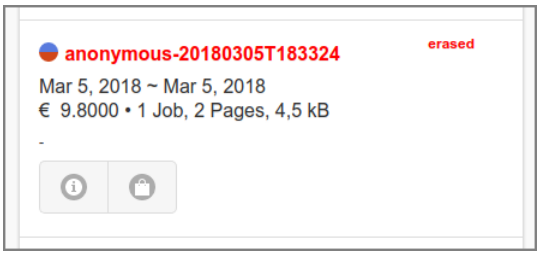


Figure 4.29. Admin Web App: Erased User

See Section 18.2.2, “Data Erasure” [296].

4.4.4. Registration Requests

Self Registration Requests for users that have not yet been verified are identied with **Pending** or **Expired**. Requests can be selected in the Select and Sort section.

4.4.5. Edit User

This chapter describes the editable sections of the User entity.

Caution

Some data you edit, like the **Name**, **Primary email**, **Card Number** and **ID Number** might be overwritten by values from the user source during synchronization. See Section 4.11.1.2, “LDAP” [150] and Section 4.11.3, “User Creation” [155].

Note

Users can also be edited and deleted with the Server Command Tool. See Section C.1.20, “setUserProperties” [338] and Section C.1.5, “deleteUser” [333].

4.4.5.1. Identity

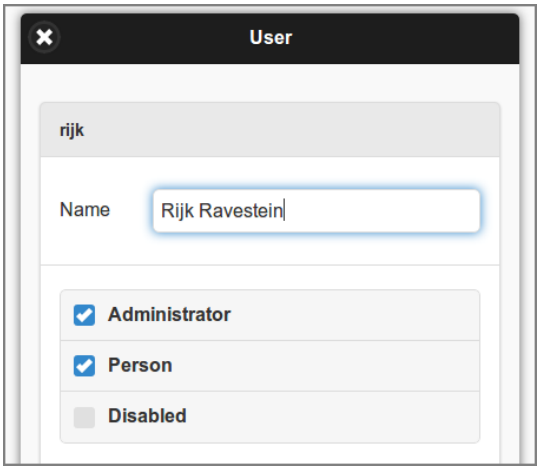


Figure 4.30. Admin Web App: Edit External User - Identity

- The user's full **Name** can be edited. Remember this name can be overwritten for an *external* User as a result of user synchronisation. See Section 4.11.3, “User Creation” [155].
- Assign the Administrator role by ticking the checkbox.
- Users are regarded as Person by default. Un-tick the Person checkbox if this user represents a generic functional account [277]. This will make the user Abstract.
- Tick the Disabled checkbox to deny the user access to the SavaPage application.

Warning

When a User becomes Abstract its SafePages are removed.

4.4.5.2. User Roles

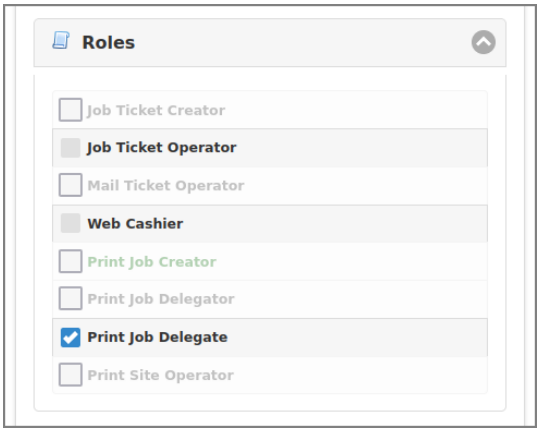


Figure 4.31. Admin Web App: Edit User - Roles

User Roles are needed to access certain application objects, as shown in the table below.

Role	Access
Job Ticket Creator	Job Ticket Printer
Job Ticket Operator	Job Tickets Web App
Mail Ticket Operator	Mail Tickets Web App
Web Cashier	Point-of-Sale Web App
Print Job Creator	A Proxy Printer that is *not* a Job Ticket Printer.
Print Job Delegate	Delegated Print and Users and Groups with role “Print Job Delegator” for Delegated Print.
Print Job Delegator	This is a passive role. Delegators can be accessed by users with role “Print Job Delegate”.
Print Site Operator	Print Site Web App

Table 4.2. User Roles

Each role is set with a checkbox that has three states:

- *Checked* : The role is *enabled*.
- *Unchecked* : The role is *disabled*.

- *Unchecked and grayed out*: The role is *indeterminate*.

If a User Role is needed to access an application function, SavaPage will check if this role is *enabled* for the authenticated user.

When the role is *indeterminate* at the user level, Group Roles are checked of the groups the user belongs to. Added Groups are checked first, then the Built-in Groups, with the "All Users" group as last.

- Access is *granted* if there is at least one group where the role is *enabled*.
- Access is *denied* when the role is *indeterminate* or *disabled* in all groups.
- Print Job Creator role is special: an *indeterminate* state at "All User" top level is interpreted as *granted*.

Note

A summary of enabled/disabled roles is displayed in the User List Item.

Caution

The 3-tier group hierarchy (User Groups > Internal/External Users > All Users) is traversed bottom up, to resolve the role of individual Users only. Group hierarchy is *not* used to resolve roles for User Groups: roles defined at group level are fixed, and are not interpreted in the context of other groups, or individual members.

4.4.5.3. Email

Figure 4.32. Admin Web App: Edit User - Email

- The **Primary email** and **Other emails** addresses are editable and must each be *unique*: they can be associated to just one User. Multiple emails must be separated by any of the characters *space*, *comma*, *semicolon*, or by carriage return or line feed.
- When saving, all email addresses are converted to *lowercase*.

4.4.5.4. Card, Yubikey

Figure 4.33. Admin Web App: Edit User - Card, Yubikey

- The **Card Number** must be *unique*: it can be associated to just one User. It must be entered in HEX/LSB format. See Section B.1, "Card Number Format" [326]. **Card Number** is used for Web App Login.

- The **YubiKey Public ID** is used for YubiKey Authentication.

4.4.5.5. ID number

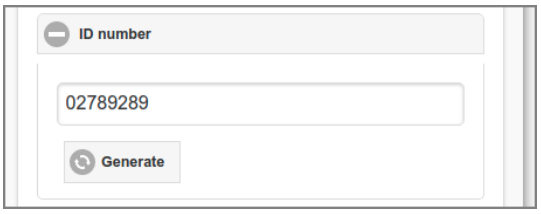


Figure 4.34. Admin Web App: Edit User - ID number

ID number must be *unique*: it can be associated to just one User. It is automatically generated when a user successfully logs in for the first time. The number can be re-generated by pushing the Generate button. The number is used for Web App Login and as authentication token for Internet Print. The following configuration properties apply:

Configuration property	Description
user.id-number-length-min	The minimum length of User ID number (manual entry). Default: 4.
user.id-number-generate.enable	Set to Y (default) or N to enable/disable automatic generation of User ID number.
user.id-number-generate.length	Length of generated User ID number. Minimal value: 6. Default: 8.

Table 4.3. Configuration Properties for User ID number

See Section 4.11.15, “Config Editor” [189] on how to change these values.

4.4.5.6. UUID

Figure 4.35. Admin Web App: Edit User - UUID

The UUID[4] is used as authentication token for Internet Print and RESTful Web Services. It is automatically created when a user successfully logs in for the first time. A new UUID can be created by pushing the Generate button.

Note

Users can replace their own UUID. See Section 3.10.1, “Internet Printer” [79].

4.4.5.7. OpenPGP

In this section the OpenPGP Public Key ID of the user can be entered. This key will be used to encrypt email send to the user. The Search and Check buttons are shown when the OpenPGP key server URL is configured. They

[4] A universally unique identifier (UUID) is an identifier standard used in software construction. See https://en.wikipedia.org/wiki/Universally_unique_identifier

open a tab to the key server, where public keys can be searched and checked (verified). See Section 13.3, “OpenPGP Settings” [247].

Figure 4.36. Admin Web App: Edit User - OpenPGP

4.4.5.8. Financial

This section shows the personal User Account. Initialization of this account is based on Group Membership as explained in the Edit Group section.

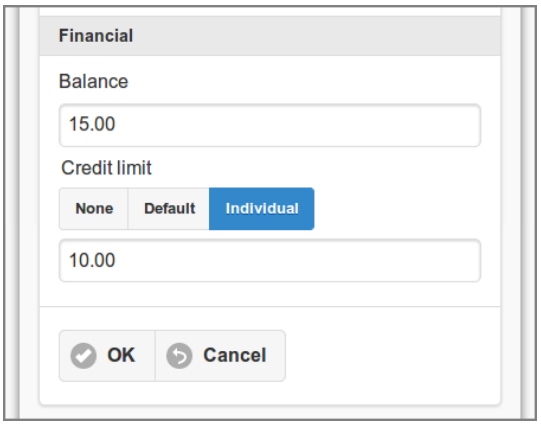


Figure 4.37. Admin Web App: Edit User - Financial

- A new value for the user's account **Balance** results in a financial transaction that corrects the previous account balance. See Section 3.8.2, “Transactions” [73]. Users are notified by a pop-up message in their active User Web App when their balance is adapted.
- Set the **Credit limit** with one of these buttons:
  - None : user has no credit limit, and is not restricted.
  - Default : user has default credit limit. See Section 4.11.11.2, “General Financial Options” [173].
  - Individual : when selected a custom credit limit can be entered.

4.4.5.9. PIN

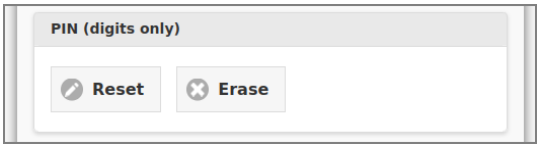


Figure 4.38. Admin Web App: Internal User - PIN Actions

A tap on the Reset button shows the PIN Reset Dialog. The Erase button is shown when a **PIN** is set. When pressed, it erases the **PIN** and makes itself disappear again. Use this dialog to initially set or change the value.

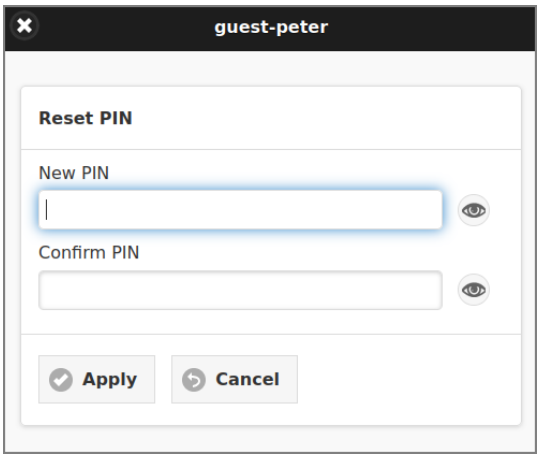


Figure 4.39. Admin Web App: Internal User - PIN Reset

The **PIN** must be digits only. The minimum and maximum length of a **PIN** are contained in the configuration keys `user.pin-length-min` and `user.pin-length-max`. A maximum value `0` (zero) indicates the maximum is unspecified. Use the Eye button on the right side of the entry fields to show/hide the PIN.

See Section 4.11.15, “Config Editor” [189] on how to change these values.

4.4.5.10. Password

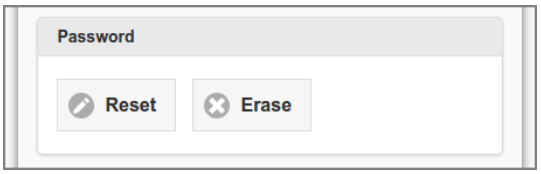


Figure 4.40. Admin Web App: Internal User - Password Actions

For an Internal User **Password** actions are shown.

The Erase button is shown when a password is set. When pressed, it erases the password and makes itself disappear again. Without an initial password, users cannot reset their password in the User Web App. *This gives administrators a means to disable login by user name/password, in favor of other authentication methods.*

A tap on the Reset button shows the Password Reset Dialog. This dialog is also used to initially set or change a password. Use the Eye button on the right side of the entry fields show/hide the password.

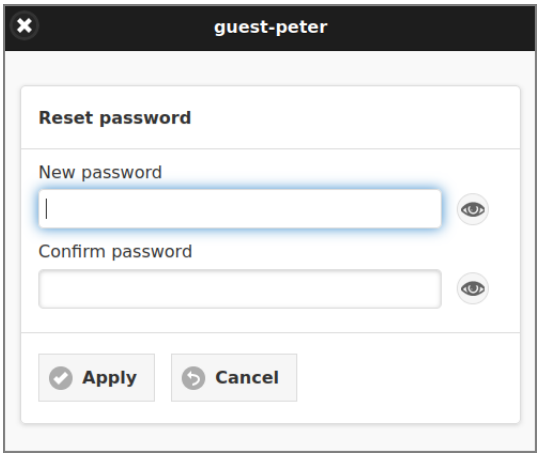


Figure 4.41. Admin Web App: Internal User - Password Reset

4.4.5.11. User Delete

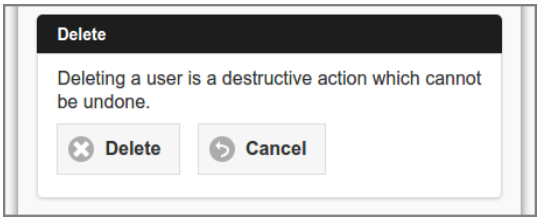


Figure 4.42. Admin Web App: Edit User - Delete

- Press the Delete button to delete the user and return to the User List. The next section describes the effect of this action.
- The Cancel button bring you back to the User List without changing anything.

4.4.6. Create Internal User

A tap on the New ... button at the top of the User List gives this dialog to create a new Internal User. Apart from the regular User data, the attributes **ID**, **PIN** and **Password** can be entered.

- The prefix of **ID** is contained in the configuration key `internal-users.username-prefix`.
- The **PIN** must be digits only. The minimum and maximum length of a **PIN** are contained in the configuration keys `user.pin-length-min` and `user.pin-length-max`. A maximum value `0` (zero) indicates the maximum is unspecified.
- The minimum length of the **Password** is contained in the configuration key `internal-users.password-length-min`.
- See Section 4.11.15, “Config Editor” [189] on how to change these configuration values.
- The **Financial** data are initialized with the New User Settings of the Built-in Internal Users Group. If these new user settings are disabled the **Balance** is set to zero with an Individual Credit limit of zero.

Note

Internal Users can also be created by Self Registration.

Tip

Internal Users can also be added with the Server Command Tool. See Section C.1.2, “addInternalUser” [331].

4.4.7. Deleted Users

Deleting a User makes sense if they are not part of the user source anymore and they were not deleted as part of a bulk delete during a manual synchronization. As long as job history or account transactions for a User are present [5], SavaPage applies a logical delete. Any logical deleted User will be physically deleted from the database when no related job history and account transactions are present anymore. This situation will automatically occur when you enabled automatic backup in combination with the delete of old document and transaction logs.

Important

If SavaPage synchronizes a new User from the user source, a new user instance will be created in the database, despite the fact that a logical deleted User exists with the same identifying name, i.e. the logical delete status of the "identical" user will remain as it is.

4.4.8. Administrator Role

SavaPage sets up a dedicated account called `admin`. This is the master administrator account, with access to all application functions, whose password is assigned during configuration. In large organizations it is likely that the administrator role needs to be granted to more than one person. One solution is to give all those persons the master password; however a better approach is to assign the administrator role to the network user accounts of these individual's. The advantages of this approach are:

- Administrators can access the Admin Web App with their own username and password.
- Since most administrative activity is logged in an audit trace, changes can easily be tracked back to an individual.

Note

Access to certain parts of the Admin Web App can be set on User Group level with Admin Privileges.

Tip

Administrative users should login via `https://savapage:8632/admin` rather than `https://savapage:8632/` or `https://savapage:8632/user` so that they are directed to the correct interface.

4.5. Groups

Groups are collections of users. You can Add and Remove groups as present in the external User Source or Internal Group definition.

Note

SavaPage caches group members for performance reasons. Therefore, when group membership changes at the source, it may not be immediately known in SavaPage. The membership cache is updated automatically

[5] When a user does not print on their own, but is printed for via Delegated Print, no job history is present for that user, but (pending) transactions are.

> according to the “Import new users overnight” option in the User Creation section, but can be also be refreshed manually at any time by a push on a button in the same section.

4.5.1. Built-in Groups

There are three built-in groups:

- *All Users* : all users in the system.
- *External Users* : all users synchronized from the external User Source.
- *Internal Users* : all users created inside SavaPage. See Section 4.11.1.3, “Internal Users” [153].

4.5.2. Group List

After a *tap* on the Groups button in the main menu this panel is shown. See Section 4.2, “Menu” [89].

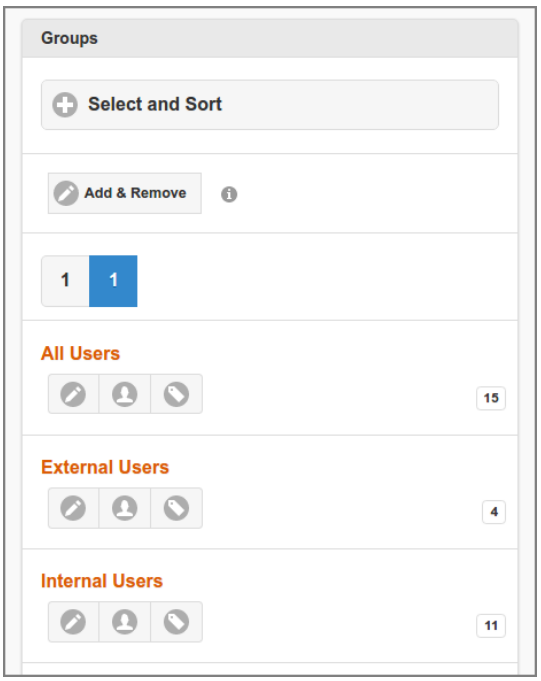


Figure 4.43. Admin Web App: User Group - List

Built-in groups are depicted in orange. Press the Add & Remove button to add additional groups.

Each item in the list shows the number of members and has buttons to jump to other dialogs. From left to right, these buttons bring you to:

- The Edit Group dialog.
- The User List with the group preselected. *Note: the button is not visible when number of group members is zero.*
- The Account List with the Group Type and Name preselected. *Note: the button is not visible when the (lazy created) Group Account is not present yet.*

> **Note**
>
> Due to Admin Privileges certain buttons might *not* be visible. Also, some buttons are *not* visible if System Status says **Setup is needed**.

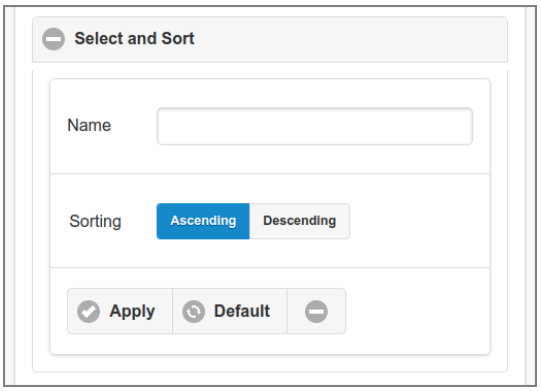


Figure 4.44. Admin Web App: Group - Select and Sort

- Groups can be selected by entering a part (fragment) of their name.
- The list can be sorted Ascending or Descending on group name.
- *Tap* the Apply button to (re)display the list.
- A *tap* on the Default button resets the selection and sort fields to their default values.
- Use the “minus” button to collapse the Select and Sort section.

4.5.2.1. Group List Item

Each Group List Item displays the number of enabled and disabled Group Roles, User Privileges and Admin Privileges objects with their respective icon. A green color means that the objects are enabled, a red color means they are disabled. Additionally, the icon is shown with the number of Shared Accounts the User Group has access to. A description of the objects is displayed as a tooltip when pointed to with the mouse. An example item is shown below.

Figure 4.45. Admin Web App: Group - List Item

Note

Roles and Privileges that are resolved by User Group membership are shown in the User List Item of a member.

4.5.3. Add & Remove Groups

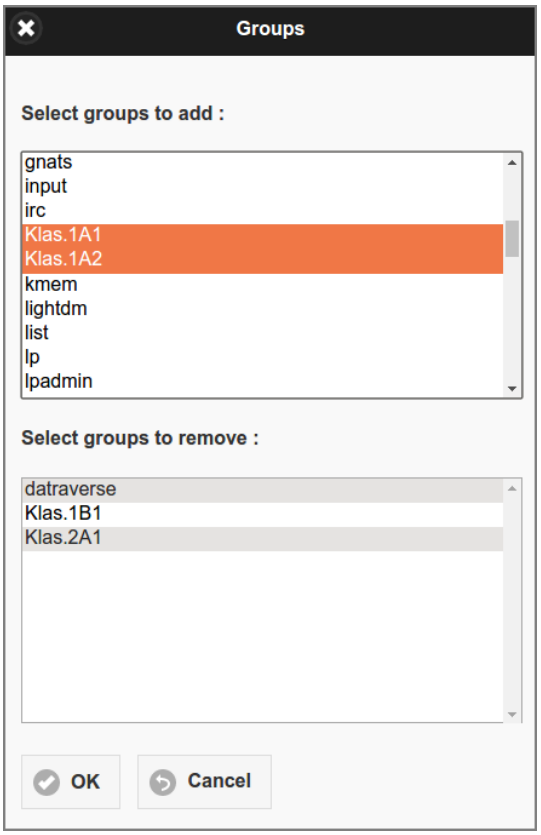


Figure 4.46. Admin Web App: User Groups - Add & Remove

Select the groups to add and to remove and press the OK button to commit the selection.

> **Note**
>
> The group list is a mix from the ones present in the external User Source and the ones defined as Internal Group. When adding a user group from Microsoft Active Directory, members from nested groups are included.

4.5.4. Edit Group

The Group Edit Dialog has several sections. Press the OK button at the bottom to commit all changes.

4.5.4.1. Group Roles

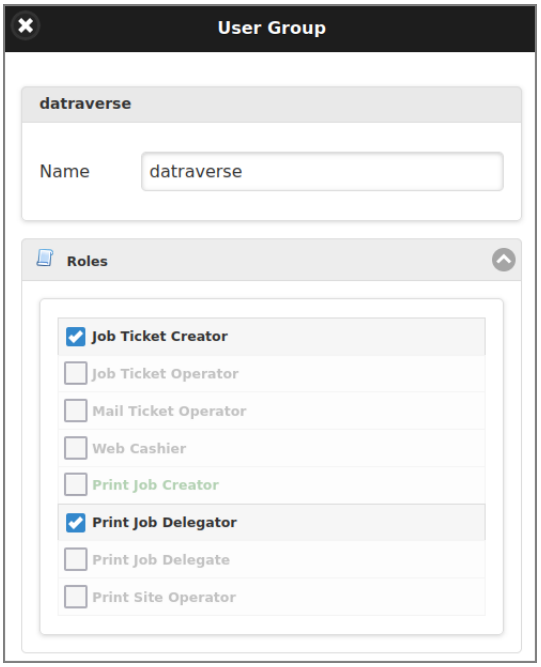


Figure 4.47. Admin Web App: User Group - Edit - Roles

In the **Roles** section you can set the user roles for group members. See Section 4.4.5.2, "User Roles" [103] for an explanation of the roles and how role based user access works.

Note

A summary of enabled/disabled roles is displayed in the Group List Item. Roles that are resolved by User Group membership are shown in the User List Item of a member.

4.5.4.2. User Privileges

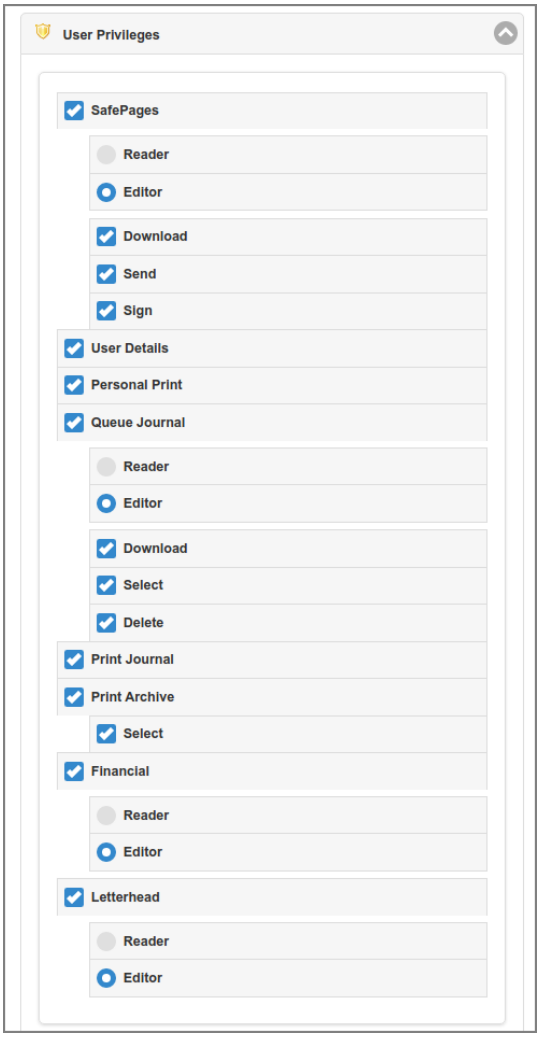


Figure 4.48. Admin Web App: User Group - Edit - User Privileges

In the **User Privileges** section you can set group member access to User Web App domain objects. Privileges are set by means three-state buttons. An unselected *grayed out* button means "indeterminate", plain unselected means "non-privileged" and selected means "privileged". When a privilege on a domain object is selected a role like Reader and Editor might be selected, as well as extra actions like Download, Send and Sign . The type of Roles and Actions offered depend on the type of domain object. This is how choices work out:

- If SafePages is non-privileged, the PDF and Sort buttons are *not* displayed in the Main Page. When privileged:
  - Reader role will display the PDF button if Download or Send action is enabled. Reader role will *not* display the Sort button.
  - Editor role will display both PDF and Sort buttons.
  - If Download and Send actions are both *disabled* the PDF button will not show.
  - Enabled Download and Send actions display the respective buttons in the PDF dialog. The enabled Sign action displays the same option in the PDF Security section.
- If User Details is non-privileged the footer button for the User Details dialog is replaced with a simple indicator holding the id of the authenticated user.

- If Personal Print is non-privileged, use of Personal Account for printing is not allowed. User can use Shared Accounts though, when permitted by Access Control. When printing with Personal and Shared Account is *not* permitted, role Print Job Creator is assumed, even when this role is explicitly selected.
- If Queue Journal is privileged a Print-in Job is journalled. Beware, that this function can be disabled for individual queues: see Section 4.7.2, “Edit Queue” [123]. When privileged:
  - Reader role will show the Download and Select privileges.
  - Editor role will additionally show the Delete privilege.
  - Enabled Download, Select and Delete privileges will display the respective buttons in the Document Log.
- If Print Journal is privileged a Print Job will be silently journalled. When Print Archive is privileged, the Print Job Archive option is active: when Select is privileged, the user is allowed to (de) select the archive, when not, the Print Job will be silently archived. Beware, that these functions can be disabled for individual printers: see Section 4.8.2, “Edit Proxy Printer” [132].
- If Financial is non-privileged, the account balance will not show in the footer, the Transactions button will not show in the Log page, and Financial data will not show in the User Details dialog. When selected, the Reader and Editor role will display all. However, only the Editor role is allowed account transactions in the User Details dialog. Editor role is also required to login to the Payment Web App.
- If Letterhead is non-privileged, the Letterhead button is not displayed. When privileged the Reader and Editor role allows user to choose a Letterhead in the PDF and Print dialog. The Editor role allows users to add letterheads themselves. See Section 3.6, “Letterheads” [68].
- The open spots left by buttons that are not displayed are taken by: the Upload button (moved from the footer), a Browse button pointing to the Browser, and an Info button (pointing to the About dialog), in that order. See Section 3.3.2, “Footer” [32].

This is how a privilege is evaluated on runtime:

- To be compatible with existing installations the “indeterminate” state for top level group “All Users” is interpreted as fully “privileged”. Of course, privileges can also be set at "lower" group levels. When determining privileges for a domain object, SavaPage looks at the lowest group first, and bubbles up to higher groups till a “non-indeterminate” privilege for the domain object is found.
- A denial of access due to a privilege takes precedence over any other configuration property.

Note

A summary of enabled/disabled privileges is displayed in the Group List Item. Privileges that are resolved by User Group membership are shown in the User List Item of a member.

4.5.4.3. Admin Privileges

In the **Administrator Privileges** section you can set group member access to Admin Web App domain objects. The objects correspond to the choices in the main menu. Any user with Administrator Role is assigned privileges by group membership.

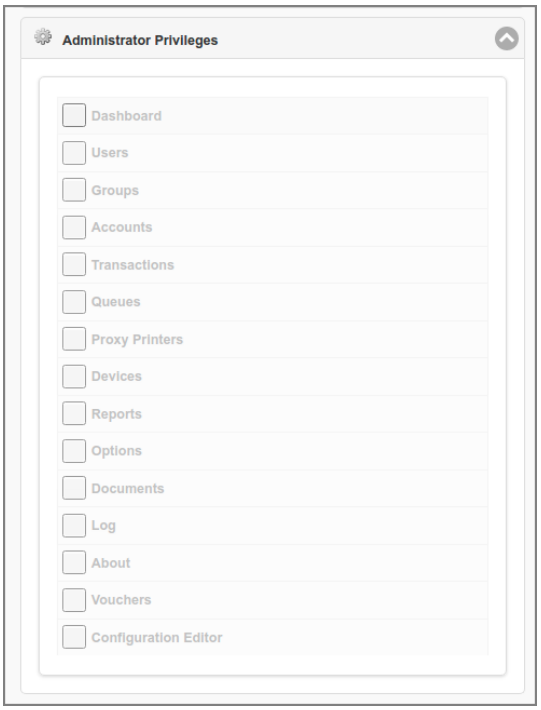


Figure 4.49. Admin Web App: User Group - Edit - Admin Privileges

Privileges are set and evaluated by means three-state buttons, just as User Privileges. For most domain objects a Reader and Editor role can be selected. Access to domain objects will be shown or hidden according to the privileges.

Note

A summary of enabled/disabled privileges is displayed in the Group List Item. Privileges that are resolved by User Group membership are shown in the User List Item of a member.

4.5.4.4. New User Settings

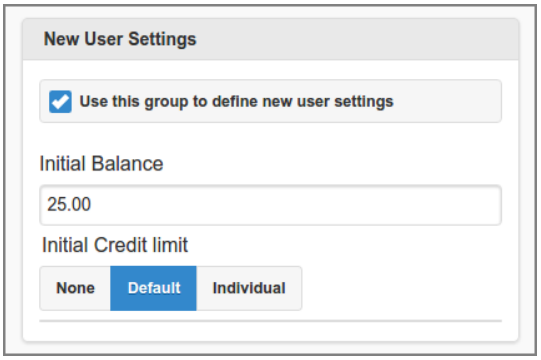


Figure 4.50. Admin Web App: User Group - Edit - New User Settings

When **New User Settings** are *enabled* they are automatically applied upon User Creation for members of this group. Note that these settings do **not** affect existing user members. See the Financial section of the Edit User dialog for a description of the Balance and Credit Limit fields.

When a user belongs to **multiple** groups, the New User Settings of these groups is applied as follows:

- The user is assigned an initial Balance that is the *sum* of the Initial Balances of *all* matching groups (with the exception of the Built-in Groups).
- If *any* of the matching groups has Initial Credit Limit “None” the user is assigned this status.
- Since the New User Settings are applied in alphabetical group name order, the Initial Credit Limit “Default” and “Individual” are assigned from the *last* group.

When a user does **not** belong to any group with New User Settings enabled, user is assigned the settings of the “External Users” or “Internal Users” Built-in Group (depending on the type of User Source).

> **Note**
>
> New User Settings are not shown for Built-in Group “All Users” because they are never used.

4.6. Accounts

This section is about *Shared Accounts* as explained in Chapter 12, *SavaPage Financial* [243]. The accounts are utilized in Delegated Print and PaperCut print scenarios.

4.6.1. Account List

After a tap on the Accounts in the main menu this panel is shown. See Section 4.2, “Menu” [89].

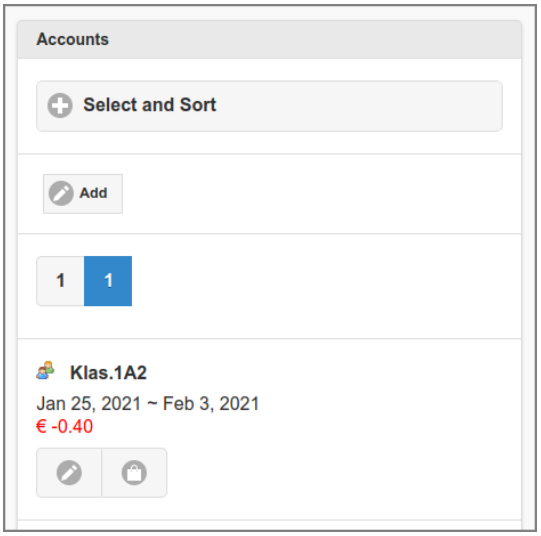


Figure 4.51. Admin Web App: Account - List

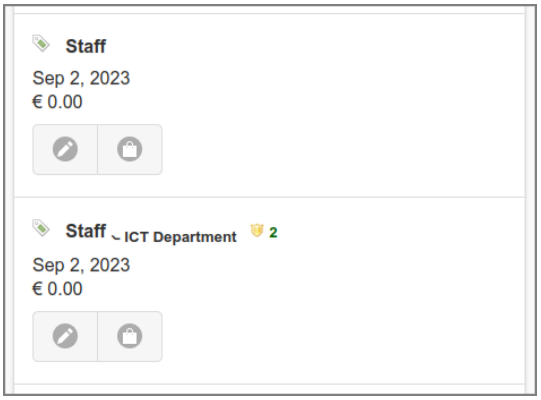


Figure 4.52. Admin Web App: Account - List - Sub Accounts

- All accounts are listed alphabetically by default. A different selection and sorting can be entered: see Figure 4.54, “Admin Web App: Account - List - Select and Sort” [118].
- The icon signifies a plain Shared Account and a Group Account.
- Press the Add button to create and Edit a new Account. Note: the button is *not* visible if System Status says **Setup is needed**.
- The list can be traversed by tapping one of the buttons at the pager at the top or bottom of the page.
- An entry is displayed for each Account, with the account type icon and name, the transaction period and the balance amount.

Figure 4.53. Admin Web App: Account - SavaPage Account Balance

If PaperCut Personal User Account is enabled, the amount is prefixed with the SavaPage icon, as shown in the screenshot above.

- Parent __ child accounts are depicted with a subscripted smaller font child name.
- Plain Shared Account items display a "shield" icon with the number of User Groups (if any) whose members can use the account to charge (printing) costs on (see the **Access control** section in the Edit Dialog). The group names are displayed as tooltip when pointed to with the mouse.
- Tap the Edit button to change or delete the account. See Section 4.6.2, “Edit Account” [119].
- The Transactions button brings you to the list of transactions. This list has an identical setup as the one for User Account transactions: see Section 3.8.2, “Transactions” [73].

Note

Due to Admin Privileges certain buttons might *not* be visible.

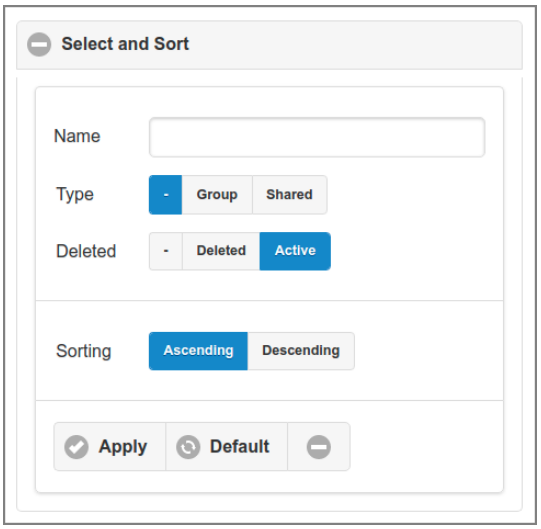


Figure 4.54. Admin Web App: Account - List - Select and Sort

- Accounts can be selected by entering the containing text (fragment) of their **Name**.
- Select the **Type** and **Deleted** Status. The - button will select both option values.

- The list can be sorted Ascending or Descending on Name.
- Tap the Apply button to (re)display the list. A tap on the Default button resets the selection and sort to their default values. The minus button collapses the section.

4.6.2. Edit Account

Group Accounts are ad-hoc created in the Delegated Print scenario, and have names identical to their User Group. Editing of Groups Accounts is limited to **Balance**, **Notes** and the Delete option. **Shared Accounts** can be freely created, and can fully be edited as explained below.

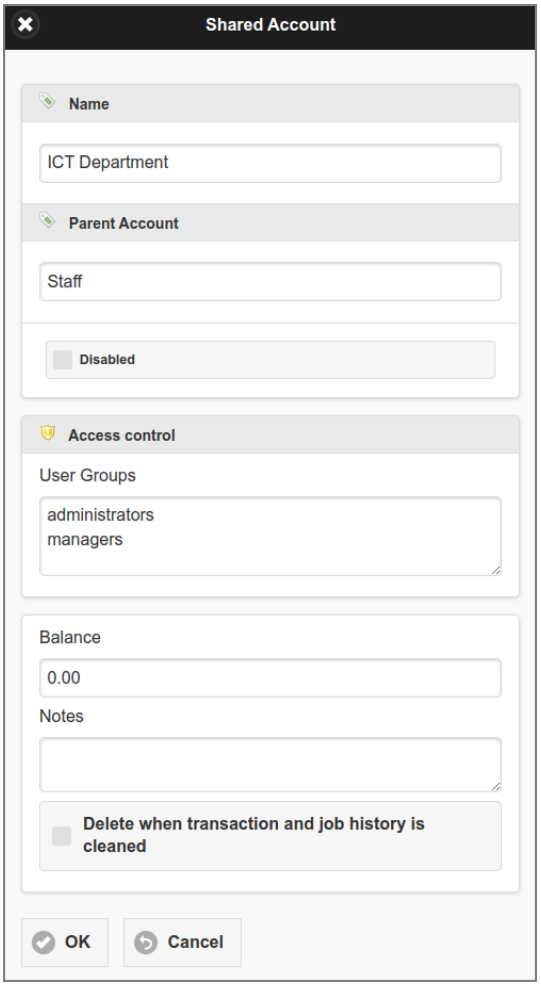


Figure 4.55. Admin Web App: Account - Edit

- The **Name** must be case insensitive unique for parents and for children within a parent. For instance, the following `Parent\Child` name combinations can coexist: `A\C`, `B\C`, `C\C`. Shared and Group Accounts have different name spaces, so account name “A” can exist as Shared *and* Group Account. A `"."` (point) character can not be part of an account name.
- *Group Accounts cannot act as Parent Account*. Therefore the **Parent Account** *must* be a Shared Account.
- An Account that acts as Parent Account for other accounts can have no Parent Account itself.
- In the **Access control** section names of User Groups can be entered, whose members can use this account to charge (printing) costs on. See Section 3.5.5, “Print Job Settings” [53] and Section 3.5.8, “Delegated Print Edit” [57]. The number of User Groups is displayed with a "shield" icon at the item in the Account List.

- A new **Balance** value results in a financial transaction that corrects the previous account balance. See Section 3.8.2, “Transactions” [73].
- Tick the Delete checkbox to delete the Account. This will be a logical delete as long as transactions are present. Any logical deleted Account will be physically deleted from the database when no related transactions are present anymore. This situation will automatically occur when you enabled automatic backup in combination with the delete of old account transactions. See Section 4.11.13, “Backups” [177].
  - A delete of a parent account cascades to all of its child accounts. An un-delete *does not*.
- Press the OK button to commit the changes and return to the Account List.
- The Cancel button brings you back to the Account List without changing anything.

4.7. Queues

Queues are Print-in channels for acquiring SafePages.

Note

Rate Limiting is applied to unauthorized access and invalid requests.

4.7.1. Queue List

After a *tap* on the Queues button in the main menu this panel is shown. See Section 4.2, “Menu” [89].

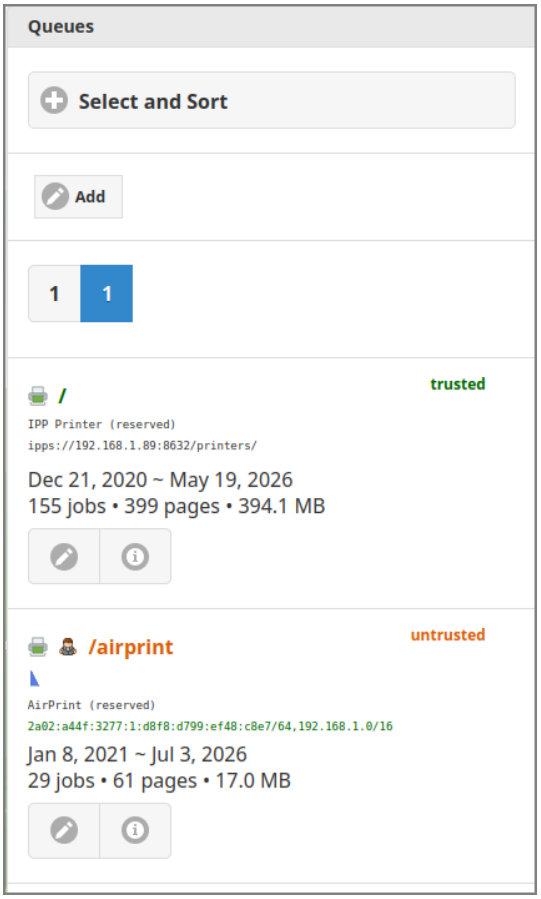


Figure 4.56. Admin Web App: Queue - List

- All non-deleted queues are listed alphabetically by default. A different selection and sorting can be entered: see Figure 4.57, “Admin Web App: Queue - Select and Sort” [122].
- Press the New button to create and edit a new queue.
- The list can be traversed by tapping one of the buttons at the *pager* at the top or bottom of the page.
- An entry is displayed for each queue, with identifying data and some usage statistics. From top to bottom:
  - The queue's trust or status (at the top right corner).
  - The URL Path of the queue. The path is relative to the /printers [363] URL base.
  - An inline Line-Graph showing the day pagometers of the printed pages over the last 30 days.
  - The full IPPS URL variant of the queue.
    - SavaPage takes the first IPv4 address of the first network interface on the server platform as URL host. You can overrule this behavior by setting a fixed IPv4 address or DNS name in configuration item `system.server.dns-name`.
  - Optionally, the allowed client IPv4 and/or IPv6 addresses as a CIDR Set.
  - The period in which activity was accumulated on the pagometer.
  - The pagometer of the queue including the number of jobs and bytes printed.
- Tap the Edit button to change or delete the queue. See Section 4.7.2, “Edit Queue” [123]
- The Log button brings you to the list of documents printed to the queue. See Figure 4.148, “Admin Web App: Document - List” [191]

Note

Due to Admin Privileges certain buttons might *not* be visible.

Tip

The pagometers of all queues can be reset at Options → Advanced → Reset Pagometers

When applicable, the following icons are shown:

	IPP Printer Queue.
	Basic Authentication enabled. • See Section 4.7.2, “Edit Queue” [123].
	Queue Journal enabled. • See Section 4.7.1.1, “Queue Journal” [121].

Table 4.4. Queue Icon

4.7.1.1. Queue Journal

The Queue Journal is the `doc.store.journal.in.print` branch of the Document Store. It must explicitly be enabled.

4.7.1.2. Select and Sort

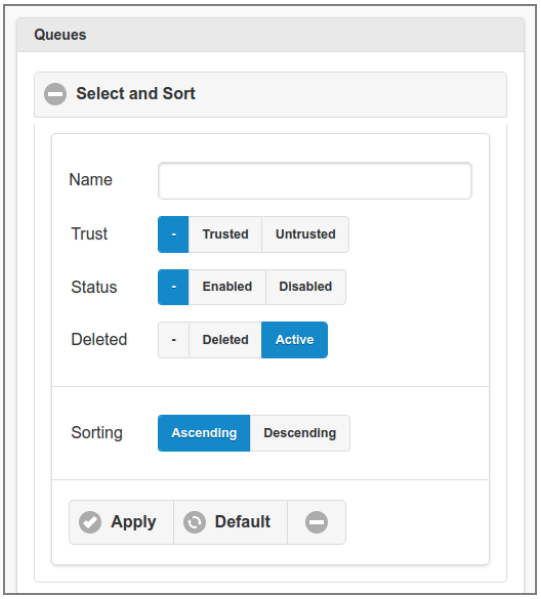


Figure 4.57. Admin Web App: Queue - Select and Sort

- Queues can be selected by entering the containing text (fragment) of their URL Path **Name**.
- Select the queue's **Trust** and (**Deleted**) **Status**. The `-` button will select both option values.
- The list can be sorted `Ascending` or `Descending` on URL Path Name.
- *Tap* the `Apply` button to (re)display the list.
- A *tap* on the `Default` button resets the selection and sort fields to their default values.

4.7.2. Edit Queue

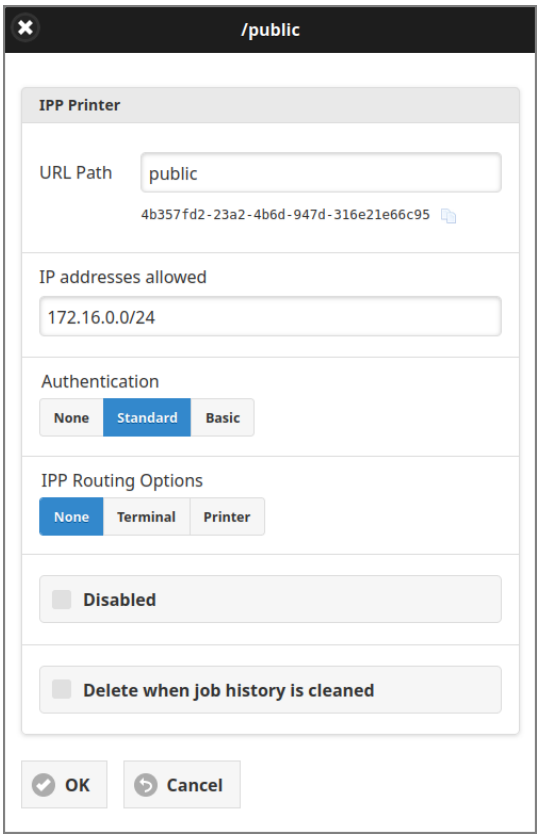


Figure 4.58. Admin Web App: Queue - Edit

- The **URL Path** is editable for non-reserved queues. Renaming the URL path name will permanently overwrite the old name in all related job history records with the new name.
  - In response to an IPP Get-Printer-Attributes request the "printer-name" and "printer-make-and-model" are returned as "SavaPage-urlpath". So, in this example the IPP printer-name will be "SavaPage-public". The UUID of the Queue, in this case `4b357fd2-23a2-4b6d-947d-316e21e66c95`, is used for IPP attribute "printer-uuid".
- Enter IPv4 and/or IPv6 address ranges as a CIDR Set at **IP addresses allowed** to restrict access to the queue based on requesting IP address. If the field is empty all requesting IP addresses from private networks are allowed to print to the queue[6]. See Section 13.5.5, "HTTP XFF Header Settings" [251].
- For **Authentication**, select the None option to make this a Trusted Queue. If this option is *not* selected, the queue will be a Public Queue with Standard IP Based Authentication or Basic Authentication.
- **IPP Routing Options** are discussed in the section below.
- Tick the Disabled checkbox to disable access to the queue for all users.
- When the Queue **Journal** function is enabled it can be Disabled for this queue. See Section 4.11.14.10, "Document Store" [185].
- Tick the Delete checkbox to delete the Queue. This will be a logical delete as long as related job history is present. Any logical deleted Queue will be physically deleted from the database when no related job history is present anymore. This situation will automatically occur when you enabled automatic backup in combination with the delete of old document logs. See Section 4.11.13, "Backups" [177].
- Press the OK button to commit the changes and return to the Queue List.
- The Cancel button brings you back to the Queue List without changing anything.

[6] IANA [https://www.iana.org/] reserved the following CIDR ranges for private networks: 10.0.0.0/8, 172.16.0.0/12 and 192.168.0.0/16.

Important

Some reserved queues like Web Print and Mail Print can not be edited. Other reserved queues like AirPrint and Internet Print are untrusted by nature, hence Authentication None cannot be selected.

Note

IPP printing from public IP addresses is denied by default. You can override this behavior by setting specific public IP address ranges at **IP addresses allowed**.

4.7.3. IPP Routing

With IPP Routing the PDF of incoming PostScript documents on a **non** reserved and driver-print Queue is routed to a Proxy Printer. There are two options for referring to this printer:

1. Terminal : refers to the **Single Printer** Target in the Custom Proxy Print of the Terminal that matches the client IP address.
2. Printer : refers to the printer as configured in this IPP Routing dialog.

When the `ipp.routing.enable` configuration property and both `ipp.routing.*.enable` options are enabled (see table below), the following section appears in the Edit Queue dialog:

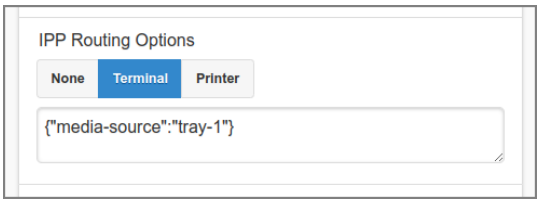


Figure 4.59. Admin Web App: Queue - IPP Routing Terminal Options

Select the Terminal option to enable “IPP Routing by Terminal” for this Queue.

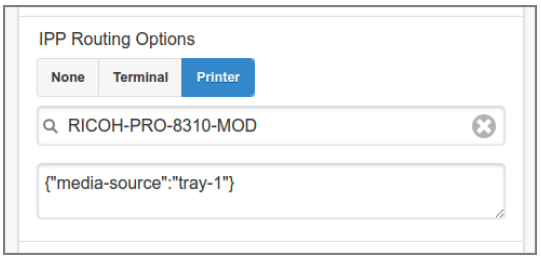


Figure 4.60. Admin Web App: Queue - IPP Routing Printer Options

Select the Printer option to enable “IPP Routing to Printer” and enter the target printer using the quick search field below it.

Optionally enter an IPP key/value map in `JSON` format to override printer defaults. In this example IPP attribute `media-source` is assigned value `tray-1`.

Important

Use IPP attributes for routing and optionally use PPD to IPP Mappings in the printer PPD Extension File for the target printer. For example:

```
{"copies":"1", "media-color":"white", "print-color-mode":"monochrome",
 "sides":"two-sided-long-edge", "media":"iso_a4_210x297mm"}
```

Note

If IPP Routing leads to a Job Ticket Printer or to a Hold Print Mode Printer, the print job is handled accordingly.

IPP Routing Queues are marked with a small printer icon in the Queue List with an additional icon in case of an IPP Routing Printer Queue.

Figure 4.61. Admin Web App: Queue - List - IPP Routing: Terminal Queue

Figure 4.62. Admin Web App: Queue - List - IPP Routing: Printer Queue

Figure 4.63. Admin Web App: Queue - List - IPP Routing: Printer Hold Queue

Figure 4.64. Admin Web App: Queue - List - IPP Routing: Job Ticket Print Queue

For a sample screenshot of the financial transaction see Figure 3.74, “User Web App: Log - IPP Routing Transaction” [74].

Configuration property	Description
ipp.routing.enable	Enable IPP Routing: `Y` or `N` (default)
ipp.routing.terminal.enable	Enable IPP Routing to Terminal: `Y` (default) or `N`
ipp.routing.printer.enable	Enable IPP Routing to Printer: `Y` (default) or `N`

Table 4.5. Configuration Properties for IPP Routing

See Section 4.11.15, “Config Editor” [189] on how to set these items.

Important

IPP Routing Plug-in instances are called on an IPP Routing event, just before the PDF is send to the Proxy Printer. These plug-ins edit the PDF, for example by adding a header, footer and QR code[7] image.

[7] https://en.wikipedia.org/wiki/QR_code

4.7.3.1. IPP Routing Script - Terminal

1. Alice logs into a computer at the Reception.
   - The computer has fixed IP address 192.168.1.50
2. Alice prints the “Sales Report” document to SavaPage Queue `/tray1`
3. The PostScript content is received by SavaPage and converted to PDF.
4. SavaPage notices that ...
   - `/tray1` is configured for IPP routing.
   - The requester's IP address 192.168.1.50 is linked to Terminal “Reception”.
   - Terminal “Reception” defines a Custom Proxy Print to the Single Printer Target “HP-Reception”.
5. SavaPage decides to print the PDF to Proxy Printer “HP-Reception”.
6. SavaPage notices that an IPP Routing Plug-in instance is present. Therefore, it calls the plug-in with the print job parameters (User id "Alice", Client IP address 192.168.1.50, CUPS printer name “HP-Reception”, Print Job Name “Sales Report”, date/time of the print) together with the PDF to be printed.
7. The IPP Routing Plug-in calls a RESTful service of a third-party QR code provider, passes the print job parameters, obtains the QR code in return, and draws an image of the QR-code on the PDF to be printed.
8. SavaPage prints the PDF as edited by the IPP Routing Plug-in to Proxy Printer “HP-Reception” using the IPP Routing Options of `/tray-1`.
9. Alice collects the printed “Sales Report” document from the “HP-Reception” printer next to her desk.

4.7.3.2. IPP Routing Script - Hold Printer

1. Bob logs into the User Web App to make sure he can print to untrusted SavaPage Queue `/tray2` with IP Based Authentication.
2. Bob prints the “Sales Report” document from his BYOD to `/tray2`
3. SavaPage notices that the requester's IP address 192.168.1.51 is authenticated and converts the PostScript content to PDF.
4. SavaPage notices that `/tray2` is configured for IPP Routing to Printer “RICOH-PRO-8310-MOD”.
5. SavaPage notices that “RICOH-PRO-8310-MOD” is a Hold Print Mode Printer and holds the print job using the IPP Routing Options of `/tray-2`.
6. Bob walks to the “RICOH-PRO-8310-MOD” printer and swipes his NFC token along the reader next to it. As a result the held job is printed.

4.7.3.3. IPP Routing Script - Job Ticket Printer

1. Alice prints 100 copies of the “Event Flyer” from her BYOD to untrusted SavaPage Queue `/tray3` with Basic Authentication.
2. A dialog pops up and Alice enters her user name and password.
3. SavaPage notices that the requester's IP address 192.168.1.52 is authenticated, that Alice has role Job Ticket Creator and converts the PostScript content to PDF.
4. SavaPage notices that `/tray3` is configured for IPP Routing to Printer “TICKET”.
5. SavaPage notices that “TICKET” is a Job Ticket Printer and uses the IPP Routing Options of `/tray-3` to create a Job Ticket assigned to Alice.
6. Bob is a Job Ticket Operator in the Print Room. He notices Alice's Job Ticket in the Job Ticket Web App, approves the ticket and prints the job to one of the printers in the room.
7. Alice walks to the Print Room to pick up the printed result.

4.7.4. Reserved Queues

Dedicated queues are reserved and pre-installed for:

- The default IPP "`/`" queue.
- The Raw IP Printer (Port 9100) queue "`/raw`"
- AirPrint
- Mail Print
- Web Print
- Internet Print
- RESTful Print

4.8. Proxy Printers

4.8.1. Proxy Printer List

After a *tap* on the Proxy Printers button in the main menu this panel is shown. See Section 4.2, "Menu" [89].

SavaPage automatically detects CUPS printer queues, and uses their unique name to replicate corresponding Proxy Printers in its database.

When a printer queue is deleted in CUPS it is real-time detected by SavaPage, and as a result the corresponding Proxy Printer is marked as *not present*. Proxy Printers which are *not present* are hidden in the User Web App, so they cannot be used for printing.

When a printer queue is renamed in CUPS, two events occur in SavaPage. First, the new name is detected as a new Proxy Printer, and second, the Proxy Printer with the old name is detected as a deleted CUPS queue. Proxy Printers which are not present anymore can either be *deleted* or *renamed*.

SavaPage selects *local* CUPS printer queues by default. This is a sensible policy since local CUPS queues are able to connect to locally attached printers (USB, LPT) or network enabled printers. However, in some odd cases you might want to proxy print to a *remote* CUPS queue, i.e. a queue shared by another machine. In that case you can set the value of the `cups.ipp.remote-enabled` config item to `Y`. See Section 4.11.15, "Config Editor" [189] on how to change this value.

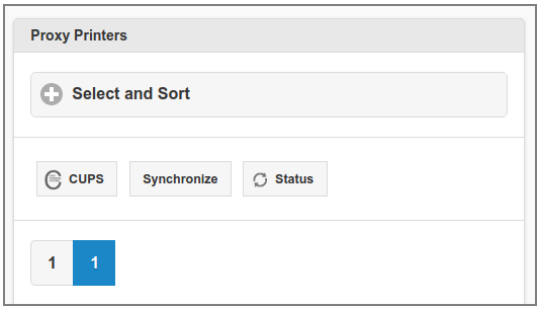


Figure 4.65. Admin Web App: Proxy Printer - List Header

- The CUPS button is shown when the browser points to "localhost", "127.0.0.1", a ".local" address, or to the intranet IP address of the SavaPage server, and host port matches the configured SavaPage port. Click the button to open the CUPS Administration web page in a new browser tab.
- Click the Synchronize button to synchronize all CUPS printer options to SavaPage. Since SavaPage does not detect changed CUPS printer defaults, PPD driver and PPD Extension File content (yet), you need to perform this action after you change any of these parameters.
- The Status button is shown if Printer SNMP is enabled. Push the button to refresh toner and printer state information for all printers.

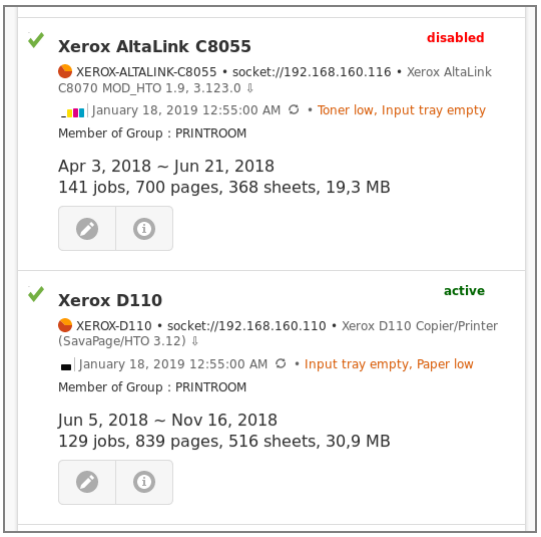


Figure 4.66. Admin Web App: Proxy Printer - List Items

- All non-deleted Proxy Printers are listed alphabetically by default. A different selection and sorting can be entered: see Figure 4.67, “Admin Web App: Proxy Printer - Select and Sort” [131].
- The list can be traversed by tapping one of the buttons at the *pager* at the top or bottom of the page.
- An entry is displayed for each Proxy Printer, with several identifying and usage data. Two sample items are show above (not all data apply). Possible items are, from top to bottom:
  - The Proxy Printer status (at the top right corner).
  - The display name (alias) of the printer as shown to the user.
  - An inline pagometer Pie-Chart followed by the CUPS printer name, URI and PPD driver name.
    - The *red* color in the chart represents the number of *pages* printed. The *orange* color represents the number of printed *sheets*.
    - A double-click on the CUPS *printer* name downloads a text file with detailed printer properties, and is used by SavaPage developers and administrators to debug PPD to IPP mapping issues.
    - The PPD Extension File configured in the Proxy Printer Edit Dialog is shown in green: a double-click on its name will download the PPDE file. See Appendix K, *PPD Extensions* [382] and Appendix L, *IPP Extensions* [397].
    - A double-click on the PPD *driver* name downloads the CUPS PPD file.
    - In case of a Raw Print the configured PPD File and (optional) PDL Transform File are shown: a double-click on their name will download the file.
  - If Printer SNMP is enabled, the toner and status information. Click on the icon at the most right to refresh this information. If the printer is marked as *not present*, this information is not shown.
  - The Printer Groups this Proxy Printer is member of.
  - The User Groups for which access is allowed or denied. See Section C.1.17, “printerAccessControl” [337].
  - An inline Line-Graph showing the day pagometers of the printed pages over the last 30 days.
  - An optional red colored “Configuration needed” tag, to remind you to configure at least one (1) of the printer's Media Sources. If the printer is marked as *not present*, configuration information is not shown. If marked as *disabled*, configuration information is colored in orange.
  - The period in which activity was accumulated on the pagometer.
  - The pagometer of the Proxy Printer including the number of jobs, sheets and bytes printed.
- Tap the Edit button to edit the entry. Proxy Printers which are present in CUPS can be edited: see Figure 4.68, “Admin Web App: Proxy Printer - Edit - Identity” [132]. Proxy Printers that are *not* present in CUPS can also be *deleted* or *renamed*: see Figure 4.77, “Admin Web App: Proxy Printer - Rename” [139]

- The Log button brings you to the list of documents printed to the Proxy Printer. See Figure 4.148, “Admin Web App: Document - List” [191]
- The Home button is shown under the same condition as the CUPS button. Click to open the CUPS Administration web page for the printer in a new browser tab.

Note

Due to Admin Privileges certain buttons might *not* be visible.

Important

The CUPS Administration web page must be accessible as explained in Section 2.3, “Step 3 - Configure CUPS and Samba” [12]. When CUPS authentication is required you can log in with user name `root` or with a user name that belongs to the admin group.

Tip

The pagometers of all Proxy Printers can be reset at Options → Advanced → Reset Pagometers

Each Proxy Printer in the list is marked with a *primary* icon:

	A non-secure Proxy Printer that **anyone** can print to. • See Section 4.11.9, “Proxy Print” [168].
	A non-secure Proxy Printer that **no one** can print to. • See Section 4.11.9, “Proxy Print” [168].
	A secured Proxy Printer whose jobs needs to be authorized with a NFC Card swipe on a Network Card Reader. • See Section 4.9.1.2, “Proxy Print Authentication” [142]. • A referral to the Reader and the enabled release functions are shown in the list item.
	A Proxy Printer that can only be used from certain Terminals. • See Section 4.9.2.1, “Custom Proxy Print” [146].
	A Proxy Printer that can only be used from certain Terminals and whose jobs needs to be authorized with a NFC Card swipe on a Network Card Reader on other Terminals.
	A Job Ticket Printer. • See Section 4.8.2.1, “Proxy Printer Identity” [132].

Table 4.6. Primary Printer Icons

When applicable, the following *secondary* icons are shown:

	A PaperCut Managed Printer. • See Appendix N, *PaperCut Integration* [412].
	Print Archive enabled. • See Section 4.8.1.2, “Print Archive” [130].
	Print Journal enabled. • See Section 4.8.1.3, “Print Journal” [130].

Table 4.7. Secondary Printer Icons

4.8.1.1. Printer SNMP

SavaPage can use the SNMP protocol to query information about networked printers, like type/model, serial number, toner and operation status. SNMP is disabled by default. Use the configuration items below to enable and set operational properties. If SNMP is enabled, printer data are read automatically at the daily maintenance cycle at 55 minutes past midnight. See Section O.4, “Daily Maintenance” [421].

Configuration property	Description
printer.snmp.enable	Enable SNMP: `Y` or `N` (default)
printer.snmp.read.trigger-mins	Number of minutes since last printer print after which a next print triggers a new SNMP read of printer data (default 240).
printer.snmp.read.retries	Number of retries after SNMP read failure (default 2).
printer.snmp.read.timeout-msec	SNMP read timeout in milliseconds (default 1500).
printer.snmp.marker.percent.warn	*reserved for future use*
printer.snmp.marker.percent.alert	*reserved for future use*

Table 4.8. Configuration Properties for Printer SNMP

See Section 4.11.15, “Config Editor” [189] on how to set these items.

Important

Avahi must be installed to retrieve the hostname of CUPS printers whose URI refers to a DNS-SD instance (`dnssd://` scheme) in order to establish an SNMP connection. See Section 1.2.1.10, “Avahi” [6].

Ensure that DNS-SD (Bonjour) is enabled on these printer devices and that IPP printing has the highest priority. Without these settings, SavaPage cannot trace the host name required for SNMP queries.

For example: in the web interface of an HP printer, enable Bonjour and set “Bonjour Highest Priority Service” to “IPP Printing”.

Tip

Use Server Command printerSnmp to SNMP query any printer available in the network.

4.8.1.2. Print Archive

The Print Archive is the `doc.store.archive.out.print` branch of the Document Store. It must explicitly be enabled.

4.8.1.3. Print Journal

The Print Journal is the `doc.store.journal.out.print` branch of the Document Store. It must explicitly be enabled.

4.8.1.4. DNS-SD Printers

Printers that advertise themselves via DNS-SD[8] (also known as Bonjour / Zeroconf) are *not* automatically part of CUPS. You can easily check: printer *devices* that show up with the **lpinfo -v** command do not show up as printer with the **lpstat -v** command.

[8] https://en.wikipedia.org/wiki/Zero-configuration_networking

Add your DNS-SD printers in CUPS by selecting them there from the list of “Discovered Network Printers”. Make sure to select a Model that uses CUPS filtering and is compatible with your printer. Do *not* select an IPP Everywhere variant as Model, as this will give you a "Local Raw Printer" throughput without filtering and print job options, which is unusable for SavaPage.

> **Tip**
>
> Placing your DNS-SD printers in a VLAN, which only the SavaPage server can access, prevents these printers from advertising themselves on user devices. If you subsequently add these DNS-SD printers in CUPS, users are forced to use them as Proxy Printer in the controlled environment of the SavaPage User Web App.

4.8.1.5. CUPS Printer Class

A CUPS Class is a group of printers. When printing to it, CUPS will redirect the job to one of the printer members. Which printer would depend on user rights or which printer currently is available.

A CUPS Class appears as a regular Proxy Printer in the list: the number of member printers is enclosed in parenthesis, just after its CUPS name. For CUPS class members, the CUPS classes to which they belong are shown in parentheses.

> **Important**
>
> SavaPage requires that CUPS Class members all have the same PPD.

4.8.1.6. Select and Sort

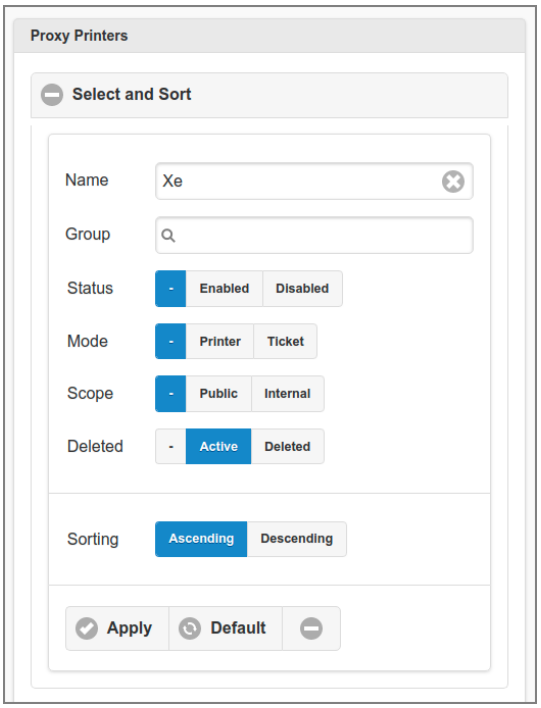


Figure 4.67. Admin Web App: Proxy Printer - Select and Sort

- Proxy Printers can be selected by entering a fragment of their CUPS or display **Name**. So, entering "Xe" will select both "Xerox AltaLink" and "Xerox D110".
- Select a **Status**, **Mode**, **Scope** or **Deleted** option. In each case the `-` button will select both option values.

- The list can be sorted Ascending or Descending on display name.
- *Tap* the Apply button to (re)display the list.
- A *tap* on the Default button resets the selection and sort fields to their default values.
- The minus button collapses the section.

4.8.2. Edit Proxy Printer

4.8.2.1. Proxy Printer Identity

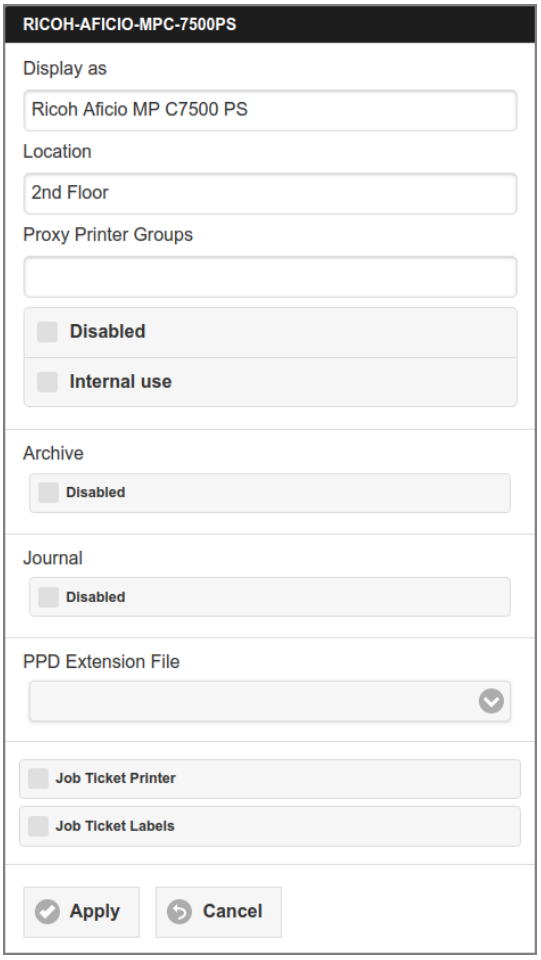


Figure 4.68. Admin Web App: Proxy Printer - Edit - Identity

- The title bar shows the CUPS Printer Name.
- The **Display** name and **Location** are editable.
- Multiple Proxy Printer Groups can be entered separated by space, comma, colon or semicolon.
- Tick the Disabled checkbox to disable access to the Proxy Printer for all use.
- Tick the Internal use checkbox to mark the printer for internal use only. This will disable access to the Proxy Printer from the User Web App, but the printer can still be used for internal print work flow scenarios.
- If the Print **Archive** and/or **Journal** function is enabled it can be Disabled for this printer. See Section 4.11.14.10, “Document Store” [185].
- If displayed, tick the Enabled checkbox to enable this printer as PaperCut front-end. See Section N.2, “Delegated Print to PaperCut Front-end” [416].

Figure 4.69. Admin Web App: Proxy Printer - PaperCut front-end

- If the Proxy Printer represents a Raw CUPS Queue, the **PPD File** and an optional **PDL Transform File** can be selected. See Section 4.8.4, "Raw Print" [138].

Figure 4.70. Admin Web App: Proxy Printer - Edit - Raw Print

- The optional **PPD Extension File** can be selected from a list with `.ppde` text files with extensions to the assigned PPD in CUPS. This is the way to map vendor specific PPD options to IPP attributes. See Appendix K, *PPD Extensions* [382].
  - Note: This file is required to configure a Raw Print.
- By ticking the Job Ticket Printer checkbox the printer will act as Job Ticket Printer and its display name will be shown to users with role Job Ticket Creator for printer selection. An entry field is shown where you must enter the **Proxy Printer Group** containing the printers the Job Ticket print job can be redirected to. See Section 4.8.3, "Printer Groups" [137] and Chapter 5, *Job Tickets Web App* [211].

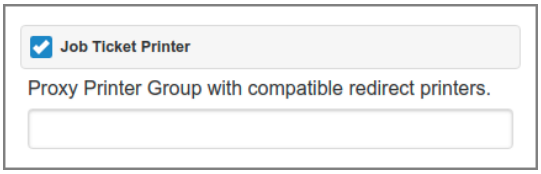


Figure 4.71. Admin Web App: Proxy Printer - Edit -Job Ticket Printer

This option can *not* be selected for PaperCut managed printers (a warning message is displayed).

- When the Job Ticket Printer option is *not* selected, you can still activate Job Ticket Labels, at least when they are enabled as configuration property: see Section 3.5.9.1, "Configuration Properties" [66] (when tags are disabled, the option is not shown). When this option is activated a user can select a Job Ticket Labels for this regular proxy printer: see Section 3.5.5, "Print Job Settings" [53]. The selected labels are shown as CUPS job ID prefix.
- Press the Apply button to commit the changes and return to the Proxy Printer List.
- The Cancel button brings you back to the Proxy Printer List without changing anything.

Important

When the content of the PPD Extension File is changed, you must Synchronize the Proxy Printers to take those changes into effect.

See Section 3.5.9, "Job Ticket Print" [64] for an explanation of Job Tickets and how users can create them. Also see Chapter 5, *Job Tickets Web App* [211] and Section A.3.1, "Delegated Print - (Non) Secure & Job Ticket Scenarios" [323].

4.8.2.2. Printer Costs

Printer costs are specified per media *side* and can be set for One-sided and Two-sided prints and differentiated for B/W (Black and White) and Color printing.

When SavaPage calculates the cost of a Proxy Print job, the two-sided (duplex) page cost is only applied to pages that are part of a sheet that is printed on both sides. So, for a document with an odd number of pages, the two-sided cost is not applied to the last page. For example, when a 7 page document is printed as two-sided, the two-sided cost is applied to the first 6 pages, and the one-sided cost to the last.

Costs are displayed, and can be entered, with a precision (number of decimals) as defined in Section 4.11.11.2, “General Financial Options” [173].

Printer costs are irrelevant and ignored in the following cases:

- When this is a Job Ticket Printer and Job Ticket Media Costs are defined in the **PPD Extension File**.
- When a printer is selected from the Job Ticket Print dialog (Job Ticket cost is leading).
- When PaperCut cost is leading. This is the case for a PaperCut Managed Printer, and when all conditions below are met:
  - This is a Non-Secure Printer (Secure Printing must be enforced by PaperCut). See Section 4.11.9, “Proxy Print” [168].
  - PaperCut Integration is enabled.
  - One of these two conditions is true:
    1. PaperCut Delegated Print Integration is *enabled*, and a Print Job is *not* charged to a Personal Account, but to a Shared Account or via the Delegated Print - PaperCut Scenario.
    2. PaperCut Delegated Print Integration is *disabled* and Personal Print to PaperCut is enabled,

  For a Document Log example, see Figure 3.53, “User Web App: Delegated Print - Group copies” [60].

4.8.2.3. Media Sources

In case Printer Costs are relevant, you can assign Media Size and Costs to Media Sources.

Warning

If no Media Source is configured, a red colored “Configuration needed” tag will show up for this printer in the Proxy Printer List and the printer will not be available for printing.

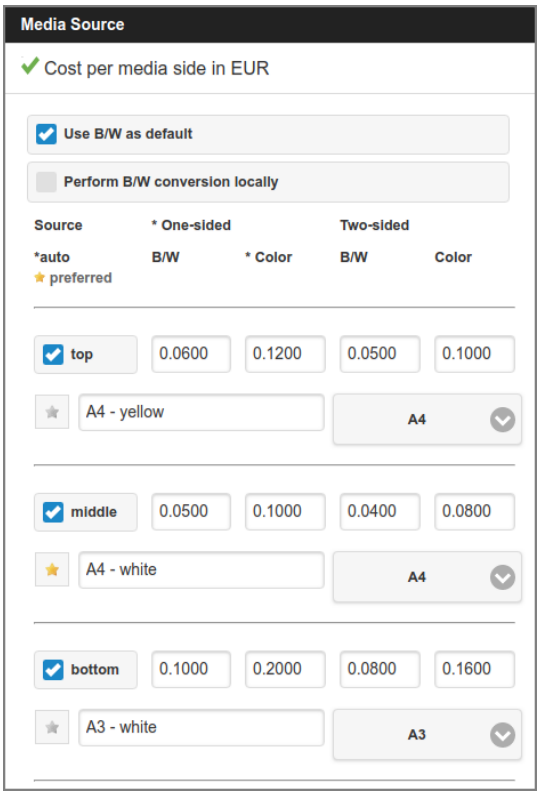


Figure 4.72. Admin Web App: Proxy Printer - Edit - Media Source

- The header text “Cost per media side ...” is prefixed with the same printer marker icon(s) as shown in the Printer List.
- CUPS printer defaults are indicated in the header with an asterisk (`*`). In some odd cases the CUPS printer color mode default “grayscale” is not correctly transferred as IPP attribute. You can easily correct this behavior by setting the Use B/W as default option at the top of the **Media Source** section. Instead, you can also use a PPD to IPP mapping to set the default for the IPP `print-color-mode` attribute: this correction is more basic, and is immediately reflected in de default `*` indicator.
- When the Perform B/W conversion locally option is set, grayscale print jobs are converted to B/W before sending them to a color capable proxy printer. This setting is needed when you observe that print jobs with color content are printed in color, despite the Grayscale Color Mode setting in the Printer Settings dialog. Indeed, some PPDs expect grayscale conversion is done client-side.
  - This option is not available for a Job Ticket Printer.
- The `*auto` text below the Source header indicates that the printer supports automatic tray selection based on media options.
- Each entry in the list has a checkbox with the IPP attribute keyword of the Media Source.
- Tick the checkbox to enable the media source, and enter a user-friendly name as will show up in the Media Source picklist of the Printer Settings dialog.
- Select the Media Size that is present in the Media Source.
- Specify the costs. Notice that only those cost cells are enabled that are applicable for the printer.
- You can mark a media source as *preferred* by pressing the “Star” button at the left. A preferred source is taken as default in situations where multiple media sources contain the same requested media size. For instance, in Job Ticket Redirect Printer settings.

Warning

Depending on the PPD file used for the CUPS printer, some media sources might not be applicable. You are advised to do some tests to make sure that media sizes are indeed applicable to the media sources as you intended.

Figure 4.73. Admin Web App: Proxy Printer - Edit - Job Sheet Media Sources

- When the printer is member of at least one Job Ticket Printer Group, and can therefore act as redirect printer, a media-source select menu for **Job Sheets** is shown. Multiple media-sources with assigned media-size can be selected.
  - When this printer is selected as redirect printer, the first **Jobs Sheets** media-source that matches the requested job-sheet media size will be selected as default. See Section 5.2.6, "Job Ticket Print" [217].
- The select menu is shown even if org.savapage-job-sheets is not configured in any job ticket PPDE file.

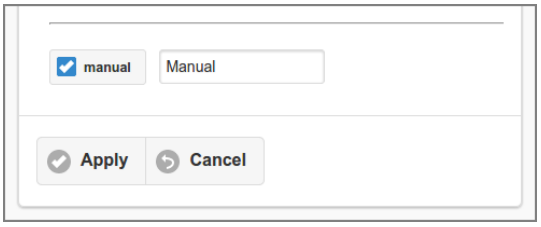


Figure 4.74. Admin Web App: Proxy Printer - Edit - Manual Media Source

- Costs for the *manual* media source can not be entered here, but must be specified as described in the next section.

4.8.2.4. Manual Media Sizes

In case Printer Costs are relevant, you can specify the Proxy Printer media costs for the *manual* media source. You can either use a `Simple` or `Advanced` definition.

Figure 4.75. Admin Web App: Proxy Printer - Edit - Manual Media Size (Simple)

The `Simple` definition allows for a single cost per media side. This is appropriate for a non-duplex monochrome Proxy Printer that can handle a single media size (Letter or A4) only.

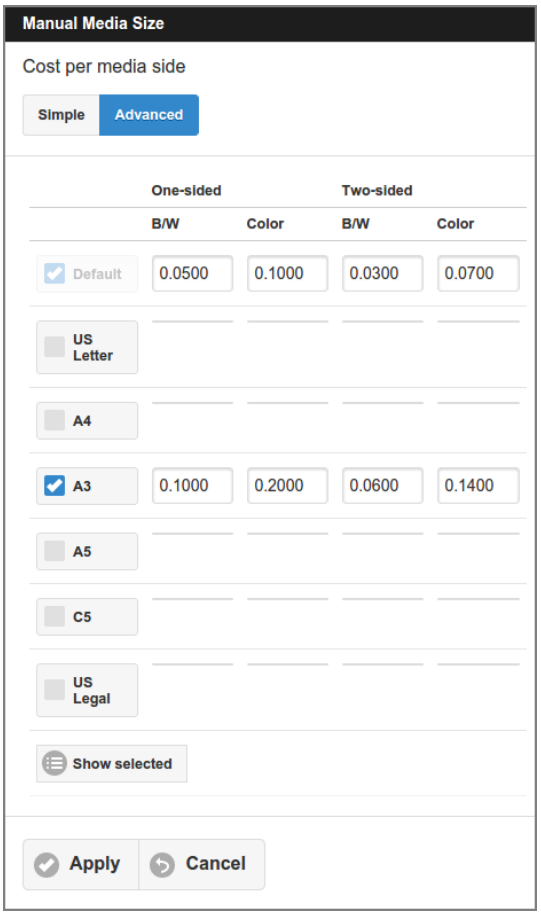


Figure 4.76. Admin Web App: Proxy Printer - Edit - Manual Media Size (Advanced)

Advanced mode is best suited for a duplex color Proxy Printer that can handle multiple media sizes.

- The list of supported media sizes is dependent on the Proxy Printer type.
- Use the check-box at a media size to enable its custom cost specification.
- Costs for unspecified (disabled) media sizes fall back to the **Default** specification.

4.8.3. Printer Groups

Printer Groups allow administrators to combine Proxy Printer instances so they can be addressed as group by a single *tag*. A Proxy Printer can have one or more groups tags. See Section 4.8.2, “Edit Proxy Printer” [132].

Printer Groups are used to customize access to Proxy Printers. See:

- Section 4.9.1.2, “Proxy Print Authentication” [142]
- Section 4.9.2.1, “Custom Proxy Print” [146]
- Section 4.11.9, “Proxy Print” [168]
- Section 4.8.2.1, “Proxy Printer Identity” [132].
- Section 5.2.1.1, “Job Ticket Group” [212].

Note

Printer Group tags are added to the database on first use. Tags without Proxy Printer members are removed from the database at the start of the application and thereafter in a Daily Job.

4.8.4. Raw Print

If the CUPS printer (or class) represents a Raw Queue, a **PPD File** and an optional **PDL Transform File** can be selected. See Section 4.8.2, “Edit Proxy Printer” [132].

4.8.4.1. PPD File

A **PPD File** must have the `ppd` file name extension and be located in the `/opt/savapage/custom/raw-print/ppd/` directory. The PPD is required so that SavaPage can generate the PDL and send it to the CUPS printer.[9] spoolfile. The **cupsfilter** command is used to produce the PDL.

Important

The **PPD File** is *not* interpreted by SavaPage to extract the IPP printer options. A PPD to IPP Mapping is needed and must be done manually, put in a `.ppde` file and selected as well. See Appendix K, *PPD Extensions* [382].

4.8.4.2. PDL Transform File

Preview

PDL Transform File is a Feature Preview that is partly operational and impermanent. The preview is offered to provoke feedback based on real world use. Please tell us your requirements.

A **PDL Transform File** can be applied to change the content of the PDL spoolfile before it is send to CUPS. The file has JSON format, must have the `json` file name extension and be located in the `/opt/savapage/custom/raw-print/transforms/` directory. Use of this type of file is optional.

The transform defines a simple `search` and `replacement` of words in the @PJL header. The `replacement` is a reference to one of the following IPP `PRINT_JOB` operation attributes: `requesting-user-name`, `job-name` and `copies`. For example:

```
{
  "@PJL" : {
    "header" : {
      "replace" : [ {
        "search" : "&user;",
        "replacement" : "requesting-user-name"
      }, {
        "search" : "&title;",
        "replacement" : "job-name"
      }, {
        "search" : "&copies;",
        "replacement" : "copies"
      } ]
    }
  }
}
```

[9] Page Description Language (PDL) is a computer language that describes the appearance of printed pages for a specific printer model or type, enabling the printer to produce the actual printout.

4.8.5. Rename Proxy Printer

If a Proxy Printer is removed from the host system it is marked in the list as *not present*. When editing new options appear, as is shown in the screenshot below.

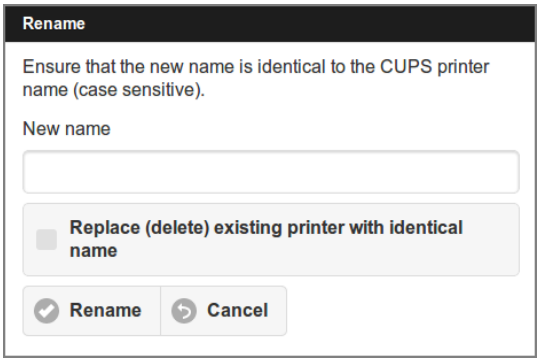


Figure 4.77. Admin Web App: Proxy Printer - Rename

- See Figure 4.68, "Admin Web App: Proxy Printer - Edit - Identity" [132] for a description of the basic edit options.
- Tick the Delete when job history is cleaned button to logically delete the Proxy Printer. It will be physically deleted from the database when no related job history is present anymore. This situation will automatically occur when you enabled automatic backup in combination with the delete of old document logs. *Deleting makes sense if the queue is permanently removed from CUPS*, and you don't want the Proxy Printer list in the Admin Web App to be cluttered with out-of-date "not present" Proxy Printers.
- Press the OK button to commit the changes and return to the Proxy Printer List.
- You can change the associated CUPS printer by entering a **New name**. *Renaming makes sense as a mirroring action of renaming a CUPS queue*. After renaming a printer in CUPS, the Proxy Printer associated with the old CUPS name will be identified by SavaPage as "not present", and a new Proxy Printer for the new CUPS queue will be created. At this point you can re-associate (rename) the old CUPS name of the Proxy Printer to the new one. This will work as long as no job history is already accumulated on the Proxy Printer associated with this new CUPS name. To overrule this constraint you can tick the **Replace (delete) existing printer with identical name** checkbox, so an existing Proxy Printer associated with the same (new) CUPS name will be deleted and replaced.
- Press the Rename button to commit the renaming action and return to the Proxy Printer List.
- Both Cancel buttons bring you back to the Proxy Printer List without changing anything.

Caution

If SavaPage detects a CUPS queue whose name is identical to a logical deleted Proxy Printer, the logical delete mark will be removed and the Proxy Printer will be re-activated.

4.9. Devices

A Device is a hardware component with a dedicated function.

SavaPage defines devices of type Terminal and *Network Card Reader*. They are identified by IP address and, in case of the reader, a port number. The combination of IP-address and Type must be unique.

Note

Although most of the time an IP address will harbor one device type, this constraint does allow that Terminal and Network Card Reader are combined on a single physical device.

- A Network Card Reader acts as *User Authenticator*, either at *Login* or *Proxy Print* time.
- A Terminal device runs customized SavaPage Web App Sessions overriding *global* User Authentication and Proxy Print defaults.

After a *tap* on the Devices button in the main menu this panel is shown. See Section 4.2, “Menu” [89].

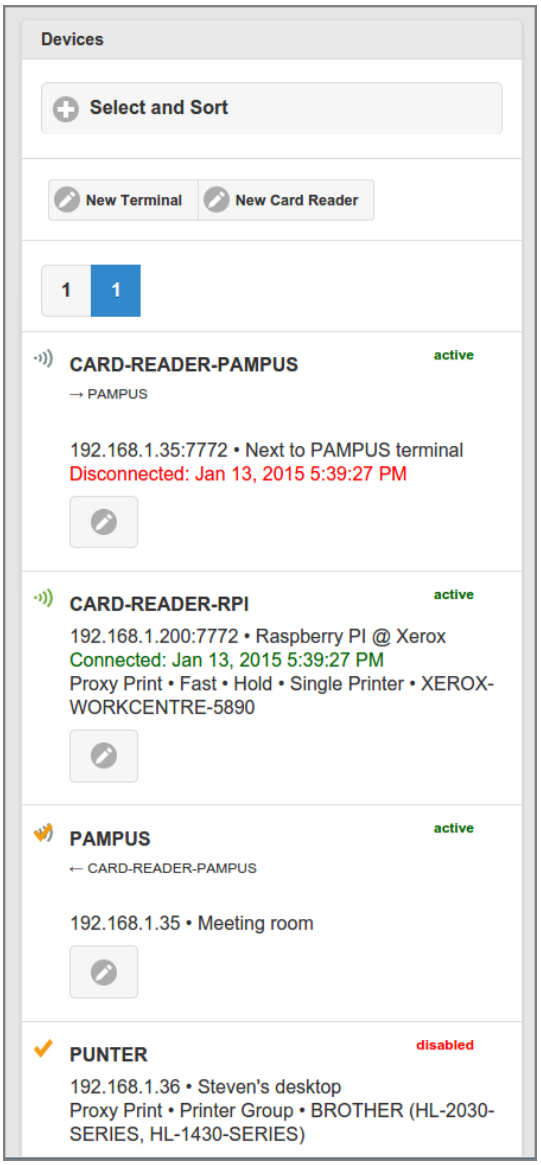


Figure 4.78. Admin Web App: Device - List

All Devices are listed alphabetically by default. A different selection and sorting can be entered: see Figure 4.79, “Admin Web App: Device - Select and Sort” [141]. Press the New Terminal... or New Card Reader... button to create a new Device.

Devices are marked with an icon and hold the following items:

A Network Card Reader used for NFC Card Login at a Terminal.

- The name of the Reader.
- A referral to the Terminal.
- The IP address, port, and location of the Reader.

- A time-stamp of the last connection attempt with the status *connected* (green) or *disconnected* (orange).

A Network Card Reader used for NFC Card *Proxy Print Authentication*.

- The name, IP address, port, and location of the Reader.
- A time-stamp of the last connection attempt with the status *connected* (green) or *disconnected* (orange).
- The Proxy Printer or Proxy Printer Group for which this reader acts as authenticator.
- The Print Mode of the proxy printer(s): Fast, Hold, Direct.

A Terminal with custom settings.

- The name, IP address and location of the Terminal.
- The Proxy Print or Proxy Printer Group for which this Terminal is entitled to print to.

A Terminal with custom settings and a Network Card Reader used for NFC Card Login.

- The name of the Terminal.
- A referral to the Reader.
- The IP address and location of the Terminal.
- The Proxy Printer or Proxy Printer Group for which this Terminal is entitled to print to.

Note

Due to Admin Privileges certain buttons might *not* be visible.

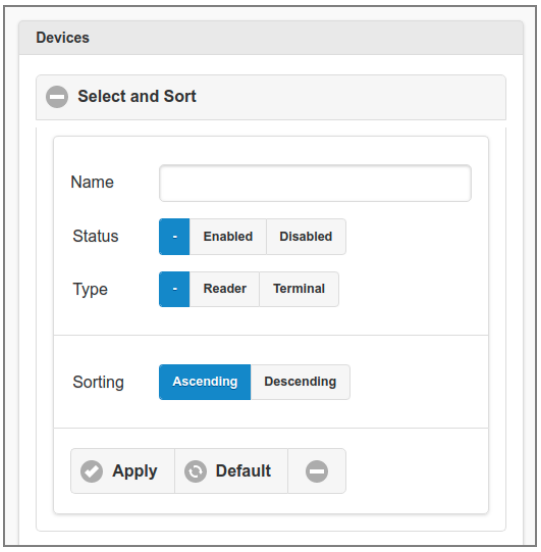


Figure 4.79. Admin Web App: Device - Select and Sort

- Devices can be selected by entering the containing text (fragment) of their **Name**. So, entering "CARD-" will select both "CARD-READER-PAMPUS" and "CARD-READER-RPI".
- Select the **Status**. The `-` button will select both Enabled and Disabled devices.
- Select the **Type**. The `-` button will select both Reader and Terminal devices.
- The list can be sorted `Ascending` or `Descending` on device name.
- *Tap* the `Apply` button to (re)display the list.
- A *tap* on the `Default` button resets the selection and sort fields to their default values.

4.9.1. Network Card Reader

A Network Card Reader device runs the SavaPage Network Card Reader Service .

4.9.1.1. Custom User Login

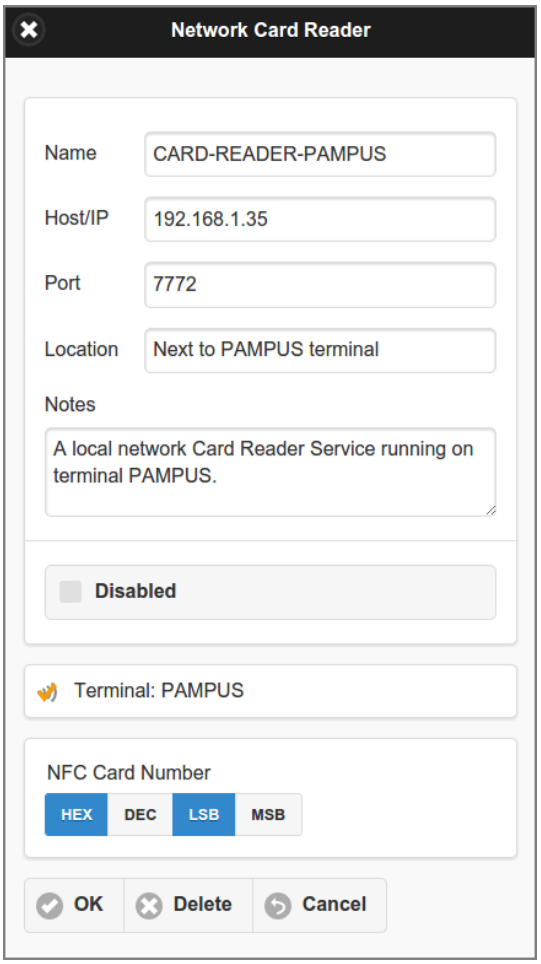


Figure 4.80. Admin Web App: Devices - Network Card Reader - Custom User Login

- A reader is associated to a Terminal in the Terminal dialog. See Section 4.9.2.2, “Custom User Login” [147]. The associated **Terminal** is shown here with icon and name.
- Tick **Disabled** to disable the Device definition.
- The format of the **NFC Card Number** must be specified. See Section B.1, “Card Number Format” [326].

> **Note**
>
> A Network Card Reader can be used as NFC Card Login authenticator by just one Terminal.

4.9.1.2. Proxy Print Authentication

A Network Card Reader can act as print job authenticator for a single Proxy Printer or a Proxy Printer Group.

When the reader device is placed next to the printer device this setup implements `Follow-me Printing`[10].

[10] https://en.wikipedia.org/wiki/Follow_Me_(printing)

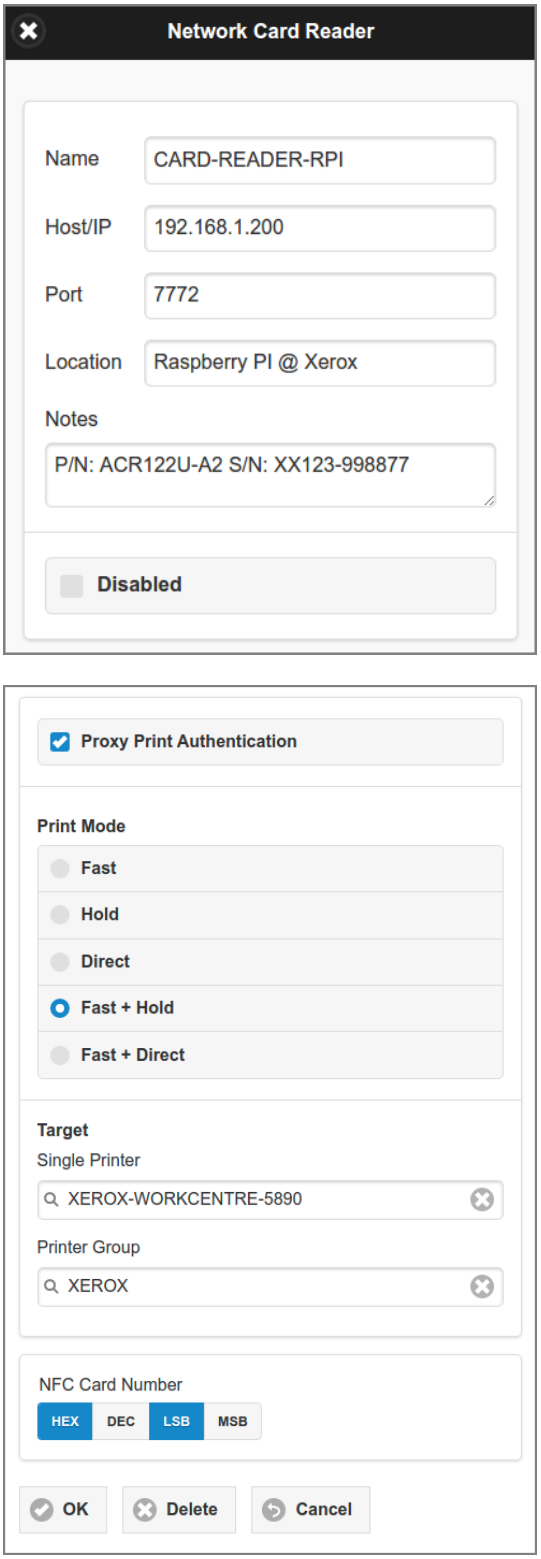


Figure 4.81. Admin Web App: Devices - Network Card Reader - Proxy Print Authentication

Checking the **Proxy Print Authentication** option shows all the detail options. The **Print Mode** determines the authentication scenario. Basically there are three modi: Fast, Hold and Direct. Fast Mode can be combined with Hold and Direct Mode. See Section 4.9.1.2.4, “Combining Print Modes” [145].

- Depending on the Print Mode, the name of a Single Printer and/or Printer Group **Target** must be selected. Which targets are needed is explained in the Print Mode subsections below.
- The format of the **NFC Card Number** must be specified. See Section B.1, “Card Number Format” [326].

Note

A Network Card Reader can be deployed for just one purpose. Either for Proxy Print Authentication or for Custom User Login at a Terminal Device.

Important

When using Proxy Print Authentication concurrently with the User Web App and User Client you are strongly advised to use an external database like PostgreSQL. See Chapter 21, *Using an External Database* [311].

4.9.1.2.1. Fast Print Mode

Fast Print Mode applies to a **Single Printer** and supports the following scenario:

1. User prints one or more jobs to SavaPage. See Chapter 14, *SavaPage as Printer* [260].
2. User walks to the printer.
3. User swipes NFC token along the reader.
4. As a result he gets one (1) printed copy of all SafePages jobs according to the default printer settings.

Note

If Inherit IPP options from print-in document is enabled, the IPP Options from PostScript Driver Print-in document(s) are used.

Expired SafePages documents are *skipped* and *cleared*. The expiry period is set in Section 4.11.9, “Proxy Print” [168]. A user can extend fast job expiry in the User Web App. See Section 3.5.1, “Printer Selection” [47].

Expired documents whose deletion is not triggered by Fast Print release are eventually deleted during User Home Clean.

- A Fast Print clears *all* SafePages after printing.
- This scenario does not need any action in the User Web App. Therefore, it is the shortest way to proxy print with SavaPage.

Caution

Fast Print Mode is meant for straightforward proxy printing without intervening editing. So, when users open the Web App and perform a Delete or Sort on their SafePages they are supposed to use Hold Print instead. When a Fast Print is tried on edited content, nothing is printed and all SafePages will be cleared as a result.

Important

When Fast Mode is combined with Hold or Direct Mode, and a **Printer Group** is specified, one of the printers from the group must be specified as **Single Printer**.

Tip

For Fast Mode printers with a single tray holding A4 media, PPD Extensions can be used to configure an extra virtual tray for Letter media. See Section K.6.1, “Fast Print A4 and Letter to Single Tray” [396].

4.9.1.2.2. Hold Print Mode

Hold Print Mode applies to a **Single Printer** or **Printer Group** and supports the following scenario:

1. User prints one or more jobs to SavaPage. See Chapter 14, *SavaPage as Printer* [260].
2. User opens the User Web App and proxy prints to either the Single Printer or one of the printers from the Printer Group.
3. As a result the proxy print job is held.
4. User walks to the chosen printer.
5. User swipes NFC token along the reader.
6. As a result all hold jobs for the printer are printed.

Expired hold jobs are *skipped* and *cleared*. The expiry period is set in Section 4.11.9, "Proxy Print" [168]. A user can delete hold jobs or extend expiry in the User Web App. See Section 3.3.2.3, "Hold Print Jobs" [33].

Expired hold jobs whose deletion is not triggered by print release or an active user session are eventually deleted during User Home Clean.

4.9.1.2.3. Direct Print Mode

Direct Print Mode applies to a **Single Printer** or **Printer Group** and supports the following scenario:

1. User prints one or more jobs to SavaPage. See Chapter 14, *SavaPage as Printer* [260].
2. User opens the User Web App in their mobile device.
3. User selects either the Single Printer or one of the printers from the Printer Group. See Section 3.5.1, "Printer Selection" [47].
4. User selects printer and job settings.
5. User walks to the chosen printer.
6. User presses the Print button in the Web App.
7. As a result a dialog pops up prompting the user to swipe an NFC Card for authentication. See Section 3.5.6, "Direct Print Release" [57]. User has a 10 second time limit to swipe their card. The time limit is set in Section 4.11.9, "Proxy Print" [168].
8. User swipes NFC token along the reader.
9. As a result the job is printed.

4.9.1.2.4. Combining Print Modes

Combining Fast and Hold Print Mode either in *one* proxy printer, or over *different* printers, needs special attention. Namely, if a user creates a Hold Print Job and does *not* delete the origin SafePages, the same SafePages will remain the target of a Fast Print, leading to duplicate prints. As this may be intended in rare cases, in the majority of cases this will not be intended by the user. To protect the user from unintended duplicate printing the following rules are applied:

- If Hold jobs exist for **any** printer, a Fast Print is only done when Proxy Printing is configured to "Always remove SafePages after printing". This way, because all SafePages are cleared after creating a Hold Job, we know for sure that present SafePages are not part of any Hold Print Job. See Section 4.11.9, "Proxy Print" [168].
- As a consequence, when Proxy Printing is **not** configured to "Always remove SafePages after printing", *all SafePages will be ad-hoc cleared after a Hold Print release*. This will prevent that, after all Hold jobs are released at Hold-only printers, and no Hold jobs exist for **any** printer, remaining SafePages will be Fast Printed anyway.

The Fast + Hold Print Mode supports the following scenario:

1. User creates a Hold Proxy Print Job.
   - All SafePages are removed after printing.

2. User prints one or more jobs to SavaPage.
3. User walks up to the Fast + Hold target printer of the Hold Print Job.
4. User swipes NFC token along the reader.
5. As a result:
   a. All Hold jobs for the printer are printed.
   b. One (1) copy of all SafePages jobs is printed with the default printer settings.

When a user created Hold Print jobs for different printers, each supporting Fast + Hold Print Mode, obviously the Fast Print occurs at the first printer visited.

4.9.2. Terminal

A Terminal runs customized SavaPage Web App Sessions on a specific device.

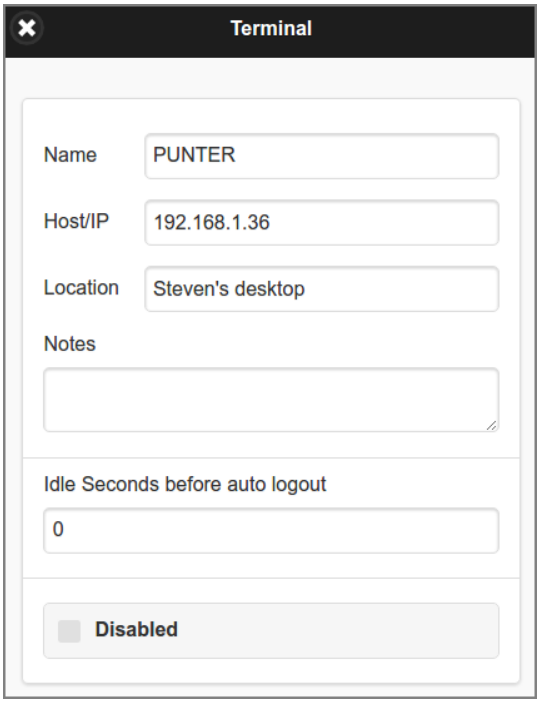


Figure 4.82. Admin Web App: Devices - Terminal - Custom Proxy Print

- The **Idle Seconds before auto logout** is targeted at public Terminals and is meant to protect users who forget to logout when done. Enter a number of seconds: the Web App will automatically logout after this period of inactivity. Enter 0 (zero) when no auto logout is desirable.

4.9.2.1. Custom Proxy Print

A Terminal can restrict printing to a single Proxy Printer or Proxy Printer Group. This is usually done for printers that need to be secured according to *global* Proxy Print defaults for Non-Secure Proxy Print. When the Terminal device is placed next to a (group of) printer(s), this setup implements `Pull Printing`[11].

[11] https://en.wikipedia.org/wiki/Follow_Me_(printing)

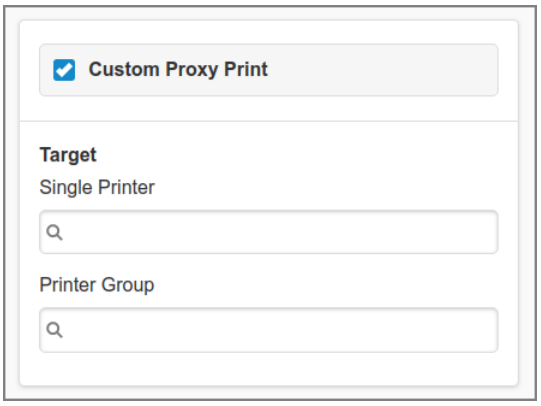


Figure 4.83. Admin Web App: Devices - Terminal - Custom Proxy Print

Check the **Custom Proxy Print** option and search/select the name of a Single Printer *or* Printer Group **Target**.

Note

A Proxy Printer assigned to a terminal, directly or by Printer Group membership, can not be accessed on other terminals.

Important

Assigning printers to a terminal does not automatically imply that these printers are available to all users. Eventually, Printer Access Control, User Roles, User Group Roles and User Group Privileges, determines whether or not a user sees a printer.

Tip

A Proxy Printer Group can have members that all point to the same printer device. By restricting access to each member with Printer Access Control you can show a single printer with a tailored set-up to specific User Groups.

4.9.2.2. Custom User Login

Check the **Custom User Login** option to override *global* User Authentication defaults just for this Terminal.

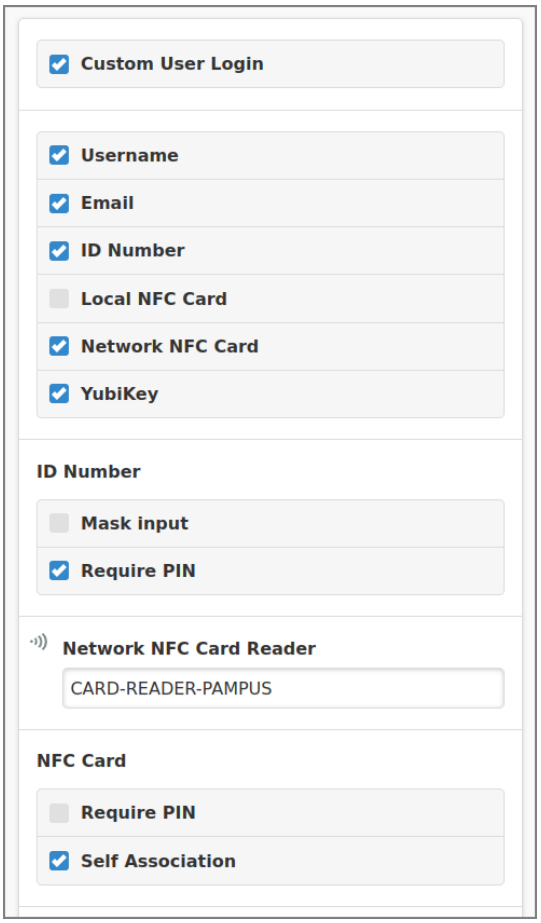


Figure 4.84. Admin Web App: Devices - Terminal - Custom User Login

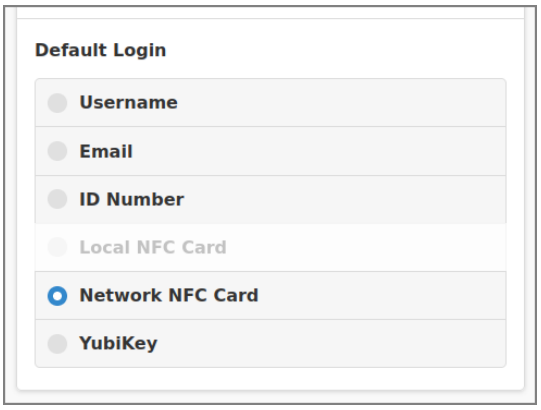


Figure 4.85. Admin Web App: Devices - Terminal - Custom User Login - Default

- The options in these section are identical to the ones in Section 4.11.4, “User Authentication” [157] with the addition of the **Network NFC Card** option.
- Enter the name of the Network Card Reader below the **Network NFC Card Reader** label.

Note

A Network Card Reader can be deployed for just one purpose. Either for Proxy Print Authentication or for Custom User Login at a Terminal Device.

4.10. Reports

Note

This section contains selections for extra PDF reports and CSV exports, apart from the ones presented in other sections, like Users. Currently a single report is defined. Please tell us your requirements so we can add more.

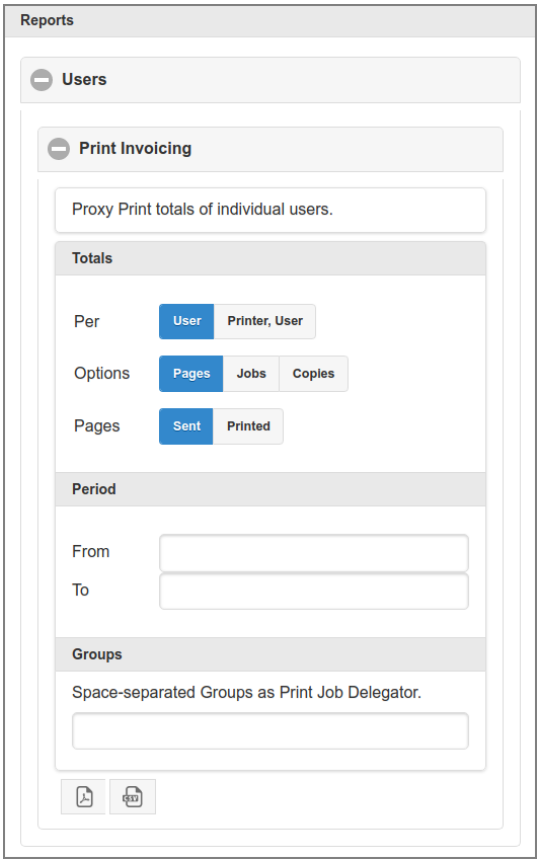


Figure 4.86. Admin Web App: Print Invoicing Report

Print Invoicing shows the proxy print cost **Totals** of individual users within a time **Period**, optionally from selected User Groups that acted as Delegator in a Delegated Print scenario. Lines are ordered **Per** User (id) or Printer, User (Printer display name, User id), and specify the cost total of completed print jobs within the period and extra data like Delegator Group and total number of Pages (Sent or Printed), Jobs or Copies per job type, like duplex/simplex, color/grayscale, page format A4, A3, etc.

4.11. Options

A *tap* on the Options button in the main menu shows a panel options organized in sections. A *tap* on one of the sections expands (or collapsed) the underlying details. Please see the sections below for a detailed description:

- Section 4.11.1, “User Source” [150].
- Section 4.11.3, “User Creation” [155].
- Section 4.11.4, “User Authentication” [157]
- Section 4.11.5, “Mail” [162].
- Section 4.11.6, “Mail Print” [164].
- Section 4.11.7, “Web Print” [166].
- Section 4.11.8, “Internet Print” [167].

- Section 4.11.9, “Proxy Print” [168].
- Section 4.11.10, “Eco Print” [170].
- Section 4.11.11, “Financial” [172].
- Section 4.11.12, “Integration” [175].
- Section 4.11.13, “Backups” [177].
- Section 4.11.14, “Advanced” [178].

Note

Due to Admin Privileges the Option panel might be *disabled* and certain buttons *not* visible.

4.11.1. User Source

This section is about the configuration of external and internal sources for user data and authentication.

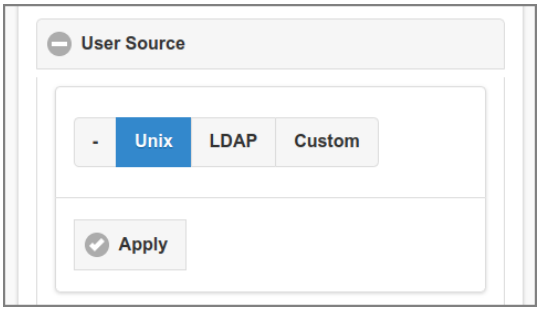


Figure 4.87. Admin Web App: Options - User Source

- *Tap* the - button if you do *not* want to import users from a standard external source.
  - If PaperCut is enabled as User Source, it will show up below the radio buttons. See Section N.5.2, “PaperCut as User Source in SavaPage” [418].
  - In other cases, remember to enable the Internal Users feature if you want to acquire any user into the system.
- *Tap* the Unix button if you want to import Unix user accounts defined on the SavaPage host.
- *Tap* the LDAP button to import users from an existing LDAP domain. This includes OpenLDAP, FreeIPA, Apple Open Directory, Novell eDirectory, Microsoft Active Directory and Google Cloud Directory. When this option is selected the LDAP connection data are shown.
- *Tap* the Custom button if you want to interface with external user sources using custom programs (plug-ins).
- Press the Apply button to commit the changes.

Note

If the **User Source** changes, the restriction **Import users from group** is reset to [All Users]. See Section 4.11.3, “User Creation” [155].

4.11.1.1. Unix

This option imports user accounts that are setup and defined on the local system as standard Unix accounts or mapped into the system from a central directory service such as LDAP via `nsswitch.conf` and PAM. Most large established Unix networks will support this option.

4.11.1.2. LDAP

LDAP (Lightweight Directory Access Protocol) directories usually store information about user and groups in an organization. One of the most common uses of LDAP is to provide single sign-on on a network that comprises multiple platforms and applications. When a network consists of Windows computers only, then an Active Directory domain

can be used. But when there is a mix of Windows, Apple and GNU/Linux machines then LDAP can provided the single source of user, group and authentication information. (It is worth noting that both Active Directory and Novell eDirectory implement the LDAP protocol).

SavaPage can use an LDAP directory for user authentication and as a source of user and group information. LDAP can either be enabled at installation time, or by changing the user source at this point. When enabling LDAP, a number of configuration properties must be specified to allow the application to connect to the LDAP server. Please ask your LDAP administrator what values to use for the various options.

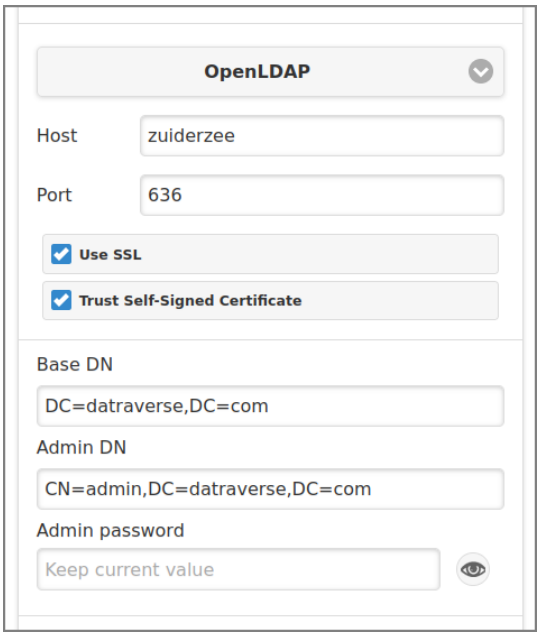


Figure 4.88. Admin Web App: Options - User Source - LDAP

The Server Type determines which LDAP fields are used to get user and group information. Select one of the following server types that SavaPage supports out-of-the-box:

- OpenLDAP.
- FreeIPA.
- Novell eDirectory.
- Apple Open Directory.
- Microsoft Active Directory.
- Google Cloud Directory.
  - This option is available when a Google LDAP Client Certificate is present.

When needed it is easy to support other server types by adjusting the fields SavaPage uses for LDAP searches. This is discussed in Appendix J, *Advanced LDAP Configuration* [376].

Enter the IP parameters (for Google Cloud Directory parameters are read-only).

- Enter the hostname or IP address of the LDAP server at the **Host** prompt.
- Enter the IP port of the LDAP server at the **Port** prompt. The value defaults to `389`.
- Tick the Use SSL checkbox to use encrypted SSL connection to connect to the LDAP server. The LDAP server requires SSL support to be enabled and should accept connections on the standard LDAPS port `636`.
- In case you connect to a Global Catalog server (Microsoft Active Directory) [12] use **Port** `3268` or `3269` (SSL).
- Check if you want to Trust Self-Signed Certificate of the LDAP server.

[12] A Global Catalog (GC) server is a domain controller that stores copies of all Active Directory objects in the forest. It stores a complete copy of all objects in the directory of your domain and a partial copy of all objects of all other forest domains. Global catalogs allow users and applications to find objects in any domain of the current forest by searching for attributes included in the GC.

And the regular settings:

- Enter the Base DN of the LDAP server at the **Base DN** prompt. This is the equivalent of the “suffix” config setting of the OpenLDAP server. For example, if the domain hosted by the LDAP server is “domain.com” then the Base DN might be: `DC=domain,DC=com`

  The format of the Base DN can differ significantly depending on the configuration. Some older Novell eDirectory installations may require a blank Base DN to operate. Some examples:

  ```
  DC=myorganization,DC=com
  DC=mycompany,DC=co,DC=uk
  OU=OrgUnit,DC=domain,DC=com
  DC=local
  ```

- The **Admin DN** is the DN of the user who has permission to connect to and query the LDAP server. This is typically an administrative user, although it can be a user that only has read-only access to the LDAP server. An example of the DN of the Administrator user on a Windows AD domain "domain.com", would be `CN=Administrator,CN=Users,DC=domain,DC=com`. The exact format of the DN depends on the LDAP server. Some examples:
  - *Microsoft Active Directory (in organizational unit)*

    ```
    CN=administrator,OU=OrgUnit,DC=domain,DC=com
    ```

  - *Apple Open Directory*

    ```
    uid=diradmin,CN=users,DC=domain,DC=com
    ```

  - *OpenLDAP*, *FreeIPA*

    ```
    uid=root,DC=domain,DC=com
    uid=ldapadmin,DC=domain,DC=com
    ```

  - *Microsoft Active Directory*

    ```
    CN=Administrator,CN=Users,DC=domain,DC=com
    ```

  - *Novell eDirectory*

    ```
    CN=root,DC=domain,DC=com
    CN=ldapadmin,OU=users,DC=domain,DC=com
    ```

- The **Admin password** is the password for the administrator specified in the Admin DN above. *Leave empty to keep the current value.* Use the Eye button on the right side of the entry field to show/hide the password.

> **Tip**
>
> Some LDAP servers are configured to allow “anonymous” LDAP query access. In these situations, the **Admin DN** and **Admin password** can be left blank.

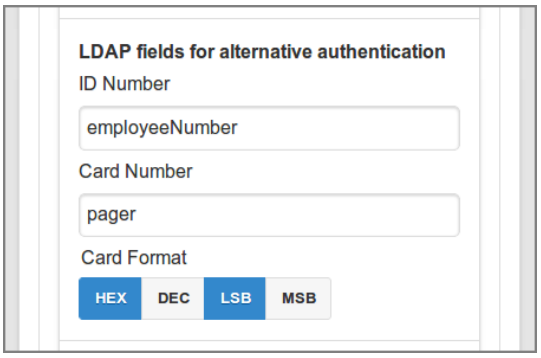


Figure 4.89. Admin Web App: Options - User Source - LDAP

At the **LDAP fields for alternative authentication** section LDAP field names can be entered for the two alternative user authentication methods **ID Number** and **Card Number**, as described in Section 4.11.4, “User Authentication” [157]. Field names are optional and can be left empty. The **Card Format** is relevant when the **Card Number** is specified. See Section B.1, “Card Number Format” [326].

Important

The ID and Card Number must each uniquely identify a user, so make sure that no two users have the same number. This means that the numbers defined in LDAP should be unique. If SavaPage encounters a non-unique ID or Card Number that user will not be updated.

Note

ID Number and **Card Number** LDAP fields are currently not supported for Google Cloud Directory.

4.11.1.3. Internal Users

With the internal users feature you can directly manage users inside SavaPage. Enabling this feature removes the obligation to define an external User Source to create and manage Users. Of course you can enable this feature as an addition to an external source.

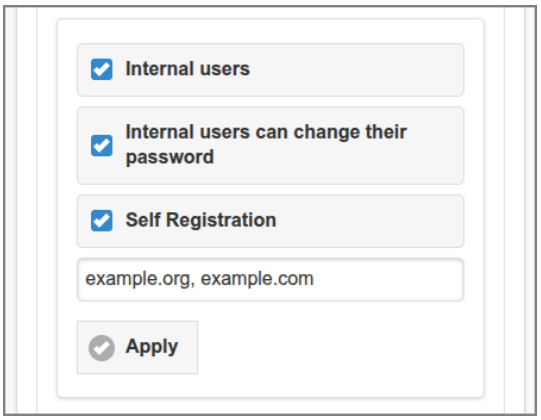


Figure 4.90. Admin Web App: Options - Internal Users

When **Internal Users** is selected extra options appear to enable **Internal Users can change their password** and **Self Registration**. See Section 3.10, “User Details” [78] and Section 3.1.3, “Self Registration” [24].

To enable **Self Registration**, you must also enter a comma-separated list of email domains that users can use to register. In addition you can set configuration properties to override defaults.

Configuration property	Description
internal-users.registration.ip-addresses-allowed	A CIDR Set of Client IPv4 and/or IPv6 addresses from which it is allowed to Register. Default: void (registration is allowed from *all* clients addresses).
internal-users.registration.email.expiry-mins	The number of minutes within which a pending registration must be verified. Default: `60`.
internal-users.registration.verify-url.authority	The authority of the verification link. Default: void (the authority of the User Web App URI is used).

Table 4.9. Self Registration: Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these items.

Tip

Use the Server Command Tool to batch import internal users. See Section C.1.2, “addInternalUser” [331]

4.11.1.4. Internal Groups

SavaPage has the ability to define internal user groups. Just like internal users these groups are internal to the SavaPage system. Internal groups are created in addition to groups already provided by the external user source and are useful in the following situations:

- You have configured the system to import users from a source that does not support groups.
- You do not have permission to create new groups in the user source.
- You'd like to create small groups just for use within SavaPage and it's not appropriate to great a new group in the user source.

Internal Groups are defined in a plain text file and composed of members who are either synchronized from the external user source or who were created as internal user. A fully annotated template text file is present here:

```
/opt/savapage/server/data/conf/internal-groups.txt.tmpl
```

... copy this file to ...

```
/opt/savapage/server/data/conf/internal-groups.txt
```

... and start defining your groups.

Internal Groups are fully emancipated to their external fellows and can be moved in and out of scope. See Section 4.5.3, “Add & Remove Groups” [112].

Warning

Internal Groups should have a name distinctive to any groups defined in your external user source. If case of a name clash, the internal group takes precedence.

4.11.2. Custom

Some organizations use software systems other that Unix and LDAP as the primary source of user data and authentication. In these situations, SavaPage can use custom user **Synchronization** and **Authentication** Command Line Interface (CLI) programs (plug-ins). Program paths are relative to `/opt/savapage/server/ext` .

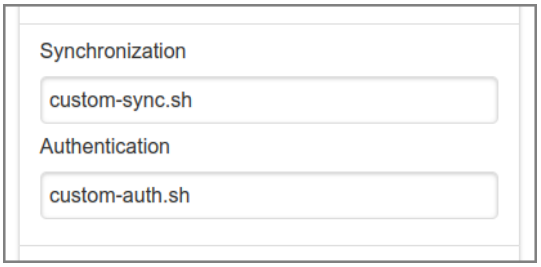


Figure 4.91. Admin Web App: Options - User Source - Custom

See Section M.6, “User Source Plug-ins” [410] :

- Section M.6.1, “User Synchronization Plug-in” [410]
- Section M.6.2, “User Authentication Plug-in” [411]

4.11.3. User Creation

This section is about the creation and synchronization of *external* users. *Internal* Users are created in the User Web App or with the Server Command Tool. See Section 4.4.6, “Create Internal User” [108] and Section C.1.2, “addInternalUser” [331].

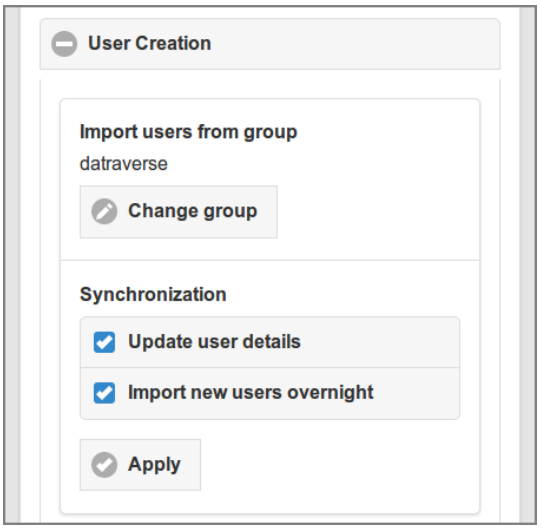


Figure 4.92. Admin Web App: Options - User Creation - Import

The **Import users from group** section holds an option to import a subset of users from the source by selecting a group. This option is relevant if you want to restrict access to SavaPage for a subset of *external* users.

- A *tap* on the Change group button shows a list of available groups as seen in Figure 4.93, “Admin Web App: Options - User Creation - From Group” [155].
- Select a group from the list and press the Apply button to commit the change, or press the Cancel button to roll it back.
- Use the [All Users] button to cancel the group restriction.

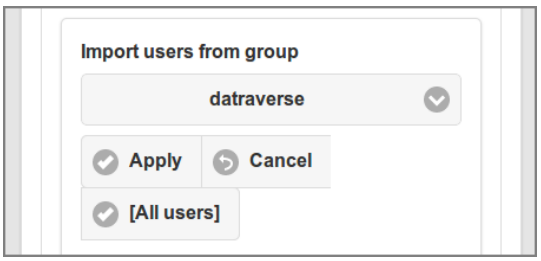


Figure 4.93. Admin Web App: Options - User Creation - From Group

Caution

In Active Directory, user group membership comes in two flavors. It can either explicitly be assigned, or be implied by the user's Primary Group ID, i.e. the RID of the primary group the user is member of. Because primary group membership is *implicit*, the Active Directory API will return an empty `member` attribute for this group. When users are *explicitly* assigned as member to groups the API will return group members as expected.

For example, because Active Directory sets the Primary Group ID of all users to the built-in “Domain Users” group, the Active Directory API will *not* report any members for the “Domain Users” group.

This issue is discussed in the following Microsoft Knowledge Base article: https://support.microsoft.com/kb/275523

Note

Active Directory supports an advanced feature called “Nested Groups”. This allows a group to have other groups as member. Since a sub-group can again have sub-group members, nesting can be several levels deep. When importing users from a group, SavaPage traverses the nested group tree to collect all containing users.

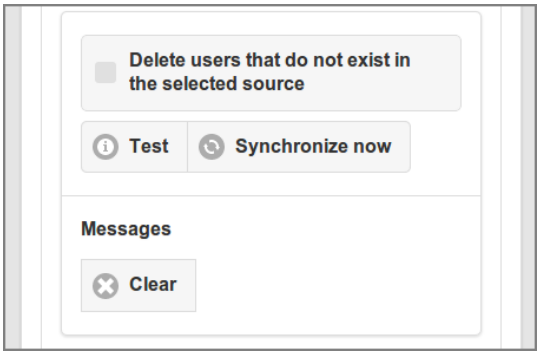


Figure 4.94. Admin Web App: Options - User Creation - Synchronize

The **Synchronization** section holds options for the *external* user synchronization process.

- Tick the Update user details checkbox if you want to overwrite user details in the database with details from the source.

Caution

An *external* User will overwrite an *internal* User with the same user id: as a result the User will become *external*.

- Select Import new users overnight to automatically synchronize daily at 10 minutes past midnight. See Section O.3, “Daily Jobs” [420].
- Press the Apply button to commit the changes.

Press the Synchronize now button to manually start a synchronization.

- Tick the Delete users that do not exist in the selected source checkbox to (logically) delete users in the database that were removed from the source. Note that this checkbox is deselected again after each synchronization.
- Feedback messages from the synchronization process are real-time displayed in the **Messages** section. Press the Clear button to remove them.
- Use the Test button to check the effect of the synchronization without updating the database. Messages are shown with a "test" prefix.

Note

Disabled Active Directory users will *not* be imported by default. If you want to change this behavior you can set the value of configuration key `ldap.disabled-users.allow` to `Y` (or enter an empty value to fall back to the default). See Section 4.11.15, “Config Editor” [189] on how to change this value.

Tip

To delete all external users, select - as User Source and use Synchronize now with the Delete users option.

Caution

The SafePages of *external* users not present in the source are deleted.

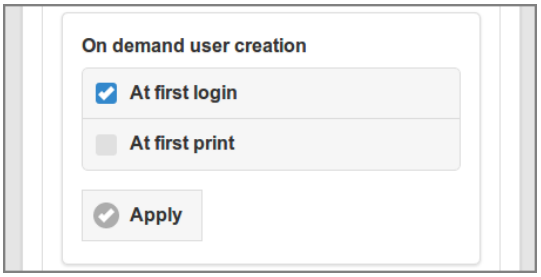


Figure 4.95. Admin Web App: Options - User Creation - On Demand

On demand user creation specifies which events, apart from regular user synchronization, will ad-hoc create new *external* users in the database.

- If the user associated with these events is not in the database, SavaPage will check if the user is part of the User Source and thereby a sure Synchronized User candidate. If so, it will ad-hoc synchronize the user into the database.
- Select At first login to ad-hoc create a user when they successfully Login to SavaPage for the first time.
  - This also includes an OAuth login if the OAuth provider uses the same (LDAP) user source that is configured in SavaPage as external User Source. This situation can be configured in the OAuth Plug-in properties file.
- Select At first print to ad-hoc create a user when they print to a SavaPage queue for the first time.
- Press the Apply button to commit the selection.

4.11.4. User Authentication

This sections describes the global defaults for User Authentication.

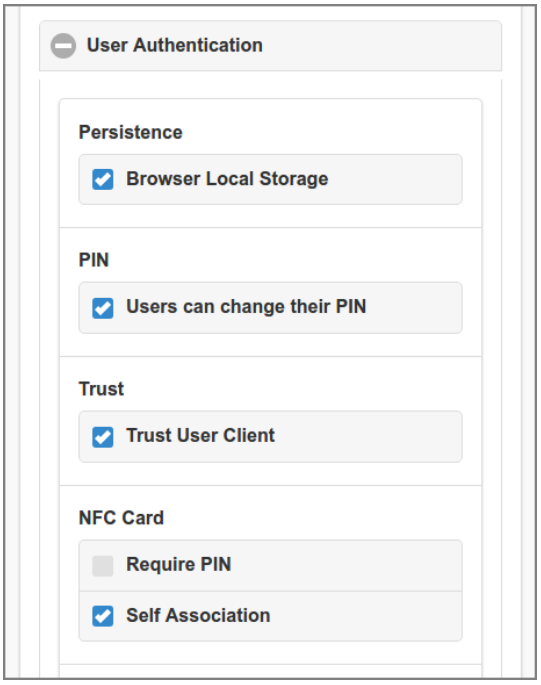


Figure 4.96. Admin Web App: Options - User Authentication

The **Persistence** section holds a options to persist authentication in Browser Local Storage. When *enabled*, a successful login to the SavaPage Web App will store an authentication token in the "Local Storage"[13] of the browser. When the user closes the browser after a successful login and opens it again, login will be automatic, without the need to authenticate again. Separate authentication tokens are held for the User and Admin Web App context. See Section 17.2.3, "Authentication Tokens" [287].

Note

The presence of an authentication token is essential for the iOS Web Clip to function, and is pure convenience in other environments.

When Browser Local Storage is *disabled*, authentication persistence is implemented with Web Sessions.

The **PIN** section holds the defaults for PIN usage.

- When **User can change their PIN** is enabled users are granted the option to change their PIN. See Section 3.10, "User Details" [78].

When Trust User Client is *enabled* User Web App authentication is automatic when:

- An authenticated User Client session is present at the same IP address.
- The User Web App is opened with an URL parameter identifying the user from the User Client session. See Appendix E, *URL Cheat Sheet* [362].

The **NFC Card** section holds the defaults for card swipe logins using a Local Card Reader or Network Card Reader.

- With **Require PIN** enabled the user must also provide their associated PIN. This provides additional security for swipe card logins.
- When the **Self Association** option is selected, users are allowed to swipe cards previously not used or registered. When swiping such an unregistered card, SavaPage will ask the user if to associate the new card to their account. When the user agrees SavaPage will switch to User Name authentication. After successful authentication the new card will be registered, thereby replacing any previously associated card. This feature is available for User Web App Login only. See Section 3.1.5, "Card Self Association Dialog" [26].

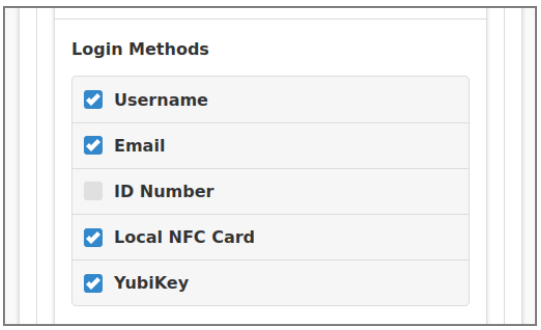


Figure 4.97. Admin Web App: Options - User Authentication - Login Methods

In the **Login Methods** section several login methods can be activated. When a method is activated a detailed section is shown. Detailed sections are explained in:

- Section 4.11.4.1, "Username Login" [159]
- Section 4.11.4.2, "Email Login" [159]
- Section 4.11.4.3, "ID Number Login" [159]
- Section 4.11.4.4, "Local NFC Card Login" [160]
- Section 4.11.4.5, "YubiKey Login" [160]
- Section 4.11.4.6, "Default Login" [161]

[13] Local Storage is a W3C standard and stores data in the browser with no expiration date. The data will not be deleted when the browser is closed, and will be available the next day, week, or year.

Important

The globally active Login Methods can be overridden for Access over Internet, and by defining *Custom User Login* settings for a specific Terminal.

Note

ID Numbers and NFC Card Numbers can be synchronized with an external source like LDAP, or imported from file.

4.11.4.1. Username Login

The **Username** login method allows a Person to use their regular username and password to login.

Figure 4.98. Admin Web App: Options - User Authentication - Username Login

If the **Show in Dialog** option is selected, the Username login method is part of the Login dialog. When this option is disabled this login can only be achieved by use of the `login` URL parameter. See Appendix E, *URL Cheat Sheet* [362].

4.11.4.2. Email Login

The **Email** login method allows a Person to use their email address and password to login.

Figure 4.99. Admin Web App: Options - User Authentication - Email Login

If the **Show in Dialog** option is selected, the Email login method is part of the Login dialog. When this option is disabled this login can only be achieved by use of the `login` URL parameter. See Appendix E, *URL Cheat Sheet* [362].

4.11.4.3. ID Number Login

The **ID Number** login method allows a Person to use their identity number. Identity numbers are convenient when usernames are too long or cumbersome to enter. For example, rather than entering a username like “steven.brown-002”, it is more convenient to enter the employee or student ID Number “3624”.

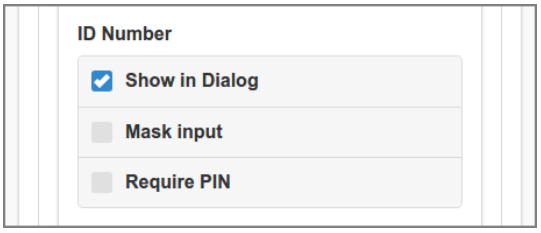


Figure 4.100. Admin Web App: Options - User Authentication - ID Number Login

If the **Show in Dialog** option is selected, the ID Number login method is part of the Login dialog. When this option is disabled this login can only be achieved by use of the `login` URL parameter. See Appendix E, *URL Cheat Sheet* [362].

When **Mask input** is set the ID Number will be masked when entered (like a password).

With **Require PIN** set the user must also provide their associated PIN. This provides additional security for ID Number logins.

4.11.4.4. Local NFC Card Login

The **Local NFC Card** login method allows a Person to login by swiping an NFC Card across a Local Card Reader.

Figure 4.101. Admin Web App: Options - User Authentication - Local NFC Card Login

If the **Show in Dialog** option is selected, the Local NFC Card login method is part of the Login dialog. When this option is disabled this login can only be achieved by use of the `login` URL parameter. See Appendix E, *URL Cheat Sheet* [362].

The **Format** of the card number must be specified. See Section B.1, “Card Number Format” [326].

> **Warning**
>
> This setting applies to any Local Card Reader hooked up to any device. If a card reader is used that produces a different format a Terminal definition with a *Custom User Login* needs to be created for the device the reader is hooked up to.

4.11.4.5. YubiKey Login

The **YubiKey** login method allows a Person to login with a YubiKey[14] Token.

Figure 4.102. Admin Web App: Options - User Authentication - YubiKey Login

If the **Show in Dialog** option is selected, the YubiKey login method is part of the Login dialog. When this option is disabled this login can only be achieved by use of the `login` URL parameter. See Appendix E, *URL Cheat Sheet* [362].

Get the YubiKey API credentials from yubico.com[15], and enter them as configuration property.

Configuration property	Description
auth-mode.yubikey.api.client-id	The YubiKey API client ID.
auth-mode.yubikey.api.secret-key	The YubiKey API secret key.

Table 4.10. YubiKey Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these items.

[14] https://www.yubico.com/
[15] https://upgrade.yubico.com/getapikey/

4.11.4.6. Default Login

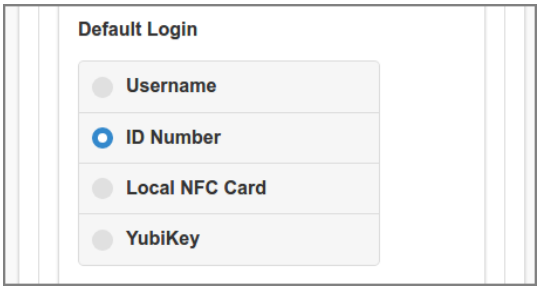


Figure 4.103. Admin Web App: Options - User Authentication - Default Login

Select the Login method that is displayed as default in the Login dialog.

4.11.4.7. 2-Step Verification

2-step verification is a method of confirming a user's claimed identity by utilizing an extra factor after a successful login as discussed in the previous sections. SavaPage uses Time-based One-time Password (TOTP) as described in RFC 6238[16].

Authenticator apps to generate TOTP codes are widely available.

Figure 4.104. Admin Web App: Options - User Authentication - 2-Step Verification

When **2-step verification** is enabled, users can open the Authentication Dialog to activate their TOTP secret. Then they can utilize their Authenticator app to generate TOTP codes by scanning the QR-code. Or instead, if Telegram TOTP is enabled (see table below), they can choose to have SavaPage send a TOTP code to their Telegram ID. See Section 3.10.2, “Telegram Configuration” [80].

Configuration property	Description
user.totp.issuer	Issuer in `otpauth://totp/[issuer]` URI. When void (default) your Organization name is used.
user.ext.telegram.totp.enable	Set to `Y` (default) or `N`, to enable/disable sending TOTP code to user Telegram ID.

Table 4.11. 2-Step Verification Issuer Properties

See Section 4.11.15, “Config Editor” [189] on how to set these properties.

Note

2-step verification is not applied when login to the Web App is implicit because of a redirect from a trusted OAuth provider, or trusted and authenticated User Client, or because of a One-Time Authentication Token.

Deprecated

Telegram support is **deprecated** and will be removed in a next release.

[16] https://tools.ietf.org/html/rfc6238

4.11.5. Mail

This section holds the settings for outgoing mail.

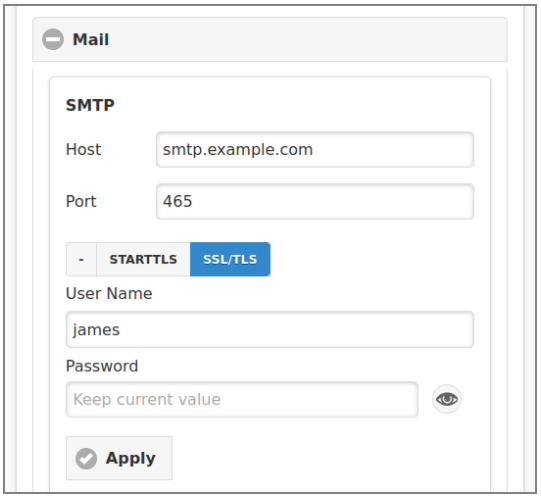


Figure 4.105. Admin Web App: Options - Mail - SMTP

Enter the **SMTP** connection parameters:

- The host name or IP address of the **Host**.
- The IP port at **Port**. The standard SMTP ports are: `25` (insecure), `465` (SSL/TLS) and `587` (STARTTLS). The value defaults to `465` (SSL/TLS).
- Select the connection security: **-** for an insecure connection, and **STARTTLS**[17] or **SSL/TLS**[18] for a secure connection. TLSv1, TLSv1.1 and TLSv1.2 are supported.
- Enter the **User Name** and **Password** if authentication is required. Use the Eye button on the right side of the **Password** field to show/hide the password. *Leave empty to keep the current value.*

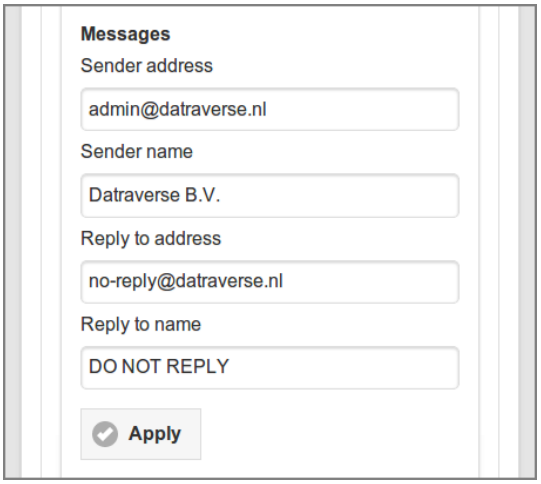


Figure 4.106. Admin Web App: Options - Mail - Messages

The **Messages** section holds the sender and reply parameters used for email messages send by the system:

[17] STARTTLS is a way to take an existing insecure connection, and upgrade it to a secure connection using SSL/TLS.
[18] SSL and TLS both provide a way to encrypt the communication between a client and a server computer. TLS is the successor to SSL and the terms SSL and TLS are used interchangeably.

- **Sender address** : enter a valid email address representing the sender of the message.
- **Sender name** : the name default to SavaPage.
- **Reply address** : enter a valid email address the recipient can reply to.
- **Reply to name** : the name to reply to.

Press the Apply button to commit the changes.

Important

The **Sender address** must be part of the OpenPGP key used for PGP/MIME email signing and encryption. See Section 13.3, “OpenPGP Settings” [247].

Warning

It is strongly recommended that you use Postfix[19], an Open Source Mail Transfer Agent (MTA), for SMTP email transmission. This is the reference SMTP server used in the SavaPage tests. Other (proprietary) SMTP services may not function as expected.

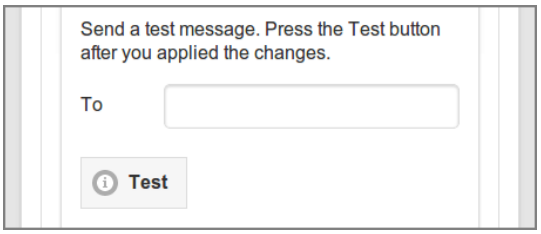


Figure 4.107. Admin Web App: Options - Mail - Test

Check if all mail parameters are valid by sending a test email.

- Enter a valid email address to send a message **To** and press Test. Check the mailbox of the recipient to see if the message arrived.

Timeout and timing parameters can be configured as shown in the table below.

Configuration property	Description
mail.smtp.connectiontimeout	Timeout in milliseconds until an SMTP connection is established. Default: 5000.
mail.smtp.timeout	Timeout in milliseconds to send mail messages after the SMTP connection is established, i.e. maximum time of inactivity between two data packets. Default: 5000.
mail.smtp.debug	Enable (Y) or disable (N) extra logging for troubleshooting SMTP problems. Default: N.
mail.outbox.poll.heartbeat-msec	Heartbeat (milliseconds) to poll the store-and-forward mail outbox for new messages. Default: 10000.
mail.outbox.send.interval-msec	Interval (milliseconds) between sending each mail message. A reasonable interval is needed to prevent mail being qualified as spam by the SMTP host. Default: 1000.

Table 4.12. Mail Configuration Properties

[19] https://www.postfix.org/

See Section 4.11.15, “Config Editor” [189] on how to set these properties.

4.11.6. Mail Print

Mail Print is an implementation of Driverless File Transfer Print which allows users to print documents by mailing them to SavaPage. The email address from the sender is used to find the corresponding Person. See Section 15.1.14, “Mail Print Authentication” [276].

> **Note**
>
> Mail Print maps to the reserved Queue `/mailprint`.

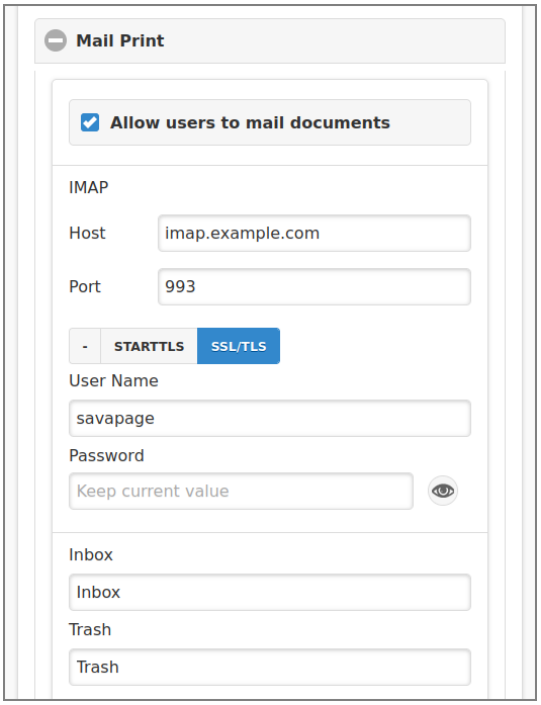


Figure 4.108. Admin Web App: Options - Mail Print (IMAP)

Check the Allow user to mail documents to enable the Mail Print function. Then enter the **IMAP** connection parameters:

- The host name or IP address of the **Host**.
- The IP port at **Port**. The standard IMAP ports are: `143` (insecure), `993` (SSL/TLS) and `143` (STARTTLS). The value defaults to `993` (SSL/TLS).
- Select the connection security: **-** for an insecure connection, and **STARTTLS** or **SSL/TLS** for a secure connection.
- Enter the **User Name** and **Password** for the required authentication. *Leave password empty to keep the current value*. Use the Eye button on the right side of the **Password** field to show/hide the password.

> **Note**
>
> If IMAP OAuth Plug-in is active, **User Name** and **Password** are replaced with text “PLUGIN” and the plug-in email address.

Important

The IMAP host *must* support the IDLE Command, which is a widely implemented standard extension to the core IMAP protocol. See RFC2177[20].

Warning

It is strongly recommended that you use Dovecot[21], an Open Source IMAP Server, to host Mail Print messages. This is the reference IMAP server used in the SavaPage tests. Other (proprietary) IMAP services may not function as expected.

Print jobs are read from the **Inbox** and moved to the **Trash** folder after processing. Enter the name of both folders:

- **Inbox** : the name of the Inbox folder.
- **Trash** : the name of the Trash folder.
  - When Trash is a *sub-folder*, enter a point separated folder path. For instance, when `Trash` is a sub-folder of `Inbox`, enter: `Inbox.Trash`

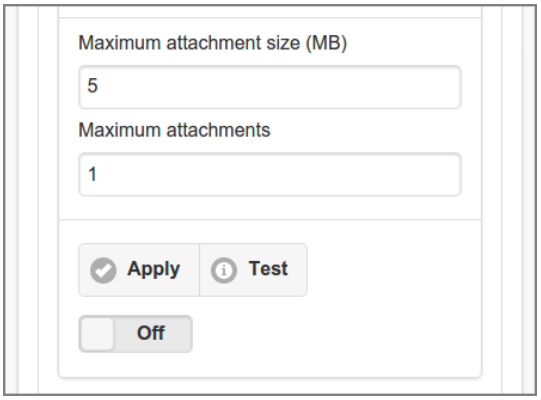


Figure 4.109. Admin Web App: Options - Mail Print (Attachments)

Limit the print job size per email message by setting the **Maximum attachment size (MB)** and **Maximum attachments**. Use integers as value. Leave empty to allow unlimited size.

- Press the Apply button to commit the changes.
- Press the Test button to test the connection. A feedback message will be displayed with the result.
- Use the flip-switch to turn the Mail Print service On and Off. Note that after disabling the service it is automatically turned Off.

Note

- Because Mail Print is an open channel, SavaPage does *not* reply to unknown users.
- For uploaded file types that do not have a page size defined (HTML, TXT) the default paper size is used.
- The Report Font is used for plain text files (TXT).
- See Section 14.9.1, “Processing Driverless PDF Transfer” [269] on how mailed PDF documents are processed.

[20] https://tools.ietf.org/html/rfc2177
[21] https://dovecot.org/

- When no file attachments are present the message body is printed.

Configuration property	Description
print-imap.content.eml.detain	Temporarily detain any printed message body as `.eml` file in the JVM Temporary Files folder. `OFF` = never detain (default), `ON` = always detain, `AUTO` = detain when conversion to `pdf` via `html`produces `stderr` messages.
print.imap.connectiontimeout-msecs	Socket timeout in milliseconds while establishing a connection to IMAP inbox folder. Default: 10000 milliseconds (10 seconds).
print.imap.session.duration-secs	Duration in seconds that the IMAP connection is monitored and kept alive by NOOP commands. When the duration has expired, the IMAP connection is closed and reopened for another period of the same duration. Default: 3000 seconds (50 minutes).
print.imap.session.heartbeat-secs	Interval in seconds between IMAP NOOP commands on inbox folder to keep IMAP connection alive. Default: 300 seconds (5 minutes).
print.imap.timeout-msecs	Socket timeout in milliseconds while communicating on an established IMAP connection. Default: 10000 milliseconds (10 seconds).

Table 4.13. Mail Print Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these properties.

4.11.7. Web Print

Web Print is an implementation of Driverless File Transfer Print which allows users to print documents to SavaPage by simply uploading them from their User Web App. See Section 3.11, “Upload” [85].

Note

Web Print maps to the reserved Queue `/webprint`.

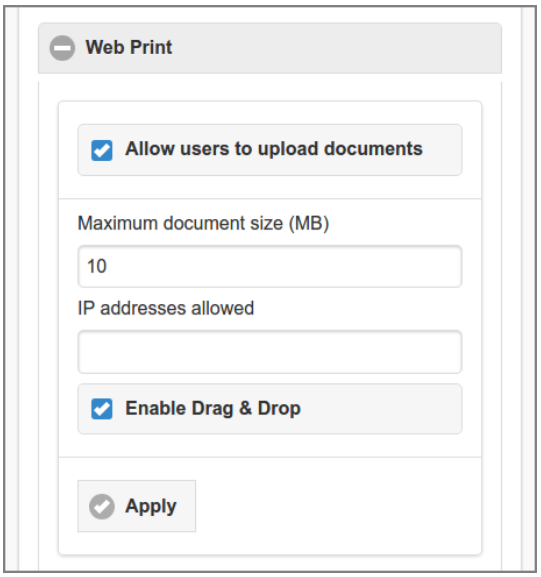


Figure 4.110. Admin Web App: Options - Web Print

Check the Allow user to upload documents to enable the Web Print function. Then enter the restriction parameters:

- Limit the print job size by setting the **Maximum document size (MB)**. Use an integer as value. Leave empty to allow unlimited size.
- Enter IPv4 and/or IPv6 address ranges as a CIDR Set at **IP addresses allowed** to restrict upload based on the requesting IP address. If the field is empty all requesting IP addresses are allowed to upload. See Section 13.5.5, "HTTP XFF Header Settings" [251].
- Enable Drag & Drop creates a Drop Zone in the User Web App main view. See Section 3.12, "Upload Drop Zone" [87].

Note

See Section 14.9.1, "Processing Driverless PDF Transfer" [269] on how uploaded PDF documents are processed.

4.11.8. Internet Print

Secure Driver Printing to SavaPage over public Internet is activated when port `443` of a public IP address is forwarded to port `8632` of the private intranet IP address of the SavaPage server. To authenticate the requesting user a special Printer URI format is used:

```
ipps://[host]/printers/internet/[number]/[uuid]
```

… where `[host]` is the public DNS name or IP address, and `[number]` and `[uuid]` are the ID Number and UUID of the user. See Section 4.4.5.5, "ID number" [105], Section 4.4.5.6, "UUID" [105] and Appendix E, *URL Cheat Sheet* [362].

Figure 4.111. Admin Web App: Options - Internet Print

Enter the `protocol://authority` of the Internet Printer Device URI as shown to users and press the Apply button to commit. When the value is left blank users won't be able to see their private Internet Printer Device URI. See Section 3.10.1, "Internet Printer" [79].

Important

Internet Print maps to the reserved Queue `/internet`. To accept prints from specific IP addresses see Section 4.7.2, "Edit Queue" [123].

Caution

Beware that by enabling Internet Print the SavaPage Web Apps are also accessible over public Internet, so take extra care to protect access to these Apps. See Section 17.3, "Access over Internet" [287].

4.11.9. Proxy Print

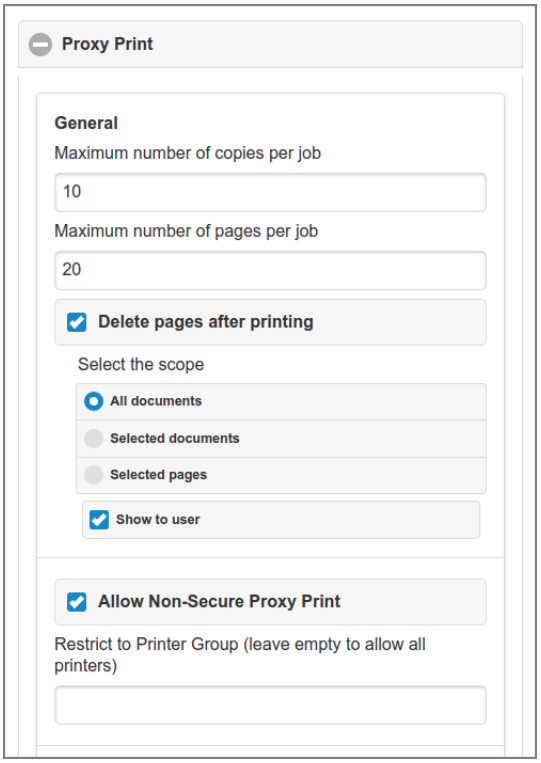


Figure 4.112. Admin Web App: Options - Proxy Print General

The **Maximum number of copies per job** restricts the number of *copies* a user can select in the Print Job Settings. Enter a positive number.

The **Maximum number of pages per job** restricts the number of *pages* for proxy print jobs. A proxy print job that exceeds this maximum will be denied. Leave empty to allow unrestricted printing.

To enforce that input documents or pages are deleted after a proxy print, enable **Delete pages after printing**, and select one of the options below. Also see Section 3.5.5, “Print Job Settings” [53].

- **All documents**: all input documents are deleted.
- **Selected documents**: documents for which pages were printed are deleted.
- **Selected pages**: all pages selected for printing are deleted.
- Check the Show to user option to inform the user about the delete scope in the Print Dialog.

Check the **Allow Non-Secure Proxy Print** option if you want to allow users to print to any enabled Proxy Printer from any device. You can optionally restrict non-secure printer access by entering a Proxy Printer Group.

Non-Secure means that users are able to initiate print jobs from locations (desktop, mobile device) remote from the actual printer. This implies that jobs will sit uncollected at the printer, at least for a while. In the mean time, prints containing sensitive information may be read by unauthorized eyes. Or jobs may be forgotten at all, adding up to paper and toner waste.

Any printer that falls beside the non-secure printer pool must be secured by Terminal or Network Card Reader Devices that have a fixed position at the target printer . See Section 4.9.1.2, “Proxy Print Authentication” [142] and Section 4.9.2.1, “Custom Proxy Print” [146].

Deprecated

Configuration item `proxy-print.repair.enable` is **deprecated** and will be removed in a next release. It defaults to `N`, but if set to `Y` a repair action is triggered just before a proxy print. The "font embedding" and overall cleaning of this pdftocairo repair will most likely secure a correct rendering in those cases where the **pdf2ps** filter executed by CUPS/Ghostscript will not. Use PDF cleaning during Driverless PDF Transfer Processing instead.

Tip

Read all about Appendix A, *Proxy Print Scenarios* [321].

4.11.9.1. Proxy Print Modes

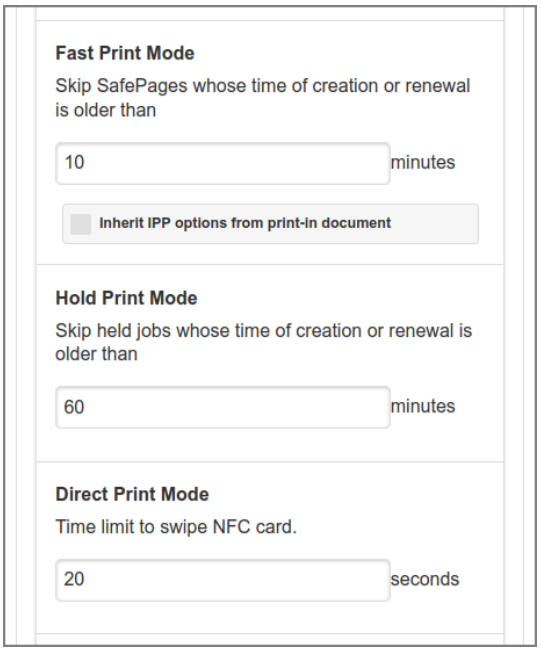


Figure 4.113. Admin Web App: Options - Proxy Print Modes

The expiration period for each Print Mode can be entered. See:

- Section 4.9.1.2.1, “Fast Print Mode” [144],

  Tick the **Inherit IPP options from print-in document** checkbox to use IPP options for Fast Print as captured from the PostScript Driver Print-in document. These IPP Options from PostScript are visible in Document Details and Document List Print-in items.
- Section 4.9.1.2.2, “Hold Print Mode” [145]
- Section 4.9.1.2.3, “Direct Print Mode” [145]

4.11.9.2. Proxy Print Delegation

Figure 4.114. Admin Web App: Options - Proxy Print Delegation

In this section you can Enable Delegated Print: see Section 3.5.4, “Delegated Print” [53] and Section A.3, “Delegated Print Scenarios” [322].

4.11.9.3. Proxy Print Preprocessing

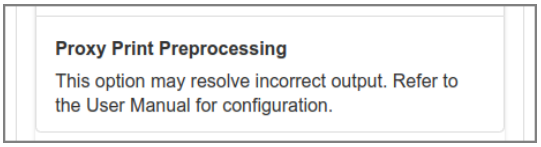


Figure 4.115. Admin Web App: Options - Proxy Print Preprocessing

If your Proxy Printer instances consistently deliver incorrect print output, preprocessing (converting) the PDF before sending it to CUPS could be a solution. The following global configuration properties apply for all printers:

Configuration property	Description
proxy-print.convert	`Y`, `N` (default). Set to `Y` to enable Proxy Print Preprocessing, i.e. PDF conversion.
proxy-print.convert.suspend	`Y`, `N` (default). Set to `Y` to suspend conversion just before sending the job to CUPS. Set to `N` if conversion is to be applied as early as possible, e.g. when creating a Job Ticket or Hold Print Job.
proxy-print.convert.grayscale	`Y`, `N` (default). Set to `Y` to convert PDF to grayscale.
proxy-print.convert.rasterize	`Y`, `N` (default). Set to `Y` to rasterize the PDF.
proxy-print.convert.rasterize.zero.fonts	`Y`, `N` (default). Set to `Y` to rasterize even if no fonts are detected in the PDF.
proxy-print.convert.rasterize.dpi	Enum DPI values of rasterized PDF: `DPI_100`, `DPI_150`, `DPI_200`, `DPI_300` (default), `DPI_600`.
proxy-print.convert.rasterize.max.bytes	PDF file size (bytes) as limit above which no conversion is done. Value `0` (default) sets no limit.
proxy-print.convert.rasterize.max.pages	Number of PDF pages as limit above which no conversion is done. Value `0` (default) sets no limit.

Table 4.14. Proxy Print Preprocessing: Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to change these properties.

4.11.10. Eco Print

Eco Print is a filter that converts PDF pages to images for eco-friendly proxy printing. The result, including ink and toner savings, is comparable to Ecofont[22]. There is a difference though. While Ecofont uses True Type Font technology

[22] https://www.ecofont.com/

at the *start* of the print chain (document editing), SavaPage Eco Print uses bitmap technology at the *end* of the chain. Eco Print intelligently punches holes in *all* non-white areas of the PDF version of a document, just before proxy printing, downloading or emailing it.

Since Eco Print processes bitmap patterns it is font agnostic and therefore can handle all font types. And, as an extra, it punches graphics along the way. Contrary to Ecofont, which is a non-free Windows only solution, Eco Print is a Libre solution that works for all client platforms since filtering is performed server side.

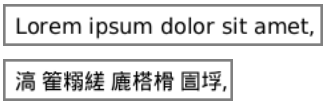
Warning

The downside of ad-hoc filtering is performance. Eco Print takes about 3 seconds per page (i5 processor, 300 DPI), but is done unobtrusive in the background and need only be done once per PDF document, since the result is cached. Anyhow, Eco Print is not suitable for very large documents.

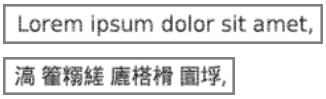
4.11.10.1. Eco Print Examples

A few Eco Print examples are depicted below.

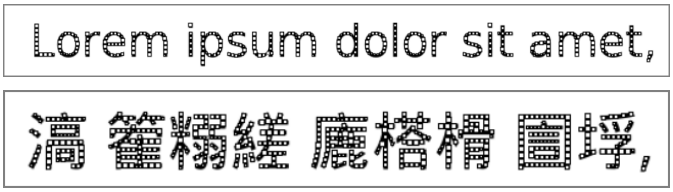
Plain Print:



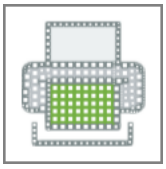
Eco Print:



Eco Print magnified:



Eco Print Graphics:



4.11.10.2. Eco Print Settings

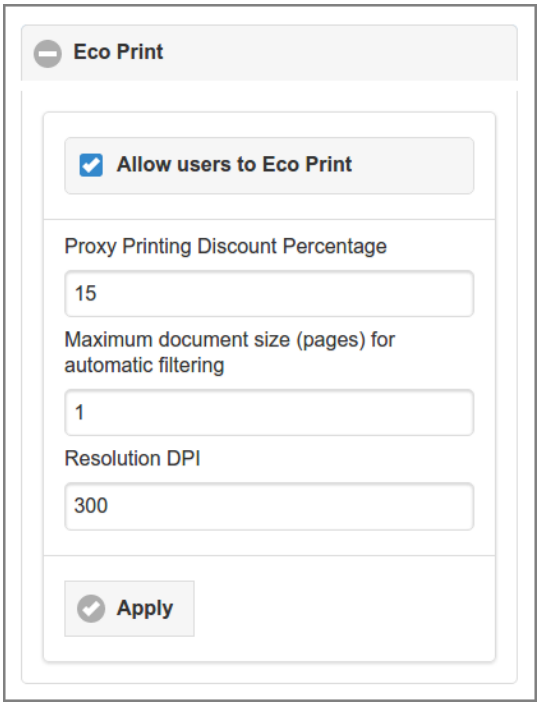


Figure 4.116. Admin Web App: Options - Eco Print

Check the Allow users to Eco Print to enable the Eco Print function. Then specify:

- A **Proxy Printing Discount Percentage** (integer) to be applied to proxy printing costs as specified for any Proxy Printer. See Section 4.8.2, "Edit Proxy Printer" [132].
- The **Maximum document size (pages) for automatic filtering**. In this example the value of "1" means that any document printed to SavaPage with 1 page is automatically filtered in the background. A value of "3" will automatically filter incoming documents of 3 pages or less. A value of "0" disables this automatism.
- The **Resolution DPI** of the Eco Print page image.

4.11.11. Financial

This section holds the options for SavaPage Financial.

4.11.11.1. Currency Code

Figure 4.117. Admin Web App: Options - Financial - Currency

The ISO 4217[23] currency code of the financial subsystem can be entered here during installation. When the application status is "ready-to-use" the code can only be changed by using a Server Command. See Section C.1.4, "changeBaseCurrency" [332].

[23] https://www.iso.org/iso/home/standards/currency_codes.htm

4.11.11.2. General Financial Options

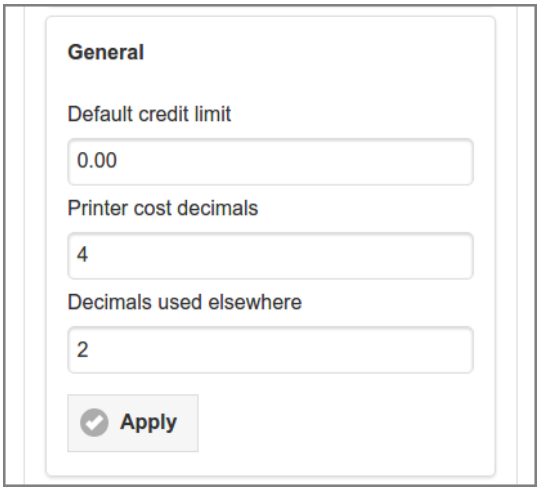


Figure 4.118. Admin Web App: Options - Financial - General

General options are:

- **Default credit limit**: this is the default value referenced in Section 4.4.5.8, "Financial" [106]. The value must be zero or greater.
- **Printer cost decimals**: the number of decimals (max 6) used to specify and display printer costs. See Section 4.8.2.3, "Media Sources" [134] and Section 4.8.2.4, "Manual Media Sizes" [136].
- **Decimals used elsewhere**: the number of decimals (max 6) used to specify financial amounts other than printer costs (e.g. for displaying user account balance).

Note

SavaPage stores financial amounts with a precision of 6 decimals.

4.11.11.3. Point-of-Sale

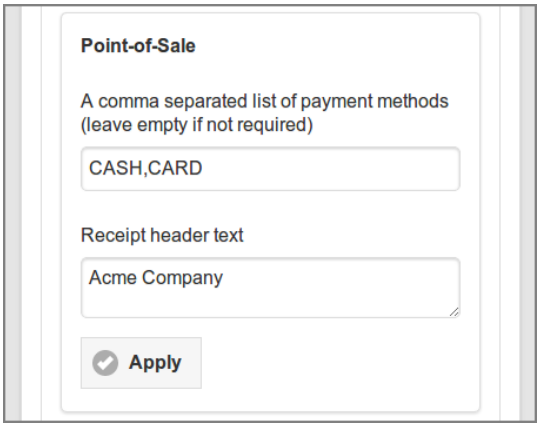


Figure 4.119. Admin Web App: Options - Financial - POS

These are the options for the Point-of-Sale:

- Payment methods: see Section 7.3, "Deposit" [227].
- **Receipt header text**: this typically contains a legal text placed in the Receipt header.

4.11.11.4. Vouchers

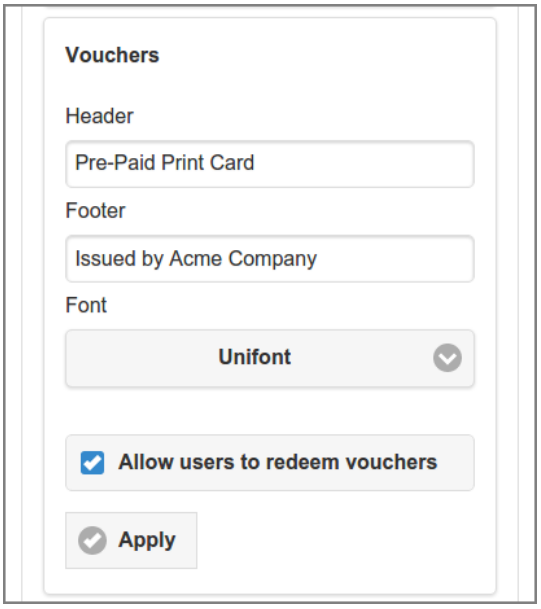


Figure 4.120. Admin Web App: Options - Financial - Vouchers

These are the options for the Voucher System:

- **Header**: the header text of the voucher card.
- **Footer**: the footer text of the voucher card.
- **Font**: the font used for the PDF Document with vouchers. See Section 19.2, “Internal Fonts” [297].
- You need to explicitly **Allow users to redeem vouchers**.

4.11.11.5. Transfer Funds

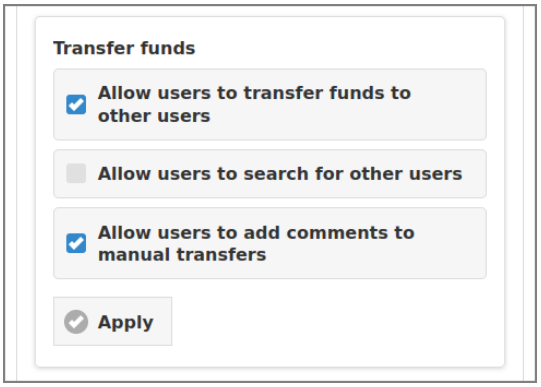


Figure 4.121. Admin Web App: Options - Financial - Transfer funds

These settings apply to Transfer Credit dialog in the User Web App. Check the options to **Allow users to transfer funds to other users**, to **Allow users to search for other users** and to **Allow users to add comments to manual transfers**.

The minimum and maximum amount to transfer are held in the configuration properties `financial.user.transfers.amount-min` and `financial.user.transfers.amount-max`. They can be changed with the Configuration Editor.

4.11.12. Integration

From a user organization point of view, integration means combining separate software components (subsystems) into one functional system. Currently SavaPage is able to connect to PaperCut to create an integrated Print Management solution.

4.11.12.1. PaperCut Integration

PaperCut is a popular print and copy management software product developed by PaperCut Software[24]. Some functions not present in PaperCut can be implemented with SavaPage as pre-processor and integrator. See Appendix N, *PaperCut Integration* [412].

When PaperCut Integration is enabled, connectivity parameters for the **PaperCut Server** (XML-RPC API) and **PaperCut Database** (JDBC) can be entered. Press the Apply button to commit the changes. Press the Test button to test the PaperCut connectivity: a message confirming the connection status is displayed.

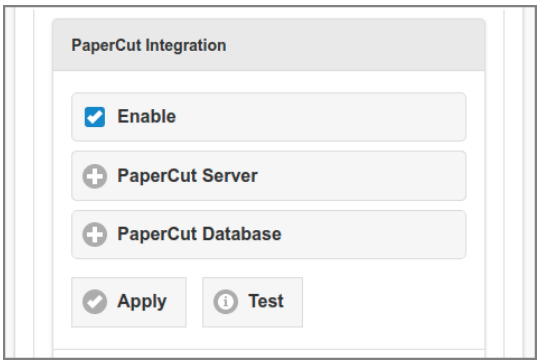


Figure 4.122. Admin Web App: Options - PaperCut Integration

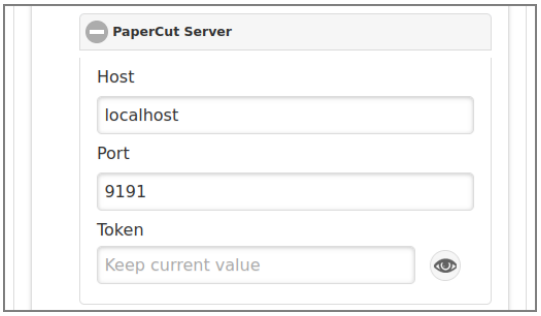


Figure 4.123. Admin Web App: Options - PaperCut Server

The **Token** must match the value of the `auth.webservices.auth-token` item in the PaperCut Config Editor. Use the Eye button on the right side of the entry field to show/hide the token. *Leave empty to keep the current value.*

[24] https://www.papercut.com/

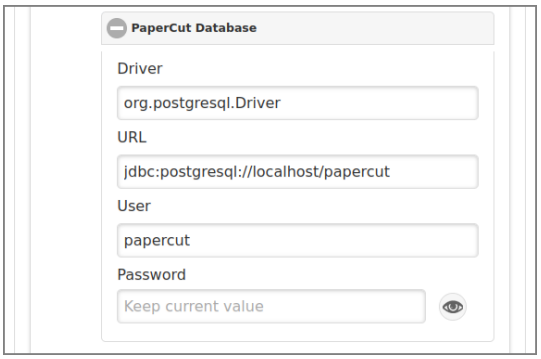


Figure 4.124. Admin Web App: Options - PaperCut Database

Use the Eye button on the right side of the **Password** field to show/hide the password. *Leave empty to keep the current value.*

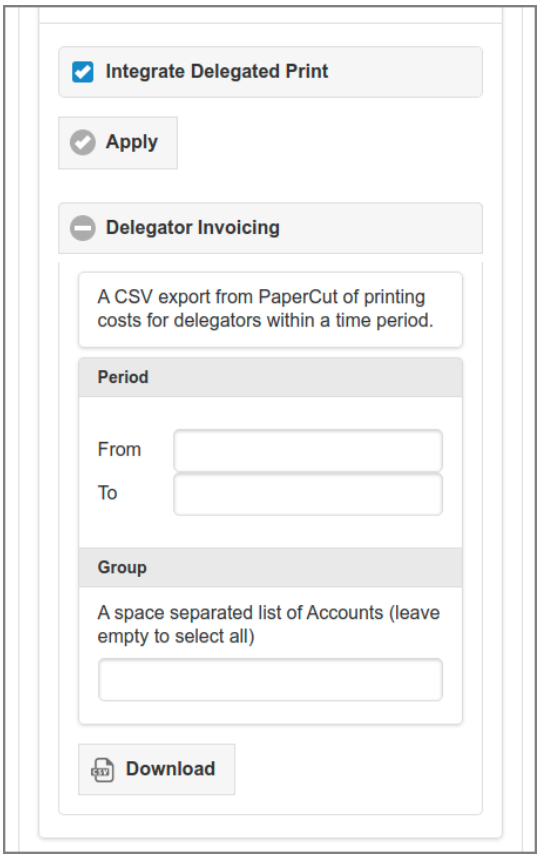


Figure 4.125. Admin Web App: Options - PaperCut Delegated Print Integration

When Proxy Print Delegation is *enabled*, the Integrate Delegated Print option is shown. If this option is selected, the **Delegator Invoicing** subsection is exposed where printing cost totals for delegators from selected Accounts, within a certain time period, can be exported. The result is a CSV file with a line for each delegator. Lines are ordered by user id and specify the cost total within the period and extra data like account and number of transactions per job type, like duplex/simplex,color/grayscale, page format A4, A3, etc. See PaperCut User Prints.

> **Note**
>
> For each Delegator line in the CSV file, a "class" is specified, which is one of the user groups used as selection context for the Delegator total (actually it is the alphanumeric MAX of all contexts involved in the total). Therefore, "class" is only an indication and does not stand for unique group membership. If you want to know what has been charged within a certain selection context, you can specify one or more accounts (classes) as selection.

References:

- Section N.1, “Delegated Print to PaperCut” [412]
- Section A.3.3, “Delegated Print - PaperCut Scenario” [325]
- Section A.3.2, “Delegated Print - Job Ticket - PaperCut - Scenario” [324]

Figure 4.126. Admin Web App: Options - PaperCut Personal Print Integration

When Proxy Print Delegation is *disabled*, Integrate Personal Print comes into play. With this setting *enabled*, Personal Print jobs are monitored in PaperCut.

References:

- Section N.3, “Personal Print to PaperCut” [416]
- Section A.2.3, “Personal Print - PaperCut Scenario” [321]

4.11.13. Backups

The **Backups** section shows the backup location and time of the last backup.

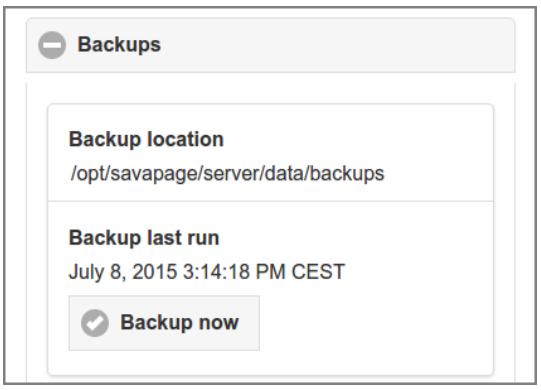


Figure 4.127. Admin Web App: Options - Backups

Press the Backup now button to launch the backup process in the background. The progress and result of the process is not echoed real-time in this section, but can be monitored in the Real-time Activity section of the Dashboard. Also see Section C.5.6, “db-export and db-export-to” [355].

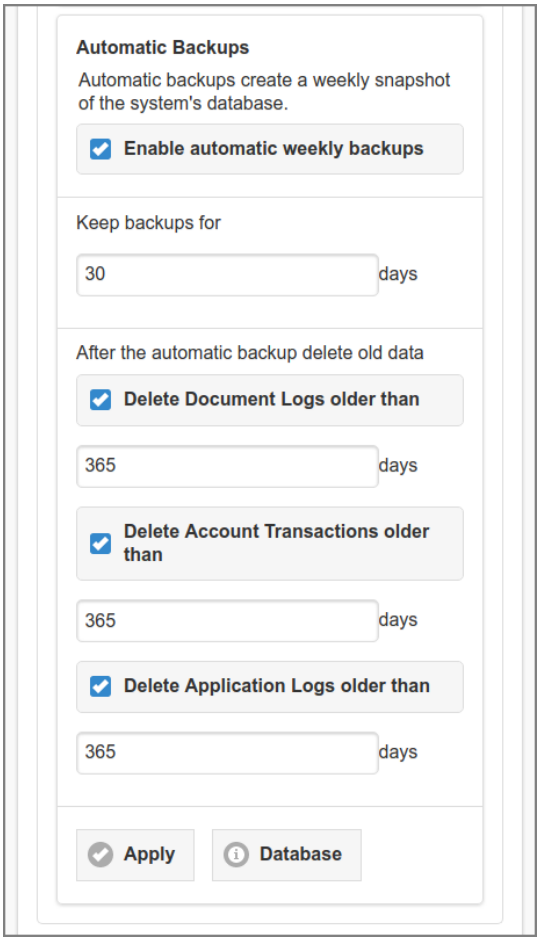


Figure 4.128. Admin Web App: Options - Automatic Backups

The **Automatic Backups** section holds options for creating weekly snapshots of the database.

- Tick the Enable automatic weekly backups checkbox to enable the process. See Weekly Jobs.
- The number of **days** a backup should be kept, must be entered at **Keep backups for**.
- A purge of old log data, executed after the backup, can be activated by selecting the **Delete older than** check-boxed for Application Logs, Document Logs and Transaction Logs. Please enter the number of **days** the logs should be held.
- Press the Apply button to commit the changes.
- Press the Database button to show a pop-up with the number of documents (received, printed, downloaded) and transactions in the database.

4.11.14. Advanced

4.11.14.1. User Client Authentication

The User Client uses the system account name of the user to identify itself to the SavaPage server. In a strict Single Sign-On (SSO) environment, where a user is already logged in and authenticated as domain user, the system account name can be trusted by default. In environments where non-domain systems are allowed, local accounts are not authenticated by domain services, and therefore can *not* be trusted.

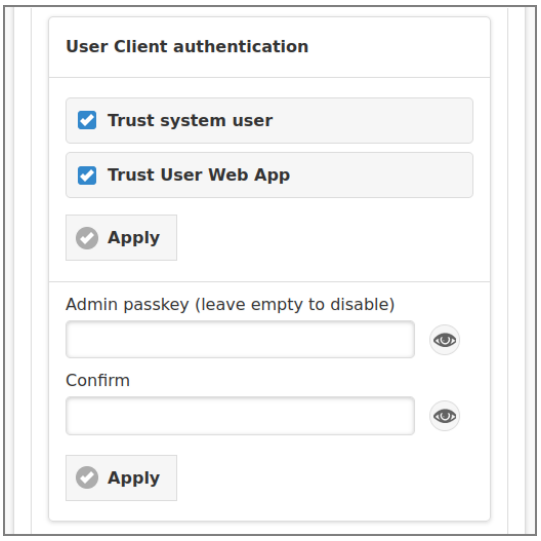


Figure 4.129. Admin Web App: Options - Advanced - User Client

User Client uses the system account name as user identification (unless overridden by a command line option).

- If Trust system user is *enabled* the User Client will trust the system account name.
- If Trust system user is *disabled* the User Client will pop-up a login dialog to authenticate the user, unless the following trust sources are available:
  - If Trust User Web App is *enabled* and the user is already authenticated in a User Web App on the same IP address, User Client will trust the Web App user as user identification.
  - If an administrator uses the secret **Admin passkey** in the start-up script it will enforce trust of the offered user identification. See Chapter 11, *User Client* [240]. Use the Eye button on the right side of the entry fields to show/hide the passkey.
- Press the Apply buttons to commit the changes.

4.11.14.2. User Name Aliases Cache

User Name Aliases are stored in a file whose content is read and cached when the SavaPage service is started. You can Refresh the cache after you edited the file content.

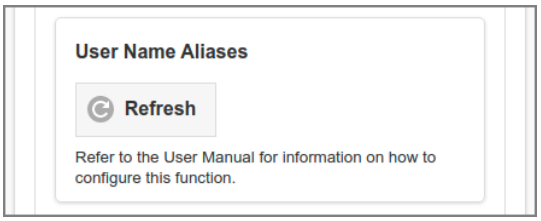


Figure 4.130. Admin Web App: Options - Advanced - User Name Aliases

4.11.14.3. Admin Password

The **Reset internal admin password** section is the place to change the master password for the built-in admin account.

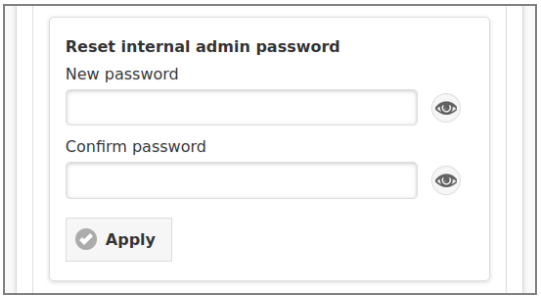


Figure 4.131. Admin Web App: Options - Advanced - Reset Admin Password

- Enter the new password twice at **New password** and **Confirm password**.
- The password must contain the same minimum number of characters as defined for Internal Users. See Section 4.4.6, "Create Internal User" [108].
- Use the Eye button on the right side of the entry fields to show/hide the password.
- Press the Apply button to commit the change.

> **Caution**
>
> Keep the new password at a secure place, since it is the master key to your system.

4.11.14.4. JMX Agent

SavaPage runs in a Java Virtual Machine, which has built-in instrumentation that enables client applications to monitor and manage it with the help of Java Management Extensions (JMX). SavaPage configures the built-in JMX agent for *remote monitoring*, so it can be *securely* accessed by remote client management applications, such as Java VisualVM or JConsole.

This section shows the JMX remote process connection string, and enables you to reset the `admin` connection password. Use the Eye button on the right side of the entry fields to show/hide the password.

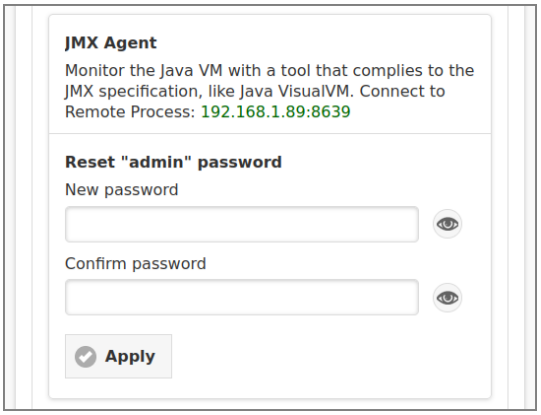


Figure 4.132. Admin Web App: Options - Advanced - JMX Agent

Java VisualVM is the standard Java JMX client that was first bundled with the Java Development Kit (JDK) version 6, update 7. It can be found in `JDK_HOME/bin`, where `JDK_HOME` is the directory where the JDK is installed.

If `JDK_HOME/bin` is in your system path, you can start Java VisualVM by simply typing **jvisualvm** in a command (shell) prompt. Otherwise, you have to type the full path to the executable file.

Since SavaPage enforces SSL for remote JMX communication, **jvisualvm** needs to be started with two special parameters referring to the Java truststore, holding the trusted SSL certificate, and the truststore password.

The shared client directory `/opt/savapage/client/jmx` contains the JMX server certificate `jmxremote.crt`, a ready to use `jmxremote.ts` truststore, and a sample GNU/Linux and Mac shell script `jmxremote.sh` and Windows command file `jmxremote.cmd` to start **jvisualvm** with the right parameters.

Note

The password of the provided `jmxremote.ts` truststore is: `savapage`. Of course you are free to import `jmxremote.crt` into your own truststore and use it with your own password.

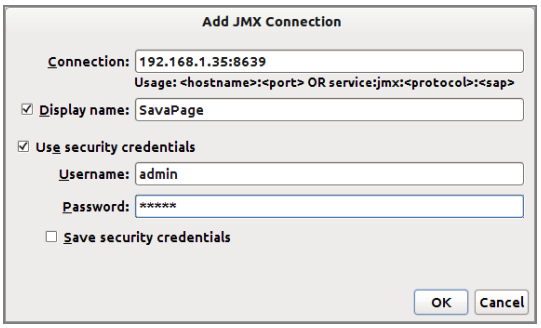


Figure 4.133. Add JMX Connection with Java VisualVM

Add a new JMX Connection and enter the IP address and port number of the **Connection** and as shown in the JMX Agent section, in our case this would be `192.168.1.35:8639`.

Enter the **Username** `admin` and the **Password** as set in the JMX Agent section above. Press the OK button to save the connection and start it from the Java VisualVM Applications pane.

Older JDK versions have JConsole as standard JMX client. If you want to use JConsole copy and edit the scripts in `/opt/savapage/client/jmx` so **jconsole** is used instead of the default **jvisualvm**.

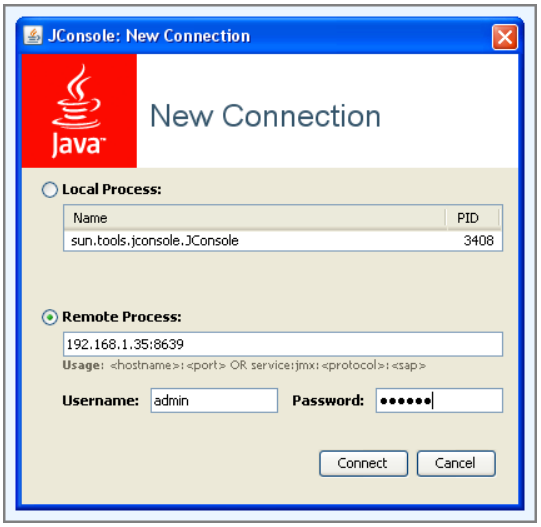


Figure 4.134. Connecting to Remote Process with JConsole

When starting JConsole it prompts you to enter the parameters for the New Connection. Select the **Remote Process** option and enter the IP address and port number as shown in the JMX Agent section, in our case this would be `192.168.1.35:8639`.

Enter the **Username** `admin` and the **Password** as set in the JMX Agent section above. Press the Connect button to open the connection.

More information about the JMX configuration can be found in Section 17.7, “Secured JMX Connection” [291].

4.11.14.5. Locale

Enter the System Locale string at the **Locale** section.

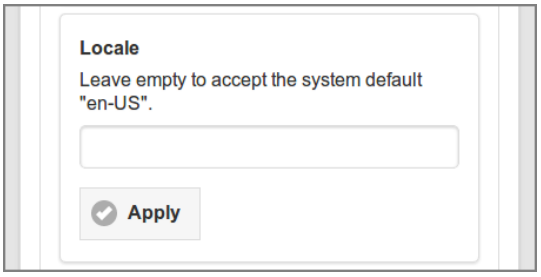


Figure 4.135. Admin Web App: Options - Advanced - Locale

- The format of the locale conforms to `IETF BCP 47`[25].
- Some examples are: `en`, `en-GB`, `en-US`, `nl`, `nl-NL`, `nl-BE`.
- You can leave the locale empty to accept the system default.
- The locale is applied to all system messages which are logged in the system log or send by email.
- Press the Apply button to commit the change.

> **Note**
>
> This system locale is *not* used for the language and country default used in the Web App. The Web App default is picked up from the locale settings of the Web browser, optionally overruled by the `language` and `country` URL parameters. See Appendix E, *URL Cheat Sheet* [362].

4.11.14.6. Default Paper Size

The **Default Paper Size** is used as the paper size for the printed document of a Printable File Type which itself does not have a document structure with a clearly defined page size. These types typically include HTML, TXT and images offered via Web Print and Mail Print. The **Default Paper Size** is also used as `media-default` for the SavaPage IPP Everywhere and AirPrint printer.

Choose Letter or A4, or accept the system default.

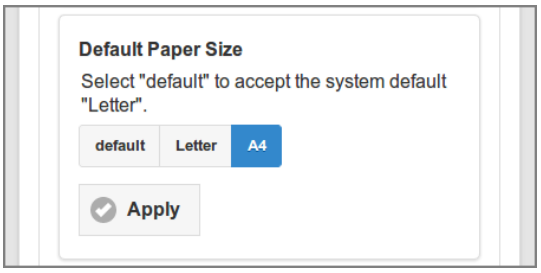


Figure 4.136. Admin Web App: Options - Default Paper Size

See Section 2.4.1, “Set Default Paper Size” [16] on how to set the system default.

4.11.14.7. Report Font

The **Report Font** is used as default font for all PDF reports.

[25] https://tools.ietf.org/html/bcp47

Figure 4.137. Admin Web App: Options - Default Paper Size

See Section 19.2, “Internal Fonts” [297].

4.11.14.8. Converters

Converters are used to convert files offered for printing via Web Print or Mail Print to PDF. This is the place to enable the converters. For installation see:

- Section G.1.1, “XPS to PDF Installation Instructions” [370]
- Section G.2, “Advanced File Types” [370]

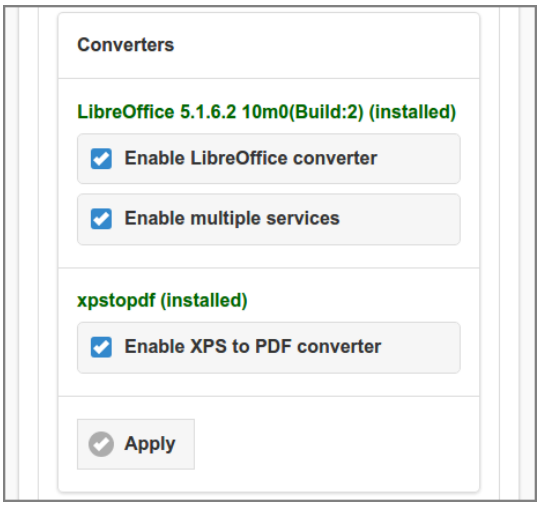


Figure 4.138. Admin Web App: Options - Converters

When Enable multiple services is checked, the LibreOffice converter acts as multi-threaded load-balancing service for easy upscaling of conversion throughput. The configuration properties that determine the behavior of this service are summarized in the table below. The defaults will work fine in most situations. By adding extra `soffice.connection.ports` you can enhance conversion throughput, as long as hardware resources permit.

Warning

Tuning LibreOffice configuration values is an advanced task. Please consult your SavaPage Service Provider about which values give the best performance in *your* situation. Then use the Configuration Editor to change the defaults.

Configuration property	Description
soffice.home	The LibreOffice home location. When empty, a probe to likely candidates is performed to retrieve the location. Default: empty.
soffice.profile.template-dir	When empty, a temporary profile directory is created for each UNO connection process with its own defaults settings. Otherwise, this configuration property must provide a profile

Configuration property	Description
	directory containing customized settings. This template directory will be copied to the temporary profile. Default: empty.
soffice.connection.ports	A comma/space separated list of TCP/IP ports to localhost LibreOffice (UNO) connection instances to be launched by SavaPage. Default: 2002,2003
soffice.connection.restart-task-count	The number of executed tasks after which a UNO connection is restarted. When `0` (zero) the connection is *never* restarted. Default: 200
soffice.task.queue-timeout-msec	Wait time (milliseconds) for a UNO connection to become available for task execution. Default: 10000
soffice.task.exec-timeout-msec	Wait time (milliseconds) for a conversion task to complete. Default: 20000
soffice.respond.retry-msec	Retry interval (milliseconds) for host process to respond. Default: 250
soffice.respond.timeout-msec	Wait time (milliseconds) for host process to respond (after retries). Default: 30000
soffice.start.retry-msec	Retry interval (milliseconds) for host process to start. Default: 1000
soffice.start.timeout-msec	Wait time (milliseconds) for host process to start (after retries). Default: 120000

Table 4.15. LibreOffice Configuration Properties

4.11.14.9. SafePages

This section contains advanced options regarding encrypted PDF and the expiration of **SafePages** input documents.

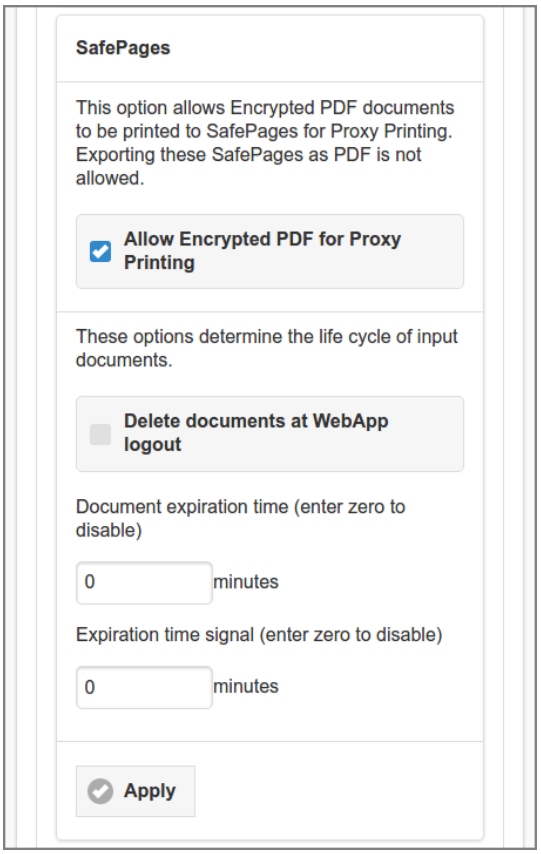


Figure 4.139. Admin Web App: Options - Advanced - Proxy Printing

- When the **Allow Encrypted PDF for Proxy Printing** option is enabled, Encrypted PDF documents with Printing as allowed action, are accepted as SafePages. However, SavaPage will protect the encryption of the original document, i.e. its pages are not allowed to be exported (downloaded or send) as PDF, directly, or as part of a composite document. When this option is disabled, all Encrypted PDF files are rejected. Encrypted PDF can be offered by:
  - SavaPage as Printer: see explanation in Section 14.10.2, "Printing Encrypted PDF" [272].
  - Web Print or Mail Print: host package QPDF must be installed so PDF can be decrypted. If not installed, Encrypted PDF is rejected despite allowing it with the option above.
  - Configuration property print-in.pdf.encrypted.allow with value Y (default) or N is responsible for allowing Encrypted PDF for Proxy Printing.
- When **Delete documents at Web App logout** is checked all print-in documents are deleted when the users logs out.
- **Document expiration time** manages the input document life cycle. Any document older than the number of entered minutes is considered expired and will be automatically deleted. For instance, a value of `1440` will delete the SafePages document `24` hours after it was printed. The expiration time is shown in the Document Details dialog. The user is notified by pop-up after an expired document is auto-deleted. User action is required to close the pop-up. This way we are sure the user noticed the delete and expectation is set right.
  - When a user logs out and logs in again after some time, expired documents will be auto-deleted to begin with, but the user will *not* be notified of this event.
  - Expired documents whose deletion is not triggered by an active User Session or Fast Print release are eventually deleted during User Home Clean.
- Use the **Expiration time signal** value to signal the user when expiration is due. For instance, a value of `15` will mark the document thumbnails with a clock icon in a colored (orange) footer, 15 minutes before expiration. This will alert the user, so he can do some last minute actions on old documents.
- Press the Apply button to commit the changes.

4.11.14.10. Document Store

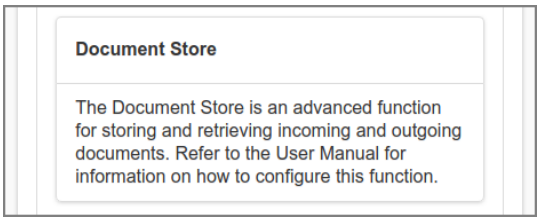


Figure 4.140. Admin Web App: Options - Advanced - Document Store

Document Store is a generic solution to store and retrieve different kind of documents. Namely, documents Printed to SavaPage, Downloaded as PDF or printed to Proxy Printers. Each kind of document is persisted in a separate storage *branch*, based on its source or destination.

- Documents are either stored for an extended or limited period of time. These two period types correspond with two types of stores: *Archive* for long-term, and *Journal* for short-term storage.
  - When in a use case, Archive and Journal are both applicable for the same branch, the Archive branch takes precedence to store the document.
- Document Store is persisted on the host file system, as described in Appendix F, *File Locations* [365]: path `/opt/savapage/server/data/doc-archive` and `/opt/savapage/server/data/doc-journal`.
  - Documents are stored with their metadata in a unique identifying file path of year, month, day, hour and document UUID. Expired documents are deleted at a daily schedule. Deletion is a technical operation where file path branches are pruned according to their timestamp fingerprint [26]. *If you want to keep a branch beyond the time limit, make sure to back-up the corresponding file path in time.*

[26] If a third party deletes a file from the Document Store it has no impact on pruning expired branches. However, as long as the directory holding the deleted file is present, SavaPage will tell that the document is stored. If, in that case, the document is accessed, retrieval will fail and an appropriate user message is displayed.

- Stored documents can be retrieved (downloaded) from the Document Log in all Web Apps.
- Currently Print Archive and Print Journal branches are supported for Proxy Print Jobs. When needed other branches can be implemented.
  - Consult Section 4.8.2.1, “Proxy Printer Identity” [132] on how to disable Print Archive and Print Journal for individual printers.
  - See Section 4.5.4.2, “User Privileges” [114] on how to enable storage access for User Groups.
  - Section 3.5.5.1, “Print Archive” [56] shows how the Archive option is presented in the Print Job Settings dialog.

Use the following configuration properties:

Configuration property	Description
doc.store.enable	Set to `Y` or `N` (default), to enable/disable the Document Store.
doc.store.free-space-limit-mb	The minimum MB of free space needed. When free space falls below this value, the store will be disabled till more space is available. Default: `5000`.
Archive branch	
doc.store.archive.enable	Set to `Y` or `N` (default), to enable/disable the `doc-archive/` branch.
doc.store.archive.out.enable	Set to `Y` or `N` (default), to enable/disable the `doc-archive/out/` branch.
doc.store.archive.out.print.enable	Set to `Y` or `N` (default), to enable/disable the `doc-archive/out/print/` branch.
doc.store.archive.out.print.days-to-keep	Number of days a Print Out Archive document is kept in store. Default: `30`. Cleaning is a Daily Job.
doc.store.archive.out.pdf.enable	*reserved for future use*
doc.store.archive.out.pdf.days-to-keep	*reserved for future use*
doc.store.archive.in.enable	*reserved for future use*
doc.store.archive.in.print.enable	*reserved for future use*
doc.store.archive.in.print.days-to-keep	*reserved for future use*
Journal branch	
doc.store.journal.enable	Set to `Y` or `N` (default), to enable/disable the `doc-journal/` branch.
doc.store.journal.out.enable	Set to `Y` or `N` (default), to enable/disable the `doc-journal/out/` branch.
doc.store.journal.out.print.enable	Set to `Y` or `N` (default), to enable/disable the `doc-journal/out/print/` branch.
doc.store.journal.out.print.days-to-keep	Number of days a Print Out Journal document is kept in store. Default: `2`. Cleaning is a Daily Job.
doc.store.journal.out.pdf.enable	*reserved for future use*
doc.store.journal.out.pdf.days-to-keep	*reserved for future use*
doc.store.journal.in.enable	Set to `Y` or `N` (default), to enable/disable the `doc-journal/in/` branch.
doc.store.journal.in.print.enable	Set to `Y` or `N` (default), to enable/disable the `doc-journal/in/print/` branch.
doc.store.journal.in.print.days-to-keep	Number of days a Print In Journal document is kept in store. Default: `2`. Cleaning is a Daily Job.

Table 4.16. Document Store Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these properties.

4.11.14.11. Telegram Messaging

Deprecated

Telegram support is **deprecated** and will be removed in a next release.

This option allows SavaPage to send Telegram[27] messages. For instance, 2-Step Verification Codes. In order to send messages, first follow the steps below to register a Telegram Bot[28].

1. Open your Telegram client, search for `@BotFather` and start a conversation.
2. Send `/newbot` to start the bot setup process.
3. Send the name of the bot, e.g. “My Org SavaPage bot”.
4. Send the username of the bot, e.g. “com_example_savapage_bot”.
5. BotFather confirms that a new bot has successfully been set-up and provides the HTTP API access token to you, e.g.

```
123456789:AAbbCCddEEffGGhhIIjjKKllMMnnOOppQQ
```

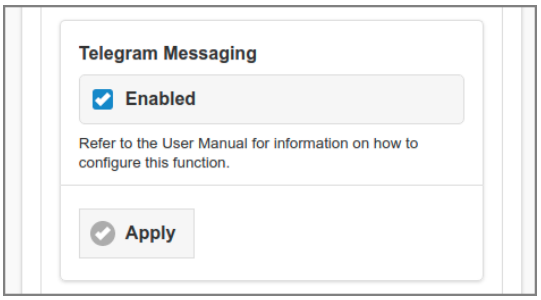


Figure 4.141. Admin Web App: Options - Advanced - Telegram Messaging

Activate Telegram Messaging by checking the Enabled button and by entering the Bot username and access token as configuration properties (see table below).

Now users can open Telegram Configuration and follow the instructions to initiate a conversation with the Telegram Bot and register their Telegram ID.

Configuration property	Description
ext.telegram.bot.username	Telegram Bot username.
ext.telegram.bot.token	Telegram HTTP API access token provided by BotFather.

Table 4.17. Telegram Messaging Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these properties.

4.11.14.12. Atom Feed

Press the Refresh button to publish an Atom Feed outside the publication schedule. See Section C.3, “Atom Feed Service” [352].

[27] https://www.telegram.org
[28] https://core.telegram.org/bots

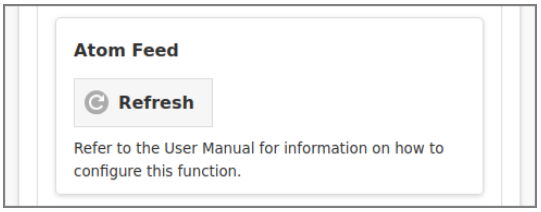


Figure 4.142. Admin Web App: Options - Advanced - Atom Feed

4.11.14.13. Matrix Messaging

This section is visible depending on Matrix configuration properties. See Section C.4, "Matrix Messaging Service" [352].

Press the Send button to send a message, optionally formatted as HTML and/or as Error. Press the Send Daily Summary button to send an ad hoc daily summary outside the messaging schedule.

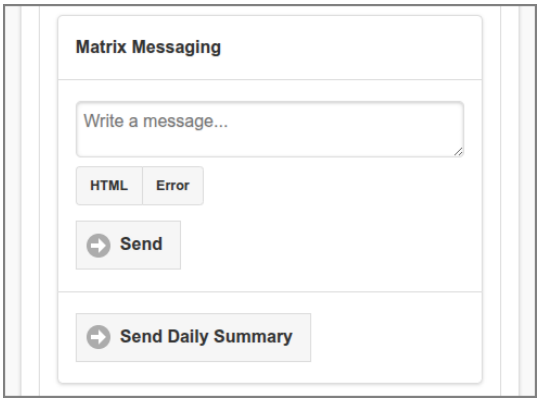


Figure 4.143. Admin Web App: Options - Advanced - Matrix Messaging

4.11.14.14. Pagometers

In analogy with the term *Odometer*, the term *Pagometer* is coined as an instrument to count the number of processed pages of SavaPage input and output documents. Pagometers are used to display usage statistics and Printing Impact from a global viewpoint as in the Dashboard, or in specialized views for User and Users, Queues and Proxy Printers. The counters can be reset in the **Reset Pagometers** section.

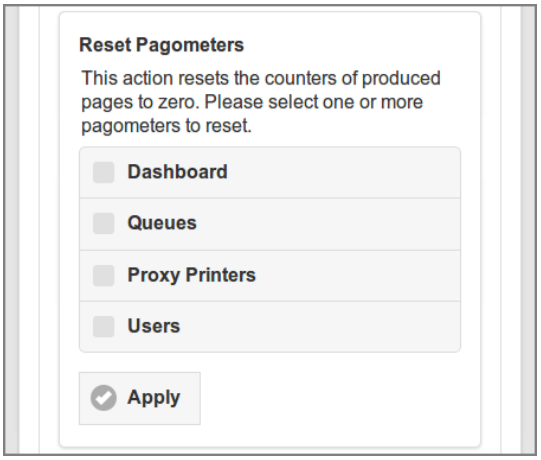


Figure 4.144. Admin Web App: Options - Advanced - Pagometers

- Tick the checkboxes of the pagometers to reset.
- Press the Apply button to execute the action.

4.11.15. Config Editor

Most of the SavaPage configuration properties can be edited in dedicated sections of the Admin Web App. However, many extra properties are present without an editing interface. Luckily a generic Configuration Editor is available for editing individual configuration properties, so the defaults of "hidden" properties can be changed when needed.

Warning

If you use the Config Editor incorrectly, you may cause serious problems which can only be fixed by re-installation of the application. Use the Config Editor at your own risk.

Tap the **Configuration Editor (advanced) button** to start the editor. See Figure 4.145, “Admin Web App: Configuration Editor - List” [189] for a detailed description.

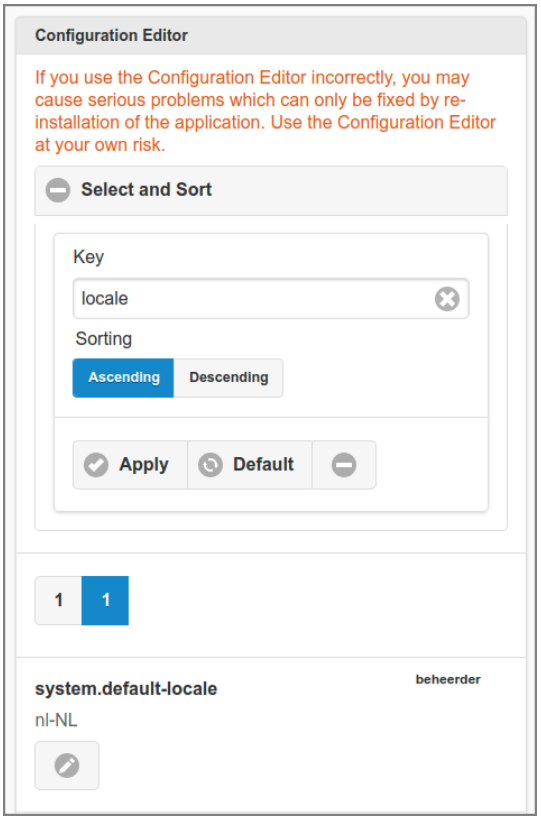


Figure 4.145. Admin Web App: Configuration Editor - List

- All configuration properties are listed alphabetically by default with their key and value. Secret values are encrypted and shown as `********` in the list, see Section 17.8, “Encrypted Secrets” [292].
- Push the Select and Sort button to expand (collapse) the section.
- The list can be traversed by tapping one of the buttons at the pager at the top or bottom of the page.
- Select items by entering a fragment) of their key. So, entering "ldap" will select "auth.ldap.port" and "ldap.schema.group-member-field".

- The list can be sorted Ascending or Descending on name.
- *Tap* the Apply button to (re)display the list.
- A *tap* on the Default button resets the selection and sort field to their default values.
- *Tap* the Edit button to edit the item. See Figure 4.146, “Admin Web App: Configuration Property - Edit” [190].

Important

Due to Admin Privileges the Edit button might *not* be visible.

Note

The Configuration Editor can also be launched from the Action Pop-up Menu at the footer of Admin Web App.

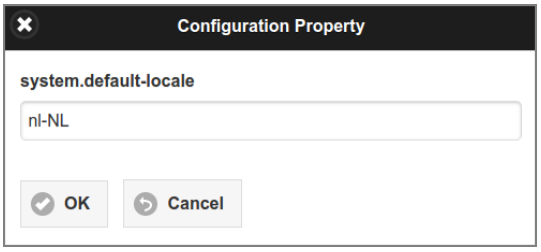


Figure 4.146. Admin Web App: Configuration Property - Edit

- The value of the item is shown in the entry field and can be edited.
- Press the OK button to commit the change and return to the list.
- The Cancel button brings you back to List without changing anything.

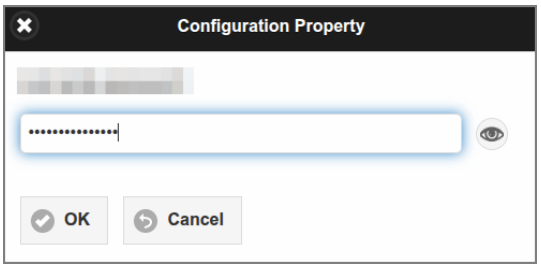


Figure 4.147. Admin Web App: Configuration Property - Edit Secret

- Secret values are shown in a password entry field.
- Use the Eye button on the right side of the entry field to show/hide the secret.

4.12. Documents

This section is shown after:

- A *tap* on the Documents button in the Main Menu : all processed documents are shown.
- A *tap* on the Log button in the User List: all documents processed by the selected user are shown. The user's name is shown in the header and the Select and Sort is within the scope of this user.
- A *tap* on the Log button in the Queues List: the Select and Sort is initialized with, and applied for the selected Queue.

- A *tap* on the Log button in the Proxy Printers List: the Select and Sort is initialized with, and applied for the selected Proxy Printer.

4.12.1. Document List

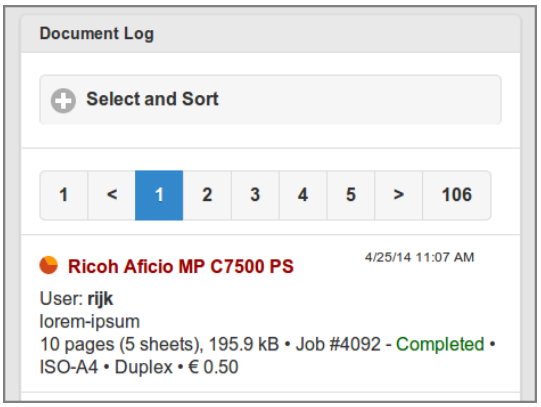


Figure 4.148. Admin Web App: Document - List

The list can be traversed by tapping one of the buttons at the pager at the top or bottom of the page.

Some samples of other entries...

Figure 4.149. Admin Web App: Document - IPP Routing

- A Proxy Print as a result of IPP Routing. The entry is marked as “Auto” (automatic) with QR code “1fa51673...” delivered by an IPP Routing Plug-in.
- The button at the bottom opens a pop-up with financial details.

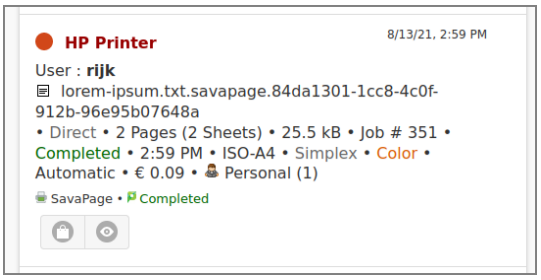


Figure 4.150. Admin Web App: Document - Proxy Print PaperCut Costs

- A Proxy Print to a PaperCut managed printer where PaperCut Printer Costs were leading.

- The left button at the bottom opens a pop-up with financial details, the right button downloads the Print Journal.
- Note the SavaPage (print supplier) and PaperCut (third-party print manager) icons just above the buttons.
- For a sample screenshot of the financial transaction, see Figure 3.73, “User Web App: Log - Proxy Print PaperCut Transaction” [74].

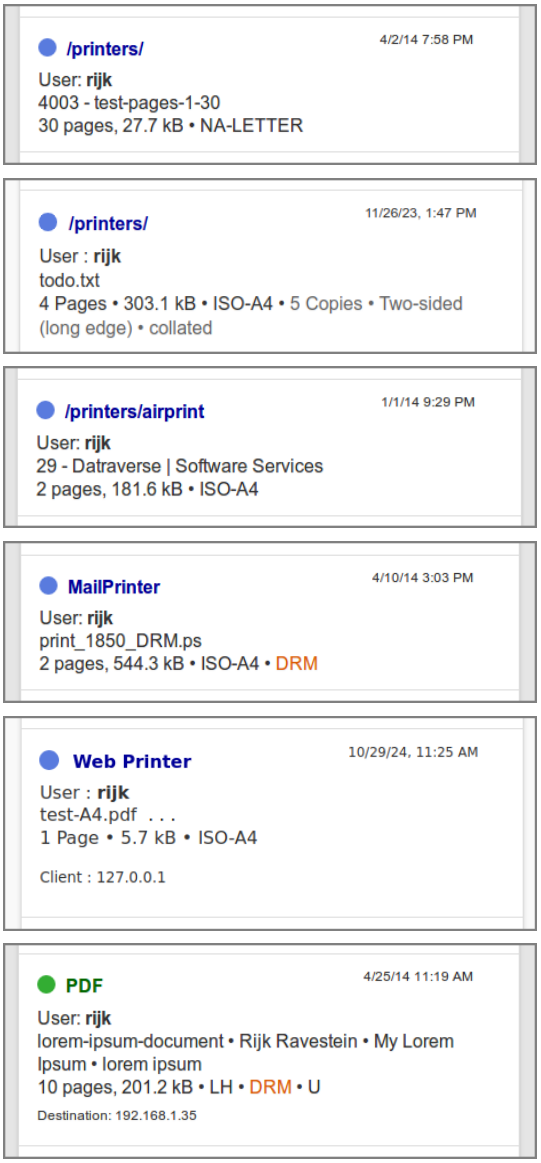


Figure 4.151. Admin Web App: Documents - Print-in - PDF out

Each document is displayed with data depending on its input source (SavaPage Print Queue) or output target (PDF export, Proxy Printer print). From top to bottom:

- The creation date at the top right corner.
- Source or destination, shown in a color depending on its type.
  - A SavaPage Queue, like "/printers/", is displayed in *blue* (when RAW printed to the default Queue the word “Printer” is displayed, Driverless File Transfer Print shows “Web Printer” or “Mail Printer”).
  - A PDF is shown in *green*.

  - A Proxy Printer, like “Ricoh Aficio MP C7500”, is displayed in *red* prefixed with an inline pagometer Pie-Chart. The *red* color in the chart represents the number of *pages* in the job. The *orange* color represents the number of printed *sheets*.
- User: the user id as creator of the document.
- The document name, with optionally (PDF export) the PDF author name, subject and keywords.
  - In case of a Proxy Print Job, icons 🗀 or 🗎 indicate the presence of a Print Archive or Print Journal. Pressing the “Eye” button at the bottom of the document entry will download the printed PDF. See Section 4.11.14.10, “Document Store” [185].
  - In case of a Print-in Job the 🗎 icon indicates the presence of a Queue Journal. Depending on User Privileges, action buttons are displayed at the bottom of the document entry. Pressing the “Eye” button will download and pressing the "Cross" button will delete the journaled PDF. Also see Section 3.8.1.1, “Queue Journal” [72].

    If Inherit IPP options from print-in document is enabled for Fast Print Mode the IPP Options from PostScript Driver Print-in are displayed.
  - If a PDF was uploaded at “Web Printer” the document name is appended with ". . ." , a click on the dots shows the properties of the uploaded file. For example:

```
Producer : Apache PDFBox 3.0.3
Creator  : SavaPage 1.6.0-rc • Open Print
           Portal • savapage.org
Author   : Rijk Ravestein
Title    : test-A4
Format   : 1.6
Fonts    : standard/embedded
```

- The number of pages and size (bytes). With optionally...
  - The CUPS Job number (Proxy Printer only).
  - The CUPS printing status: *Pending, Held, Processing, Stopped, Canceled, Aborted, Completed* (Proxy Printer only).
  - The paper size, like "ISO-A4" (Queue and Proxy Printer only).
  - "LH" indicator in case a letterhead was applied (Proxy Printer and PDF only).
  - "Duplex" indicator (Proxy Printer only).
  - "DRM" indicator when exported PDF was encrypted (PDF only), or printed document was an encrypted PDF. (Queue only).
  - "Denied" indicator when printed document was an encrypted PDF and such printing is not permitted (Queue only). See Section 14.10.2, “Printing Encrypted PDF” [272].
  - Owner ("O") and User ("U") password indicators (PDF only).
- Destination (PDF only). The client IP address (PDF Download) or the recipient email address (Send PDF)
- An gear icon with text “Web Service” in case a RESTful Print Service printed to a Queue other than the reserved `/webservice` queue.

Tap the Select and Sort button to expand the section, and select a document **Type** :

- Option - selects all document types: more...
- Option In selects documents printed to a SavaPage Queue: more...
- Option Out selects documents printed to a Proxy Printer or exported to PDF: more...
- Option PDF selects documents exported to PDF: more...
- Option Print selects documents printed to a Proxy Printer: more...

Depending on the selected **Type**, selection and sort options are shown or hidden. However, there are common selections for all document types as discussed at the screenshot below.

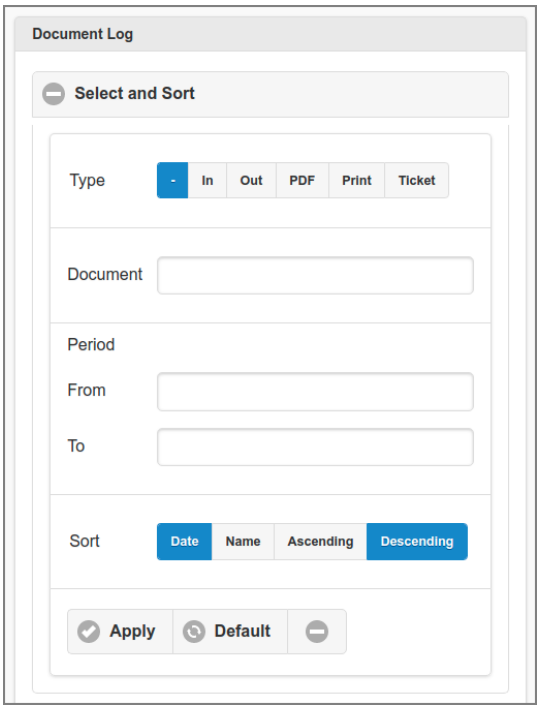


Figure 4.152. Admin Web App: Documents - Select and Sort - All

- Select a **Document** by entering a part (fragment) of its name.
- Select a creation **Period** by entering a `From` and `To` date. Tap the `x` button after a date to clear it. See this example Data Selection Dialog.
- Documents can sorted `Ascending` or `Descending` by creation `Date` or `Name`.

Figure 4.153. Admin Web App: Documents - Select and Sort - In

As an extra to the common options, the `In` **Type** offers:

- A selection on **Queue**.
- A **Sort** on `Queue`.

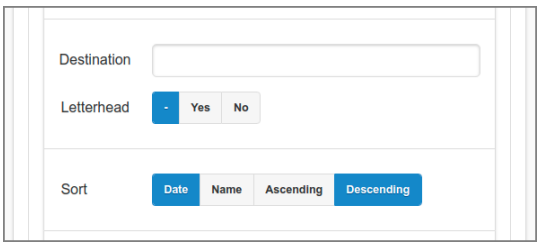


Figure 4.154. Admin Web App: Documents - Select and Sort - Out

As an extra to the common options, the Out **Type** offers:

- A selection on **Destination**.
- A selection on **Letterhead**.

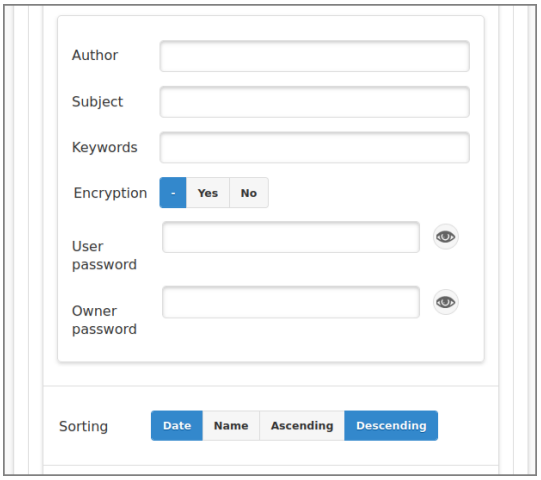


Figure 4.155. Admin Web App: Documents - Select and Sort - PDF

As an extra to the Out options, the PDF **Type** offers:

- A selection on **Author**.
- A selection on **Subject**.
- A selection on **Keywords**.
- A selection on **Encryption**.
- A selection on **User password** and **Owner password**.
  - Use the Eye button on the right side of these entry fields to show/hide the password.

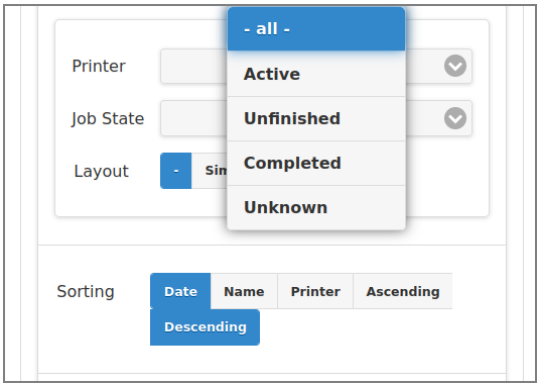


Figure 4.156. Admin Web App: Documents - Select and Sort - Print

As an extra to the Out options, the Print **Type** offers:

- A selection on **Printer**.
- A selection on **Job State**.
  - Status **Unkown** is explained in Section 2.3.3, “CUPS Job ID” [13].
- A selection on **Layout**: Simplex or Duplex.

- A **Sort** on Printer.

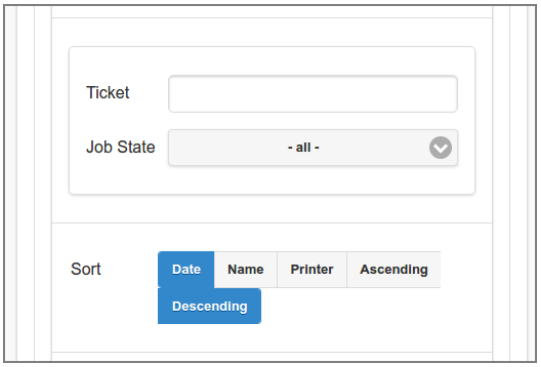


Figure 4.157. Admin Web App: Documents - Select and Sort - Ticket

The Ticket **Type** shows a slightly changed version of the Print Type options:

- A selection on **Ticket** is added (at the expense of the Printer selection). You can enter any part of a Ticket Number.
- Selection on **Layout** is dropped.

4.12.2. Proxy Print Reversal

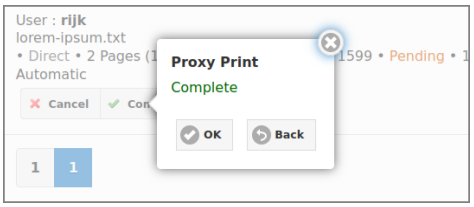
For a Proxy Print that is Canceled or Aborted a reverse button (at the right) is available in the document item to adapt counters and refund accounts charged. The button is available for administrators with Editor permission for Accounts. In case of a Job Ticket print the button is disabled as long as the Job Ticket is Open (its Close button is not pressed yet).

After the refund it looks like this:

See Section A.1, “Proxy Print Costs” [321].

4.12.3. Proxy Print Status

For an Active or Unknown Proxy Print, buttons are available to simulate a Cancel or Complete of a print job whose status is not automatically updated due to a duplicate CUPS Job ID. See Section 2.3.3.1, “CUPS Job ID Duplicates” [14].



The buttons are shown in the Admin Web App for administrators that have permission to edit accounts if Configuration property `proxy-print.job-state.edit.enable = Y`.

4.13. Log

After a *tap* on the Log button in the main menu this panel is shown. See Section 4.2, “Menu” [89].

The Log shows Info, Warning or Error messages for application events.

Tip

The size of the Log can be limited by purging old log data after an automatic database backup. See Figure 4.128, “Admin Web App: Options - Automatic Backups” [178].

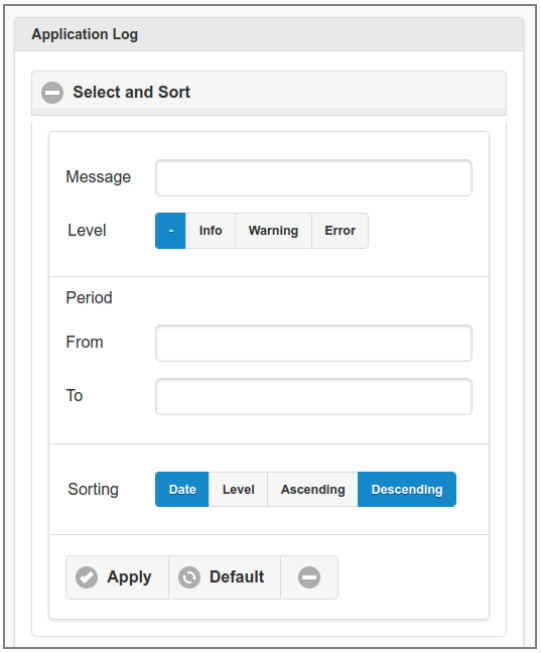


Figure 4.158. Admin Web App: Log - List

- All events are listed by default, ordered by descending date.
- The list can be traversed by tapping one of the buttons at the pager at the top or bottom of the page.
- A different selection and sorting can be entered: see Figure 4.159, “Admin Web App: Log - Select and Sort” [198].

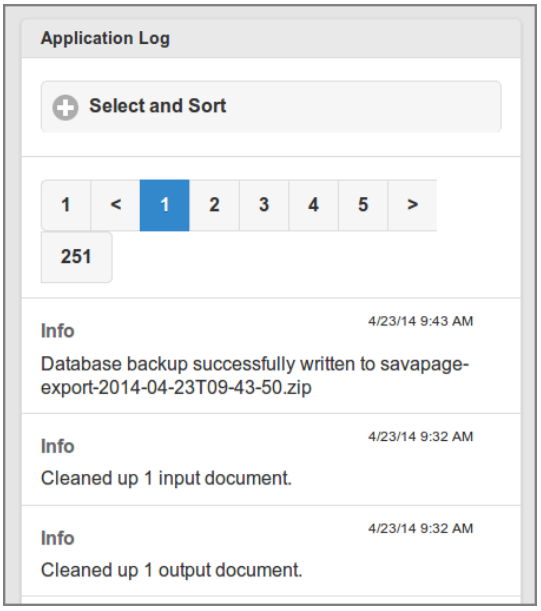


Figure 4.159. Admin Web App: Log - Select and Sort

- Events can be selected by entering a fragment of the **Message**.
- Select the message **Level**. The `-` button will select all levels.
- Select the time **Period** by pushing the **From** and **To** labels or entry fields. See date picker dialog is shown at Figure 4.160, “Admin Web App: Log - Select Date” [198]. Tap the `x` button after the date to clear it.
- The list can be sorted `Ascending` or `Descending` on `Date` or `Level`.
- *Tap* the `Apply` button to (re)display the list.
- A *tap* on the `Default` button resets the selection and sort fields to their default values.

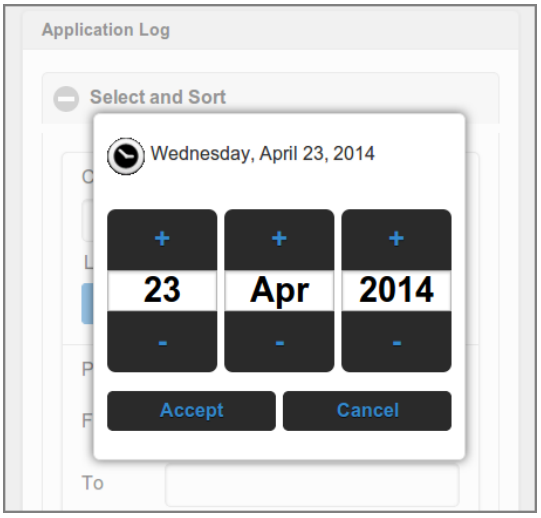


Figure 4.160. Admin Web App: Log - Select Date

- User the `+` and `-` buttons to select the Month, Day and Year.
- Push the `Accept` button to apply the date.
- Push the `Cancel` button to ignore the date and return to the original setting.

4.14. About

After a *tap* on the About button in the main menu this panel is shown. See Section 4.2, “Menu” [89].

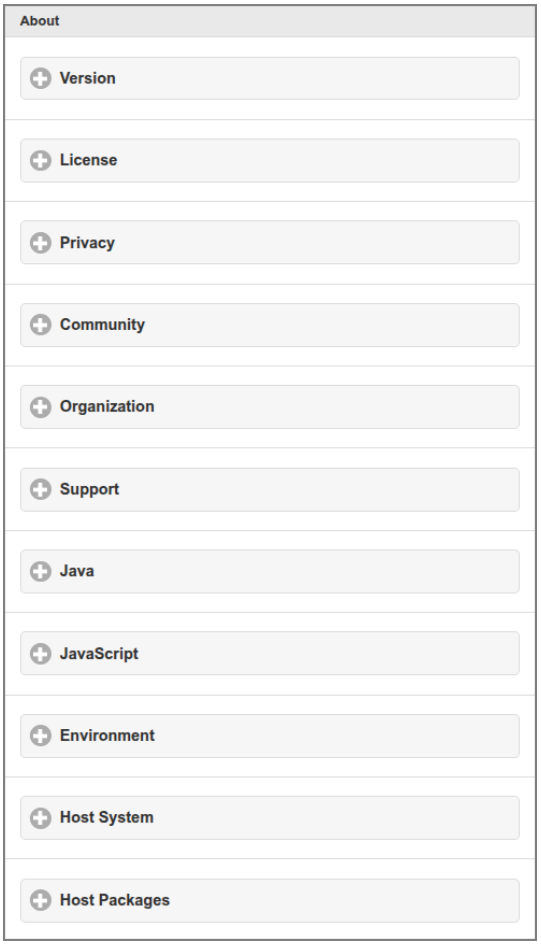


Figure 4.161. Admin Web App: About

A *tap* on one of the options expands (or collapsed) the underlying section. The sections are described in the paragraphs below.

4.14.1. Version

The Version Info identifies the application, database and files version. Please include this information when you contact support.

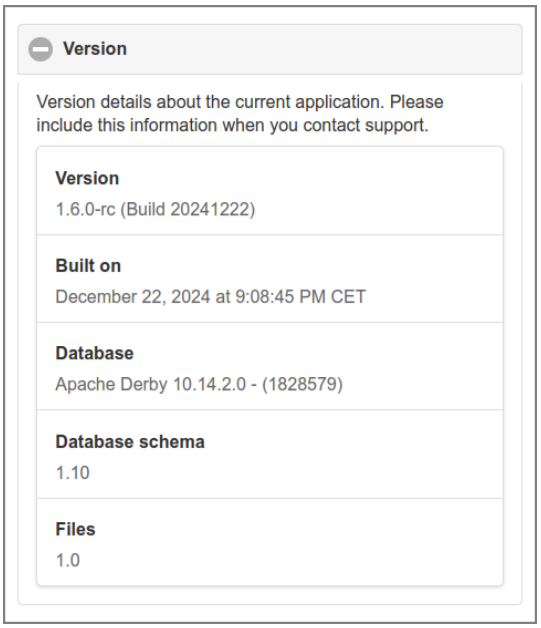


Figure 4.162. Admin Web App: About - Version

4.14.2. License

This section displays the license information with related links in green.

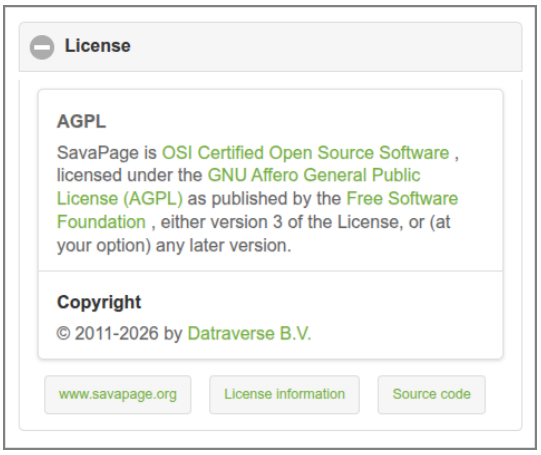


Figure 4.163. Admin Web App: About - License

4.14.3. Privacy

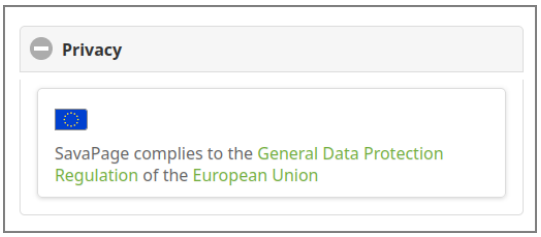


Figure 4.164. Admin Web App: About - Privacy

4.14.4. Community

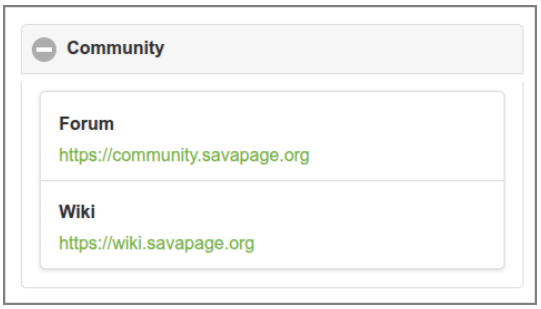


Figure 4.165. Admin Web App: About - Community

4.14.5. Organization

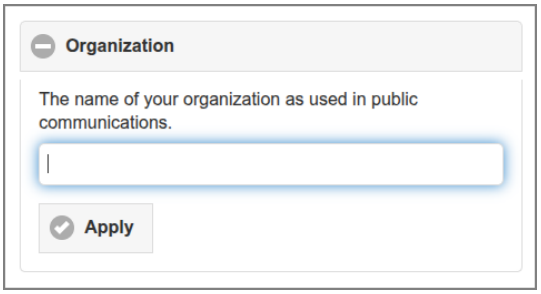


Figure 4.166. Admin Web App: About - Organization

> **Note**
>
> Due to Admin Privileges the `Apply` button might *not* be visible.

4.14.6. Support

The `Support` section shows Service Provider data from the imported Service Card.

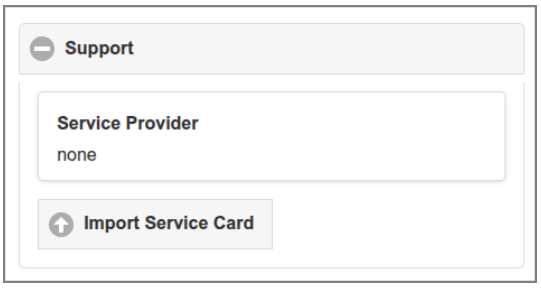


Figure 4.167. Admin Web App: About - Support

The `Import Service Card` button shows if the Organization name has been entered. Press the button to start the Import Dialog.

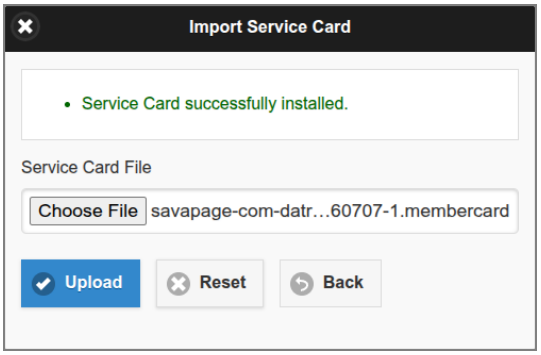


Figure 4.168. Admin Web App: About - Import Service Card

- Select the Service Card file to be uploaded. The actual file selection trigger differs from browser to browser.
- Press the Import button to start the import.
- The progress of the import is displayed at the top of the dialog box.
- Just in case, the Clear button clears the messages and selected file.

As a result Service Card information is shown, as in the example below.

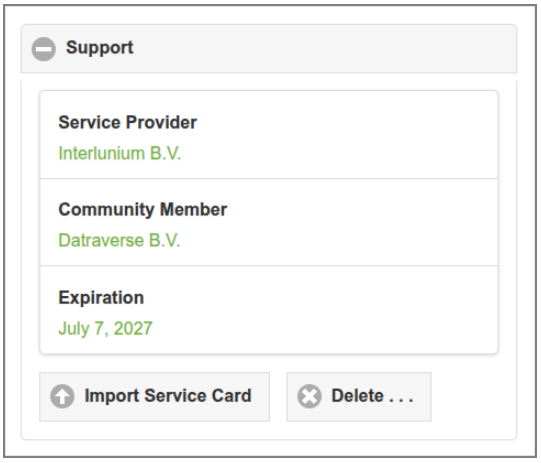


Figure 4.169. Admin Web App: About - Support - Service Provider

As you wish you can press the Delete button to remove the Service Card.

> **Note**
>
> Due to Admin Privileges the Import Service Card and Delete buttons might *not* be visible.

4.14.7. Java

This section displays Java runtime information, file locations and other server.properties.

Figure 4.170. Admin Web App: About - Java

Press the Clear button to clear the i18n cache, so changes in custom i18n files immediately take effect.

4.14.8. JavaScript

Figure 4.171. Admin Web App: About - JavaScript

JavaScript code run in this Web App is GNU LibreJS[29] compliant. Code and license of each JavaScript module can be inspected via the "JavaScript License Information" dialog that is opened by pressing the JavaScript information button. See Section 18.1.1, "Libre JavaScript" [295].

4.14.9. Environment

This section displays the SavaPage specific environment values.

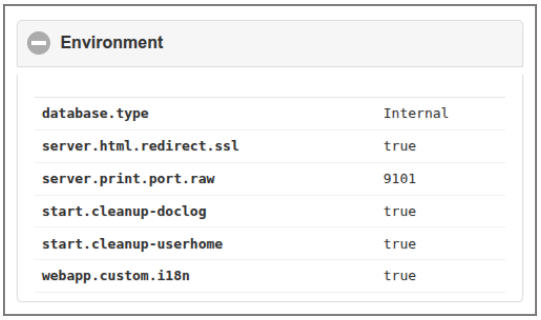


Figure 4.172. Admin Web App: About - Environment

[29] https://www.gnu.org/software/librejs/

> **Note**
>
> In case a CONTAINER is specified, its type is displayed on the first row. For instance: DOCKER.

See Section 13.5.15, “System Environment” [257] and Section 4.3.1, “Status” [92].

4.14.10. Host System

This section displays host name and IP address, the name, version and architecture of the host operating system and the GNU/Linux parameters described in Section 22.1, “Linux Kernel Parameters” [314] and Section 22.2, “Linux User Limits” [316].

4.14.11. Host Packages

The Host Packages gives version information about the required and optional third-party software installed on the SavaPage host.

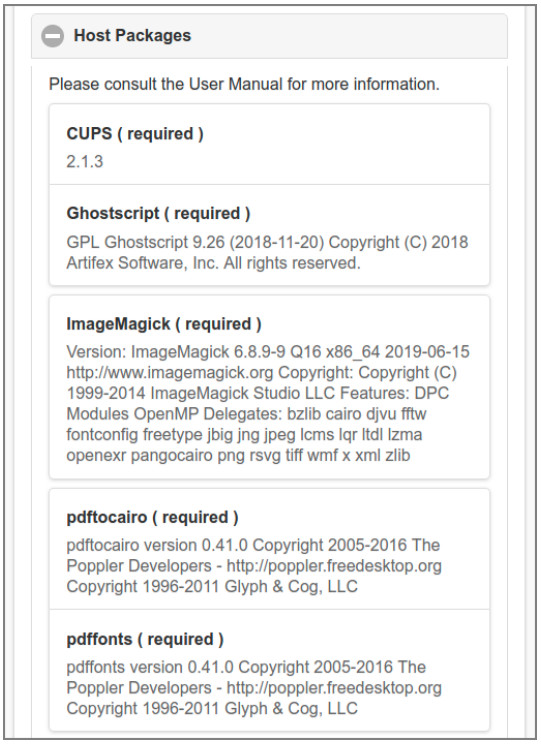
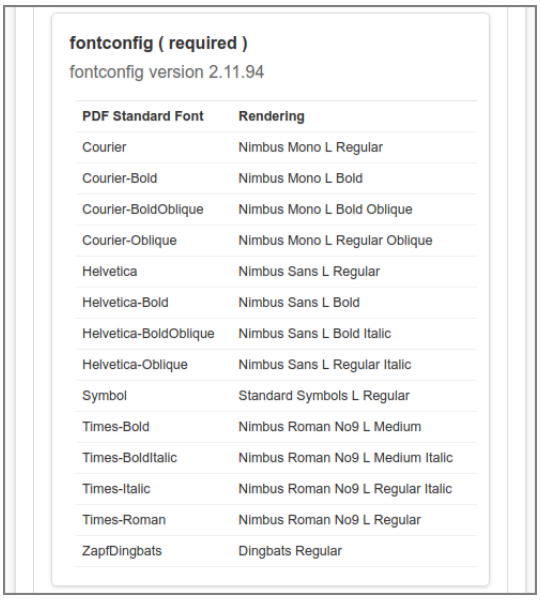


Figure 4.173. Admin Web App: About - Host Packages - 1/3

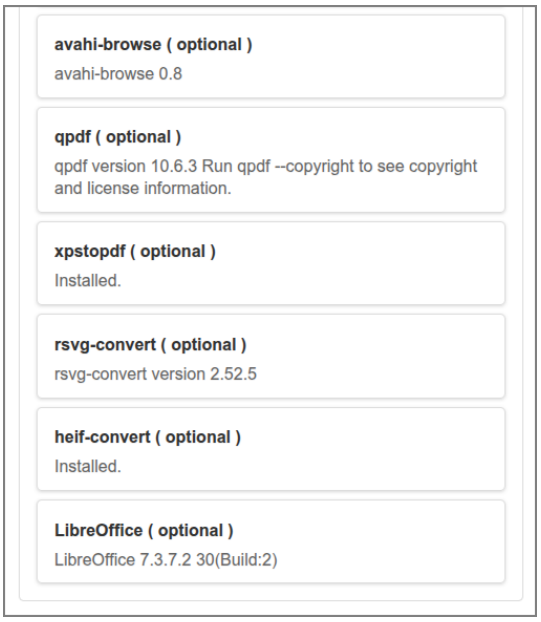


fontconfig (required)

fontconfig version 2.11.94

PDF Standard Font	Rendering
Courier	Nimbus Mono L Regular
Courier-Bold	Nimbus Mono L Bold
Courier-BoldOblique	Nimbus Mono L Bold Oblique
Courier-Oblique	Nimbus Mono L Regular Oblique
Helvetica	Nimbus Sans L Regular
Helvetica-Bold	Nimbus Sans L Bold
Helvetica-BoldOblique	Nimbus Sans L Bold Italic
Helvetica-Oblique	Nimbus Sans L Regular Italic
Symbol	Standard Symbols L Regular
Times-Bold	Nimbus Roman No9 L Medium
Times-BoldItalic	Nimbus Roman No9 L Medium Italic
Times-Italic	Nimbus Roman No9 L Regular Italic
Times-Roman	Nimbus Roman No9 L Regular
ZapfDingbats	Dingbats Regular

Figure 4.174. Admin Web App: About - PDF Standard Font Substitutes - 2/3



avahi-browse (optional)

avahi-browse 0.8

qpdf (optional)

qpdf version 10.6.3 Run qpdf --copyright to see copyright and license information.

xpstopdf (optional)

Installed.

rsvg-convert (optional)

rsvg-convert version 2.52.5

heif-convert (optional)

Installed.

LibreOffice (optional)

LibreOffice 7.3.7.2 30(Build:2)

Figure 4.175. Admin Web App: About - Host Packages - 3/3

More information can be found at:

- Section 1.2.1.2, “CUPS” [3]
- Section 1.2.1.9, “ImageMagick” [5]
- Section 1.2.1.5, “Poppler” [4]
- Section 1.2.1.10, “Avahi” [6]
- Section 1.2.1.6, “QPDF” [5]
- Section G.1.1, “XPS to PDF Installation Instructions” [370]
- Section 1.2.1.7, “rsvg-convert” [5]
- Section 1.2.1.8, “heif-convert” [5].

- Section G.2, “Advanced File Types” [370].

> **Note**
>
> Before **xptopdf** and LibreOffice can be used they must be *enabled*. See Figure 4.138, “Admin Web App: Options - Converters” [183].

4.15. Vouchers

This option is available from the pop-up menu at the footer of the Main Menu.

Vouchers provide a straightforward and cost effective solution for users to upgrade their account balance. Vouchers are common value tokens in many applications, like for instance pre-paid mobile phones. Unlike solutions that use smart cards, micro-payments or vending machines, voucher systems require no hardware investment. While manual processing is needed to generate, print, distribute and sell the voucher cards, redemption is fully end-user driven and can be processed automatically.

A voucher system is fully integrated in SavaPage, and includes:

- A Web App dialog for administrators to Create Vouchers.
- A Web App dialog for end-users to Redeem Vouchers.
- A Voucher Security safety net for voucher tracking and fraud prevention.

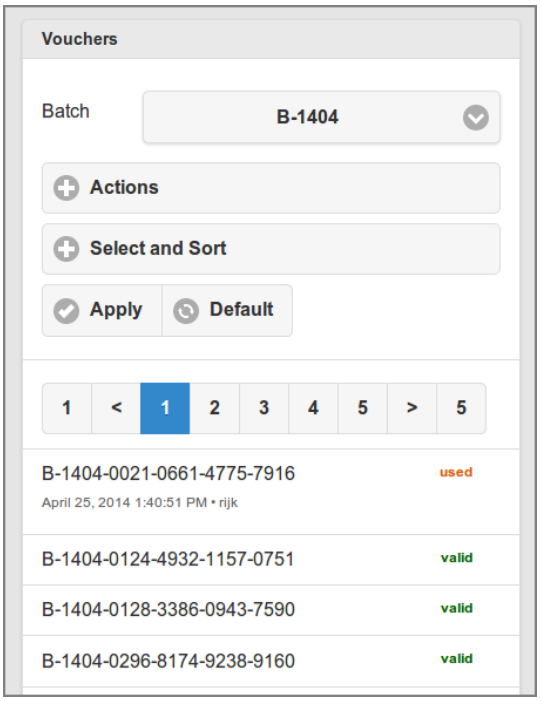


Figure 4.176. Admin Web App: Voucher List

The list shows the vouchers of a selected Batch-ID and the extra selections shown in the next figure.

- The list is refreshed, and the selection applied, after you push the Apply button.
- The Default button resets the selection items to their default values.

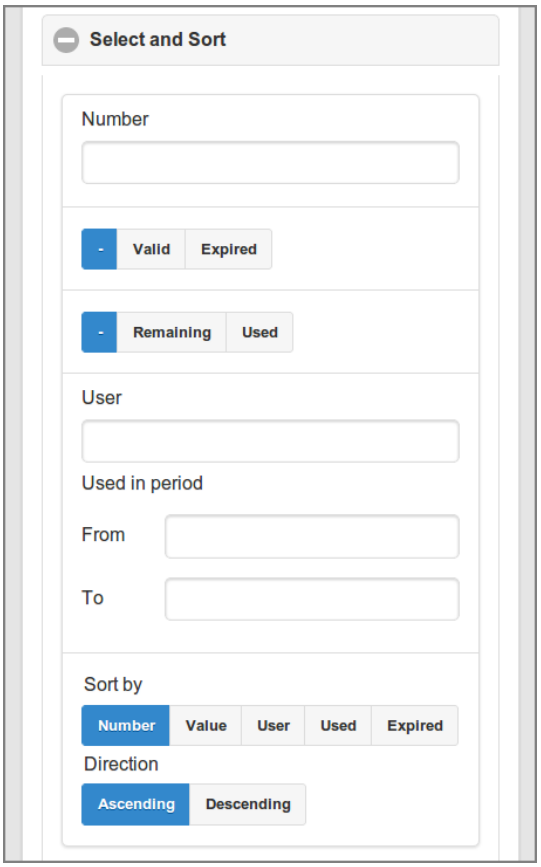


Figure 4.177. Admin Web App: Vouchers - Select and Sort

Select vouchers by entering:

- Part of their **Number**.
- Their Valid or Expired status.
- Their Remaining or Used status.
- Part of the **User** ID that redeemed any voucher.
- The **From** and **To** date of their usage period.

4.15.1. Voucher Actions

Note

Due to Admin Privileges this section might *not* be visible.

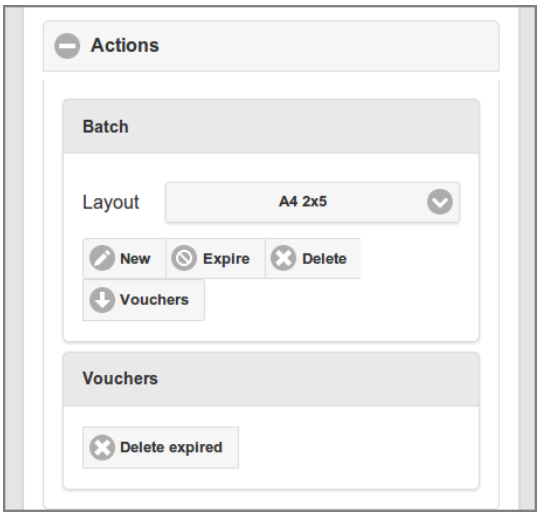


Figure 4.178. Admin Web App: Voucher Actions

- New : pops up the dialog to Create Vouchers.
- Delete expired : deletes vouchers whose expiry date is before the current date (today).

When a Batch-ID is selected extra buttons appear:

- Expire : Expires all remaining vouchers of the selected Batch-ID by setting the expiry date to today 00:00.
- Delete : Deletes all remaining vouchers of the selected Batch-ID.
- Vouchers : downloads a printable PDF document with remaining (non-redeemed) vouchers of the selected Batch-ID according to the selected **Layout**.

4.15.2. Create Vouchers

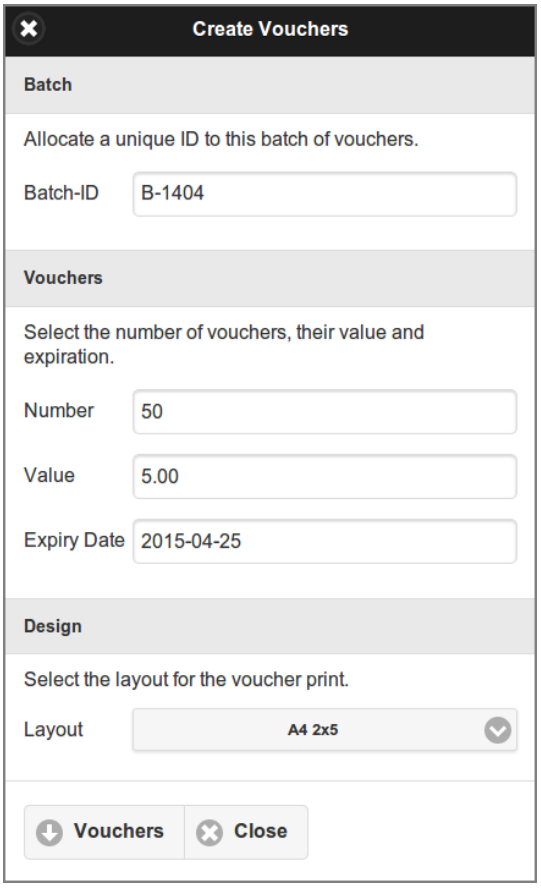


Figure 4.179. Admin Web App: Create Vouchers

Enter the data for the batch as follows:

- **Batch-ID**: a user defined ID that will be assigned to all vouchers in a batch. The ID is prefixed to each voucher number to easily identify its source. A unique ID should be assigned to each batch.
- **Number**: the number of vouchers in the batch.
- **Value**: the monetary value of each voucher.
- **Expiry Date**: the date after which a voucher can no longer be used. This enforces that vouchers are valid for a limited period of time.
- **Layout**: the page format of the PDF output with the number of voucher columns and rows. Some fixed variants are offered.

Press the Vouchers button to create the batch. As a result:

- Each voucher in the batch is assigned a formatted random unique number, for example `B-1404-0021-0661-4775-7916`, and is stored in the database.
- A printable PDF document is downloaded with all vouchers from the batch according to the selected **Layout**.

4.15.3. Voucher Usage

This is what end-users should know about vouchers.

- Purchase a voucher from an authorized person at an assigned location. Vouchers are unique for your organization and cannot be used elsewhere.

- Use a web browser to open the SavaPage User Web App. After logging in, your current account balance is shown at the footer bar.
- Push the account balance button to pop-up the User Details dialog, and push the Redeem Voucher button in the pop-up.
- Enter the voucher Number in the next dialog box and press Redeem. Make sure to enter the number exactly as listed on the voucher including any dashes (-).
- If you entered the number correctly, the value as shown on the voucher will be transferred to your account and a new entry will list in your transaction log.

Chapter 5. Job Tickets Web App

The Job Tickets Web App can be accessed by users with role Job Ticket Operator to dispatch job tickets. Job Tickets are created by users with role Job Ticket Creator in the Print Dialog of the User Web App. Operators optionally add or edit print options and select a suitable proxy printer to print the job.

The Web App can be reached at `https://savapage:8632/jobtickets`. See Appendix E, *URL Cheat Sheet* [362].

5.1. Login

The Job Tickets Login screen is a variant of the User Login screen, with the following exception:

- The internal `admin` user and Persons with role Job Ticket Operator are allowed to log in. See Section 4.4.5.2, "User Roles" [103] how to assign admin rights to users.
- After a successful login the Open Tickets are shown.

5.2. Open Tickets

5.2.1. Select and Sort

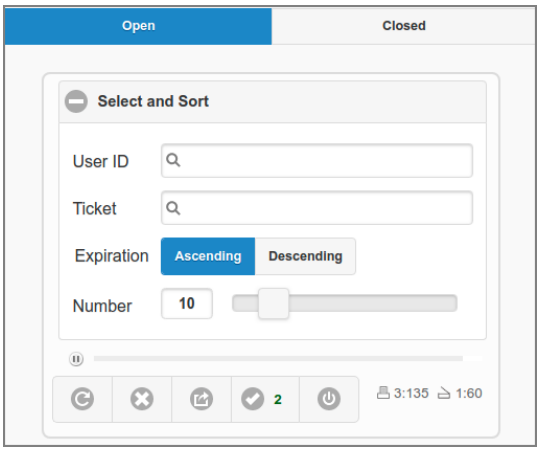


Figure 5.1. Job Tickets: Open Tickets - Select and Sort

A list of pending Job Tickets is shown that refreshes automatically every 60 seconds. The tiny progress bar at the bottom of the **Select and Sort** section shows when the refresh is due.

- You can pause/resume the refresh by toggling the tiny button at the left of the progress bar.
- You can Quick Search Job Tickets for a single **User ID**. Leave empty to select all users.
- You can Quick Search for a single **Ticket** Number. When a User ID is selected, tickets of just this user are shown. Leave empty to select all tickets.

- Tickets are sorted in the following way:
  1. Ascending or Descending on Expiration time.
  2. Ascending on Submit Date.
  3. Ascending on Job Ticket Creator.
  4. Ascending on “chunk-index”.

  Tickets that have identical values for 1, 2 and 3, are part of a single chunk, and are displayed together, even if this exceeds the **Number** limit of tickets shown. When a chunk contains more than one (1) ticket, the “chunk-index” and “chunk-size” are displayed for each ticket in the chunk. For example, for a chunk-size of 3, tickets in the chunk are marked as: “1 / 3”, “2 / 3” and “3 / 3” .
- The **Number** of tickets shown in the list is limited, and can be controlled with a slider. The slider range and default can be set with configuration properties.
- At the bottom right, a small printer and copier icon is shown with the total number of pending print/copy tickets and number of sheets to be printed.

5.2.1.1. Job Ticket Group

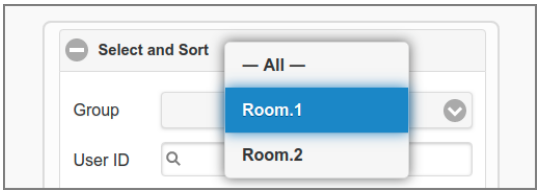


Figure 5.2. Job Tickets: Open Tickets - Group

When one or more Job Ticket printers are member of a Proxy Printer Group, these groups can be used to select tickets. When a **Group** is selected, just the tickets from printers that are member of this group are shown. If no groups are defined in Job Ticket printers, the Group selection is not visible.

5.2.2. Ticket List

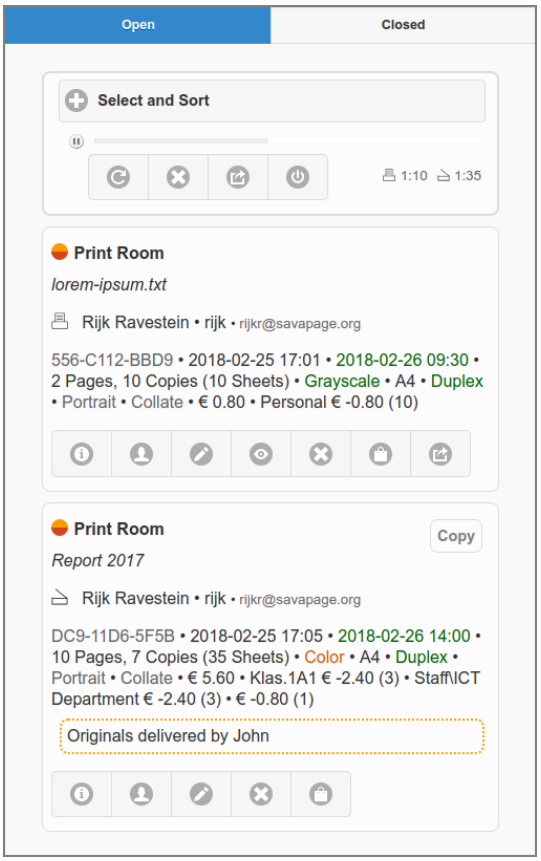


Figure 5.3. Job Tickets: Open Tickets - List

Each Job Ticket in the list shows a summary of options. Some are marked with outstanding colors.

- For instance, paper size (media) that differs from Default Paper Size (usually A4 or Letter) is marked with an orange-brown color.

When the page size of the PDF document to be printed differs from the Media Size of the print, both sizes are displayed. For instance, “A3 > A4 • Scaled” means that the PDF A3 source is scaled down to an A4 print.

Below the details there are buttons for further actions. From left to right:

- Settings... shows a dialog with a summary of printer settings.
- Transactions... shows the Transactions Dialog with accounting details where number of copies can be edited.
- Edit... shows the Edit Dialog, so job options can be changed.
- Preview downloads the PDF document to be printed.
- Cancel... shows a dialog to confirm cancellation of the Ticket. A reason can be entered. After cancellation the user is optionally notified by email or a Notification Plug-in: see Job Ticket Configuration Properties.
- Settle... shows the Settle Dialog, so the job can be charged, when not printed from the Print Dialog, but from the PDF viewer, after Preview.
- Print... shows a Print Dialog with compatible printers to redirect the job to.

Note

When pop-up dialogs are visible, automatic list refresh is paused.

> **Note**
>
> Reopened Job Tickets are marked as such.

5.2.3. Job Ticket Bulk Actions

The *scope* of the Cancel All... , Print All... and Close All... is the *visible* list selection. This bulk action must be confirmed in a pop-up.

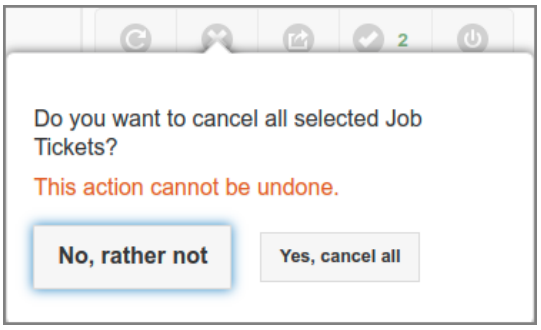


Figure 5.4. Job Tickets: Cancel All

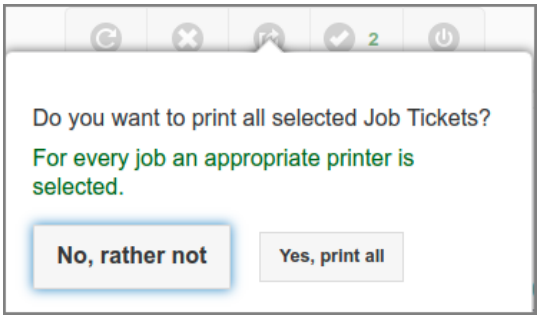


Figure 5.5. Job Tickets: Print All

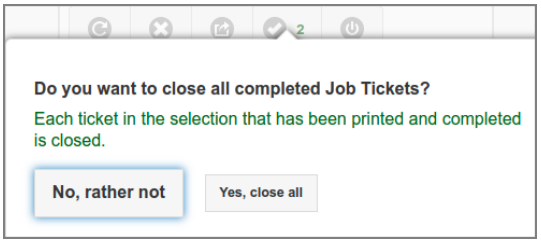


Figure 5.6. Job Tickets: Close All

The Cancel All... and Close All... actions are enabled by default and can be disabled by setting a configuration property.

The Print All... action selects a random compatible printer from the Job Ticket Proxy Printer Group for each Job Ticket.

- The selected printer *must* support automatic media source selection, since this option will be used for the print job. In this way, media (implicitly) selected through the job ticket media-source, will be printed from the intended tray.
- The Print All... action is enabled by default and can be disabled by setting a configuration property.

5.2.4. Job Ticket Transactions

This dialog pops up after pressing the Transactions... button in a Job Ticket List Item.

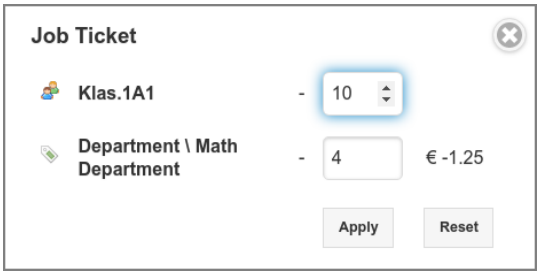


Figure 5.7. Job Tickets: Transactions

The number of copies for each account can be edited. If a number is changed, its transaction cost is hidden and the Apply and Reset buttons are enabled.

Note

A single Personal account transaction is display only. Its copies can be edited in the Job Ticket Edit Dialog as job option.

5.2.5. Job Ticket Edit

This dialog pops up after pressing the Edit... button in a Job Ticket List Item.

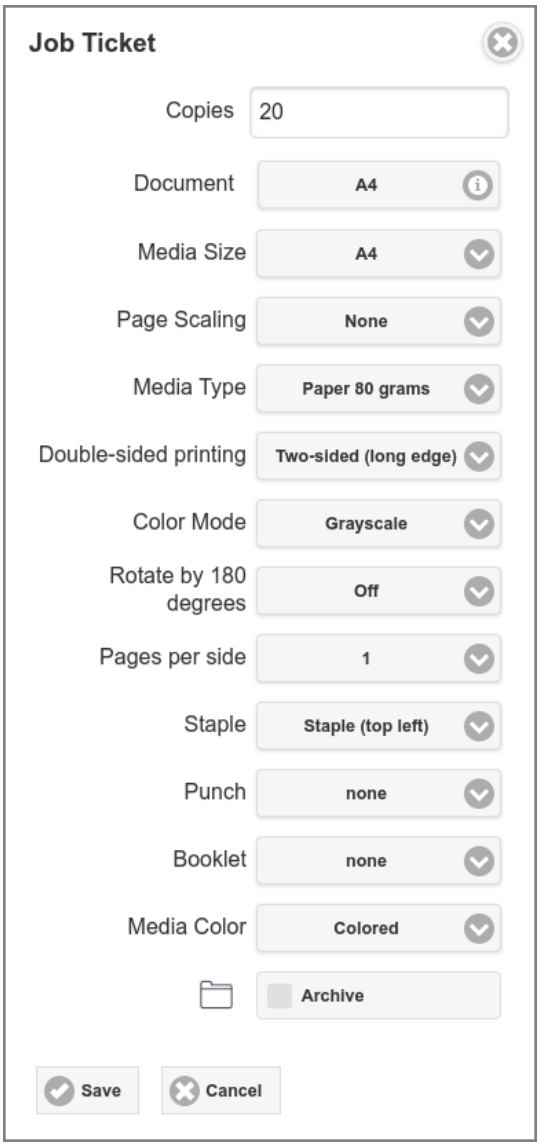


Figure 5.8. Job Tickets: Edit Ticket

Job Ticket Options can be changed (overruled) by the Job Ticket Operator. Costs are recalculated after Save.

Note

Document displays the page size of the PDF source document. The number of **Copies** can *not* be edited when Print Delegation is applied for *more than one* account. So, as long as a single account is involved the number of copies can be edited.

Page Scaling displays the selected value of the print-scaling option.

Important

For Reopened Job Tickets all options are read-only, except for the number of copies.

5.2.6. Job Ticket Print

This dialog pops up after pressing the Print... button in a Job Ticket List Item.

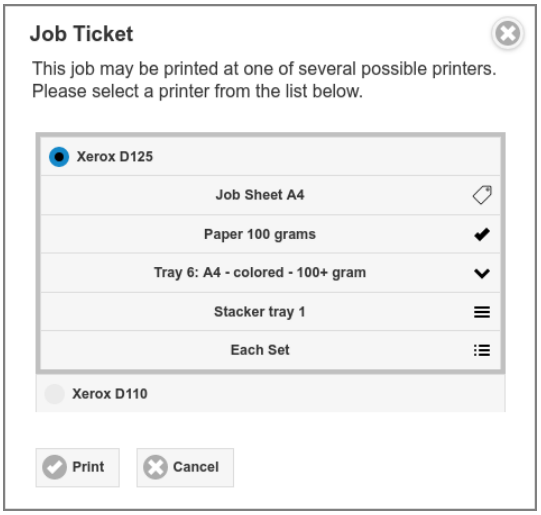


Figure 5.9. Job Tickets: Print Ticket

Redirect printers are shown from the Job Ticket Proxy Printer Group, that are compatible with ticket options. See Section 4.8.2.1, “Proxy Printer Identity” [132].

A selected printer may support the following settings, from top to bottom:

- Job Sheet Media Source : pick a source that is configured on the printer to hold the job sheet media-type.
- media-type : This setting is taken from the Ticket. It is a fixed choice in this context and can only be changed in the Ticket Edit Dialog. It is displayed for the operator to verify that the correct media type is loaded in the target media source.
- media-source: pick a source that is configured on the printer to hold the media-type.
- output-bin: pick a suitable output bin.
- org.savapage-finishings-jog-offset : select a finishing that shifts “Each Set” in the output-bin from the previous one by a small amount (or chose “none”).

After pressing the Print button, constraints are validated. When settings are valid, the dialog closes, and the status of the Ticket will be shown in the list as either Pending, Processing, or Completed.

Note

Printer settings are saved in the Web Session, and used as default for next Job Ticket prints.

- The last selected printer is used as default for the next print.
- The last jog-offset choice is used as default for *all* printers.
- The last media-source choice of a printer is used as default, if the requested media size can be assigned to this media-source. If not, the first (*preferred*) media-source that is compatible with the requested media size is used. See Section 4.8.2.3, “Media Sources” [134].

5.2.6.1. Job Ticket Print Pending

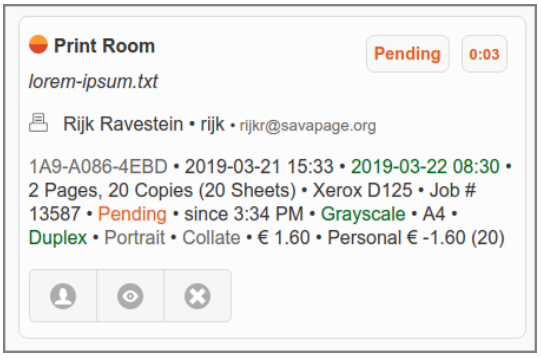


Figure 5.10. Job Tickets: Print Pending

When status is **Pending**, the job is waiting to be printed. By pressing Cancel, the job will be removed from the print queue and the Ticket will show status Canceled in the list.

Optionally the Print Job Supplier and Print Status Supplier are shown at the bottom. In the example below SavaPage supplied the job and PaperCut, as external print manager, supplied the status. When SavaPage supplies both Print Job and Status, this information is not shown.

Figure 5.11. Job Tickets: Print Pending (external)

> **Note**
>
> The top right corner of the job ticket entry displays the elapsed time since the Print command was issued.

5.2.6.2. Job Ticket Print Processing

When the job is actually printing, status **Pending** changes to **Processing**. By pressing Cancel printing is stopped, the job is removed from the print queue, and the Ticket will show status Canceled in the list.

5.2.6.3. Job Ticket Print Canceled

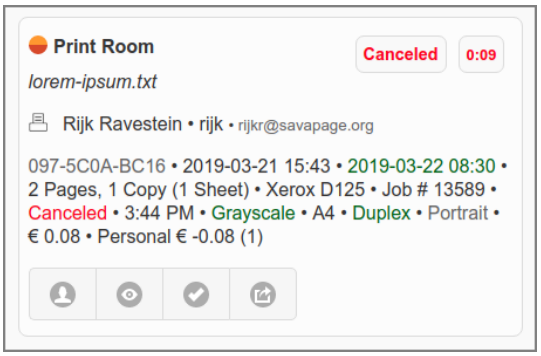


Figure 5.12. Job Tickets: Print Canceled

The print job is canceled. By pressing Close, the Ticket is closed and removed from the list, and the user is optionally notified by email or a Notification Plug-in. Retry to Print again, optionally to another printer.

5.2.6.4. Job Ticket Print Completed

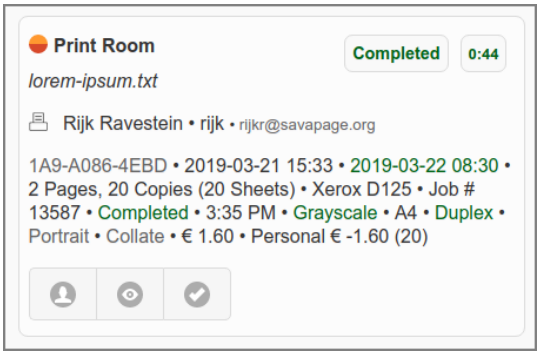


Figure 5.13. Job Tickets: Print Completed

When a ticket has been Completed, the proxy print succeeded and the print costs have been settled successfully with the relevant accounts. By pressing Close, the Ticket is closed and removed from the list, and the user is optionally notified by email or a Notification Plug-in: see Section 3.5.9, “Job Ticket Print” [64].

Optionally the Print Job Supplier and Print Status Supplier are shown at the bottom. See remarks at Section 5.2.6.1, “Job Ticket Print Pending” [218].

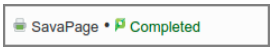


Figure 5.14. Job Tickets: Print Completed (external)

5.2.7. Job Ticket Settle

This dialog pops up after pressing the Settle... button in a Job Ticket List Item.

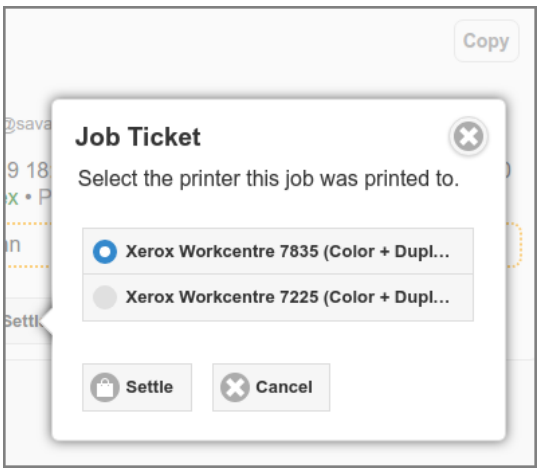


Figure 5.15. Job Tickets: Settle Ticket

After Settle, the ticket is effectuated without proxy printing. This implements the scenario where the Job Ticket Operator prints the attached PDF with an external program, and registers the printing occurrence (including financial cost) in SavaPage. After settlement, the Ticket is automatically closed and removed from the list. And, the user is optionally notified by email or a Notification Plug-in: see Section 3.5.9, “Job Ticket Print” [64].

> **Note**
>
> Settle is the only option to charge a Copy Job Ticket.

5.3. Closed Tickets

The Closed Ticket List is identical to the Documents List with an implicit selection of Type Ticket. Here you can query Job Tickets that were closed by the Job Ticket Operator, optionally by User ID of the Job Ticket Creator.

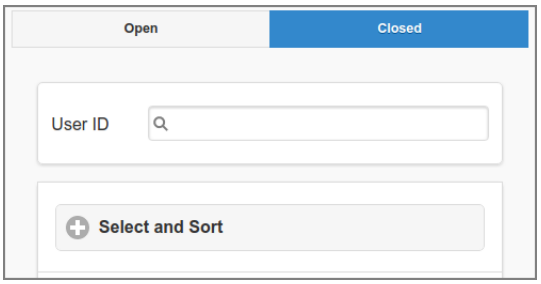


Figure 5.16. Job Tickets: Closed Ticket List

5.3.1. Job Ticket Refund

In case the print completed but costs were incorrectly charged, they can be refunded by pressing the refund button. The button is disabled as long as the Job Ticket is Open (the Close button is not pressed yet).

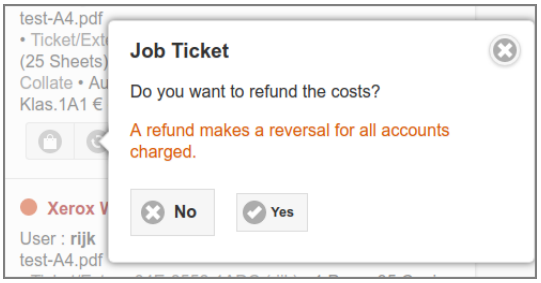


Figure 5.17. Job Ticket: Refund

Refunds are shown in the document transaction details (see screenshot below), and in the transaction list of the Personal, Group and Shared Accounts involved. See Section 3.8.2, “Transactions” [73] and Section 4.6.1, “Account List” [117].

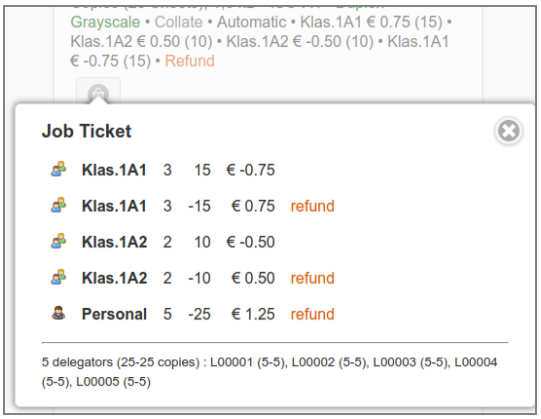


Figure 5.18. Job Ticket: Refund Transactions

> **Note**
>
> When both PaperCut Integration and Delegated Print with PaperCut are enabled, and the Job Ticket was printed to a PaperCut managed printer, refund transactions will also be created in PaperCut. See Section N.1.3, “PaperCut Delegated Print Accounting” [414].

5.3.2. Job Ticket Reversal

In case the print was Canceled or Aborted a reverse button is available to adapt counters and refund accounts charged. The button is disabled as long as the Job Ticket is Open (its Close button is not pressed yet). This is identical to the option shown in the Admin Web App: see Section 4.12.2, “Proxy Print Reversal” [196].

5.3.3. Job Ticket Reopen

If webapp.jobtickets.reopen.enable is set to Y, a closed single account job ticket that is archived or journalled in the Document Store can be reopened. This option is active in the Job Ticket Web App only.

- A reopened ticket has the same ticket number as the original with a + character appended.
- Just the number of copies (initialized to zero) of the reopened ticket can (must) be edited: other IPP options are read-only.
- A Print or Settle of a reopened ticket is *not* archived or journalled.
- The calculation of the delivery date/time is according to jobticket.delivery-* configuration properties.
- A refunded Ticket can't be reopened.
- The reopened ticket is tied to the Job Ticket Printer of the closed ticket. So, this printer must still be present.

The reopen option is offered with a + button in the document log item, as shown in the screenshot below.

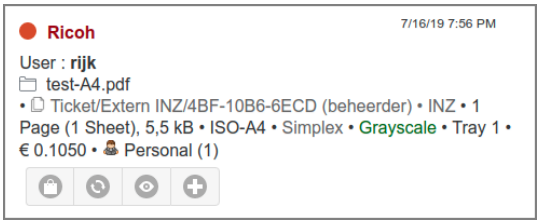


Figure 5.19. Job Ticket: Reopen Print Ticket

When the + button is pushed, the ticket is reopened and the button disabled to prevent reopening a second copy. Copy Jobs can be reopened as well, as shown below.

Figure 5.20. Job Ticket: Reopen Copy Ticket

The reopened ticket will show up in the Open Tickets list with zero copies, and thereby the Print and Settle options will be disabled, as shown in screenshot below.

Figure 5.21. Job Ticket: Reopened Print Ticket

After the reopened ticket is printed and closed it appears in the Closed Tickets list marked as “Reopened”. Note that this instance isn't archived or journalled, and therefore can't be reopened.

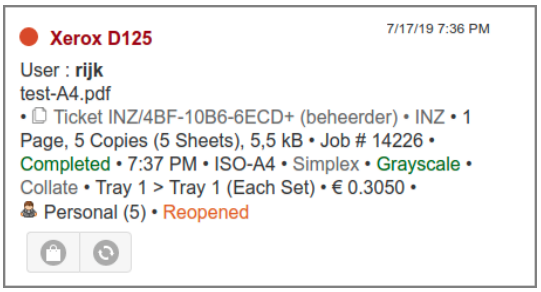


Figure 5.22. Job Ticket: Reopened Ticket Printed

5.4. Job Ticket Configuration Properties

Configuration property	Description
jobticket.notify-email.content-type.html	Set to `Y` (default) or `N`, to send Job Ticket email notifications as "text/html" (`Y`) or "text/plain (`N`).
jobticket.notify-email.completed.enable	Set to `Y` (default) or `N`, to enable/disable notification by email to owner of job ticket when ticket is completed. A standard email message is sent. Email content and layout can be customized if needed. See Section 20.2, “Email Templates” [304] and Section 20.2.6.2, “JobTicketCompleted Email” [309].
jobticket.notify-email.canceled.enable	Set to `Y` (default) or `N`, to enable/disable notification by email to owner of job ticket when ticket is canceled. A standard email message is sent. Email content and layout can be customized if needed. See Section 20.2, “Email Templates” [304] and Section 20.2.6.1, “JobTicketCanceled Email” [309].
webapp.jobtickets.list-size	The Number of Job Tickets shown in the list. Default: `10`.
webapp.jobtickets.list-size-min	The minimum Number of Job Tickets that can be shown in the list. A value of zero means all pending tickets are shown. Default: `5`.
webapp.jobtickets.list-size-max	The maximum Number of Job Tickets that can be shown in the list. Default: `50`.
webapp.jobtickets.cancel-all.enable	Set to `Y` (default) or `N`, to enable/disable the Cancel All... bulk action.
webapp.jobtickets.print-all.enable	Set to `Y` (default) or `N`, to enable/disable the Print All... bulk action.
webapp.jobtickets.close-all.enable	Set to `Y` (default) or `N`, to enable/disable the Close All... bulk action.

Configuration property	Description
webapp.jobtickets.reopen.enable	Set to `Y` or `N` (default), to enable/disable option to reopen a closed single account job ticket that is archived or journalled is the Document Store. See Section 5.3.3, “Job Ticket Reopen” [221].

Table 5.1. Job Ticket Print Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these items.

Chapter 6. Mail Tickets Web App

The Mail Tickets Web App can be accessed by users with role Mail Ticket Operator to dispatch Mail Print tickets.

The Web App can be reached at `https://savapage:8632/mailtickets`. See Appendix E, *URL Cheat Sheet* [362].

Mail Ticket Print is active when:

- Configuration property `print.imap.ticket.enable` is set to `Y`.
- Configuration property `print.imap.ticket.operator` is set.
- The Queue Journal function is enabled.
- Mail Print is configured and enabled.

> **Note**
>
> Mail Ticket Print can be used to allow unknown (guest) users to Mail Print and collect their print at a desk equipped with this Web App. A Mail Ticket Operator selects the Ticket ID as notified in an email reply to the Mail Print sender. Since the costs are shown before printing, the operator can charge the amount due to the guest user.

6.1. Login

The Mail Tickets Login screen is a variant of the User Login screen, with the following exception:

- Persons with role Mail Ticket Operator are allowed to log in. See Section 4.4.5.2, "User Roles" [103] how to assign admin rights to users.
- After a successful login the Mail Ticket Selection is shown.

6.2. Mail Ticket Selection

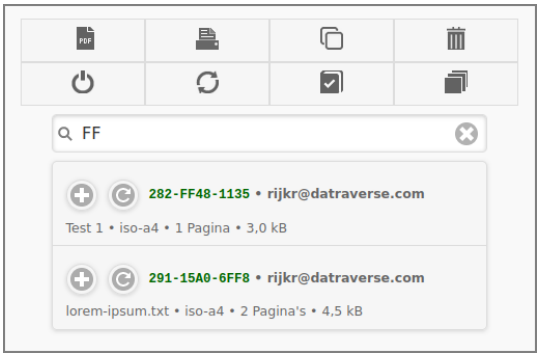


Figure 6.1. Mail Tickets Web App: Ticket selection

The Mail Ticket can be selected by ID. Press the `+` button to add the document to the SavaPages, or the button at the right to replace them.

Printing scenarios can start from here. All costs are charged to the User set in the `print.imap.ticket.operator` configuration property. Printing costs are shown and must be confirmed.

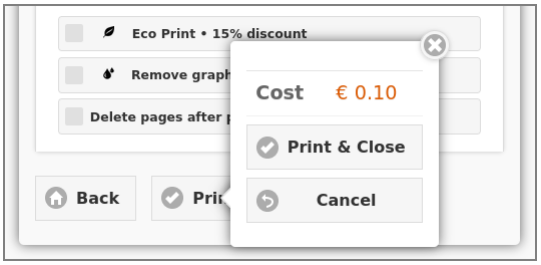


Figure 6.2. Mail Tickets Web App: Print Confirmation

6.3. Mail Ticket Configuration Properties

Configuration property	Description
print.imap.ticket.enable	Set to `Y` or `N` (default) to enable Mail Ticket Print.
print.imap.ticket.operator	The User (identified by User Name) to whom the Mail Ticket Print is delivered, if print.imap.ticket.include-known-users is `Y`, or if no user email match is found. Note: the User (operator) must have Queue Journal privilege.
print.imap.ticket.include-known-users	If `N` (default), the Mail Print is delivered to the user that matches the email adres of the sender, and no Mail Print ticket email notification is sent. If `Y`, or no user email match is found, the Mail Print is delivered at the print.imap.ticket.operator and a Mail Ticket Print email notification is returned to the sender.
print.imap.ticket.no-files-content.enable	When no attachments are present: if `Y` (default), the body of the email message is printed, if `N` the Mail Print is rejected.
print.imap.ticket.reply.content-type.html	Set to `Y` (default) to send Mail Print ticket email notification with content-type "text/html". If `N` notifications are send as plain text.

Table 6.1. Mail Ticket Print Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these items.

Chapter 7. Point-of-Sale Web App

The Point-of-Sale (POS) is a Web App used to deposit funds to a user's printing account, usually after receiving a cash or electronic payment. It can be accessed by users with role Web Cashier.

The POS Web App can be reached at `https://savapage:8632/pos`. See Appendix E, *URL Cheat Sheet* [362].

7.1. Login

The Point-of-Sale Login screen is a variant of the User Login screen, with the following exception:

- The internal `admin` user and Persons with role Web Cashier are allowed to log in. See Section 4.4.5.2, "User Roles" [103] how to assign admin rights to users.
- After a successful login the dialog in the leftmost tab is shown.

7.2. Sales

The **Sales** tab is the place to handle a user's purchase of an Item in a Shop on a Location. Buyers must use a Local Card Reader to identify themselves.

Note

- The **Sales** tab is disabled by default. The Swipe Card area can be enhanced with a password entry field where the RFID is captured. See Section 7.5, "Configuration Properties" [230].

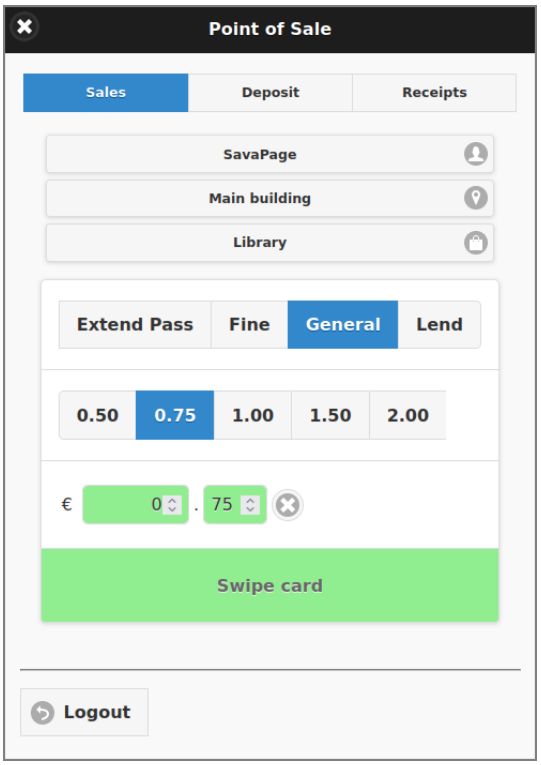


Figure 7.1. Point-of-Sale: Sales

If PaperCut Personal User Account is enabled, the amount is charged to the personal `PaperCut"` user account instead of `SavaPage` (as will be shown in the top of the dialog).

7.3. Deposit

The **Deposit** tab is the place to handle a user's deposit to their personal SavaPage account.

Note

The **Deposit** tab is enabled by default. See Section 7.5, “Configuration Properties” [230].

The figure below shows the initial content when first used or after the `Clear` button was pushed.

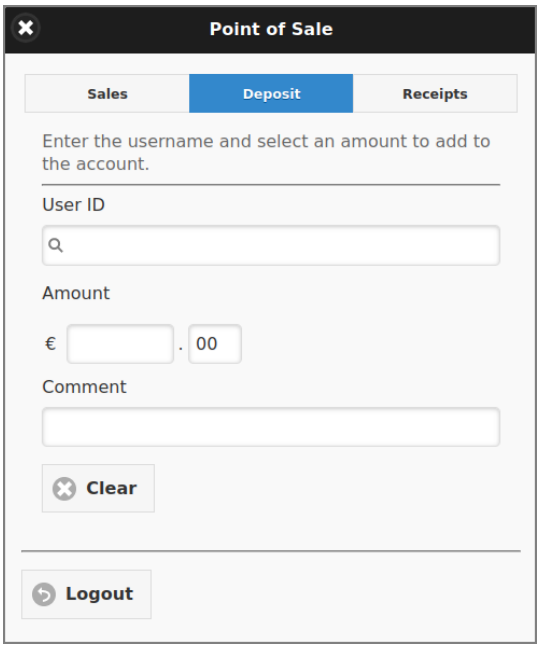


Figure 7.2. Point-of-Sale: Deposit Start

- **User ID**: a quick search entry field to select the User. See the figure below for a description.
- **Amount**: enter the integer and cents of the amount separately. The currency sign is taken from the Financial settings.
- **Payment method**: select one of the payment methods as specified in the Financial settings. When no methods are specified this field will not show.
- **Comment**: a short comment to denote the deposit.
- **Receipt**: select the email address if you want to send de receipt as PDF.

- **User ID** is a quick search entry field. By entering part of the user id or name, a pick-list of matching users is displayed below. The list is refreshed real-time as characters are entered (or deleted).
- By selecting the user from the list the entry field is replaced by the selected user id. Also, the current account balance, the name and email of the user is shown.

When all required field are entered the Deposit button will show.

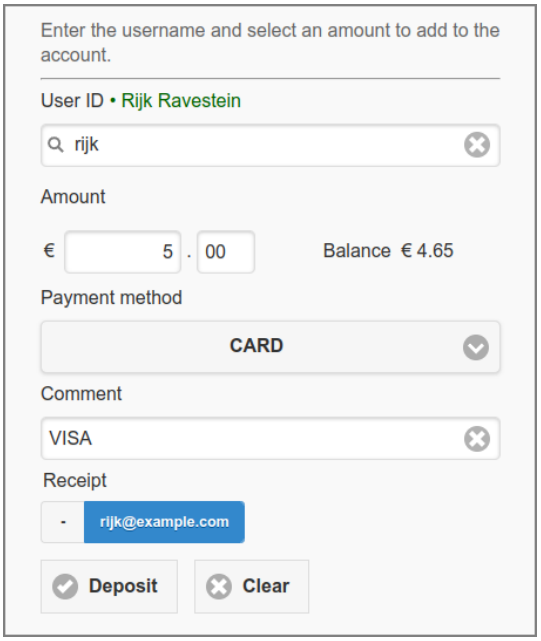


Figure 7.3. Point-of-Sale: Deposit Completed

Push the Deposit to make the deposit. As a result:

- When user's email address was selected, the receipt will be emailed to the User.
- The form will be cleared, with the focus on the User ID quick search field.

7.4. Receipts

The **Receipts** tab is the place to query deposit history.

> **Note**
>
> The **Receipts** tab is shown if the Deposit tab is enabled.

The figure below shows a content sample.

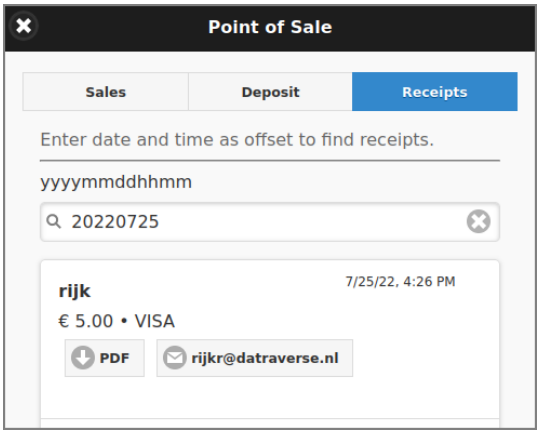


Figure 7.4. Point-of-Sale: Receipts

- By entering a date/time offset in the prescribed **yyyymmddhhmm** format, a pick-list of matching receipts is displayed, sorted in descending date/time order. The list is refreshed real-time as characters are entered (or deleted). Note: the date/time defaults to the current yyyymmdd date (today).
- Each entry in the list has buttons to download the Receipt PDF or email it to the User (again).

7.5. Configuration Properties

Configuration property	Description
financial.pos.sales.enable	Set to `N` (default) or `Y`, to disable/enable the Sales tab.
financial.pos.sales.credit-limit.enable	Set to `Y` (default) or `N`, to enable/disable the `financial.pos.sales.credit-limit` amount.
financial.pos.sales.credit-limit	Credit limit amount applied for all users (Default: 0.00). This overrules any individual user credit limit. See Section 3.10.5, “Financial” [83].
financial.pos.sales.card-as-password	Determines if the Swipe Card area is enhanced with a password entry field where the RFID is captured: `OFF` (default) | `ON` | `AUTO`. The `AUTO` value will turn the entry field `ON` for mobile devices. Note: This value can be overruled for a POS with a special URL parameter. See Appendix E, *URL Cheat Sheet* [362].
financial.pos.deposit.enable	Set to `Y` (default) or `N`, to enable/disable the Deposit and Receipts tab.
webapp.pos.sound.success	Path to a sound file (mp3, wav, ...) relative to `/opt/savapage/server/custom/web/` that is played after a POS transaction finished successfully.
webapp.pos.sound.failure	Path to a sound file (mp3, wav, ...) relative to `/opt/savapage/server/custom/web/` that is played after a POS transaction failed.
Sales labels	
financial.pos.sales.label.locations	A comma separated list of POS Sales Locations to be applied as sales label prefix. Each location on the list is formatted as "LOC/location", where "LOC" is a unique N-letter upper-case mnemonic, "/" is a fixed separator, "location" is a case-sensitive single word used in UI context. E.g. "A/LocationA,B/LocationB,C/LocationC". If a Web Cashier selects location "B", the sales prefix is "B". Note: A POS can be fixed to a Location with a special URL parameter. See Appendix E, *URL Cheat Sheet* [362].
financial.pos.sales.label.locations.enable	Set to `Y` or `N` (default) to enable/disable `financial.pos.sales.label.locations`.
financial.pos.sales.label.shops	A comma separated list of POS Sales Shops in a Location, to be applied as sales label *after* the location label. Each shop on the list is formatted as "SHOP/shop", where "SHOP" is a unique N-letter upper-case mnemonic, "/" is a fixed separator, and "shop" is a case-sensitive single word used in UI context. E.g. "F/Frontdesk,L/Library". If a Web Cashier selects shop "L" in location "B", the sales label is "B/L". A shop can be restricted to one or more locations by appending the location mnemonics. E.g. "`F/Frontdesk/A/C`" restricts the "`F/Frontdesk`" shop to "`A`" and "`C`" locations. Note: A POS can be fixed to a Shop with a special URL parameter. See Appendix E, *URL Cheat Sheet* [362].
financial.pos.sales.label.shops.enable	Set to `Y` or `N` (default) to enable /disable `financial.pos.sales.label.shops`.

Configuration property	Description
financial.pos.sales.label.items	A comma separated list of POS Sales Items in a Shop to be applied as sales label *after* the shop label. Each item on the list is formatted as "ITEM/item", where "ITEM" is a unique N-letter upper-case mnemonic, "/" is a fixed separator, and "item" is a case-sensitive single word used in UI context. E.g. "BK/Book,CA/Cafetaria,HW/Hardware". If a Web Cashier selects item "CA" in shop "L" of location "B", the sales label is "B/L/CA". An item can be restricted for use with one or more shops by appending the shop mnemonics. E.g. "`BK/Book/F/L`" restricts the "`BK/Book`" item to "`F`" and "`L`" shops.
financial.pos.sales.label.items.enable	Set to `Y` or `N` (default) to enable/disable `financial.pos.sales.label.items`.
financial.pos.sales.label.prices	A comma separated list of POS Sales Prices for Items. Each price on the list is formatted as "PRICE/price", where "PRICE" is a unique N-digit amount in cents, "/" is a fixed separator, and "price" is the case-sensitive single word price in UI context. Prices are shown as preset choices. E.g. "50/0.50,100/1.00,150/1.50,850/8.50". A price can be restricted for use with one or more items by appending the item mnemonics. E.g. "`850/8.50/BK/HW`" restricts the "`850/8.50`" price to "`BK`" and "`HW`" items.
financial.pos.sales.label.prices.enable	Set to `Y` or `N` (default) to enable/disable `financial.pos.sales.label.prices`.
webapp.pos.sales.label.items.button-max	The maximum number of `financial.pos.sales.label.items` that make items show as buttons before a select list is shown. Default: 7.
webapp.pos.sales.label.prices.button-max	The maximum number of `financial.pos.sales.label.prices` that make prices show as buttons before a select list is shown. Default: 10.

Table 7.1. Point-of-Sale Configuration Properties

Chapter 8. Payment Web App

The Payment Web App is a specialized User Web App with the sole function to increment a personal account via a Payment Gateway. It can be accessed by any user with Financial Editor privilege. The Web App is unavailable if a Generic Payment Gateway is not configured or is set offline in the Admin Dashboard.

The Web App can be reached at `https://savapage:8632/payment`. See Appendix E, *URL Cheat Sheet* [362].

8.1. Login

The Payment Login screen is like a User Login with the exception that users must have Financial Editor privilege to log in.

8.2. Payment Page

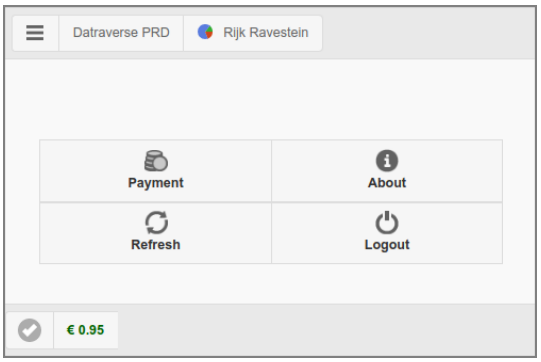


Figure 8.1. Payment Web App: Main Page

The main page is restricted to just a Payment button with a selection of generic options as present in the User Web App. The button opens the Payment dialog.

A Help button that opens a custom URL in a new tab is shown instead of the About button, if the URL is set with configuration property "`webapp.payment.help.url`" and enabled with "`webapp.payment.help.url.enable`" default value `Y` (disable with value `N`).

By default, button text on mobile devices only: on desktops a hover text is shown. You can change this behavior by setting configuration property `webapp.payment.main.nav-button-text` to value `ON` (button text is always shown), `OFF` (button text is never shown) or `AUTO` (button text is shown on mobile devices only).

See Section 4.11.15, “Config Editor” [189] on how to change configuration properties.

If PaperCut Personal User Account is enabled, the account balance in the footer is prefixed with the PaperCut icon.

8.2.1. Payment Dialog

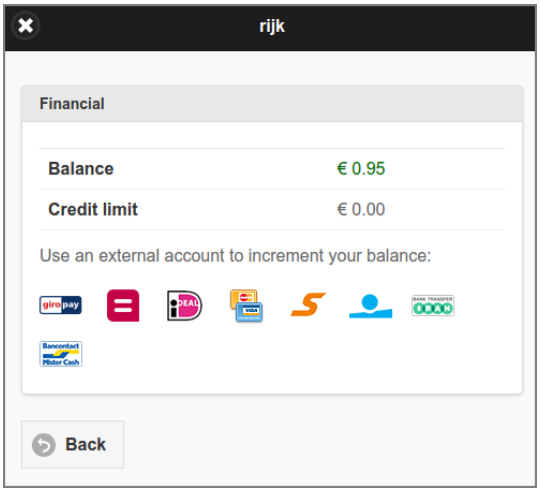


Figure 8.2. Payment Web App: Payment

This dialog is a restricted version of Section 3.10.5, “Financial” [83]. Just the external accounts to increment user balance are shown. If PaperCut Personal User Account is enabled, the **Balance** label is prefixed with the PaperCut icon and **Credit limit** is not shown.

Chapter 9. Print Site Web App

Preview

Print Site Web App is a Feature Preview that is partly operational and impermanent. The preview is offered to provoke feedback based on real world use. Please tell us your experience.

A Print Site is a location where printers are set up for self-service.

In the typical use case, SavaPage Print Portal is set up as public Internet service, where authorized users choose a printer to print the documents that were collected with Internet Print, Web Print or Mail Print. To enforce Secure Printing, print jobs are held and must be released by the user after authentication at the very printer.

The Print Site Web App is meant for users with role Print Site Operator to support users with self-service printing scenario's.

The Web App can be reached at `https://savapage:8632/printsite`. See Appendix E, *URL Cheat Sheet* [362].

9.1. Login

The Print Site Login screen is a variant of the User Login screen, with the following exception:

- The internal `admin` user and Persons with role Print Site Operator are allowed to log in. See Section 4.4.5.2, “User Roles” [103] how to assign admin rights to users.
- After a successful login the menu is shown.

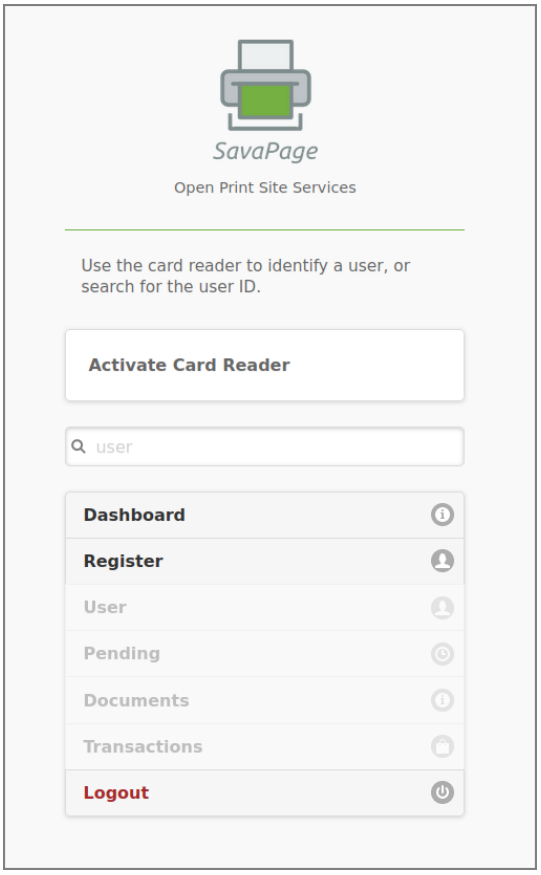


Figure 9.1. Print Site Web App: Menu

9.2. PaperCut Integration

If PaperCut Integration is enabled, PaperCut follow-me printing and copying of the glass can be integrated, with all costs charged to the personal PaperCut account of the user.

In that case SavaPage Print Site will be the single application the Print Site Operator has to deal with. Any interaction with the PaperCut back-end is handled behind the scenes.

9.2.1. Users

When SavaPage and PaperCut share a common User Source, user synchronization and authentication can be set-up identically in both environments.

When a common User Source is absent, SavaPage Internal Users can be synchronized with and authenticated in PaperCut via the SavaPage as Sync Source in PaperCut.

9.2.2. Financial

The PaperCut Personal User Account must activated as the leading account for personal financial transactions and credit.

Users can recharge their account balance via Manual Payment at the counter, with a Voucher or Payment Gateway.

9.2.3. Printing

Personal Print to PaperCut must be activated. This makes PaperCut printing cost leading. See Section A.2.3, “Personal Print - PaperCut Scenario” [321].

Chapter 10. PDF/PGP Verification

Preview

PDF/PGP Verification originated and is prototyped in the SavaPage Community. It has no formal status outside, but could become a more widely accepted Open Standard. This Feature Preview is offered to provoke feedback on the verification method and reference implementation. Please share your experiences and requirements.

10.1. PDF/PGP in a Nutshell

Many organizations who are bound to legal and regulatory requirements use PKI[1] based services to verify *authenticity* and *integrity* of PDF documents. X.509[2] is the "de facto" standard for this security measure.

While some have a self-imposed X.509 policy, most organizations are not aware of security issues or are deterred by PKI requirements, that include third-party Certificate Authorities (CAs), Time Stamping Authorities (TSAs), and PDF signature-compliant PDF readers.

For those organizations PDF/PGP Verification is a simple OpenPGP[3] based PKI alternative. It works as follows:

1. The PDF document is *signed* by including its detached OpenPGP signature as PDF comment.
2. The PDF is *verified* by stripping the comment OpenPGP signature, and using it to verify the remainder PDF.

Organizations can easily use their own Web Site to implement a PDF/PGP Verification Service. This is the use-case:

1. Published PDF documents are PDF/PGP signed and have a visible URL link to the Verification Service at the top of the first page.
2. A User, who received the PDF through whatever channel, wants to verify its authenticity and integrity, and clicks the link.
3. The Web App opens in the browser and invites the user to upload the PDF for verification.
4. The User trusts the Secure Connection and Website Identity (SSL certificate) and uploads the PDF.
5. The Web App uses the Public OpenPGP Key of the trusted signer to verify the uploaded PDF and communicates the verdict to the User.

SavaPage implements a reference PDF/PGP Verification Service where PDF documents are signed and verified with the server's OpenPGP Key Pair.

Caution

PDF/PGP Verification will surely not hold against stringent certificate-based legal and regulatory requirements. On the other hand, for many organizations it will lower the threshold to adopt a simple and pretty-good security policy.

10.2. PDF/PGP Signature

PDF/PGP verification can be configured with the following properties:

[1] https://en.wikipedia.org/wiki/Public_key_infrastructure
[2] https://en.wikipedia.org/wiki/X.509
[3] https://www.openpgp.org/

Configuration property	Description
pdfpgp.verification.enable	Set to `Y` or `N` (default), to enable/disable PDF/PGP verification. When enabled *and* the OpenPGP Key Pair is present on the server, users who have the Privilege to Sign can apply an OpenPGP Signature when creating a PDF for Download or Send.
pdfpgp.verification.host	DNS host name of the Verification Service. When blank, the IPv4 address of the SavaPage server is used
pdfpgp.verification.port	Optional TCP/IP port of the Verification Service. Host and port are used to compose the URL: `https://host[:port]/verify/pdf` See Appendix E, *URL Cheat Sheet* [362].

Table 10.1. PDF/PGP Signing Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to enter these properties.

A signed PDF file has the `Verify . . .` button and the SavaPage server's OpenPGP Fingerprint displayed at top of the first page. When the button is pressed, the Verification Web App opens.

Figure 10.1. PDF/PGP Verify Link

Also, the signed PDF contains one or two Public Key file Attachments:

1. The OpenPGP Public Key file `creator.asc` of the SavaPage server.
2. If the User who created the PDF has a registered OpenPGP Public Key ID, the public key file `author.asc` as retrieved from the configured Public Key Server (and persisted in the user's SafePages home).

10.3. PDF/PGP Verification Web App

The PDF/PGP Verification Web App can be configured with the following properties:

Configuration property	Description
webapp.pdfpgp.enable	Set to `Y` or `N` (default), to enable/disable the PDF/PGP Verification Web App.
webapp.pdfpgp.max-upload-file-mb	The maximum MB size of the PDF that can be uploaded for verification. Default: 10

Table 10.2. PDF/PGP Verification Web App Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to enter these properties.

The Web App invites to choose a PDF for verification. Multiple files can be selected for upload.

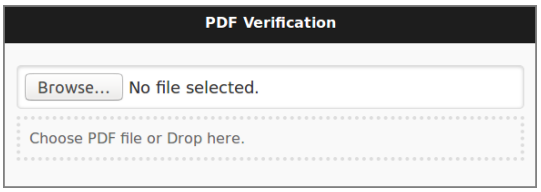


Figure 10.2. PDF/PGP Verification - Upload

And shows success or failure when done.

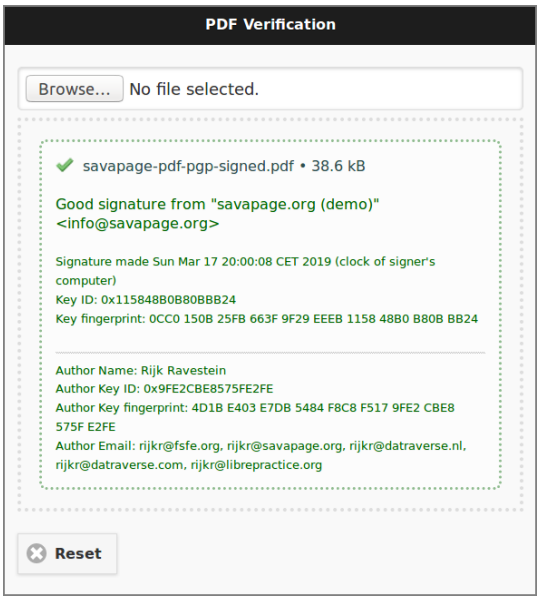


Figure 10.3. PDF/PGP Verification - Good signature

Note

When the OpenPGP Public Key file `author.asc` is found as PDF attachment, its data is displayed as Author identification. See Section 10.2, “PDF/PGP Signature” [237].

Figure 10.4. PDF/PGP Verification - Public key not found

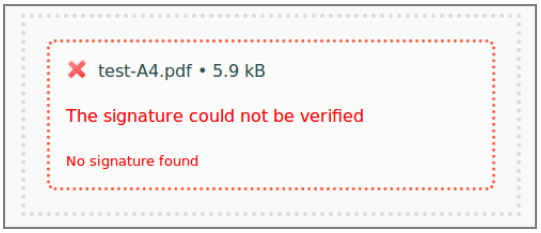


Figure 10.5. PDF/PGP Verification - Signature not found

Chapter 11. User Client

The User Client is a Java application for desktops and notebooks that resides in the *system tray*.

According to the Oracle Java Documentation[1]:

"The system tray is a specialized area of the desktop where users can access currently running programs. This area may be referred to differently on various operating systems. On Microsoft Windows, the system tray is referred to as the Taskbar Status Area, while on the GNU Network Object Model Environment (GNOME) Desktop it is referred to as the Notification Area. On K Desktop Environment (KDE) this area is referred to as the System Tray. However, on each system the tray area is shared by all applications running on the desktop."

The SavaPage User Client is provided as a notifier of personal user events like:

- A successful print to SavaPage. See Chapter 14, *SavaPage as Printer* [260].
- A Proxy Printer started printing one of your jobs. See Section 4.8, "Proxy Printers" [127]

A notification message is typically displayed near the SavaPage tray icon in the form of a balloon (Windows) or message box (GNU/Linux, macOS).

The User Web App opens for the authenticated user at a double-click on the tray icon, a click in the notification message or selecting the Open Web App... item from the tray icon context menu. When the User Client is trusted as authentication source no extra login is needed.

User Client authentication is explained in Section 4.11.14.1, "User Client Authentication" [178].

Client access can be restricted by IP address with these configuration properties:

Configuration property	Description
cliapp.ip-addresses-allowed	A CIDR Set of Client IP addresses that are allowed to use the User Client App (when void, all client addresses are allowed).
cliapp.auth.ip-addresses-denied.enable	Enable (Y) or Disable (N) User Client Authentication for clients that are denied for their IP address.

Table 11.1. User Client Access Configuration Properties

See Section 4.11.15, "Config Editor" [189] on how to enter these properties.

Note

Rate Limiting is applied to login failures.

Important

When using the User Client concurrently with the User Web App and Proxy Print Authentication you are strongly advised to use an external database like PostgreSQL. See Chapter 21, *Using an External Database* [311].

[1] https://docs.oracle.com/javase/tutorial/uiswing/misc/systemtray.html

Warning

Java applications with system tray icons do not work properly with GNOME Shell. This is a persistent Java bug[2] that is still not resolved. As a workaround, use the `--anchor` command line option for an alternative display.

11.1. User Client Options

In order of precedence, User Client options can be set ...

1. On the Command-line.
2. As value in a `client.properties` file. An annotated template is available in the `/opt/savapage/client/app/config/` directory.
3. As Configuration Property.

```
usage: savapage-client <options>
      --anchor <ne|nw|se|sw>     Show on desktop at anchor position instead of tray.
                                 --notify-send switch is auto activated (Linux only).
   -d,--debug                    Write debug messages to the log file.
   -h,--help                     Display help text in GUI.
      --help-tui                 Display help text in TUI.
      --log-dir <dir>            Log file directory. Default: $HOME
      --notify-send              Use 'notify-send' command to send desktop
                                 notifications (Linux only).
   -p,--print-dialog             Show action dialog at print-in event.
      --passkey <key>            The admin passkey (optional).
      --print-dialog-btn <arg>   Button text on print-in action dialog for opening
                                 User Web App (optional).
      --print-url-query <arg>    URL query for opening User Web App at print-in event
                                 (optional).
      --properties <file>        File with default command-line options (optional).
                                 Default: $APP/config/client.properties
      --server-host <arg>        IP address or name of SavaPage server
      --server-port <number>     SSL port of SavaPage server (optional). Default:
                                 8632.
      --user <name>              A different username than current user $USER
                                 (optional).
   -x,--hide-exit                Hide the "Exit" menuitem (optional). Default: false.
```

Note

The `passkey` option can also be applied as environment variable `SAVAPAGE_CLIAPP_ADMIN_PASSKEY`. This is the preferred way to use in generic login scripts, since the command-line option might be visible in system process viewers.

On Debian, the `notify-send` utility is part of the `libnotify-bin` package.

Configuration property	Command-Line Option
cliapp.print-in.url-querycliapp.print-in.url-query	`--print-url-query` The query string to be appended to the base URL when opening the User Web App in response to a print-in event. Do *not* prefix the value with a ? or & character.
cliapp.print-in.dialog.button-open	`--print-dialog-btn`

[2] https://bugzilla.redhat.com/show_bug.cgi?id=1014448

Configuration property	Command-Line Option
	Action button text on print-in action dialog for opening User Web App.

Table 11.2. User Client Options Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to enter these items.

11.2. User Client Deployment

Since the software is written in Java it can be deployed on any platform where Java SE 11 or higher is installed, like Microsoft Windows, macOS and GNU/Linux. The User Client software is installed on the server in the `/opt/savapage/client/app` directory.

Caution

Running the User Client directly from a network share can cause performance problems, resulting in an unresponsive application. Therefore is it recommended to run the client from a local copy.

Note

The User Client is resilient in the event that the connection with the server is (temporarily) lost. The connection will be restored automatically after a short break. If the connection is lost for a longer period of time, the user is notified with a pop-up message with the choice to try again or to close the application.

11.2.1. Deployment on Windows

The `savapage-client.bat` batch file that starts the application is available in the `/opt/savapage/client/app` directory on the server.

Note

Make sure to enable Balloon Notification in Windows Group Policy to allow users to see notification messages.

11.2.2. Deployment on macOS

The `savapage-client` shell script that starts the application is available in the `/opt/savapage/client/app` directory on the server.

11.2.3. Deployment on GNU/Linux

The `savapage-client` shell script that starts the application is available in the `/opt/savapage/client/app` directory on the server.

Chapter 12. SavaPage Financial

SavaPage Financial captures many aspects of user activity. Obviously, proxy printing is the main trigger for financial accountability and monitoring, since it consumes tangible resources like paper, ink and toner. This chapter introduces the main financial concepts with references to more detailed parts of the manual.

- Account are used to register financial status (balance) and history (transactions). SavaPage has three types of accounts:
  - *User Account* : The personal account of an User, optionally restricted by a Credit Limit. See Section 4.4.5.8, “Financial” [106].
  - *Shared Account* : Shared accounts act as *cost center* to track printing expenses in a specific area. It does not have a credit Limit: its balance is initialized to zero and is allowed to count down into the negative.
  - *Group Account* : A shared account that is tied to a User Group by name.

  Shared Accounts are discussed in Section 4.6, “Accounts” [117].

- Printing costs are configured per Proxy Printer. Pay-per-Print is active for each Proxy Printer that has costs greater than zero.
  - Section 4.8.2.3, “Media Sources” [134]
  - Section 4.8.2.4, “Manual Media Sizes” [136]

- Printing costs are charged to Accounts.

- Users get feedback about printing costs and their personal account balance.
  - Section 3.5.6, “Direct Print Release” [57]
  - Section 3.3.2.3, “Hold Print Jobs” [33].
  - Section 3.3.2, “Footer” [32]
  - Section 3.10.5, “Financial” [83]

- Restricted users can upgrade their account balance with vouchers (pre-paid printing cards), by making a deposit at a point-of-sale, or by transferring money from an external account.
  - Section 4.15, “Vouchers” [206]
  - Section 3.10.6, “Redeem Voucher” [84]
  - Chapter 7, *Point-of-Sale Web App* [226]
  - Section M.1.1, “Payment Gateway Plug-in” [405]

- All financial transactions can be inspected by administrators. Users can inspect their own transactions.
  - Section 4.4.1, “User List” [98]
  - Section 3.8.2, “Transactions” [73]

- Global financial options can be set by administrators.
  - Section 4.11.11, “Financial” [172]

Chapter 13. SavaPage on GNU/Linux

This section is a supplement to the Install Guide (see Chapter 2, *Server Installation* [11]). It provides an in-depth explanation of the GNU/Linux installation process, the directory layout and tools involved.

Information in this chapter is technical in nature. It is expected that readers have prior experience with:

- The Unix command line environment
- Unix file permissions

13.1. The Installation Process

SavaPage is supplied as a pre-compiled self-installing application. The installation process is designed to work with all major GNU/Linux distributions. To be sure if your GNU/Linux distro is supported out of the box, please check Section 1.2, "System Requirements" [3]. Due to the varied nature of some installations and administrator preferences, often some manual configuration is required. This section describes the installation process in detail as well as some additional options available to system administrators.

13.1.1. Manual extraction

SavaPage is supplied in a self-extracting, self-installing archive. The archive is simply a `tar` archive compressed with `gzip`, and headed with a shell script to facilitate self-extracting. After extraction is complete, the installation script named `install` is executed to begin the install process in the directory where the archive resides (usually `/opt/savapage`). Some system administrators may like to inspect the contents of the archive, and possibly the installation process itself prior to the actual install.

The self-extracting installer takes a number of command line arguments. The `-e` option will extract the archive into the current working directory ready for inspection. With the `-n` switch the `-i` install will be non-interactive. With this mode you implicitly agree with the AGPL license, and root tasks are collected in a `MUST-RUN-AS-ROOT` script located in the install directory. This script must be run manually as `root` after the installation. Further usage information is available via the `-h` switch.

```
_____________________________________________________________________________
SavaPage Install (c) 2010-2018, Datraverse B.V.

License: GNU AGPL version 3 or later <https://www.gnu.org/licenses/agpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Usage: savapage-setup-1.0.0-linux-x64.bin [-h|-i|-e|-l] [-n] [-v] [FILE]...
        -h      This help text.
        -i      Install after extracting the files (default).
        -n      Non-interactive install: execute MUST-RUN-AS-ROOT afterwards.
        -e      Extract all files or a FILE list and exit without installing.
        -l      List the contents of the archive and exit without extracting.
        -v      Verbose. Print the names of the files as they are extracted.
```

13.1.2. The install process

Even though the majority of the installation process is completed under the identity of the system user account called `savapage`, most administrators would like to know what the install process does. The main steps are outlined in the next paragraphs.

13.1.2.1. Extraction

The first stage in the install process extracts the archive to `/tmp` or a location as defined by an environment variable `TMPDIR`. The command-line programs **tar** and **gunzip** are used during this phase.

13.1.2.2. Installation

After extraction is complete the **install** script is called. The current directory is passed as `-d` option argument, to be used as install location. Also, the `-n` switch, used at the setup binary, is propagated to this script. The extracted files are copied to the install location. *Care is taken not to overwrite any existing data or configuration files if this is an install-over-the-top upgrade.*

```
_____________________________________________________________________________
SavaPage Install (c) 2010-2018, Datraverse B.V.

License: GNU AGPL version 3 or later <https://www.gnu.org/licenses/agpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Usage: install [OPTION]...
        -d <dir>  The target location, usually the /opt/savapage directory.
        -h        This help text.
        -n        Non-interactive install: execute MUST-RUN-AS-ROOT afterwards.
```

13.1.2.3. Permissions

To ensure the default installation is secure by default, permissions are applied to key files. The following `/opt/savapage/server` areas of the application are restricted to the `savapage` user only:

Area	Description
`data/`	This directory contains application data including database files. Some of this data may contain sensitive information.
`data/server.properties`	Contains server configuration properties. See `server.properties` in Section 13.5, “Advanced Configuration” [248].
`data/admin.properties`	Contains the hashed password of the reserved internal `admin` user.
`data/jmxremote.password`	Contains the plain text password of the reserved JMX `admin` user. See Section 4.11.14.4, “JMX Agent” [180].
`data/jmxremote.ks`	The private keystore used by the JMX Agent.
`bin/linux-x64/`	This directory contains the `savapage-pam` setuid-root binary. Even though the binary is no use to an end-user or hacker, good security practice stipulates that we should only allow the `savapage` user access to this directory.

Table 13.1. Secured Application Areas

Permissions can be checked and re-applied any time after the installation by running the script:

```
/opt/savapage/server/bin/linux-x64/setperms
```

13.1.2.4. Firewall

The SavaPage Application Server runs in a Java Virtual Machine (JVM) process and listens on ports 8631 and 8632 and 8633 (SSL). These ports are used for Web App access, printing and other services. Ensure that any firewall or local IP filtering software such as `iptables` is set to allow local network traffic access to this ports.

13.1.2.5. Root Level Tasks

A small part of the install process needs to run as the `root` account. The tasks conducted as root include:

- Setting the `/opt/savapage/server/bin/linux-x64/savapage-pam` binary as setuid-root. This binary is used for password verification.
- Installing the `/opt/savapage/providers/cups/linux-x64/savapage-notifier` binary as CUPS event notifier. This binary is used to send CUPS printer and print job status events to the central SavaPage server. See Section 13.5.7.1, "CUPS Notifier" [253].
- Setting up a **systemd** unit for GNU/Linux systems that use the systemd service manager. This is done by creating a `savapage.service` file in the systemd unit library. Depending on the distribution the unit will either be created in the `/lib/systemd/system/` or `/usr/lib/systemd/system/` directory. The unit is started and enabled.
- Setting up a *custom* **systemd** unit for the CUPS scheduler **cupsd**.
  - When the scheduler is run from systemd some systems pass the `-l` parameter so `cupsd` is run on demand by socket and path activation. The advantage of this setup is that CUPS is activated when needed, saving precious boot time and resources, and deactivated again after being idle for a while. This lazy activation scenario is efficient for desktop systems that print occasionally and for which printing is not time critical.
  - However, dedicated print systems like SavaPage, that intensively use IPP to communicate with CUPS, need CUPS to be full-time activated. Therefore a custom systemd `cups.service` unit is installed in `/etc/systemd/system/` to override the default `/lib/systemd/system/cups.service` shipped with the CUPS package. This custom unit starts `cupsd` with the `-f` parameter so it runs steadily in the foreground (without dependencies for `cups.socket` and `cups.path`).
- Setting up **SysV** style start scripts for Debian based systems that use the SysV service manager. This is done by placing symlinks in the:

```
/etc/init.d/
/etc/rc3.d/
/etc/rc5.d/
and so on...
```

If the administrator decides not to run the root-level tasks during the install process, the tasks can be run again post-install by executing the shell scripts:

`/opt/savapage/server/bin/linux-x64/roottasks`

and ...

`/opt/savapage/providers/cups/linux-x64/roottasks`

Alternatively the administrator can view the script and make the required changes by hand.

13.2. Logs

The main application logging is available via the Application Log section of the Administrator Web App. Additional advanced level logging is maintained in plain text files located at:

`/opt/savapage/server/logs/*`

Administrators may wish to consult these logs when attempting to diagnose or troubleshoot problems.

Log file	Description
`adminpublisher.log`	Messages published to the Admin Web App Dashboard. See Section 4.3.8, "Real-time Activity" [98].

Log file	Description
`gc.log.*`	JVM Garbage Collect log files.
`paymentgateway.log`	Payment Gateway events. See Section M.1.1, “Payment Gateway Plug-in” [405].
`proxy-print.log`	See Section K.7, “proxy-print.log” [396]
`server.log`	General purpose log file for all server activities.
`service.log`	Messages written to `stderr` and `stdout` from server process.

Table 13.2. Log Files

13.3. OpenPGP Settings

SavaPage can be configured to use OpenPGP[1] for PGP/MIME email signing and encryption. See Section 4.11.5, “Mail” [162].

The first step is to set the Public and Secret Key of the SavaPage server instance in the `server.properties` configuration file, as shown in the table below.

Key	Description
`pgp.publickey.file`	The path of the ASCII Armored public OpenPGP key. Relative to: `/opt/savapage/server/data/`
`pgp.secretkey.file`	The path of the ASCII Armored secret (private) OpenPGP key. Relative to: `/opt/savapage/server/data/`
`pgp.secretkey.passphrase`	The pass-phrase of the `openpgp.secretkey.file`

Table 13.3. OpenPGP server.properties

OpenPGP server information is shown on the Admin Dashboard.

The Secret Key of the server is used to sign e-mail content. Content encryption is done with the recipient's public key, when present. See Section 4.4.5.7, “OpenPGP” [105].

Next, PGP/MIME must be activated with the following configuration properties:

Configuration property	Description
mail.pgp.mime.sign	Set to `Y` (default) or `N`, to enable/disable PGP/MIME signing.
mail.pgp.mime.encrypt	Set to `Y` (default) or `N`, to enable/disable PGP/MIME encryption.
pgp.pks.url	URL of an OpenPGP Public Key Server. Required if OpenPGP is configured.

Table 13.4. OpenPGP Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these properties.

Preview

OpenPGP Server Properties are also used for the PDF/PGP Verification Feature Preview.

[1] https://www.openpgp.org/

13.4. IP Address Restriction

SavaPage uses CIDR Notation for IP address restriction where needed.

13.4.1. CIDR Notation

Classless Inter-Domain Routing[2] (CIDR) is developed in the 1990s as a standard scheme for routing network traffic across the Internet.

CIDR notation is a syntax for specifying IP addresses and their associated prefix size, the latter being equivalent to the number of leading 1 bits in the routing prefix mask. The notation starts with an IP address expressed according to the standards of IPv4 or IPv6[3]. It is followed by a separator character, the slash (/) character, and the prefix size expressed as a decimal number.

Some examples:

- `172.16.0.0/24` represents the given IPv4 address and its associated routing prefix `172.16.0.0`, or equivalently, its subnet mask `255.255.255.0`. This represents the host address range `172.16.0.0` - `172.16.0.255`.
- CIDR `192.168.1.40/32` represents the single IPv4 address `192.168.1.40`.
- CIDR `2021:661:4cc6:1:5561:96ff:ca11:d4a/128` represents the single IPv6 address `2021:661:4cc6:1:5561:96ff:ca11:d4a`.
- CIDR `2021:661:4cc6:1:5561:96ff:ca11:d00/120` represents a block size of `256` with First IP `2021:661:4cc6:1:5561:96ff:ca11:d00` and a Last IP `2021:661:4cc6:1:5561:96ff:ca11:dff` .

13.4.2. CIDR Set

For SavaPage use only we define a CIDR Set as a concatenation of single CIDR notations separated by any of the characters *space*, *comma* or *semicolon*.

13.5. Advanced Configuration

The majority of SavaPage configuration is conducted in the Administrator Web App. Additional configuration options can be set in the `server.properties` and `log4j.properties` files, and in the Config Editor.

13.5.1. server.properties

Static server properties are configured in `/opt/savapage/server/data/server.properties` and read at server startup.

Tip

`/opt/savapage/server/examples/data/server.properties.template` is a copy of the default `server.properties` of your current installation. If your actual `server.properties` is missing key/value pairs that are required for advanced configuration, you can copy them from the corresponding `.template` file.

Note

Instead of using `server.properties` to configure SavaPage, the same keys from this file can also be used as enviroment variables. See Section 13.5.15, “System Environment” [257].

[2] https://tools.ietf.org/html/rfc4632
[3] CIDR notation for IPv6 is described in RFC 4291 [https://tools.ietf.org/html/rfc4291#section-2.3].

13.5.2. log4j.properties

Static logging properties are configured in `/opt/savapage/server/data/log4j.properties` and read at server startup.

> **Note**
>
> `/opt/savapage/server/examples/data/log4j.properties.template` is a copy of the default `log4j.properties` of your current installation. If your actual `log4j.properties` is missing key/value pairs that are required for advanced configuration, you can copy them from the corresponding `.template` file.

13.5.3. Alternative TCP/IP Settings

13.5.3.1. Alternative TCP/IP Interface and Ports

Alternative port are set in the `server.properties` file as shown in the table below.

Key	Description
`server.host`	Hostname or IPv4/6 address of the network interface the server binds to. Default: bind to all interfaces.
`server.port`	Server http port. Default: `8631`
`server.ssl.port`	Main https port. Connected to an externally supplied SSL certificate, if installed. If *not* installed, the port is connected to the internal self-signed SSL certificate for local use. Default: `8632`
`server.ssl.port.local`	Extra https port for local use. Connected to the internal self-signed SSL certificate Default: `8633` *This port is relevant if an external SSL certificate is installed and `server.host` is not specified.*
`server.html.redirect.ssl`	Redirect HTML of non-SSL port to SSL: `true` or `false` (default). This concerns HTML only, IPP traffic is not redirected.
`server.print.port.raw`	The RAW Print Server port. Default: `9100` • Value `0`: RAW printing is disabled.

Table 13.5. Alternative TCP/IP Ports : server.properties

> **Important**
>
> Since SavaPage is run by user `savapage`, you can't use ports below 1024, because these ports can only be bound to by the superuser (root). If you want SavaPage Web Apps to be accessible through port 80 and 443 at all costs, you can use Apache[4] or NGINX[5] server to forward (or "reverse-proxy") requests to the IP address of the SavaPage host and apply the `server.host` key to bind the server to this single IP address.

13.5.3.2. Server ThreadPool Settings

A high reliability server process like SavaPage must reject excess requests *immediately* (fail fast) by using a Request Queue with a *bounded capacity*.

[4] https://httpd.apache.org/docs/2.4/howto/reverse_proxy.html
[5] https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/

When a request is rejected, the user browser will show a diagnostic message, for example that “The connection to the server was reset while the page was loading” because “The site could be temporarily unavailable or too busy”, with the advise to “Try again in a few moments”.

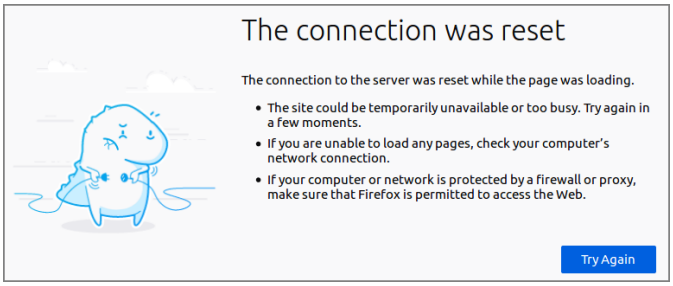


Figure 13.1. Rejected request in Firefox browser

Request Queue capacity is calculated according to a tolerable “no-response” time.

For example, if the server is capable of handling 100 requests per second, and 30 seconds of patience is accepted by users in the event of excessive high load, you can set the queue capacity to `30*100=3000`.

If queue capacity is set too low, the server will reject requests too soon and won't be able to handle normal load spikes. If set too high, a high load, that exceeds the processing power of the server application, will continue to stack requests on the queue. And thus, even after the load stops, the application will appear to have stopped responding to new requests as it still has lots of requests on the queue to handle.

Requests are FIFO processed by threads contained in a ThreadPool. The maximum number of threads needed, in order to achieve the best performance, depends on host resources (RAM and CPU cores) assigned to the SavaPage application. The maximum value will typically be between `50` and `500`.

The keys to set thread pool and queue capacity in the `server.properties` file are shown in the table below.

Key	Description
`server.threadpool.queue.capacity`	The capacity (maximum length) of the queue holding client requests to the server. Requests are FIFO processed by threads in the pool. Default: `3000`
`server.threadpool.maxthreads`	Maximum number of threads in the pool. Default: `200`
`server.threadpool.minthreads`	Minimum number of threads in the pool. Default: `20`
`server.threadpool.idle-timeout-msec`	Maximum time a thread may be idle in milliseconds after it is terminated. Default: `30000`

Table 13.6. ThreadPool Settings: server.properties

ThreadPool settings are part of application performance tuning. See Chapter 22, *Tuning* [314] in general and Section 22.1, “Linux Kernel Parameters” [314] in particular.

Note

The browser “connection reset” message can be easily provoked by setting very low threadpool values, like:

```
server.threadpool.queue.capacity=4
server.threadpool.minthreads=1
server.threadpool.maxthreads=5
server.threadpool.idle-timeout-msec=10000
```

... and frantically pressing the F5 browser key.

13.5.3.3. Server Session Settings

Server Web Sessions expire after a certain period of inactivity: see Section 17.4.1, "Web Session Timeout" [289]. Expired sessions are removed by a cyclic scavenge process. The cycle interval can be set with this property:

Key	Description
`server.session.scavenge.interval-sec`	The interval (seconds) at which expired server sessions are scavenged. Default: `600`

Table 13.7. Session Settings: server.properties

13.5.4. HTTP/2 Settings

SavaPage supports both HTTP/1.1 and HTTP/2[6] communication protocols. HTTP/2 can be disabled or customized as desired. In the `server.properties` configuration file ...

Key	Description
`server.http2`	Enable HTTP/2: `true` (default) or `false`.
`server.http2.max-requests-per-sec`	Maximum number of requests from an HTTP/2 connection per second. Requests in excess of this will encounter an ERR_CONNECTION_CLOSED. Default: `128`. Also see Section 17.1.1, "Denial of Service Filter" [284].

Table 13.8. HTTP/2 Settings: server.properties

... and in the Config Editor.

Configuration property	Description
system.http2.max-requests.warn-interval.applog.mins	Interval (minutes) within which HTTP/2 requests that exceed the rate control max are totaled and written as warning in the Application Log. The expiration of the interval is triggered by an HTTP/2 event that exceeds the limit. Default: `5`.
system.http2.max-requests.warn-interval.webapp.secs	Interval (seconds) within which HTTP/2 requests that exceed the rate control max are totaled and send as Real-time Activity warning with popup message to the Admin Web App. Default: `60`.

Table 13.9. HTTP/2 Configuration Properties

13.5.5. HTTP XFF Header Settings

SavaPage uses the Remote IP address from an HTTP Request as Client IP address. This usually works out well. However, when client requests go through an HTTP Proxy or Load Balancer, the Remote IP address will be the IP address of the proxy or load balancer only. In that case the Remote IP address can't be used to define Devices or to restrict access to SavaPage objects, like Queues, Web Services, etc. To solve this issue you can set the properties below to use the HTPP X-Forwarded-For[7] header to retrieve the correct Client IP address.

Configuration property	Description
webserver.http.header.xff.enable	Use the X-Forwarded-For (XFF) HTTP header to retrieve Client IP address. Values: `N` (default) | `Y`. If `Y`, the HTPP "X-Forwarded-For" header is used to retrieve the Client IP address.

[6] https://en.wikipedia.org/wiki/HTTP/2
[7] https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Forwarded-For

Configuration property	Description
webserver.http.header.xff.proxies.allowed	A CIDR Set of allowed XFF proxies. If empty all proxies are allowed. Default: empty.
webserver.http.header.xff.debug	Enable X-Forwarded-For (XFF) HTTP header debugging. Values: `N` (default) | `Y`. If `Y`, INFO messages are send to the Admin Dashboard for each retrieved Client IP address.

Table 13.10. HTTP Configuration Properties

See Section 4.11.15, "Config Editor" [189] on how to set these properties.

13.5.6. Database Connection Settings

Database connections are both expensive to create and maintain over time. Therefore, they are an ideal resource to pool. That is just what SavaPage does, with the help of JDBC Connection Pooling[8].

The keys to set the Connection parameters in the `server.properties` file are shown in the table below.

Key	Description
`database.connection.pool.max`	Maximum number of connections in the pool. Default: `200` **Important**: if you are using PostgreSQL as database back-end this value must align with the maximum number of client connections allowed by the database. See Section 13.5.6.1, "PostgreSQL Settings" [252].
`database.connection.pool.min`	Minimum number of connections in the pool. Default: `5`
`database.connection.idle-timeout-secs`	Maximum time a connection may be idle in *seconds* after it is closed. Default: `600`
`database.connection.idle-timeout-test-secs`	Idle time in *seconds* before a connection is checked for timeout. Default: `120` This value must be **less** than `database.connection.idle-timeout-secs`. If not, connections closed by the database will not be detected.
`database.connection.statement-cache`	Number of prepared SQL statements that will be cached. Default: `50` Value `0` disables caching.

Table 13.11. Database Connection: server.properties

Note

Database Connection Settings do *not* apply to the internal database.

13.5.6.1. PostgreSQL Settings

PostgreSQL ships with a basic configuration tuned for wide compatibility rather than performance. Depending on available host resources and Database Connection Settings as described in the section above, there is a good chance the default parameters are very undersized.

[8] SavaPage uses Java Database Connectivity (JDBC) Connection Pooling from c3p0 [https://www.mchange.com/projects/c3p0/]: an easy-to-use library for making traditional JDBC drivers "enterprise-ready".

Tuning Your PostgreSQL Server[9] is an expert job, but there are two obvious parameters in `postgresql.conf`[10] you need to consider:

1. `max_connections` **must** at least be equal to `database.connection.pool.max` plus the value of `superuser_reserved_connections`. For example:

   ```
   #------------------------------------------------------------------
   # If SavaPage: database.connection.pool.max = 200
   # and in this file: superuser_reserved_connections = 3
   # and SavaPage is the only process accessing the Database ...
   #------------------------------------------------------------------
   max_connections = 203    # (change requires restart)

   # Defaults to 3
   #superuser_reserved_connections = 3
   ```

   See PostgreSQL Connection Settings[11].

2. The `shared_buffers` default of 128MB can be replaced by a higher value, depending on the total RAM in your system. For example:

   ```
   #------------------------------------------------------------------
   # System has 16GB RAM, we take a conservative 6.25% ...
   #------------------------------------------------------------------
   #shared_buffers = 128MB  # min 128kB
                            # (change requires restart)
   shared_buffers = 1024MB
   ```

   A higher value allocates more shared memory during inter-process communication (IPC) between the database server and the requesting client, resulting in speedier data transfer. See PostgreSQL Memory[12].

Important: *restart the database after changing any of these settings.*

13.5.7. CUPS Settings

13.5.7.1. CUPS Notifier

CUPS notifier is a custom SavaPage binary that pushes printer and print job status events from CUPS to SavaPage server. It is the default CUPS notification method. A pull method can be activated as alternative.

Occasionally the notifier may fail to deliver job state notifications, as can be seen reported in `/var/log/cups/error_log`. Since a failure can result in no job end states being observed, a heartbeat monitor pulls print status from CUPS for active jobs that have not been notified since the previous beat of the heart. The heartbeat period can be configured: see Section 13.5.7.4, “CUPS Configuration Properties” [254].

13.5.7.2. CUPS Job Status

CUPS Job Status is handled in the following way:

- Normally a print job that is stopped in CUPS, for instance because “job-completed-with-errors”, must be manually controlled (reprint, cancel, move) by an operator. However, when SavaPage identifies a stopped job it cancels it automatically.
- At system (application) start-up, job status is pulled from CUPS for every proxy printed document that has not reached end-state.

[9] https://wiki.postgresql.org/wiki/Tuning_Your_PostgreSQL_Server
[10] On Debian based systems `postgresql.conf` is located in `/etc/postgresql/[version]/main/`
[11] https://www.postgresql.org/docs/current/runtime-config-connection.html
[12] https://www.postgresql.org/docs/current/runtime-config-resource.html

See Section 13.5.7.4, “CUPS Configuration Properties” [254] on how to change the defaults.

13.5.7.3. CUPS IPP connections

See Section 13.5.7.4, “CUPS Configuration Properties” [254] on how to change defaults for the number of IPP connections to local CUPS and their IPP connect and read timeouts.

13.5.7.4. CUPS Configuration Properties

Configuration property	Description
cups.ipp.notification.method	CUPS event notification method. Value `PUSH` (default) activates *pushed* print job status notification by CUPS Notifier. If value is `PULL`, print job status is *pulled* from CUPS.
cups.ipp.notification.push.notify-lease-duration	Duration of the subscription lease (seconds) for `PUSH` notification with the CUPS Notifier. Value must be greater than one (1) hour, since the renewal of the subscription is scheduled by SavaPage every hour. Default: 4200. See Hourly Jobs.
cups.ipp.notification.push.heartbeat-msec	Number of milliseconds to wait before evaluating *pushed* CUPS job status notifications. Default: 4000.
cups.ipp.notification.push.pull-fallback-msec	Number of milliseconds since the last *pushed* print job status notification by CUPS Notifier after which a job status update is *pulled* from CUPS. Default: 30000.
cups.ipp.notification.pull.heartbeat-msec	Number of milliseconds to wait before *pulling* CUPS job status. Default: 5000.
cups.ipp.job-state.cancel-if-stopped.enable	Set to Y (default) or N, to enable/disable automatic cancellation of stopped CUPS Job Status.
system.startup.cups.ipp.sync-print-jobs.enable	Set to Y (default) or N, to enable/disable CUPS Job Status synchronization at system start-up.
cups.ipp.local-connect-timeout-msec	IPP connect timeout in milliseconds on local CUPS. Default: 5000.
cups.ipp.local-socket-timeout-msec	IPP read timeout in milliseconds on local CUPS. Default: 9000.
cups.ipp.max-connections	Max number of IPP connections on CUPS. Default: 10.
cups.ipp.exception.stacktrace	Set to Y (default) or N to enable/disable writing a call stack trace to the `server.log` when an exception is thrown that breaks the IPP CUPS connection. See Section 13.2, “Logs” [246].

Table 13.12. CUPS Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these properties.

13.5.8. Server-side Download

A Web Page or Document can be printed by a Drag & Drop of its URL. In that case its content is server-side downloaded and rendered as PDF.

Configuration property	Description
download.connect-timeout-msec	Timeout in milliseconds until a connection is established. Default: 4000.
download.socket-timeout-msec	Timeout in milliseconds to receive data from remote host after the connection is established, i.e. maximum time of inactivity between two data packets. Default: 2000.

Configuration property	Description
download.max-connections	Max number of download connections. Default: 10.
download.max-connections-per-route	Max number of download connections per route (host) Default: 2.

Table 13.13. Server-side Download Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these properties.

> **Note**
>
> Rendering a Web Page by a Drag & Drop of its URL might not give accurate results.

13.5.9. RESTful Client

When SavaPage makes RESTful API calls to third-party servers, for instance in the IPP Routing Plug-in, these are the configuration properties to consider.

Configuration property	Description
restful.client.connect-timeout-msec	Timeout in milliseconds until a connection is established. Default: 4000.
restful.client.read-timeout-msec	Timeout in milliseconds to receive data from remote host after the connection is established, i.e. maximum time of inactivity between two data packets. Default: 2000.
restful.client.max-connections	Max number of RESTful client. Default: 100.
restful.client.max-connections-per-route	Max number of RESTful client connections per route (host) Default: 20.
restful.client.trust-self-signed	Trust self-signed certificate of RESTful servers: N (default) or Y.

Table 13.14. RESTful Client Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these properties.

13.5.10. User Home Clean

When Document expiration time is enabled, any SafePages document older than the number of configured minutes is considered expired and will be automatically deleted in an active User Web App session. Moreover, when a user logs out and logs in again, expired documents will be auto-deleted. So, when a user does not log in for a long time, old documents will stay put in the user's home directory. These ignored documents take up unnecessary space. The same applies to PDF documents that exist long after their due date, because their deletion is triggered far too late (never) at Fast Print or Hold Print release.

During Daily Maintenance user home directories are scanned and ignored documents deleted. The result of the scan is shown in the Admin Dashboard. These are the configuration properties to consider:

Configuration property	Description
print-in.job-expiry-ignored.mins	Number of minutes added to print-in job expiry (if applicable) after which job is considered ignored. Ignored jobs are removed by Daily Maintenance. Default: 10.
proxy-print.hold-ignored-mins	Number of minutes added to Hold Print job expiry after which job is considered ignored. Ignored jobs are removed by Daily Maintenance. Default: 10.

Table 13.15. User Home Clean Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these properties.

> **Note**
>
> User Home Clean is also executed at server start-up. You can disable this behavior in a separate Server Setting.

13.5.11. Database Locking

When using an External Database, table row locking is used to serialize access to individual user resources. Production grade databases like PostgreSQL should be able to handle row locking without problems. However, if you do suspect problems in this area, please contact SavaPage support. In the meantime, you can disable row locking with the property below.

Configuration property	Description
webapp.user.database-user-row-locking.enabled	Enable user table row locking in database to serialize access of a user to database or user home file system: Y (default) | N.

Table 13.16. Database Locking Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these properties.

13.5.12. pdf2cairo Image Strategy

SavaPage uses **pdftocairo** to create PNG images from SafePages. On centos-release-7-7.1908.0.el7, pdftocairo 0.26.5 fails due to a bug. Therefore **pdftocairo -v** is executed at server start-up to determine the best way to create images. If version is 0.26.5 is encountered workaround statements are applied. This default probe strategy can be overwritten with this configuration property:

Configuration property	Description
system.host.cmd.pdftocairo.img.strategy	Strategy for executing `pdftocairo` command to create a PNG image. Values: `AUTO` = use version probe to determine optimal command (default), `STREAM` = use one (1) stdout/stdin stream command, `SPLIT` = split action into separate commands.

Table 13.17. pdf2cairo Image Strategy Configuration Property

See Section 4.11.15, “Config Editor” [189] on how to set these properties.

13.5.13. Alternative File Locations

To separate files created at runtime from installation files you can set alternative locations for temporary files, SafePages, public Letterheads and Document Store in the `server.properties` file.

> **Caution**
>
> All alternative file locations must reside on the same disk partition[13].

Key	Description
`app.dir.tmp`	Location of temporary files created by SavaPage. It is created when the application starts and removed when stopped, so make sure it is exclusively used by the SavaPage application.

[13] This constraint is needed because files are initially created in the temporary location and *atomically* moved to their destination. Atomic moves do *not* work cross-partition.

Key	Description
	The location is not used by third-party Java components: they use `java.io.tmpdir` to store their own temporary files. Default: `/opt/savapage/server/data/.tmp` • User `savapage` must have permission to create the location.
`app.dir.safepages`	Location where the user's SafePages are stored. Default: `/opt/savapage/server/data/internal/safepages` • The location directory must be owned by user `savapage` and have permission `700`.
`app.dir.letterheads`	Location where the public Letterheads are stored. Default: `/opt/savapage/server/data/internal/letterheads` • The location directory must be owned by user `savapage` and have permission `700`.
`app.dir.doc.store.archive`	Location where the *Archive branch* of the Document Store is stored. Default: `/opt/savapage/server/data/doc-archive` • User `savapage` must have permission to create the location.
`app.dir.doc.store.journal`	Location where the *Journal branch* of the Document Store is stored. Default: `/opt/savapage/server/data/doc-journal` • User `savapage` must have permission to create the location.

Table 13.18. Alternative File Locations: server.properties

13.5.14. Miscellaneous Settings

These are the extra settings that can be configured in the `server.properties` file.

Key	Description
`start.cleanup-doclog`	Enable database cleanup at server start-up. See Section 4.11.13, “Backups” [177]. Values: `true` (default) or `false`
`start.cleanup-userhome`	Enable User Home Clean at server start-up. Values: `true` (default) or `false`
`system.cleanup-userhome.test`	Put User Home Clean in test mode: i.e. cleanable files are signaled on the Admin Dashboard but not deleted. Values: `false` (default) or `true`
`webapp.custom.i18n`	See Section 20.1.1.4, “Custom i18n” [303]. Values: `true` or `false` (default).

Table 13.19. Miscellaneous Settings: server.properties

13.5.15. System Environment

Instead of using `server.properties` to configure SavaPage, the same keys from the properties file can also be used as enviroment variables, as follows:

```
# Namespace as prefix for SavaPage envvar names (mandatory)
SAVAPAGE_NS=SP_
```

```
# Required if run in Docker container
SP_CONTAINER=DOCKER

# Unique SP_SRV_* variables with server.properties key/value
# separated by ':' like SP_SRV_n=key:value

# For example:
SP_SRV_1=visitor.organization:Acme Corporation
SP_SRV_2=server.port:8080
SP_SRV_3=server.ssl.port:8443

# etc ...
```

Important

Environment variables take precedence over those configured in `server.properties`

The environment is shown in the Admin Web App. See Section 4.14.9, “Environment” [203] and Section 4.3.1, “Status” [92].

13.6. Upgrading SavaPage

Upgrading SavaPage is just an install of the new version “over-the-top” of the old version. It follows the same procedure as a first time installation, as described in Section 13.1, “The Installation Process” [244].

The process keeps existing data and configuration files as they are.

Important

Always check the Release Notes[14] after upgrading, to see if additional actions are needed before you can actually use the new version.

13.7. Removing SavaPage from a GNU/Linux server

SavaPage can be completely removed from a system with the following procedure:

- Remove all files from the `/opt/savapage` install directory.
- Remove the `savapage` system account.
- Remove the `savapage` binary from the CUPS `notifier` directory.
- For **systemd** installations:
  - Remove the `savapage.service` file in the `systemd` unit library. Depending on the distribution the unit will either be found in the `/lib/systemd/system` or `/usr/lib/systemd/system` directory.
  - Remove the custom `/etc/systemd/system/cups.services` file.
- For **SysV** style installations remove any matching start script:

```
/etc/init.d/savapage
/etc/rc*.d/*savapage
```

Note

Removing SavaPage can also be done by executing the `uninstall` program like this:

[14] https://wiki.savapage.org

```
cd /opt/savapage
sudo ./uninstall
```

The installation will be reverted including the CUPS `notifier` installation and the server's `systemd` or `SysV` service scripts.

As a final action the `savapage` system account and the `/opt/savapage` install directory should be removed manually.

```
sudo rm -rf /opt/savapage/
sudo userdel savapage
```

Chapter 14. SavaPage as Printer

14.1. IPP Everywhere and AirPrint

IPP Everywhere™[1] is a PWG[2] standard that allows personal computers and mobile devices to find and print to networked (and USB) printers without using vendor-specific software (printer drivers). It is based on IPP/2.0 and Zero-configuration networking[3] (Zeroconf) for automatic printer discovery on Internet Protocol (IP) networks. IPP Everywhere printers accept PWG Raster Format and JPEG (required) and PDF (recommended) print content.

IPP Everywhere™	• The terms "Zeroconf" and "Bonjour" are often used interchangeably, although Zeroconf refers to the technology in general and Bonjour is the Apple branded implementation. • In analogy, the terms "IPP Everywhere" (Zeroconf printers) and "AirPrint" (Bonjour printers) are also used interchangeably. • *To advertise SavaPage as Zeroconf printer follow the steps described in the sections below.*
SavaPage AirPrint is discussed for each platform separately: • Section 14.3, "Printing from GNU/Linux" [263] • Section 14.4, "Printing from macOS" [264] • Section 14.5, "Printing from Chrome OS" [264] • Section 14.6, "Printing from Android" [264] • Section 14.7, "Printing from iOS and iPadOS" [267] • Section 14.8, "Printing from Windows" [267]	

Table 14.1. IPP Everywhere and AirPrint

The IPP Printer attributes of SavaPage can be checked with the **ipptool** command as follows:

```
# install ipptool
sudo apt install cups-ipp-utils

# get printer attributes
ipptool -tv ipps://savapage.local:8443/printers/airprint get-printer-attributes.test
```

Note

The Default Paper Size is used as `media-default`.

14.1.1. Step 1: Check Avahi configuration

To advertise SavaPage as Zeroconf printer, Avahi needs to be installed on your GNU/Linux host.

To service a broad range of devices, both IPv4 and IPv6 should be enabled in the avahi-daemon. This is normally the case, but you can check by editing the Avahi configuration file:

```
sudo vi /etc/avahi/avahi-daemon.conf
```

[1] https://www.pwg.org/ipp/everywhere.html
[2] https://www.pwg.org/
[3] https://en.wikipedia.org/wiki/Zero-configuration_networking

Make sure settings are as follows:

```
use-ipv4=yes
use-ipv6=yes
```

If you made changes to the configuration file, restart the daemon as follows:

```
sudo systemctl restart avahi-daemon.service
```

14.1.2. Step 2: Create AirPrint Queue

The URL path of Reserved Queue called `/airprint` is available by default.

You can create a new SavaPage Queue (see Section 4.7, "Queues" [120]) with another path if you like. It is important though to select Standard or Basic for Authentication, since the queue must be *untrusted* to enforce either IP Based Authentication or Basic Authentication. This is needed because AirPrint is inherently *unauthenticated*.

Important

IP addresses from private networks are allowed by default. If you need to allow addresses from public networks, you must override the default setting by adding one or more CIDR sets containing allowed IP ranges (including those from private networks). See Section 4.7.2, "Edit Queue" [123].

14.1.3. Step 3: Create Avahi Service File

Copy the `/opt/savapage/server/examples/linux/avahi/savapage.service` file to your personal home directory.

`savapage.service` is an Avahi service file with annotations explaining how to adapt it to your own situation. Follow the `$Customize$` annotation to insert your settings.

Copy your tailored service file to Avahi, with this command (assuming the source file resides in your home directory):

```
sudo cp ~/savapage.service /etc/avahi/services
```

Check if Avahi has published the SavaPage printer as intended by typing this command:

```
avahi-browse -a -t
```

Assuming your GNU/Linux host is called `savapage` and you named your Avahi print service `SavaPage`, you should find entries in the output like this :

```
+   eth1 IPv6 SavaPage on savapage     _ipps._tcp           local
+   eth1 IPv4 SavaPage on savapage     _ipps._tcp           local
+   eth1 IPv6 SavaPage on savapage     Internet Printer     local
+   eth1 IPv4 SavaPage on savapage     Internet Printer     local
```

To check if the SavaPage DNS printer can be found, use this command on a macOS or GNU/Linux workstation :

```
ippfind _ipp._tcp _ipps._tcp
```
[4]

Assuming your GNU/Linux server host is called `savapage` in the Avahi service file, you should see these entries like this as output:

[4] `ippfind` is part of `cups-ipp-utils` package.

```
ipp://savapage.local:8631/printers/airprint
ipps://savapage.local:8632/printers/airprint
```

14.2. PostScript Driver Print

Any desktop system can print to SavaPage with a PostScript driver. The driver can either be generic, or a mainstream one from a vendor as shipped with the OS, or a dedicated one provided by SavaPage. When printing from public Internet a private Device URI must be used. See Section 3.10.1, “Internet Printer” [79].

Caution

Although the SavaPage driver is not required, beware that vendor-specific drivers might offer options that are irrelevant, or not supported by the SavaPage Print Server.

The SavaPage Printer Driver comes as a PostScript Printer Description (PPD), as captured in the `SAVAPAGE.ppd` file located in the shared client directory `/opt/savapage/client`. The driver is optimized for SavaPage printing. Irrelevant options, like *Duplex Printing* are stripped, other options like *Paper Size* and *Resolution*, are narrowed down to the most common choices. If you feel options are missing please let us know.

Important

`SAVAPAGE.ppd` is a PostScript-only driver, and does *not* produce PDF. This is on purpose, since PDF producing drivers print landscape orientated documents with 270 page rotation. As a result these documents will show as portrait orientated SafePages. [5]

The driver file can be downloaded from the About section of the User Web App and Admin Web App.

The installation scripts below use the SavaPage printer driver. When you want to use a PostScript driver already present in the OS, please use the proper selection dialogs.

Caution

The SavaPage JetDirect Server accepts PostScript print jobs only. So do not use the JetDirect protocol unless you are absolutely sure that the print client uses PostScript as Print Job Format. Windows clients can safely use JetDirect. On macOS and GNU/Linux systems IPP and IPPS are the obvious choices.

14.2.1. IPP Options from PostScript

Preview

IPP Options from PostScript is a Feature Preview that is fully operational but impermanent. The preview is offered to provoke feedback based on real world use. Please tell us your requirements.

[5] SavaPage acts as a Virtual PostScript Printer Device. This is separate from the decision on the OSDL Printing Summit in 2006 to switch the GNU/Linux standard print job transfer format from PostScript to PDF. See `https://wiki.linuxfoundation.org/openprinting/pdf_as_standard_print_job_format`

During PostScript Driver Print the PostScript stream is scanned by SavaPage for elements that can be mapped to IPP options.

PostScript element	value	IPP option	value
%RBINumCopies: %%BeginNonPPDFeature: NumCopies		copies	(number of copies)
%%BeginFeature: *Duplex	None DuplexNoTumble DuplexTumble	sides	one-sided two-sided-long-edge two-sided-short-edge
%%BeginFeature: *Collate	True False	sheet-collate	collated uncollated
featurebegin{<< /Collate	true false	sheet-collate	collated uncollated
<</ProcessColorModel	/DeviceCMYK /DeviceGray	print-color-mode	color monochrome
featurebegin{<< /ProcessColorModel	/DeviceCMYK /DeviceGray	print-color-mode	color monochrome
@PJL SET LCOLORMODEL=	CMYK BLACK	print-color-mode	color monochrome

Table 14.2. PostScript elements mapped to IPP options

Note

If Inherit IPP options from print-in document is enabled extracted IPP options are used for Fast Print Mode.

14.3. Printing from GNU/Linux

GNU/Linux supports native printing through the Common UNIX Printing System (CUPS) and Internet Printing Protocol (IPP).

IPP Everywhere™	If SavaPage IPP Everywhere is configured in a Local Network, the corresponding printer is automatically detected by CUPS 2.2.2 (or later) and can directly be used without any configuration. In older CUPS versions the SavaPage IPP Everywhere printer is recognized as “Network Printer”, but you must choose a driver either from the database (Generic > Generic PostScript Printer Foomatic) or provide a PPD file, i.e. `SAVAPAGE.ppd`. • IP Based Authentication or Basic Authentication is needed before submitting a print job.
UNIX PRINTING SYSTEM	You can add any SavaPage Local IPP Printer according to the SavaPage Printer URI syntax [363]. For example: • Choose a driver either from the database (Generic > Generic PostScript Printer Foomatic) or provide a PPD file, i.e. `SAVAPAGE.ppd`. • Enter `ipps://savapage:8632/printers` at **Device URI** for the default queue, or `ipps://savapage:8632/printers/[queue]` for any other specific queue. Notes: • IP Based Authentication or Basic Authentication might be needed before submitting a print job.

	• For Internet Printer settings, see Section 3.10.1, “Internet Printer” [79] (no additional authentication is needed).

Table 14.3. Printing from GNU/Linux

14.4. Printing from macOS

MacOS supports native printing through the Common UNIX Printing System (CUPS) and Internet Printing Protocol (IPP).

IPP Everywhere™	If SavaPage IPP Everywhere is configured in a Local Network, the corresponding printer is automatically detected as `Default` in the “Printers & Scanners > Add” dialog and can directly be used without any configuration. • IP Based Authentication or Basic Authentication is needed before submitting a print job.
UNIX PRINTING SYSTEM	You can add any SavaPage Local IPP Printer according to the SavaPage Printer URI syntax [363]. Add a new printer and enter data in the “Printers & Scanners > Add” dialog as follows: • Click the `IP` printer button and select IPP for **Protocol**. • At **Address**, enter the IP address or host name of the SavaPage Print Server *including* the port number. • Enter `printers` at **Queue** for the default queue, or `printers/[queue]` for any other specific queue. See SavaPage URI syntax [363]. • Enter the **Name** of the queue. SavaPage is the obvious choice here. • **Use** the default “Generic PostScript Printer”, or use “Other” to select the `SAVAPAGE.ppd` you downloaded earlier. Notes: • IP Based Authentication or Basic Authentication might be needed before submitting a print job. • For Internet Printer settings, see Section 3.10.1, “Internet Printer” [79] (no additional authentication is needed).

Table 14.4. Printing from macOS

14.5. Printing from Chrome OS

Chrome OS supports native printing through the Common UNIX Printing System (CUPS) and Internet Printing Protocol (IPP).

IPP Everywhere™	If SavaPage IPP Everywhere is configured in a Local Network, the corresponding printer is automatically detected and can directly be used without any configuration. • IP Based Authentication is needed before submitting a print job.
UNIX PRINTING SYSTEM	You can add any SavaPage Local IPP Printer according to the SavaPage Printer URI syntax [363] using Manufacturer “Generic” and Model “Generic PostScript Printer”. • IP Based Authentication is needed before submitting a print job. For Internet Printer settings, see Section 3.10.1, “Internet Printer” [79] (no additional authentication is needed).

Table 14.5. Printing from Chrome OS

14.6. Printing from Android

IPP Everywhere™	If SavaPage IPP Everywhere is configured in a Local Network, the corresponding printer is automatically detected and can directly be used without any configuration. • IP Based Authentication is needed before submitting a print job.

	SavaPage Android Print App is a native Android printing service. Users can copy the URL and their credentials from the Internet Printer dialog. The app makes use of Section C.2.3.3, “RESTful User Services” [350].

Table 14.6. Printing from Android

14.6.1. SavaPage Android Print App

ANDROID GET IT ON F-DROID [6]

Preview

SavaPage Android Print App[7] is a Feature Preview that is fully operational but impermanent. The preview is offered to provoke feedback based on real world use. Please tell us your requirements.

Users can copy the URL and their credentials from the Internet Printer dialog.

The app makes use of Section C.2.3.3, “RESTful User Services” [350].

To illustrate the App, composite Android screen-shots below show three SavaPage Printers. A tap on a printer entry shows the details for editing.

Figure 14.1. SavaPage Android Print App - Printer List

Printers were added by selecting Add Printer from the top-right menu or by pressing the bottom-right + button. Global Settings were used to trust a self-signed SSL certificate of any SavaPage Printer URL.

[6] https://f-droid.org/packages/org.savapage.android.print/
[7] https://f-droid.org/packages/org.savapage.android.print/

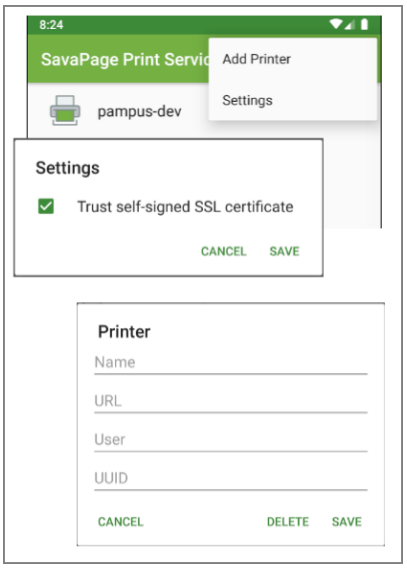


Figure 14.2. SavaPage Android Print App - Menu

Important

After the SavaPage App is enabled in the “Print services” section of the Settings App, Android Apps that offer Print Services will see the SavaPage printers as target.

Figure 14.3. SavaPage Android Print Service

14.7. Printing from iOS and iPadOS

IPP Everywhere™	If SavaPage IPP Everywhere is configured in a Local Network, the corresponding AirPrint printer is automatically detected by and can directly be used without any configuration. • 64-bit iOS (iPadOS) is supported, 32-bit is **not**. • IP Based Authentication or Basic Authentication is needed before submitting a print job.
	For your convenience `https://savapage:8632/ios/install` is available to add a SavaPage icon to the Home Screen automatically. See Appendix E, *URL Cheat Sheet* [362]. A click on the icon opens the SavaPage User Web App full-screen and will therefore be part of the multitasking bar. This convenience comes with a penalty though, since Apple treats full-screen WebApps in a “special” way, i.e. when they are selected from the multitasking bar and regain focus the Web App is reloaded. Luckily, since SavaPage utilizes an authentication token, an automatic login is performed. However, the page needs to be retrieved from the server again, giving some performance penalty. **Note**: Only one SavaPage Web Clip can be present on an iOS device. A new install overwrites the previous one. **Warning**: When using the Payment Gateway Plug-in, the redirect URL as forwarded to and applied by the payment provider does not show in the same User WebApp as where the payment started, but is shown on a new tab in the default browser. **Tip**: If you don't care about the full-screen User Web App, and want optimal performance, you can just add any SavaPage Web App to the Home Screen manually by surfing to the URL, then click the Share button and choose Add to Home Screen. Clicking the home screen icon will not open the Web App full-screen, but as a tabbed instance in the browser. Also, it will not be reloaded by definition as the browser regains focus.

Table 14.7. Printing from iOS and iPadOS

14.8. Printing from Windows

IPP Everywhere™	If SavaPage IPP Everywhere is configured in a Local Network, the corresponding printer is automatically detected by Windows and can be used without any configuration by executing “Add a printer or scanner”. As a response Windows will be “Searching for Printers and Scanners”: pick the detected SavaPage printer and select “Add device". • Make sure Windows IPP printing is enabled as explained in Section 14.8.1, “IPP Printer” [267]. • IP Based Authentication or Basic Authentication is needed before submitting a print job.
	Installing a IPP Printer or Raw Printer is similar for different Windows versions with slight variations in dialog appearance. When asked for a *printer driver*, pick a plain PostScript driver from the list. Any type/model will do, as long as it generates standard PostScript spool files. It makes sense to select just a simple, no-frills type/model. **Caution**: Although a printer driver may render printed documents correctly on first sight, its SafePages PDF might not be *searchable*. You can simply Download the PDF result and check with a simple copy/paste of text. When the pasted result is garbled, the printer driver delivered so-called “Mojibake” [a]

[a] Mojibake [https://en.wikipedia.org/wiki/Mojibake] is the garbled text that is the result of text being decoded using an unintended character encoding. The result is a systematic replacement of symbols with completely unrelated ones, often from a different writing system.

Table 14.8. Printing from Windows

14.8.1. IPP Printer

Before adding a SavaPage IPP Printer make sure IPP printing is enabled:

1. Go to Control Panel
2. Select Programs and Features
3. Select Turn Windows features on or off
4. In the features list, scroll down to Print and Document Services and verify that the Internet Printing Client check-box is checked. If not, check this option and click OK.

To add SavaPage as IPP Printer, start the "Add a printer or scanner" dialog, choose Add a new device manually and follow these steps:

1. Find a printer by other options
   - Select Add a printer using an IP address or hostname.
   - Click Next.
2. Type a printer hostname or IP address
   - Select IPP Device as **Device type**.
   - At **Hostname or IP address**, enter the URL of a SavaPage IPP Queue. For example:
     - `https://my-savapage:8632/printers/my-queue`
     - `http://my-savapage:8631/printers` (default IPP "`/`" queue)
   - Click Next
3. Click Print a test page
   - Choose a plain PostScript driver from the list.
   - Click Finish

14.8.2. Raw Printer

To add SavaPage as Raw Printer, start the "Add a printer or scanner" dialog, choose Add a new device manually and follow these steps:

1. Find a printer by other options
   - Select Add a printer using an IP address or hostname.
   - Click Next.
2. Type a printer hostname or IP address
   - Device type : TCP/IP Device
   - Hostname or IP address : `my-savapage:9100`
   - Click Next
3. Detecting TCP/IP port ... (may take a while)
4. Additional port information required
   - Device Type : Custom
   - Click Settings . . .
5. Configure Standard TCP/IP port Monitor
   - Protocol: Raw
   - Port Number: 9100 (or the alternative port as configured in server.properties).
   - Click OK and Next
6. Install the printer driver
   - Choose a plain PostScript driver from the list.

- Click Next

7. Which version of the driver do you want to use
   - Use the driver that is currently installed
   - Click Next
8. Type a printer name
   - For instance “SavaPage” (it is possible to rename the printer after it is installed).
   - Click Next
9. Printer Sharing
   - If you want to share this printer, you must provide a share name.
   - Click Next
10. Print a test page to check if everything works as expected.
    - Print jobs are send to the Reserved Queue named “`/raw`” .
    - Click Finish

14.9. Driverless File Transfer Print

Driverless Printing by file transfer is one of the gateways to authenticated printing from devices belonging to unauthenticated anonymous users, where either native or IPP Everywhere printing support is lacking or where user trust cannot be enforced by a Single Sign-On User Domain. This situation is typical for a BYOD[8] context.

Web Print and Mail Print offer casual printing without a hassle, since users are familiar with either file uploading or email and no printer driver needs to be installed. Since rendering of the document content is not handled by a know-it-all printer driver, not all document types are supported. See Appendix G, *Printable File Types* [369] for a list of supported types. Driverless File Transfer Print is also implemented in the SavaPage Android Print App and RESTful Print API.

Note

Mail Print uses `127.0.0.1` (localhost) as the IP address of the requesting user.

14.9.1. Processing Driverless PDF Transfer

PDF files received from Driverless File Transfer Print are validated, repaired and processed before admitted as SafePages. These are the processing steps:

1. *Uploaded PDF document is read to check validity.*

   If the PDF is assessed as invalid by the PDF library used by SavaPage, a repair is executed with pdftocairo. If repair fails, the document is rejected.

   Note

   Although shown correctly as SafePages, PDF files might not have a completely valid PDF format. When these invalid PDFs are proxy printed, CUPS filtering may deliver Page Description Language (PDL) that incorrectly describes the PDF content. As a result, printed output will not be as expected. Text can be

[8] Bring Your Own Device (BYOD) is the policy of permitting employees to bring personally owned mobile devices (laptops, tablets, and smart phones) to their workplace, and use those devices to access privileged company information and applications. The term is also used to describe the same practice applied to students using personally owned devices in an educational setting.

omitted, and in some cases we observed only a single blank page was printed. To prevent this situation, SavaPage tries to repair these files, which is a modest action with minor performance impact. *Disclaimer: repair is a best-effort attempt and is no guarantee for success.*

2. *If PDF is encrypted, it is decrypted.* QPDF is used for this step.
3. *PDF is optimized* with the **ps2pdf** system command.
4. *PDF font syntax is verified*. If font errors are encountered, a repair is executed with pdftocairo, when not already executed in a previous step. When font errors persist, the PDF document is rejected.
5. *Non-embedded/non-standard PDF fonts are embedded with best-fit fonts available on the host system.*

> **Note**
>
> This is because font embedding performed by CUPS/Ghostscript might result in missing text as explained in Step 1.

6. *PDF is cleaned* by executing pdftocairo command (when not already executed in a previous step) and optionally a "Ghostscript prepress" command.

> **Note**
>
> This is because the **pdf2ps** filter as executed by CUPS/Ghostscript might result in missing text and annotations as explained in Step 1.

7. A final check is performed by reading the page properties of the PDF. If the check fails, a PDF optimization is performed as explained in Step 3.

You can enable/disable each step with the following configuration properties:

Configuration property	Description
print-in.pdf.optimize	Set to `Y` or `N` (default) to enable/disable optimization (cleaning) of PDF files with the **ps2pdf** system command in Step 3. See Optimizing PDFs with Ghostscript[9]. Note: Step 7 is always performed, regardless of the value of this configuration property.
print-in.pdf.invalid.repair	Set to `Y` (default) or `N` to enable/disable repair of invalid PDF files with pdftocairo. When N, invalid PDF files are rejected.
print-in.pdf.encrypted.allow	Set to `Y` (default) or `N` to enable/disable decryption of encrypted PDF files with QPDF. When N, encrypted PDF files are rejected. **Important**: this property also applies when Section 14.10.2, "Printing Encrypted PDF" [272] with a printer driver. Also see Section 4.11.14.9, "SafePages" [184].
print-in.pdf.fonts.verify	Set to `Y` (default) or `N` to enable/disable PDF font syntax validation. If `Y`, PDF files are checked with the **pdffonts** system command. When font syntax errors or warnings are encountered a repair is executed with pdftocairo (when not already executed in a previous step). If font errors persist, the PDF document is rejected if `print-in.pdf.fonts.verify.reject` is `Y`.
print-in.pdf.fonts.verify.reject	Set to `Y` (default) or `N` to enable/disable rejection of PDF files with font issues that cannot be repaired. If `N`, PDF files with font issues are accepted with an appropriate user/log message.
print-in.pdf.fonts.embed	Set to `Y` (default) or `N` to enable/disable embedding of non-embedded/non-standard PDF fonts by executing pdftocairo command (when not already executed in a previous step).

[9] https://ghostscript.com/blog/optimizing-pdfs.html

Configuration property	Description
print-in.pdf.clean	Set to `Y` (default) or `N` to enable/disable PDF cleaning by executing pdftocairo command (when not already executed in a previous step). This is a replacement for the deprecated "proxy-print.repair.enable" property. See Section 4.11.9, “Proxy Print” [168].
print-in.pdf.prepress	Set to `N` (default) or `Y` to disable/enable PDF cleaning by executing `gs -sDEVICE=pdfwrite -dPDFSETTINGS=/prepress` command.

Table 14.9. Driverless PDF Transfer Processing Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set this property.

Note

The pdftocairo and `gs` command embeds all fonts, even standard PDF fonts.

Tip

Driverless PDF Transfer statistics are shown on the Admin Web App Dashboard. The final result of a Driverless PDF Transfer Print on the display of fonts is shown in the SafePages Document Details popup dialog.

14.10. PostScript Print Processing

The PostScript that arrives at SavaPage by PostScript Driver Print or Driverless File Transfer Print is the first to be converted to PDF by the **ps2pdf** program from the Ghostscript suite.

Note

ps2pdf is a work-alike for nearly all the functionality (but not the user interface) of Adobe's Acrobat™ Distiller™ product: it converts PostScript files to Portable Document Format (PDF) files, and is implemented as a command script that invokes Ghostscript[10].

Tip

The final result of PostScript Print Processing on the display of fonts is shown in the SafePages Document Details popup dialog.

14.10.1. Licensed Fonts

Using a font requires a license, just like a software program. There are fonts that require a special (paid) license for embedding them into documents so these documents can be used outside the scope of the licensed computer or licensed document viewer.

Be careful when you print a document containing license restricted fonts to SavaPage, since font rendering might not be as expected. It all depends on how the print toolchain renders non-embeddable fonts to PostScript at the client side. Some toolchains will convert these fonts to bitmaps. Others, will embed them unconditionally, which is fine for local printing, but not for SavaPage because the **ps2pdf** converter might give a Warning message saying the font “cannot be

[10] https://www.ghostscript.com/doc/doc/Use.htm

embedded because of licensing restrictions". As a result font substitution might be flawed and SafePages preview will show garbled text. In those cases rendering PostScript pages to *images* is a solution. See Section 14.10.3, "PostScript Configuration Properties" [272].

> **Important**
>
> Windows computers are more prone to font licensing issues than GNU/Linux workstations where Open Source Fonts are the norm. When printing from Windows use Open Source Fonts wherever possible. Many of these fonts come with the increasingly popular Open Font License[11] (OFL) from SIL International[12], a nonprofit organization committed to serving language communities worldwide. The OFL is "a free, libre and open source license specifically designed for fonts and related software based on our experience in font design and linguistic software engineering."

14.10.2. Printing Encrypted PDF

When you print an encrypted PDF document from Adobe Reader to a PostScript printer like SavaPage, it creates a PostScript file that contains a notice telling the recipient that it is not permitted to convert (re-distill) it to PDF again. The Ghostscript **ps2pdf** program respects this notice, and throws an error saying "This PostScript file was created from an encrypted PDF file. Re-distilling encrypted PDF is not permitted". If, for example, an encrypted PDF allows printing only, it should not be re-distilled to a plain PDF equivalent, where all intended protection is removed.

SavaPage respects this policy. Moreover, on behalf of its users SavaPage would like its own encrypted PDF documents to be respected in the same way.

However, when an encrypted document is allowed to be printed, SavaPage would like to be able to receive it as printer, so it can be previewed and Proxy Printed. But, for that to happen we need to convert it to SafePages, i.e. to PDF format. That's where we are facing the **ps2pdf** barrier.

To solve this issue SavaPage has an optional workaround that ignores the PostScript notice at the point where we need the **ps2pdf** program to create the PDF, so SafePages can be displayed and Proxy Printed as intended.

The workaround is activated when Allow Encrypted PDF for Proxy Printing is enabled.

14.10.3. PostScript Configuration Properties

Configuration property	Description
Driver Print	
print-in.ps.driver.images.trigger	Trigger to render driver printed PostScript pages to images: `AUTO` = trigger when `ps2pdf` produces `stderr` messages (default), `OFF` = never trigger, `ON` = always trigger.
print-in.ps.driver.images.dpi	DPI for triggered images of PostScript driver print. Default `300`.
print-in.ps.driver.detain	Temporarily detain driver printed PostScript file in the JVM Temporary Files folder. `OFF` = never detain (default), `ON` = always detain, `AUTO` = detain when `ps2pdf` produces `stderr` messages.
Driverless File Transfer Print	
print-in.ps.driverless.images.trigger	Trigger to render driverless transferred PostScript file pages to images: `AUTO` = trigger when `ps2pdf` produces `stderr` messages (default), `OFF` = never trigger, `ON` = always trigger.

[11] http://scripts.sil.org/ofl
[12] https://software.sil.org/fonts/

Configuration property	Description
print-in.ps.driverless.images.dpi	DPI for triggered images of PostScript driverless file transfer. Default `300`.

Table 14.10. PostScript Print Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these properties.

14.11. IP Restricted Printing

Beware that SavaPage printer access may be restricted based on the requesting IP address. See:

- Figure 4.58, “Admin Web App: Queue - Edit” [123]
- Figure 4.110, “Admin Web App: Options - Web Print” [166]

CIDR Notation is used to specify the allowed IP address ranges.

14.12. Printer Availability

When SavaPage is temporarily unavailable, for instance during database backup or user synchronization, all printing requests are rejected.

- Driverless File Transfer Print will communicate a message saying “The application is temporarily unavailable, please try again later”.
- IPP Driver Print responds with HTTP status code 503 (service unavailable).
- JetDirect Driver Print requests are silently ignored.
- All rejected requests are logged as real-time activity in the Admin Web App Dashboard.

See Section 4.7, “Queues” [120].

Chapter 15. Authenticated Printing

Authentication in a printing environment is the act of confirming the digital identity of the person who issued a print job. Knowledge of this identity is crucial for SavaPage to securely offer its services to the right user. The next sections discuss authenticated printing in:

- Single Sign-On (SSO) Domains
- Peer to Peer Networks

But first, let us introduce the key authentication concepts where our discussion is based upon.

15.1. Key Concepts

This section lists the main authentication concepts headed with a short *term*. Each term is defined with a concise description, optionally followed with more details and a list of invariants.

15.1.1. User

An actor with a unique identity.

15.1.2. Person

A User who represents a real human being, as opposed to an abstract human role, software service or hardware device.

- Only Persons can login to SavaPage.
- Any User can print to a SavaPage Printer. However, SavaPage assigns a print job to a Person.

15.1.3. Domain User

A User defined in a SSO domain.

15.1.4. Synchronized User

A SavaPage User synchronized from a User Source.

- SavaPage assumes each Synchronized User is a Person, but Administrators can mark a user as Abstract.

15.1.5. Synchronized Person

A Synchronized User that is a Person.

15.1.6. Internal Person

A Person who is internally defined in SavaPage (opposed to a Synchronized Person).

15.1.7. Authenticated User

A User authenticated on a SSO domain by a workstation login.

15.1.8. Authenticated Abstract User

An Abstract User authenticated on a SSO domain by a workstation login.

- Before Authenticated Abstract Users can print to a SavaPage Printer they need to login to the SavaPage Web App on the same device from which they use the printer.

15.1.9. Authenticated Person

A Synchronized Person authenticated on a SSO domain by a workstation login.

- Authenticated Persons can print to SavaPage without being logged in to the Web App.

15.1.10. Trusted SavaPage Queue

A SavaPage Print Queue whose print jobs are trusted to originate from Authenticated Users.

- Each SavaPage Print Queue is untrusted by default. However, administrators can mark SavaPage Print Queues as *trusted* by setting its Authentication to `None`.
- Every job of a trusted SavaPage Queue is checked for the originating User. When this user is an Abstract User, SavaPage uses IP Based Authentication to deduce the associated Person. When the Person cannot be deduced the job is ignored.
- Note that the "trust" qualification is SavaPage internal use only, and not related to network domain trust in any way.
- SavaPage Print Queues are IPP based and, from a network point of view, are publicly accessible by nature.
- SavaPage does not bet on native domain trust alone, and accepts public network access as a given fact. But even in this case, SavaPage Print Queues can still be *internally* trusted if access is limited to authorized users on a network level. Stated the other way round: administrators need to prevent that users who connect to the network unauthenticated, e.g. with their personal laptop, use Trusted SavaPage Queues. SavaPage adds a helping hand here with an option to restrict print queue usage to a specific range of IP addresses. This makes it possible for instance to deny trusted queue access for wireless users who get their IP addresses from a distinct DHCP server issuing leases from a distinct IP range.

Caution

When non-domain users are allowed to print to trusted SavaPage Printers an accidental match with a Synchronized Person may lead to undesirable results.

15.1.11. Public SavaPage Queue

A SavaPage Print Queue where print jobs are not trusted to originate from Authenticated Users.

- A public queue must be explicitly marked for `Standard` or `Basic` Authentication in the SavaPage Admin Web App.

15.1.12. Standard IP Based Authentication

Deduction of the printing Person by matching the IP address of the originating host of the print job with the authenticated SavaPage Web App Session on the same host.

- This type of authentication is applied for jobs printed to a Public SavaPage Printer with `Standard` Authentication, or from a Trusted SavaPage Queue where the originating User is Abstract.
- When no authenticated Web App session is found the job is ignored.

15.1.13. Basic Authentication

Getting the identity of the printing Person by an IPP enforced Basic Authentication[1] (user name/password) dialog.

- This type of authentication is applied for jobs printed to a Public SavaPage Printer with `Basic` Authentication,

[1] https://en.wikipedia.org/wiki/Basic_access_authentication

15.1.14. Mail Print Authentication

Deduction of the printing Person using the email address of the sender.

- This type of authentication is applied for print jobs coming in from Mail Print.
- When no unique matching Person is found, or when the Person is disabled, authentication fails. Consult Section 4.4.5, "Edit User" [102] on how to mark a User as (enabled) Person.

15.1.15. Local User

A User defined on a local device.

15.1.16. Local Abstract User

An Abstract User defined on a local device.

15.1.17. Local Person

A Person defined on a local device.

15.1.18. User Alias

An alternative name for a User.

- A single User can have several aliases.
- An alias is applied at the following levels:
  - User login to the User and Admin Web App via the Login dialog, or the XML Web Services API.
  - Print jobs arriving in the SavaPage queues under the alias name.

For more information see Section 15.5, "User Name Aliases" [280].

15.2. Abstract User

A User who is not a Person.

15.3. Single Sign-On Domains

In a *Single Sign-On* (SSO) network user authentication is achieved in a login dialog where the User supplies unique credentials, usually by entering their user name and password[2]. SSO networks establish a web-of-trust between *users* and *domain services*. System administrators like SSO domains, because they provide a single interface to control access of Domain Users to servers and services.

Making SavaPage part of an SSO domain is the most sophisticated setup possible. In this way access to the SavaPage queues can be controlled on a domain defined user and group level, and by doing so we can create *authenticated queues*.

Authenticated SavaPage *IPP* Print Queues can exclusively be achieved in a macOS or GNU/Linux SSO domain using LDAP or NIS (Network Information Service) as authentication services[3].

Windows Domains have their own rules regarding IPP Print Queues. An alternative is to install local printers connected to SavaPage by the JetDirect/RAW protocol. These RAW IP printers are typically installed on a Windows Print Server.

[2] Of course methods like a smartcard and biometric (fingerprint) authentication should be mentioned as alternative.

[3] NIS (Network Information Service) protocol, also known as Sun Yellow Pages (YP) allows the information contained in the files `/etc/passwd`, `/etc/group` and `/etc/shadow` to be hosted on a central server. Administrators can enter, edit and delete the information on the NIS server so that it is automatically available on all Unix nodes. Authentication services are usually delegated to a Kerberos server, which thanks to tickets and authenticators eliminates the need to move passwords over an open and insecure network. NIS operates on "flat" domains and is therefore unsuitable for large organizations which due to their nature may be organized hierarchically. In such cases, the use of the LDAP (Lightweight Directory Access Protocol) is the way to go.

To exclusively grant access to the SavaPage printer by the print server, enter the server IP address (as CIDR) as the allowed client IP address in the Default / Queue definition. See Figure 4.58, “Admin Web App: Queue - Edit” [123].

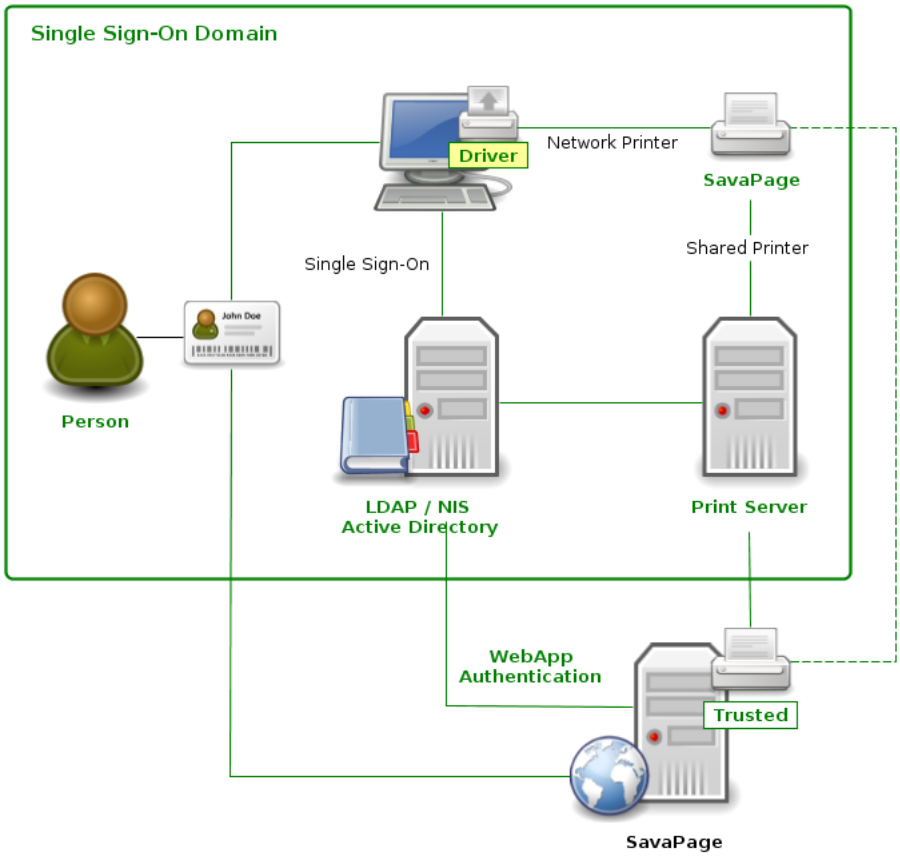


Figure 15.1. SavaPage in a Single Sign-On Domain

Trust is indeed essential to match the print job with the user's *SafePages* as previewed in their authenticated browser session. But, as we shall see in the next section, even in trust relations there are some loopholes to consider, and in networks where access is not fully guarded by SSO, unauthenticated users also need our special attention.

15.3.1. Authentication Loopholes

Although fully closed SSO domains provide unambiguous trust, there are common authentication loopholes that needs to be addressed. These loopholes are generic in nature and not related to SavaPage.

1. A loophole is introduced when multiple users use the same account (user name) to authenticate to the network. Because the login is based on the person's *role* we can not retrieve the *unique user identity*. If for example, both John and Mary logged in with the generic `student` account, there is no way to find out if a SavaPage print job from this session was issued by John or Mary. By default the print jobs of John or Mary will end up in the SafePages of the one and only unknown `student`. In situations where printing content is private this might pose a problem. In SavaPage this loophole can be solved by marking the generic user account as *abstract*. See Section 15.1.12, “Standard IP Based Authentication” [275].
2. A similar loophole is introduced when different users (sequentially) use the same machine, which was started in auto-login mode. Because the login is based on the *machine identity* we can not retrieve the *unique user identity*. In SavaPage this loophole is solved by the marking the auto-login user account as *abstract* . See Section 15.1.12, “Standard IP Based Authentication” [275].

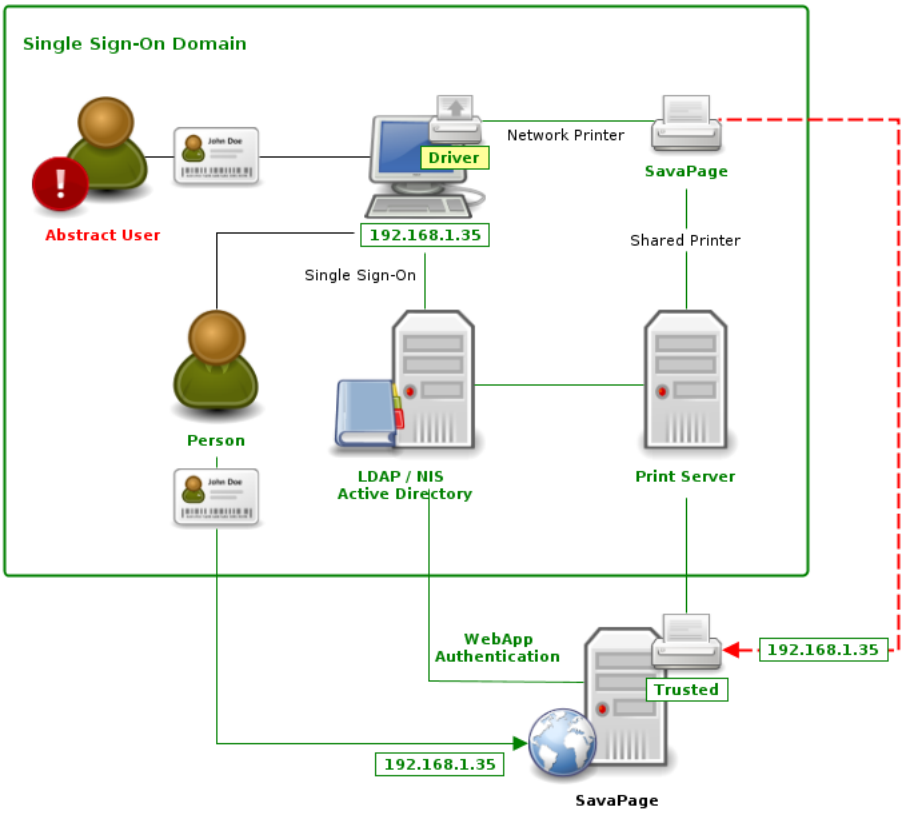


Figure 15.2. IP Based Authentication for Abstract User

15.3.2. Unauthenticated Users

In networks where access is not fully guarded by SSO, SavaPage queues introduce a security issue when they are used by unauthenticated non-domain users. For example, consider a guest user who connects a personal laptop to the network, and installs and prints to a SavaPage printer. In SavaPage this loophole can be solved by marking the *queue* as *untrusted*, i.e. by creating a Public SavaPage Queue. See Section 15.1.12, “Standard IP Based Authentication” [275] and Section 15.1.13, “Basic Authentication” [275]. In addition the Internal Users feature can be used to offer out of domain Web App authentication for guest users.

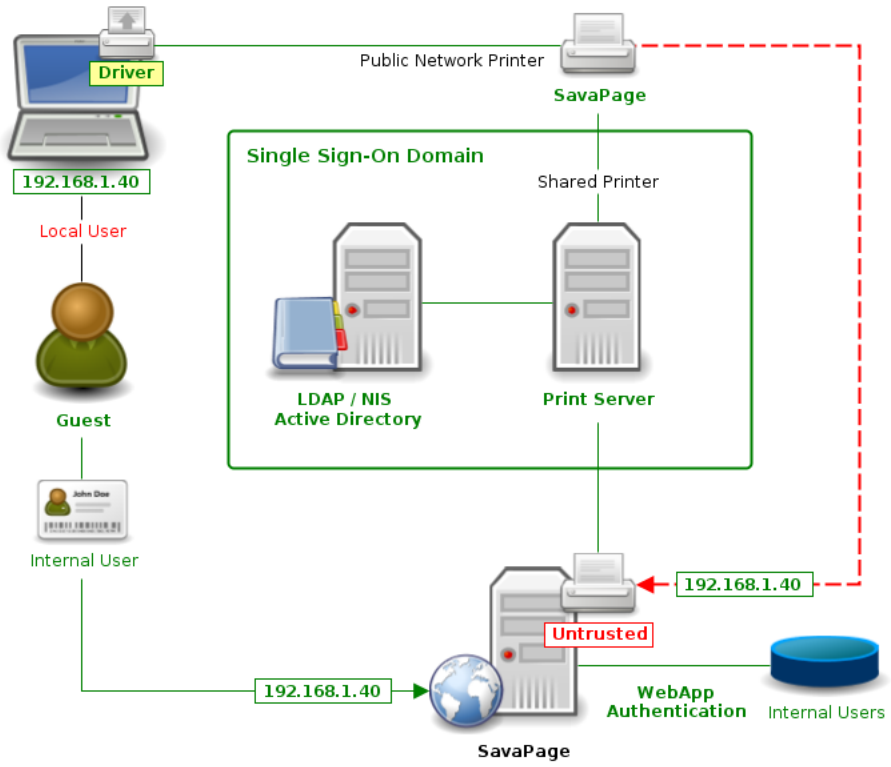


Figure 15.3. Authentication for Unauthenticated User

15.4. Peer to Peer Networks

A peer-to-peer or workgroup environment differs fundamentally from the network domain model. In the domain model, users authenticate with a unique (password protected) user name, as defined in a central server, while in a workgroup user identity is validated against a Local User rather than a central authority. The workstations are either set up to automatically login as a general "user", or user accounts are created locally as required.

Trust can be enforced by creating a Public SavaPage Queue, see Section 15.1.12, “Standard IP Based Authentication” [275] and Section 15.1.13, “Basic Authentication” [275], and by using the Internal Users feature for Web App authentication.

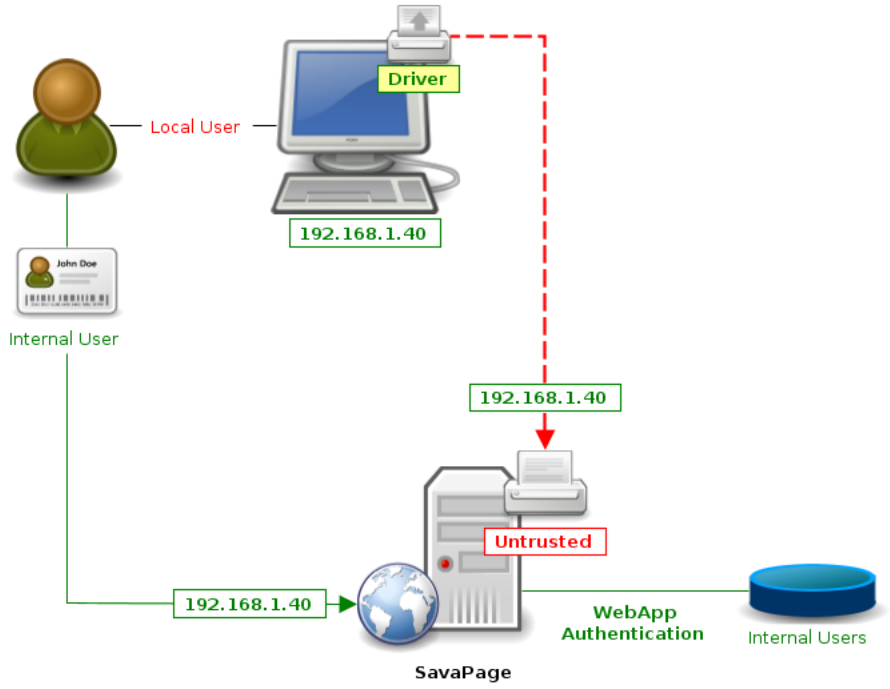


Figure 15.4. IP Based Authentication in Peer-to-Peer Network

15.5. User Name Aliases

A User Alias is a way of mapping a user name in one format to a name in another. It is useful in the following situations:

- Providing extra convenience for users to log into the system with a user name formatted in a different way. So Georg Friedrich Händel can have a User Alias of “georg_handel”, “gf_handel”, and “gfhandel”.
- Used as a temporary tool to manage domain or user name format changes. For example, you may have changed names from j.brown to john.brown. An alias can help forgetful users to log in with their old name.

Name aliases are applied at the following levels:

- User Login to the User and Admin Web App.
- Print jobs arriving under the alias name.

The aliases information is kept in the file,

`/opt/savapage/server/data/conf/username-aliases.txt`

and can be created based on the provided template file,

`/opt/savapage/server/data/conf/username-aliases.txt.tmpl`

You can create your own custom alias file as follows:

- Go to the directory `/opt/savapage/server/data/conf/`
- Open your favorite text editor with the file `username-aliases.txt.tmpl`
- For example, add the following lines to the end:

```
j.brown : john.brown
jbrown  : john.brown
```

- Save file as `username-aliases.txt`

The format of the alias file is:

```
aliasname1: username1
aliasnameA: username2
aliasnameB: username2
```

where `aliasname` is mapped to `username` in the system database. A user may have multiple aliases. In this example, `username2` is known both as `aliasnameA` and `aliasnameB`. The separator between `aliasname` and `username` can be ":", "=" or tab.

Warning

If an offered user name does not match an alias in the alias file, it is assumed it represents the user's real name. If this user is new to the system he might be created automatically in SavaPage, according to the user creation policy defined in the Options → User Creation → On demand user creation section of the Admin Web App. So please take care that your alias list is valid and up-to-date.

Tip

The content of the user name aliases file is read and cached when the SavaPage service is started. See Section 4.11.14.2, "User Name Aliases Cache" [179] on how to refresh the cache after you edited the file content.

Chapter 16. Printing Impact

One of the goals of SavaPage is to reduce hard-copy printing by facilitating the use of soft PDF copies instead. Above that, if printing is needed after all, SavaPage offers easy n-up, gray-scale and duplex proxy-printing to reduce printing even more. Giving feedback to users about the costs and environmental impact of their printing habits is used to arouse awareness and achieve behavioral change.

16.1. Financial Impact

In any organization the costs of unrestricted access to office printers can be substantial. With SavaPage Financial feedback about the costs from different perspectives (User, ProxyPrinter, Period) is within reach. Future releases will indeed process log data, and present financial statistics from all possible angles.

16.2. Environmental Impact

Environmental issues like global warming, waste management, paper production and consumption are an area of debate and interest to many. Highlighting the environmental impact of printing is one of the ways to influence user behavior.

SavaPage uses the number of printed *Sheet Units* to calculate three impact metrics: trees and energy consumption, and carbon production.

> **Important**
>
> The default values SavaPage uses for environmental metrics can be the subject of debate. Of course you are free to set the metric to any value that works for you. Please inform us about facts and findings you feel confident about.

16.2.1. Printed Sheet Units

A *Sheet Unit* (SU) is the size equivalent of an A4 sheet. So,

```
A4 == 1.00 SU
A3 == 2.00 SU
A2 == 4.00 SU
A1 == 8.00 SU
```

and ...

```
A5 == 0.50 SU
A6 == 0.25 SU
A7 == 0.12 SU
A8 == 0.06 SU
```

Note that SU precision is 2 decimals. As environmental impact is concerned, A4 and US Letter sheets are handled as equivalent, so ...

```
US Letter == 1.00 SU
```

SavaPage uses the media size chosen by the user to calculate the printed Sheet Units of a Proxy Printer print job. Some print examples:

- 6 pages double-sided on A4 : `3 SU`
- 6 pages double-sided on A3 : `6 SU`
- 6 pages 2-up on A4 : `3 SU`
- 6 pages 2-up double-sided on A4 : `2 SU`

Warning

SavaPage is not able to anticipate printer intelligence, for instance, when a printer uses different trays (with different media sizes) for different page sizes within the job document.

16.2.2. Trees

This metric is the percentage of a tree used to produce the paper of the printed Sheet Units. The metric is adopted from `Conservatree.org`[1] and is as follows:

- A prototypical tree is 40 feet tall and 6-8 inches in diameter.
- One tree makes 16.67 reams of copy paper or 8,333.3 SU.

The metric `83333` is set as default for the configuration key: `environment.sheets-per-tree`

16.2.3. Energy

This metric is the energy used to produce the paper of the printed Sheet Units. The metric is adopted from `Paper-online.org`[2] and is as follows:

- Around 500 kWh of energy are required in Europe to make 200 kg of paper.
- So one A4 or Letter sheet of office paper costs 12.5Wh to manufacture [3].

The metric `12.5` is set as default for the configuration key: `environment.watt-hours-per-sheet`

16.2.4. Carbon

This metric is the amount of CO_2 released for producing the paper of the printed Sheet Units. The metric is adopted from the `Swiss Federal Institute of Technology Zurich`[4] and is as follows:

- One A4 or Letter sheet costs 5.1g of CO_2 to production.

The metric `5.1` is set as default for the configuration key: `environment.co2-grams-per-sheet`

Note

This metric takes into account the CO_2 produced as a byproduct of paper production only. It does not take into account the CO_2 related to the production and operation of the printer and the ink or toner cartridges. Defining broader system boundaries and tracking down all parameters involved requires a major effort, and is beyond the scope of this manual. Of course you are free to set the parameter value for this metric to any value that works for you.

[1] http://conservatree.org/learn/EnviroIssues/TreeStats.shtml
[2] http://www.paperonline.org/uploads/AskGuenter/MR_produce%20energy.pdf
[3] A sheet of 5g requires (5/200.000) * 500 = 0.0125 kWh = 12.5 Wh.
[4] http://www.umwelt.ethz.ch/dokument/factsheets/sustainable_conference_compensation_en.pdf

Chapter 17. Security

This chapter discusses how SavaPage secures sensitive user and application data, and how it communicates with external Information Providers.

17.1. HTTP Service Filters

HTTP Service Filters enforce access policies at the point where remote clients connect to the SavaPage server. Connection requests may be delayed, throttled, rejected, aborted or denied. Statistics are shown in the Admin Web App. See Section 4.3.7, “Filtered Requests” [97].

> **Note**
>
> Details of the filtered requests are persisted in the rotating log file:
>
> `/opt/savapage/server/logs/security.log`
>
> This file has a tab separated value (TSV) format for easy import and manipulation into spreadsheet programs. See `log4j.properties` for more information.
>
> If your `log4j.properties` file does not contain the "Security Logger" section, copy it from the `log4j.properties.template`.

17.1.1. Denial of Service Filter

A Denial of Service Filter keeps track of the number of requests from a connection to the SavaPage server per second. If a limit is exceeded, the request is either delayed, throttled, rejected or aborted. A throttled request is placed in a queue and will only proceed when there is capacity. HTTP status code `429` is sent if there are too many requests.

Configuration property	Description
system.dosfilter.enable	Set to `Y` (default) or `N` to enable/disable the Denial of Service Filter.
system.dosfilter.delay-msec	Delay (in milliseconds) that is applied to all requests over the rate limit `system.dosfilter.max-requests-per-sec`, before they are considered at all. `-1` means just **reject** request, `0` means no delay (**throttled**), otherwise it is the **delay**. Default: `100`.
system.dosfilter.max-idle-tracker-msec	Maximum amount of time (in milliseconds) to keep track of request rates for a connection, before deciding that the user has gone away, and discarding it. Default: `30000`.
system.dosfilter.max-request-msec	Maximum amount of time (in milliseconds) to allow the request to process. Default: `30000`. If a request exceeds this threshold it is interrupted and HTTP status code `503` is returned.
system.dosfilter.max-requests-per-sec	Maximum number of requests from a connection per second. Requests in excess of this are first delayed, then throttled. Default: `75`. Also see Section 13.5.4, “HTTP/2 Settings” [251].

Configuration property	Description
system.dosfilter.throttle-msec	Amount of time (in milliseconds) to async wait for semaphore. Default: `30000`.
system.dosfilter.throttled-requests	Number of requests over the rate limit able to be considered at once. Default: `5`.
system.dosfilter.too-many-code	HTTP status code to send if there are too many requests. Default: 429 (too many requests). 503 (service unavailable) is another option.
system.dosfilter.whitelist	Comma-separated list of host names, IPv4 addresses in decimal notation `A.B.C.D`, and canonical (non-compressed) IPv6 addresses that will not be rate limited. IPv4 CIDR notation `A.B.C.D/M` is supported, IPv6 CIDR notation is **not**. Note: `localhost` , `127.0.0.1` and `0:0:0:0:0:0:0:1` are implicitly whitelisted.
system.dosfilter.warn-interval.applog.mins	Interval (minutes) within which filter events are totaled and written as warning in the Application Log. The expiration of the interval is triggered by a filter event. Default: `5`.
system.dosfilter.warn-interval.webapp.secs	Interval (seconds) within which filter events are totaled and send as Real-time Activity warning with popup message to the Admin Web App. Default: `60`.

Table 17.1. Denial of Service Filter : Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these properties.

Caution

Please exercise extreme caution when configuring these properties, as doing so may result in losing access to all HTTP communication with SavaPage. For example, setting `system.dosfilter.max-request-msec` too low can result in a permanent HTTP status code `503`.

If such a lockout occurs, you can enter the correct values using a Database Command, as in this example:

```
sudo su - savapage
cd server/bin/linux-x64
./savapage-db --db-config-set system.dosfilter.max-request-msec=30000
```

See Section C.5.4, “db-config-set” [354].

Restart SavaPage to apply the changes. See Section C.6, “Stopping and Starting the Server” [356].

17.1.2. Inet Access Filter

An Inet Access Filter allows/denies remote access to the SavaPage server according to a list of IP addresses. HTTP error code `403` is sent if a request is denied.

Configuration property	Description
system.inetfilter.enable	Set to `Y` or `N` (default) to enable/disable the Inet Access Filter.
system.inetfilter.whitelist	A list of comma-separated list of IPv4/6 addresses, individual or in CIDR notation that are allowed access. Requests from remote addresses that do not match the whitelist are denied with HTTP error code `403`. Note: `127.0.0.1` and `0:0:0:0:0:0:0:1` are implicitly whitelisted.
system.inetfilter.whitelist.empty-allow-all	Set to `Y` (default) or `N` to enable/disable empty white list to be treated as "match all". If `N`, an empty whitelist means that only `127.0.0.1` and `0:0:0:0:0:0:0:1` are allowed.

Configuration property	Description
system.inetfilter.blacklist	A list of comma-separated IPv4/6 addresses, individual or in CIDR notation. Requests from remote addresses that match the list are denied with HTTP error code `403`. Black list entries are always applied, so that even if an entry matches the white list, a black list entry will override it. Note: `127.0.0.1` and `0:0:0:0:0:0:0:1` are *not* allowed on the list.
system.inetfilter.warn-interval.applog.mins	Interval (minutes) within which denied requests are totaled and written as warning in the Application Log. The expiration of the interval is triggered by a InetAccessFilter event. Default: `5`.
system.inetfilter.warn-interval.webapp.secs	Interval (seconds) within which denied requests are totaled and send as Real-time Activity warning with popup message to the Admin Web App. Default: `60`.

Table 17.2. Inet Access Filter : Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these properties.

17.1.3. HTTP/2 Rate Control

SavaPage supports both HTTP/1.1 and HTTP/2[1] communication protocols. HTTP/2 can be disabled or customized as desired. See Section 13.5.4, “HTTP/2 Settings” [251].

17.2. User Authentication

This section discusses how user credentials are protected.

17.2.1. Login Passwords

This section is about the passwords and PIN codes entered in the Web App Login Dialog.

Tip

Users can use the HTTPS protocol for connecting to the Web App, so data is encrypted to and from the server.

17.2.1.1. User Domain Passwords

SavaPage does *not* store or cache user domain login passwords. These passwords are always checked real-time at the source.

17.2.1.2. Internal User Passwords

Passwords of Internal Users are stored as SHA1 hash in the database.

17.2.1.3. Internal Admin Password

The SHA1 hashed password of the internal administrator `admin` is stored in a text file located at `/opt/savapage/server/data/admin.properties`. Access to this file is restricted to the `savapage` user.

SavaPage installs with `admin` as initial password for user `admin`.

[1] https://en.wikipedia.org/wiki/HTTP/2

Tip

If you forgot the internal admin password, you can reset it by editing the `admin.password` property in the `/opt/savapage/server/data/admin.properties` text file. Ignore the existing HASH value. SavaPage will hash your password upon first use.

17.2.2. PIN Codes

User PIN codes are stored in the database as encrypted secret.

17.2.3. Authentication Tokens

When Authentication Persistence is enabled for Browser Local Storage, authentication tokens are stored in the “Local Storage” of the browser. See Section 4.11.4, “User Authentication” [157].

Separate authentication tokens are held for the User, Admin, POS and Job Tickets Web App context and the same token is used for different sessions (on different devices) of a single user. A explicit logout in the Web App destroys the token. Authentication tokens are managed in memory on the SavaPage server. So, when the server restarts all local tokens are implicitly invalidated.

17.2.4. One-time Authentication Tokens

A Trusted Third Party (TTP) can acquire a one-time token for Web App user authentication by calling an XML-RPC method. The expiration time of the token should be as short as possible to minimize the risk that an accidentally exposed token can be misused. See Section C.2.1.1, “onetime-auth.createToken” [342] for details.

17.2.5. User Dialog

When authentication fails a neutral "Authentication failed" message is communicated to the user to prevent “Account Enumeration” and “Guessable User Account”.

17.3. Access over Internet

Take extra care when SavaPage is accessible over public Internet, as a result of enabled Internet Print or explicit WAN to LAN routing, since authentication falls back to global defaults for User Authentication. At least make sure that access to the Admin Web App is solidly secured.

Internet access to each Web App can be restricted with configuration properties in the table below:

Configuration property	Description
webapp.internet.enable	Set to `Y` (default) or `N` to enable/disable Internet access for **all** Web Apps.
webapp.internet.*.enable * = admin | jobtickets | mailtickets | payment | pos | printsite | user	Set to `Y` (default) or `N` to enable/disable Internet access for a specific Web App.
webapp.internet.*.auth-mode.enable * = admin | jobtickets | mailtickets | payment | pos | printsite | user	Set to `Y` or `N` (default) to enable/disable webapp.internet.*.auth-modes for a specific Web App. Note: if webapp.internet.admin.auth-mode.enable = `Y`, then user `admin` is denied access to Admin WebApp over the internet.

Configuration property	Description
webapp.internet.*.auth-modes * = admin | jobtickets | mailtickets | payment | pos | printsite | user	A comma-separated list of authentication methods (`name`, `email`, `id`, `nfc-local`, `yubikey`, `oauth`) for a specific Web App. The first in the list is the default method.

Table 17.3. Web App Internet Access Configuration Properties

See Section 4.11.15, "Config Editor" [189] on how to set these items.

Important

If `webapp.internet.admin.auth-mode.enable` = `Y`, then user `admin` is denied access to the Admin Web App over the internet.

Note

Exceptions to these generic restrictions can be configured by creating a Terminal Device for each trusted client IP address and configure Custom User Login methods.

17.3.1. security.txt

security.txt[2] is a proposed standard which allows websites to define security policies. The file is available via `https://your-savapage-server/.well-known/security.txt` and its content is set with configuration properties as outlined in the table below:

Configuration property	Description
securitytxt.enable	Set to `Y` or `N` (default) to enable/disable security.txt for **all** Web Apps.
securitytxt.contact.mailto	E-mail address for people to contact you about security issues.
securitytxt.contact.tel	Phone number for people to contact you about security issues.
securitytxt.contact.url	URL for people to contact you about security issues. Must start with `https://`. *At least one securitytxt.contact.* property must be configured.*
securitytxt.encryption.uri	URI to a key which security researchers should use to securely talk to you (optional).
securitytxt.acknowledgments.url	URL to a web page where you say thank you to security researchers who have helped you (optional). Must start with `https://`.
securitytxt.preferred-languages	A comma-separated list of language codes that your security team speaks (optional).
securitytxt.securitytxt.policy.url	URL to a policy detailing what security researchers should do when searching for or reporting security issues (optional). Must start with `https://`.
securitytxt.securitytxt.hiring.url	URL to any security-related job openings in your organization (optional). Must start with `https://`.

Table 17.4. Configuration Properties for security.txt

[2] https://securitytxt.org/

See Section 4.11.15, “Config Editor” [189] on how to change these properties.

> **Note**
>
> The “Expires:” property in `security.txt` is generated by SavaPage and set one month in the future.

17.4. Web Sessions

This section discusses the security and performance of web sessions.

17.4.1. Web Session Timeout

Web Sessions guard persistent authorized access to SavaPage. By default, all sessions expire after a certain period of inactivity. Each interaction with the Web App that results in a call to the SavaPage Web Server resets the inactivity timer. Explicitly logging out of any SavaPage Web App will *immediately* end the session. However, closing the browser window or tab won't: the session will be marked for removal after expiration. The actual removal of expired sessions is performed by a cyclic scavenge process. The cycle interval can be set in the server.properties file.

The default timeout periods for different login types are shown in the table below:

Login type	Default value
Admin Web App	1440 minutes (24 hours)
User Web App	60 minutes (1 hour)

Table 17.5. Default Web Session Timeout Values

The timeout value (in *minutes*) can be changed using the configuration properties below. A value of 0 indicates that the session will never time out: the downside is that these sessions are not scavenged ever. A positive value will make sure that sessions are scavenged at some point in time.

Configuration property	Description
web-login.admin.session-timeout-mins	Inactivity timeout for the Admin Web App
web-login.user.session-timeout-mins	Inactivity timeout for the User Web App

Table 17.6. Web Session Timeout Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to change these properties.

Changed inactivity timeout values take effect for new sessions.

When a session has expired in an open Web App, users are given a warning message, and must login again. When Authentication Tokens are used, users will see the message, but will be able to continue without the need to login.

Note that some pages periodically refresh the page (or data on the page), such as the Dashboard. A session will not time out if a browser is left on these pages, as it will be considered active.

17.4.2. Web Session Cookies

Session tracking cookies like `SP_JSESSIONID` and `SP_BAYEUX_BROWSER` are marked as *HttpOnly*. An HttpOnly cookie cannot be accessed by client-side APIs, such as JavaScript, and may therefore help mitigate certain kinds of cross-site scripting attacks.

17.4.3. Web App WatchDog

A WatchDog within a Web App continuously monitors browser performance and restores server connections if necessary. Optionally, a message can be displayed to draw attention to connectivity or performance issues.

Configuration property	Description
webapp.watchdog.interval-secs	The interval in seconds after which the Web App WatchDog is supposed to report. Default: `3` seconds. Note: The WatchDog is implemented as a JavaScript timer configured to emit a signal (callback) at fixed intervals. In practice the callback regularity is subject to the browser's rendering and scripting pipeline. When a timer's callback is scheduled, it's placed in a queue. If other, higher-priority tasks are being performed, the timer callback waits, causing the WatchDog report to be delayed. The delay is an indication of performance or connectivity issues.
webapp.watchdog.allowed-delay-secs	The maximum allowed delay (in seconds) for WatchDog reports. If the maximum is exceeded, connectivity is checked and restored if needed. Default: `7` seconds.
webapp.watchdog.message.enable	Set to `Y` (default) or `N` to display/hide (popup) messages after the maximum allowed delay has been exceeded.

Table 17.7. Web App WatchDog Configuration Properties

See Section 4.11.15, "Config Editor" [189] on how to change these properties.

Note

Web App WatchDog is implemented in User Web App.

17.5. Rate Limiting

Rate Limiting[3] with Token Bucket[4] algorithm is used to suspend the processing of requests from an IP address when a maximum number of processing failures (unauthorized access or processing errors) has been reached within a given time period. Rate Limiting events are reported in the Admin Web App Dashboard as Real-time Activity and in the `server.log`.

Configuration property	Description
system.ratelimiting.enable	Set to `Y` (default) or `N` to enable/disable Rate Limiting.
system.ratelimiting.user-auth.enable	Set to `Y` (default) or `N` to enable/disable Rate Limiting for User Authentication in Web Apps and the User Client Desktop Application. See Section 3.1, "Login" [22] and Chapter 11, *User Client* [240].
system.ratelimiting.user-auth.failures-per-min	The maximum number of User Authentication failures in Web Apps per IP address per minute. The processing of any attempt that exceeds this maximum will be suspended until there is room again within the one minute time frame. Default: 5.
system.ratelimiting.user-reg.enable	Set to `Y` (default) or `N` to enable/disable Rate Limiting for Internal User Self Registration. See Section 4.11.1.3, "Internal Users" [153].

[3] https://en.wikipedia.org/wiki/Rate_limiting
[4] https://en.wikipedia.org/wiki/Token_bucket

Configuration property	Description
system.ratelimiting.user-reg.max-per-hour	The maximum number of User Self Registrations per IP address per hour. Any registration attempt that exceeds this maximum is rejected with a "Registration is currently unavailable" message, until there is room again within the one hour time frame. Default: 10.
system.ratelimiting.api.enable	Set to `Y` (default) or `N` to enable/disable API Rate Limiting. See Section C.2, "Web Services" [341], Section C.3, "Atom Feed Service" [352] and Section N.5.1, "SavaPage as Sync Source in PaperCut" [417].
system.ratelimiting.api.failures-per-min	The maximum number of API failures per IP address per minute. The processing of any attempt that exceeds this maximum will be suspended until there is room again within the one minute time frame. Default: 5.
system.ratelimiting.print-in.enable	Set to `Y` (default) or `N` to enable/disable Print-in Rate Limiting. See Section 4.7, "Queues" [120].
system.ratelimiting.print-in.failures-per-min	The maximum number of Print-in failures per IP address per minute. The processing of any attempt that exceeds this maximum will be suspended until there is room again within the one minute time frame. Default: 5.
print-in.ipp.default.wait-after-failure-msec	A wait time in milliseconds before returning after an IPP Print-in failure. This causes a next request, which may result in the same failure, to be delayed. Default: 5000. Note: the wait time is *only* applied if `system.ratelimiting.print-in.enable` = `N`.

Table 17.8. Rate Limiting Configuration Properties

See Section 4.11.15, "Config Editor" [189] on how to change these properties.

17.6. SSL Certificate and Password

Each time SavaPage server starts it checks if the default self-signed key and certificate and password file are present. If not, they are generated (a missing password file will generate both) with defaults described in Section C.7.1, "Re-Create the Self-Signed Certificate" [357].

The key is used by default when the system is accessed via HTTPS on port 8632. The password of the `default-ssl-keystore` is generated in `/opt/savapage/server/data/default-ssl-keystore.pw`. Access to this file is restricted to the `savapage` user.

The passwords for the installed keystore created from an imported Existing SSL Certificate are set in the `server.properties` file. Access to this file is restricted to the `savapage` user.

17.7. Secured JMX Connection

During the install process, SavaPage generates a dedicated self-signed key and certificate for securing JMX connections with SSL. The generated Java keystore is: `/opt/savapage/server/data/jmxremote.ks` with password `savapage`. Access to this file is restricted to the `savapage` user.

The JMX password for the `admin` user is held in `/opt/savapage/server/data/jmxremote.password`. Java needs the password to be provided in plain text, so access to this file is restricted to the `savapage` user.

The initial password is a random string generated by the installation process. It needs to be changed in the Admin Web App as described in Section 4.11.14.4, “JMX Agent” [180].

Note

The default SavaPage JMX port is 8639. This can be changed by editing the file:

```
/opt/savapage/server/data/jmxremote.properties
```

You need to restart SavaPage for this change to take effect.

17.8. Encrypted Secrets

Data in the database will not contain any explicit secret data. All secret data is stored encrypted with encryption keys held in file `/opt/savapage/server/data/encryption.properties`. This file is generated by SavaPage at initial installation and contains randomized encryption keys unique for this particular installation instance. Access to this file is restricted to the `savapage` user.

Important

Make a backup of `encryption.properties` immediately after installation and store it at a secure place, so you can restore it in case of a server crash or when you need to migrate to a new server.

Caution

The `encryption.properties` file is *crucial* for decrypting secret data in the database and verifying the authenticity of document signatures. So, when you restore a database backup on a different server, be sure to also restore this file.

The following secret data is stored encrypted in the database:

- Passwords.
- PIN codes.
- Authentication tokens.
- API keys.

17.9. Document Signature

SavaPage generates a digital signature for every document printed-out or downloaded. Digital signatures are generated using a cryptographic technique called Hash-based Message Authentication Code (HMAC)[5]. The algorithm takes various output job attributes such as job time, user name, document name and UUID, and combines them with a secret key. The result is then passed through the MD5 digest algorithm. The resulting signature is unique to the document instance [6]. The applied secret key ensures the *authenticity* of the signature.

The algorithm used is:

- `Digest = Hash(date time || user name || document name || document UUID)`
- `Signature = Hash(Digest || Key)`

where

[5] https://en.wikipedia.org/wiki/HMAC.
[6] The SHA1 digest algorithm is a stronger alternative, but MD5 is secure enough for our application and generates shorter signatures, which are easier to enter as argument to find the matching document.

- `Key` is a random string generated by SavaPage at initial installation. It is stored as `hmac.key` property in the `/opt/savapage/server/data/encryption.properties` file, which is also used for Encrypted Secrets.
- `Hash` is the MD5 function.
- `date time` is formatted in `ISO 8601` basic format from year to second (`yyyyMMddTHHmmss`). The time is local time (not UTC). E.g. `20120906T151231`.

Note

The signature is stored in the database for future use.

17.10. User Client

The User Client is a notifier of personal user events: see Chapter 11, *User Client* [240]. The following security measures apply:

- An SSL connection is used to communicate with the server.
- The server only accepts SSL connections.
- An API key is used as client identification.
- Access can be restricted by client IP addresses.

17.11. Server Commands

The Server Command tool provides a command-line interface to SavaPage Server methods. The following security measures apply:

- Only users with read access to the `server.properties` file have the right to execute the command.
- An SSL connection is used to communicate with the server.
- An API key is used as client identification.
- SavaPage server accepts SSL connections from local host only. When remote access to SavaPage is proxied, e.g. by Apache redirect, the remote address will be 127.0.0.1 in all cases. Therefore, requests from local loop-back address 127.0.0.1 are **not** accepted. The request origin must be a "real" IP address identical to the one of the SavaPage server.
- At the client, SSL host name verification is turned *off*. This allows a mismatch between the server host name and the SSL certificate CN.

17.12. Web Services

Public XML-RPC, JSON-RPC and RESTful API services are protected by secure endpoints with API keys.

17.13. Log Files

Log files are located in `/opt/savapage/server/logs`. This directory can be accessed by user `savapage` only. Although log files may contain sensitive user and application data, they never contain secrets like passwords and access codes.

Log file rotation is fully customizable in `log4j.properties`. See the documentation in this file for more information. A server restart is required for changes to take effect.

17.14. Network Card Reader

A setup where a *public* and *unattended* IP device communicates with a central server is inherently prone to security breaches. Our Network Card Reader device is no exception to this rule.

Although SavaPage validates the reader's IP address, the reader could be replaced and mimicked. Also, since communication is non-SSL, the Card Number (UID) of swiped NFC Cards could be hijacked. However, since the only content transmitted is the Card Number, misuse will be limited to a Card Number being offered from an unexpected origin at an unexpected time. Since offered Card Numbers are always processed in well defined transient contexts with short time limits, the risk of unnoticed abuse can be considered minimal.

A security breach of a fundamentally different nature is the rare scenario where it is possible to manipulate the UID of an NFC Card. A hacker could then use the hijacked card number to make a duplicate authentication token.

Tip

As an extra security measure *two-step authentication* can be implemented by requiring an additional PIN (over an SSL connection) after the initial NFC Card authentication. See Section 4.11.4, “User Authentication” [157] and Section 4.9.2.2, “Custom User Login” [147]

17.15. Internal Services

SSL transport is used for Internal Services, like publishing notifications for Real-time Activity.

17.16. External Services

Communication with external services can optionally be secured with SSL/TLS. See Section 4.11.1.2, “LDAP” [150], Section 4.11.5, “Mail” [162] and Section 4.11.6, “Mail Print” [164]

17.17. Vouchers

The Voucher System is designed for optimal security. Vouchers are assigned a random 16-digit number which makes a guess statistically near impossible. Above that, all unsuccessful (potentially fraudulent) voucher redemption attempts are detected and logged.

Like all security systems, the human factor is the most critical. Remember that vouchers represent *cash*, so take special care to protect the vouchers from unauthorized use.

- Delete the generated PDF voucher document after the vouchers are printed. You can always regenerate the PDF document when needed.
- Keep printed vouchers in a secure location.
- Put vouchers in envelopes to prevent exposure of voucher numbers.
- Check the Application Log regularly for unsuccessful (potentially fraudulent) voucher redemption attempts.
- Use the Voucher List to monitor successful voucher redemption.
- Delete or Expire a voucher batch immediately when vouchers are reported lost or stolen. See Section 4.15.1, “Voucher Actions” [207].

Caution

Voucher numbers are *not* encrypted in the database, so be careful to store database backup files at a save location.

Chapter 18. Freedom and Privacy

This chapter explains how Digital Freedom and Privacy is secured in the SavaPage domain. The privacy domain partially overlaps security (confidentiality), which include the protection of information. See Chapter 17, *Security* [284].

The About dialog in each Web App has links to the Free Software source code and legal context.

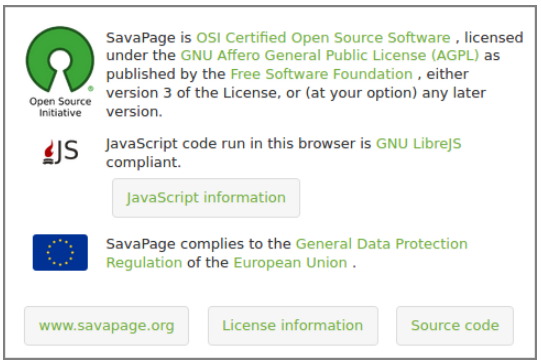


Figure 18.1. Web App - About Dialog - Software & Privacy Section

18.1. Open Source

SavaPage is Open Source Software, and complies to Free Software as defined by the Free Software Foundation[1]. The use of the AGPL Software License testifies of this compliance. Being free open source software, everyone has the freedom to study how SavaPage works and monitor how digital privacy is enforced. Code and licenses can be accessed by pressing the License information and Source code button in the About dialog.

18.1.1. Libre JavaScript

For a true Libre Web App, it is crucial to free all of its JavaScript code, as explained in The JavaScript Trap[2] by Richard Stallman. All JavaScript code run in SavaPage Web Apps is GNU LibreJS[3] compliant. Code and license of each JavaScript module can be inspected via the "JavaScript License Information" dialog that is opened by pressing the JavaScript information button in the About dialog.

Figure 18.2. Web App - JavaScript Info (first 2 of 20+ sources)

18.2. General Data Protection Regulation

Digital Privacy is a fundamental right in the primary law of the European Union, and the General Data Protection Regulation[4] (GDPR) is one of the main legal instruments. GDPR uses the following terminology (in short):

[1] https://www.fsf.org
[2] http://www.gnu.org/philosophy/javascript-trap.html
[3] https://www.gnu.org/software/librejs/
[4] https://ec.europa.eu/info/law/law-topic/data-protection_en

- *Data Subject* : a natural person, as in Section 15.1.2, “Person” [274].
- *Personal Data* : information related to a Data Subject. See Section 4.4.5, “Edit User” [102] and Section 3.8, “Log” [71].
- *Data Controller* : an entity that controls the Personal Data, and determines the purpose and means of processing. Members and Visitors of the SavaPage Community are Data Controllers.
- *Data Processor* : an organization that processes the Personal data on behalf of the Data Controller (e.g. in an outsourcing situation).

18.2.1. Data Portability

Data Portability is the Data Subject’s right to obtain information related to the processing of their Personal Data from the Data Controller. This allows the Data Subject to move, copy or transfer personal data easily from one IT environment to another.

SavaPage Data Portability is implemented as a download of personal data as CSV files in a ZIP container from the User List in the Admin Web App or optionally from a GDPR Dialog in the User Web App.

18.2.2. Data Erasure

Data Erasure is the Data Subject’s “right to be forgotten”. This right enables a person to request the removal of personal data when there is no compelling reason for its continued processing. The request can optionally be send from the GDPR Dialog in the User Web App.

Data Erasure is performed with a Server Command. See Section C.1.8, “eraseUser” [334] and Section 4.4.3, “Erased Users” [101].

18.3. Secure Print Release

SavaPage supports several scenario's where documents are not printed directly, but only after the user walks up to the printer and authenticates himself. This enables the rightful owner to directly retrieve printed copies of private or confidential documents. See Section 4.9.1.2, “Proxy Print Authentication” [142].

18.4. CUPS Privacy

CUPS makes “job-name” “job-originating-host-name” and “job-originating-user-name” private by default. This means that personal data are anonymized in the CUPS Web interface. See Section 2.3.4, “CUPS Job Privacy” [14].

Chapter 19. Internationalization

Internationalization is the process of designing a software application so that it can potentially be adapted to various languages and regions without engineering changes.

Localization is the process of adapting internationalized software for a specific region or language by adding locale-specific components and translating text.

SavaPage is internationalized software and can easily be localized to different languages, countries or regions.

19.1. Localization

The following localization rules apply:

- The Web App user interface localizes according to either the locale saved in the browser's local storage during a previous visit[1], or to the country/region code of the browser, or finally to the language chosen at Login.
- Reports generated from the Web App localize according to the language of the Web App, except for Vouchers and Receipts at the Point-of-Sale and User Web App, which use the System Locale.
- The System Locale is applied to all system messages as broadcasted to the Admin Web App Dashboard, written to the Application Log, or send by email.
- Text in log and audit files is fixed to the English language. So, when in need for support, these files can easily be understood by SavaPage Support.
- Dates and numbers are formatted according to the localization context.
- The currency symbol of the localization context is used.
- When localized text is not found the fall-back language will be English. So, in case SavaPage is partially translated for a selected locale, the user experience will be fragmented, as part of the text will fall back to English.

SavaPage handles all localized text and user entered data as Unicode. The Web Browser, and therefore the Web App, natively displays Unicode correctly. However, the correct display of Unicode in PDF reports, needs special attention. Therefore, Internal Fonts are available to customize PDF generation.

19.1.1. Notes for Translators

Currently SavaPage is fully localized to *English*, *German*, *French* and *Dutch*. If you want to localize the software to another region, the easiest way to get started is by using the Custom i18n feature with the `*.xml` files from the savapage-i18n-en[2] repository as a reference. In this way your translation is directly visible in the application.

Important

Please tell us about your localization effort by sending an email[3], so we can give you all the support needed.

19.2. Internal Fonts

The Unicode range of the displayed text in PDF documents must be covered by the embedded font.

[1] After login, the locale of the WebApp is saved in the browser's local storage, together with the Authentication Tokens.
[2] https://gitlab.com/savapage-i18n/savapage-i18n-en
[3] mailto:support@savapage.org

Unfortunately, at present there is no native *outline* font that can display all Unicode characters. The one exception is GNU Unifont, which *does* support the full 65,536 Unicode code point range. However, the glyphs originate from a *bitmap* of 16 pixels high and either 8 or 16 pixels wide, which gives the font a coarser look.

SavaPage contains internal fonts covering specific `Unicode Scripts`[4]. These fonts can be selected to customize PDF output to the content locale.

19.2.1. Default Font

The default font for PDF output is DejaVu Sans[5], which supports a broad set of Unicode scripts:

- Latin (including European and African alphabets, IPA, ...)
- Greek (including polytonic)
- Cyrillic
- Armenian
- Georgian
- Hebrew
- Arabic
- N'ko
- Lao
- Canadian Aboriginal Syllabics
- Ogham
- Tifinagh
- Lisu

Next to that, DejaVu also contains a lot of mathematical and other symbols, arrows, braille patterns, etc.

Tip

Coverage of the default font can be seen in `DejaVuSans.pdf`[6].

19.2.2. CJK Font

Support for Chinese, Japanese and Korean (CJK) is provided by the `Droid Sans "fallback" font`[7]. This font contains over 43,000 glyphs and includes support for Simplified Chinese (GB2312), Traditional Chinese (Big 5), Japanese (JIS 0208) and Korean (KSC 5601). The font uses the Simplified Chinese ideographs for shared Unicode code points.

Tip

Coverage of the CJK font can be seen in `DroidSansFallbackFull.pdf`[8].

19.2.3. Unifont

GNU Unifont[9] is a Unicode font with a glyph for every visible Unicode Basic Multilingual Plane code point. The Unicode Basic Multilingual Plane covers the first 65,536 (or 2^16) Unicode code points. GNU Unifont originates

[4] https://www.unicode.org/charts/
[5] https://dejavu-fonts.org
[6] https://www.savapage.org/docs/fonts/DejaVuSans.pdf
[7] https://www.droidfonts.com/droidfonts
[8] https://www.savapage.org/docs/fonts/DroidSansFallbackFull.pdf
[9] https://savannah.gnu.org/projects/unifont

from a bitmap font with glyphs of 16 pixels high and either 8 or 16 pixels wide. Therefore it has a coarser look than native outline fonts.

Chapter 20. Customization

SavaPage can be customized to match your organizational identity. Customization makes SavaPage an integral part of your organization rather than an external tool.

> **Note**
>
> Customization is an advanced topic. If you need help, contact your SavaPage Service Provider.

20.1. Custom Web App

Web App customization is controlled in the `/opt/savapage/server/custom/web.properties` file. An annotated `web.properties.template` file is installed for your convenience.

> **Tip**
>
> Each key value in the `/opt/savapage/server/custom/web.properties` file can be overruled at runtime by specifying the key value in the Configuration Editor. When the configuration key value is left empty customization falls back to the value in the properties file.

20.1.1. Web App Look-and-feel

The default look-and-feel of Web Apps has a simple drawing of green hills with a blue sky as background. You can disable this default by setting the value of the `webapp.style.default.enable` config item to `N`. See Section 4.11.15, "Config Editor" [189] on how to change this value.

The look-and-feel of Web Apps can be customized by theming and CSS tailoring.

20.1.1.1. Web App Theming

SavaPage uses jQuery Mobile[1] as user interface system to create responsive Web Apps that are accessible on all smartphone, tablet and desktop devices. jQuery Mobile supports theming. Themes can be built online with the ThemeRoller for Mobile[2] tools and deployed in SavaPage by downloading the zipped theme file and extracting the content of the `/themes/` folder into the `/opt/savapage/server/custom/web/themes` directory.

The `web.properties` file contains entries to specify a separate CSS theme for each Web App, as shown in the example below:

```
webapp.theme.admin=admin.min.css                ❶
webapp.theme.jobtickets=jobtickets.min.css      ❷
webapp.theme.pos=admin.min.css                  ❸
webapp.theme.user=user.min.css                  ❹
```

[1] https://jquerymobile.com
[2] https://themeroller.jquerymobile.com

❶ CSS theme file name for the Admin Web App.
❷ CSS theme file name for the Job Tickets Web App.
❸ CSS theme file name for the POS Web App.
❹ CSS theme file name for the User Web App.

SavaPage uses swatch[3] “a” for all pages and dialogs. Swatch “b” is used for page and dialog headers, and in some cases for list dividers.

You can store a theme in a subdirectory of `/opt/savapage/server/custom/web/themes` and use its relative path to reference a CSS theme file.

20.1.1.2. Custom CSS

Advanced tailoring can be done with custom CSS files. They are rendered as last, so they have the final say about styling.

The `web.properties` file contains entries to specify a custom CSS file for each Web App, as illustrated in the example below:

```
webapp.custom.admin=admin.css              ❶
webapp.custom.jobtickets=jobtickets.css    ❷
webapp.custom.pos=pos.css                  ❸
webapp.custom.user=user.css                ❹
```

❶ Custom CSS file for the Admin Web App.
❷ Custom CSS file for the Job Tickets Web App.
❸ Custom CSS file for the POS Web App.
❹ Custom CSS file for the User Web App.

Custom CSS files are stored in `/opt/savapage/server/custom/web/`. Subdirectories are allowed, and you can use their relative path to reference the custom CSS file.

Any content placed in `/opt/savapage/server/custom/web/`, such as images, can be accessed in CSS via a URL beginning with `/custom/web/`. For example if a file named `logo.png` is placed in `/opt/savapage/server/custom/web/images` it can be accessed via the URL `/custom/web/images/logo.png`.

[3] A swatch is one of several colour schemes that can be provided by a jQuery Mobile theme. Single-letter designations are used for swatches. The default theme provides two swatches. The "a" swatch is a neutral, gray swatch, and the "b" swatch has a darker color scheme designed to contrast with the "a" swatch. Swatch "b" is used to draw special attention to certain elements in a user interface styled with "a".

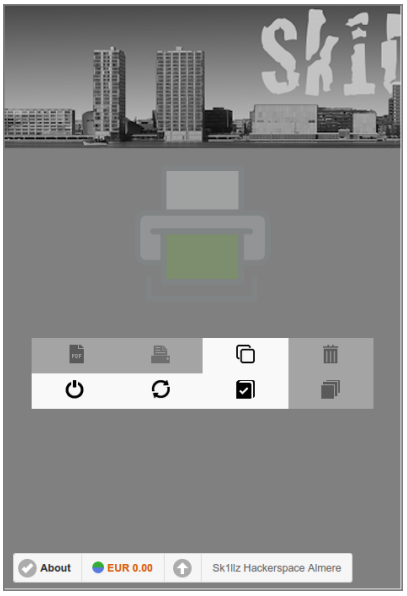


Figure 20.1. User Web App: Custom CSS - Sample #1

Figure 20.2. User Web App: Custom CSS - Sample #2

20.1.1.3. Custom HTML

Extra tailoring can be done with HTML snippet files to be injected into the Web App. Injection points are defined at the top of the Login and About Page for each Web App. Snippet files must be placed in the `/opt/savapage/server/custom/html/` directory. The default snippet is for the English locale. You can create i18n variants by appending the locale to the base file name. For example: `user-login_de.html` is the German variant of `user-login.html`.

Snippet files are assigned in the `web.properties` file, as shown below. Snippets must refer to the default English language variant. At runtime the locale variant (when applicable and available) is used.

```
webapp.html.admin.about=admin-about.html          ❶
webapp.html.admin.login=admin-login.html          ❷
```

```
webapp.html.jobtickets.about=jobtickets-about.html   ❸
webapp.html.jobtickets.login=jobtickets-login.html   ❹

webapp.html.mailtickets.about=mailtickets-about.html ❺
webapp.html.mailtickets.login=mailtickets-login.html ❻

webapp.html.payment.about=payment-about.html         ❼
webapp.html.payment.login=payment-login.html         ❽

webapp.html.pos.about=pos-about.html                 ❾
webapp.html.pos.login=pos-login.html                 ❿

webapp.html.printsite.about=printsite-about.html     ⓫
webapp.html.printsite.login=printsite-login.html     ⓬

webapp.html.user.about=user-about.html               ⓭
webapp.html.user.login=user-login.html               ⓮
```

❶ snippet for Admin About Dialog.
❷ snippet for Admin Login Page.
❸ snippet for Job Tickets About Page.
❹ snippet for Job Tickets Login Page.
❺ snippet for Mail Tickets About Page.
❻ snippet for Mail Tickets Login Page.
❼ snippet for Payment About Dialog.
❽ snippet for Payment Login Page.
❾ snippet for POS About Dialog.
❿ snippet for POS Login Page.
⓫ snippet for Print Site About Dialog.
⓬ snippet for Print Site Login Page.
⓭ snippet for User About Dialog.
⓮ snippet for User Login Page.

You can store custom HTML files in a subdirectory of `/opt/savapage/server/custom/html/` and use its relative path to reference the HTML file.

Note

The keys in the `web.properties` file are also present as Configuration Property, the latter taking precedence over the former. See Section 4.11.15, “Config Editor” [189] on how to enter these property values.

Important

Please use this template when creating snippets for Login pages. The CSS classes are needed to toggle visibility of sub-parts in different Login modes.

```
<h3 class="sp-login-dialog">Header when in Login Mode</h3>
<h3 class="sp-login-dialog-assoc">Header when in Card Self Association Mode</h3>
<div class="sp-login-dialog">
    <!-- Your custom HTML when in Login Mode -->
</div>
```

20.1.1.4. Custom i18n

In rare cases an application manager might want to override i18n text of Web App pages, dictionaries or messages.

Overrides of packaged i18n `*.xml` files are placed in `/opt/savapage/server/custom/i18n/`. Each override must be located in a subdirectory with a path identical to the original `*.xml` file. The content of the override file can be restricted to just the overridden keys.

Overrides of packaged `*.properties.xml` files (Web App HTML parts) must have a `*.xml` name **without** the "`.properties.`" part. For all other `*.xml` files the name can remain the same. The base name of English i18n files must end with "`_en`".

Example:

```
/opt/savapage/server/custom/i18n/org/savapage/core/i18n/PrintOutNounEnum_de.xml
/opt/savapage/server/custom/i18n/org/savapage/core/services/impl/messages_de.xml
/opt/savapage/server/custom/i18n/org/savapage/server/pages/Login_de.xml

/opt/savapage/server/custom/i18n/org/savapage/core/i18n/PrintOutNounEnum_en.xml
/opt/savapage/server/custom/i18n/org/savapage/core/services/impl/messages_en.xml
/opt/savapage/server/custom/i18n/org/savapage/server/pages/Login_en.xml
```

Custom i18n must be activated in server.properties with:

```
webapp.custom.i18n=true
```

Warning

Custom i18n is an advanced feature and should be implemented in consultation with SavaPage Tech Support.

Note

Added or changed custom i18n files have immediately effect when the i18n cache is cleared in the Admin Web App About : Java section, or when SavaPage is restarted.

Tip

Custom i18n is a great way to get started with localizing SavaPage to your own region. See Section 19.1.1, “Notes for Translators” [297].

20.1.1.5. Custom i18n for IPP

IPP attribute localization can be overridden with special i18n XML files. See Section L.3, “IPP Localization” [404].

20.2. Email Templates

Email templates are a powerful instrument to customize layout and content of email messages. Templates are defined as XML files, and are located in the `/opt/savapage/server/custom/template/` directory by default (alternative locations can be configured).

Default templates are present in the SavaPage i18n jars. Therefore, after a first-time installation the custom `template/` directory will be empty. However, when SavaPage finds a suitable i18n XML file in the custom directory, that belongs to an Email Message Type, it will use that template.

You can create i18n variants of template XML files by appending the locale to the base file name. For example: `template_de.html` is the German variant of `template.xml`.

20.2.1. Email Template Syntax

The base syntax of an Email Template[4] is described by example in the following XML file:

```
<?xml version= "1.0" encoding= "UTF-8" ?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd"> ❶

<properties>
    <!-- $obj.attr$ placeholders are replaced at runtime --> ❷

    <!-- TEXT --> ❸
    <entry key="text"><![CDATA[Text with $obj.attr$ placeholders.]]></entry>

    <!-- HTML --> ❹
    <entry key="html"><![CDATA[<b>HTML</b><p>
      with $obj.attr$ and other placeholders</p>]]>
    </entry>

</properties>
```

❶ DTD[5] of XML persisted `java.util.Properties` file.
❷ Placeholder object attributes are identified by a `$` character at the beginning and the end. Valid objects are defined for each Email Message Type. The global Application Object is valid for all types.
❸ Content for plain `text` message body.
❹ Content for `html` message body.

20.2.1.1. Embedded Images

HTML content may contain Embedded (Inline) Images.

Images are first defined in separated template entries with keys that have "`cid_`" prefix. Those keys can then be used as placeholders for `cid` scheme values. SavaPage email processor will embed the image file and replace `cid` placeholders with a unique Content-ID for the Embedded (Inline) Image.

Custom image files are referred to by their relative path from the custom `template/` directory. Internal stock images can simply be referred to by their upper-case identifier, as shown in the table below.

Object.attribute	Value
`SAVAPAGE_ICON`	SavaPage Icon: 32 x 32 pixels, 705 bytes.
`SAVAPAGE_LOGO`	SavaPage Logo with "SavaPage" text at the bottom: 148 x 174 pixels, 5.6 kB.

Table 20.1. Stock Image Identifiers

The XML snippet below shows how it works:

```
<entry key="cid_1">SAVAPAGE_ICON</entry>          ❶
<entry key="cid_2">images/mysite-logo.png</entry> ❷

<entry key="html">><![CDATA[
  <img src="cid:$cid_1$" alt="$app.name$">        ❸

  <a href="https://mysite.xyz">
    <img alt="mysite.xyz" src="cid:$cid_2$"/>     ❹
```

[4] Email Template syntax is based on StringTemplate [https://www.stringtemplate.org].
[5] http://java.sun.com/dtd/properties.dtd

```
  </a>

]]></entry>
```

❶ `cid_1` holds a stock image.
❷ `cid_2` holds the relative path to a custom image file.
❸ Note that the URL of the image has scheme `cid`. SavaPage email processor will embed the image and replace `$cid_1$` with a unique Content-ID for the Embedded (Inline) Image with key `cid_1`.
❹ The `cid_2` image is handled just as its `cid_1` sibling.

20.2.2. Email Stationary Template

An Email Stationary Template is used by an Email Message Template as container to embed its content in. A stationary is typically used by many Email Messages Types, and thus gives a common look-and-feel to various email output.

A Stationary Template contains the following placeholder objects:

- Stationary.
- Application.

And this is how it looks like:

```
<?xml version= "1.0" encoding= "UTF-8" ?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">

<properties>

    <!-- TEXT -->
    <entry key="text"><![CDATA[Plain text with $stationary.header$  ❶
$stationary.content$ and additional app placeholders]]></entry>     ❷

    <!-- HTML -->
    <entry key="html"><![CDATA[HTML with <b>$stationary.header$</b>
      <p>$stationary.content$ and additional cid_ and app placeholders</p>]]> ❸
    </entry>

</properties>
```

❶ `$stationary.header$` is replaced by the `header` entry of the embedded template.
❷ `$stationary.content$` is replaced by the `text` entry of the embedded template.
❸ `$stationary.content$` is replaced by the `html` entry of the embedded template.

> **Note**
>
> SavaPage uses its own stationary types. These types can be overwritten. See Section 20.2.5, “Email Stationary Types” [308].

20.2.3. Email Message Template

Email Message Templates are used by Email Message Types.

Template and Type are tied by name.

For example: the JobTicketCompleted message type will look for the nearest i18n version of a `JobTicketCompleted.xml` template file. In this way, `JobTicketCompleted_de.xml` will be the perfect match when a German email message is requested.

Email Message Template adds the entry key "`subject`" to the syntax. Optionally, entry keys "`stationary`" and "`header`" can be used to link to an Email Stationary Template.

The syntax is described by example in the following XML file:

```
<?xml version= "1.0" encoding= "UTF-8" ?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>

    <!-- Email subject -->        ❶
    <entry key="subject">Text with $obj.attr$ placeholders</entry>

    <!-- Optional stationary -->  ❷
    <entry key="stationary">EmailStationary</entry>
    <entry key="header">Stationary header with placeholders</entry>

    <!-- TEXT -->
    <entry key="text"><![CDATA[Plain text with placeholders.]]></entry>

    <!-- HTML -->
    <entry key="html"><![CDATA[<p>HTML with placeholders.</p>]]></entry>

</properties>
```

❶ The `subject` text of the email.

❷ The `stationary` value (just the template file basename, without the `.xml` suffix) refers to `EmailStationary.xml`, including i18n `EmailStationary_*.xml` variants. When a stationary is linked this way, the `header` value is injected into the `stationary.header` placeholder of the target stationary, and the `text` and `html` values are injected into the `stationary.content` placeholder of their counterpart stationary entry.

20.2.4. Email Placeholders Objects

Placeholder Objects are used in Email Stationary, and in Email Message Templates tied to Email Messages Types.

20.2.4.1. Stationary

Object uniquely used by Email Stationary Template.

Object.attribute	Value
`stationary.header`	The `header` entry of the embedded template.
`stationary.content`	The `text` or `html` entry of the embedded template.

Table 20.2. Placeholder: Stationary

20.2.4.2. Application

A global object with SavaPage application attributes. This object can be used in *any* Email Message Type context.

Object.attribute	Value
`app.name`	Application name: "SavaPage"
`app.nameVersion`	Formatted name and version, like: "SavaPage Major.Minor.Revision"
`app.nameVersionBuild`	Formatted name, version and build, like:

Object.attribute	Value
	"SavaPage Major.Minor.Revision (Build)"
`app.slogan`	Short slogan describing the application: "Open Print Portal"

Table 20.3. Placeholder: Application

20.2.4.3. Ticket

A Print Job Ticket. See Chapter 5, *Job Tickets Web App* [211].

Object.attribute	Value
`ticket.number`	Ticket number.
`ticket.name`	Document name.
`ticket.operator`	Name of the Job Ticket Operator.
`ticket.returnMessage`	The return message to Job Ticket Creator (optional).

Table 20.4. Placeholder: Ticket

20.2.4.4. User

A User as Person.

Object.attribute	Value
`user.fullName`	The full name of the user.

Table 20.5. Placeholder: User

20.2.4.5. User Registration

Object	Value
`user_reg.userName`	User Name.
`user_reg.fullName`	User Full Name.
`user_reg.link`	Verification URL.

Table 20.6. Placeholder: User Registration

20.2.5. Email Stationary Types

SavaPage uses its own Email Stationary Types. As the table below shows, currently there is just one type in use. You can use this type in your own custom Email Message Template files. Or, you can override this type by creating your own `EmailStationary.xml` i18n variants in the custom `template/` directory.

Name	Scope
`EmailStationary`	All Email messages

Table 20.7. Email Stationary Types

20.2.6. Email Message Types

The sections below describe email message types that can be customized. The list is limited for now, but will grow as more messages are refactored for this purpose.

> **Note**
>
> The global Application Object is valid for each message type.

20.2.6.1. JobTicketCanceled Email

Email sent when a Job Ticket is canceled. See Table 3.8, “Job Ticket Print Configuration Properties” [66].

The following placeholder objects apply:

Object	Role
`ticket`	Canceled Job Ticket.
`user`	Job Ticket Creator.

Table 20.8. Placeholder Objects: JobTicketCanceled

20.2.6.2. JobTicketCompleted Email

Email sent when a Job Ticket is completed successfully. See Table 3.8, “Job Ticket Print Configuration Properties” [66].

The following placeholder objects apply:

Object	Role
`ticket`	Completed Job Ticket.
`user`	Job Ticket Creator.

Table 20.9. Placeholder Objects: JobTicketCompleted

20.2.6.3. UserRegVerification

Email sent to verify User Self Registration. See Section 3.1.3, “Self Registration” [24].

The following placeholder objects apply:

Object	Role
`user_reg`	Self Registration.

Table 20.10. Placeholder Objects: UserRegVerification

20.2.7. Custom Template Locations

Custom templates are located in the `/opt/savapage/server/custom/template/` directory. Sub-locations can be defined for different template groups. These sub-locations can then be activated for different applications, with the following configuration properties:

Configuration property	Description
custom.template.home	Home location of *all* template files, relative to the default location. For example: a value of "`group-1`" will resolve to: `/opt/savapage/server/custom/template/group-1`
custom.template.home.mail	Home location of Email template files, relative to the default location. For example: a value of "`group-1/mail`" will resolve to: `/opt/savapage/server/custom/template/group-1/mail`

Table 20.11. Configuration Properties for Custom Template Locations

See Section 4.11.15, "Config Editor" [189] on how to set these items.

Chapter 21. Using an External Database

By default SavaPage is packaged with Apache Derby[1] as *internal* database. This gives you the opportunity to evaluate SavaPage on a small scale right away. However, in a production environment with multiple users, we strongly advise you to use PostgreSQL[2] as *external* database server.

Warning

Using the internal database in situations with multiple users and thus concurrent use, will inevitably lead to locking, deadlock and out-of-memory errors, which can make the system totally unresponsive. See our notes on Database Locking.

Other situations can be extra reason to choose for an external database, like:

- Organizational policy dictates that all applications must be consolidated on a single database infrastructure.
- You want to take advantage of existing maintenance and backup procedures that are present on your current database infrastructure.
- You want to use third party reporting tools to view and analyze the SavaPage database.
- You want optimal (tailored) performance, since SavaPage is intensively used by a very large user population. So, for example, you want to deploy a dedicated database server as a scalable solution.

This chapter describes how to connect and migrate to an external database. For database tuning, see Section 13.5.6, “Database Connection Settings” [252].

21.1. Supported Databases

SavaPage is able to use any database that has a JDBC driver available that is supported by Hibernate Dialects[3]. *However, PostgreSQL on GNU/Linux servers is our first choice*. PostgreSQL is designed to be highly scalable, is optimized for concurrent use, and handles datasets of any size efficiently. It behave robust with thousands of users and is very well suited for large scale SavaPage implementations. Above that, PostgreSQL is Free and Open Source software, and complies to Open Standards.

Note

Of course it is possible to connect SavaPage to your existing database infrastructure other than PostgreSQL. For example, an Oracle JDBC connection template is present in `server.properties`. Contact your SavaPage Service Provider to assist you with configuring alternative connections.

21.2. Migrating to an External Database

The migration is a simple process and takes about 15-30 minutes. The sections below describe in more detail the following high-level migration scenario steps:

[1] https://db.apache.org/derby/
[2] https://www.postgresql.org/
[3] Hibernate [https://hibernate.org/] is an object-relational mapping tool for the Java programming language. It provides a framework for mapping an object-oriented domain model to a relational database. Hibernate supports several SQL Dialects. Each dialect specifies the database type and is responsible for connecting to the database and generating the appropriate SQL statements.

1. Stop the server.
2. Create a backup of the current internal database.
3. Create and initialize a new external database.
4. Change connection parameters.
5. Restore the backup into the new external database.
6. Restart the server.

21.2.1. Step 1 - Stop SavaPage

The application server must be stopped in order to make a backup of the current internal database. The command to stop the server is described in Section C.6, “Stopping and Starting the Server” [356].

21.2.2. Step 2 - Create a Backup

Run the command to backup the current (internal) as described in Section C.5.6, “db-export and db-export-to” [355]. The command echoes the name of the created backup file to `stdout`. Take a note of this because you will need this in a future step.

21.2.3. Step 3 - Create new Database in External DBMS

Creating a new database is specific to the external Database Management System and is off-topic for this manual. It is assumed that the database administrator knows how to create a new database. However, the following generic requirements must be honored:

- Create a *dedicated* database user with a strong password to be used by SavaPage to connect to the database.
- Create the new empty database with a Unicode or UTF8 character encoding to make sure that all possible characters can be stored.
- Assign the dedicated user full access to the new database, i.e. grant permission to create and drop tables, and to execute select, insert, update and delete statements in all tables.

21.2.4. Step 4 - Change SavaPage Connection Parameters

Open a terminal session on the SavaPage server as user `savapage` and edit the `server.properties` file.

- Comment out the line `database.type=Internal` by adding a hash (#) character at the start of the line.
- Uncomment the `database.` connection parameter lines for the external database (in our case PostgreSQL).
- Set the `database.url`, which describes the location and connection details of the external database.

  The PostgreSQL URL format is: `jdbc:postgresql://[server]/[database]`

  The `[server]` parameter is the name of the server running the PostgreSQL database, and must be resolvable from the SavaPage server. If the PostgreSQL instance is running on the same machine then `localhost` can be used. The `[database]` parameter is the name of the PostgreSQL database you created in the previous step.
- Set the `database.user` and `database.password` used to connect to the database.

A connection example is shown below:

```
#-------------------------------------------------------------
# Database Settings
#-------------------------------------------------------------

# Using the internal database (default)
#database.type=Internal
```

```
# PostgreSQL connection
database.type=PostgreSQL
database.driver=org.postgresql.Driver
database.url=jdbc:postgresql://localhost/savapage
database.user=your-db-user
database.password=your-db-user-password
```

21.2.5. Step 5 - Initialize new Database

This step creates the required database tables and initial data. Open a terminal session on the SavaPage server as user `savapage` and run the command as described in Section C.5.8, “db-init” [356].

21.2.6. Step 6 - Restore Backup into new Database

This step restores the backup file exported in one of the previous steps, into the newly initialized database. Open a terminal session on the SavaPage server as user `savapage` and run the command as described in Section C.5.7, “db-import” [355].

21.2.7. Step 7 - Restart SavaPage

At this point the data have been migrated to the new database and the server can be restarted. See Section C.6, “Stopping and Starting the Server” [356].

Wait a couple of seconds before logging in to the Admin Web App to verify that the migration worked successfully.

Chapter 22. Tuning

22.1. Linux Kernel Parameters

GNU/Linux distributions are generally not configured to run more demanding server processes out-of-the-box. So, running SavaPage with high load on a vanilla GNU/Linux OS can easily result in a degraded performance.

Performance bottlenecks are usually due to OS, TCP stack and network settings meant for *desktop* user sessions, and not for server processes that are intensively used by many network clients. Fortunately, it is easy to unleash the full potential of your GNU/Linux host with a few simple tweaks. The message is that SavaPage scales perfectly if you apply the right kernel settings.

Relevant kernel parameters and settings are discussed in the next sections. The last section summarizes the suggested settings and describes how to apply them. See Section 22.1.5, “Setting Linux kernel parameters with sysctl” [315].

> **Note**
>
> Kernel parameters with `ipv4` in their names also apply to TCP over IPv6.

22.1.1. IP Ports

As many outgoing connections are concurrently established from SavaPage, we must make sure Linux does not run low on *ephemeral* local ports[1] and reuse sockets with state `TIME_WAIT`.

```
net.ipv4.ip_local_port_range = 1024 65535   ❶
net.ipv4.tcp_tw_reuse = 1                   ❷
```

❶ Broaden the ephemeral local port range.

❷ Enable the reuse of sockets with state `TIME_WAIT`. This is particularly useful in environments where numerous short connections are open and left in TIME_WAIT state, such as in SavaPage.

22.1.2. TCP Buffer Sizes

Linux does a good job of auto-tuning the TCP buffers, but the default maximum sizes are still very small. Here are sample settings for 1Gb and 10Gb network.

```
# Settings for 1Gb network (16Mb buffer)
net.core.rmem_max = 16777216                ❶
net.core.wmem_max = 16777216                ❷
net.ipv4.tcp_rmem = 4096 87380 16777216     ❸
net.ipv4.tcp_wmem = 4096 16384 16777216     ❹

# Settings for 10Gb network (32Mb buffer)
```

[1] An established TCP/IP connection can be regarded as a 4-tuple (server IP, server port, client IP, client port). Three of the four are evident, i.e. the client uses its own IP address to connect to the server's IP address and service port. However, the connection also needs a port number at the client side. Unless the client program explicitly requests a port number, this port number is called an *ephemeral* port number. Ephemeral ports are temporary issued by the IP stack of the client OS from a dedicated port range.

```
net.core.rmem_max = 33554432
net.core.wmem_max = 33554432
net.ipv4.tcp_rmem = 4096 87380 33554432
net.ipv4.tcp_wmem = 4096 16384 33554432

# Settings for 10Gb network (54Mb buffer)
net.core.rmem_max = 56623104
net.core.wmem_max = 56623104
net.ipv4.tcp_rmem = 4096 87380 56623104
net.ipv4.tcp_wmem = 4096 16384 56623104
```

❶ Max size (bytes) of the TCP receive buffer as settable with setsockopt.
❷ Max size (bytes) of the TCP send buffer as settable with setsockopt.
❸ Auto-tuning limits (bytes) for TCP receive buffer: min, default, and max number of bytes.
❹ Auto-tuning limits (bytes) for TCP send buffer: min, default, and max number of bytes.

22.1.3. Queue Sizes

While a socket is listening and busy, new connection requests will pile up. The kernel keeps pending connection requests in a buffer. When the buffer is full new requests will fail. You can increase several buffer sizes.

```
net.core.somaxconn = 4096                ❶
net.core.netdev_max_backlog = 16384      ❷
net.ipv4.tcp_max_syn_backlog = 8192      ❸
net.ipv4.tcp_syncookies = 1              ❹
```

❶ Max number of queued connections on a socket. The default of `128` is too low: we raise this value substantially to support bursts of request.
❷ Max number of packets, queued on the input side, when the interface receives packets faster than the kernel can process them.
❸ Max number half open SYN requests to keep in memory.
❹ Enable SYN cookies[2] to harden the TCP/IP stack against SYN floods.

22.1.4. Congestion Control

Congestion refers to a network state where a node or link carries so much data that it may deteriorate network service quality, resulting in queuing delay, frame or data packet loss and the blocking of new connections.

In a congested network, response time slows with reduced network throughput. Congestion occurs when bandwidth is insufficient and network data traffic exceeds capacity.

Linux supports pluggable congestion control (avoidance) algorithms. To get a list of congestion control algorithms that are available in your kernel run the command:

```
sudo sysctl net.ipv4.tcp_available_congestion_control
```

If `cubic` and/or `htcp` are not listed then you will need to research the control algorithms for your kernel. If available set the control to cubic:

```
net.ipv4.tcp_congestion_control = cubic
```

22.1.5. Setting Linux kernel parameters with sysctl

Edit the file `/etc/sysctl.conf` like this:

[2] https://en.wikipedia.org/wiki/SYN_cookies

```
sudo vi /etc/sysctl.conf
```

and add the following lines, that summarize the previously discussed kernel parameters, at the end of the file:

```
#-----------------------------------------------------
# SavaPage Settings for 1Gb network
#-----------------------------------------------------
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 16384 16777216
net.core.somaxconn = 4096
net.core.netdev_max_backlog = 16384
net.ipv4.tcp_max_syn_backlog = 8192
net.ipv4.tcp_syncookies = 1
net.ipv4.ip_local_port_range = 1024 65535
net.ipv4.tcp_tw_recycle = 0
net.ipv4.tcp_tw_reuse = 1

# Only if cubic is available
net.ipv4.tcp_congestion_control = cubic
```

You can apply the settings without rebooting the server with this command:

```
sudo sysctl -p
```

22.2. Linux User Limits

SavaPage server may run out of file descriptors as the system defaults are normally very low. A file descriptor (FD) is a handle created by a process when a file is opened. Each process can use a limited number of FDs as specified per user in an OS level user limit.

Beware that apart from “regular” files that are accessed by SavaPage from disk, each incoming request that uses a TCP socket also consumes one file descriptor from the total available for the process.

22.2.1. SysVinit User Limits

On Debian based systems the number of process FDs for the `savapage` user can be increased as follows.

Edit the file Edit `/etc/security/limits.conf` like this:

```
sudo vi /etc/security/limits.conf
```

and add the following lines at the end of the file:

```
savapage    hard    nofile    65536
savapage    soft    nofile    65536
```

Next, open `/etc/pam.d/su` like this:

```
sudo vi /etc/pam.d/su
```

and uncomment the following line:

```
session    required   pam_limits.so
```

You also need to edit the `/etc/pam.d/common-session` and `/etc/pam.d/common-session-noninteractive` files. Open the files like this:

```
sudo vi /etc/pam.d/common-session
```

```
sudo vi /etc/pam.d/common-session-noninteractive
```

and for each file add the following line to the end:

```
session required pam_limits.so
```

Finally, check whether the settings are applied with this command:

```
sudo su - savapage -c "ulimit -n"
```

This should output the value `65536`.

22.2.2. Systemd User Limits

Systemd ignores **ulimit** values as described in the previous section, and has its own equivalent that can be used per service. The best way to set user limits for the `savapage.service` is by an *override*, that will have the last say over any SavaPage default.

Check the `/lib/systemd/system/savapage.service` unit to see if the value of the `LimitNOFILE` directive is to your liking. This directive corresponds to "`ulimit -n`" (open files) and defaults to `65536`.

```
cat /lib/systemd/system/savapage.service | grep LimitNOFILE
```

If this directive is absent, or if you want to override the value, edit the service unit with this command:

```
sudo systemctl edit savapage
```

This launches a text editor for creating the file:

```
/etc/systemd/system/savapage.service.d/override.conf
```

Add the following lines, using your own choice (e.g. 98304):

```
[Service]
LimitNOFILE=98304
```

Save the file and close the editor. Usually, after you edited a systemd unit file, for it to take effect, you need to run:

```
sudo systemctl daemon-reload
```

However, the `systemctl edit` command automatically did this for you. You can check the effect of the override with this command:

```
systemctl cat savapage.service | grep LimitNOFILE
```

... the last line should show:

```
LimitNOFILE=98304
```

Restart SavaPage for the changes to take effect, and check if the override has effect with this command:

```
systemctl status savapage.service
```

Notice the Drop-In override.conf as shown below:

```
savapage.service - SavaPage Open Print Portal
   Loaded: loaded (/lib/systemd/system/savapage.service; enabled; vendor preset: enabled)
  Drop-In: /etc/systemd/system/savapage.service.d
           |- override.conf
   Active: active (running) since ...
```

> **Important**
>
> Check if the “open files” override value is shown correctly in the Host System section of the Admin Web App, since this is the actual value that the SavaPage server process uses.

22.3. JVM Tuning

SavaPage runs in the Java Virtual Machine (JVM) using the class libraries and other supporting files provided in the JRE.

The SavaPage JVM settings work fine, and generally there is no customization needed. However, if needed the JVM can be tuned by adding extra JVM arguments in the file:

`/opt/savapage/server/custom/app-server.conf`

Edit this file as `savapage` user and enter the extra JVM arguments as value of the `CUSTOM_JVM_ARGS` key. The examples below shows the JVM arguments as explained in the next sections.

```
# JDK 11
# Note: enclose the value with quotes
CUSTOM_JVM_ARGS="-XX:MaxRAMFraction=2"
```

The location of temporary files can be overwritten with the `JAVA_IO_TMPDIR` key. See Section 22.3.3, “JVM Temporary Files” [319].

> **Important**
>
> Before doing any JVM customizing please consult SavaPage Support to discuss your requirements and which customization fits best.

22.3.1. JVM Memory Allocation

The JVM allocates a quarter of host system RAM to the SavaPage Server process by default. This ensures that SavaPage does not consume too many resources and does not get in the way of other applications running on the same system.

However, if the host system is dedicated to running SavaPage, you can safely allocate more memory to SavaPage. With more allocated memory SavaPage will have a better performance, particularly with many users and large printing throughput.

Add one of the following JVM parameters to allocate relative or absolute memory:

```
-XX:MaxRAMFraction=3          ❶
-XX:MaxRAMFraction=2          ❷
-Xmx864m                      ❸
```

❶ Allocate one third of host system RAM.
❷ Allocate one half of host system RAM.
❸ Allocate 864MB of host system RAM.

22.3.2. JVM Garbage Collection

Customizing Java Garbage Collection (GC) depends on the characteristics of the application involved. The JVM provides proper defaults for SavaPage most of the time. See OpenJDK Wiki[3] for an introduction to GC options.

[3] https://wiki.openjdk.org/

22.3.3. JVM Temporary Files

The Java system property `java.io.tmpdir` determines where the JVM writes temporary files. The default value typically points to the world readable `/tmp` directory. As a result, all Java components that are part of the SavaPage application will write their temporary files to that directory. You can override the default in:

`/opt/savapage/server/custom/app-server.conf`

Edit this file as `savapage` user and specify the alternative temp directory at the `JAVA_IO_TMPDIR` variable.

```
# Overwrite of JVM system property 'java.io.tmpdir'
# User 'savapage' must have mode 700 access to this directory.
#JAVA_IO_TMPDIR=
```

Caution

Use the `JAVA_IO_TMPDIR` setting with the *utmost care*. Make sure the directory is *exclusively used* by the SavaPage application, and is accessible by the `savapage` system user only. If these conditions are not met, SavaPage will not start and might get corrupted.

Among all Java third-party components, SavaPage components take a special position because they write their own temp files in `/opt/savapage/server/data/.tmp` . This directory is created when the application starts and removed when stopped. You can override this special location with the `app.dir.tmp` property in the `server.properties` file. See Section 13.5.13, “Alternative File Locations” [256].

Caution

Make sure that the special temp directory used by SavaPage resides on the same disk partition as the other file locations that are used to store SavaPage data on runtime. See:

- Section 13.5.13, “Alternative File Locations” [256].
- Appendix F, *File Locations* [365].

22.4. Server Thread Pooling

A high reliability server process like SavaPage must reject excess requests *immediately* (fail fast) by using a request queue with a *bounded capacity*. Requests are FIFO processed by threads in a ThreadPool. The maximum number of threads needed, in order to achieve the best performance, depends on host resources (RAM and CPU cores) assigned to SavaPage. See Section 13.5.3.2, “Server ThreadPool Settings” [249] on how to configure the relevant parameters.

22.5. Database Connection Pooling

Database connections are both expensive to create and maintain over time. Therefore, they are an ideal resource to pool. That is exactly what SavaPage does. See Section 13.5.6, “Database Connection Settings” [252] on how to configure the relevant parameters.

Chapter 23. SavaPage Community

You can actively join the SavaPage Community by purchasing support from an authorized SavaPage Service Provider (SSP). You will get a SavaPage Service Card file in return. This file holds your Member name and is your token as resident of the SavaPage Community. It can be uploaded to show your status in the SavaPage Software. Members have the right to request new features and are entitled to high-priority Technical Support.

The Service Card has an expiration date. This date is displayed on the Admin Web App Dashboard. Exceeding this date serves as a reminder to extend the service period. The absence of a Service Card or its expiration does not affect the software's functionality in any way.

Whether you have a Service Card or not, you must always enter the name of your organization. See Section 4.14.5, “Organization” [201].

The community status is shown on the Admin Web App Dashboard and the Support section of the About page.

23.1. Importing the Service Card

The Service Card is issued by a SavaPage Service Provider as a digitally signed file. Installing the file into the application confirms your community status. To install the file:

1. Log into the SavaPage Admin Web App and navigate to the `About` page.
2. Scroll down to the `Support` section and click the `Import Service Card` button.
3. See Figure 4.168, “Admin Web App: About - Import Service Card” [202] how to proceed in the import dialog.
4. Verify that your Membership is correctly listed in the About page.

If you have a question about your Service Card or need assistance please contact your SavaPage Service Provider.

Note

The file supplied is simply a digitally signed and zipped text file containing your Membership information. It's converted to ZIP format to minimize size. If you're interested in viewing the contents of the file, rename the file to `.zip` and simply open it in any ZIP extraction utility.

Appendix A. Proxy Print Scenarios

This chapter summarizes several Proxy Print scenarios in a shorthand catalogue. Please follow the hyperlinks in the summaries for more detail.

A.1. Proxy Print Costs

If a Proxy Printer is exclusively managed by SavaPage, the costs of its print jobs are immediately charged after a job is submitted. This is done with full confidence that the jobs will reach status “Completed”. In this way account balances are up to date in advance and the corresponding credit limits can be applied in a timely manner.

Note

In the exceptional case that a print job is Canceled or Aborted (by manual intervention in CUPS) the costs can be refunded in the Documents section of the Admin Web App.

If the Proxy Printer CUPS queue is managed by a *Third Party Print Management System* (TPPMS) like PaperCut, costs will not be charged until the TPPMS notifies that the print job is Completed. In this way we exempt ourselves from third party print policy and only charge cost when we are absolutely sure that the print was successful. We accept the inherent latency of an updated account balance in SavaPage which might introduce a small chance for unjustified prints in between (according to the related credit limit).

A.2. Personal Print Scenarios

A.2.1. Personal Print - Non-Secure Scenarios

See Section 4.11.9, “Proxy Print” [168].

A.2.2. Personal Print - Secure Scenarios

See Section 4.9.1.2, “Proxy Print Authentication” [142].

A.2.2.1. Direct Print Scenario

See Section 4.9.1.2.3, “Direct Print Mode” [145].

A.2.2.2. Hold Print Scenario

See Section 4.9.1.2.2, “Hold Print Mode” [145].

A.2.2.3. Fast Print Scenario

See Section 4.9.1.2.1, “Fast Print Mode” [144].

A.2.2.4. Automatic Print Scenario

See Section 4.7.3, “IPP Routing” [124].

A.2.3. Personal Print - PaperCut Scenario

Personal Print to PaperCut	
User Story	*"As a **Print Job Creator**, I can create a Personal Print, so that PaperCut printing costs are charged to my personal account in SavaPage."* *"As an **Operator**, I can release or cancel a pending Print Job in PaperCut or CUPS."*
	*"As a **Job Ticket Creator**, I can create a Personal Print."* This User Story is identical to the Personal Print track as described in Section A.3.2, "Delegated Print - Job Ticket - PaperCut - Scenario" [324], and is not discussed here.
Preconditions	• PaperCut Integration is *enabled*. • Delegated Print is *disabled*. • Personal Print with PaperCut is *enabled*. • Proxy Printer used for printing is managed by PaperCut.

Process	1. **Print Job Creator** creates Personal Print Job.
	2. **Print Job Creator** prints job to PaperCut Managed printer.
	3. Print-out document for Print Job Creator is created with status **Pending (external)**
	  • A Shadow SavaPage Personal Account Transaction is created and linked.
	  • Transactions are displayed as **Printer Usage** with zero amount and status **Pending (external)**.
	  • Transactions Summary without cost data is displayed with the Print-out document.
	4. PaperCut print status is ***monitored*** till end state is reached.
	  • Status is monitored for a maximum number of minutes, after which the Print-out document gets status **Error**. See Section N.4.1, "PaperCut Print Log Monitoring" [417].

Scenario 1	*Print **Completed**.*
	
	Given the printer is a PaperCut Hold/Release queue *And* the **Operator** or **Print Job Creator** Released the job (in PaperCut) , *Or* the job was printed directly, *Then*:
	
	• Print-out document gets status **Completed**.
	• *PaperCut Printing Cost* is used for the Shadow SavaPage Personal Account Transaction.
	  • The transactions are displayed as **Printer Usage** with the PaperCut amount and status **Completed**.
	  • A Transaction Summary *with* cost data is now displayed with the Print-out document.

Scenario 2	*Print is **Canceled** or **Timed out**.*
	
	Given the printer is a PaperCut Hold/Release queue *And* the **Operator** or **Print Job Creator** Canceled the job (in PaperCut or CUPS) or the job Expired (in PaperCut) , *Then*:
	
	• Print-out document gets status **Canceled**.
	  • The Shadow Transaction *continues* to be displayed as **Printer Usage** with zero amount, but now has status **Canceled**.
	  • A Transaction Summary without cost data *continues* to be displayed at the Print-in document.

References	• Section N.3, "Personal Print to PaperCut" [416].

Table A.1. Personal Print - PaperCut Scenario

A.3. Delegated Print Scenarios

The scenarios in this section elaborate situations where users act as Print Job Delegate for other users (Delegators) or entities (Shared Accounts).

Note

A Personal Job Ticket Print is handled as Delegated Print, where the Job Ticket Creator has role Delegate and Delegator at the same time.

A.3.1. Delegated Print - (Non) Secure & Job Ticket Scenarios

Delegated Print	
User Story	*"As a **Delegate**, I can print for other users (delegators) and entities, so that costs are charged to individual delegators, delegator groups and shared accounts."* *"As a Print Job Creator, I can act as **Delegate** for another entity, so that costs are charged to a shared account."* *"As a Job Ticket Creator, I can create a Personal Print, and act as **Delegate** and Delegator at the same time."* *"As a **Job Ticket Operator**, I can release or cancel the Job Ticket."* *"As an **Operator**, I can cancel a pending Print Job in CUPS."*
Preconditions	• Delegated Print is *enabled*. • Delegated Print with PaperCut is *disabled*. • Job is printed as *Job Ticket*, or in *Non-secure* or *Authenticated* (Hold or Direct) Print Mode.
Process	1. **Delegate** creates Delegated Print Job. 2. **Delegate** prints the job.

Scenario 1	***Non-secure** Print*
	
	Given the Delegate printed to a non-secure Proxy Printer, *Then*:
	
	• The job is send to the printer.
	• Print-out document for Delegate is created.
	  • SavaPage Account Transactions are created and linked.
	  • Transactions are for personal delegators, delegator groups, and shared accounts involved.
	  • Printing costs are pro-rata divided over the Account Transactions.
	  • Transactions are displayed as **Printer Usage** with the pro-rata amount.
	  • Transactions Summary with cost data is displayed with the Print-out document.
	  • The actual print status from "Pending", "Processing" ("Held", "Stopped") to end state "Completed", "Canceled" or "Aborted" is displayed in the Print-out document and each transaction involved.

Scenario 2	***Non-secure** Print Canceled or Aborted* *Given* the Delegate printed to a non-secure Proxy Printer, *And* the print job is canceled or aborted by an Operator in CUPS, *Then*: • Print status "Canceled" or "Aborted" is displayed in the Print-out document and each transaction involved. • A `Reverse` button is available for the Print-out document in the Admin Web App for administrators with Editor permission for Accounts to reverse print counters and cost transactions as created in "Scenario 1".
Scenario 3	***Authenticated** Print Release*. *Given* the Delegate printed to a Hold or Direct Mode Proxy Printer, *And* released the job by NFC Authentication, *Then*: • The Print-out document is created as in "Scenario 1".
Scenario 4	***Job Ticket** Released*. *Given* the Delegate printed a Job Ticket, *And* the ticket is released by a Job Ticket Operator, *Then*: • The Print-out document is created as in "Scenario 1", using *Job Ticket Cost* as printing cost.
Scenario 5	***Job Ticket** Canceled*. *Given* the Delegate printed a Job Ticket, *And* the ticket is canceled by the Delegate or an Operator (in SavaPage or CUPS), *Then*: • The Job Ticket is just removed.
References	• Section 3.5.4, "Delegated Print" [53]. • Section 4.9.1.2, "Proxy Print Authentication" [142].

Delegated Print	
	• Section 3.5.9, “Job Ticket Print” [64].

Table A.2. Delegated Print - (Non) Secure & Job Ticket Scenarios

A.3.2. Delegated Print - Job Ticket - PaperCut - Scenario

Delegated Print via Job Ticket to PaperCut	
User Story	*“As a **Delegate**, I can create a SavaPage Job Ticket for other users (delegators) and entities, so that costs are charged to individual delegators, delegator groups and entities in PaperCut.”* *“As a Job Ticket Creator, I can create a Personal Print, and act as **Delegate** and Delegator at the same time.”* *“As a **Job Ticket Operator**, I can release or cancel the Job Ticket.”* *“As an **Operator**, I can release or cancel a pending Print Job in PaperCut or CUPS.”*
Preconditions	• PaperCut Integration is *enabled*. • Delegated Print with PaperCut is *enabled*. *Charging costs to PaperCut entities is unconditional. Balance and overdraft preconditions are not checked for restricted PaperCut accounts. This can result in an overdraft beyond what is allowed by PaperCut printing rules.*

Process	1. **Delegate** creates Delegated Print Job.
	2. **Delegate** prints job as *Job Ticket*.
	3. **Job Ticket Operator** releases the ticket to a PaperCut managed printer.
	4. Print-out document for Delegate is created with status **Pending (external)**
	  • Shadow SavaPage Account Transactions are created and linked.
	  • Transactions are for personal delegators, delegator groups, and shared accounts involved.
	  • Transactions act as template for transactions created in PaperCut after the print is completed.
	  • Transactions are displayed as **Printer Usage** with zero amount and status **Pending (external)**.
	  • Transactions Summary without cost data is displayed with the Print-out document.
	5. PaperCut print status is ***monitored*** till end state is reached.
	  • Status is monitored for a maximum number of minutes, after which the Print-out document gets status **Error**, and the ticket can be closed. See Section N.4.1, “PaperCut Print Log Monitoring” [417].

Scenario 1	*Print **Completed**.*
	
	Given the printer is a PaperCut Hold/Release queue *And* the **Operator** Released the job (in PaperCut), *Or* the job was printed directly, *Then*:
	
	• Print-out document gets status **Completed**.
	• *SavaPage Job Ticket Cost* is pro-rata divided over the Shadow SavaPage Account Transactions.
	  • Transactions are displayed as **Printer Usage** with the pro-rata amount and status **Completed**.
	  • Transactions Summary *with* cost data is now displayed with the Print-out document.
	• PaperCut Account Transactions are created according to the Shadow Accounts template.

Scenario 2	*Print is **Canceled** or **Timed out**.*
	
	Given the printer is a PaperCut Hold/Release queue *And* the **Operator** Canceled the job (in PaperCut or CUPS) or the job Expired (in PaperCut), *Then*:
	
	• Print-out document gets status **Canceled**.
	  • Transactions *continue* to be displayed as **Printer Usage** with zero amount but have status **Canceled**.
	  • Transactions Summary without cost data *continues* to be displayed at the Print-in document.

References	• Section 3.5.4, “Delegated Print” [53]. • Section 3.5.9, “Job Ticket Print” [64]. • Section N.1, “Delegated Print to PaperCut” [412].

Table A.3. Delegated Print - Job Ticket - PaperCut Scenario

A.3.3. Delegated Print - PaperCut Scenario

Delegated Print to PaperCut	
User Story	*"As a **Delegate**, I can print for other users (delegators) and entities, so that costs are charged to individual delegators, delegator groups and entities in PaperCut."* *"As a Print Job Creator, I can act as **Delegate** for another entity, so that costs are charged to this entity in PaperCut."* *"As a Print Job Creator, I can create a Personal Print, and act as **Delegate** and Delegator at the same time."* *"As an **Operator**, I can release or cancel a pending Print Job in PaperCut or CUPS."*
Preconditions	• PaperCut Integration is *enabled*. • Delegated Print with PaperCut is *enabled*. • Proxy Printer used for printing is managed by PaperCut. • Proxy Printer used for printing is *non-secure*. • If Hold/Release printing is required it must be handled in PaperCut.

Process	1. **Delegate** creates Delegated Print Job.
	2. **Delegate** prints job to PaperCut Managed printer.
	3. Print-out document for Delegate is created with status **Pending (external)**
	  • Shadow SavaPage Account Transactions are created and linked.
	  • Transactions are for personal delegators, delegator groups and entities involved.
	  • Transactions act as template for transactions created in PaperCut after the print is completed.
	  • Transactions are displayed as **Printer Usage** with zero amount and status **Pending (external)**.
	  • Transactions Summary without cost data is displayed with the Print-out document.
	4. PaperCut print status is ***monitored*** till end state is reached.
	  • Status is monitored for a maximum number of minutes, after which the Print-out document gets status **Error**, and the ticket can be closed. See Section N.4.1, "PaperCut Print Log Monitoring" [417].

Scenario 1	*Print **Completed**.*
	
	Given the printer is a PaperCut Hold/Release queue *And* the **Operator** Released the job (in PaperCut) , *Or* the job was printed directly, *Then*:
	
	• Print-out document gets status **Completed**.
	• *PaperCut Printing Cost* is pro-rata divided over the Shadow SavaPage Account Transactions.
	  • Transactions are displayed as **Printer Usage** with the pro-rata amount and status **Completed**.
	  • Transactions Summary *with* cost data is now displayed with the Print-out document.
	• PaperCut Account Transactions are created according to the Shadow Accounts template.

Scenario 2	*Print is **Canceled** or **Timed out**.*
	
	Given the printer is a PaperCut Hold/Release queue *And* the **Operator** Canceled the job (in PaperCut or CUPS) or the job Expired (in PaperCut) , *Then*:
	
	• Print-out document gets status **Canceled**.
	  • Transactions *continue* to be displayed as **Printer Usage** with zero amount but have status **Canceled**.
	  • Transactions Summary without cost data *continues* to be displayed at the Print-in document.

References	• Section 3.5.4, "Delegated Print" [53]. • Section N.1, "Delegated Print to PaperCut" [412].

Table A.4. Delegated Print - PaperCut Scenario

Appendix B. NFC Authentication

SavaPage supports Radio Frequency Identification (RFID) as authentication method.

RFID is the technology for uniquely identifying items using radio waves. A basic RFID system comprises a *passive tag*[1], a *reader*, and an *antenna*, where the reader sends an interrogating signal to the tag via the antenna, and the tag responds with its Unique Identification (UID).

In this way RFID tags are commonly used as authentication token: the RFID reader connected to the authenticator just passes the UID (Card Number) of the tag. Applications are abundant, ranging from tags embedded into retail products to help stores keep tabs on inventory, to tags embedded into animals to keep track of life stock. RFID is also applied in passports and credit cards, as well as identification badges that let employees access secure areas.

Near Field Communication (NFC) is a more recent, finely honed version of RFID with a much broader application. While RFID is a one-way communication system only, with data flowing from tag to reader, NFC can also be set up for two-way communication. However, NFC operates at a maximum range of about 4 inches (10 centimeters) and uses High Frequency (HF) RFID readers at 13.56 MHz.

Since SavaPage is targeted at the same HF RFID readers and tags, albeit in one-way communication, this manual uses the more common terms *NFC Card* and *NFC Reader* for the tag and reader role. In some contexts the terms *Card* and *Card Reader* will be used as shorthand.

SavaPage supports two Card Reader types.

- A *Local* Card Reader: a keyboard emulating device that “types” the UID (Card Number) each time a Card is swiped.
- A *Network* Card Reader: a software component, implemented on a dedicated device (like a Raspberry Pi®), that interacts with an NFC Reader after a card swipe and sends the UID to the central SavaPage server.

B.1. Card Number Format

SavaPage stores the Card Number (UID) in lower case HEX format, with Least Significant Byte (LSB) first. So, at the interfaces where the UID is captured, the output format and byte order must be specified as HEX or DECIMAL and LSB or MSB (Least or Most Significant Byte) first. This information is used by SavaPage to convert the captured Card Number to its internal HEX/LSB standard.

B.2. Local Card Reader

A *Local Card Reader* is an NFC Reader that functions as USB Keyboard Emulator. At each card swipe the reader must react by “typing” the card's UID (Card Number) appended by a Carriage Return (CR). SavaPage makes use of this function by capturing[2] the keystrokes at Login time in the Web App.

[1] RFID tags are either Active or Passive. Active tags have their own power supply by which they can broadcast with a read range of up to 100 meters. Passive tags do not have their own power source. Instead, they are powered by the electromagnetic energy transmitted from the RFID reader. Because the radio waves must be strong enough to power the tags, passive RFID tags have a read range from near contact to a few meters.

[2] SavaPage uses a short time limit to capture the keystrokes from a Local Card Reader. The time limit (milliseconds) is contained in the configuration key `webapp.card-local.keystrokes-max-msecs`. Do not change this value, except when requested by the SavaPage support desk.

Note

The way a reader formats the UID can deviate from the SavaPage HEX/LSB standard. Therefore you need to specify the format at the interfaces where the reader's UID is used. Most keyboard emulating readers can be configured to a specific output format and byte order.

Tip

At the time of this writing StrongLink[3] sells a reliable Plug and Play USB Keyboard Emulating Card Reader (SL040A) for a competitive price. The reader supports UID reads for Mifare Mini, Mifare 1k, Mifare 4k, Mifare Plus, Ultralight, DesFire and Mifare_ProX cards.

B.3. Network Card Reader Service

SavaPage ships with a Network Card Reader Service to be deployed on a Raspberry Pi®. The service works with CCID[4] compatible NFC readers. The `ACS ACR122U`[5] and `ACR1252U`[6] readers are popular choices, but other CCID readers work just as well [7]. The installation instructions can be found in:

```
/opt/savapage/providers/nfc/linux-armv6/README
```

Any deployed service must be entered as SavaPage Device. See Section 4.9.1, “Network Card Reader” [141]. At each card swipe the UID of the card is read and send to the central SavaPage server, where it is handled in context of the device definition.

You can link sounds and scrips to various events. Sample files are provided for your own customization, for example to communicate with PiGlow[8], Pibrella[9] or PiFace Control & Display[10] add-on boards.

[3] http://www.stronglink-rfid.com
[4] https://en.wikipedia.org/wiki/CCID_(protocol)
[5] https://www.acs.com.hk/en/products/3/acr122u-usb-nfc-reader/
[6] https://www.acs.com.hk/en/products/342/acr1252u-usb-nfc-reader-iii-nfc-forum-certified-reader/
[7] A list of compatible NFC readers is available at CCID free software driver [https://ccid.apdu.fr/].
[8] https://www.pimoroni.com/
[9] http://pibrella.com/
[10] http://www.piface.org.uk/

Appendix C. Tools

C.1. Server Commands

The **savapage-cmd** tool provides a command-line interface to SavaPage Server methods. It can directly be executed on the command-line or be part of more elaborate shell scripts.

For security reasons only users with read access to the `server.properties` file have the right to execute the command. So, the sure way to go is ...

```
sudo su - savapage
cd server/bin/linux-x64
```

... and to execute `savapage-cmd` from here, and ...

```
./savapage-cmd --help
```

... will echo all methods available:

```
_______________________________
SavaPage Command Line Interface

Note: use METHOD --help for method details.

usage: [METHOD] [OPTION]...

    --add-internal-user                Creates a new or Updates an existing Internal
                                       User.
    --add-user-group                   Adds a user group from the external user
                                       source: synchronized external users belonging
                                       to this group are added as member.
    --change-base-currency             Changes the base currency of the application.
    --delete-user                      Logically deletes a User.
    --delete-user-group                Deletes a user group.
    --delete-user-group-account        Logically deletes a User Group Account.
    --erase-user                       Erases a User, complying to GDPR Data Erasure
                                       (Right to be Forgotten).
    --get-config-property              Gets configuration property value.
    --list-users                       Lists the names of all the Users in the
                                       system, sorted by user name, one per line.
    --list-user-groups                 Lists the names of all the User Groups in the
                                       system, sorted by name, one per line.
    --list-user-group-members          Lists the names of the user group members in
                                       the system, sorted by user name, one per line.
    --list-user-group-memberships      Lists the names of the groups a user belongs
                                       to, sorted by name, one per line.
    --list-user-source-groups          Lists the names of all the groups in the user
                                       source, sorted by name, one per line.
    --list-user-source-group-members   Lists the names of the (nested) user group
                                       members in the user source, sorted by user
                                       name, one per line.
    --list-user-source-group-nesting   Lists a space indented hierarchy of nested
                                       groups within a group. Nested groups are only
                                       supported by Active Directory, all other user
                                       sources return an empty list.
    --printer-access-control           Controls user groups to either allow or deny
```

```
                                        access to a proxy printer.
   --printer-snmp                       Reads SNMP info from a printer.
   --set-config-property                Sets configuration property value.
   --set-user-properties                Sets properties for an existing Internal or
                                        External User.
   --set-user-group-properties          Sets properties of an Internal or External
                                        User Group.
   --sync-user-group                    Synchronizes a user group with the external
                                        user source, updating group membership.
   --sync-users-and-groups              Starts user and group synchronization with
                                        external user source.
   --system-status                      Gets the system status enum value: READY,
                                        SETUP, UNAVAILABLE, MAINTENANCE.
-help,--help                            Displays this help text.
   --help-all                           Displays help text of all methods.
```

Note

The number of available methods will grow according to customer needs. Please contact support if you need a method that is missing.

C.1.1. Common Options

C.1.1.1. Keep Switches

`--keep-*` option switches are used to *not* overwrite existing values.

For example, the `--keep-card`, `--keep-pin` and `--keep-password` switches make their corresponding `--card`, `--pin` and `--password` options act as defaults in those cases where values have not yet been set.

Some examples:

```
# Overwrite any PIN set by user.
--add-internal-user --username "guest-john" --pin "1234"

# Preserve any PIN set by user.
--add-internal-user --username "guest-john" --pin "1234" --keep-pin
```

C.1.1.2. Remove Switches

`--remove-*` option switches are used to *clear* values. Since the *absence* of a command-line option (or an empty value in batch mode CSV/TSV files) can *not* be interpreted as *no value* (`null`), the `--remove` switch comes to help to explicitly nullify values.

This implies that blank values on the command-line and in batch mode input files are ignored. So, this command has no effect ...

```
--add-internal-user --username "guest-john" --pin ""
```

... use this command instead ...

```
--set-user-properties --username "guest-john" --remove-pin
```

When an option does *not* have a `--remove-*` switch, there is no way to clear the corresponding field. For example, since `--remove-full-name` is not available, there is no way to clear the User field “full-name” from the command-line (see Section C.1.20, “setUserProperties” [338]).

C.1.1.3. Locale Option

Some methods pass numeric values that are formatted according to the locale. In these cases the locale can be specified with a separate option like this:

```
 -locale <arg>                        The IETF BCP 47 Locale used for numeric values.
                                      Example values are: en, en-GB, en-US, nl,
                                      nl-NL, nl-BE. [defaults to system default
                                      en-US].
```

> **Note**
>
> The actual system default `locale` depends on your terminal session settings.

C.1.1.4. Batch Mode Options

Some methods have options for passing values in batch mode. Below are the standard batch mode parameters:

```
 -batch                               Enables batch mode: executing from CSV or TSV
                                      input.
 -continue                            Continues batch processing after a batch line
                                      execution error.
 -input <arg>                         Batch input file: optional with stdin as
                                      default.
 -charset <arg>                       IANA Charset Name of batch input character
                                      encoding [default: utf-8].
```

So instead of using these three commands ...

```
./savapage-cmd --add-internal-user --username john --password rTf4g
./savapage-cmd --add-internal-user --username dave --password 9j6Tw
./savapage-cmd --add-internal-user --username mick --password f75L2
```

... you can use this single batch command ...

```
./savapage-cmd --add-internal-user -batch -input /path/to/add-internal-user.csv
```

.. where the file `add-internal-user.csv` looks like this:

```
"username","password"
"john","rTf4g"
"dave","9j6Tw"
"mick","f75L2"
```

Input files must have the extension `.csv` or `.tsv` as indication for a comma or tab separated file format.

The first line in the file must be the comma or tab separated list of parameters. The convention is that the parameter names are identical to their command line counterpart, except for the `--` prefix. The next lines simply contain the comma or tab separated parameter values.

Option *switches* like applied in the command below ...

```
--set-user-properties --username "john" --pin 1234 --remove-card --full-name "John Brown"
--set-user-properties --username "carol" --pin 4713 --keep-pin --full-name "Carol Johnson"
```

... can be applied in a CSV file like this:

```
"username,"pin","keep-pin","remove-card","full-name"
"john",1234,,"true","John Brown"
"carol",4713,"true",,"Carol Johnson"
```

Important

By default, batch processing is interrupted after a batch line execution error. With the `-continue` switch set, it will instead continue processing. After the batch finishes it will return error code `5` to distinguish continuation from an immediate termination, which is reported with error return code `1`.

Note

In a CSV/TSV file any *blank* switch value is interpreted as not present (`false`), any non-blank value as present (`true`).

C.1.2. addInternalUser

```
./savapage-cmd --add-internal-user --help
```

... gives the options:

```
_________________________________
SavaPage Command Line Interface

Method  : addInternalUser
Version : 0.30

Creates a new or updates an existing Internal User.

usage: --add-internal-user [OPTION]...
    --username <text(50)>             [required] Unique user name.
    --password <text(64)>             [optional] Password.
    --full-name <text(255)>           [optional] Full user name.
    --email <text(255)>               [optional] Primary Email address.
    --email-other <list>              [optional] List of space separated other
                                      (secondary) Email addresses.
    --card <text(16)>                 [optional] NFC Card Number.
    --card-format <HEX|DEC>           [optional] NFC Card Number Format [default:
                                      HEX].
    --card-first-byte <LSB|MSB>       [optional] NFC Card Number First Byte [default:
                                      LSB].
    --id <text(16)>                   [optional] ID Number.
    --pin <text(16)>                  [optional] PIN for ID and Card.
    --yubikey <text(12)>              [optional] YubiKey Public ID.
    --uuid <text(36)>                 [optional] The user's secret UUID.
    --balance <decimal>               [optional] The user's initial account balance.
                                      This value is ignored when a balance is already
                                      assigned.
    --balance-comment <text(255)>     [optional] A comment to be associated with the
                                      --balance transaction.
    --credit-limit                    [optional] Assign default credit limit amount.
    --credit-limit-amount <decimal>   [optional] Assign custom credit limit amount.
    --credit-limit-none               [optional] no credit limit restriction (opposed
                                      to --credit-limit and --credit-limit-amount).
    --keep-card                       [optional] Keep existing Card Number, or use
                                      --card value when not present.
    --keep-id                         [optional] Keep existing ID Number, or use --id
                                      value when not present.
    --keep-email-other                [optional] Keep existing other (secondary)
                                      Email addresses, or use --email-other value
                                      when not present.
    --keep-password                   [optional] Keep existing Password, or use
                                      --password value when not present.
    --keep-pin                        [optional] Keep existing PIN, or use --pin
                                      value when not present.
```

```
    --keep-uuid                       [optional] Keep existing UUID, or use --uuid
                                      value when not present.
 -h,--help                            Displays this help text.
 -batch                               Enables batch mode: executing from CSV or TSV
                                      input.
 -continue                            Continues batch processing after a batch line
                                      execution error.
 -input <arg>                         Batch input file: optional with stdin as
                                      default.
 -charset <arg>                       IANA Charset Name of batch input character
                                      encoding [default: utf-8].
 -locale <arg>                        The IETF BCP 47 Locale used for numeric values.
                                      Example values are: en, en-GB, en-US, nl,
                                      nl-NL, nl-BE. [defaults to system default
                                      en-US].
```

C.1.3. addUserGroup

```
./savapage-cmd --add-user-group --help
```

... gives the options:

```
_________________________________
SavaPage Command Line Interface

Method  : addUserGroup
Version : 0.10

Adds a user group from the external user source: synchronized external users
belonging to this group are added as member.

usage: --add-user-group [OPTION]...

    --groupname <text(255)>   [required] Unique group name.
 -h,--help                    Displays this help text.
```

C.1.4. changeBaseCurrency

```
./savapage-cmd --change-base-currency --help
```

... gives the options:

```
_________________________________
SavaPage Command Line Interface

Method  : changeBaseCurrency
Version : 0.10

Changes the base currency of the application.

This action creates financial transactions to align each account to the new
currency: the current account balance is nullified by a debit transaction
and replaced with the new currency according to the exchange rate via a
credit transaction.

Individual credit limits are converted as well, default credit limits are not.

WARNING: Create a database back-up before executing this command!

usage: --change-base-currency [OPTION]...
    --from <text(3)>            [required] The current currency code (ISO 4217).
    --to <text(3)>              [required] The new currency code (ISO 4217).
    --exchange-rate <decimal>   [required] The exchange rate.
```

```
    --test                      [optional] Dry run, changes are not committed.
 -h,--help                      Displays this help text.
```

C.1.5. deleteUser

```
./savapage-cmd --delete-user --help
```

... gives the options:

```
_________________________________
SavaPage Command Line Interface

Method  : deleteUser
Version : 0.10

Logically deletes a User.

usage: --delete-user [OPTION]...

    --username <text(50)>   [required] Unique user name.
 -h,--help                  Displays this help text.
 -batch                     Enables batch mode: executing from CSV or TSV input.
 -continue                  Continues batch processing after a batch line
                            execution error.
 -input <arg>               Batch input file: optional with stdin as default.
 -charset <arg>             IANA Charset Name of batch input character encoding
                            [default: utf-8].
```

C.1.6. deleteUserGroup

```
./savapage-cmd --delete-user-group --help
```

... gives the options:

```
_________________________________
SavaPage Command Line Interface

Method  : deleteUserGroup
Version : 0.10

Deletes a user group.

usage: --delete-user-group [OPTION]...

    --groupname <text(255)>   [required] Unique group name.
 -h,--help                    Displays this help text.
```

C.1.7. deleteUserGroupAccount

```
./savapage-cmd --delete-user-group-account --help
```

... gives the options:

```
_________________________________
SavaPage Command Line Interface

Method  : deleteUserGroupAccount
Version : 0.10

Logically deletes a User Group Account.
```

```
usage: --delete-user-group-account [OPTION]...
    --groupname <text(255)>   [required] Unique group name.
 -h,--help                    Displays this help text.
```

This method has the same effect as ticking the Delete checkbox in the Edit Account dialog.

C.1.8. eraseUser

```
./savapage-cmd --erase-user --help
```

... gives the options:

```
_____________________________
SavaPage Command Line Interface

Method  : eraseUser
Version : 0.10

Erases a User, complying to GDPR Data Erasure (Right to be Forgotten).

usage: --erase-user [OPTION]...
    --username <text(50)>   [required] Unique user name.
 -h,--help                  Displays this help text.
 -batch                     Enables batch mode: executing from CSV or TSV input.
 -continue                  Continues batch processing after a batch line execution
                            error.
 -input <arg>               Batch input file: optional with stdin as default.
 -charset <arg>             IANA Charset Name of batch input character encoding
                            [default: utf-8].
```

This command clears the user's personal data in the active database, including all identifying content from the transaction log, document log and personal account. Transaction and document log details are still held in the database for reporting purposes, but without any user related data. The user is also logically deleted, when not already done so: see Section C.1.5, “deleteUser” [333] and Section 4.11.3, “User Creation” [155].

Caution

When a database is restored, erased users might be restored as identifiable users again. Therefore, keep Data Erasure requests in a separate administration, so that they can easily be retrieved and re-executed.

Warning

Erasing users without deleting them from their external user source has a temporary effect, since they will be created again at the next synchronization.

Note

Data Erasure does *not* include database backups and log files.

C.1.9. getConfigProperty

```
./savapage-cmd --get-config-property --help
```

... gives the options:

```
_____________________________
```

```
SavaPage Command Line Interface

Method  : getConfigProperty
Version : 0.10

Gets configuration property value.

This is an advanced command. Please contact SavaPage Support before use.

usage: --get-config-property [OPTION]...
    --name <text(100)>   [required] Property name.
 -h,--help               Displays this help text.
```

Note

Values that are stored encrypted are returned decrypted. See Section 17.8, “Encrypted Secrets” [292].

Also see Section C.5.3, “db-config-get” [354] and Section C.1.19, “setConfigProperty” [338].

C.1.10. listUsers

```
./savapage-cmd --list-users --help
```

... gives the options:

```
_______________________________
SavaPage Command Line Interface

Method  : listUsers
Version : 0.10

Lists the names of all the Users in the system, sorted by user name, one per line.

usage: --list-users [OPTION]...

 -h,--help   Displays this help text.
```

C.1.11. listUserGroups

```
./savapage-cmd --list-user-groups --help
```

... gives the options:

```
_______________________________
SavaPage Command Line Interface

Method  : listUserGroups
Version : 0.10

Lists the names of all the User Groups in the system, sorted by name, one per line.

usage: --list-user-groups [OPTION]...

 -h,--help   Displays this help text.
```

C.1.12. listUserGroupMembers

```
./savapage-cmd --list-user-group-members --help
```

... gives the options:

```
________________________________
SavaPage Command Line Interface

Method  : listUserGroupMembers
Version : 0.10

Lists the names of the user group members in the system, sorted by user name,
one per line.

usage: --list-user-group-members [OPTION]...

    --groupname <text(255)>   [required] Unique group name.
 -h,--help                    Displays this help text.
```

C.1.13. listUserGroupMemberships

```
./savapage-cmd --list-user-group-memberships --help
```

... gives the options:

```
________________________________
SavaPage Command Line Interface

Method  : listUserGroupMemberships
Version : 0.10

Lists the names of the groups a user belongs to, sorted by name, one per line.

usage: --list-user-group-memberships [OPTION]...

    --username <text(50)>   [required] Unique user name.
 -h,--help                  Displays this help text.
```

C.1.14. listUserSourceGroups

```
./savapage-cmd --list-user-source-groups --help
```

... gives the options:

```
________________________________
SavaPage Command Line Interface

Method  : listUserSourceGroups
Version : 0.10

Lists the names of all the groups in the user source, sorted by name, one per line.

usage: --list-user-source-groups [OPTION]...

 -h,--help   Displays this help text.
```

C.1.15. listUserSourceGroupMembers

```
./savapage-cmd --list-user-source-group-members --help
```

... gives the options:

```
________________________________
SavaPage Command Line Interface

Method  : listUserSourceGroupMembers
```

```
Version : 0.10

Lists the names of the (nested) user group members in the user source,
sorted by user name, one per line.

usage: --list-user-source-group-members [OPTION]...

    --groupname <text(255)>   [required] Unique group name.
    --nested                  [optional] Accumulate members from nested groups
                              (Active Directory only).
 -h,--help                    Displays this help text.
```

C.1.16. listUserSourceGroupNesting

```
./savapage-cmd --list-user-source-group-nesting --help
```

... gives the options:

```
_______________________________
SavaPage Command Line Interface

Method  : listUserSourceGroupNesting
Version : 0.10

Lists a space indented hierarchy of nested groups within a group. Nested groups
are only supported by Active Directory, all other user sources return an empty list.

usage: --list-user-source-group-nesting [OPTION]...

    --groupname <text(255)>   [required] Unique group name.
 -h,--help                    Displays this help text.
```

C.1.17. printerAccessControl

```
./savapage-cmd --printer-access-control --help
```

... gives the options:

```
_______________________________
SavaPage Command Line Interface

Method  : printerAccessControl
Version : 0.10

Controls user groups to either allow or deny access to a proxy printer.

usage: --printer-access-control [OPTION]...

    --printername <text(255)>   [required] CUPS name of the proxy printer.
    --allow                     [optional] Allow access to --groupname (existing
                                denied user groups are removed).
    --deny                      [optional] Deny access to --groupname (existing
                                allowed user groups are removed).
    --remove                    [optional] Remove --groupname from the access list.
    --groupname <text(255)>     [optional] Name of the user group to --allow, --deny
                                or --remove access
    --remove-all                [optional] Remove all user groups from the access
                                list.
    --list                      [optional] Echoes the access list to stdout in CSV
                                format.
 -h,--help                      Displays this help text.
```

User Groups for which access to a Proxy Printer is allowed or denied, are shown in the Proxy Printer List.

C.1.18. printerSnmp

```
./savapage-cmd --printer-snmp --help
```

... gives the options:

```
_______________________________
SavaPage Command Line Interface

Method  : printerSnmp
Version : 0.20

Reads SNMP info from a printer.

usage: --printer-snmp [OPTION]...
    --printername <text(255)>   [optional] CUPS printer name used to resolve host
                                name (required when --host is not set).
    --host <text>               [optional] Host name or IP address of the printer
                                (required when --printername is not set).
    --port <number>             [optional] SNMP port number (default 161).
    --community <text>          [optional] SNMP community (default "public").
    --version <1|2c>            [optional] SNMP version (default "1").
 -h,--help                      Displays this help text.
```

C.1.19. setConfigProperty

```
./savapage-cmd --set-config-property --help
```

... gives the options:

```
_______________________________
SavaPage Command Line Interface

Method  : setConfigProperty
Version : 0.10

Sets configuration property value.

The property must be present in the database. Not all properties are available for update.
This is an advanced command. Please contact SavaPage Support before use.

usage: --set-config-property [OPTION]...
    --name <text(100)>     [required] Property name.
    --value <text(1000)>   [required] Property value.
 -h,--help                 Displays this help text.
 -batch                    Enables batch mode: executing from CSV or TSV input.
 -continue                 Continues batch processing after a batch line execution
                           error.
 -input <arg>              Batch input file: optional with stdin as default.
 -charset <arg>            IANA Charset Name of batch input character encoding
                           [default: utf-8].
```

Note

Some values will be stored encrypted. See Section 17.8, “Encrypted Secrets” [292].

Also see Section C.5.4, “db-config-set” [354] and Section C.1.9, “getConfigProperty” [334].

C.1.20. setUserProperties

```
./savapage-cmd --set-user-properties --help
```

... gives the options:

```
_____________________________
SavaPage Command Line Interface

Method  : setUserProperties
Version : 0.30

Sets properties for an existing Internal or External User.

usage: --set-user-properties [OPTION]...
    --username <text(50)>             [required] Unique user name.
    --password <text(64)>             [optional] Password (Internal User only).
    --full-name <text(255)>           [optional] Full user name.
    --email <text(255)>               [optional] Primary Email address.
    --email-other <list>              [optional] List of space separated other
                                      (secondary) Email addresses.
    --card <text(16)>                 [optional] NFC Card Number.
    --card-format <HEX|DEC>           [optional] NFC Card Number Format [default:
                                      HEX].
    --card-first-byte <LSB|MSB>       [optional] NFC Card Number First Byte [default:
                                      LSB].
    --id <text(16)>                   [optional] ID Number.
    --pin <text(16)>                  [optional] PIN for ID and Card.
    --yubikey <text(12)>              [optional] YubiKey Public ID.
    --uuid <text(36)>                 [optional] The user's secret UUID.
    --balance <decimal>               [optional] The user's current account balance.
    --balance-comment <text(255)>     [optional] A comment to be associated with the
                                      --balance transaction.
    --credit-limit                    [optional] Assign default credit limit amount.
    --credit-limit-amount <decimal>   [optional] Assign custom credit limit amount.
    --credit-limit-none               [optional] No credit limit restriction (opposed
                                      to --credit-limit and --credit-limit-amount).
    --keep-card                       [optional] Keep existing Card Number, or use
                                      --card value when not present.
    --keep-id                         [optional] Keep existing ID Number, or use --id
                                      value when not present.
    --keep-email-other                [optional] Keep existing other (secondary)
                                      Email addresses, or use --email-other value
                                      when not present.
    --keep-password                   [optional] Keep existing Password, or use
                                      --password value when not present (Internal
                                      User only).
    --keep-pin                        [optional] Keep existing PIN, or use --pin
                                      value when not present.
    --keep-uuid                       [optional] Keep existing UUID, or use --uuid
                                      value when not present.
    --remove-email                    [optional] Remove Primary Email address
                                      (opposed to --email).
    --remove-email-other              [optional] Remove other (secondary) Email
                                      addresses (opposed to --email-other).
    --remove-card                     [optional] Remove NFC Card Number (opposed to
                                      --card).
    --remove-id                       [optional] Remove ID Number (opposed to --id).
    --remove-password                 [optional] Remove Password (Internal User
                                      only).
    --remove-pin                      [optional] Remove PIN (opposed to --pin).
    --remove-yubikey                  [optional] Remove YubiKey Public ID (opposed to
                                      --yubikey).
    --remove-uuid                     [optional] Remove UUID (opposed to --uuid).
 -h,--help                            Displays this help text.
 -batch                               Enables batch mode: executing from CSV or TSV
                                      input.
 -continue                            Continues batch processing after a batch line
                                      execution error.
 -input <arg>                         Batch input file: optional with stdin as
                                      default.
```

```
 -charset <arg>                        IANA Charset Name of batch input character
                                       encoding [default: utf-8].
 -locale <arg>                         The IETF BCP 47 Locale used for numeric values.
                                       Example values are: en, en-GB, en-US, nl,
                                       nl-NL, nl-BE. [defaults to system default
                                       en-US].
```

C.1.21. setUserGroupProperties

```
./savapage-cmd --set-user-group-properties --help
```

... gives the options:

```
_________________________________
SavaPage Command Line Interface

Method  : setUserGroupProperties
Version : 0.11

Sets properties of an Internal or External User Group.

usage: --set-user-group-properties [OPTION]...
    --groupname <text(255)>              [required] Unique group name.
    --balance <decimal>                  [optional] The user's initial account
                                         balance.
    --credit-limit                       [optional] Assign default credit limit
                                         amount to new users.
    --credit-limit-amount <decimal>      [optional] Assign custom credit limit amount
                                         to new users.
    --credit-limit-none                  [optional] Assign no credit limit
                                         restriction to new users (opposed to
                                         --credit-limit and --credit-limit-amount).
    --role-job-ticket-creator <Y|N|U>    [optional] Assign Job Ticket Creator role.
    --role-job-ticket-operator <Y|N|U>   [optional] Assign Job Ticket Operator role.
    --role-print-creator <Y|N|U>         [optional] Assign Print Creator role.
    --role-print-delegate <Y|N|U>        [optional] Assign Print Delegate role.
    --role-print-delegator <Y|N|U>       [optional] Assign Print Delegator role.
    --role-web-cashier <Y|N|U>           [optional] Assign Web Cashier role.
 -h,--help                               Displays this help text.
 -batch                                  Enables batch mode: executing from CSV or
                                         TSV input.
 -continue                               Continues batch processing after a batch
                                         line execution error.
 -input <arg>                            Batch input file: optional with stdin as
                                         default.
 -charset <arg>                          IANA Charset Name of batch input character
                                         encoding [default: utf-8].
 -locale <arg>                           The IETF BCP 47 Locale used for numeric
                                         values. Example values are: en, en-GB,
                                         en-US, nl, nl-NL, nl-BE. [defaults to system
                                         default en-US].
```

C.1.22. syncUserGroup

```
./savapage-cmd --sync-user-group --help
```

... gives the options:

```
_________________________________
SavaPage Command Line Interface

Method  : syncUserGroup
Version : 0.10
```

```
Synchronizes a user group with the external user source, updating group membership.

usage: --sync-user-group [OPTION]...

    --groupname <text(255)>   [required] The name of the group to synchronize.
 -h,--help                    Displays this help text.
```

C.1.23. syncUsersAndGroups

```
./savapage-cmd --sync-users-and-groups --help
```

... gives the options:

```
_______________________________
SavaPage Command Line Interface

Method  : syncUsersAndGroups
Version : 0.10

Starts user and group synchronization with external user source.

This is equivalent to clicking "Synchronize now" in the Admin Web App.
Synchronization completes in the background.

usage: --sync-users-and-groups [OPTION]...
    --delete-users   [optional] Remove users that do not exist in external user
                     source.
 -h,--help           Displays this help text.
```

See Section 4.11.3, “User Creation” [155].

C.1.24. systemStatus

```
./savapage-cmd ---system-status --help
```

... gives the options:

```
_______________________________
SavaPage Command Line Interface

Method  : systemStatus
Version : 0.10

Gets the system status enum value: READY, SETUP, UNAVAILABLE, MAINTENANCE.

usage: --system-status [OPTION]...
 -h,--help   Displays this help text.
```

This command is identical to Section C.2.2.1, “systemStatus” [344].

C.2. Web Services

SavaPage uses XML-RPC[1], JSON-RPC[2] and RESTful API[3] for its public Web Services. Both protocols are Open Standard, lightweight and have support for all major programming and scripting languages.

[1] https://en.wikipedia.org/wiki/XML-RPC
[2] https://en.wikipedia.org/wiki/JSON-RPC
[3] https://en.wikipedia.org/wiki/Representational_state_transfer

Note

Rate Limiting is applied to unauthorized access and invalid requests.

Note

The range of Web Services will grow upon request. Please tell us if you need extra methods.

C.2.1. XML-RPC

The secure endpoint is: `https://savapage:8632/xmlrpc/v1`

C.2.1.1. onetime-auth.createToken

With this method a Trusted Third Party (TTP) acquires a one-time token for Web App user authentication.

The requesting User Name and returned Token must be offered to the Web App by hidden HTTP POST input names `auth_user` and `auth_token`. The Web App decrypts the token and honours the authentication request if the offered and encrypted Users match and the token is not expired.

	Name	Type	Description
Parameters	apikey	string	The TTP API key as set in the `web-login.ttp.apikey` configuration property.
	username	string	The User Name to authenticate. See Section 3.1, “Login” [22]. User Alias mapping is applied.
Returns	token	string	The one-time authentication token holding the encrypted User Name and token creation time. The expiration criterion in milliseconds is set in the `web-login.ttp.token.expiry-msecs` configuration property.

Table C.1. XML-RPC method: onetime-auth.createToken

Note

This method is used by moodle-block_savapage[4].

Important

TTP Web App Login authentication must be enabled by setting the `web-login.ttp.enable` configuration property to `Y`.

See Section 4.11.15, “Config Editor” [189] on how to set the configuration properties `web-login.ttp.enable`, `web-login.ttp.apikey` and `web-login.ttp.token.expiry-msecs`.

```
#---------------------------
# sample request.xml
#---------------------------
```

[4] https://moodle.org/plugins/block_savapage

```
<?xml version="1.0"?>
<methodCall>
  <methodName>onetime-auth.createToken</methodName>
  <params>
    <param><value>your-apikey</value></param>
    <param><value>a-userid</value></param>
  </params>
</methodCall>

#---------------------------
# sample request
#---------------------------
curl -k -H "content-type:text/xml" https://savapage:8632/xmlrpc/v1 --data @request.xml

#---------------------------
# sample response
#---------------------------
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse xmlns:ex="http://ws.apache.org/xmlrpc/namespaces/extensions">
  <params>
    <param><value>the-onetime-token</value></param>
  </params>
</methodResponse>

#---------------------------
# sample html
#---------------------------
<form method="post" action="https://savapage:8632/user?sp-user=a-userid">
  <input type= "hidden" name= "auth_user"  value="a-userid" >
  <input type= "hidden" name= "auth_token" value="the-onetime-token">
  <input type="submit" value="Logon">
</form>
```

C.2.2. JSON-RPC

The secure endpoint is: `https://savapage:8632/jsonrpc/v1`

The following configuration properties apply:

Configuration property	Description
api.jsonrpc.secret-key	The secret key to be passed as `X-Auth-Key` HTTP header.
api.jsonrpc.ext.ip-addresses-allowed	A CIDR Set of allowed client IPv4 and/or IPv6 addresses. When void access is denied for *all* clients.

Table C.2. JSON-RPC Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these items.

The sections below discuss the individual methods. All methods default to the same response layout in case of an error.

Basic Error

jsonrpc constant string `2.0` JSON-RPC version.

id string Reference ID.

result constant boolean `false` Error indicator.

error object

Error

code enum integer `-32600` `-32601` `-32602` `-32603` `-32700`

-32600 (invalid request), -32601 (method not found), -32602 (Invalid params), -32603 (internal error), -32700 (parse error).

message string Error message.

data object

Data

@type constant string `BASIC` Data type indicator.

reason string Error reason.

Figure C.1. JSON-RPC : Basic Error

C.2.2.1. systemStatus

This method returns the system status as enum value. `READY` indicates the system is ready to use and users have access to the Web App. All other values indicate that user access is *denied*, because: system `SETUP` is needed, `MAINTENANCE` is in progress, or the system is temporarily `UNAVAILABLE` due to running batch jobs (database backup, user synchronization, database cleanup).

Note

Although the system can have multiple "deny status" at the same time, the single most important status is returned, in the order `SETUP`, `MAINTENANCE`, `UNAVAILABLE`.

systemStatus Gets System Status

jsonrpc constant string `2.0` JSON-RPC version.

id string Reference ID.

method constant string `systemStatus` Method name.

Figure C.2. JSON-RPC : systemStatus (request)

systemStatus

jsonrpc constant string 2.0 JSON-RPC version.

id string Reference ID.

result object

Result

data object

Data

@type constant string enum Data type indicator.

value

enum string READY SETUP MAINTENANCE UNAVAILABLE

System status.

Figure C.3. JSON-RPC : systemStatus (response)

```
# sample request
curl -k --request POST --header "X-Auth-Key: 7961b5ec-bee4-11e7-8731-406186940c49" \
     --data '{"jsonrpc":"2.0","id":"reference","method":"systemStatus"}' \
     https://savapage:8632/jsonrpc/v1

# sample response
 {
  "jsonrpc" : "2.0",
  "id" : "reference",
  "result" : {
    "data" : {
      "@type" : "enum",
      "value" : "READY"
    }
  }
}
```

C.2.2.2. authUserSource

This method authenticates an external user against the configured User Source.

authUserSource Authenticate External User

jsonrpc constant string 2.0 JSON-RPC version.

id string Reference ID.

method constant string authUserSource Method name.

params object

Parameters

username string User Name to authenticate.

password string Password.

Figure C.4. JSON-RPC : authUserSource (request)

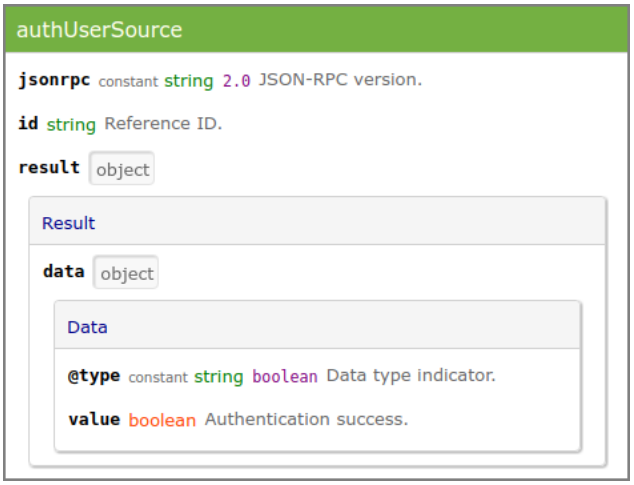


Figure C.5. JSON-RPC : authUserSource (response)

```
# sample request
curl -k --request POST --header "X-Auth-Key: 7961b5ec-bee4-11e7-8731-406186940c49" \
     --data '{"jsonrpc":"2.0","id":"reference","method":"authUserSource", \
               "params":{"username":"john","password":"AzFi7I"}}'
     https://savapage:8632/jsonrpc/v1

# sample response
{
  "jsonrpc" : "2.0",
  "id" : "reference",
  "result" : {
    "data" : {
      "@type" : "boolean",
      "value" : true
    }
  }
}
# sample error
{
  "jsonrpc" : "2.0",
  "id" : "reference",
  "result" : false,
  "error" : {
    "code" : -32603,
    "message" : "ldapserver:636 [No route to host (Host unreachable)]",
    "data" : {
      "@type" : "BASIC",
      "reason" : "ldapserver:636"
    }
  }
}
```

C.2.3. RESTful API

> **Preview**
>
> RESTful Web Services is a Feature Preview that is partly operational and impermanent. The preview is offered to provoke feedback based on real world use. Please tell us your requirements.

The secure endpoint is: `https://savapage:8632/restful/v1`

Services are described with **curl** examples.

C.2.3.1. RESTful Public Services

C.2.3.1.1. GET /system/version

```
curl -k -X GET "https://savapage:8632/restful/v1/system/version"

# sample response
SavaPage 1.2.0-rc (Build 20200121)
```

C.2.3.1.2. GET /tests/http/request

```
curl -k -X GET \
     "https://savapage.example.com/restful/v1/tests/http/request"

# sample response
{
  "remoteAddr" : "[2021:661:4cc6:1:5561:96ff:ca11:d4a]",
  "remoteHost" : "[2021:661:4cc6:1:5561:96ff:ca11:d4a]",
  "remotePort" : 53840,
  "localAddr" : "127.0.0.1",
  "serverName" : "savapage.example.com",
  "serverPort" : 443,
  "headers" : {
    "Accept" : "*/*",
    "User-Agent" : "curl/7.47.0",
    "X-Forwarded-Proto" : "https",
    "X-Forwarded-Host" : "savapage.example.com",
    "Connection" : "keep-alive",
    "X-Forwarded-For" : "2021:661:4cc6:1:5561:96ff:ca11:d4a",
    "Host" : "savapage.example.com",
    "X-Forwarded-Server" : "savapage.example.com"
  }
}
```

C.2.3.2. RESTful Admin Services

Access to Admin Services is controlled by Basic HTTP Authentication.

Configuration property	Description
api.restful.auth.username	Basic Authentication Username.
api.restful.auth.password	Basic Authentication Password.
api.restful.ip-addresses-allowed	A CIDR Set of allowed client IPv4 and/or IPv6 addresses. When void access is allowed for *all* clients.

Table C.3. RESTful Admin Services Configuration Properties

C.2.3.2.1. POST /tests/echo

```
curl -k -u admin:password -X POST \
     "https://savapage:8632/restful/v1/tests/echo" \
     -H "Content-Type:text/plain" -d "Hello World!"

# response
Hello World!
```

C.2.3.2.2. GET /documents

```
# GET JSON info of first 5 documents sorted by creation datetime descending
curl -k -u admin:password -X GET \
     "https://savapage:8632/restful/v1/documents?sort=-created"

# ... sorted ascending
curl -k -u admin:password -X GET \
     "https://savapage:8632/restful/v1/documents?sort=+created"
```

```
# GET JSON info of first page chunk of 2 proxy printed documents
# type = all (default) | in | out | pdf | print | ticket

curl -k -u admin:password -X GET \
     "https://savapage:8632/restful/v1/documents?sort=-created&type=print&limit=2&page=0"

# sample response
[ {
  "id" : 6263,
  "ext_id" : "",
  "type" : "print",
  "title" : "Document A",
  "created" : "2019-09-13T09:18:38+0200",
  "store" : false
}, {
  "id" : 6258,
  "ext_id" : "c5a9dd70e931",
  "type" : "print",
  "title" : "Document B",
  "created" : "2019-09-12T09:49:48+0200",
  "store" : false
} ]
```

C.2.3.2.3. GET /documents/<id>

```
# GET JSON info of document id 6272
curl -k -u admin:password -X GET \
     "https://savapage:8632/restful/v1/documents/6272"

# sample response
{
  "id" : 6272,
  "ext_id" : "d2a3a71c11e9",
  "type" : "print",
  "title" : "Invoice 20190248",
  "created" : "2019-09-16T12:29:15+0200",
  "store" : true
}
```

C.2.3.2.4. GET /documents/ext_id=<ext_id>

```
# GET JSON info of document with ext_id d2a3a71c11e9
curl -k -u admin:password -X GET \
     "https://savapage:8632/restful/v1/documents/ext_id=d2a3a71c11e9"
```

C.2.3.2.5. GET /documents/<id>/pdf

```
# GET stored PDF document 6272
curl -k -u admin:password -X GET \
     "https://savapage:8632/restful/v1/documents/6272/pdf" > 6272.pdf
```

C.2.3.2.6. GET /documents/ext_id=<ext_id>/pdf

```
# GET stored PDF document with ext_id d2a3a71c11e9
curl -k -u admin:password -X GET \
     "https://savapage:8632/restful/v1/documents/ext_id=d2a3a71c11e9/pdf" \
     > d2a3a71c11e9.pdf
```

C.2.3.2.7. GET /financial/account/balance

```
# GET JSON info of account balance
# type = USER | GROUP | SHARED

curl -k -u admin:password -X GET \
     "https://savapage:8632/restful/v1/financial/account/balance?type=USER&name=rijk"

# sample response
{
  "succes" : true,
  "result" : "9.260000"
}

# sample response
{
  "succes" : false,
  "error" : "error message"
}
```

See Chapter 12, *SavaPage Financial* [243].

C.2.3.2.8. POST /financial/account/balance

```
# Sets or adjusts account balance of an account.
#
# URL parameters (by example):
#
# Mandatory :
#
# type=USER   : USER | GROUP | SHARED
# name=rijk   : identifying name of account
# amount=2.50 : can be negative, like -2.50
#
# Optional:
#
# details=my%20text : transaction details "my text"
# adjust=true       : 'true' creates a transaction to adjust the balance with amount
#                     (a negative amount reduces the balance), 'false' (default)
#                     creates a transaction to set balance to amount

curl -k -u admin:password -X POST \
  "https://savapage:8632/restful/v1/financial/account/balance?type=USER&name=rijk&amount=2.50"

# sample response
{
  "succes" : true
  "result" : "2.50"    # resulting balance
}
```

```
# sample response
{
  "succes" : false,
  "error" : "error message"
}
```

See Chapter 12, *SavaPage Financial* [243].

> **Note**
>
> The value of URL parameter `details` is visible in the transaction logs. See Section 3.8.2, "Transactions" [73] and Section 4.6.1, "Account List" [117].

C.2.3.2.9. POST /reports/accounttrx

```
# POST selection to download Account Transactions of user "rijk"
curl -k -u admin:password -X POST \
     "https://savapage:8632/restful/v1/reports/accounttrx" \
     -H "Content-Type: application/json" \
     -d '{"select":{"account_type":"USER","account_name":"rijk"}}' > AccountTrxList.csv
```

The POSTed data is a JSON string. Full # annotated example:

```
{
  "select" : {
    "account_type" : "USER",  # USER | GROUP | SHARED
    "account_name" : "rijk",  # optional for USER (all users)
                              # mandatory for GROUP | SHARED
    "text" : null,            # (optional) transaction comment snippet
    "trx_type" : "PRINT_OUT", # (optional )
                              # ADJUST | DEPOSIT | GATEWAY | INITIAL |
                              # PDF_OUT | PRINT_IN | PRINT_OUT |
                              # PURCHASE | TRANSFER | VOUCHER
    "date_from" : {           # (optional) Sun Jan 01 00:00:00 CET 2023
      "ccyy" : 2023,
      "mm" : 1,
      "dd" : 1,
      "zzz" : "+01"
    },
    "date_to" : {             # (optional) Thu Jun 01 00:00:00 CEST 2023
      "ccyy" : 2023,
      "mm" : 6,
      "dd" : 1,
      "zzz" : "+02"
    }
  },
  "sort" : {                  # (optional)
    "field" : "TRX_DATE",     # TRX_DATE (default) | TRX_TYPE
    "ascending" : true        # true (default)| false
  },
  "media_type" : "CSV"        # CSV (default) | PDF
}
```

> **Note**
>
> The output of this service is identical to the PDF and CSV reports from Section 3.8.2, "Transactions" [73].

C.2.3.3. RESTful User Services

Access to User Services is controlled by User ID and UUID.

C.2.3.3.1. POST /documents/print

```
# Print file "lorem-ipsum.txt" to SavaPage as user "janet" with UUID credentials
curl -v -k -u janet:97008e2c-d8b2-11e9-aa13-406186940c49 \
     -H "Content-Type: multipart/form-data" -H "Accept: application/json" \
     -F "file=@lorem-ipsum.txt" \
      https://savapage:8632/restful/v1/documents/print
```

You can add a selection of IPP options just like they would be distilled from a driver printed job. See Section 14.2.1, “IPP Options from PostScript” [262].

```
# Add IPP options in JSON format
-F "file=@lorem-ipsum.txt" \
-F 'ipp_options={"copies":"2","print-color-mode":"monochrome","sides":"one-sided"}'
```

Important

A RESTful Print maps to the reserved Queue `/webservice` by default.

An alternative *non-reserved* Queue can be specified by adding an extra `-F` parameter, like this:

```
# Print two collated copies to alternative queue "test"
-F "file=@lorem-ipsum.txt" -F "queue=test"
-F 'ipp_options={"copies":"2","sheet-collate":"collated"}'
```

If IPP Routing is enabled for the alternative Queue, the RESTful `ipp_options` have priority over and are merged with the IPP Routing options.

To exclusively accept prints from specific IP addresses see Section 4.7.2, “Edit Queue” [123].

Note

RESTful Print is used by the SavaPage Android Print App.

C.2.3.3.2. GET /financial/balance

```
# GET JSON balance of user account "janet"

curl -k -u janet:97008e2c-d8b2-11e9-aa13-406186940c49 -X GET \
     "https://savapage:8632/restful/v1/financial/balance"

# sample response
{
  "succes" : true,
  "result" : "5.350000"
}

# sample response
{
  "succes" : false,
  "error" : "error message"
}
```

See Section 3.10.5, “Financial” [83].

C.2.3.3.3. POST /financial/transfer

```
# Transfers balance amount from user "janet" to "rijk"
```

```
curl -k -u janet:97008e2c-d8b2-11e9-aa13-406186940c49 -X POST \
     "https://savapage:8632/restful/v1/financial/transfer?name=rijk&amount=15.00"

# sample response
{
  "succes" : true
}

# sample response
{
  "succes" : false,
  "error" : "error message"
}
```

See Section 3.10.7, “Transfer Credit” [84].

C.3. Atom Feed Service

SavaPage publishes system information of the past day in Atom Syndication Format[5]. The info is meant for administrators and includes runtime status, statistics, errors and warnings. The feed is available at URL path feed/admin/ and protected by Basic HTTP Authentication. It can be enabled and configured with the properties shown below.

Configuration property	Description
feed.atom.admin.enable	Set to `Y` or `N` (default) to enable or disable.
feed.atom.admin.username	Basic HTTP Authentication User name.
feed.atom.admin.password	Basic HTTP Authentication User password.
feed.atom.admin.schedule	Feed publication schedule according to the default Cron Trigger Expression `"0 0 3 ? * 3-7"`. The previous feed is overwritten: no history is held in SavaPage. See Atom Feed Job. A feed can be published ad hoc in the Admin Web App. See Section 4.11.14.12, “Atom Feed” [187].
feed.atom.admin.uuid	Universally unique identifier of the Atom Feed (generated by SavaPage).

Table C.4. Atom Feed Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to change these properties.

Note

Rate Limiting is applied to unauthorized access.

Tip

QuiteRSS[6] is a open-source cross-platform RSS/Atom news feeds reader.

C.4. Matrix Messaging Service

SavaPage can be configured to publish realtime and summary messages to an *unencrypted* Matrix Room as shown in the table below.

[5] https://en.wikipedia.org/wiki/Atom_(Web_standard)
[6] https://quiterss.org/

Configuration property	Description
system.matrix.enable	Set to `Y` or `N` (default) to enable or disable Matrix services.
system.matrix.homeserver	Matrix home server. For example: `matrix.org`
system.matrix.access-token	Matrix access token.
system.matrix.roomid	ID of the *unencrypted* Matrix room messages are send to. For example: `!CAfJdIOajJsRVhGJfk:matrix.org`
system.matrix.publish.enable	Set to `Y` (default) or `N` to enable or disable publishing.
system.matrix.publish.html	Set to `Y` (default) or `N` to publish in HTML (default) or plain text format.
system.matrix.publish.realtime.enable	Set to `Y` (default) or `N` to enable or disable realtime publishing of runtime errors.
system.matrix.publish.realtime.timeslot	An hourly timeslot, within which runtime errors are published, formatted as hour `from-to`. For example: `8-18` (8 AM till 6 PM). Default: empty (equals to `0-0`, 24 hours a day).
system.matrix.publish.summary.enable	Set to `Y` (default) or `N` to enable or disable publishing of summaries.
system.matrix.publish.summary.day.enable	Set to `Y` (default) or `N` to enable or disable publishing of a daily summary.
system.matrix.publish.summary.day.schedule	Daily summary schedule according to the default Cron Trigger Expression `"0 0 8 ? * 3-7"`. See Matrix Summary Job.

Table C.5. Matrix Messaging Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to change these properties.

Note

The configuration can be tested in the Admin Web App. See Section 4.11.14.13, “Matrix Messaging” [188].

C.5. Database Commands

The **savapage-db** command-line tool provides functions for manipulating the database. The tool is located in `/opt/savapage/server/bin/linux-x64/` and needs to be executed from a command prompt. The syntax of the command is:

```
usage: [OPTION]
    --db-check                    Checks database integrity. This is an advanced
                                  option. Please contact SavaPage Support before
                                  use.
    --db-check-fix                Checks and fixes database integrity. NOTE: Only
                                  perform as requested by SavaPage Support.
    --db-config-get <NAME>        Gets configuration property value. This is an
                                  advanced option. Please contact SavaPage Support
                                  before use.
    --db-config-set <NAME=VALUE>  Sets configuration property value. This is an
                                  advanced option. Please contact SavaPage Support
                                  before use.
    --db-delete-logs <DAYS>       Deletes application, account transaction and
                                  document log data older than DAYS. A DAYS value of
                                  zero (0) will remove all log data from the system.
    --db-export                   Exports the database to the default backup
                                  location.
    --db-export-to <FILE|DIR>     Exports the database to the specified file or
                                  directory.
    --db-import <FILE>            Imports the database from the specified file.
                                  Deletes any existing data before loading the data.
```

```
    --db-init                       Re-initializes the database even if it already
                                    exists.
    --db-run-script <FILE>          Runs SQL statements from the specified script
                                    file. NOTE: Only perform as requested by SavaPage
                                    Support.
    --db-run-sql <STATEMENT>        Runs an SQL statement. NOTE: Only perform as
                                    requested by SavaPage Support.
    --db-schema-version             Echoes database schema version on stdout.
 -h,--help                          Displays this help text.
    --log4j                         Use log4j. NOTE: Only perform as requested by
                                    SavaPage Support.
```

The command must be run as the `savapage` user. An example:

```
sudo su - savapage
cd server/bin/linux-x64
./savapage-db --db-import /home/john/savapage-backup.zip
```

savapage-db needs exclusive access to the database. It is important that any SavaPage services and processes are stopped before executing a command. Failure to do so will result in a "database in use" error message. The **savapage-db** command is a powerful low-level utility and its use on a production system should be carefully considered. Details of the available commands are discussed below.

C.5.1. db-check

Checks database integrity. This is an advanced option. Please contact SavaPage Support before use.

C.5.2. db-check-fix

Checks and fixes database integrity.

> **Note**
>
> Only perform as requested by SavaPage Support.

C.5.3. db-config-get

With this option the (decrypted) value of a Configuration Property can be retrieved. Also see Section C.1.9, “getConfigProperty” [334].

C.5.4. db-config-set

With this option the value of a Configuration Property can be set (and encrypted). Also see Section C.1.19, “setConfigProperty” [338].

C.5.5. db-delete-logs

This option delete old log data from the system. This command will permanently delete the following data.

- Document logs - Record all document history and statistics
- Transaction logs - Record all financial history and statistics
- Application logs - Record application status and error messages

The DAYS option determines what data will be deleted. If DAYS is 90, then all log data more than 90 days old will be deleted. A value of zero (0) will remove all historical log data from the system. This is an example:

```
savapage-db --db-delete-logs 90
```

C.5.6. db-export and db-export-to

This option exports the data from the database. The application server must be stopped before performing the export. See Section C.6, “Stopping and Starting the Server” [356].

Tip

If you want to perform an online backup without stopping the application server you can use the backup function in the Admin Web App. See Section 4.11.13, “Backups” [177].

```
savapage-db --db-export    ❶
savapage-db --db-export-to /home/john    ❷
savapage-db --db-export-to /home/john/savapage-backup.zip    ❸
```

❶ The database export file is created in the `/opt/savapage/server/data/backups` directory and the file is named `savapage-export-[date-time].zip`.

❷ This option is used to override the default backup directory. The filename will still be named `savapage-export-[date-time].zip`.

❸ The full path and filename where the backup is saved is specified.

Note

When executing the command the last line echoed to `stdout` is the canonical path of the database export file.

Tip

The database export zip file contains an unformatted XML file with the database content (tables and rows). An XML formatter can be enabled by setting configuration item `system.backup.xmlformatter.enable` to `Y`. See Section 4.11.15, “Config Editor” [189] on how to set this property.

Caution

If the directory or filename parameters contain spaces, then the argument needs to be quoted.

C.5.7. db-import

This option imports the data (from a previous export) into the database. The application server must be stopped to perform the import. This is an example:

```
savapage-db --db-import /home/john/savapage-backup.zip
```

Note

Progress and statistics of the import process are written to `stdout`.

Warning

Before import the database is initialized. Thereby existing data in the database are erased.

C.5.8. db-init

This option initializes a database, creating the required tables and initial data. The application server must be stopped before you initialize the database. This is the command:

```
savapage-db --db-init
```

Warning

Be careful, existing data in the database are erased.

C.6. Stopping and Starting the Server

Normally there is no need to stop or start the server. This is only required when:

- Performing an off-line backup.
- Migrating the an external database.
- Upgrading the application.

The SavaPage application server may be stopped or started by executing these **systemd** commands:

```
sudo systemctl start savapage.service   ❶
sudo systemctl stop savapage.service   ❷
sudo systemctl restart savapage.service   ❸
systemctl status savapage.service        ❹
```

❶ Starts the application server.
❷ Stops the application server.
❸ Stops and starts the application server in one go.
❹ Echoes the run status of the application server.

When SavaPage is installed on a SysV system, use the appropriate **service** commands.

Important

When you start the application server, wait approximately 10 seconds for the service to initialize before accessing the Web App interface.

Note

When the server is stopped, any SavaPage Web App loses connection and will be unloaded, and a browser message like "This site can’t be reached" will be shown. The user must wait till the server is started again, and press any Reload (Try Again) button to resume.

C.7. SSL Key Generation

Each time SavaPage server starts it checks if the default Self-Signed Key and SHA-2 certificate and password file are present. If not, they are generated (a missing password file will generate both) with defaults described in Section C.7.1, “Re-Create the Self-Signed Certificate” [357]. This certificate is used by default when the system is accessed via HTTPS on port 8632.

This SSL certificate provides good security. However, there are two downsides to using a self-signed certificate, since users accessing the HTTPS site will encounter warnings from the browser:

1. When accessing the HTTPS site with a fully-qualified domain name that does not match the Certificate Subject Alternative Name, the browser will issue a “Domain mismatch warning”. To avoid this warning, re-create the

self-signed certificate with the machine's fully qualified domain name as common name with optional lists of Certificate Subject Alternative DNS Names and IP addresses. See Section C.7.1, “Re-Create the Self-Signed Certificate” [357].

2. The browser will also warn that the certificate is not signed by a trusted authority. To overcome this you have two options:
   a. *Set trust for the Self-Signed Certificate*. For Chrome based browsers:
      - Select the certificate from the browser's URL entry field, and Export as “Base64-encode ASCII, single certificate” `savapage.pem` file.
      - Import `savapage.pem` as Certificate Authority and “Trust this certificate for identifying websites”.

      For Firefox you need to create a permanent server exception for each SavaPage `https://fqdn:port` via Certificate Manager > Servers > "Add Exception".

   b. *Obtain a certificate signed by a trusted authority*. See Section C.7.2, “Importing an Existing SSL Certificate” [358].

C.7.1. Re-Create the Self-Signed Certificate

The tool **create-ssl-keystore** can be used to re-create the key/certificate (stored in a keystore file) for a different hostname eliminating the browser domain mismatch warning. An example of the command's use:

```
cd /opt/savapage/server/bin/linux-x64
./create-ssl-keystore -f --default --common-name "savapage.mycompany.com"
```

More information is available via the `--help` command line option.

```
usage: [OPTION]...
    --common-name <NAME>   Subject and Issuer CN of the SSL Certificate (optional).
                           Default: server hostname [hostname].
    --create <FILE>        Creates a specific keystore file.
 -d,--default              Creates keystore file in default location
                           /opt/savapage/server/data/default-ssl-keystore.
    --dns-name <LIST>      Comma-separated list of DNS names to use as SSL
                           Certificate Subject Alternative Name (optional). Default:
                           value of --common-name if specified, if not
                           [hostname,hostname.local,hostname.local.,localhost].
 -f,--force                Force. Overwrites any existing keystore file.
 -h,--help                 Displays this help text.
    --ip-addr <LIST>       Comma-separated list of IP addresses to use as SSL
                           Certificate Subject Alternative Name (optional). Default:
                           server IPv4/IPv6 addresses [ip-addr,127.0.0.1,::1]
    --localhost            Creates a keystore for [localhost,127.0.0.1,::1] only.
                           (this option is incompatible with --common-name,
                           --dns-name and --ip-addr)
    --no-ca                SSL Certificate Issuer is not Certificate Authority (CA).
```

Important

`localhost` is always present as Subject Alternative Name, whatever the options are. This host is needed because local clients use it to send messages to the SavaPage server.

Note

DNS name `hostname.local.` belongs to top-level domain `.local.` and is the Multicast DNS (mDNS) host name according to RFC 6762[7].

[7] https://datatracker.ietf.org/doc/html/rfc6762

Note

The Issuer in the self-signed certificate is also the Certificate Authority (CA). In this way the certificate can be exported/imported as trusted CA in a browser. Use the `--no-ca` switch if you do not want the issuer to be CA.

Tip

If the Self-Signed Certificate (or its password file) is not found at server startup it is re-created ad hoc with default values described above. So, if you encounter an "HTTP ERROR 400 Invalid SNI" error when accessing a SavaPage webapp, re-create the certificate file with tailored values or remove the file, restart the server and try again. Also see Section 17.6, “SSL Certificate and Password” [291].

C.7.2. Importing an Existing SSL Certificate

If you have an existing SSL certificate you can import it into a Java keystore to be used by SavaPage. Reasons for having an existing signed key include:

- You have obtained a dedicated SSL certificate for use with your SavaPage Application Server.
- Your organization's intranet as served by Internet Information Server (Windows), Apache (GNU/Linux) or another web server uses a certificate that can be re-used for SavaPage.

Note

Unless your intranet server and SavaPage run on the same server (i.e. on different ports), the server name of your intranet server will be different from your SavaPage Application Server. E.g. the intranet address might be `internal.mycompany.com` while the SavaPage Application Server can be reached at `savapage.mycompany.com`. In this case the certificate can only be re-used if it is a so-called wild-card certificate that allows arbitrary subdomains under the `mycompany.com` domain name that it was issued for.

If the SSL certificate is held in a Windows environment you will have a certificate with an attached private key in a so-called PCKS #12 file with `*.p12` or `.pfx` extension[8]. Please convert this PCKS #12 file to a separate `PEM` private key and `PEM` certificate.

Note

If the certificate with key exist in the certificates store of Windows or IIS Server, you need to export it as a .PFX file first.

On GNU/Linux you will typically already have separate PEM encoded[9] key and certificate files. In this example, they are called `your_private_key.pem` and `your_certificate.pem` respectively.

However, we are not quite done yet, since we should add the intermediate certificate(s) of the Certificate Authority to the keystore as well. These certificates should be supplied by your CA or are available for download on the CA's web site as a file ending with `.pem` or `.crt`. A single PEM file has to be made, containing your certificate and all the intermediate certificates of your CA.

Use these commands to combine your certificate and the intermediates into one `PEM` file:

[8] PKCS #12 is one of the family of standards called Public-Key Cryptography Standards (PKCS), published by RSA Laboratories. It defines a (binary) file format commonly used to store X.509 private keys with accompanying public key certificates, protected with a password-based symmetric key, and is the successor to PFX from Microsoft.

[9] PEM or Privacy Enhanced Mail is a Base64 encoded DER certificate. PEM certificates are frequently used for web servers as they can easily be translated into readable data using a simple text editor. Generally when a PEM encoded file is opened in a text editor, it contains very distinct headers and footers.

```
cat your_certificate.pem > savapage.pem
cat intermediate_cert_1.pem >> savapage.pem
cat intermediate_cert_2.pem >> savapage.pem
[...]
```

Use this command to combine the private key and the certificates into a single `.p12` file:

```
openssl pkcs12 -export -inkey your_private_key.pem -in savapage.pem -out savapage.p12

Enter pass phrase for your_private_key.pem: (Enter your private key password, if present)
Enter Export Password:                      (Make up a password)
Verifying - Enter Export Password:          (Repeat this password)
```

The **keytool** command used in this section is part of the OpenJDK package as installed on the host. Now, use this command to create a new Java keystore and import the `.p12` or `.pfx` file at hand:

```
keytool -importkeystore -destkeystore my-ssl-keystore.jks -srckeystore savapage.p12 \
-srcstoretype PKCS12

Enter destination keystore password:        (Make up another password)
Re-enter new password:                      (Repeat this password)
Enter source keystore password:             (Enter Export Password from the previous openssl command)
```

Note

keytool “Warning: The JKS keystore uses a proprietary format” can be ignored.

At this point your keystore file is ready to use, so follow the instructions in Section C.7.3, “Installing the Keystore” [359] to install it and start serving up your new SSL certificate.

C.7.3. Installing the Keystore

The previous section described how to create a keystore file from an existing SSL certificate. This section describes how to install your keystore so that SavaPage can start serving up your new certificate.

To configure the SavaPage Application Server to use the new key/certificate:

1. Copy your signed keystore onto the server running the SavaPage Application Server. The suggested location is `/opt/savapage/server/custom/my-ssl-keystore.jks`
2. Open the file `/opt/savapage/server/data/server.properties` with a text editor.
3. Locate the section titled `SSL/HTTP Configuration`.
4. Remove the # (hash) comment marker from all lines starting with "`server.ssl`".
5. Define the location of your keystore, keystore password and key password as chosen previously. The file should look something like this:

   ```
   server.ssl.keystore=custom/my-ssl-keystore.jks
   server.ssl.keystore-password=password
   server.ssl.key-password=password
   ```

6. Restart the SavaPage Application Server and verify all is working. If the server fails to start, error messages will be recorded in logs located in the server's logs directory.

Note

Section 13.5.3.1, “Alternative TCP/IP Interface and Ports” [249] explains how IP ports are assigned to internal self-signed and external SSL certificates.

Appendix D. Capacity Planning

This section discusses capacity planning considerations so administrators can plan future infrastructure requirements and make decisions about how to deploy the application. SavaPage is designed to be self-maintaining, however it is important that the administrator understands the disk-space requirements and how this changes overtime.

D.1. Database Sizing and Growth

Special attention is needed to make sure there is enough disk space to hold a growing database. Database growth is very dependent on the usage patterns and therefore can differ significantly from site to site.

Although, there is some overhead for data like Users, Proxy Printers and Queues, this data is static and does not grow over-time. The majority of database growth is caused by logging the document flow.

So, the best prediction of database growth is based on the estimated number of documents printed to SavaPage and Proxy Printers, and exported to PDF.

The table below provides an indication of growth per 10,000 jobs for SavaPage and Proxy Printing and PDF export. Combining these numbers with your estimate of user activity result in a growth estimate.

Job type	Increase per 10,000 jobs
SavaPage printing	15 MB
Proxy Printer printing	20 MB
SavaPage Financial	5 MB
PDF export	20 MB

Table D.1. Database size increase metrics per document flow.

To demonstrate how to estimate database growth we make a number of assumptions. Please adjust these assumptions to suit your organization. The assumptions are:

- 1 job for each job type per user per day
- 20 working days in a month
- Therefore, 20 jobs for each job type per user per month

Here is a sample database growth calculation based on a 500 user site:

1. Calculate the total number of jobs expected for the month (i.e. the total number of users multiplied by the number of jobs). So in this example, SavaPage is handling 10,000 jobs for each job type a month.
2. Calculate the monthly growth rate by dividing the jobs per month by 10,000 and then multiplying by the database growth rate:
   - SavaPage printing : `10,000/10,000*15=15MB` per Month
   - Proxy Printer printing: `10,000/10,000*20=20MB` per Month
   - SavaPage Financial: `10,000/10,000*5=5MB` per Month
   - PDF export: `10,000/10,000*20=20MB` per Month

Therefore in this situation the database will grow by approximately `15+20+5+20=60MB` per month.

3. To estimate the growth per year, multiply the above by 12. Therefore in this situation, the database will grow by `60*12=720MB` per year.

Tip

You can limit database growth by purging old log data after an automatic database backup. In our example, when you set the number of days document logs are held to 365, database increase will maximize to 720MB. See Figure 4.128, “Admin Web App: Options - Automatic Backups” [178]

D.2. SafePages Sizing and Growth

SafePages are transient. They serve as scratchpad to accumulate and edit SavaPage print jobs. After ProxyPrinting and PDF exporting is done the user will normally clear the scratchpad for a new session. Of course this does not hold for personal Letterheads. Take in consideration that SafePages (including Letterheads) from logically deleted users are removed. This all makes the calculation of required disk space a simple linear function of the number of active users times the average size of a user's SafePages home. Since an average 10-page print job takes about 1 MB to hold in store, making a reservation of 10 MB per user seems fair enough.

D.3. Network Bandwidth Planning

With modern switched Ethernet networks, bandwidth is rarely a factor when planning SavaPage deployments. The bandwidth consumed by SavaPage is usually dwarfed by the print document data - e.g. the PostScript and PDF spool data sent across the network. Bandwidth does however become a consideration when planning deployments crossing physical site boundaries such as networks linked via a WAN.

SavaPage uses JSON based HTTP Requests for communication between browser-to-server (Ajax)[1] and server-to-browser (Comet)[2]. This protocol is very bandwidth efficient and designed to work well on low bandwidth and high latency networks.

[1] *Ajax* (an acronym for Asynchronous JavaScript and XML) is a group of techniques to create asynchronous web applications. With Ajax, web applications send data to, and retrieve data from, a server asynchronously using an XMLHttpRequest object. Despite the name, the use of XML is not needed, and JSON is often used instead. Also, requests do not need to be asynchronous.

[2] *Comet* (or “Reverse Ajax”) is a web application model in which a long-held HTTP request allows a web server to push data to a browser, without the browser explicitly requesting it.

Appendix E. URL Cheat Sheet

> **Note**
>
> For Internet access restrictions see Section 17.3, “Access over Internet” [287].

Path	Description / Parameters / Examples
`/`	User Web App `https://savapage:8632/`

`user`	User Web App
	
	`sp-user=[user]`
	`sp-lang=[de|en|es|fr|nl|ru|sv|zh|..]`
	`sp-ctry=[TW|..]`
	`sp-login=[name|email|id|nfc-local|yubikey]`
	`sp-login-local (exclude OAuth, use local login modes only)`
	`sp-log=[warn|info|debug]`
	
	`https://savapage:8632/user?sp-user=tom&sp-lang=en`
	`https://savapage:8632/user?sp-user=tom`
	`https://savapage:8632/user?sp-lang=zh&sp-ctry=TW`
	
	`Login with Username, Email, ID Number, Local NFC Card or YubiKey:`
	
	`https://savapage:8632/user?sp-login=name`
	`https://savapage:8632/user?sp-login=email`
	`https://savapage:8632/user?sp-login=id`
	`https://savapage:8632/user?sp-login=nfc-local`
	`https://savapage:8632/user?sp-login=yubikey`
	`https://savapage:8632/user?sp-login-local`
	
	`Verification link in Self Registration email`
	
	`https://savapage:8632/user?sp-verify=UUID`

`oauth/[provider]/`
 	
`user/oauth/[provider]`
 	
`oauth/[provider]/[id]`
 	
`user/oauth/[provider]/[id]`
	User Web App login with OAuth [provider] and optional plug-in [id].
	
	`https://savapage:8632/oauth/azure`
	`https://savapage:8632/oauth/keycloak`
	`https://savapage:8632/oauth/google`
	`https://savapage:8632/oauth/smartschool`
	
	`https://savapage:8632/user/oauth/google`
	`https://savapage:8632/user/oauth/smartschool`
	`https://savapage:8632/user/oauth/smartschool/myschool`

`admin`
 	
`jobtickets`
 	
`mailtickets`
 	
`payment`
	Admin Web App
	
	Job Tickets Web App
	
	Mail Tickets Web App
	
	Payment Web App

Path	Description / Parameters / Examples

`pos`
 	
`printsite`
	Point-of-Sale Web App
	
	Print Site Web App
	
	All Web Apps
	`sp-user=[user]`
	`sp-lang=[de|en|es|fr|nl|ru|..]`
	`sp-ctry=[DE|US|ES|FR|NL|RU|..]`
	`sp-login=[name|id|nfc-local|yubikey]`
	`sp-log=[warn|info|debug]`
	
	`https://savapage:8632/admin?sp-user=admin`
	`https://savapage:8632/jobtickets?sp-user=mary`
	`https://savapage:8632/pos?sp-user=dmitri&sp-lang=ru`
	`https://savapage:8632/printsite?sp-login=id`
	
	Point-of-Sale Web App
	`sp-sales-location=<sales-label-location-mnemonic>`
	`sp-sales-shop=<sales-label-shop-mnemonic>`
	`sp-sales-card-as-password=[OFF|ON|AUTO]`
	
	`https://savapage:8632/pos?sp-sales-location=A&sp-sales-shop=L`
	
	See Section 7.5, “Configuration Properties” [230]

`printers/[queue]`
 	
`printers/`
	Printer Queue
	
	Default Printer Queue
	
	`ipp://savapage:8631/printers/` ❶
	`ipps://savapage:8632/printers/` ❷
	`http://savapage:8631/printers/` ❸
	`https://savapage:8632/printers/` ❹
	
	❶ IPP 1.1 scheme: supported by all major operating systems except Windows.
	❷ The SavaPage SSL certificate needs to be trusted by the client workstation[a] . See Section C.7.3, “Installing the Keystore” [359].
	❸ IPP 1.0 scheme: supported by all major operating systems.
	❹ The SavaPage SSL certificate needs to be trusted by the client workstation. See Section C.7.3, “Installing the Keystore” [359]
	
	In response to an IPP Get-Printer-Attributes request the “printer-name” is returned as “SavaPage” (default queue) or “SavaPage-[queue]”.

`printers/internet/[number]/[uuid]`	The Printer Device URI path for Internet Print. Parameters are *not* query parameters but are part of the path.
	• `[number]` : the User ID Number.
	• `[uuid]` : the User UUID.
	
	`ipps://example.com/printers/internet \`
	`  /1111/b0a2f092-8c5b-11e5-a6fb-406186940c49`

`ios/install`	iOS Web Clip Install `https://savapage:8632/ios/install`
`docs/manual`	User Manual
`docs/licenses`	License Information
`callback/`	Web API Callback

Path	Description / Parameters / Examples
`callback/payment/[live|test]/[pluginId]`	Web API Callback for Payment Gateway
`client/`	Download of shared Client Files. A link to a directory downloads the zipped content.
`xmlrpc/`	The secure only XML-RPC endpoint: `https://savapage:8632/xmlrpc` See Section C.2.1, “XML-RPC” [342].
`ext/papercut/`	Section N.5.1, “SavaPage as Sync Source in PaperCut” [417].
`feed/admin/`	Section C.3, “Atom Feed Service” [352].
`verify/pdf/`	Section 10.2, “PDF/PGP Signature” [237].

[a] When the SSL certificate is not trusted a "client-error-not-possible" situation will occur when adding the printer.

Table E.1. SavaPage URL Cheat Sheet

Appendix F. File Locations

The table below summarizes the SavaPage file locations.

> **Important**
>
> When overriding defaults, make sure the location for SavaPage temporary files resides on the same disk partition as the locations used to store data on runtime. See:
> - Section 13.5.13, “Alternative File Locations” [256].
> - Section 22.3.3, “JVM Temporary Files” [319].

Path	Description
`/opt/savapage/`	Install directory.
`client/`	Home of client applications. Sibling of home of server application, or passed as system property `client.home` at the start of a `java` client application.
`app/`	Chapter 11, *User Client* [240].
`config/`	Section 11.2, “User Client Deployment” [242] • `client.properties`
`jmx/`	Section 4.11.14.4, “JMX Agent” [180].
`linux/`	.desktop templates.
`providers/`	Section 1.3.1.6, “Information Provider” [7]
`cups/`	Section 13.5.7.1, “CUPS Notifier” [253].
`nfc/`	Section B.3, “Network Card Reader Service” [327].
`server/`	Home of server application. Passed as system property `server.home` at the start of the `java` server application.
`bin/`	• Section C.1, “Server Commands” [328]. • Section C.5, “Database Commands” [353].
`custom/`	Section 22.3, “JVM Tuning” [318] • `app-server.conf` Section C.7, “SSL Key Generation” [356] • `my-ssl-keystore.jks` Section 20.1, “Custom Web App” [300] • `web.properties`
`cups/`	Appendix K, *PPD Extensions* [382]

Path	Description
`i18n/`	Section L.3, “IPP Localization” [404]
`html/`	HTML injectable files.
`i18n/`	Section 20.1.1.4, “Custom i18n” [303] Section L.3, “IPP Localization” [404]
`rawprint/`	Raw Print files.
`ppd/`	Raw Print PPD files.
`transforms/`	Raw Print PDL Transform files.
`template/`	Section 20.2, “Email Templates” [304]
`web/`	Section 20.1.1, “Web App Look-and-feel” [300].
`themes/`	Section 20.1.1.1, “Web App Theming” [300].
`data/`	Section 17.2.1.3, “Internal Admin Password” [286] • `admin.properties` Section 17.6, “SSL Certificate and Password” [291] • `default-ssl-keystore.*` Section 17.8, “Encrypted Secrets” [292] • `encryption.properties` Section J.1.6, “Google Cloud Directory” [380] • `google-ldap-client-cert.*` Section 4.11.14.4, “JMX Agent” [180] • `jmxremote.*` Log4j configuration • `log4j.properties` Section 4.14.4, “Community” [201] • `savapage.membercard` Section 13.5, “Advanced Configuration” [248] • `server.properties`
`.tmp/`	Default location for temporary files, created when the application starts and removed when stopped. Location can be changed with the `app.dir.tmp` property in the `server.properties` file. See Section 13.5.13, “Alternative File Locations” [256].
`backups/`	Section 4.11.13, “Backups” [177].
`conf/`	Section 4.11.1.4, “Internal Groups” [154] • `internal-groups.txt` Section 15.1.18, “User Alias” [276] • `username-aliases.txt`
`doc-archive/` `out/print/`	Print Archive

Path	Description

`[CCYY]/`
`[MM]/`
`[DD]/`
`[HH]/`
`[UUID]`
	Printed PDF and Job Options (in JSON format) are stored in a unique identifying file path of year, month, day, hour and UUID of the print job.
	
	Also see Section 13.5.13, “Alternative File Locations” [256].

`doc-journal/`
`out/print/`
`[CCYY]/`
`[MM]/`
`[DD]/`
`[HH]/`
`[UUID]`
	Print Journal
	
	Analogous to `doc-archive`.

`doc-journal/`
`in/print/`
`[CCYY]/`
`[MM]/`
`[DD]/`
`[HH]/`
`[UUID]`
	Queue Journal
	
	Analogous to `doc-jounal/out`.

`email-outbox/`	Outgoing email queue.
`internal/`	
`Derby/`	Section 1.2.1.4, “Database” [4].
`letterheads/`	Section 3.6, “Letterheads” [68] Also see Section 13.5.13, “Alternative File Locations” [256].
`safepages/`	Section 3.3, “SafePages” [29]. Also see Section 13.5.13, “Alternative File Locations” [256] and Section 13.5.10, “User Home Clean” [255].
`print-jobtickets/`	Chapter 5, *Job Tickets Web App* [211].
`docs/`	Accessible via URL: Appendix E, *URL Cheat Sheet* [362]
`examples/`	Section 14.1, “IPP Everywhere and AirPrint” [260]
`data/`	Configuration file templates that may contain additional features compared to their `/opt/savapage/data` counterparts.
`papercut/`	Section N.5.1, “SavaPage as Sync Source in PaperCut” [417]
`ext/`	Appendix M, *SavaPage Plug-ins* [405] • `savapage-ext-*.properties`
`lib/`	Extension JAR files.
`logs/`	Extension log files.
`lib/`	Main WAR and JAR files including supporting SQL scripts.

Path	Description
`logs/`	Main log files.

Table F.1. SavaPage File Locations

Appendix G. Printable File Types

G.1. Standard File Types

SavaPage printer supports a number of common file types out of the box as summarized in Table G.1, “Standard Printable File Types” [369].

Important

Non-embedded fonts in original or produced PDF documents are subject to Font Substitution.

Extension	Type
`pdf`	**Portable Document Format** • The PDF must not be password protected and not have dynamic XFA[a] form content. • For acceptance of encrypted PDF the Allow Encrypted PDF for Proxy Printing option must explicitly be enabled. • Non-embedded fonts are subject to PDF Font Substitution.
`ps`	**PostScript** DRM protected PostScript is rendered for ProxyPrinting only. See Section 14.10.2, “Printing Encrypted PDF” [272].
`oxps`, `xps`	**Open XML Paper Specification** See Section G.1.1, “XPS to PDF Installation Instructions” [370].
`html`, `htm`	**Hypertext Markup Language**
`txt`	**Text File**
`bmp`	**Bitmap**
`gif`	**Graphic Interchange Format** For Animated GIF each image is rendered separately.
`heif`, `heic`	**High Efficiency Image File Format** See Section 1.2.1.8, “heif-convert” [5].
`jpg`, `jpeg`, `jpe`	**JPEG/JIFF Image**
`png`	**Portable (Public) Network Graphic**
`md`	**Markdown**
`svg`	**Scalable Vector Graphics** rsvg-convert is needed for this file type.
`tiff`, `tif`	**Tagged Image Format File**

Extension	Type
	Multi-page tiff is supported.
`eml`	**Electronic Mail Format** EML is a file extension for an email message saved to a file in the Internet Message Format protocol for electronic mail messages. EML files comply with industry standard RFC 5322[1] and can be exported from most email clients.

[a] XML Forms Architecture (XFA) is a proprietary format for forms introduced by Adobe in PDF 1.5 that is not compatible with ISO 32000's AcroForms feature. Most PDF processors do not handle XFA content. The XFA specification is referenced from ISO 32000-1 / PDF 1.7 as an external proprietary specification, and was entirely deprecated from PDF with ISO 32000-2 (PDF 2.0). Dynamic PDF-XFA can change its layout and appearance based on the data entered by the user. Static forms have a fixed layout.

Table G.1. Standard Printable File Types

Note

The Default Paper Size, as shown in Figure 4.136, “Admin Web App: Options - Default Paper Size” [182], is used as the paper size for the printed document of a Printable File Type which itself does not have a document structure with a clearly defined page size. These types typically include HTML, TXT and *images*.

G.1.1. XPS to PDF Installation Instructions

Open XML Paper Specification (XPS) is an XML based electronic paper format originally developed by Microsoft to serve as a PDF alternative. XPS files are usually created using "Microsoft XPS Document Writer" in a Windows environment.

SavaPage uses the **xpstopdf** command from the libgxps[2] package to convert XPS documents to PDF format. Check if this package is installed by entering the command: `xpstopdf --help`

On Debian based systems you can install the package with the command:

```
sudo apt install libgxps-utils
```

Note

Before XPS to PDF can be used it must be enabled. See Figure 4.138, “Admin Web App: Options - Converters” [183].

G.2. Advanced File Types

SavaPage printer supports additional file types using the PDF converter of LibreOffice as summarized in Table G.2, “Advanced Printable File Types” [371]. Check if LibreOffice is installed by entering the command:

```
libreoffice --version
```

LibreOffice can easily be installed with the standard installer of the GNU/Linux host. On Debian based systems you can use the command line to install the packages needed. For example:

```
# The core package
sudo apt install libreoffice-core unoconv

# To convert text, spreadsheet and presentation documents
```

[1] https://datatracker.ietf.org/doc/html/rfc5322
[2] libgxps [https://wiki.gnome.org/libgxps] is a library for handling and rendering XPS documents.

```
sudo apt install libreoffice-writer libreoffice-calc libreoffice-impress
```

Extension	Type
rtf	**Rich Text Format**
doc	**Microsoft Word 97/2000/XP/2003**
xls	**Microsoft Excel 97/2000/XP/2003**
ppt	**Microsoft PowerPoint 97/2000/XP/2003**
docx	**Microsoft Word 2007/2010 XML**
xlsx	**Microsoft Excel 2007/2010 XML**
pptx	**Microsoft PowerPoint 2007/2010 XML**
odt	**ODF Text Document**
ods	**ODF Spreadsheet**
odp	**ODF Presentation**
sxw	**OpenOffice.org 1.0 Text Document**
sxc	**OpenOffice.org 1.0 Spreadsheet**
sxi	**OpenOffice.org 1.0 Presentation**

Table G.2. Advanced Printable File Types

Warning

PDF conversion of Microsoft documents may not give correct results in all cases.

Note

Before LibreOffice can be used it must be enabled. See Figure 4.138, “Admin Web App: Options - Converters” [183].

G.3. PDF Font Substitution

The `fontconfig`[3] package, installed on the GNU/Linux host, is used by the PDF printing renderer to get the best matching installed font as substitute for non-embedded fonts in the PDF document.

The PDF "Standard 14 fonts"[4] aka "Base 14 Fonts" are never embedded. Every PDF renderer is required to provide these fonts, either directly or via appropriate substitutes. Current substitutes are shown in the Admin Web App, About Host Packages section.

When a PDF document contains non-embedded fonts that are not from the Standard 14 fonts, the substitute fonts selected by `fontconfig` may not be the same as the fonts used to create the PDF, and so rendering may not be correct. Therefore, it is wise to install additional fonts, especially the much used Microsoft True Type Core Fonts with Andale Mono, Arial, Comic Sans MS, Courier New, Georgia, Impact, Times New Roman, Verdana, etc.

```
# DejaVu provides an expanded version of the Vera font family aiming for
```

[3] https://fontconfig.org/
[4] The "Standard 14 Fonts” of PDF documents are described in the ISO Standard 32000-1:2008(E) Section 9.6.2.2.

```
# quality and broader Unicode coverage while retaining the original Vera style.
sudo apt install fonts-dejavu

# Other fonts for more accurate matching, so non-embedded fonts
# don't fallback to DejaVu font.

# Microsoft True Type Core Fonts
sudo apt install ttf-mscorefonts-installer

# Free look-alike fonts of the Adobe PostScript fonts:
# Nimbus Roman, Nimbus Mono, Nimbus Sans, Century Schoolbook, Dingbats,
# Standard Symbols, URW Chancery, URW Palladio, URW Gothic, URW Bookman
sudo apt install gsfonts

# Cantarell sans serif font family designed for on-screen readability.
sudo apt install fonts-cantarell

# Width-compatible fonts for improved on-screen readability
# Arimo, Cousine and Tinos are metrically compatible with Arial,
# Courier New and Times New Roman, respectively.
sudo apt install fonts-croscore

# Caladea is metric-compatible with Cambria font.
sudo apt install fonts-crosextra-caladea

# Carlito is metric-compatible with Calibri font.
sudo apt install fonts-crosextra-carlito

# Extensive character set coverage including Western Europe, Eastern/Central Europe,
# Baltic, Cyrillic, Greek and Turkish support. The Droid Sans regular font also
# includes support for Arabic, Simplified and Traditional Chinese, Hebrew, Japanese,
# Korean and Thai.
sudo apt install fonts-droid-fallback

# Fonts with the same metrics as Times, Arial and Courier
sudo apt install fonts-liberation

# "Noto" is short for "No Tofu", describing the aim of covering
# all living Unicode scripts.
sudo apt install fonts-noto

# OpenSymbol TrueType font included in LibreOffice.
sudo apt install fonts-opensymbol
```

Appendix H. Upgrading from a Previous Version

This appendix describes the SavaPage standard upgrade procedure.

H.1. Upgrading the Server

SavaPage supports upgrades using a simple *install-over-the-top* procedure. We recommend reviewing all steps prior to commencing the upgrade procedure.

1. Download the SavaPage installer for your platform. In accordance with best practice we recommend that you archive your install programs just in case you need to reinstall in the future or roll back to a previous version.
2. Take some time to read the release notes for this version as they may highlight considerations during upgrades.
3. Schedule approximately 10 minutes downtime. It is suggested to choose a time of day with minimal network activity. If there is a large volume of data in the system (for example if the system has been running for more than a year, or there are several thousands of users) the upgrade may take longer. With very large installations it may be appropriate to schedule an hour or more of downtime.
4. Take a point-in-time backup of the data by pressing the Backup Now located under Options → Backups. This will ensure you have a copy of the important data.
5. As a precaution on very large systems, we recommend backing up the whole SavaPage install directory. Existing overnight backups may have taken care of this task, however take a few moments to grab an up-to-date backup now. For example, create a zip archive of the directory `/opt/savapage`
6. Run the installer downloaded in step 1 and install into the *same* location as the existing install, like /opt/savapage.
7. After the install has completed allow a few minutes before accessing the system. The system may need to perform a database upgrade and this will be performed in the background. If you try to access the application while a database upgrade is in progress a message displaying the upgrade status will be displayed.

Important

Do not shutdown the application while an upgrade is in progress. Wait for the upgrade to complete.

Note

Sometimes a new SavaPage version performs changes on the database schema. In that case a database backup is performed automatically before the upgrade. The backup file is located at `/opt/savapage/server/data/backups/`. The file name is formatted as `schema-[nn]-upgrade-backup-[timestamp].zip`, where `[nn]` is the database schema version before the upgrade.

H.2. Upgrading Client Printer Drivers

Although upgrading locally installed SavaPage Printer Drivers is not strictly required, we strongly recommend doing so. We strive to maintain backwards compatibility between versions, so in most cases these drivers will continue to function, but to take advantage of new features they must be upgraded.

H.3. Testing the Upgrade

After the install is complete, log into the system and perform some tests to ensure all is working as expected.

Appendix I. Migrating to a New Server

Migrating to a new server is a major task. Administrators should block out at a minimum two hours, and should select a time where downtime will be of minimum disruption to end-users.

This section describes how to migrate SavaPage to a new system so that all data is moved to the new system. To ensure a smooth migration it is strongly recommended that the versions of SavaPage on both the old and new servers are the same. The easiest way to achieve this is to upgrade the old server to the latest version, and then install the latest version on the new server.

Please read the sections below in full before conducting your migration.

I.1. Upgrade Old Server

Upgrade the old server to the latest version:

1. Download the latest available version available from the `SavaPage Website`[1]
2. Install the upgrade by following the steps in Appendix H, *Upgrading from a Previous Version* [373].
3. After the upgrade is complete, check that everything is working as expected.

I.2. Install New Server

Install the latest version of SavaPage on the new server.

1. Make your CUPS printers on the new server identical as the ones on the old server, and test the CUPS queues before installing SavaPage. Use exactly the same names for the CUPS queues. If you deviate you might need to rename Proxy Printers after installation, as explained in Section I.5, “Rename Printers” [375].
2. Follow the instructions at Chapter 2, *Server Installation* [11]. Complete the configuration steps, including the user import. Although importing the users is not strictly required, as this data will be overridden after data migration, it does confirm that your new server has the correct network connectivity. Of course you are free to synchronize with a smaller user group to proof the connectivity.
3. Compare the content of the `server.properties` file from the old server with the one on the new server, and change the file on the new server where needed.
4. Import your Service Card file into the new server. See Section 4.14.4, “Community” [201].

I.3. Freeze Old Server

Prevent creation of new Proxy Print jobs and bring all active CUPS jobs to end-state. For more information about the reason for this action see Section 2.3.3, “CUPS Job ID” [13].

1. Put SavaPage into Maintenance Mode to block user access.
2. Wait for print jobs to complete. Cancel all jobs that can't complete. Check the Documents Log: no print job should be active.

I.4. Migrate Data to New Server

The simplest way to migrate the data to the new server is to use the backup and restore process.

[1] https://www.savapage.org

1. Backup the data from the old system using the Admin Web App, or see the instructions at Section C.5.6, "db-export and db-export-to" [355] for the command-line variant.
2. Copy the backup zip file created in the backup step onto the new server.
3. Stop the SavaPage Server by running the stop script. See Section C.6, "Stopping and Starting the Server" [356].
4. Restore the data from the old system into SavaPage on the new server by following the Database Import instructions. The import commands need to be run as the `savapage` user.
5. Migrate the database Encryption Keys by copying the `encryption.properties` file at location `/opt/savapage/server/data/` from the old server to the same location on the new server. Overwrite the existing file on the new server. Make sure this file is owned by `savapage` , and restrict access by executing:

   ```
   sudo chown savapage:savapage encryption.properties
   sudo chmod 600 encryption.properties
   ```

6. If present, migrate the User Name Aliases and Internal Groups by copying the `username-aliases.txt` and `internal-groups.txt` files at location `/opt/savapage/server/data/conf/` from the old server to the same location on the new server. If the alias file depends on users from a User Source on the local Unix system, be sure that these users also exist on the new server.
7. If present, migrate any messages in the email outbox by copying the files at location `/opt/savapage/server/data/email-outbox/` from the old server to the same location on the new server.
8. If present migrate the Document Store from the old server to the same location on the new server.
9. Migrate all customization files to the new server. See Chapter 20, *Customization* [300] and Appendix K, *PPD Extensions* [382].
10. Migrate all plug-in files to the new server (except for the `lib/.jar` files already present on the new server). See Appendix M, *SavaPage Plug-ins* [405].
11. Check if all essential files are migrated: see Appendix F, *File Locations* [365]. Take special attention to Section 13.5.13, "Alternative File Locations" [256].
12. Start the SavaPage Server by running the start script. See Section C.6, "Stopping and Starting the Server" [356].
13. Check that all data has been migrated correctly and the system works as expected by comparing Users and Documents data in the old and new Admin Web App.

I.5. Rename Printers

If you changed the CUPS printer names on the new server, you may want to rename the existing Proxy Printer entries in SavaPage so that the printing history and settings are maintained. See Figure 4.77, "Admin Web App: Proxy Printer - Rename" [139] for details about the rename action.

I.6. Update SavaPage Printers

If the server's name and/or IP address has changed then it is necessary to update the connection details for SavaPage Printers on user workstations. See Section 14.2, "PostScript Driver Print" [262] for details.

Appendix J. Advanced LDAP Configuration

SavaPage supports the following LDAP server types out-of-the-box:

- OpenLDAP
- FreeIPA
- Apple Open Directory
- Novell eDirectory
- Microsoft Active Directory
- Google Cloud Directory

The basic configuration options for these types are discussed at Section 4.11.1.2, “LDAP” [150]. However, other server/schema types can be supported by defining the fields to query and the LDAP searches to perform. These options are configured by adjusting entries in the Config Editor of the Admin Web App. The following configuration properties are available:

Configuration property	Description
ldap.schema.user-name-field	The LDAP field that contains the user's username. • Content of this field is converted to lowercase when synchronized with the SavaPage database.
ldap.schema.user-full-name-field	The LDAP field that contains the user's full name.
ldap.schema.user-email-field	The LDAP field that contains the user's email address. • Content of this field is converted to lowercase when synchronized with the SavaPage database.
ldap.schema.user-department-field	The LDAP field that contains the user's department.
ldap.schema.user-office-field	The LDAP field that contains the user's office location.
ldap.schema.user-name-search	The LDAP search to retrieve the user. The `{0}` in the search is replaced with `*` when listing all users, and `[username]` when searching for a specific user. If no search is defined the default is `(userNameField={0})`, where `userNameField` is the value of configuration property "ldap.schema.user-name-field". IMPORTANT: The search must include the `{0}` value and involve the LDAP field as specified in configuration property "ldap.schema.user-name-field".
ldap.schema.group-name-field	The LDAP field that contains the group's name.
ldap.schema.group-full-name-field	The LDAP field that contains the group's full name.
ldap.schema.group-member-field	The LDAP field that contains the group members.
ldap.schema.group-search	The LDAP search to retrieve the group. The `{0}` in the search is replaced with `*` for all group searches. If no search is defined, the default is `([groupMemberField]={0})`, which means get all entries with at least one member.

Configuration property	Description
	IMPORTANT: The search must include the `{0}` value.
ldap.schema.posix-groups	If `Y`, then the group member field contains the user's username. If `N`, then the group member field contains the user's DN.

Table J.1. LDAP Configuration Properties

J.1. LDAP Server Default Configuration

When a particular LDAP server type is selected (e.g. Novell eDirectory), SavaPage uses the following defaults to query the LDAP server. These defaults can be used as a starting point for customizing the LDAP searches or for supporting other server types.

J.1.1. OpenLDAP

If the LDAP server is configured to support OpenLDAP[1] based authentication then this schema type can be used. The following defaults are used.

Configuration property	Default value
ldap.schema.user-name-field	`uid`
ldap.schema.user-full-name-field	`cn`
ldap.schema.user-email-field	`mail`
ldap.schema.user-department-field	`departmentNumber`
ldap.schema.user-office-field *This item is not set.*	
ldap.schema.user-name-search	`(uid={0})`
ldap.schema.group-name-field	`cn`
ldap.schema.group-full-name-field	`displayName`
ldap.schema.group-member-field	`member`
ldap.schema.group-search	`(&(cn={0})(objectClass=groupOfNames))`
ldap.schema.posix-groups	`N`

Table J.2. OpenLDAP Default Settings

J.1.2. FreeIPA

The defaults for FreeIPA[2] based authentication are identical to OpenLDAP.

J.1.3. Apple Open Directory

If the LDAP server is configured to support Apple Open Directory based authentication then this schema type can be used. The following defaults are used.

[1] https://www.openldap.org
[2] https://www.freeipa.org

Configuration property	Default value
ldap.schema.user-name-field	`uid`
ldap.schema.user-full-name-field	`cn`
ldap.schema.user-email-field	`mail`
ldap.schema.user-department-field	`departmentNumber`
ldap.schema.user-office-field *This item is not set.*	
ldap.schema.user-name-search	`(uid={0})`
ldap.schema.group-name-field	`cn`
ldap.schema.group-full-name-field	`displayName`
ldap.schema.group-member-field	`memberUid`
ldap.schema.group-search	`(memberUid={0})`
ldap.schema.posix-groups	`Y`

Table J.3. Apple Open Directory Default Settings

J.1.4. Novell eDirectory Defaults

If the LDAP server is a Novell eDirectory then the following defaults are used[3].

Configuration property	Default value
ldap.schema.user-name-field	`cn`
ldap.schema.user-full-name-field	`fullName`
ldap.schema.user-email-field	`mail`
ldap.schema.user-department-field	`OU`
ldap.schema.user-office-field	`l`
ldap.schema.user-name-search	`(&(cn={0})(objectClass=person))`
ldap.schema.group-name-field	`cn`
ldap.schema.group-full-name-field	`fullName`
ldap.schema.group-member-field	`member`
ldap.schema.group-search	`(&(member={0})(objectClass=groupOfNames))`
ldap.schema.posix-groups	`N`

Table J.4. Novell eDirectory Default Settings

J.1.5. Microsoft Active Directory Defaults

If the LDAP server is a Microsoft Active Directory then the following defaults are used[4].

[3] The list of standard Novell eDirectory user fields can be found on `NDK: Novell eDirectory Schema Reference [https://www.novell.com/documentation/developer/ndslib/schm_enu/data/h4q1mn1i.html#h4q1mn1i]`.
[4] The list of standard Active Directory user fields can be found on the Microsoft `Active Directory Schema [https://docs.microsoft.com/en-us/windows/desktop/ADSchema/active-directory-schema]` web site .

Configuration property	Default value
ldap.schema.user-name-field	`sAMAccountName`
ldap.schema.user-full-name-field	`displayName`
ldap.schema.user-email-field	`mail`
ldap.schema.user-department-field	`department`
ldap.schema.user-office-field	`physicalDeliveryOfficeName`
ldap.schema.user-name-search	`(&(sAMAccountName={0})(objectCategory=person)(objectClass=user)(sAMAccount-Type=805306368){1})` The extra `{1}` in the search is replaced with an optional filter to fetch enabled users only (see ldap.disabled-users.allow).
ldap.schema.group-name-field	`sAMAccountName`
ldap.schema.group-full-name-field	`displayName`
ldap.schema.group-member-field	`member`
ldap.schema.group-search	`(&(sAMAccountName={0})(objectCategory=group))`
ldap.schema.posix-groups	`N`

Table J.5. Microsoft Active Directory Default Settings

Configuration property	Default value / Description
ldap.disabled-users.allow	`N` If `Y`, then disabled users are accepted in user name searches. If `N`, they are ignored.
ldap.schema.dn-field	`distinguishedName` The LDAP field that contains the Distinguished Name (DN).
ldap.schema.user-name-group-search	`(&(memberOf={0})(objectCategory=person) (objectClass=user)(sAMAccountType=805306368){1})` This is the LDAP search to retrieve the users from a group. The `{0}` in the search is replaced with the DN of the user. The `{1}` in the search is replaced with an optional filter to fetch enabled users only (see ldap.disabled-users.allow). IMPORTANT: The search must include the `{0}` and `{1}` value.
ldap.schema.nested-group-search	`(&(memberOf={0})(objectCategory=group))` This is the LDAP search to retrieve the nested groups from a group. The `{0}` in the search is replaced with the DN of the group. IMPORTANT: The search must include the `{0}` value.

Table J.6. Microsoft Active Directory Custom Settings

Important

Active Directory field names must be in the `Ldap-Display-Name` format. For example, if you want to use the *Employee-Number* field, then the field name entered should be *employeeNumber* as shown on the Employee-Number attribute page `https://docs.microsoft.com/en-us/windows/desktop/ADSchema/a-employeenumber`.

J.1.6. Google Cloud Directory

The defaults for Google Cloud Directory based authentication are identical to OpenLDAP.

If your organization is allowed to use LDAP for G Suite or Google Cloud Identity services, you can create a certificate that clients like SavaPage can use for user synchronization and authentication. To create the certificate, log in to admin.google.com[5] as Super Admin and add an LDAP Client. Set your Access permissions, download the certificate zip file and save it in a secure location. Finally, switch the LDAP Service status ON in the Client Details. Note that it may take up to 24 hours for the LDAP Client to be active.

The downloaded certificate zip file name contains its expiration date and time: we use `Google_2022_10_27_37420.zip` as an example. The zip contains the PEM formatted public certificate and private key file.

```
$ unzip Google_2022_10_27_37420.zip

# Have a look at the certificate
$ openssl x509 -in Google_2022_10_27_37420.crt -text -noout

Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1563872646762 (0x16c1e12d66a)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: O=Google Inc., L=Mountain View, CN=LDAP Client, OU=GSuite, C=US, ...
        Validity
            Not Before: Oct 28 10:23:40 2019 GMT
            Not After : Oct 27 10:23:40 2022 GMT
        Subject: O=Google Inc., L=Mountain View, CN=LDAP Client, OU=GSuite, C=US, ...
        Subject Public Key Info:
    ....

# Check the private key
$ openssl rsa -in Google_2022_10_27_37420.key -check -noout
RSA key ok
```

Use the certificate and private key to create the `p12` container needed by SavaPage. In addition store the p12 export password in a separate file. Set strict read permissions on both files.

```
$ openssl pkcs12 -inkey Google_2022_10_27_37420.key -in Google_2022_10_27_37420.crt \
               -export -out google-ldap-client-cert.p12

Enter Export Password:
Verifying - Enter Export Password:

$ sudo chown savapage:savapage google-ldap-client-cert.p12
$ sudo chmod 400 google-ldap-client-cert.p12
$ sudo mv google-ldap-client-cert.p12 /opt/savapage/server/data
sudo su - savapage
$ cd server/data
```

[5] https://admin.google.com

```
# Create file with Export Password to unlock p12 file
$ echo "password=export-password" > google-ldap-client-cert.pw
$ chmod 400 google-ldap-client-cert.pw
```

If `google-ldap-client-cert.p12` and `google-ldap-client-cert.pw` are present in `/opt/sava-page/server/data` the Google Cloud Directory option will be available at Section 4.11.1.2, “LDAP” [150].

Appendix K. PPD Extensions

Vendor specific PPD option keywords are generally not mapped to IPP attributes by CUPS. That's why we do not get IPP attributes for finishings (staple, punch, fold, booklet) or collating delivered, when we ask CUPS for an IPP printer description.

To bridge this gap we built our own mapping by means of a so-called PPD Extension `.ppde` file. With this mapping we are able to identify printer capabilities based on IPP and feed CUPS the vendor specific PPD options as IPP attributes when sending a print job. When printing, these IPP disguised PPD options are neatly applied by CUPS in the context of the PPD driver, so the right PostScript / PCL snippets are injected in the spool file.

PPD Extension files reside in the `/opt/savapage/server/custom/cups/` directory. An annotated `type-model-version.ppde.template` file is installed there for your convenience. The `.ppde` file can be linked to a Proxy Printer. See Section 4.8.2, “Edit Proxy Printer” [132]. When linked, the mapped PPD options will appear in the Printer Settings Dialog.

Important

When the content of a PPD Extension File, assigned to any Proxy Printer, is changed, you must Synchronize the Proxy Printers to take those changes into effect.

Warning

PPD Extensions is an advanced feature. Consult your SavaPage Service Provider before implementing.

K.1. PPD to IPP Mappings

The PPDE file holds mappings of original PPD file options to their IPP attribute and values counterparts. Mapped IPP attributes and values can either be IANA[1] registered, or Internal SavaPage Extensions. This is the syntax:

```
*VENOption IppAttribute                  ❶
*VENOption *VENOptionValue-1 IppValue-1  ❷
...
*VENOption *VENOptionValue-n IppValue-n
```

❶ `VENOption` must be replaced by its `PPD` equivalent (the `VEN` prefix stands for *vendor* specific).

❷ `VENOption` / `VENOptionValue` pairs, whose values must be copied from the vendor `PPD`.

`IppValue` is the IPP value equivalent. The optional asterix `*` prefix of `VENOptionValue` tells if the value is the default. The `IppValue` *must* be unique in the `VENOption` set, while the `VENOptionValue` can be used more than once. *In this way different IPP values can be mapped to the same PPD value.*

Note

`VENOption` / `VENOptionValue` pairs are relevant for IPP attributes that have “keyword”, “enum” or “boolean” syntax, because their values are confined to a predefined set. These pairs are not needed for IPP attributes that have unconfined values, because of their “integer”, “name”, or “text” syntax.

[1] https://www.iana.org/assignments/ipp-registrations/ipp-registrations.xhtml

K.1.1. Mapping PPD to IPP

IANA IPP attributes[2] available for mapping are presented in the sections below.

K.1.1.1. copies

```
*VENCopies copies
```

K.1.1.2. media

```
*VENMedia media
*VENMedia  VENA3 iso_a3_297x420mm
*VENMedia *VENA4 iso_a4_210x297mm
*VENMedia  VENA5 iso_a5_148x210mm
*VENMedia  VENA6 iso_a6_105x148mm
# ...
```

K.1.1.3. media-source

```
*VENMediaSource media-source
*VENMediaSource *VENAuto auto
*VENMediaSource VENMain main
*VENMediaSource VENTop top
*VENMediaSource VENMiddle middle
*VENMediaSource VENBottom bottom
*VENMediaSource VENBypassTray by-pass-tray
*VENMediaSource VENManual manual
*VENMediaSource VENTray1 tray-1
# tray-2 ... tray-10
```

K.1.1.4. media-type

```
*VENMediaType media-type
*VENMediaType *VENPaper paper
*VENMediaType VENTransparency transparency
*VENMediaType VENLabels labels
*VENMediaType VENLetterhead letterhead
```

K.1.1.5. output-bin

```
*VENOutputBin output-bin
*VENOutputBin *VENAuto     auto
*VENOutputBin VENBottom    bottom
*VENOutputBin VENCenter    center
*VENOutputBin VENTop       top
*VENOutputBin VENFaceDown  face-down
*VENOutputBin VENFaceUp    face-up
*VENOutputBin VENLargeCap  large-capacity
*VENOutputBin VENLeft      left
*VENOutputBin VENMiddle    middle
*VENOutputBin VENRear      rear
*VENOutputBin VENSide      side
*VENOutputBin VENStacker1  stacker-1
# stacker-2 ... stacker-5
*VENOutputBin VENTray1 tray-1
# tray-2 ... tray-5
```

[2] https://www.iana.org/assignments/ipp-registrations/ipp-registrations.xhtml

K.1.1.6. print-color-mode

```
*VENPrintColorMode print-color-mode
*VENPrintColorMode *VENMonochrome monochrome
*VENPrintColorMode VENColor       color
```

K.1.1.7. print-scaling

```
*VENPrintScaling print-scaling
*VENPrintScaling *VENFit fit   ❶
*VENPrintScaling VENNone none  ❷
*VENPrintScaling VENAuto auto  ❸
```

❶ Scale the document to fit the printable area of the requested media size, preserving the aspect ratio of the document data without cropping the document.

❷ Do not scale the document to fit the requested media size. If the document is larger than the requested media, center and clip the resulting output. If the document is smaller than the requested media, center the resulting output.

❸ Value `auto` applies a 'fit', 'fill' or 'none' method to get an optimized imposition of the document onto the printable area of the requested media size. This is the best method to scale documents with mixed portrait/landscape pages.

See Section 3.5.2.1, “Page Scaling” [49].

Note

If CUPS does not provide the IPP `print-scaling` option for a printer, SavaPage falls back to the CUPS `fit-to-page`[3] boolean attribute to scale documents. Value `1` (true) scales the document up or down to fit the selected media. Value `0` (false) preserves the physical size of the printed document and crops any content outside the selected media.

K.1.1.8. printer-resolution

```
*VENResolution printer-resolution
*VENResolution *VEN150dpi 150x150 DPI      ❶
*VENResolution  VEN300dpi 300x300 DPI      ❷
*VENResolution  VENnnndpi nnnxnnn DPI|DPC  ❸
```

❶ `150x150` Dots Per Inch (DPI) as default.

❷ `300x300` Dots Per Inch (DPI).

❸ General format: any combination of Dots Per Inch (DPI) or Dots Per Centimeter (DPC).

K.1.1.9. sheet-collate

```
*VENCollate sheet-collate
*VENCollate *VENCollated  collated
*VENCollate VENUncollated uncollated
```

When `sheet-collate` is not mapped, SavaPage generates a single PDF and applies a one-copy print. The PDF is a concatenation of the requested number of copies with pages arranged in the right collate order.

K.1.1.10. sides

```
*VENSides sides
```

[3] https://www.cups.org/doc/options.html

```
*VENSides VENOneSided          one-sided
*VENSides *VENTwoSidedLongEdge two-sided-long-edge
*VENSides VENTwoSidedShortEdge two-sided-short-edge
```

K.1.2. Mapping PPD to IPP Extensions

PPD Options can be mapped to Internal IPP Extensions. For example:

```
*VENStapleOption org.savapage-finishings-staple  ❶
*VENStapleOption *VENNone       3  ❷
*VENStapleOption VENTopLeft    20  ❸
*VENStapleOption VENBottomLeft 21  ❹
```

❶ `VENStapleOption` is mapped to org.savapage-finishings-staple.
❷ `VENNone` value of `VENStapleOption` is mapped to IPP enum value 3 (none). The asterix `*` tells this options is the default.
❸ `VENTopLeft` value of `VENStapleOption` is mapped to IPP enum value 20 (staple-top-left).
❹ `VENBottomLeft` value of `VENStapleOption` is mapped to IPP enum value 21 (staple-bottom-left).

K.1.3. Restricting Standard Options

Standard CUPS/IPP options, like `number-up`, are independent of PPD, and available for all printers types. The values of these options can be restricted with PPDE syntax. Below are examples for options currently supported by SavaPage.

K.1.3.1. number-up

For example, option `number-up` can be restricted to values `1` (default), `2` and `4` like this:

```
*number-up number-up
*number-up *1 1
*number-up 2 2
*number-up 4 4
```

K.1.3.2. print-scaling

For example, option `print-scaling` can be restricted to value `none` like this:

```
*print-scaling print-scaling
*print-scaling  none none
```

In this case the Fit option is not shown in the Printer Settings dialog. See Section 3.5.2.1, “Page Scaling” [49].

Assuming `tray-1` contains `A4` and `tray-2` contains `A3` media, you can enforce a scaling policy with SPSubst rules.

```
# Expose attribute in full
*print-scaling print-scaling
*print-scaling  fit  fit
*print-scaling  none none
*print-scaling *auto auto

# Do NOT scale in tray-1 (A4)
*SPSubst/print-scaling/fit: A4-scale-none \
   media-source/tray-1 \
   *none

# Scale in tray-2 (A3)
*SPSubst/print-scaling/none: A3-scale-fit \
   media-source/tray-2 \
   *fit
```

K.2. PPD Rules

Sometimes, simple one-to-one PPD to IPP mappings do not suffice to get the right options to the PPD driver. In that case PPD rules can be of help. PPD rules take independent IPP (or PDF document) options as input, and return dependent PPD (or CUPS) options.

K.2.1. Generic PPD Rules

Generic PPD Rules add, assign or substitute PPD option values.

K.2.1.1. SPExtra

The `SPExtra` rule adds one or more PPD options to a print request, depending on one or more IPP options. The rule is formatted like this:

```
*SPExtra/<attrib>/<value>: <mnemonic>  \  ❶
   <attrib>/[!]<value> ...    \  ❷
   *<option>/<value> ...  \  ❸
```

❶ The rule prefix, an IPP attribute/value pair as main independent variable, and identifying `<mnemonic>`. Note: the `!` prefix before the IPP value to negates it, is **not** applicable in this case: you need to specify an exact IPP value to identify the rule.

❷ Optionally one or more extra IPP attribute/value pairs as independent variables. An optional `!` before a value negates it, and selects all other attribute values.

❸ One or more PPD option/value pairs as dependent variables. These option pairs are added to the print request, when all independent variables are present.

As an example, some org.savapage-finishings-punch rules for “Canon iR-ADV 8285/8295 UFR II” are shown below.

```
*SPExtra/org.savapage-finishings-punch/3: punch-none \
   *CNPunch/None

*SPExtra/org.savapage-finishings-punch/74: punch-dual-left \
   *CNPunch/Left

*SPExtra/org.savapage-finishings-punch/82: punch-quad-left \
   *CNPunch/Left
```

The rule can also be used to *replace* existing options. For example: org.savapage-finishings-booklet rules, for “Canon iR-ADV 8505 PS”. The snippet below *adds* extra options `BindMode` and `Booklet`, and *replaces* the `media`, `number-up`, `sides` and `fit-to-page` options.

```
*SPExtra/org.savapage-finishings-booklet/toleft-totop: booklet-toleft-totop-a3 \
   media/iso_a3_297x420mm \
   *media/iso_a4_210x297mm \
   *number-up/1 \
   *sides/one-sided \
   *BindMode/SaddleStitch \
   *Booklet/Left

*SPExtra/org.savapage-finishings-booklet/toleft-totop: booklet-toleft-totop-a4 \
   media/iso_a4_210x297mm \
   *media/iso_a5_148x210mm \
   *fit-to-page/1 \
   *number-up/1 \
   *sides/one-sided \
   *BindMode/SaddleStitch \
   *Booklet/Left
```

You can even replace a single IPP option value. For example, if you want to print “`NA Letter`” as `A4` (relying on default print-scaling) you can use this rule:

```
*SPExtra/media/na_letter_8.5x11in: letter-to-a4 \
   *media/iso_a4_210x297mm
```

K.2.1.2. SPSubst

This rule is meant to assign a native PPD value to an IPP attribute, and is formatted like this:

```
*SPSubst/<attrib>/<value>: <mnemonic>  \  ❶
   <attrib>/[!]<value> ...  \  ❷
   *<value>                    ❸
```

❶ The rule prefix, an IPP attribute/value pair as main independent variable, and identifying `<mnemonic>`.

❷ Optionally one or more extra IPP attribute/value pairs as independent variables. An optional `!` before a value negates it, and selects all other attribute values.

❸ The PPD option value as dependent variable. This value is assigned as PPD option to the main IPP attribute, when all independent variables are present.

As an example, some sheet-collate substitution rules for “Canon iR-ADV 8285/8295 UFR II” are shown below. Also see Section L.1.1.1, “org.savapage-finishings-staple” [397]

```
*SPSubst/sheet-collate/collated: collate \
   org.savapage-finishings-staple/3 \
   *True

*SPSubst/sheet-collate/uncollated: group \
   org.savapage-finishings-staple/3 \
   *Group

*SPSubst/sheet-collate/collated: staple-and-collate \
   org.savapage-finishings-staple/!3 \
   *StapleCollate

*SPSubst/sheet-collate/uncollated: staple-and-group \
   org.savapage-finishings-staple/!3 \
   *StapleGroup
```

K.2.2. Custom PPD Rules

Custom PPD Rules are restricted to specific situations.

K.2.2.1. LandscapeOrientation

Most PPD files contain an attribute called `LandscapeOrientation`, with value `Plus90` (default) or `Minus90`, that describes how landscape oriented pages are rotated to fit the portrait mode "Finished Page" [4].

In `Plus90` mode, landscape pages are -90 rotated to fit the "Finished Page", and the user must +90 rotate after printing, to get it into tangible landscape view again. `Minus90` is vice versa. The terms "Minus" and "Plus" are a bit confusing. They do not refer to the rotate direction before printing, but refer to the "manual" user rotation *after* printing, to get the "Finished Page" into landscape view again.

SavaPage creates IPP print jobs that are effectively printed according to Minus90. In this way, `staple-top-right` makes sense for landscape viewed pages.

[4] According to PWG5100.3: a Finished Page is "One side of a sheet in a Finished Document, i.e., one side of a sheet as perceived by a person after any cutting, folding, and/or booklet making".

As the CUPS `number-up` option still behaves according to the original PPD `LandscapeOrientation` attribute, SavaPage corrects with appropriate `number-up-layout` and `orientation-requested` options, to get the overall `Minus90` effect for `number-up` printed sheets that have landscape view.

Because SavaPage assumes that all target proxy printers behave according to the `Plus90` default, any exception must be entered in a PPDE like this:

```
*LandscapeOrientation: Minus90
```

When nevertheless `number-up` printed sheets do not turn out as expected, corrections can be made with `SPRule/number-up`.

> **Note**
>
> SavaPage does **not** correct with `number-up-layout` and `orientation-requested` options when a org.savapage-finishings-booklet finishing is chosen, since MFP booklet finishers are supposed to apply the correct orientation and layout.

K.2.2.1.1. SPRule/number-up

This is a fallback rule to make corrections, in the rare case that `LandscapeOrientation`, as discussed in the previous section, does not work out as expected. Each rule is prefixed with `*SPRule/number-up:` and formatted like this:

```
*SPRule/number-up: <mnemonic>     \  ❶
   pdf-orientation/<value>        \  ❷
   pdf-rotation/<value>           \  ❸
   pdf-content-rotation/<value>   \  ❹
   user-rotate/<value>            \  ❺
   number-up/<value>              \  ❻
   *orientation-requested/<value> \  ❼
   *number-up-layout/<value>      \  ❽
   *org.savapage-landscape           ❾
```

❶ The rule prefix and identifying `<mnemonic>`.
❷ The orientation `<value>` of the first PDF page to be printed: `portrait`, `landscape` .
❸ The rotation `<value>` of the first PDF page to be printed: `0`, `90`, `180` , `270` .
❹ The content rotation `<value>` of the first PDF page to be printed: `0`, `90`, `180` , `270` . When not specified, value `0` is assumed.
❺ The user rotation `<value>` on the PDF document to be printed: `0`, `90`. See Section 3.3.3.2, “Rotation” [36]. When not specified, value `0` is assumed.
❻ The selected `number-up` `<value>` : `1`, `2`, `4`, `6`, `9`.
❼ As the previous independent variables describe the situation “as is”, this is the first *dependent* variable telling CUPS the `orientation-requested` `<value>` : `4` (landscape), `5` (reverse landscape) , `6` (reverse portrait). Note: a value of `-` indicates that this CUPS attribute is *not* used.
❽ The second *dependent* variable telling CUPS the `number-up-layout` `<value>` : `btlr`, `btrl`, `lrbt`, `lrtb`, `rlbt`, `rltb`, `tblr`, `tbrl`. Note: a value of `-` indicates that this CUPS attribute is *not* used.
❾ The third optional *dependent* variable is used for the Document Log only, telling that the resulting n-up layout has logical landscape orientation. When not specified, portrait orientation is assumed.

As an example, some rules are shown below.

```
*SPRule/number-up: portrait-90-0-0-1 \
   pdf-orientation/portrait pdf-rotation/90 user-rotate/0 number-up/1  \
  *orientation-requested/- *number-up-layout/-
```

```
*SPRule/number-up: portrait-90-0-0-2-270 \
    pdf-orientation/portrait pdf-rotation/90 user-rotate/0 number-up/2  \
   *orientation-requested/5 *number-up-layout/tbrl

*SPRule/number-up: portrait-90-0-0-4 \
   pdf-orientation/portrait pdf-rotation/90 user-rotate/0 number-up/4  \
   *orientation-requested/- *number-up-layout/tbrl

*SPRule/number-up: portrait-90-0-0-6-270 \
   pdf-orientation/portrait pdf-rotation/90 user-rotate/0 number-up/6  \
   *orientation-requested/5 *number-up-layout/lrtb

*SPRule/number-up: portrait-90-0-0-9 \
   pdf-orientation/portrait pdf-rotation/90 user-rotate/0 number-up/9  \
   *orientation-requested/- *number-up-layout/tbrl

*SPRule/number-up: landscape-0-0-0-2-270 \
   pdf-orientation/landscape pdf-rotation/0 user-rotate/0 number-up/2  \
  *orientation-requested/5 *number-up-layout/tbrl

*SPRule/number-up: landscape-0-0-0-6-270 \
   pdf-orientation/landscape pdf-rotation/0 user-rotate/0 number-up/6  \
   *orientation-requested/5 *number-up-layout/lrtb

*SPRule/number-up: landscape-270-270-0-6-270 \
   pdf-orientation/landscape pdf-rotation/270 pdf-content-rotation/270 \
   user-rotate/0 number-up/6  \
  *orientation-requested/5 *number-up-layout/tbrl

*SPRule/number-up: landscape-270-0-90-2 \
   pdf-orientation/landscape pdf-rotation/270 pdf-content-rotation/0 \
   user-rotate/90 number-up/2  \
  *orientation-requested/- *number-up-layout/tbrl

*SPRule/number-up: landscape-270-0-90-4 \
   pdf-orientation/landscape pdf-rotation/270 pdf-content-rotation/0 \
   user-rotate/90 number-up/4  \
  *orientation-requested/- *number-up-layout/tbrl *org.savapage-landscape

*SPRule/number-up: landscape-270-0-90-6 \
   pdf-orientation/landscape pdf-rotation/270 pdf-content-rotation/0 \
   user-rotate/90 number-up/6  \
  *orientation-requested/- *number-up-layout/tbrl
```

K.2.2.1.2. Number-up semantics and limitations

- PDF input from Web Print can lead to unexpected number-up print results, when the semantic (perceived) orientation of a page, does not match the actual orientation/rotation of the PDF page. For instance, a mismatch occurs when a landscape oriented PDF page, has rotated portrait content. *There is no way SavaPage can identify this situation to make intelligent corrections.*
- An `SPRule/number-up` is applied when its independent variables (orientation, rotation) match the *first* page in the PDF document. When these variables differ for subsequent pages, the PPD has the final say on how page rotation and n-up layout turns out. *A PDF document with different page orientations may produce an unexpected number-up result.*

K.2.2.2. Booklet Imposition

Booklet imposition is assumed to be handled by the PPD. However, while some PPD's have an option for booklet folding, they do not handle booklet imposition. Therefore, a special PPDE option is available to prepare booklet page ordering, before sending PDF to CUPS. Add the following line to a .ppde file to enable pre-processing the PDF to booklet page ordering:

```
*SPLocalBooklet: True
```

> **Tip**
>
> You can use this option in combination with IPP option org.savapage-finishings-booklet to offer duplex printing in booklet page ordering, without the presence of a printer booklet finisher.

K.3. IPP Rules

IPP Rules specify relations between IPP attributes.

K.3.1. SPConstraint

An `SPConstraint` rule specifies values of two incompatible IPP attributes, for example two-sided printing on transparency media. The rule is formatted like this:

```
*SPConstraint: <mnemonic>  \  ❶
   <attrib>/[!]<value>     \  ❷
   <attrib>/[!]<value>        ❸
```

❶ The rule prefix and identifying `<mnemonic>`.
❷ The first IPP attribute/value pair. An optional `!` before a value negates it, and selects all other attribute values.
❸ The second IPP attribute/value pair, that is incompatible with the first.

Some examples:

```
*SPConstraint: transparency-two-sided \
    media-type/transparency \
    sides/!one-sided

*SPConstraint: booklet-punch \
    org.savapage-finishings-booklet/!none \
    org.savapage-finishings-punch/!3
```

> **Note**
>
> `SPConstraint` is the IPP counterpart of the `UIConstraints` directive found in PPD files.
>
> "CUPS does not enforce constraints when printing. Constraints must be managed and resolved by the user interface, because there is usually no way to specify preferences or intentions for automatic resolution of constraints by the driver or other filters." From "CUPS: Common Unix Printing System" by Michael R. Sweet, Sams Publishing, 2002 (page 337).
>
> `SPConstraint` rules are used by SavaPage to validate user input when specifying a print job.

K.3.1.1. Internal SPConstraint Rules

SavaPage has predefined rule sets. When activated, they are applied globally, independent of any PPDE file.

K.3.1.2. Booklet SPConstraint Set

This set is for org.savapage-finishings-booklet and must be activated by setting configuration property `ipp.ext.constraint.booklet.enable` to `Y`.

```
*SPConstraint: sp-booklet-number-up-1 \
```

```
    org.savapage-finishings-booklet/!none \
    number-up/1

*SPConstraint: sp-booklet-number-up-4 \
    org.savapage-finishings-booklet/!none \
    number-up/4

*SPConstraint: sp-booklet-number-up-6 \
    org.savapage-finishings-booklet/!none \
    number-up/6

*SPConstraint: sp-booklet-number-up-9 \
    org.savapage-finishings-booklet/!none \
    number-up/9

*SPConstraint: sp-booklet-one-sided \
    org.savapage-finishings-booklet/!none \
    sides/one-sided

*SPConstraint: sp-booklet-rotate-180-on \
    org.savapage-finishings-booklet/!none \
    org.savapage.int-page-rotate180/1
```

Important

When Media Cost is specified for Proxy Printers or Job Tickets supporting booklet finishing, activating this set is **crucial** for calculating the right cost for booklet print jobs. See Section 4.8.2.2, “Printer Costs” [134] and Section K.5.1, “Job Ticket Media Options” [392].

Caution

When this set is activated, user choices will be restricted to `number-up/2` and `side/!one-sided`. As this is correct for cost calculation, these options will probably **not** deliver the intended booklet. So, make sure to review the required IPP options for the booklet job, and use SPExtra rules to customize proxy printer instances where needed.

K.4. Job Extensions

In the PPD Extension file, special extensions are available to define Job IPP options. These options can be defined for scope *Copy* (sheet collection of a printed copy).

K.4.1. Job Copy Options

Job Copy options specify *finishing* actions, performed on a single printed copy. Each option is prefixed with `*SPJob/Copy:`

See Section L.1.2, “Internal IPP Job Extensions ” [400].

K.5. Job Ticket Extensions

Job Tickets are created with a dedicated Proxy Printer marked as Job Ticket Printer. As with any printer, a PPD extension file can be linked to this virtual printer.

In the PPD Extension file, special extensions are available to define Job Ticket IPP options, including *Cost Rules*. These IPP options are generic and abstracted from physical printers. When handling the ticket, the operator interprets the option values, selects a suitable printer, assures the right media are present in the target tray, and redirects the job to it.

Job Ticket IPP options can be defined for scope *Media* (single side of a page sheet), *Sheet* (single page sheet), *Copy* (sheet collection of a printed copy) and *Set* (the complete set of copies). The syntax is as follows:

```
*SPJobTicket/<scope>: <attr> \ ❶
  [*|+]<value> ...   ❷
```

❶ The `*SPJobTicket` option with scope `Media` | `Sheet` | `Copy` | `Set`, and attribute name.
❷ One or more option values. An `*` before a value marks it as default. A `+` prefix signifies an *extended* value, that is available for Job Ticket Operators only, so they can ad-hoc assign this value when editing a job ticket.

> **Note**
>
> The `+` prefix can also be assigned to regular option values. For example, if a Job Ticket Creator is not allowed to select color mode, the following code snippet can be added to the PPD extension file:
>
> ```
> # Virtual print color mode, with dummy PPD option and values.
> *VPrintColorMode print-color-mode
> *VPrintColorMode dummy monochrome
> *VPrintColorMode dummy +color
> ```

Cost Rules are introduced for each option scope in the sections below.

> **Important**
>
> IPP Options defined in the Job Ticket context are *not* mapped to their PPD counterparts. Therefore, their chosen values will *not* be send with the print job.

K.5.1. Job Ticket Media Options

IPP attributes describing media characteristics, like `media-color` and `media-type` are supported. Each option is prefixed with `*SPJobTicket/Media:`

Some examples:

```
*SPJobTicket/Media: media-color *white int.colored  ❶
*SPJobTicket/Media: media-type *paper transparency labels ext.letterhead-1   ❷
```

❶ Option to select white and colored. `white` is an IANA media-color and is the default. `int.colored` is an internal IPP value extension, denoting a non-white color. Other values could be `blue`, `red`, `green`, `orange`, etc.
❷ Option to select media types: `ext.letterhead-1` is an external IPP value extension.

Media Cost is charged per media *side*, and specified for a combination of IPP values for media attributes. Each cost rule is prefixed with `*SPJobTicket/Media/Cost:` and formatted like this:

```
*SPJobTicket/Media/Cost: <cost> <mnemonic> \  ❶
   <media*>/[!]<value> ... \  ❷
   <attrib>/[!]<value> ... \  ❸
```

❶ Decimal point `<cost>` and identifying `<mnemonic>`.
❷ One or more IPP `<media*>/<value>` pairs. An optional `!` before a value negates it, and selects all other attribute values.
❸ Optionally one or more IPP non-media `<attrib>/<value>` pairs.

> **Note**
>
> The IPP `<media*>/<value>` pairs referred to in `*SPJobTicket/Media/Cost` must either be automatically picked up from the PPD, or be defined as PPD to IPP Mapping or `*SPJobTicket/Media`.

Some examples:

```
*SPJobTicket/Media/Cost: 0.0430 white-A4-080-S \  ❶
   media-type/ext.paper-80 \
   media-color/white \
   media/iso_a4_210x297mm \
   sides/one-sided

*SPJobTicket/Media/Cost: 0.0610 color-A4-120-S \  ❷
   media-type/ext.paper-120 \
   media-color/!white \
   media/iso_a4_210x297mm \
   sides/one-sided

*SPJobTicket/Media/Cost: 0.0390 white-A4-080-D \  ❸
   media-type/ext.paper-80 \
   media-color/white \
   media/iso_a4_210x297mm \
   sides/!one-sided

*SPJobTicket/Media/Cost: 0.0790 letterhead-A4-S \ ❹
   media-type/ext.letterhead-1 \
   media/iso_a4_210x297mm \
   sides/one-sided
```

❶ Single-sided A4 print on 80 grams white paper: 0.0430 per side.
❷ Single-sided A4 print on 120 grams colored paper: 0.0610 per side.
❸ Double-sided A4 print on 80 grams white paper: 0.0390 per side.
❹ Single-sided A4 print on letterhead: 0.0790 per side.

> **Important**
>
> When *SPJobTicket/Media/Cost items are present, they acts as constraint. When a user sets print job properties, a cost rule must be present that matches chosen media options. When no rule is found, a warning message is displayed to the user.
>
> The calculated cost of the *first* cost rule, that applies to the Job Ticket option values, is used as media cost.

K.5.2. Job Ticket Sheet Options

Job Ticket Sheet options specify *finishing* actions, performed on a single printed sheet. Each option is prefixed with `*SPJobTicket/Sheet:`

For example:

```
*SPJobTicket/Sheet: org.savapage-finishings-ext *none laminate  ❶
```

❶ org.savapage-finishings-ext option to select an extra finishing to be performed manually by Job Ticket operator.

Sheet Cost is charged per media *sheet*, and specified for a combination of `SPJobTicket/Sheet` and other (`media*`) attribute values. Each cost rule is prefixed with `*SPJobTicket/Sheet/Cost:` and formatted like this:

```
*SPJobTicket/Sheet/Cost: <cost> <mnemonic> \  ❶
```

```
   <sheet>/[!]<value> ... \  ❷
   <attr>/[!]<value> ... \  ❸
```

❶ Decimal point `<cost>` and identifying `<mnemonic>`.
❷ One or more IPP `<sheet>/<value>` pairs of type `SPJobTicket/Sheet`. An optional `!` before a value negates it, and selects all other attribute values.
❸ Optionally one or more other IPP `<attr>/<value>` pairs.

For example:

```
*SPJobTicket/Sheet/Cost: 0.5000 laminate-A4 org.savapage-finishings-ext/laminate \
    media/iso_a4_210x297mm
*SPJobTicket/Sheet/Cost: 0.7500 laminate-A3 org.savapage-finishings-ext/laminate \
    media/iso_a3_297x420mm
```

Important

The calculated cost of *all* sheet cost rules, that apply to the Job Ticket option values, are *accumulated* as cost per sheet.

K.5.3. Job Ticket Copy Options

Job Ticket Copy options specify *finishing* actions, performed on a single printed copy. Each option is prefixed with `*SPJobTicket/Copy:`

Some examples:

```
*SPJobTicket/Copy: org.savapage-cover-type *no-cover ext.printfront-1  ❶
*SPJobTicket/Copy: org.savapage-finishings-ext *none laminate bind adhesive ext.binder  ❷
```

❶ org.savapage-cover-type option to select a cover type. `ext.printfront-1` is an External IPP extension.
❷ org.savapage-finishings-ext option to select an extra finishing to be performed manually by Job Ticket operator. `ext.binder` is an External IPP extension.

Copy Cost is charged per job *copy*, and specified for a combination of `SPJobTicket/Copy` and other (`media*`) attribute values. Each cost rule is prefixed with `*SPJobTicket/Copy/Cost:` and formatted like this:

```
*SPJobTicket/Copy/Cost: <cost> <mnemonic> \  ❶
   <copy>/[!]<value> ... \  ❷
   <attr>/[!]<value> ... \  ❸
```

❶ Decimal point `<cost>` and identifying `<mnemonic>`.
❷ One or more IPP `<copy>/<value>` pairs of type `SPJobTicket/Copy`. An optional `!` before a value negates it, and selects all other attribute values.
❸ Optionally one or more other IPP `<attr>/<value>` pairs.

For example:

```
*SPJobTicket/Copy/Cost: 0.5000 folder-A4 org.savapage-finishings-ext/ext.folder \
    media/iso_a4_210x297mm
```

Important

The calculated cost of *all* copy cost rules, that apply to the Job Ticket option values, are *accumulated* as cost per copy.

K.5.3.1. Using External IPP Extensions

Job Ticket Copy Options are not confined to regular IPP attributes and Internal IPP Extensions. By utilizing External IPP Extensions, you can fully customize your Job Tickets Copy options and cost rules. For example:

```
*SPJobTicket/Copy: org.savapage.ext-myoption \  ❶
   *none ext.choice-1 ext.choice-2

*SPJobTicket/Copy/Cost: 0.1000 myrule-1 \       ❷
   org.savapage.ext-myoption/ext.choice-1 media/iso_a4_210x297mm

*SPJobTicket/Copy/Cost: 0.1500 myrule-2 \       ❸
   org.savapage.ext-myoption/ext.choice-2 media/iso_a3_297x420mm
```

❶ Definition of External IPP extension `myoption` with two custom choices. The default choice `none` is a reserved internal value, indicating that the option is not selected.
❷ Cost rule for `ext.choice-1` and A4 media.
❸ Cost rule for `ext.choice-2` and A3 media.

K.5.4. Job Ticket Set Options

Job Ticket Set options specify actions performed on the complete set of copies[5]. Each option is prefixed with `*SPJobTicket/Set:`

For example:

```
*SPJobTicket/Set: org.savapage-job-sheets       \ ❶
    +none *job-start-sheet
*SPJobTicket/Set: org.savapage-job-sheets-media \ ❷
    *iso_a4_210x297mm +iso_a3_297x420mm
```

❶ org.savapage-job-sheets option to select a job-sheet.
❷ org.savapage-job-sheets-media option to select the job-sheet media.

Set Cost is charged per *job*, and specified for a combination of `SPJobTicket/Set` and other (`org.savapage-job-sheets-*`) attribute values. Each cost rule is prefixed with `*SPJobTicket/Set/Cost:` and formatted like this:

```
*SPJobTicket/Set/Cost: <cost> <mnemonic> \  ❶
   <set>/[!]<value> ... \   ❷
   <attr>/[!]<value> ... \  ❸
```

❸ Decimal point `<cost>` and identifying `<mnemonic>`.
❷ One or more IPP `<set>/<value>` pairs of type `SPJobTicket/Set`. An optional `!` before a value negates it, and selects all other attribute values.
❸ Optionally one or more other IPP `<org.savapage-job-sheets-*>/<value>` pairs.

For example:

```
*SPJobTicket/Set/Cost: 0.05 banner-A4 org.savapage-job-sheets/!none \
    org.savapage-job-sheets-media/iso_a4_210x297mm
```

[5] The Job Ticket Set includes all copies, and therefore differs from the *Set* as defined in RFC8011, where it is "a logical boundary between the delivered Media Sheets of a printed Job. For example, in the case of a ten-page single Document with collated pages and a request for 50 copies, each of the 50 printed copies of the Document constitutes a Set. If the pages were uncollated, then 50 copies of each of the individual pages within the Document would represent each Set."

Important

The calculated cost of *all* cost rules, that apply to the Job Ticket option values, are *accumulated* as cost per set.

K.6. Tips and Tricks

K.6.1. Fast Print A4 and Letter to Single Tray

For Fast Mode printers with a single tray holding A4 media, PPDE can be used to configure an extra (virtual) IPP media-source entry for Letter, that maps to the same A4 tray. The two media sources will be visible in the Proxy Printer Edit dialog, so A4 and Letter media size can be assigned to them, and both A4 and Letter jobs will be Fast Print candidates. With an SPExtra rule the Letter media is substituted with A4. And, since `fit-to-page` print-scaling is applied for Fast Mode printing by default, the Letter job will correctly scale to A4.

The .ppde of the Fast Mode printer looks like this:

```
#-----------------------------------------------------------------------
# Two trays mapped to a single media-source holding A4 media.
# In SavaPage Admin Web App: assign A4 to tray-1 and US Letter to tray-2.
# -----------------------------------------------------------------------
*VENInputSlot media-source
*VENInputSlot *Auto tray-1
*VENInputSlot  Auto tray-2

# -----------------------------------------------------------------------
# When job is US Letter, send as A4. The fit-to-page strategy
# for Fast Print will make Letter scale to A4.
# -----------------------------------------------------------------------
*SPExtra/media/na_letter_8.5x11in: letter-to-a4 \
   *media/iso_a4_210x297mm
```

K.7. proxy-print.log

The IPP request and response details of every Proxy Print job send to CUPS is logged in the rotating log file:

`/opt/savapage/server/logs/proxy-print.log`

This file is an indispensable resource for debugging PPDE related issues. See `log4j.properties` for more information.

Appendix L. IPP Extensions

Internet Printing Protocol (IPP) attributes and values are registered by IANA. See the IANA site[1] or the PWG site[2] for a full list.

SavaPage uses two types of extensions:

- *Internal* extensions, which are intrinsic to SavaPage.
- *External* extensions, as defined in implementation specific configuration files.

L.1. Internal IPP Extensions

Internal IPP Extension attributes are *intrinsic* to SavaPage. To distinguish them from IANA registrations, their names have a `org.savapage-` prefix. Attribute value extensions with "type2 keyword" syntax are `int.` prefixed.

Attributes and values are summarized in the sections below. Attribute values are IANA registered, and the semantics can be found in the Reference documents. Attribute value `int.` extensions are described separately.

L.1.1. Internal IPP - PPD Mapping Extensions

These Internal IPP extensions are used to map vendor specific PPD options to an independent common denominator. They are *never* send to CUPS as print job descriptor as such.

L.1.1.1. org.savapage-finishings-staple

Staple positions are specified with respect to portrait media orientation. See RFC3806[3].

Attribute	Value	Name	Syntax	Reference
org.savapage-finishings-staple	3	none	1setOf type2 enum	RFC8011
org.savapage-finishings-staple	20	staple-top-left	1setOf type2 enum	RFC8011
org.savapage-finishings-staple	21	staple-bottom-left	1setOf type2 enum	RFC8011
org.savapage-finishings-staple	22	staple-top-right	1setOf type2 enum	RFC8011
org.savapage-finishings-staple	23	staple-bottom-right	1setOf type2 enum	RFC8011
org.savapage-finishings-staple	24	edge-stitch-left	1setOf type2 enum	RFC8011
org.savapage-finishings-staple	25	edge-stitch-top	1setOf type2 enum	RFC8011
org.savapage-finishings-staple	26	edge-stitch-right	1setOf type2 enum	RFC8011
org.savapage-finishings-staple	27	edge-stitch-left-bottom	1setOf type2 enum	RFC8011
org.savapage-finishings-staple	28	staple-dual-left	1setOf type2 enum	RFC8011
org.savapage-finishings-staple	29	staple-dual-top	1setOf type2 enum	RFC8011
org.savapage-finishings-staple	30	staple-dual-right	1setOf type2 enum	RFC8011
org.savapage-finishings-staple	31	staple-dual-bottom	1setOf type2 enum	RFC8011
org.savapage-finishings-staple	32	staple-triple-left	1setOf type2 enum	PWG5100.1

[1] https://www.iana.org/assignments/ipp-registrations/ipp-registrations.xhtml
[2] https://www.pwg.org/ipp/ipp-registrations.xml
[3] https://tools.ietf.org/html/rfc3806

Attribute	Value	Name	Syntax	Reference
org.savapage-finishings-staple	33	staple-triple-top	1setOf type2 enum	PWG5100.1
org.savapage-finishings-staple	34	staple-triple-right	1setOf type2 enum	PWG5100.1
org.savapage-finishings-staple	35	staple-triple-bottom	1setOf type2 enum	PWG5100.1

Table L.1. Internal IPP Attribute: org.savapage-finishings-staple

L.1.1.2. org.savapage-finishings-punch

Punch positions are specified with respect to portrait media orientation. See RFC3806[4].

Attribute	Value	Name	Syntax	Reference
org.savapage-finishings-punch	3	none	1setOf type2 enum	RFC8011
org.savapage-finishings-punch	70	punch-top-left	1setOf type2 enum	PWG5100.1
org.savapage-finishings-punch	71	punch-bottom-left	1setOf type2 enum	PWG5100.1
org.savapage-finishings-punch	72	punch-top-right	1setOf type2 enum	PWG5100.1
org.savapage-finishings-punch	73	punch-bottom-right	1setOf type2 enum	PWG5100.1
org.savapage-finishings-punch	74	punch-dual-left	1setOf type2 enum	PWG5100.1
org.savapage-finishings-punch	75	punch-dual-top	1setOf type2 enum	PWG5100.1
org.savapage-finishings-punch	76	punch-dual-right	1setOf type2 enum	PWG5100.1
org.savapage-finishings-punch	77	punch-dual-bottom	1setOf type2 enum	PWG5100.1
org.savapage-finishings-punch	78	punch-triple-left	1setOf type2 enum	PWG5100.1
org.savapage-finishings-punch	79	punch-triple-top	1setOf type2 enum	PWG5100.1
org.savapage-finishings-punch	80	punch-triple-right	1setOf type2 enum	PWG5100.1
org.savapage-finishings-punch	81	punch-triple-bottom	1setOf type2 enum	PWG5100.1
org.savapage-finishings-punch	82	punch-quad-left	1setOf type2 enum	PWG5100.1
org.savapage-finishings-punch	83	punch-quad-top	1setOf type2 enum	PWG5100.1
org.savapage-finishings-punch	84	punch-quad-right	1setOf type2 enum	PWG5100.1
org.savapage-finishings-punch	85	punch-quad-bottom	1setOf type2 enum	PWG5100.1

Table L.2. Internal IPP Attribute: org.savapage-finishings-punch

L.1.1.3. org.savapage-finishings-fold

Attribute	Value	Name	Syntax	Reference
org.savapage-finishings-fold	3	none	1setOf type2 enum	RFC8011
org.savapage-finishings-fold	90	fold-accordion	1setOf type2 enum	PWG5100.1
org.savapage-finishings-fold	91	fold-double-gate	1setOf type2 enum	PWG5100.1
org.savapage-finishings-fold	92	fold-gate	1setOf type2 enum	PWG5100.1

[4] https://tools.ietf.org/html/rfc3806

Attribute	Value	Name	Syntax	Reference
org.savapage-finishings-fold	93	fold-half	1setOf type2 enum	PWG5100.1
org.savapage-finishings-fold	94	fold-half-z	1setOf type2 enum	PWG5100.1
org.savapage-finishings-fold	95	fold-left-gate	1setOf type2 enum	PWG5100.1
org.savapage-finishings-fold	96	fold-letter	1setOf type2 enum	PWG5100.1
org.savapage-finishings-fold	97	fold-parallel	1setOf type2 enum	PWG5100.1
org.savapage-finishings-fold	98	fold-poster	1setOf type2 enum	PWG5100.1
org.savapage-finishings-fold	99	fold-right-gate	1setOf type2 enum	PWG5100.1
org.savapage-finishings-fold	100	fold-z	1setOf type2 enum	PWG5100.1

Table L.3. Internal IPP Attribute: org.savapage-finishings-fold

L.1.1.4. org.savapage-finishings-booklet

Attribute	Keyword Value	Syntax	Reference
org.savapage-finishings-booklet	none	type2 keyword	RFC8011
org.savapage-finishings-booklet	toleft-totop	type2 keyword	PWG5100.3
org.savapage-finishings-booklet	toright-tobottom	type2 keyword	PWG5100.3

Table L.4. Internal IPP Attribute: org.savapage-finishings-booklet

Also see Section K.2.2.2, “Booklet Imposition” [389]

Important

For the right booklet cost calculation, make sure to activate the Booklet SPConstraint Set.

L.1.1.5. org.savapage-finishings-jog-offset

A finishing that shifts Each Set in the output-bin from the previous one by a small amount which is device dependent (PWG5100.1).

Attribute	Value	Name	Syntax	Reference
org.savapage-finishings-jog-offset	3	none	1setOf type2 enum	RFC8011
org.savapage-finishings-jog-offset	14	EachSet	1setOf type2 enum	PWG5100.1

Table L.5. Internal IPP Attribute: org.savapage-finishings-jog-offset

L.1.1.6. org.savapage-requesting-user-name

The internal IPP extension `org.savapage-requesting-user-name` is the equivalent of IPP PRINT_JOB operation attribute `requesting-user-name`. In the example below it is mapped to PPD Option `VENRequestingUserName`.

```
*VENRequestingUserName org.savapage-requesting-user-name
```

Tip

`org.savapage-requesting-user-name` can be used to pass the value of `requesting-user-name` as option in a `*JCLOpenUI` section of a PPD file.

L.1.2. Internal IPP Job Extensions

These options specify actions performed on a single printed copy. Each option is prefixed with `*SPJob/Copy:`

See Section K.4.1, “Job Copy Options” [391].

L.1.2.1. org.savapage-cover-front-type

This attribute indicates whether front covers are wanted for a printed copy and which sides of the cover must contain print-stream pages.

Attribute	Keyword Value	Syntax	Reference
org.savapage-cover-front-type	no-cover	type2 keyword	PWG5100.3
org.savapage-cover-front-type	print-none	type2 keyword	PWG5100.3
org.savapage-cover-front-type	print-front	type2 keyword	PWG5100.3
org.savapage-cover-front-type	print-back	type2 keyword	PWG5100.3
org.savapage-cover-front-type	print-both	type2 keyword	PWG5100.3

Table L.6. Internal IPP Attribute: org.savapage-cover-front-type

For example:

```
*SPJob/Copy: org.savapage-cover-front-type \
  *no-cover print-none print-front print-back print-both  ❶
```

❶ Option to select `no-cover` as default and four other keyword values.

L.1.2.2. org.savapage-cover-front-media-source

The IPP media-source the cover is printed from.

For example:

```
*SPJob/Copy: org.savapage-cover-front-media-source \
  *tray-3 tray-4  ❶
```

❶ Option to select `tray-3` as default media source for the cover.

L.1.2.3. org.savapage-back-front-type

This attribute indicates whether back covers are wanted for a printed copy and which sides of the cover must contain print-stream pages.

Attribute	Keyword Value	Syntax	Reference
org.savapage-cover-back-type	no-cover	type2 keyword	PWG5100.3

Attribute	Keyword Value	Syntax	Reference
org.savapage-cover-back-type	print-none	type2 keyword	PWG5100.3
org.savapage-cover-back-type	print-front	type2 keyword	PWG5100.3
org.savapage-cover-back-type	print-back	type2 keyword	PWG5100.3
org.savapage-cover-back-type	print-both	type2 keyword	PWG5100.3

Table L.7. Internal IPP Attribute: org.savapage-cover-back-type

Also see Section L.1.2.1, “org.savapage-cover-front-type” [400].

L.1.2.4. org.savapage-cover-back-media-source

The IPP media-source the cover is printed from. Also see Section L.1.2.2, “org.savapage-cover-front-media-source” [400].

L.1.2.5. Mapping Internal IPP Job Options to PPD

Internal IPP Job Copy options can be mapped to PPD with SPExtra rules as illustrated in the example below.

```
*SPExtra/org.savapage-cover-front-media-source/tray-3: cover-front-tray-3 \
    org.savapage-cover-front-type/!no-cover \
    *FrontCover/Yes \
    *FrontCoverPrintTray/TRAY3

*SPExtra/org.savapage-cover-front-media-source/tray-4: cover-front-tray-4 \
    org.savapage-cover-front-type/!no-cover \
    *FrontCover/Yes \
    *FrontCoverPrintTray/TRAY4
```

L.1.2.6. IPP Job Copy Costs

Copy Cost is charged per job *copy*, and specified for a combination of `SPJob/Copy` and other (`media*`) attribute values. Each cost rule is prefixed with `*SPJob/Copy/Cost:` and formatted like this:

```
*SPJob/Copy/Cost: <cost> <mnemonic> \  ❶
   <copy>/[!]<value> ... \  ❷
   <attr>/[!]<value> ... \  ❸
```

❶ Decimal point `<cost>` and identifying `<mnemonic>`.
❷ One or more IPP `<copy>/<value>` pairs of type `SPJob/Copy`. An optional `!` before a value negates it, and selects all other attribute values.
❸ Optionally one or more other IPP `<attr>/<value>` pairs.

For example:

```
*SPJob/Copy/Cost: 0.0200 cost-cover-front-A4 \
    org.savapage-cover-front-type/!no-cover \
    media/iso_a4_210x297mm

*SPJob/Copy/Cost: 0.0400 cost-cover-front-A3 \
    org.savapage-cover-front-type/!no-cover \
    media/iso_a3_297x420mm
```

L.1.3. Internal IPP Job Ticket Extensions

These Internal IPP extensions are used in Job Ticketing Cost Rules. See Section K.5, “Job Ticket Extensions” [391].

L.1.3.1. org.savapage-finishings-ext

This attribute represents a finishing action, on a set of pages (a copy), executed externally by an operator, on the printed output.

Attribute	Keyword Value	Syntax	Reference
org.savapage-finishings-ext	none	type2 keyword	RFC8011
org.savapage-finishings-ext	laminate	type2 keyword	PWG5100.1
org.savapage-finishings-ext	bind	type2 keyword	PWG5100.1
org.savapage-finishings-ext	adhesive	type2 keyword	Candidate Standard PWG 5100.1-2014: “sheets are bound using glue or adhesive.”

Table L.8. Internal IPP Attribute: org.savapage-finishings-ext

L.1.3.2. org.savapage-cover-type

Caution

This attribute is intended for externally printed Job Tickets only.

The same PDF document must be printed twice with different page ranges:

1. The first page(s) are printed on a single cover page. The cost for this action is calculated according to the Job Ticket Copy Cost Rules.
2. The rest of the pages are printed on selected media. The cost for this action is calculated according to the Job Ticket Media Cost Rules.

Attribute	Keyword Value	Syntax	Reference
org.savapage-cover-type	no-cover	type2 keyword	PWG5100.3
org.savapage-cover-type	int.printfront	type2 keyword	SavaPage Extension: a single-sided printed cover page.
org.savapage-cover-type	int.printboth	type2 keyword	SavaPage Extension: a double-sided printed cover page.

Table L.9. Internal IPP Attribute: org.savapage-cover-type

For example:

```
*SPJobTicket/Copy: org.savapage-cover-type \
  *no-cover int.printfront int.printboth  ❶
```

❶ Option to select `no-cover` as default and two internal values.

The color of the cover can be specified with org.savapage-cover-type-color

L.1.3.3. org.savapage-cover-type-color

Thias attribute specifies the color of the org.savapage-cover-type. For example:

```
*SPJobTicket/Copy: org.savapage-cover-type-color \
  *white yellow orange red pink blue green  ❶
```

❶ Option to select `white` as default and several other IANA defined `media-color` values.

L.1.3.4. org.savapage-job-sheets

This attribute determines which Job start/end sheet(s), if any, must be printed with a Job. Contrary to the IPP `job-sheets` definition, where sheets are part of the job, `org.savapage-job-sheets` are printed as a *separate* job. In this way sheets can be printed from a different `media-source` with distinctive org.savapage-job-sheets-media and `media-color`.

A job sheet is a single `monochrome` page with job detail information.

Attribute	Keyword Value	Syntax	Reference
org.savapage-job-sheets	none	type2 keyword	RFC8011
org.savapage-job-sheets	job-start-sheet	type2 keyword	PWG5100.3: “A job sheet MUST be printed to indicate the start of the job”.
org.savapage-job-sheets	job-end-sheet	type2 keyword	PWG5100.3: “A job sheet MUST be printed to indicate the end of the job”.

Table L.10. Internal IPP Attribute: org.savapage-job-sheets

L.1.3.5. org.savapage-job-sheets-media

The IPP `media` for org.savapage-job-sheets. Multiple keyword values are allowed, like `iso_a4_210x297mm`, `iso_a3_297x420mm`, etc.

L.1.4. Internal IPP Marker

These Internal IPP extensions are used to add a special marker to a print job. They are *never* send to CUPS as such, but translated by SavaPage to "real" IPP or CUPS options. IPP marker names have a `org.savapage.int-` prefix.

L.1.4.1. org.savapage.int-page-rotate180

SavaPage creates IPP print jobs that are printed according to LandscapeOrientation `Minus90`. This is convenient most of the time. For example, when a landscape oriented PDF is 4-up printed, and `staple-top-right` and `punch-dual-right` are available as finishing. But, what if `staple-top-left` and `punch-dual-left` are the only options available? In that case a 180 degrees rotation of the "Finished-Page" is needed. This is where the `org.savapage.int-page-rotate180` boolean option comes in. This marker is added to each proxy printer as option to make the printed result "Rotate by 180 degrees", and translated by SavaPage to the proper CUPS `orientation-requested` and `number-up-layout` values.

Attribute	Keyword Value	Syntax	Description
org.savapage.int-page-rotate180	0	boolean	Do not rotate.
org.savapage.int-page-rotate180	1	boolean	Rotate by 180 degrees.

Table L.11. Internal IPP Marker Attribute: org.savapage.int-page-rotate180

Note

`org.savapage.int-page-rotate180` can be used in SPConstraint rules.

L.2. External IPP Extensions

External IPP Extension attributes as defined in implementation specific configuration files. To distinguish them from IANA registrations, their names have a `org.savapage.ext-` prefix.

Important

External IPP attributes can be used as IPP Job Ticketing Extension only.

The syntax is as follows:

- External *value* extensions with "type2 keyword" syntax must be `ext.` prefixed.
- Value `none` is a reserved internal value, indicating that the attribute option is not selected.
- External attribute values can be added to an Internal IPP Job Ticket Extension.
  - For example: to charge cost for a finishing to wrap a printed copy into a folder, a `ext.folder` value could be added to org.savapage-finishings-ext.

External IPP Extensions open the way to full customization of Job Ticketing options. Any option with any number of choices can be configured in a PPDE file as `*SPJobTicket/Copy` finishing, and thereupon be used in any cost rule. See Section K.5.3.1, "Using External IPP Extensions" [395] and Section L.3, "IPP Localization" [404].

L.3. IPP Localization

User interface localization of regular IPP attributes and Internal IPP Extensions are part of SavaPage.

Localization of External IPP Extensions text, and optionally 16x16 pixel icons, must be supplied by XML files in the `/opt/savapage/server/custom/cups/i18n` directory. The default `ipp-i18n.xml` file is reserved for the English locale. Variants are created by appending the language locale to the base file name. For example: `ipp-i18n_de.xml` is the German version.

An annotated `ipp-i18n.xml.template` file is available in the target directory.

See Section 3.5.2.3, "Custom Text and Icons" [51] for an example screenshot.

Tip

`ipp-i18n*.xml` files can also be used to override IPP localization that is part of SavaPage.

Warning

After creating or updating any of the `ipp-i18n*.xml` files you might need to restart the server to see the effect.

Appendix M. SavaPage Plug-ins

A *plug-in* (aka “extension”) is a software component that adds a specific feature to SavaPage. Plug-ins have a well-defined interface so partner developers can easily create isolated components that extend the application with new features. Extension interfaces are defined in the savapage-ext[1] project.

A plug-in is installed by copying its property file in `/opt/savapage/server/ext` and its jar files in the `/opt/savapage/server/ext/lib` directory. For example, the Mollie Payment Plug-in is installed with these two files:

```
/opt/savapage/server/ext/savapage-ext-mollie.properties
/opt/savapage/server/ext/lib/savapage-ext-mollie-<version>.jar
```

Important

Since property files contain sensitive data make sure they are protected by executing commands like:

```
sudo chown savapage:savapage savapage-ext-mollie.properties
sudo chmod 600 savapage-ext-mollie.properties
```

M.1. Web API Callback Plug-in

The Web API Callback method is used by many third-party providers who offer their services via HTTP. SavaPage supports this method with the `/callback` URL *path*. However, URL *protocol* and *authority* for the callback needs to be configured. Configuration is done by using the Config Editor of the Admin Web App. The following configuration property is available:

Configuration property	Description
ext.webapi.callback.url-base	The publicly accessible base URL, i.e. `protocol://authority` *without* the *path*, of the Web API callback interface (no trailing slash). When SavaPage is implemented as intranet application to be accessed with a local URL, take care to configure proper port forwarding of the public base URL to the local SavaPage server host name or IP address in your router or firewall. ***Important**: use `https` protocol, and make sure the SSL Certificate for public access is valid, i.e. not self-signed. When the certificate is invalid, a third-party service provider might not be able to deliver call-back messages.*

Table M.1. Web API Callback Configuration Property

M.1.1. Payment Gateway Plug-in

The Payment Gateway Plug-in is based on the Web API Callback Plug-in and uses the `/callback/payment` URL *path*. The following configuration property can be set by using the Config Editor of the Admin Web App.

Configuration property	Description
ext.webapi.redirect.url-webapp-user	The User Web App URL used by the Web API to redirect to after a *remote* Web App dialog is finished.

[1] https://gitlab.com/savapage/savapage-ext

Configuration property	Description
	Configuration is optional. SavaPage uses the local URL from which the remote excursion started as default.

Table M.2. Payment Gateway Configuration Property

Payment Gateway events are persisted in the rotating log file:

```
/opt/savapage/server/logs/paymentgateway.log
```

This file has a tab separated value (TSV) format for easy import and manipulation into spreadsheet programs.

M.1.1.1. Generic Payment Plug-in

A Generic Payment Plug-in implements several payment methods behind a single Web API. Only one generic plug-in can be active. See Section 3.10.5, “Financial” [83], Section 3.10.8, “Transfer Money” [85] and Chapter 8, *Payment Web App* [232].

M.1.1.1.1. Mollie Payment Plug-in

Mollie[2] is a Dutch payment provider that offers a single Web API for various payment methods, including:

- Creditcard
- PayPal
- paysafecard
- SOFORT Banking (Europe)
- SEPA bank transfers (Europe)
- Bancontact/Mister Cash (Belgium)
- Belfius Direct Net (Belgium)
- iDEAL#|#Wero (Netherlands).

Mollie supports the European Economic Area and Switzerland and accepts EUR payments only.

See the `README.md` of the savapage-ext-mollie[3] project for more information.

Note

The Mollie Payment Plug-in is shipped with the SavaPage install binary.

M.1.1.1.2. Generic Payment Pitfalls

Callback messages for generic payments return the identity of the user that started the payment in the Transfer Money dialog. So, when a payment is acknowledged, we can easily add the amount paid to the user's balance. In some very unlikely cases a user might not be found. For example, when a database export is restored or a user was deleted, all in the short lifecycle of a payment transaction.

In the rare case a user is not found, a warning message containing the user and transaction identification are written to the Application Log. With this information the user balance can be updated manually, after the user has been added again, either in the Admin WebApp or with a Server Command.

M.2. OAuth Client Plug-in

With OAuth[4] Client Plug-ins Login Alternatives can be activated. Currently the following providers are supported:

[2] https://www.mollie.com/en/
[3] https://gitlab.com/savapage-ext/savapage-ext-mollie
[4] https://en.wikipedia.org/wiki/OAuth

- Keycloak[5]
- Google[6]
- Microsoft Azure[7]
- Smartschool[8]

See the `README.md` of the savapage-ext-oauth[9] project for more information.

> **Note**
>
> A special URL path is available to use OAuth as SSO for User Web App. This URL can be presented on the site of an OAuth provider, to seamlessly link to SavaPage. See Appendix E, *URL Cheat Sheet* [362].

M.3. Mail Print IMAP OAuth Plug-in

This plug-in supports IMAP OAuth[10] authentication for Mail Print. Currently the following providers are supported:

- outlook.office365.com[11]

  An annotated `savapage-ext-office365-oauth-imap.properties.template` is present in the `/opt/savapage/ext` directory. Copy this template to a file with a `.properties` extension in the same directory and restrict permissions, since the file contains confidential information. An example:

  ```
  $ sudo su - savapage
  $ cd /opt/savapage/server/ext
  $ cp savapage-ext-office365-oauth-imap.properties.template office365-oauth-imap.properties
  $ chmod 600 office365-oauth-imap.properties
  ```

  Edit the .properties file to specify the necessary data and restart SavaPage to test the plug-in.

M.4. Notification Plug-in

Notification Plug-ins listen to application events, and processes them in a custom way. Currently Job Ticket Close and Cancel events are supported, for the following clients platforms:

- Smartschool[12]

See the `README.md` of the savapage-ext-notification[13] project for more information.

M.5. IPP Routing Plug-in

This plug-in is called on an IPP Routing event, just before the PDF is send to the Proxy Printer.

The plug-in supports PDF edit actions, like adding header and footer text, and a QR code[14] image. The QR code itself can optionally be a constant, a generated UUID or obtained by executing a third-party RESTful POST request that registers the print job and delivers the QR code in return.

> **Note**
>
> The QR code is shown in the Documents and Transactions Log.

[5] https://www.keycloak.org/
[6] https://www.google.com/
[7] https://login.microsoftonline.com/
[8] https://www.smartschool.be/
[9] https://gitlab.com/savapage-ext/savapage-ext-oauth.git
[10] https://en.wikipedia.org/wiki/OAuth
[11] https://outlook.office365.com
[12] https://www.smartschool.be/
[13] https://gitlab.com/savapage-ext/savapage-ext-notification.git
[14] https://en.wikipedia.org/wiki/QR_code

> **Tip**
>
> If the Document Store is enabled for Print Out, and Archive/Journal is not disabled for the target Proxy Printer, a third-party can retrieve the printed PDF by calling a SavaPage RESTful service with the QR code as `ext_id`. See Section C.2.3.2.6, “GET /documents/ext_id=<ext_id>/pdf” [349].

An annotated `savapage-ext-ipp-routing.properties.template` is present in the `/opt/savapage/ext` directory. Copy this template to a file with a `.properties` extension in the same directory and restrict permissions, since the file contains confidential information. An example:

```
$ sudo su - savapage
$ cd /opt/savapage/server/ext
$ cp savapage-ext-ipp-routing.properties.template savapage-ext-ipp-routing.properties
$ chmod 600 savapage-ext-ipp-routing.properties
```

Edit the .properties file to specify the necessary data and restart SavaPage to test the plug-in.

Apart from the standard plug-in keys, the plug-in file contains specific `routing.*` keys as explained in the table below.

Key	Description
routing.id	A fixed or generated routing ID stored as document ext_id (max 64 chars). See Section C.2.3.2.4, “GET /documents/ext_id=<ext_id>” [348] *Placeholder values:* • `$uuid$` : Generated UUID.
RESTful POST request to register print job and obtain QR code.	
routing.rest.uri	RESTful resource URI.
routing.rest.user routing.rest.password	HTTP Basic Authentication User and Password.
routing.rest.post.request.entity	The entity to be POST-ed. *Placeholder values:* • `$uuid$` : Generated UUID. • `$user_id$` : User id • `$client_ip$` : Client IP address • `$queue_name$` : Print-in Queue name • `$printer_name$` : CUPS printer name • `$printer_uri$` : CUPS printer URI • `$printer_uri_host$` : CUPS printer URI host component • `$job_name$` : Print Job Name (document name) • `$job_time$` : Print Job datetime in ISO_8601 format • `$page_width_mm$` : Print Job first page width (millimeters). • `$page_height_mm$` : Print Job first page height (millimeters).
routing.rest.post.request.mediatype	Media type[15] of request entity. There are tywo options: • `text/plain` • `application/json`
routing.rest.post.response.mediatype	Media type of request entity. There are two options: • `text/plain` : delivers the QR code as plain text.

[15] https://en.wikipedia.org/wiki/Media_type

Key	Description
	• `application/json` : delivers the JSON representaton of `org.savapage.ext.print.IppRoutingDto` POJO[a]. The routing "id" is mandatory. All other values override the ones that are configured in their respective `routing.pdf.*` keys of the .properties file (see keys below).
QR image	
routing.pdf.qrcode.content	QR code content. *Placeholder values:* • `$routing_id$` : routing.id
routing.pdf.qrcode.size	Square QR code size in millimeters (width == height).
routing.pdf.qrcode.qz	Quiet zone [b] in millimeters. When not specified, zero (0) is applied.
routing.pdf.qrcode.pos.anchor	Position in PDF document. `TL` (top-left, default), `TR` (top-right), `BL` (bottom-left), `BR` (bottom-right).
routing.pdf.qrcode.pos.margin.x routing.pdf.qrcode.pos.margin.y	Margins in millimeters.
Header	
routing.pdf.header.text	Header text: Helvetica Font, color Gray.
routing.pdf.header.font.size	Font size points (pt).
routing.pdf.header.margin.top	Top margin in millimeters.
Footer	
routing.pdf.footer.text	Footer text: Helvetica Font, color Gray.
routing.pdf.footer.font.size	Font size points (pt).
routing.pdf.footer.margin.bottom	Bottom margin in millimeters.
PDF info (optional)	
routing.pdf.info.title	PDF Title
routing.pdf.info.subject	PDF Subject
routing.pdf.info.author	PDF Author
routing.pdf.info.keywords	A space separated list of PDF keywords.

[a] `org.savapage.ext.print.IppRoutingDto` POJO can be found in https://gitlab.com/savapage/savapage-ext.

[b] The Quiet Zone (QZ) is the area around the QR code that is clear of text, graphics, and any other printing. The QZ helps isolate a QR code of its surroundings in order to guarantee a correct readability.

Table M.3. IPP Routing Plug-in: .properties keys

Some examples:

- A plug-in using SavaPage RESTful service to echo the generated `routing.pdf.qrcode.content`. This has no practical use other than to test the RESTful interface.

```
routing.id=$uuid$

routing.pdf.qrcode.rest.uri=https://localhost:8632/restful/v1/tests/echo
routing.pdf.qrcode.rest.user=admin
routing.pdf.qrcode.rest.password=password
routing.pdf.qrcode.rest.post.request.entity=$uuid$
```

```
routing.pdf.qrcode.rest.post.request.mediatype=text/plain
routing.pdf.qrcode.rest.post.response.mediatype=text/plain

routing.pdf.qrcode.size=12
routing.pdf.qrcode.qz=2
routing.pdf.qrcode.pos=BR
routing.pdf.qrcode.pos.margin.x=9
routing.pdf.qrcode.pos.margin.y=35

routing.pdf.header=Sample Header Text
routing.pdf.header.font.size=8
routing.pdf.header.margin.top=15

routing.pdf.footer=Sample Footer Text
routing.pdf.footer.font.size=6
routing.pdf.foooter.margin.bottom=10
```

- A plug-in calling a third-party RESTful service to obtain the `routing.pdf.qrcode.content`. See RESTful Client configuration properties.

```
routing.id=$uuid$

routing.pdf.qrcode.rest.uri=https://192.168.1.51:8080/register
routing.pdf.qrcode.rest.user=savapage
routing.pdf.qrcode.rest.password=elah7deV
routing.pdf.qrcode.rest.post.request.entity={"printUser":"$user_id$",\
    "printIp":"$client_ip$",\
    "printerName":"$printer_name$",\
    "docTitle":"$job_name$",\
    "printTime":"$job_time$"}
routing.pdf.qrcode.rest.post.request.mediatype=application/json
routing.pdf.qrcode.rest.post.response.mediatype=text/plain

routing.pdf.qrcode.size=12
routing.pdf.qrcode.qz=2
routing.pdf.qrcode.pos=BR
routing.pdf.qrcode.pos.margin.x=9
routing.pdf.qrcode.pos.margin.y=35
```

Note

Multiple IPP Routing Plug-ins can be active in `/opt/savapage/ext` directory. Each one will be executed upon an IPP Routing event. The order of execution is undefined.

M.6. User Source Plug-ins

See Section 4.11.2, “Custom” [154].

Tip

A sample CLI program `custom-auth-sync-sample.sh` with both plug-in interfaces combined is found in `/opt/savapage/server/examples/ext/` .

M.6.1. User Synchronization Plug-in

Arguments	stdout
`--list-users`	List of all users. Each user is separated by `\n`. User details must be supplied in the correct order separated by `\t` like this: `UserName\tFullName\tEmail\tCardNumber\tIDNumber\n`

Arguments	stdout
	`UserName` identifies the user and is mandatory. Other fields can be void with respect to the mandatory `\t` field separator. Note: Email is the **Primary email**.
`--list-groups`	List of all groups. Each group is separated by `\n`. Group details must be supplied in the correct order separated by `\t` like this: `GroupName\tFullName\n` `GroupName` identifies the group and is mandatory. `FullName` can be void with respect to the mandatory `\t` field separator.
`--get-user-details [UserName]`	`UserName` details separated by `\t` as in `--list-users`.
`--get-group-details [GroupName]`	`GroupName` details separated by `\t` as in `--list-groups`.
`--list-group-members [GroupName]`	List of users as in `--list-users` that are member of `GroupName`.
`--is-group-present [GroupName]`	`Y` if `GroupName` is present, `N` if not.
`--is-user-in-group [GroupName] [UserName]`	`Y` if `UserName` is a member of `GroupName`, `N` if not.
`--rfid-format`	The Card Number Format as `LSB|MSB` and `HEX|DEC`. For example: `LSB\tHEX\n`

Table M.4. User Synchronization Plug-in CLI

M.6.2. User Authentication Plug-in

stdin	stdout
`UserName\nPassword\n`	`OK` if authentication succeeded, `ERROR` if not.

Table M.5. User Authentication Plug-in CLI

Appendix N. PaperCut Integration

PaperCut is a popular print and copy management software product developed by PaperCut Software[1] based in Melbourne, Australia.

Some functions that are not present in PaperCut can be implemented with SavaPage as pre-processor and integrator.

Note

See Section 4.11.12.1, “PaperCut Integration” [175] on how to set the PaperCut Connectivity options.

N.1. Delegated Print to PaperCut

Delegated Print is integrated with PaperCut when the following conditions are met:
- PaperCut Integration is enabled: see Section 4.11.12.1, “PaperCut Integration” [175]
- Delegated Print and Delegated Print integration with PaperCut are enabled: see Section 4.11.9.2, “Proxy Print Delegation” [170] and Figure 4.125, “Admin Web App: Options - PaperCut Delegated Print Integration” [176].

The Proxy Printer must meet the following requirements:
- The Proxy Printer is managed by PaperCut.
- The Proxy Printer is configured as non-secure. See Section 4.11.9, “Proxy Print” [168]. If secure printing is required it must be configured in PaperCut and not in SavaPage.

Warning

A Delegated Print job is denied when the delegate or any of the delegators does not exist in PaperCut. This limitation can easily be solved when both systems synchronize from the same user source. See Section 4.11.1, “User Source” [150].

Tips for further reading:
- Section A.3.3, “Delegated Print - PaperCut Scenario” [325].
- Section A.3.2, “Delegated Print - Job Ticket - PaperCut - Scenario” [324].

N.1.1. PaperCut Configuration

N.1.1.1. Step 1 - Create shared account

Create a shared parent account called `SavaPage`. This top-level account *must* be present, since several sub-accounts will be lazy created by SavaPage.

In addition, any printer used for Delegated Print must be configured to charge to this account. See Section N.1.1.3, “Step 3 - Configure Printers” [413].

Note

The PaperCut shared account name is known by the SavaPage configuration key:

[1] https://www.papercut.com/

> `proxy-print.delegate.papercut.account.shared.parent`
>
> The value defaults to `SavaPage`. See Section 4.11.15, “Config Editor” [189] on how to change this value.

N.1.1.2. Step 2 - Enable Multiple Personal Accounts

Enable the PaperCut “Multiple Personal Accounts” option and add the personal account `SavaPage`. This account *must* be present for it is used by SavaPage to charge printing costs to individual persons.

> **Important**
>
> The PaperCut personal account name is known by the SavaPage configuration key:
>
> `proxy-print.delegate.papercut.account.personal`
>
> The value defaults to `SavaPage`.
>
> The account type for this account as determined by PaperCut in its own configuration key `multi-personal-accounts.definitions` (with values like `USER-001`, `USER-002`) can be set with this key:
>
> `proxy-print.delegate.papercut.account.personal-type`
>
> When a value is specified it is used to filter personal account transactions in JDBC queries (CSV downloads) for the Delegated Print context.
>
> See Section 4.11.15, “Config Editor” [189] on how to change these values.

N.1.1.3. Step 3 - Configure Printers

Take a moment to consider how you want the PaperCut printers that are used for SavaPage Delegated Print to act. A likely scenario is that you want these printers to be virtual hold/release queues so users can enjoy follow-me printing, and release print jobs at a series of physical printers. Or, may be you want these jobs to be released by administrators only. Consult the PaperCut User Manual on how implement the desired scenario.

There is one crucial printer configuration property though that must be addressed. Make sure that **Override user-level settings** is set, and activate **Do not show account popups and allocate jobs to:** a single shared account. Use the Shared Account `SavaPage` (as created in Step 1) and select the Charge shared account option.

Now, when a Delegated Print job is successfully printed by PaperCut, the cost will be automatically charged to the shared `SavaPage` account.

N.1.2. PaperCut Delegated Print Processing

SavaPage monitors the print job status in PaperCut and, when printing is successful, charges the costs, as reported by PaperCut, to the proper PaperCut accounts, as explained in the section below.

> **Important**
>
> In addition to PaperCut Account processing, account transactions are still added to SavaPage as explained in Section 3.5.8, “Delegated Print Edit” [57]. However, the printing costs reported by PaperCut overrule any costs defined in SavaPage for regular printing. *For Job Tickets, SavaPage cost is leading.*

N.1.3. PaperCut Delegated Print Accounting

SavaPage uses the PaperCut cost total of the Delegated Print job to add extra PaperCut account transactions. The *comment* of each account transaction holds a combination of pipe-separated (`|`) job parameter fields, with the following meaning:

- *delegate* : the user (delegate) who printed the job on behalf of the target users (delegators).
- *class* : the "class" a target user belongs to. A value of "`-`" (minus) means the user does not belong to a class. A class can be of type:
  - *shared* : a SavaPage Shared Account.
  - *group* : a SavaPage Group Account.
- *copies* : the number of document copies printed (a *negative* number for a Refund).
- *pages* : number of pages within the document.
- *size* : the page size of the document (A4, A3).
- *duplex* : D for duplex, S for simplex.
- *color* : C for color, G for grayscale.
- *id* : CUPS job ID, optionally preceded by Job Ticket Number or Tag.
- *document* : the document name.
- *comment* : any comment entered by the delegate (CR+LF, CR, LF and TAB are replaced by one space).

As a rule SavaPage charges target users individually for print copies.

As an extra, solely for reporting purposes, dedicated PaperCut shared accounts are used to accumulate cost and job information globally (the `SavaPage\Jobs` account) and per User Class.

Important

No transaction appears in any PaperCut account when the cost of a print job is *zero*. For transactions to appear you need to specify page cost at the PaperCut printer configuration.

N.1.3.1. PaperCut User Accounting

SavaPage proportionally splits the cost total of the printed Delegated Print job over individual users (delegators), directly or indirectly by group membership. Costs are charged to their *personal* user account named `SavaPage`. The comment format of the transaction is:

```
group | delegate | copies | pages | size | duplex | color | id | document | comment
```

Copies for users not belonging to a group have dummy group "`-`" (minus).

N.1.3.2. PaperCut User Class Accounting

SavaPage proportionally splits the cost total of the printed Delegated Print job over "classes". Class cost is proportional to the sum of the print copies for users belonging to the class. This cost is charged to a shared child account of the `SavaPage` parent account, with format:

```
savapage.[class].[name]
```

The `[class]` placeholder stands for the class *type* and can have value `group` or `shared`. The `[name]` is placeholder for the class *name*. The *name* of a `shared` child account is prefixed with the name of its parent, separated by a dot character (as `parent.child`).

PaperCut shared child accounts are *ad-hoc* created by SavaPage on first use.

The comment format of the transaction is:

```
delegate | copies | pages | size | duplex | color | id | document | comment
```

N.1.3.3. PaperCut Job Accounting

As an extra, for each Delegated Print job, SavaPage adds a transaction to the shared child account "`Jobs`" of the SavaPage parent account. The comment format of the transaction is:

```
delegate | copies | pages | size | duplex | color | id | delegate@class-1 |
copies-1 | ... | delegate@class-n | copies-n | document | comment
```

The `delegate@class | copies` field group shows the printed copies per class and is repeated for each class that was printed for. Copies for users not belonging to a class are accumulated in dummy class "`-`" (minus).

Note

The PaperCut shared child account name is known by the SavaPage configuration key:

```
proxy-print.delegate.papercut.account.shared.child.jobs
```

The value defaults to `Jobs`. See Section 4.11.15, "Config Editor" [189] on how to change this value.

N.1.4. PaperCut Queries and Reports

Use the on-line queries and run the reports in the PaperCut Admin Web App to answer questions about Delegated Print jobs and transactions from the following perspectives.

N.1.4.1. User Prints

SavaPage documents printed for a user.

Users → User List → User Details → Transactions gives a quick on-line view of personal account transactions. Sort and filter to find the SavaPage transactions.

Use Reports → Transaction Reports → Transaction logs to generate a SavaPage transaction report for a user or user group over a period of time. The report shows the individual transaction details and the total amount charged.

Important

PaperCut does *not* have a Transaction Report to accumulate transaction totals per user over a period of time. Please use the Delegator Invoicing from PaperCut export function in SavaPage for that purpose.

N.1.4.2. Class Prints

SavaPage documents printed for a class.

Select Accounts → Shared Account List → Account Details → Transaction for a SavaPage class account to get a quick on-line view of account transactions. Sort and filter to find transactions.

The Reports → Shared Account Reports section contains many reports that summarize printing activity charged to shared accounts. Select the SavaPage class account to get a transaction summary report for a period of time.

N.1.4.3. Delegate Prints

SavaPage documents printed by a delegate.

Users → User List → User Details → Job Log gives a quick on-line view of the documents printed by a delegate user. Sort and filter to find the print jobs charged to the shared SavaPage account.

N.1.4.4. Delegate Class Prints

SavaPage documents printed by a delegate for a class.

Select Accounts → Shared Account List → Account Details → Transaction for the SavaPage Jobs account to get a quick on-line view of account transactions. Since the comment contains formatted information about classes, you can select transactions with the "Comment containing" filter.

Likewise you can use Reports → Transaction Reports → Transaction logs to generate a SavaPage\Jobs transaction report for over a period of time. The report shows the individual transaction details and the total amount charged.

N.2. Delegated Print to PaperCut Front-end

If the requirements for Delegated Print to PaperCut are *not* met, SavaPage can still generate PaperCut Account Transactions for print jobs from a printer *not* managed by PaperCut, provided the following conditions are met:

- Configuration Property `proxy-print.delegate.papercut.front-end.enable` = `Y` (default `N`).
- The Enable checkbox for **PaperCut front-end** is checked in the Proxy Printer Edit dialog for the printer to which the job was sent.

When these conditions are met, SavaPage printing costs are applied within PaperCut, in the same way as described in Delegated Print to PaperCut, after CUPS notifies that the print job has been successfully completed.

N.3. Personal Print to PaperCut

Personal Print is integrated with PaperCut when the following conditions are met:

- PaperCut Integration is *enabled*.
- Delegated Print is *disabled*.
- Personal Print integration with PaperCut is *enabled*.
- The Proxy Printer is managed by PaperCut.

The job can be printed Non-Secure or in Hold or Direct Print Mode.

Tip

A proxy printer can be configured as hold/release in both SavaPage and PaperCut with different expiration times. In this way SavaPage can act as long-term gateway to short-term PaperCut follow-me printing. This can be an efficient strategy, because SavaPage storage is lean (just the PDF document with print job parameters), while PaperCut stores bulky spool files. *Since SavaPage cost is used to communicate PaperCut cost before print release, make sure SavaPage cost matches PaperCut cost.*

See Section A.2.3, “Personal Print - PaperCut Scenario” [321].

N.4. Advanced Print Configuration

N.4.1. PaperCut Print Log Monitoring

PaperCut print job status is monitored for a maximum number of minutes, as set in configuration item `proxy-print.papercut.print-log.max-mins` (default `7200`). See Section 4.11.15, “Config Editor” [189] on how to change this value.

Important

Disable any setting in PaperCut that requires a user to run the PaperCut client software. For instance, for pop-up authentication, shared account selection or print confirmation. Beware, that when a user prints to a PaperCut managed printer from the SavaPage User Web App, and action is required in a PaperCut popup dialog, the PaperCut Print Log Monitoring will never see the print job cancellation, when the user does not respond within time.

N.5. User Source Solutions

With the absence of a common User Source, SavaPage can act as Sync Source in PaperCut. Conversely, PaperCut can act as User Source for SavaPage. However, since PaperCut does not provide an API for User Authentication, in this case an extra shared Authentication Service is required.

N.5.1. SavaPage as Sync Source in PaperCut

With the absence of a common User Source, SavaPage can provide User Sync and Authentication Services for PaperCut. In this way Internal Users can be synchronized to and authenticated in PaperCut.

The interface is implemented as HTTP Basic Auth service. This interface can easily be used in Python or Linux `curl` scripts. Sample scripts for use in PaperCut Custom Sync[2] are available in `/opt/savapage/server/examples/papercut/`

Note

Rate Limiting is applied to unauthorized access.

Configuration property	Description
ext.papercut.user.sync.enable	Set to `Y` or `N` (default) to enable/disable PaperCut Custom User Sync Integration.
ext.papercut.user.sync.username	Basic HTTP Authentication User name.
ext.papercut.user.sync.password	Basic HTTP Authentication User password.
ext.papercut.user.sync.ip-addresses-allowed	The allowed client IP addresses as a CIDR Set. *When void, not a single client is allowed.*

Table N.1. SavaPage as Sync Source in PaperCut: Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these properties.

[2] https://www.papercut.com/kb/Main/CaseStudyCustomUserSyncIntegration

N.5.2. PaperCut as User Source in SavaPage

PaperCut can be set as User Source in SavaPage by setting configuration property `custom.user-sync.papercut` to `Y`. There are three extra settings needed to make this work:

1. PaperCut Integration must be enabled, because a connection to the PaperCut Server (XML-RPC API) is needed to select the users that were synchronized by PaperCut from their User Source.
2. Because User Authentication is not part of the PaperCut API, authentication must be configured as OAuth Client Plug-in that connects to the same User Source configured in PaperCut.
3. User Source must be disabled by selecting the `-` option. This will show PaperCut as User Source.

Figure N.1. Admin Web App: Options - PaperCut User Source

Configuration property	Description
custom.user-sync.papercut	Set to `Y` or `N` (default) to enable/disable PaperCut as User Source.
custom.user-sync.papercut.user-card-number.first-byte	NFC Card Number First Byte: `MSB` or `LSB` (default).
custom.user-sync.papercut.user-card-number.format	NFC Card Number Format: `DEC` or `HEX` (default).

Table N.2. PaperCut as User Source in SavaPage: Configuration Properties

See Section 4.11.15, “Config Editor” [189] on how to set these properties.

N.6. PaperCut Personal User Account

Preview

PaperCut Personal User Account is a Feature Preview that is partly operational and impermanent. The preview is offered to provoke feedback based on real world use. Please tell us your experience.

Visibility of the PaperCut Personal User Account can be enabled.

Configuration property	Value
financial.user.account.papercut.enable	Set to `Y` or `N` (default) to enable/disable visibility of PaperCut Personal Account. The effect of enabled visibility is that a SavaPage and PaperCut icon is used as amount prefix to distinguish between the two account types. See the following sections: • Figure 4.27, “Admin Web App: User - PaperCut Account Balance” [101]. • Figure 4.53, “Admin Web App: Account - SavaPage Account Balance ” [118]. • Figure 3.72, “User Web App: Log - SavaPage User Account Transaction” [74]. • Section 3.10.5, “Financial” [83].

Configuration property	Value
	• Section 8.2, “Payment Page” [232]. • Section 8.2.1, “Payment Dialog” [233]. Also see Section 7.2, “Sales” [226].

Table N.3. PaperCut Personal User Account Configuration Settings

See Section 4.11.15, “Config Editor” [189] on how to enter this property.

N.7. Integration Pitfalls

The state of the two systems involved in the print chain (SavaPage, PaperCut) is tightly coupled. Restoring an earlier backup of either system can break the work-flow for pending jobs and lead to unwanted results. For instance:

- *When a backup of SavaPage is restored*, it will not handle jobs that were submitted to PaperCut after the backup point. In these cases SavaPage will show a print job status that does not reflect the real situation. On the other side, jobs that were already fully processed, might be re-processed by SavaPage, leading to extra charges on the shared PaperCut accounts.
- *When a backup of PaperCut is restored*, SavaPage will not find PaperCut print status information for *pending* jobs that were submitted to PaperCut after the backup point. In these cases SavaPage will show an *error* print job status, when in real the job is *completed* or *cancelled*.

To avoid integration problems, review your backup and restore strategy carefully.

Make sure you create backups of both SavaPage and PaperCut at the same point in time. Also, be sure that at the time of backup all pending print jobs are fully processed. When you need to restore, use backups of the same snapshot time, first restore PaperCut and than SavaPage.

Appendix O. Job Scheduling

SavaPage background jobs are scheduled with Cron Trigger Format expressions.

O.1. Cron Trigger Format

Cron Trigger Format is similar to GNU/Linux `cron` scheduling expressions. An extensive tutorial can be found here[1], and a simplified introduction is presented in this section.

A Cron Trigger is an expression comprised of 6 mandatory fields separated by white space.

Field Name	Range	Special	Remark
Seconds	`0-59`	`, - *`	Not relevant in our case: can be zero.
Minutes	`0-59`	`, - *`	
Hours	`0-23`	`, - *`	
Day of month	`1-31`	`, - * ?`	
Month	`1-12`	`, - *`	Range is identical to `JAN-DEC`.
Day of week	`1-7`	`, - * ?`	Range is identical to `SUN-SAT`. Note: GNU/Linux `cron` uses range `0-6`.

Table O.1. Cron Trigger Format - simplified

Special character values are:

- **`,`** : used to specify *additional values*. For example, “2,4,6” in the day-of-week field means the days Monday, Wednesday, and Friday.
- **`-`** : used to specify *ranges*. For example, “2-6” in the day-of-week field means the days Monday to Friday.
- **`*`** : used to select *all values* within a field. For example: `*` in the Hours field means “every hour”.
- **`?`** : used to select *no specific value*. This is useful when you need to specify something in one of the two fields in which the character is allowed, but not the other. For example, if I want a trigger to fire on a particular day of the week (say, Sunday), but don’t care what day of the month that happens to be, you would put `1` in the day-of-week field, and `?` in the day-of-month field.

O.2. Hourly Jobs

Hourly Jobs execute according to the default Cron Trigger Expression `"0 0 * * * ?"` contained in configuration key `schedule.hourly`. See Section 4.11.15, “Config Editor” [189] on how to change this value. The following jobs are executed:

- Renewal of CUPS notification subscription for CUPS Notifier. See CUPS Configuration Properties.

O.3. Daily Jobs

Regular Daily Jobs execute 10 minutes past midnight, according to the default Cron Trigger Expression `"0 10 0 * * ?"` contained in configuration key `schedule.daily`. See Section 4.11.15, “Config Editor” [189] on how to change this value. The following jobs are executed:

- User Synchronization

[1] https://www.quartz-scheduler.org/documentation/quartz-2.3.0/tutorials/tutorial-lesson-06.html

- Document Store Cleanup.
- Printer Group Cleanup.
- Community Membership Check.

O.4. Daily Maintenance

Overnight maintenance synchronization takes place 55 minutes past midnight, according to the default Cron Trigger Expression `"0 55 0 * * ?"` contained in configuration key `schedule.daily-maintenance`. See Section 4.11.15, "Config Editor" [189] on how to change this value. The following jobs are executed:

- Printer SNMP refresh.
- User Home Clean.

O.5. Weekly Jobs

Default weekly backups take place at 20 minutes past midnight on Sunday morning [2], as in the Cron Trigger Expression `"0 20 0 ? * 1"` contained in configuration key `schedule.weekly`. See Section 4.11.15, "Config Editor" [189] on how to change this value. The following jobs are executed:

- Database Backup and Cleanup.

O.6. Monthly Jobs

This schedule is reserved for future monthly jobs and will be performed 30 minutes past midnight on the first day of each month, according to the default Cron Trigger Expression `"0 30 0 1 * ?"` contained in configuration key `schedule.monthly`.

O.7. Atom Feed Job

Data for Atom Feed Service is prepared by an Atom Feed job that runs on Tuesday till Saturday at 3:00 AM, as in the Cron Trigger Expression `"0 0 3 ? * 3-7"` contained in configuration key `feed.atom.admin.schedule`. See Section 4.11.15, "Config Editor" [189] on how to change this value.

O.8. Matrix Summary Job

A daily summary of operational data (equivalent to data published by the Atom Feed Service) is send by a Matrix Summary job that runs on Tuesday till Saturday at 8:00 AM, as in the Cron Trigger Expression `"0 0 8 ? * 3-7"` contained in configuration key `system.matrix.publish.summary.day.schedule`. See Section C.4, "Matrix Messaging Service" [352].

[2] The 20 minutes past midnight for the weekly job is chosen to avoid a potential performance bottleneck with server processes that occur exactly at midnight.

Appendix P. GNU Affero General Public License (AGPL)

```
                    GNU AFFERO GENERAL PUBLIC LICENSE
                       Version 3, 19 November 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The GNU Affero General Public License is a free, copyleft license for
software and other kinds of works, specifically designed to ensure
cooperation with the community in the case of network server software.

  The licenses for most software and other practical works are designed
to take away your freedom to share and change the works.  By contrast,
our General Public Licenses are intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

  Developers that use our General Public Licenses protect your rights
with two steps: (1) assert copyright on the software, and (2) offer
you this License which gives you legal permission to copy, distribute
and/or modify the software.

  A secondary benefit of defending all users' freedom is that
improvements made in alternate versions of the program, if they
receive widespread use, become available for other developers to
incorporate.  Many developers of free software are heartened and
encouraged by the resulting cooperation.  However, in the case of
software used on network servers, this result may fail to come about.
The GNU General Public License permits making a modified version and
letting the public access it on a server without ever releasing its
source code to the public.

  The GNU Affero General Public License is designed specifically to
ensure that, in such cases, the modified source code becomes available
to the community.  It requires the operator of a network server to
provide the source code of the modified version running there to the
users of that server.  Therefore, public use of a modified version, on
a publicly accessible server, gives the public access to the source
code of the modified version.

  An older license, called the Affero General Public License and
published by Affero, was designed to accomplish similar goals.  This is
a different license, not a version of the Affero GPL, but Affero has
released a new version of the Affero GPL which permits relicensing under
this license.
```

```
  The precise terms and conditions for copying, distribution and
modification follow.

                       TERMS AND CONDITIONS

  0. Definitions.

  "This License" refers to version 3 of the GNU Affero General Public License.

  "Copyright" also means copyright-like laws that apply to other kinds of
works, such as semiconductor masks.

  "The Program" refers to any copyrightable work licensed under this
License.  Each licensee is addressed as "you".  "Licensees" and
"recipients" may be individuals or organizations.

  To "modify" a work means to copy from or adapt all or part of the work
in a fashion requiring copyright permission, other than the making of an
exact copy.  The resulting work is called a "modified version" of the
earlier work or a work "based on" the earlier work.

  A "covered work" means either the unmodified Program or a work based
on the Program.

  To "propagate" a work means to do anything with it that, without
permission, would make you directly or secondarily liable for
infringement under applicable copyright law, except executing it on a
computer or modifying a private copy.  Propagation includes copying,
distribution (with or without modification), making available to the
public, and in some countries other activities as well.

  To "convey" a work means any kind of propagation that enables other
parties to make or receive copies.  Mere interaction with a user through
a computer network, with no transfer of a copy, is not conveying.

  An interactive user interface displays "Appropriate Legal Notices"
to the extent that it includes a convenient and prominently visible
feature that (1) displays an appropriate copyright notice, and (2)
tells the user that there is no warranty for the work (except to the
extent that warranties are provided), that licensees may convey the
work under this License, and how to view a copy of this License.  If
the interface presents a list of user commands or options, such as a
menu, a prominent item in the list meets this criterion.

  1. Source Code.

  The "source code" for a work means the preferred form of the work
for making modifications to it.  "Object code" means any non-source
form of a work.

  A "Standard Interface" means an interface that either is an official
standard defined by a recognized standards body, or, in the case of
interfaces specified for a particular programming language, one that
is widely used among developers working in that language.

  The "System Libraries" of an executable work include anything, other
than the work as a whole, that (a) is included in the normal form of
packaging a Major Component, but which is not part of that Major
Component, and (b) serves only to enable use of the work with that
Major Component, or to implement a Standard Interface for which an
implementation is available to the public in source code form.  A
"Major Component", in this context, means a major essential component
(kernel, window system, and so on) of the specific operating system
(if any) on which the executable work runs, or a compiler used to
produce the work, or an object code interpreter used to run it.

  The "Corresponding Source" for a work in object code form means all
```

```
the source code needed to generate, install, and (for an executable
work) run the object code and to modify the work, including scripts to
control those activities.  However, it does not include the work's
System Libraries, or general-purpose tools or generally available free
programs which are used unmodified in performing those activities but
which are not part of the work.  For example, Corresponding Source
includes interface definition files associated with source files for
the work, and the source code for shared libraries and dynamically
linked subprograms that the work is specifically designed to require,
such as by intimate data communication or control flow between those
subprograms and other parts of the work.

  The Corresponding Source need not include anything that users
can regenerate automatically from other parts of the Corresponding
Source.

  The Corresponding Source for a work in source code form is that
same work.

  2. Basic Permissions.

  All rights granted under this License are granted for the term of
copyright on the Program, and are irrevocable provided the stated
conditions are met.  This License explicitly affirms your unlimited
permission to run the unmodified Program.  The output from running a
covered work is covered by this License only if the output, given its
content, constitutes a covered work.  This License acknowledges your
rights of fair use or other equivalent, as provided by copyright law.

  You may make, run and propagate covered works that you do not
convey, without conditions so long as your license otherwise remains
in force.  You may convey covered works to others for the sole purpose
of having them make modifications exclusively for you, or provide you
with facilities for running those works, provided that you comply with
the terms of this License in conveying all material for which you do
not control copyright.  Those thus making or running the covered works
for you must do so exclusively on your behalf, under your direction
and control, on terms that prohibit them from making any copies of
your copyrighted material outside their relationship with you.

  Conveying under any other circumstances is permitted solely under
the conditions stated below.  Sublicensing is not allowed; section 10
makes it unnecessary.

  3. Protecting Users' Legal Rights From Anti-Circumvention Law.

  No covered work shall be deemed part of an effective technological
measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
measures.

  When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

  4. Conveying Verbatim Copies.

  You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
```

```
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

  You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

  5. Conveying Modified Source Versions.

  You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

    a) The work must carry prominent notices stating that you modified
    it, and giving a relevant date.

    b) The work must carry prominent notices stating that it is
    released under this License and any conditions added under section
    7.  This requirement modifies the requirement in section 4 to
    "keep intact all notices".

    c) You must license the entire work, as a whole, under this
    License to anyone who comes into possession of a copy.  This
    License will therefore apply, along with any applicable section 7
    additional terms, to the whole of the work, and all its parts,
    regardless of how they are packaged.  This License gives no
    permission to license the work in any other way, but it does not
    invalidate such permission if you have separately received it.

    d) If the work has interactive user interfaces, each must display
    Appropriate Legal Notices; however, if the Program has interactive
    interfaces that do not display Appropriate Legal Notices, your
    work need not make them do so.

  A compilation of a covered work with other separate and independent
works, which are not by their nature extensions of the covered work,
and which are not combined with it such as to form a larger program,
in or on a volume of a storage or distribution medium, is called an
"aggregate" if the compilation and its resulting copyright are not
used to limit the access or legal rights of the compilation's users
beyond what the individual works permit.  Inclusion of a covered work
in an aggregate does not cause this License to apply to the other
parts of the aggregate.

  6. Conveying Non-Source Forms.

  You may convey a covered work in object code form under the terms
of sections 4 and 5, provided that you also convey the
machine-readable Corresponding Source under the terms of this License,
in one of these ways:

    a) Convey the object code in, or embodied in, a physical product
    (including a physical distribution medium), accompanied by the
    Corresponding Source fixed on a durable physical medium
    customarily used for software interchange.

    b) Convey the object code in, or embodied in, a physical product
    (including a physical distribution medium), accompanied by a
    written offer, valid for at least three years and valid for as
    long as you offer spare parts or customer support for that product
    model, to give anyone who possesses the object code either (1) a
    copy of the Corresponding Source for all the software in the
    product that is covered by this License, on a durable physical
    medium customarily used for software interchange, for a price no
    more than your reasonable cost of physically performing this
    conveying of source, or (2) access to copy the
    Corresponding Source from a network server at no charge.
```

```
    c) Convey individual copies of the object code with a copy of the
    written offer to provide the Corresponding Source.  This
    alternative is allowed only occasionally and noncommercially, and
    only if you received the object code with such an offer, in accord
    with subsection 6b.

    d) Convey the object code by offering access from a designated
    place (gratis or for a charge), and offer equivalent access to the
    Corresponding Source in the same way through the same place at no
    further charge.  You need not require recipients to copy the
    Corresponding Source along with the object code.  If the place to
    copy the object code is a network server, the Corresponding Source
    may be on a different server (operated by you or a third party)
    that supports equivalent copying facilities, provided you maintain
    clear directions next to the object code saying where to find the
    Corresponding Source.  Regardless of what server hosts the
    Corresponding Source, you remain obligated to ensure that it is
    available for as long as needed to satisfy these requirements.

    e) Convey the object code using peer-to-peer transmission, provided
    you inform other peers where the object code and Corresponding
    Source of the work are being offered to the general public at no
    charge under subsection 6d.

  A separable portion of the object code, whose source code is excluded
from the Corresponding Source as a System Library, need not be
included in conveying the object code work.

  A "User Product" is either (1) a "consumer product", which means any
tangible personal property which is normally used for personal, family,
or household purposes, or (2) anything designed or sold for incorporation
into a dwelling.  In determining whether a product is a consumer product,
doubtful cases shall be resolved in favor of coverage.  For a particular
product received by a particular user, "normally used" refers to a
typical or common use of that class of product, regardless of the status
of the particular user or of the way in which the particular user
actually uses, or expects or is expected to use, the product.  A product
is a consumer product regardless of whether the product has substantial
commercial, industrial or non-consumer uses, unless such uses represent
the only significant mode of use of the product.

  "Installation Information" for a User Product means any methods,
procedures, authorization keys, or other information required to install
and execute modified versions of a covered work in that User Product from
a modified version of its Corresponding Source.  The information must
suffice to ensure that the continued functioning of the modified object
code is in no case prevented or interfered with solely because
modification has been made.

  If you convey an object code work under this section in, or with, or
specifically for use in, a User Product, and the conveying occurs as
part of a transaction in which the right of possession and use of the
User Product is transferred to the recipient in perpetuity or for a
fixed term (regardless of how the transaction is characterized), the
Corresponding Source conveyed under this section must be accompanied
by the Installation Information.  But this requirement does not apply
if neither you nor any third party retains the ability to install
modified object code on the User Product (for example, the work has
been installed in ROM).

  The requirement to provide Installation Information does not include a
requirement to continue to provide support service, warranty, or updates
for a work that has been modified or installed by the recipient, or for
the User Product in which it has been modified or installed.  Access to a
network may be denied when the modification itself materially and
adversely affects the operation of the network or violates the rules and
protocols for communication across the network.
```

```
  Corresponding Source conveyed, and Installation Information provided,
in accord with this section must be in a format that is publicly
documented (and with an implementation available to the public in
source code form), and must require no special password or key for
unpacking, reading or copying.

  7. Additional Terms.

  "Additional permissions" are terms that supplement the terms of this
License by making exceptions from one or more of its conditions.
Additional permissions that are applicable to the entire Program shall
be treated as though they were included in this License, to the extent
that they are valid under applicable law.  If additional permissions
apply only to part of the Program, that part may be used separately
under those permissions, but the entire Program remains governed by
this License without regard to the additional permissions.

  When you convey a copy of a covered work, you may at your option
remove any additional permissions from that copy, or from any part of
it.  (Additional permissions may be written to require their own
removal in certain cases when you modify the work.)  You may place
additional permissions on material, added by you to a covered work,
for which you have or can give appropriate copyright permission.

  Notwithstanding any other provision of this License, for material you
add to a covered work, you may (if authorized by the copyright holders of
that material) supplement the terms of this License with terms:

    a) Disclaiming warranty or limiting liability differently from the
    terms of sections 15 and 16 of this License; or

    b) Requiring preservation of specified reasonable legal notices or
    author attributions in that material or in the Appropriate Legal
    Notices displayed by works containing it; or

    c) Prohibiting misrepresentation of the origin of that material, or
    requiring that modified versions of such material be marked in
    reasonable ways as different from the original version; or

    d) Limiting the use for publicity purposes of names of licensors or
    authors of the material; or

    e) Declining to grant rights under trademark law for use of some
    trade names, trademarks, or service marks; or

    f) Requiring indemnification of licensors and authors of that
    material by anyone who conveys the material (or modified versions of
    it) with contractual assumptions of liability to the recipient, for
    any liability that these contractual assumptions directly impose on
    those licensors and authors.

  All other non-permissive additional terms are considered "further
restrictions" within the meaning of section 10.  If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further
restriction, you may remove that term.  If a license document contains
a further restriction but permits relicensing or conveying under this
License, you may add to a covered work material governed by the terms
of that license document, provided that the further restriction does
not survive such relicensing or conveying.

  If you add terms to a covered work in accord with this section, you
must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating
where to find the applicable terms.
```

```
  Additional terms, permissive or non-permissive, may be stated in the
form of a separately written license, or stated as exceptions;
the above requirements apply either way.

  8. Termination.

  You may not propagate or modify a covered work except as expressly
provided under this License.  Any attempt otherwise to propagate or
modify it is void, and will automatically terminate your rights under
this License (including any patent licenses granted under the third
paragraph of section 11).

  However, if you cease all violation of this License, then your
license from a particular copyright holder is reinstated (a)
provisionally, unless and until the copyright holder explicitly and
finally terminates your license, and (b) permanently, if the copyright
holder fails to notify you of the violation by some reasonable means
prior to 60 days after the cessation.

  Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies you of the
violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.

  Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
this License.  If your rights have been terminated and not permanently
reinstated, you do not qualify to receive new licenses for the same
material under section 10.

  9. Acceptance Not Required for Having Copies.

  You are not required to accept this License in order to receive or
run a copy of the Program.  Ancillary propagation of a covered work
occurring solely as a consequence of using peer-to-peer transmission
to receive a copy likewise does not require acceptance.  However,
nothing other than this License grants you permission to propagate or
modify any covered work.  These actions infringe copyright if you do
not accept this License.  Therefore, by modifying or propagating a
covered work, you indicate your acceptance of this License to do so.

  10. Automatic Licensing of Downstream Recipients.

  Each time you convey a covered work, the recipient automatically
receives a license from the original licensors, to run, modify and
propagate that work, subject to this License.  You are not responsible
for enforcing compliance by third parties with this License.

  An "entity transaction" is a transaction transferring control of an
organization, or substantially all assets of one, or subdividing an
organization, or merging organizations.  If propagation of a covered
work results from an entity transaction, each party to that
transaction who receives a copy of the work also receives whatever
licenses to the work the party's predecessor in interest had or could
give under the previous paragraph, plus a right to possession of the
Corresponding Source of the work from the predecessor in interest, if
the predecessor has it or can get it with reasonable efforts.

  You may not impose any further restrictions on the exercise of the
rights granted or affirmed under this License.  For example, you may
not impose a license fee, royalty, or other charge for exercise of
rights granted under this License, and you may not initiate litigation
(including a cross-claim or counterclaim in a lawsuit) alleging that
any patent claim is infringed by making, using, selling, offering for
sale, or importing the Program or any portion of it.
```

```
  11. Patents.

  A "contributor" is a copyright holder who authorizes use under this
License of the Program or a work on which the Program is based.  The
work thus licensed is called the contributor's "contributor version".

  A contributor's "essential patent claims" are all patent claims
owned or controlled by the contributor, whether already acquired or
hereafter acquired, that would be infringed by some manner, permitted
by this License, of making, using, or selling its contributor version,
but do not include claims that would be infringed only as a
consequence of further modification of the contributor version.  For
purposes of this definition, "control" includes the right to grant
patent sublicenses in a manner consistent with the requirements of
this License.

  Each contributor grants you a non-exclusive, worldwide, royalty-free
patent license under the contributor's essential patent claims, to
make, use, sell, offer for sale, import and otherwise run, modify and
propagate the contents of its contributor version.

  In the following three paragraphs, a "patent license" is any express
agreement or commitment, however denominated, not to enforce a patent
(such as an express permission to practice a patent or covenant not to
sue for patent infringement).  To "grant" such a patent license to a
party means to make such an agreement or commitment not to enforce a
patent against the party.

  If you convey a covered work, knowingly relying on a patent license,
and the Corresponding Source of the work is not available for anyone
to copy, free of charge and under the terms of this License, through a
publicly available network server or other readily accessible means,
then you must either (1) cause the Corresponding Source to be so
available, or (2) arrange to deprive yourself of the benefit of the
patent license for this particular work, or (3) arrange, in a manner
consistent with the requirements of this License, to extend the patent
license to downstream recipients.  "Knowingly relying" means you have
actual knowledge that, but for the patent license, your conveying the
covered work in a country, or your recipient's use of the covered work
in a country, would infringe one or more identifiable patents in that
country that you have reason to believe are valid.

  If, pursuant to or in connection with a single transaction or
arrangement, you convey, or propagate by procuring conveyance of, a
covered work, and grant a patent license to some of the parties
receiving the covered work authorizing them to use, propagate, modify
or convey a specific copy of the covered work, then the patent license
you grant is automatically extended to all recipients of the covered
work and works based on it.

  A patent license is "discriminatory" if it does not include within
the scope of its coverage, prohibits the exercise of, or is
conditioned on the non-exercise of one or more of the rights that are
specifically granted under this License.  You may not convey a covered
work if you are a party to an arrangement with a third party that is
in the business of distributing software, under which you make payment
to the third party based on the extent of your activity of conveying
the work, and under which the third party grants, to any of the
parties who would receive the covered work from you, a discriminatory
patent license (a) in connection with copies of the covered work
conveyed by you (or copies made from those copies), or (b) primarily
for and in connection with specific products or compilations that
contain the covered work, unless you entered into that arrangement,
or that patent license was granted, prior to 28 March 2007.

  Nothing in this License shall be construed as excluding or limiting
```

```
any implied license or other defenses to infringement that may
otherwise be available to you under applicable patent law.

  12. No Surrender of Others' Freedom.

  If conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License.  If you cannot convey a
covered work so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you may
not convey it at all.  For example, if you agree to terms that obligate you
to collect a royalty for further conveying from those to whom you convey
the Program, the only way you could satisfy both those terms and this
License would be to refrain entirely from conveying the Program.

  13. Remote Network Interaction; Use with the GNU General Public License.

  Notwithstanding any other provision of this License, if you modify the
Program, your modified version must prominently offer all users
interacting with it remotely through a computer network (if your version
supports such interaction) an opportunity to receive the Corresponding
Source of your version by providing access to the Corresponding Source
from a network server at no charge, through some standard or customary
means of facilitating copying of software.  This Corresponding Source
shall include the Corresponding Source for any work covered by version 3
of the GNU General Public License that is incorporated pursuant to the
following paragraph.

  Notwithstanding any other provision of this License, you have
permission to link or combine any covered work with a work licensed
under version 3 of the GNU General Public License into a single
combined work, and to convey the resulting work.  The terms of this
License will continue to apply to the part which is the covered work,
but the work with which it is combined will remain governed by version
3 of the GNU General Public License.

  14. Revised Versions of this License.

  The Free Software Foundation may publish revised and/or new versions of
the GNU Affero General Public License from time to time.  Such new versions
will be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

  Each version is given a distinguishing version number.  If the
Program specifies that a certain numbered version of the GNU Affero General
Public License "or any later version" applies to it, you have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software
Foundation.  If the Program does not specify a version number of the
GNU Affero General Public License, you may choose any version ever published
by the Free Software Foundation.

  If the Program specifies that a proxy can decide which future
versions of the GNU Affero General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

  Later license versions may give you additional or different
permissions.  However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

  15. Disclaimer of Warranty.

  THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW.  EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
```

```
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.  THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU.  SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. Limitation of Liability.

  IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

  17. Interpretation of Sections 15 and 16.

  If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

                      END OF TERMS AND CONDITIONS
```