

# Altova FlowForce Server 2021

ユーザーマニュアル

# Altova FlowForce Server 2021 ユーザーマニュアル

All rights reserved. No parts of this work may be reproduced in any form or by any means – graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems – without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

公開日: 2015–2021

(C) 2015–2021 Altova GmbH

---

# 目次

|          |                            |           |
|----------|----------------------------|-----------|
| <b>1</b> | <b>はじめに</b>                | <b>13</b> |
| 1.1      | 新規機能                       | 14        |
| 1.2      | 基本概念                       | 18        |
| 1.3      | セキュリティ概念                   | 20        |
| 1.4      | 動作の説明                      | 21        |
| 1.5      | Web 管理インターフェイス             | 23        |
| 1.6      | FlowForce Server へのログオン    | 25        |
| <br>     |                            |           |
| <b>2</b> | <b>インストール</b>              | <b>27</b> |
| 2.1      | システム必要条件                   | 27        |
| 2.2      | プロセッサコアとライセンス              | 28        |
| 2.3      | 重要なパス                      | 29        |
| 2.4      | Linux                      | 30        |
| 2.4.1    | FlowForce Server のインストール   | 30        |
| 2.4.2    | FlowForce Server 2021 への移行 | 31        |
| 2.4.3    | 現在インストールされている製品の確認         | 32        |
| 2.4.4    | FlowForce Server のアンインストール | 32        |
| 2.4.5    | インストール ディレクトリ              | 32        |
| 2.5      | macOS                      | 33        |
| 2.5.1    | FlowForce Server のインストール   | 33        |
| 2.5.2    | FlowForce Server 2021 への移行 | 34        |
| 2.5.3    | FlowForce Server のアンインストール | 35        |
| 2.5.4    | インストールディレクトリ               | 35        |
| 2.6      | Windows                    | 36        |
| 2.6.1    | FlowForce Server のインストール   | 36        |
| 2.6.2    | Windows サービスの構成            | 40        |
| 2.6.3    | ネットワーク構成 (Windows のみ)      | 41        |
| 2.6.4    | データを手動で移行する                | 44        |
| 2.6.5    | インストールフォルダー                | 45        |

|          |  |           |
|----------|--|-----------|
| 2.7      | Altova Server の移行.....                                       | 46        |
| <b>3</b> | <b>サーバーの構成</b> .....   | <b>48</b> |
| 3.1      | セットアップ.....  | 49        |
| 3.2      | ネットワーク設定の定義.....   | 53        |
| 3.2.1    | ファイル参照の構成.....   | 56        |
| 3.3      | SSL 暗号化のセットアップ.....  | 61        |
| 3.3.1    | 証明書機関により SSL 証明書を署名する方法.....                                 | 61        |
| 3.3.2    | 自己署名済みの SSL 証明書の作成方法.....                                    | 63        |
| 3.3.3    | 秘密キーの必要条件.....   | 73        |
| 3.3.4    | FlowForce Web サーバーのために SSL を有効化する方法.....                     | 74        |
| 3.3.5    | FlowForce サーバーのために SSL を有効化する方法.....                         | 76        |
| 3.3.6    | FlowForce Web Server と FlowForce サーバー間に SSL<br>を有効化する方法..... | 77        |
| 3.4      | デフォルトのタイムゾーンの設定.....   | 79        |
| 3.5      | 電子メールのパラメーターの設定.....   | 80        |
| 3.6      | ディレクトリサービスの設定.....   | 81        |
| 3.7      | ログ設定.....  | 84        |
| 3.8      | サービスの開始と停止 (Linux).....                                      | 89        |
| 3.9      | サービスの開始と停止 (macOS).....                                      | 90        |
| 3.10     | サービスの開始と停止 (Windows).....                                    | 91        |
| 3.11     | FlowForce Server アプリケーションデータ.....                            | 92        |
| 3.12     | FlowForce Server のローカライズ.....                                | 94        |
| 3.13     | バックアップと復元.....   | 95        |
| <b>4</b> | <b>ユーザーアクセスの管理</b> .....                                     | <b>98</b> |
| 4.1      | ユーザーとロール.....  | 99        |
| 4.1.1    | ユーザーの作成.....   | 99        |
| 4.1.2    | ロールの作成.....  | 100       |
| 4.1.3    | ドメインユーザーとロールをインポート.....                                      | 100       |
| 4.1.4    | デフォルトのユーザーとロール.....  | 101       |
| 4.1.5    | ユーザーとロールの名前の変更.....  | 102       |
| 4.1.6    | ユーザーにロールを割り当てる.....  | 103       |
| 4.1.7    | ロールを他のロールに割り当てる.....   | 103       |

|       |                       |     |
|-------|-----------------------|-----|
| 4.1.8 | ルートパスワードの再設定          | 104 |
| 4.2   | 特権                    | 105 |
| 4.2.1 | 特権の作動方法               | 105 |
| 4.2.2 | 特権レポートの閲覧             | 108 |
| 4.3   | パーミッションとコンテナ          | 111 |
| 4.3.1 | パーミッションの動作方法          | 111 |
| 4.3.2 | コンテナを理解する             | 114 |
| 4.3.3 | コンテナの作成、名前変更、および移動    | 116 |
| 4.3.4 | コンテナパーミッションの閲覧        | 118 |
| 4.3.5 | コンテナパーミッションの設定        | 119 |
| 4.3.6 | /public コンテナへのアクセスの制限 | 120 |
| 4.4   | パスワードポリシー             | 122 |
| 4.4.1 | パスワードポリシーの動作方法        | 122 |
| 4.4.2 | パスワードポリシーの作成と割り当て     | 122 |

## 5 ジョブの構成 124

|       |                     |     |
|-------|---------------------|-----|
| 5.1   | 資格情報                | 125 |
| 5.1.1 | 資格情報の定義             | 125 |
| 5.1.2 | OAuth 2.0 資格情報      | 127 |
| 5.1.3 | ジョブから資格情報を参照する方法    | 128 |
| 5.2   | ジョブの作成              | 130 |
| 5.3   | ジョブの複製              | 131 |
| 5.4   | 入力パラメーターの管理         | 132 |
| 5.5   | ステップの管理             | 134 |
| 5.6   | 順番にステップを処理する        | 136 |
| 5.7   | 条件付でステップを処理         | 138 |
| 5.8   | ステップのエラーの処理         | 140 |
| 5.8.1 | エラー時に再試行する          | 142 |
| 5.8.2 | 実行する際の注意点           | 144 |
| 5.9   | 延期されたステップ           | 147 |
| 5.10  | ジョブの戻り値の型を宣言する      | 152 |
| 5.11  | データ型をステップで処理する      | 153 |
| 5.12  | Windows ネットワークパスの参照 | 156 |
| 5.13  | トリガーの管理             | 157 |

---

|          |                                     |            |
|----------|-------------------------------------|------------|
| 5.13.1   | タイマートリガー.....                       | 158        |
| 5.13.2   | ファイルシステムトリガー.....                   | 159        |
| 5.13.3   | HTTPトリガー.....                       | 161        |
| 5.14     | ジョブを Web サービスとして公開する.....           | 163        |
| 5.15     | ジョブの結果をキャッシュする.....                 | 167        |
| 5.16     | キュー設定の定義.....                       | 170        |
| 5.17     | Windows ネットワークパスの参照.....            | 173        |
| 5.18     | FTP サポート.....                       | 174        |
| 5.19     | ジョブのインポートとエクスポート.....               | 175        |
| 5.19.1   | 構成データのエクスポート.....                   | 180        |
| 5.19.2   | 機密な情報を含む、または含まないオプション.....          | 184        |
| 5.19.3   | 構成データのインポート.....                    | 186        |
| 5.19.4   | 不足する依存関係の処理方法.....                  | 186        |
| <b>6</b> | <b>ジョブの実行のモニタリング</b> .....          | <b>188</b> |
| 6.1      | ジョブのステータス.....                      | 190        |
| 6.2      | ジョブの停止.....                         | 193        |
| 6.3      | メインログの確認.....                       | 194        |
| 6.4      | インスタンスログの確認.....                    | 196        |
| <b>7</b> | <b>FlowForce 式</b> .....            | <b>198</b> |
| 7.1      | 式の計算.....                           | 199        |
| 7.2      | 式言語のルール.....                        | 200        |
| 7.3      | 文字列フィールド内に式を埋め込む.....               | 203        |
| 7.4      | 式関数の呼び出し.....                       | 204        |
| 7.5      | FlowForce データ型.....                 | 205        |
| 7.6      | 演算子.....                            | 207        |
| <b>8</b> | <b>Altova 製品との統合</b> .....          | <b>208</b> |
| 8.1      | サーバー実行のためにファイルを準備する.....            | 209        |
| 8.2      | FlowForce Server にマッピングをデプロイする..... | 214        |
| 8.3      | マッピングと変換をジョブとして実行する方法.....          | 218        |
| 8.3.1    | マッピング関数内の資格情報.....                  | 221        |

|       |                             |     |
|-------|-----------------------------|-----|
| 8.3.2 | 動的な認証.....                  | 223 |
| 8.3.3 | リソース.....                   | 224 |
| 8.4   | .マッピング/変換結果へのアクセス.....      | 231 |
| 8.5   | .RaptorXML Server との統合..... | 233 |
| 8.6   | .ツールファイル.....               | 236 |
| 8.6.1 | 環境変数の設定.....                | 237 |

## 9 コマンドライン インターフェイス 239

|      |                             |     |
|------|-----------------------------|-----|
| 9.1  | .help.....                  | 243 |
| 9.2  | .assignlicense.....         | 244 |
| 9.3  | .compactdb.....             | 245 |
| 9.4  | .createdb.....              | 246 |
| 9.5  | .debug.....                 | 247 |
| 9.6  | .exportresourcestrings..... | 248 |
| 9.7  | .foreground.....            | 249 |
| 9.8  | .initdb.....                | 250 |
| 9.9  | .install.....               | 251 |
| 9.10 | .licenseserver.....         | 252 |
| 9.11 | .migratedb.....             | 253 |
| 9.12 | .repair.....                | 254 |
| 9.13 | .resetpassword.....         | 255 |
| 9.14 | .setdeflang (sdl).....      | 256 |
| 9.15 | .start.....                 | 257 |
| 9.16 | .uninstall.....             | 258 |
| 9.17 | .upgradedb.....             | 259 |
| 9.18 | .verifylicense.....         | 260 |

## 10 ビルトイン関数 261

|        |                          |     |
|--------|--------------------------|-----|
| 10.1   | ./system.....            | 262 |
| 10.1.1 | abort.....               | 262 |
| 10.1.2 | compute.....             | 263 |
| 10.1.3 | compute-string.....      | 265 |
| 10.2   | ./system/filesystem..... | 268 |

---

|           |                           |            |
|-----------|---------------------------|------------|
| 10.2.1    | copy.....                 | 268        |
| 10.2.2    | delete.....               | 269        |
| 10.2.3    | mkdir.....                | 269        |
| 10.2.4    | move.....                 | 270        |
| 10.2.5    | rmdir.....                | 271        |
| 10.3      | ./system/ftp.....         | 273        |
| 10.3.1    | delete.....               | 274        |
| 10.3.2    | delete-wildcard.....      | 276        |
| 10.3.3    | mkdir.....                | 278        |
| 10.3.4    | move.....                 | 281        |
| 10.3.5    | list.....                 | 283        |
| 10.3.6    | retrieve.....             | 286        |
| 10.3.7    | retrieve-wildcard.....    | 289        |
| 10.3.8    | rmdir.....                | 291        |
| 10.3.9    | store.....                | 294        |
| 10.3.10   | store-wildcard.....       | 298        |
| 10.4      | ./system/mail.....        | 301        |
| 10.4.1    | send.....                 | 301        |
| 10.4.2    | send-mime.....            | 302        |
| 10.5      | ./system/maintenance..... | 307        |
| 10.5.1    | archive-log.....          | 307        |
| 10.5.2    | cleanup-files.....        | 307        |
| 10.5.3    | truncate-log.....         | 308        |
| 10.6      | ./system/shell.....       | 309        |
| 10.6.1    | commandline.....          | 309        |
| <b>11</b> | <b>式関数</b>                | <b>311</b> |
| 11.1      | ステップ結果関数.....             | 312        |
| 11.1.1    | error-message.....        | 312        |
| 11.1.2    | exitcode.....             | 312        |
| 11.1.3    | failed-step.....          | 313        |
| 11.1.4    | results.....              | 314        |
| 11.1.5    | retry-count.....          | 316        |
| 11.1.6    | stdout.....               | 316        |

---

|        |                              |     |
|--------|------------------------------|-----|
| 11.1.7 | stderr.....                  | 317 |
| 11.2   | ストリーム関数.....                 | 318 |
| 11.2.1 | as-file.....                 | 318 |
| 11.2.2 | content.....                 | 319 |
| 11.2.3 | empty-stream.....            | 320 |
| 11.2.4 | stream-from-string.....      | 320 |
| 11.2.5 | stream-open.....             | 320 |
| 11.3   | ファイルシステム関数.....              | 322 |
| 11.3.1 | list-files.....              | 322 |
| 11.3.2 | list-directories.....        | 322 |
| 11.3.3 | read-lines.....              | 323 |
| 11.4   | ファイルパス関数.....                | 325 |
| 11.4.1 | extension.....               | 325 |
| 11.4.2 | filename.....                | 325 |
| 11.4.3 | filename-with-extension..... | 326 |
| 11.4.4 | join-paths.....              | 326 |
| 11.4.5 | parent-directory.....        | 327 |
| 11.5   | リスト関数.....                   | 329 |
| 11.5.1 | char.....                    | 329 |
| 11.5.2 | code.....                    | 330 |
| 11.5.3 | from-to.....                 | 331 |
| 11.5.4 | join.....                    | 331 |
| 11.5.5 | length.....                  | 333 |
| 11.5.6 | list.....                    | 333 |
| 11.5.7 | nth.....                     | 334 |
| 11.5.8 | slice.....                   | 334 |
| 11.6   | 文字列関数.....                   | 336 |
| 11.6.1 | concat.....                  | 336 |
| 11.6.2 | contains.....                | 336 |
| 11.6.3 | ends-with.....               | 337 |
| 11.6.4 | find-all.....                | 338 |
| 11.6.5 | number.....                  | 338 |
| 11.6.6 | split.....                   | 339 |
| 11.6.7 | starts-with.....             | 339 |
| 11.6.8 | string.....                  | 340 |

---

|         |   |     |
|---------|---|-----|
| 11.6.9  | string-join.....                        | 340 |
| 11.6.10 | string-length.....                      | 341 |
| 11.6.11 | substring.....                          | 341 |
| 11.6.12 | trim.....                               | 342 |
| 11.6.13 | trim-start.....                         | 343 |
| 11.6.14 | trim-end.....                           | 343 |
| 11.7    | ブール値関数.....                             | 344 |
| 11.7.1  | all.....                                | 344 |
| 11.7.2  | any.....                                | 344 |
| 11.7.3  | false.....                              | 345 |
| 11.7.4  | if.....                                 | 345 |
| 11.7.5  | not.....                                | 346 |
| 11.7.6  | true.....                               | 346 |
| 11.8    | ランタイム情報関数.....                          | 347 |
| 11.8.1  | instance-id.....                        | 347 |
| 11.8.2  | log.....                                | 347 |
| 11.8.3  | slot-number.....                        | 349 |
| 11.9    | MIME 式関数.....                           | 351 |
| 11.9.1  | add-mime-header.....                    | 351 |
| 11.9.2  | add-mime-headers.....                   | 351 |
| 11.9.3  | current-message-id.....                 | 352 |
| 11.9.4  | get-mime-content-disposition-param..... | 353 |
| 11.9.5  | get-mime-content-id.....                | 353 |
| 11.9.6  | get-mime-content-type-param.....        | 354 |
| 11.9.7  | get-mime-header.....                    | 355 |
| 11.9.8  | get-mime-headers.....                   | 355 |
| 11.9.9  | get-stream-filename.....                | 356 |
| 11.9.10 | is-file.....                            | 356 |
| 11.9.11 | is-mime-content-type.....               | 356 |
| 11.9.12 | mime-content-encode.....                | 357 |
| 11.9.13 | mime-flatten.....                       | 358 |
| 11.9.14 | mime-multipart.....                     | 358 |
| 11.9.15 | mime-multipart-related.....             | 359 |
| 11.9.16 | mime-multipart-from-list.....           | 361 |
| 11.9.17 | mime-parse.....                         | 361 |

|           |  |            |
|-----------|--|------------|
| 11.9.18   | mime-split-multipart.....              | 361        |
| 11.9.19   | new-message-id.....                    | 362        |
| 11.9.20   | reset-mime-headers.....                | 362        |
| 11.9.21   | set-mime-content-disposition.....      | 363        |
| 11.9.22   | set-mime-content-id.....               | 363        |
| 11.9.23   | set-mime-header.....                   | 364        |
| 11.9.24   | set-mime-headers.....                  | 365        |
| <b>12</b> | <b>ジョブの構成の例</b>                        | <b>366</b> |
| 12.1      | “Hello, World!” ジョブの作成.....            | 367        |
| 12.2      | パスが存在するか確認する.....                      | 370        |
| 12.3      | ファイルをコピーする.....                        | 374        |
| 12.4      | MapForce マッピングからジョブを作成する.....          | 379        |
| 12.5      | ジョブを他のジョブのステップとして使用する.....             | 386        |
| 12.6      | ディレクトリ ポーリング ジョブを作成する.....             | 389        |
| 12.7      | エラー処理をジョブに追加する.....                    | 395        |
| 12.8      | Web サービスとしてジョブを公開する.....               | 401        |
| 12.9      | JSON を FlowForce Web サービスへポストする方法..... | 406        |
| 12.10     | ジョブの結果をキャッシュする.....                    | 413        |
| 12.11     | StyleVision 変換からジョブを作成する.....          | 417        |
| 12.12     | RaptorXML を使用してドキュメントを検証する.....        | 425        |
| 12.13     | エラーログのある XML を検証する.....                | 427        |
| 12.14     | RaptorXML を使用して XSLT を実行する.....        | 431        |
| 12.15     | 複数の PDF を複数の XML から生成する.....           | 436        |
| <b>13</b> | <b>用語</b>                              | <b>448</b> |
| 13.1      | C.....                                 | 449        |
| 13.2      | E.....                                 | 450        |
| 13.3      | F.....                                 | 451        |
| 13.4      | I.....                                 | 452        |
| 13.5      | J.....                                 | 453        |
| 13.6      | M.....                                 | 454        |
| 13.7      | P.....                                 | 455        |

---

|       |        |     |
|-------|--------|-----|
| 13.8  | Q..... | 456 |
| 13.9  | R..... | 457 |
| 13.10 | S..... | 458 |
| 13.11 | T..... | 459 |
| 13.12 | U..... | 460 |
| 13.13 | W..... | 461 |

|               |            |
|---------------|------------|
| <b>インデックス</b> | <b>462</b> |
|---------------|------------|

## 1 はじめに

FlowForce Server は Windows、Linux、および macOS サーバー上、および Web インターフェイスを介したワークステーションでタスクを自動化するために使用されるクロスプラットフォームソフトウェアソリューションです。

FlowForce Server は他の Altova サーバー製品 (MapForce Server、StyleVision Server、および Raptor XML Server の両方の種類) と統合することができます。Web サービスとして作動するジョブを含む再帰的なオンデマンドのジョブとして機能を拡張することができます。例えば、MapForce Server と StyleVision Server との統合のために、MapForce マッピング、または StyleVision 変換を再帰的な FlowForce ジョブとして作動することができます。RaptorXML Server との統合でも同様に、XML または JSON ファイルを Web サービスとして公開されているオンデマンドのジョブとしてファイルを検証することができます。

FlowForce Server を使用すると、電子メールの送信、ローカルシステムまたはネットワーク上でのファイルの管理、ファイル転送プロトコル (FTP) を使用したファイルの管理、シェルスクリプトの実行など他の共通サーバータスクを作成し自動化することができます。



最終更新日: 2021 年 02 月 25 日

## 1.1 新規機能

### FlowForce Server 2021 R2

- [/system/ftp](#) ライブラリからの既存のFlowForce ビルトイン関数はFTPS (SSL を介したFTP) を介したサーバーへの接続のオプションをサポートします。
- ワイルドカードを使用してリモートのFTP サーバー上でファイルをアップロード、抽出、削除することを可能にする新規のFTP 関数を使用できるようになりました。具体的にはFTP サーバーをFTP またはFTPS を介して接続する場合、関数 `delete-wildcard`、`retrieve-wildcard`、と `store-wildcard` を [/system/ftp](#) ライブラリから使用することができます。
- [ファイルシステムリガー](#) を作成する際、最小のポーリング間隔を1秒に設定できるようになりました(以前は、最小の間隔は30秒でした)。
- [/system/mail/send](#) 関数は[メッセージボディ] フィールドのための改善された複数行のエディターを提供します。
- 統計詳細ページ: 色スキームラベル内の変更
- ファイルパス関数: [join-paths](#) は引数として提供されたパスを一つに結合する新規機能です。

### FlowForce Server 2021

- 更に素早くレコードをロードするために[ログビュー](#) ページは最適化され、新規のナビゲートとフィルターオプション、およびログの現在の状態を永久的リンクとして保存する機能が搭載されました。
- 一度にいくつかのログされたジョブをビューするための専用の新規の[ログインスタンス](#) ページを使用することができます。後に他の受取人に送信するためにこのページからログされた情報を.zip アーカイブにエクスポートすることができます。以前にエクスポートされたジョブインスタンスをログインスタンスページで検索し、後でダウンロードするためにビューすることができます。
- ログの特定のログの詳細が保管またはスキップされるかを決定できる新規の[ログ設定](#) を使用できるようになりました。ジョブの結果をベースとしてログ詳細のレベルを構成することもできます。ログのトレース情報を保管することが奨励されます。例えば、ジョブの失敗では、ログ内でトレース情報全体を保管することが奨励されますが、実行の成功の場合、最も基本的な項目のみを保管することが奨励されます。
- 特定のログ設定をアプリケーションレベルだけでなく、特定のFlowForce Server ジョブレベルで構成できるようになりました。[オブジェクトレベルでのログルール](#) を参照してください。

### FlowForce Server 2020 R2

- エラーが発生した場合、1つまたは複数のステップの実行を再試行することが可能です。[エラー時に再試行](#) を参照してください。
- ジョブはステップを延期して実行することができます。これはジョブがWeb サービスの呼び出しを介して呼び出される場合に適しています。[延期されたステップ](#) を参照してください。
- 新規の任意のホスト名 フィールドがセットアップページで使用できるようになりました。[ネットワーク設定の定義](#) を参照してください。SSL 構成が更に柔軟になり、Web サービスを直接ジョブ構成ページからテストできるようになりました。
- 新規のファイルまたはディレクトリが指定されたディレクトリに追加される際[ファイルシステムリガー](#) を実行するよう構成できます。このリガーはポーリングされたディレクトリが後に変更された場合トリガーを実行しない「既存の変更された日付」とは異なります。
- [セットアップページ](#) が簡素化されたかを評価するプロシージャです。

### FlowForce Server 2020

- MapForce とFlowForce Server 内の両方で資格情報を定義することができ、デザイン時にマッピングに埋め込む、またはFlowForce Server 内の実行ステップコンポーネントとして提供することができます。[マッピング関数内の資格情報](#) も参照してください。
- [資格情報オブジェクトを定義中](#) 使用の特定のドメインを制限することができます。「使用方法」は以下の1つまたは複数であることができます: ジョブ実行、FTP、HTTP。
- (「グローバルリソース」として既知の) MapForce 内で定義されているポータブルファイル、フォルダー、データベースリソースをFlowForce Server にデプロイし、マッピング関数により使用することができます。必要な場合、FlowForce 内で直接マッピング

ク関数により使用されるリソース(ファイル、フォルダー、データベースライセンス)を変更することができます。これはこの関数を使用するFlowForce ジョブのすべてに影響を与えます。FlowForce Server 内で直接リソースを作成または編集することができます。[リソース](#)を参照してください。

- ジョブ構成データを他のFlowForce Server インスタンスまたは.zip アーカイブをエクスポートする場合、機密なデータを任意で選択することもできます。[構成データのインポートとエクスポート](#)を参照してください。

## FlowForce Server 2019 R3

- FlowForce Server を使用して作成されたWeb サービスは、HTTP POST リクエストのボディをラマーターとして受け入れることができます。[Web サービスラマーター](#)を参照してください。例えば、[JSONをFlowForce Web サービスへポストする方法](#)を参照してください。
- [ログ](#)式関数と新規のログ設定により、FlowForce Server のログ機能が拡張されました。[ログ設定の変更](#)を参照してください。

## FlowForce Server 2019

- FlowForce Server をWindows Active Directory だけでなくLDAP (ライトウェイト デイレトリアクセスプロトコル)をサポートするDirectory Service プロバイダーにも統合することができます。[ディレトリサービスの設定を変更する](#)を参照してください。
- 新規のFlowForce Server 内蔵関数はFlowForce Server により送信される電子メールメッセージ本文をカスタマイズすることができる[send-mime](#) を使用することができます(例えば、StyleVision Server 変換を実行してHTML 本文、または、添付を準備する場合など)。新規関数によりSMTP ヘッダーをFlowForce で使用することのできる[MIME 式関数](#)を使用してカスタマイズすることができます。

## FlowForce Server 2018 R2

- (Windows 固有) Windows Active Directory 統合の拡張。Windows マシンがFlowForce Server がドメインの一部として作動する場合、「設定」ページ内で選択するためこれらのドメインが自動的に使用することができます(次を参照:[ディレトリサービス設定の変更](#))。

## FlowForce Server 2018

- FlowForce Server Advanced Edition が既存のFlowForce Server に加えて利用いただけます。FlowForce Server Advanced Edition のインストールパッケージは個別ライセンス供与されている次の追加製品とバンドルされています: MapForce Server Advanced Edition、RaptorXML Server、およびStyleVision Server。FlowForce Server のインストールパッケージは個別ライセンス供与されている次の追加製品とバンドルされています: MapForce Server、RaptorXML Server、StyleVision Server。
- 内部の更新と最適化。

## FlowForce Server 2017r3

- FlowForce Server とFlowForce Web Server は共に暗号化されていないプレーンな HTTP および(SSL-暗号化された) HTTPS接続を受け入れることができます。HTTPS 接続設定は、セットアップページ(次を参照:[ネットワーク設定の定義](#))、または、構成ファイルを使用して定義することができます(次を参照:[ファイル参照の構成](#))。ウィンドウでは、接続設定はインストール中に定義することができます。次を参照してください:[ネットワークの構成 \(Windows のみ\)](#)。
- Windows 上でのFlowForce Server インストール中、FlowForce Server とFlowForce Web Server に対応するWindows サービスを特定のオペレーティングシステムユーザーとして実行するよう構成することができます。サービスの開始(自動、手動)もセットアップすることができます。次を参照してください:[Windows サービス構成](#)。
- 新規のシステム管理コマンドがコマンドラインで使用できます: [compactdb](#)。

## FlowForce Server 2017

- システムシャットダウンでの改善された振る舞い

- 内部の更新と最適化。

### FlowForce Server 2016 R3

- ジョブのステップ、トリガー、およびパラメータを新規「複製」(複製) ボタンを使用することにより複製することができます。また、ステップ、トリガー、またはパラメータを削除した後「元に戻す」(元に戻す) ボタンを使用することができます。以下を参照: [入力パラメータの管理](#)、[ステップの管理](#)、および[トリガーの管理](#)。
- 他のジョブで呼び出されるジョブは、存在する場合、パラメータの説明ヒントとして表示します。パラメータの追加に関する詳しい情報は、以下を参照してください: [入力パラメータを追加する](#)。

### FlowForce Server 2016

- 既存のジョブが新規の名前で保存されることができ、新規のジョブのテンプレートとしての役割を果たします (以下を参照: [ジョブの複製](#))。
- ジョブの管理の強化: コンテナ内の特定のジョブのみだけでなく、サブコンテナを含むコンテナ全体を削除またはエクスポートすることが可能になりました (以下を参照: [ジョブのインポートとエクスポート](#))。
- コマンドラインインターフェイスから、ルート/パスワードのリセットを行うことができます (以下を参照: [ルート/パスワードの再設定](#))。
- パフォーマンスの向上: 集中したジョブの実行による Web インターフェイスのタイムアウトは大幅に削減されました。
- ビルトイン [/system/mail/send](#) 関数は、SMTP サーバーへの SSL/TLS 接続をサポートします (以下を参照: [電子メールパラメータの設定](#))。

### FlowForce Server 2015 R4

- Mac 向けの FlowForce Server インストールの ディスクイメージ (.dmg) ファイルをご利用いただけます (以下を参照: 上で実行されている [FlowForce Server のインストール](#))。
- 失敗したジョブの出力を処理する柔軟性: ジョブの実行が失敗すると、新規「エラー時は中断」パラメータは、ジョブが中断されるか、関数が false を返すかを決めます。このパラメータは `/system/filesystem`、`/system/ftp`、`/system/mail`、and `/system/shell` コンテナからの関数に適用されます。
- パフォーマンスの向上: ジョブ構成ページ内で大きなジョブが速くロードすることができます。

### FlowForce Server 2015 R3

- Web 管理インターフェイスから (ジョブまたは関数などの) オブジェクトが作成された、または変更された日時、および変更を行うユーザー名を確認することができます (以下を参照: [コンテナを理解する](#))。

### FlowForce Server 2015

- [Windows Active Directory](#) 統合
- ユーザーパスワードの [複雑性](#) の構成
- システム特権の [レポート](#)
- ジョブ [インポートとエクスポート](#)
- 実行中のジョブの [停止](#)

### FlowForce Server 2014 R2

- FlowForce [キャッシュ拡張](#)
- に名前を付ける能力 [ユーザーとロール](#)
- [コンテナオブジェクト](#) の名前変更機能
- 新規のステップ式 [関数](#)
- RaptorXML (キー値/パラメータペアのための [パラメータのリスト](#))

## FlowForce Server 2014

- 新規のサーバー [管理](#) 関数
- [ジョブの結果のキャンセル](#) 機能と、結果を他のジョブで使用する機能
- 拡張された [RaptorXML](#) へのサポート

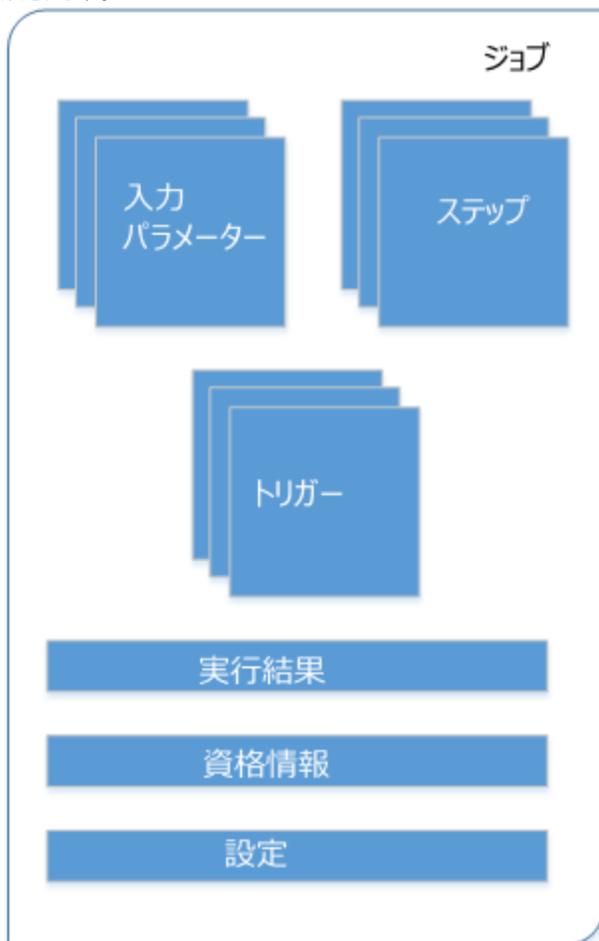
## FlowForce Server 2013 R2

- [RaptorXML Server との統合](#)
- 条件をベースにしたジョブステップジョブの実行を許可する [フローの管理](#) 実行ステップの繰り返し を無制限に行う機能
- [実行を繰り返す](#) ステップを無制限に使用する機能
- 1つのステップを次のジョブステップでを使用することを許可する [ステップ変数](#) の定義
- 電子メールの通知、FTP サーバーインタラクション、式の計算を許可する [ビルトインステップ](#) の拡張されたセット

## 1.2 基本概念

### ジョブ

ジョブとはFlowForce Server 内のコア概念です。この概念サーバー上で実行されるタスクまたはタスクのシーケンスを表します。ジョブは、電子メールの送信など1つのステップから構成されるジョブであることができます。しかしながら、複数のアクションを実行し結果(例えば、ファイル)を他のジョブの入力パラメータとして使用するジョブを作成することもできます。ジョブは入力パラメータ、ステップ、トリガーおよび他の設定から構成されます。



FlowForce ジョブの構成

### 入力パラメータ

FlowForce Server [ジョブ](#) のコンテキストで、入力パラメータは、どのジョブが特定の方法で実行されるかを表したジョブに与えられた任意の情報を意味します。ジョブの入力パラメータは、関数引数(プログラム言語の観点から類似しています。入力パラメータは多種の型(例えば、ファイルまたはディレクトリ、ファイル名、テキスト、番号、ブール値、その他)であることができます。一部のジョブの型に対しては、FlowForce Server は、入力パラメータを自動的に作成します。

### トリガー

ジョブを作成する際、トリガーとしてジョブをトリガーする条件(または基準)を指定する必要があります。FlowForce Server は、定義されたトリガーを監視し、トリガーの条件が満たされるとジョブを実行します。

## ステップ

ステップは、FlowForce Server ジョブが実際に行う内容を定義します(例えば、ファイルの削除、MapForce マッピングの実行、または電子メールの送信など)。最も簡単な形式としては、ステップは失敗または成功した結果を持つオペレーションで、実行されるために関数が必要とします(以下を参照: [実行ステップ](#))。しかしながら、ループして他のステップを条件付きで実行する方法を与えるステップも存在します(以下を参照: [選択ステップ](#)、[エラー成功の処理ステップ](#) および [For-Each ステップ](#))。同じジョブ内で必要な数だけのステップを作成し、実行される順序を設定することができます。

## 関数

[ジョブ](#) のコンテキストでは、関数は FlowForce Server により理解される、ターゲットファイルシステム上のオペレーションを実行するための命令です。関数は以下であることができます:

- ビルトイン FlowForce 関数 (以下を参照: [ビルトイン関数](#))
- StyleVision 変換
- MapForce マッピング
- ジョブの実行ステップ

関数の多くは入力パラメータを有します。必須入力パラメータは、ステップを成功裏に実行するため呼び出し側により与えられる必要があります。

## 実行の結果

FlowForce Server 内では、ステップの実行はステップの実行後に返される内容を定義します(例えば、ファイル、またはテキスト)。ジョブと作業する際、ステップの実行の結果を特定のデータ型(文字列またはブール値)になるように、または破棄されるように、明示的に宣言することができます。通常、結果を他のジョブで使用する場合、または、結果をキャッシュする場合、実行の結果を宣言する必要があります。

## 資格情報

資格情報オブジェクトは、認証に関する情報を保管します。これは FlowForce Server ジョブが作動する箇所でのオペレーティングシステム上のユーザーアカウントに関連するユーザー名とパスワードの組み合わせですが HTTP または FTP 資格情報のセットであることもできます。

## 設定

ジョブの作成時、以下の任意の設定を構成することができます:

- Web サービスとしてジョブを使用できるようにします(以下も参照してください: [ジョブを Web サービスとして表示する](#))。
- 同じジョブのため平行して実行されているインスタンスの数を制限します(以下も参照してください: [キュー設定の定義](#))。
- ジョブにより返された結果をキャンセルします(以下も参照してください: [ジョブの結果をキャンセルする](#))。

## 1.3 セキュリティ概念

FlowForce Server は、企業およびビジネスモデルのニーズと構造に従って構成されるロールをベースとしたユーザーのアクセスを管理するメカニズムを使用しています。例えば、ジョブと資格情報を特別なデータコンテナに整理し、閲覧および変更するには、アクセス権利を必要とするパッケージ化することができます。対応するアクセス権利を有するユーザーのみがコンテナ内部のデータにアクセスすることができます。

### コンテナ

名前が意味するように、コンテナはパッケージされたデータです。FlowForce Server では、コンテナはオペレーティングシステム上のフォルダと大まかに比較することができます。コンテナは以下を含むことができます：[ジョブ](#)、[資格情報](#)、[関数](#) と他のコンテナ。コンテナで [パーミッション](#) を設定することにより、コンテナ内のデータを閲覧およびデータにアクセスする権利を管理することができます。データをコンテナ別に整理し、対応する [パーミッション](#) を各コンテナのためにセットアップすることにより、セキュリティ対策となるでしょう。

### ユーザー

ユーザーとは FlowForce Server にログインジョブを構成、および MapForce および StyleVision 変換をデプロイし、FlowForce Server を管理する個人を指します。ユーザーが使用することのできるアクションは以下には影響されません

- 割り当てられた [パーミッション](#) および [特権](#)
- ユーザーがメンバーであるロールに割り当てられている [パーミッション](#) と [特権](#)

### ロール

ロールは、ビジネスニーズをベースとしたセキュリティの強化を手助けする [特権](#) のセットです。標準的なロールをベースとしたセキュリティには、最低2つのロールが含まれます：管理者と一般ユーザー。各ロールは、ロールに与えられた特権により定義されています。例えば、管理者は、自身のおよび他のユーザーのパスワードを変更することができますが、一般ユーザーは自身のパスワードのみを変更することができます。ユーザーロールを割り当て、必要に応じて、ユーザーのロールを取り消すことができます。

### 特権

特権は、FlowForce Server 内でユーザーが行うことのできる内容を定義します（例えば、自身のパスワード設定、ユーザーとロールの読み取り、ジョブの停止、など）。特権は、[パーミッション](#) は [コンテナ](#) へのユーザーアクセスを管理し、特権はグローバルに FlowForce Server 全体に対して効果がある点で [パーミッション](#) と異なります。次の簡単なルールは、特権と [パーミッション](#) の違いを理解する助けになります：特権はグローバルであり、[パーミッション](#) はローカルです。

[パーミッション](#) に類似して、特権は個々のユーザーと [ロール](#) に割り当てることができます。ですから、ユーザーが FlowForce Server にログインすると、有効な特権は以下により決定されます：

- 直接割り当てられた特権
- ユーザーがメンバーであるロールに割り当てられた特権

### パーミッション

[パーミッション](#) は、ユーザーの [コンテナ](#) へのアクセスを管理します。特権と同様、[パーミッション](#) はユーザーと [ロール](#) に与えられることができます。ですから、ユーザーがロールの一部である場合、ロールに与えられた [パーミッション](#) は、ユーザーにも自動的に適用されます。

デフォルトで、コンテナに設定された [パーミッション](#) は、親コンテナから継承されます。例えば、コンテナ A には、子コンテナ B があるとします。コンテナ A にアクセスする [パーミッション](#) を持つユーザーは、コンテナ B にもアクセスすることのできるデフォルトの [パーミッション](#) を持つことになります。しかしながら、管理者は、コンテナの階層の全てのレベルで、全てのユーザーおよびロールの [パーミッション](#) を再定義することができます。

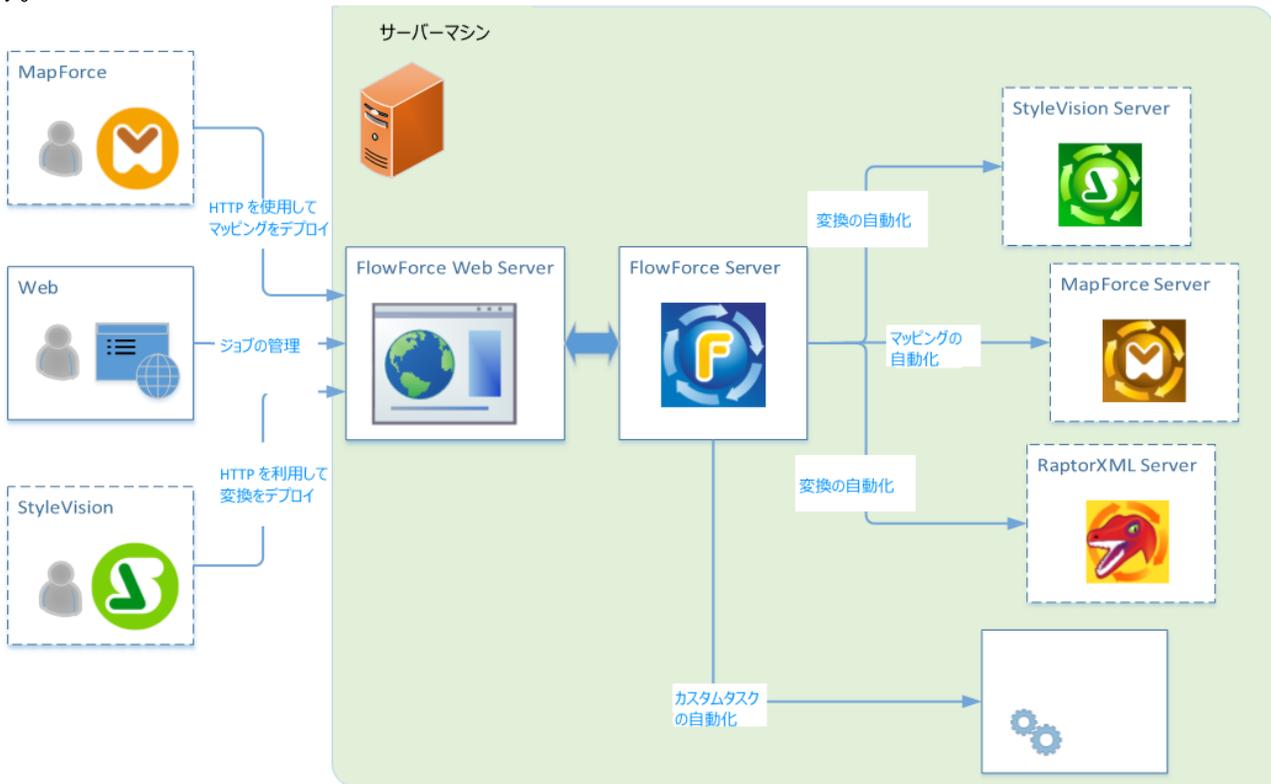
### パスワードポリシー

FlowForce Servers は、管理者が複雑なユーザーパスワードを管理するパスワードポリシーを使用します。パスワードポリシーとは、有効になるためにユーザーパスワードが必要とする最低条件です（例えば、最低 N 文字のなど）。

## 1.4 動作の説明

Altova Web サイト: [ワークフロー自動化ツール](#)

標準的な FlowForce Server インストールは、下の図に表されています。任意でライセンスが必要なコンポーネントは点線で囲まれています。



標準的な FlowForce Server インストール

図に示されているように、標準的な FlowForce Server インストールは、FlowForce Server として同じサーバーマシンにインストールされている複数のサーバー製品（または、技術的な用語では、サービス）により構成されています。FlowForce Server ソリューションは、以下の2つのサービスから構成されています: FlowForce Web Server と FlowForce Server。これらの2つのサービスは、異なるサービスとして作動し、個別に構成、開始、停止することができます。これらの2つのサービスの管理は、オペレーティングシステム (Linux、Windows、macOS) によりとなり、また、このドキュメントで更に詳しく説明されています。

FlowForce Web Server は、HTTP クライアントからのリクエストを受け入れ検証し、FlowForce Server にパスします。および FlowForce Server は FlowForce Server ソリューションのコアであり、視覚的なユーザーインターフェイスを使用することなく背景のサービスとして作動します。FlowForce Server は、継続的にトリガーの条件をチェックし、ジョブ実行を開始および監視し、詳細なログを作成します。更に、FlowForce Server は、Web サービスとして公開されているジョブのためのリクエストをリスンします (ローカルマシンとポートのクライアントからの HTTP リクエストを受け入れるように構成することができます、次を参照してください: [ネットワーク設定の定義](#))。

FlowForce Web Server、はその一方、ジョブを監視し、多種の FlowForce 設定を管理する Web 管理インターフェイスからのリクエストを処理します。FlowForce Web Server は、HTTP (または HTTPS) 接続を次のクライアントの型から受け入れます:

|           |  |
|-----------|--|
| Web ブラウザー | Web ブラウザーを使用して、FlowForce Server ジョブと他の設定を構成することができます (概要に関しては、以下を参照してください: <a href="#">Web 管理インターフェイス</a> )。 |
|-----------|--|

|   |   |
|---|---|
| MapForce Enterprise または Professional Edition    | MapForce は、Altova の視覚的なデータマッピングのためのデスクトップアプリケーションです。MapForce で作成されたマッピングを FlowForce Server にデプロイし、柔軟に構成することのできるジョブに変換することができます。例えば、マッピングジョブを毎日指定された時間、またはファイルが監視されるディレクトリに追加されると実行するように構成することができます。デプロイされた MapForce マッピングからジョブを実行するには、FlowForce Server は、役割がマッピングを実行し、結果出力ファイルを作成する MapForce Server を呼び出します。詳細に関しては、 <a href="#">MapForce</a> と <a href="#">MapForce Server</a> 製品ページを参照してください。 |
| StyleVision Enterprise または Professional Edition | StyleVision は、XML、SQL データベース および XBRL 入力をベースとしてレポートをデザインするデスクトップアプリケーションです。スタイルシートがテストされ、デバッグされると、FlowForce Server にデプロイすることができます。デプロイされたファイルは、サーバー上で変換ジョブのために使用することができます。デプロイされた StyleVision 変換から作成されたジョブを実行するには、FlowForce Server は変換と結果出力ファイルを作成する StyleVision Server を呼び出します。詳細に関しては、 <a href="#">StyleVision</a> と <a href="#">StyleVision Server</a> 製品ページを参照してください。                       |

各製品の詳細に関しては、Altova ドキュメントのページを参照してください (<https://www.altova.com/ja/documentation.html>)。

## RaptorXML Server

(略して RaptorXML と呼ばれる) Altova RaptorXML Server は、パラレルコンピューティング環境と最新の標準のために最適化された Altova の第3世代の高速 XML プロセッサです。高水準なクロスプラットフォームの機能性のためにデザインされたエンジンは、XML の高速な処理を行うために、現在のマルチコアコンピュータの利点を活用しています。RaptorXML には2つのエディションが存在します: (i) **RaptorXML Server** と (ii) **RaptorXML+XBRL Server**。RaptorXML+XBRL Server エディションは、XML に加え、XBRL (eXtensible Business Reporting Language) ドキュメントの検証と処理へのサポートを含んでいます。

RaptorXML Server が FlowForce Server と同じサーバーにインストールされている場合、関数はビルトイン FlowForce Server 関数として使用することができます。これは、XML ドキュメントの整形形式をチェックする、または検証する、または XSLT と XQuery ドキュメントを変換するジョブを作成することができることを意味します。詳細に関しては、[RaptorXML Server との統合](#)を参照してください。

## 1.5 Web 管理インターフェイス

FlowForce Server Web 管理インターフェイスは、サーバーを管理し、ジョブを構成するFlowForce Server のフロントエンドです。Web 管理インターフェイスに[アドレスとポートの構成](#)のWeb のブラウザからアクセスすることができます。

以下のセクションではWeb 管理インターフェイス内で使用することができるメニューアイテムについて説明しています。

- [ホーム](#)
- [構成](#)
- [ログ](#)
- [管理](#)
- [ヘルプ](#)

**メモ** Web 管理インターフェイスから使用することができるリソースとアクションへのアクセスは、ユーザーアクセス管理メカニズムにより管理されています。これは、割り当てられたパーミッションが許可する限り、構成データアクセスし変更することができます。同じように、対応する特権が与えられている場合、アクションを実行し（対応するメニューアイテムを閲覧することができます）。

### ホーム

、最近および実行中のジョブのリスト、アクティブトリガーとサービスのリストを表示します。詳細に関しては[ジョブの実行のモニタリング](#)を参照してください。

### 構成

現在定義されているFlowForce [コンテナ](#)、[ジョブ](#)、[資格情報](#)、および[関数](#)を表示します。オブジェクトのエントリと詳細を確認するには、対応する記録をクリックしてください。

デフォルトで以下のコンテナを使用することができます：

- `/public`
- `/system`
- `/RaptorXML` (ライセンスが与えられたRaptorXML Server を使用できる場合)

コンテナに関する詳細は、以下を参照してください [コンテナを理解する](#)。

関連するアクセスの権利があると前提して、構成ページから、コンテナ、ジョブ、資格情報および関数、を管理し、コンテナにパーミッションを設定することができます。

### ログ

サーバーログを表示します。表示されたログエントリはサーバーに関連した、また、ジョブに関連したメッセージを含みます。内容はアクセス権利により異なります。詳細に関しては[ジョブのログの確認](#)を参照してください。

### 管理

管理エリアから、サーバー構成とユーザーの管理に関連したアクションを実行することができます。管理エリアは以下のメニューアイテムから構成されます：

#### ユーザー

ユーザーの作成、削除、および、特権、ロール とパスワードの変更を含む、管理を行います。詳細に関しては、以下を参照してください [ユーザーアクセスの管理](#)。

#### レポート

アクセスコントロールのロールを作成、削除および保持することができます。詳細に関しては、以下を参照してください！ [ユーザーとロール](#)。

#### パスワードポリシー

パスワードの難易度のポリシーを作成、削除、および保持することを許可します。詳細に関しては、以下を参照してください！ [パスワードポリシー](#)。

#### レポート

現在割り当てられているユーザーと件に関するレポートの閲覧を許可します。詳細に関しては、以下を参照してください！ [特権レポートの閲覧](#)。

#### 設定

FlowForce Server とActive Directory およびLDAP 準拠したサーバーを統合し、デフォルトのタイムゾーン、電子メールサーバー設定を定義することを許可します。詳細に関しては、以下を参照してください！ [サーバーの構成](#)。

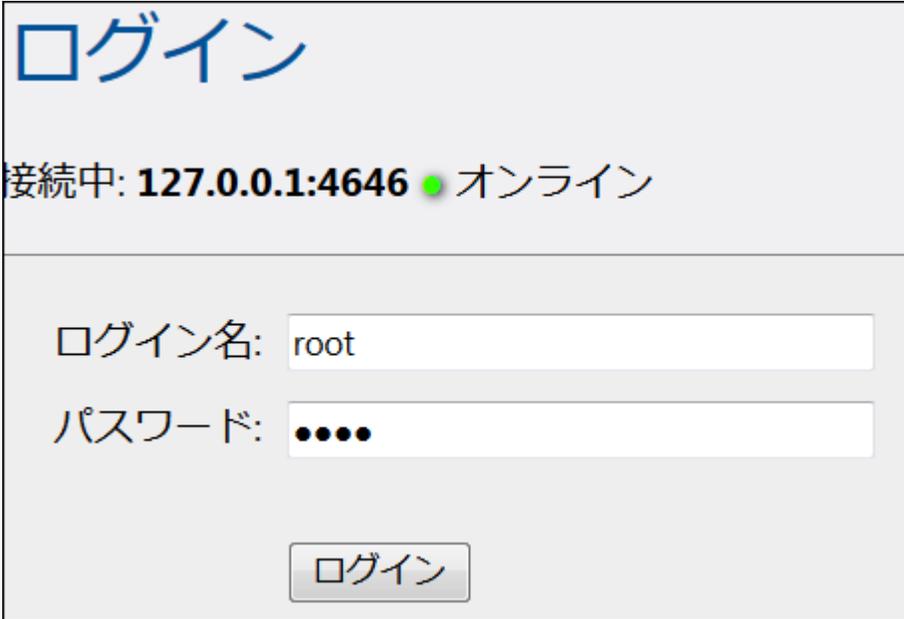
## ヘルプ

FlowForce Server ドキュメントを別のブラウザタブまたはウィンドウで開きます。

## 1.6 FlowForce Server へのログオン

(ジョブを作成し、ユーザーを追加するなどの) FlowForce Server を管理するには (例えば <http://localhost:8082> のような) 構成された HTTP(S) アドレスとポートで Web 管理インターフェイスにログオンする必要があります。この URL の構成に関しては、次を参照してください: [ネットワーク設定の定義](#) (「FlowForce Web Server」の下でグループ化された設定を検索してください)。

デフォルトでは、FlowForce Server を新規にインストールすることにより、ユーザー名 **root** とパスワード **root** を使用してログオンすることができます。



ログイン

接続中: 127.0.0.1:4646 ● オンライン

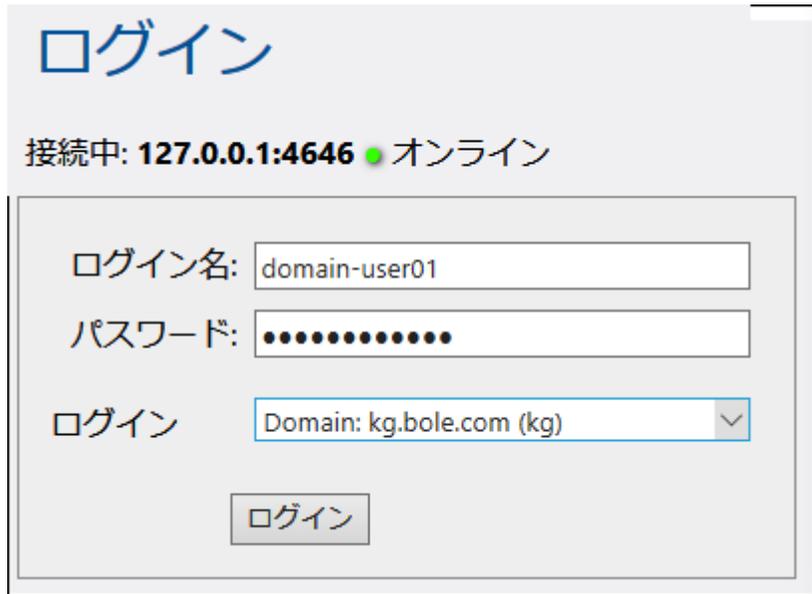
ログイン名: root

パスワード: ●●●●

ログイン

セキュリティ上の理由のため、FlowForce Server に最初ログイン後、デフォルトの root パスワードを変更してください。

(Active Directory などの) ディレクトリサービスプロバイダーが構成されている場合、ドメインユーザーは FlowForce にログオンすることができます。この場合、ログインページには、Windows ドメインを選択することのできるドロップダウンリストが含まれています。ディレクトリサービス認証の代わりに HTTP 認証を使用する場合、ログインドロップダウンリストから、ディレクトリを選択してください。



ログイン

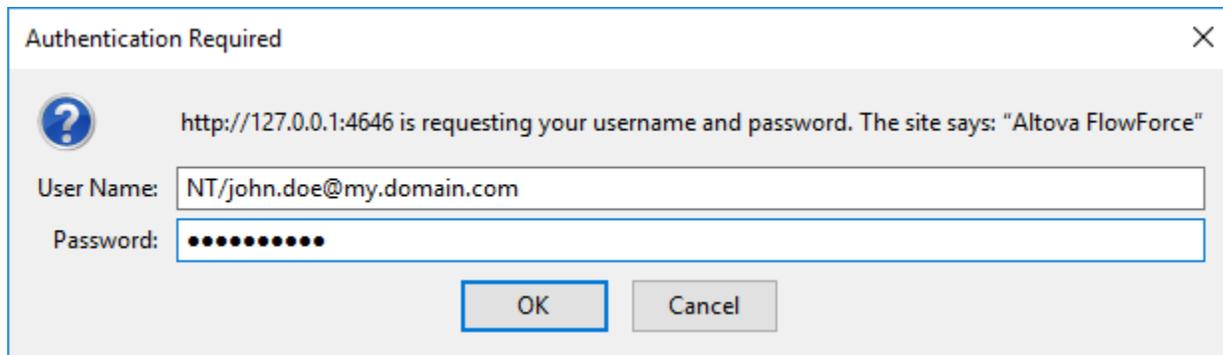
接続中: 127.0.0.1:4646 ● オンライン

ログイン名:

パスワード:

ログイン

FlowForce Server により公開される Web サービスにアクセスするクライアントは HTTP 認証への代替として Active Directory 認証を使用することができます(通常 `http://localhost:4646/service/SomeService` などの URL)。Active Directory 認証を可能にするには、ユーザー名は NT/ と共にプレフィックスされている必要があります、`NT/john.doe@my.domain.com` などのドメイン名を含む必要があります。次を参照してください: [Web サービスとしてジョブを公開する](#)。



Authentication Required

http://127.0.0.1:4646 is requesting your username and password. The site says: "Altova FlowForce"

User Name:

Password:

Windows ドメイン 認証の構成方法の詳細に関しては、次を参照してください: [ディレクトリサービス設定の変更](#)。

## 2 インストール

### 2.1 システム必要条件

|                |   |
|----------------|---|
| Windows        | プラットフォーム更新済みのWindows 7 SP1、Windows 8、Windows 10   |
| Windows Server | プラットフォーム更新済みのWindows Server 2008 R2 SP1 まで以降  |
| Linux          | <ul style="list-style-type: none"><li>• CentOS 7 まで以降</li><li>• RedHat 7 まで以降</li><li>• Debian 8 まで以降</li><li>• Ubuntu 16.04 LTS まで以降</li></ul> |
| macOS          | macOS 10.13 まで以降  |

Web 管理インターフェイスはGoogle Chrome、Mozilla Firefox、およびMicrosoft Edge の最新バージョンよりアクセス可能です。

## 2.2 プロセッサコアとライセンス

Altova サーバー製品へのライセンス供与は(ロジカルなコアの数量ではなく)製品マシン上で使用することのできる物理的なプロセッサのコアの数量をベースとしています。例えば、デュアルコアプロセッサ1コア、クワッドコアプロセッサ4コア、ヘキサコア6コアなどの2つのコアが存在します。製品にライセンス供与されるコア数は、物理マシンは仮想マシン上に関わらず、サーバーマシン上で使用することのできるコアの数量と同等、またはそれ以上である必要があります。例えば、サーバーが8コア持つ場合、8コアライセンスを購入する必要があります。コアの数量を満たすためにライセンスを組み合わせることもできます。ですから、2つの4コアライセンスを使用して、8コアライセンスの代わりに8コアを達成することもできます。

処理する容量は少ないが、CPU コアの大きなコンピューターサーバーを使用している場合、少ない数量のコア割り当てられている仮想マシンを作成し、その数量のライセンスを購入することもできます。このようなデプロイの場合、もちろん、コンピューターのすべてのコアを使用する場合に比べこのようなデプロイの処理スピードは遅いです。

**メモ** 各 Altova サーバー製品ライセンスは、使用されていないライセンス供与能力が存在する場合でも、Altova サーバー製品がインストールされているマシンで、一度1つのクライアントマシンのためのみ使用することができます。例えば、10 コアライセンスが6 CPU コアを持つマシンのためで使用されている場合、残りのライセンス許与能力の4 コアは、他のクライアントマシンのためと同時に使用することはできません。

### 単一スレッド実行

Altova サーバー製品が単一スレッドの実行を許可すると、単一スレッドの実行を使用できるようになります。この場合、Altova サーバー製品ライセンスはライセンスプール内でコアのみ使用することができます。複数のコアを持つマシンはこのコアライセンスを割り当てることができます。このような場合、その製品を作動するマシンは単一コアで作動します。(マルチコアで使用可能な複数スレッドを使用できないため処理はこのために遅くなります。製品はそのマシン上で単一スレッドモードで実行されます。

マルチコアマシンに単一コアライセンスを割り当てるには、その製品のために LicenseServer 内で「単一スレッド実行に制限する」チェックボックスを選択してください。

### 必要なコア数の見積もり

サーバーが扱うことのできるデータの量と処理回数を影響する多くの外部要素が存在します。(例えば、ハードウェア、CPU 上の現在のロード、サーバー上で作動する他のアプリケーションのメモリの割り当てなどが挙げられます)。パフォーマンスを正確に測定するために、データの量、条件、およびビジネスの環境に近い状態でアプリケーションをテストしてください。

## 2.3 重要なパス

FlowForce Server のインストール後以下のファイルに重要な情報が保管されます:

- インストールディレクトリ(INSTALLDIR)
- アプリケーションデータディレクトリ(DATADIR)

### FlowForce Server インストールディレクトリ(INSTALLDIR)

|         |  |
|---------|--|
| Linux   | /opt/Altova/FlowForceServer2021/   |
| macOS   | /usr/local/Altova/FlowForceServer2021/   |
| Windows | C:\Program Files\Altova\FlowForceServer2021\<br>C:\Program Files (x86)\Altova\FlowForceServer2021\ |

### FlowForce Server アプリケーションデータディレクトリ(DATADIR)

|         |  |
|---------|--|
| Linux   | /var/opt/Altova/FlowForceServer2021/data       |
| macOS   | /var/Altova/FlowForceServer2021/data           |
| Windows | C:\ProgramData\Altova\FlowForceServer2021\data |

## 2.4 Linux

### 2.4.1 FlowForce Server のインストール

#### システムの必要条件

##### ▼ Linux

- CentOS 7 または以降
- RedHat 7 または以降
- Debian 8 または以降
- Ubuntu 16.04 LTS または以降

次のライブラリはアプリケーションをインストール実行するために必要とされるライブラリです。下のパッケージが使用中 Linux のマシンで使用できない場合、yum (または 適用できる場合、apt-get を) コマンドを実行してインストールしてください。

| サーバー             | CentOS, RedHat    | Debian                     | Ubuntu                     |
|------------------|-------------------|----------------------------|----------------------------|
| FlowForce Server | libidn, krb5-libs | libidn11, libgssapi-krb5-2 | libidn11, libgssapi-krb5-2 |
| LicenseServer    | krb5-libs         | libgssapi-krb5-2           | libgssapi-krb5-2           |

#### 必要条件

- root ユーザーまたは sudo 特権を持つユーザーとしてインストールを行ってください。
- FlowForce Server の前のバージョンが新しいバージョンがインストールされる前にアンインストールされる必要があります。
- FlowForce Server を他のAltova サーバー製品とインストールする場合は、FlowForce Server を最初にインストールすることが奨励されます。

#### FlowForce Server のインストール

1. <https://www.altova.com/download#server> からローカルディレクトリにインストールパッケージをダウンロードします。
2. インストールパッケージをダウンロードするディレクトリを変更します。例:

```
cd /home/User/Downloads
```

3. FlowForce Server パッケージをインストールします。

```
[Debian, Ubuntu] sudo dpkg --install flowforceserver-2021-debian.deb
[CentOS, RedHat] sudo rpm -ivh flowforceserver-2021-1.x86_64.rpm
```

**メモ** 現在のリリース または サービス パッケージバージョンに一致するために、上記のパッケージの名前を調整する必要がある場合があります。

#### FlowForce Server のライセンス供与

1. Altova LicenseServer の最新バージョンがインストールされており、ローカルマシン、または ネットワーク上で実行されていること

- を確認してください (<https://www.altova.com/ja/licenseserver>)。
- FlowForce Server を Altova LicenseServer に登録します。このためには、作動中のコンピュータのホスト名、またはアドレスを知る必要があります。例えば、LicenseServer が 127.0.0.1 上で作動する場合、登録は以下のように行うことができます:

```
sudo ./flowforceserver licenseserver 127.0.0.1
```

**メモ** 上記のコマンドは、プログラムインストールディレクトリから実行可能ファイルを呼び出していることが前提です。

- LicenseServer 管理者インターフェイスログイン、FlowForce Server が実行されるマシンにライセンスを割り当てます。

ライセンスの詳細に関しては、Altova LicenseServer ドキュメントを参照してください (<https://www.altova.com/ja/documentation>)。

### インストール後のタスク

- セットアップモードを入力して [セットアップページ](#) を参照してください。
- Web 管理インターフェイスのネットワークアドレスとポートを構成します。FlowForce Server を Altova LicenseServer に登録します。 [ネットワーク設定の定義](#) を参照してください。
- Web 管理インターフェイスログイン、(前のステップ内でアドレスとポートを変更しない限り、デフォルトでは `http://localhost:8082`、です) デフォルトの FlowForce Server `root` / パスワードを変更します。デフォルトのログイン名とパスワードはそれぞれ、`root` と `root` です。
- FlowForce Server と共に他の Altova 製品がインストールされている場合、(例えば、MapForce Server、StyleVision Server、RaptorXML Server) 環境変数を設定する場合、 [環境編集の設定](#) を参照してください。

## 2.4.2 FlowForce Server 2021 への移行

必要条件:

- FlowForce Server の前のバージョンを [アンインストール](#) します。アンインストールはアプリケーション データディレクトリを削除することを意味しません。 [重要な](#) [タスク](#) も参照してください。アプリケーションデータディレクトリへのパスは FlowForce Server のバージョンにより異なります (例、`/var/opt/FlowForceServer2017`)。
- FlowForce Server 2021 を [インストール](#) します。これはデフォルトの構成データを持つ新規のアプリケーションデータディレクトリを作成します (例、`/var/opt/FlowForceServer2021`)。

FlowForce Server 2021 へのデータの移動の方法 :

- 実行中の場合 FlowForce Web Server サービスを停止します。

```
sudo systemctl stop flowforcewebserver
```

- 実行中の場合 FlowForce Server サービスを停止します。上記と同じコマンドを使用します。 `flowforcewebserver` を `flowforceserver` と置き換えます。
- インストール中に作成された新規のデータディレクトリを、削除または、別の名前を与えます。

```
sudo rm -rf /var/opt/Altova/FlowForceServer2021/data
```

- FlowForce Server のコマンドラインインターフェイス内で使用可能な [migratedb](#) コマンドを実行して EXISTING データを移行します。例:

```
sudo /opt/Altova/FlowForceServer2021/bin/flowforceserver migratedb
--olddatadir=/var/opt/Altova/FlowForceServer2017/data
--datadir=/var/opt/Altova/FlowForceServer2021/data
```

5. FlowForce Web Server サービスを開始します。

```
sudo systemctl start flowforcewebserver
```

6. FlowForce Server サービスを開始します。上記と同じコマンドを使用します。flowforcewebserver を flowforceserver と置き換えます。

### 2.4.3 現在インストールされている製品の確認

Altova サーバー製品がインストールされているか確認するには以下のコマンドを実行します:

```
[Debian, Ubuntu] dpkg --get-selections | grep Altova
[CentOS, RedHat] rpm -qa | grep server
```

### 2.4.4 FlowForce Server のアンインストール

FlowForce Server をアンインストールするには、以下のコマンドを実行します:

```
[Debian, Ubuntu] sudo dpkg --remove flowforceserver
[CentOS, RedHat] sudo rpm -e flowforceserver
```

### 2.4.5 インストール ディレクトリ

Linux 上での FlowForce Server インストールディレクトリの場所は、以下の通りです:

```
/opt/Altova/FlowForceServer2021/bin
```

## 2.5 macOS

### 2.5.1 FlowForce Server のインストール

#### システム必要条件

##### ▼ macOS

macOS 10.13 以降

#### 必要条件

- Altova LicenseServer がインストールされており、ローカルマシン、または、ネットワーク上で実行されていることを確認してください。
- 管理 (ルート) 特権を持つユーザーとしてインストールします。
- FlowForce Server を他の Altova サーバー製品とインストールする場合は、FlowForce Server を最初にインストールすることが奨励されます。
- macOS マシンは、IP アドレスに対して名前が解決されるように構成される必要があります (すなわち、コマンド `ping <hostname>` を使用してターミナルからホスト名を ping すること成功する必要があります)。

#### FlowForce Server のインストール

1. Altova Web ページから <http://www.altova.com/ja/download.html> から、ディスクイメージファイル (.dmg) をダウンロードし、開けぬをクリックします。これによりインストーラーがコンピュータ上で新規ドライブとして表示されます。
2. パッケージ (.pkg) ファイルをダブルクリックして、インストールウィザードに従いインストールします。これらのステップは、説明を必要とせず、続行する前にライセンス使用許諾書に同意するステップ1つが含まれています。ウィザードを完了すると、FlowForce Server アイコンがアプリケーション内に表示されます。パッケージは以下のディレクトリに `/usr/local/Altova/FlowForceServer2021/`。
3. Altova License Server がインストールされていない場合、または、最新バージョンにアップグレードする場合、対応するパッケージ (.pkg) ファイルをダブルクリックし、画面の説明に従ってください。Altova LicenseServer は、FlowForce Server を含む Altova サーバーアプリケーションを実行するために必須です。
4. アプリケーション内で FlowForce Server アイコンをダブルクリックします。これにより FlowForce Server 構成ページが開かれます。

Mac マシンの名前が IP アドレスを解決できない場合、ブラウザは次のメッセージを表示するページを開きます:

「`http://<hostname>:<port>` で FlowForce Web を使用できません。再起動の後、このページを再ロードしてください。」このメッセージが表示されると以下を行ってください (次を参照してください! [必要条件](#)):

- a. このメッセージ内のリンクをクリックする。
  - b. ブラウザーのアドレスバー内の `<hostname>` を使用中の Mac の IP アドレスと置き換えてください。
5. 構成ページから、Altova LicenseServer を使用して、FlowForce Server を登録します。
  6. Web 管理インターフェイスのネットワークアドレスとポートを構成します。FlowForce Server を Altova LicenseServer に登録します。 [ネットワーク設定の定義](#) を参照してください。
  7. Web 管理インターフェイスを起動し、(前のステップ内でアドレスとポートを変更しない限り、デフォルトでは `http://localhost:8082`、です) デフォルトの FlowForce Server `root` パスワードを変更します。デフォルトのログイン名とパスワードはそれぞれ、`root` と `root` です。

**メモ** FlowForce Server をインストールする前に、Altova サーバー製品がインストールされている場合、FlowForce Server `/tools` ディレクトリへ、対応する製品の `/etc` ディレクトリから .tool ファイルをコピーしてください。 .tool ファイルは、各

Altova サーバー製品の実行可能ファイルへのパスを含みます。ツールファイルをコピーする場合は、ターミナル内で `cp` コマンドを実行してください。例えば、MapForce Server インストールパスから、ファイルをコピーする場合、以下を実行します:

```
cp /usr/local/Altova/MapForceServer2021/etc/*.tool /usr/local/Altova/FlowForceServer2021/tools
```

### FlowForce Server のライセンス供与

1. Altova LicenseServer の最新バージョンがインストールされており、ローカルマシン、またはネットワーク上で実行されていることを確認してください (<https://www.altova.com/ja/licenseserver>)。
2. FlowForce Server を Altova LicenseServer に登録します。このためには、が作動中のコンピュータのホスト名、またはアドレスを知る必要があります。例えば、LicenseServer が 127.0.0.1 上で作動する場合、登録は以下のように行うことができます:

```
sudo flowforceserver licenseserver 127.0.0.1
```

3. LicenseServer 管理者インターフェイスログイン、FlowForce Server が実行されるマシンにライセンスを割り当てます。

詳細に関しては、Altova LicenseServer ドキュメントを参照してください (<https://www.altova.com/ja/documentation>)。

## 2.5.2 FlowForce Server 2021 への移行

### 必要条件

- FlowForce Server 2021 がインストールされている必要があります (以下を参照: [FlowForce Server のインストール](#))。
- データの移行を管理者 (ルート) の特権を持つユーザーとして行います。

### FlowForce Server 2021 にデータを移行する

1. FlowForce Server サービスを停止します。

```
sudo launchctl unload /Library/LaunchDaemons/com.altova.FlowForceServer.plist
```

2. FlowForce Web Server サービスを停止します。

```
sudo launchctl unload /Library/LaunchDaemons/com.altova.FlowForceWebServer.plist
```

3. インストール中に作成されたデータディレクトリの名前を変更するか削除します。これにより、手動のアップグレードを行う前のインストールの後に作成された新しいバージョン内のオブジェクトが削除されます:

```
sudo rm -rf /var/Altova/FlowForceServer2021/data
```

4. **migratedb** コマンドを実行します:

```
sudo /usr/local/Altova/FlowForceServer2021/bin/FlowForceServer migratedb  
--olddatadir=/var/Altova/FlowForceServer2017/data --  
datadir=/var/Altova/FlowForceServer2021/data
```

5. FlowForce Server サービスを開始します。

```
sudo launchctl load /Library/LaunchDaemons/com.altova.FlowForceServer.plist
```

6. FlowForce Web Server サービスを開始します。

```
sudo launchctl load /Library/LaunchDaemons/com.altova.FlowForceWebServer.plist
```

## 2.5.3 FlowForce Server のアンインストール

### FlowForce Server のアンインストール

1. FlowForce Server サービスを停止します。

```
sudo launchctl unload /Library/LaunchDaemons/com.altova.FlowForceServer.plist
```

2. FlowForce Web Server サービスを停止します。

```
sudo launchctl unload /Library/LaunchDaemons/com.altova.FlowForceWebServer.plist
```

サービスが停止下かを確認するために、アクティビティモニターターミナルを開き、サービスがリストに存在しないことを確認します。アクティビティモニターをダブルクリックすると、「マイプロセス」以外の全てのプロセスが表示されます（「表示」メニューの「全てのプロセス」をクリックします）。

3. 「アプリケーション」内のFlowForce Server アイコンを右クリックし、「ゴミ箱に移動する」を選択します。FlowForce Server 構成データは別途に保管されており、この段階では削除されないことにご注意してください（次のステップを参照してください）。
4. 任意で、FlowForce Server 構成データを削除する場合、以下のコマンドを実行します:

```
sudo rm -rf /usr/local/Altova/FlowForceServer2021/
```

**メモ** 上記のステップは /var/Altova/FlowForceServer2021 から FlowForce Server 構成データを削除しません。FlowForce Server 構成データが削除されると、後で復元できないことにご注意してください。

## 2.5.4 インストールディレクトリ

macOS 上で FlowForce Server インストールディレクトリの場所は、以下の通りです:

```
/usr/local/Altova/FlowForceServer2021
```

## 2.6 Windows

### 2.6.1 FlowForce Server のインストール

#### システム必要条件

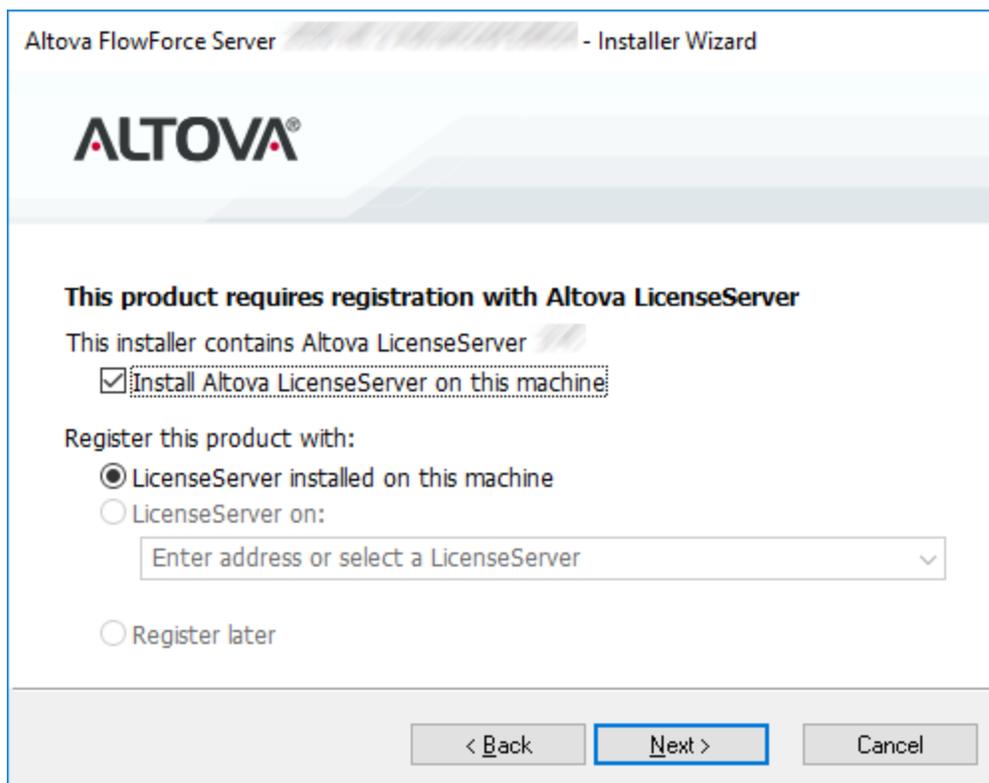
- プラットフォーム更新済みのWindows 7 SP1、Windows 8、Windows 10
- プラットフォーム更新済みのWindows Server 2008 R2 SP1 以降

#### 必要条件

- 管理特権を持つユーザーとしてインストールを行います。
- バージョン 20217.1 以降からFlowForce Server の32 ビットバージョンは64 ビットバージョンにインストールすることはできません。または64 ビットバージョンを32 ビットバージョンにインストールすることはできません。(i) 新しいバージョンをインストールする前に古いバージョンを削除する、または(ii) 古いインストールと同じビットバージョンを新規のバージョンに更新する必要があります。

#### FlowForce Server のインストール

1. Altova Web サイト(<https://www.altova.com/download#server>) からインストールファイルをダウンロードして実行します。
2. 任意で、ウィザードの左下のボックスからインストール言語を選択し、「次へ」をクリックします。ここで選択される言語は、Web ブラウザ内のFlowForce Server ユーザーインターフェイスの言語も決定します。必要であれば、後に [コマンドライン](#) から言語を変更することができます。
3. 以下の内の1つを行います:
  - a. Altova LicenseServer がまだインストールされていない場合、デフォルトの設定を保持してください。ウィザードが最新バージョンのAltova LicenseServer をウィザードを起動しているコンピュータにインストールします。
  - b. Altova LicenseServer がまだインストールされていない、他のコンピュータにAltova LicenseServer をインストールする場合、「使用中のコンピュータにAltova LicenseServer をインストールする」チェックボックスをクリアして、「後で登録」を選択します。この場合、Altova LicenseServer のインストールとFlowForce Server の登録を個別に行う必要があります。
  - c. Altova LicenseServer が既に使用中のコンピュータにインストールされているが、インストールウィザードにより示されているバージョンが低い場合は、デフォルトの設定を保持してください。この場合、インストールウィザードが自動的にLicenseServer バージョンをダイアログボックスに表示されたバージョンにアップグレードします。既存の登録およびライセンス情報はアップグレードの後も保存されます。
  - d. Altova LicenseServer が既に使用中のコンピュータ、またはネットワークインストールされており、ウィザードに示されるバージョンと同じ場合は、以下を行ってください:
    - i. 「使用中のコンピュータにAltova LicenseServer をインストールする」チェックボックスをクリックしてクリアします。
    - ii. 「この製品を登録する」からFlowForce Server を登録するAltova LicenseServer インスタンスを選択、または「後で登録」を選択してください。LicenseServer への関連を無視し、FlowForce Server のインストールを継続する場合、「後で登録」を常に選択することができます。



4. 「次へ」をクリックします。
5. 任意で、インストールする追加 Altova サーバー製品を選択してください。
6. FlowForce Server と FlowForce Web Server に適用することのできるネットワークの設定と Windows サービスの構成を任意で定義することができます。詳細に関しては [ネットワーク構成](#) と [Windows サービスの構成](#) を参照してください。FlowForce のインストール時と同じダイアログボックスから、それぞれのタブをクリックすることにより、他の Altova 製品のためオプションでネットワークとサービス設定を定義することもできます。これらの設定をインストール後に行う場合、「次へ」をクリックします。



7. 任意で、FlowForce クラスター構成オプションをセットアップします。このインストールステップは、FlowForce Server Advanced Edition のためこのみ適用することができます。を参照してください。FlowForce Server をクラスターで動作する必要がない場合、「スタンドアロン」を選択します。
8. ウィザードに従ってインストールを完了してください。インストールが完了すると、システム通知エリアに、Altova Service Controller アイコン(  ) が追加されます。

## Windows サーバーコア上でのインストール

Windows Server Core は多数の GUI 機能を使用しない最低限の Windows インストールです。Windows Server Core マシン上に FlowForce Server を以下のようにインストールすることができます。

1. Altova ダウンロードセンター (<https://www.altova.com/ja/download>) から FlowForce Server 実行可能ファイルをダウンロードして実行します。使用中のサーバープラットフォーム(32ビットまたは64ビット)にマッチする実行可能ファイルが選択されていることを確認して下さい。
2. グラフィカルなユーザーインターフェイスが搭載されている Windows コンピューターでコマンドプロンプトウィンドウを管理者として開きインストール実行可能ファイル上の以下のコマンドを実行します:

```
FlowForceServer<version>.exe /u
```

3. 次のステップのために必要な .msi ファイルが解凍されます。Windows Server コアで動作するコンピューターに .msi ファイルをコピーします。
4. FlowForce Server の前のバージョンを更新する場合 FlowForce Server を次のステップを実行する前にシャットダウンします。
5. .msi ファイルを以下のコマンドを使用して FlowForce Server をインストールします:

```
msiexec /i FlowForceServer.msi
```

インストールの戻り値をテストするには、以下に類似するスクリプトを実行する必要がある場合があります。

```
start /wait msiexec /i FlowForceServer.msi /q
echo %errorlevel%
```

インストールオペレーションの戻りコードは %errorlevel% 環境変数内で使用することができます。戻りコード 0 は成功を示しています。

戻りコードとインストールプロセスのログを持つサイレントインストールのため以下を実行します。

```
start /wait msiexec /i FlowForceServer.msi /q /L*v! <pathToInstallLogFile>
```

メモ .msi ファイルを安全な場所に保管してください。後にアンインストール、修復、または変更する場合これらが必要になります。

インストールを変更するには以下を実行します。

```
msiexec /m FlowForceServer.msi
```

インストールを修復するには以下を実行します。

```
msiexec /r FlowForceServer.msi
```

FlowForce Server をアンインストールするには以下を実行します。

```
msiexec /x FlowForceServer.msi
```

または、以下のコマンドは FlowForce Server を確認せずアンインストールしてログファイルに結果の詳細をレポートします。

```
start /wait msiexec /x FlowForceServer.msi /q /L*v! <pathToUninstallLogFile>
```

## FlowForce Server のライセンス供与

1. Altova LicenseServer の最新バージョンがインストールされており、ローカルマシン、またはネットワーク上で実行されていることを確認してください (<https://www.altova.com/ja/licenseserver>)。
2. インストール中に行っていない場合、FlowForce Server を Altova LicenseServer に登録します。このためには、作動中のコンピューターの名前、またはアドレスを知る必要があります。例えば LicenseServer が 127.0.0.1 上で作動する場合、登録は以下のように行うことができます。

```
FlowForceServer.exe licenseserver 127.0.0.1
```

3. LicenseServer 管理者インターフェイスのログイン、FlowForce Server が実行されるマシンにライセンスを割り当てます。

詳細に関しては、Altova LicenseServer ドキュメントを参照してください (<https://www.altova.com/ja/documentation>)。

## インストール後のタスク

1. インストール中に行っていない場合、[ネットワーク設定](#)を構成します。

2. Web 管理インターフェイスログイン、デフォルトのFlowForce Server root / パスワードを変更します (アドレスとポートを変更していない場合、デフォルトでは `http://localhost:8082` です)。デフォルトのログイン名とパスワードはそれぞれ `root` と `root` です。

## 2.6.2 Windows サービスの構成

Windows オペレーティングシステム上にFlowForce Server をインストール中、以下の設定を構成することができます。

- FlowForce Server をWindows サービスとして開始する方法 自動、または、オンデマンド、遅延された自動、または、無効化。
- FlowForce Server によりオペレーティングシステムユーザーアカウントをWindows サービスとして使用する方法 ローカルシステム、ローカルサービス、ネットワークサービス、または他のユーザー。
- FlowForce Server が他のユーザーとして実行されるように構成されていると、Windows Services 管理コンソールと同様にこのユーザーのユーザー名とパスワードを設定することができます。

If you select 他のユーザーを選択するとユーザーアカウント(またはメンバーであるグループ)はC:\ProgramData\Altovaへの読み取り書き込みアクセスを有している必要があります。それ以外の場合、インストールまたは起動の失敗を引き起こす可能性があります。

上にリストされる設定は、FlowForce Web Server のために構成することもできます。

設定は、Windows Services 管理コンソールからサービスプロパティを変更することと同じ意味があります。

インストール中に上記の設定を定義するには、インストールダイアログボックスの関連するタブをクリックします (FlowForce Server または FlowForce Web Server)。そして、「サービスの構成」の横の「変更」をクリックします。

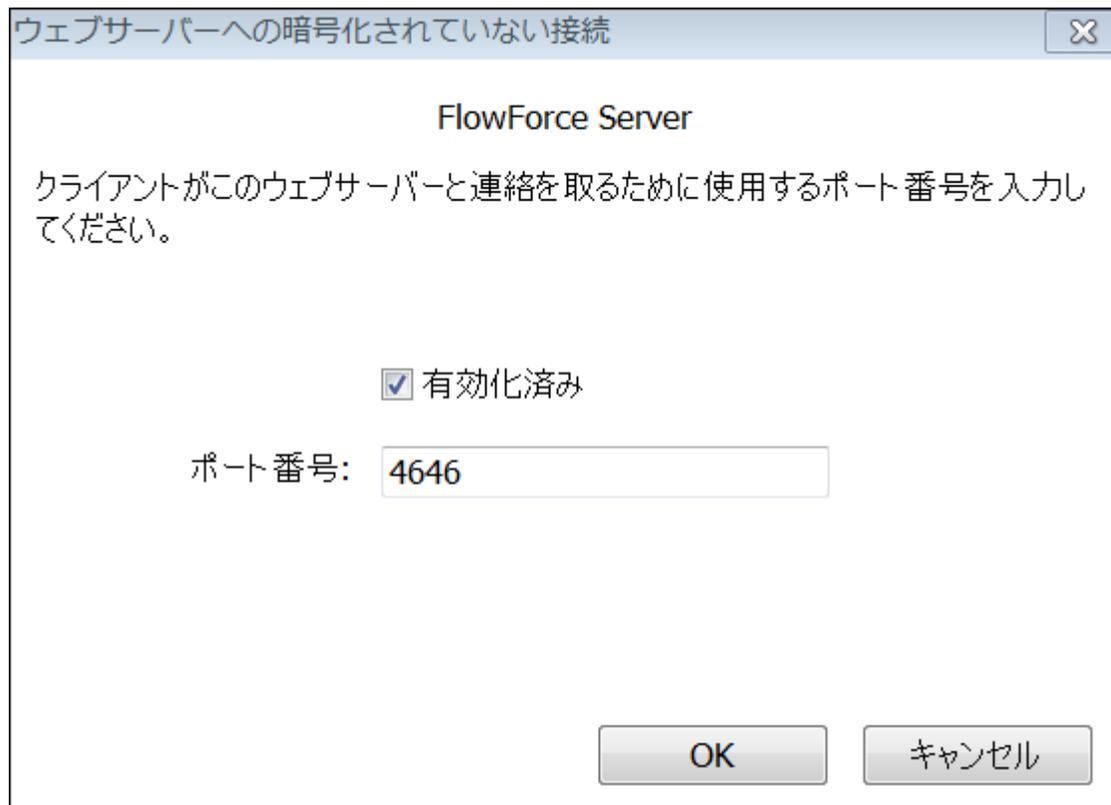


インストール後に設定を変更する場合、Windows Services 管理コンソールを開き、必要とするサービスのプロパティを変更します。Windows Services 管理コンソールを開くには、コマンドラインウィンドウを開き、`Services.msc` を入力します。

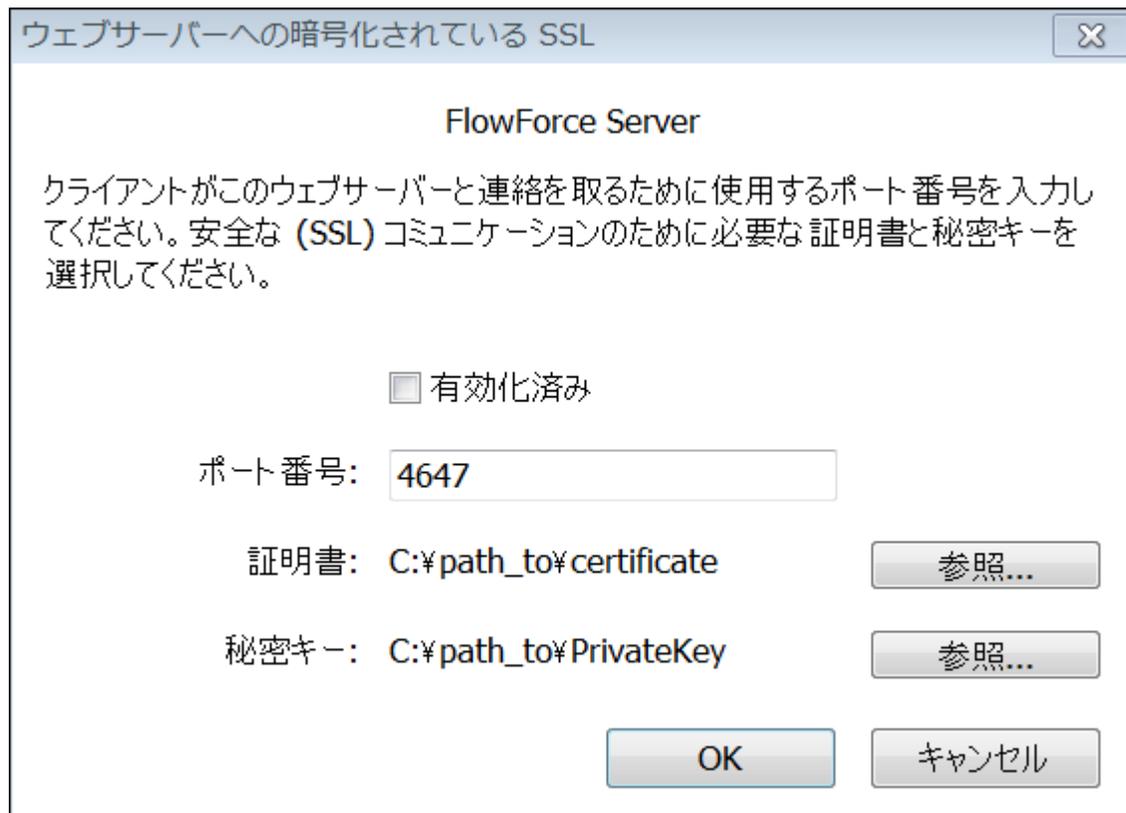
### 2.6.3 ネットワーク構成 (Windows のみ)

Windows オペレーティングシステム上に FlowForce Server をインストール中、以下の設定を構成することができます。

- FlowForce Server への暗号化されていない接続が許可されている場合のポート番号。



- FlowForce Server への安全な (SSL-暗号化済み) 接続が許可されているポート。デフォルトで安全な接続は無効化されます。SSL-暗号化済みの接続を有効化するには、適用可能な場合、証明書ファイル、秘密キー、中間証明書が必要です。[FlowForce Server のために SSL を有効化する](#) で説明されているとおり、他の必要条件が適用されます。



メモ 上にリストされる設定は FlowForce Web Server のために構成することもできます。

インストール中に上記の設定を定義する場合は、インストールダイアログボックスの関連するタブ (FlowForce Server または FlowForce Web Server) をクリックし、[変更] をクリックします。



上記の設定を後で行う場合は [次へ] をクリックします。この場合、FlowForce Server は、ダイアログボックスの対応するタブの下に表示されるデフォルトのオプションを使用してインストールされます。インストール後、[ネットワーク設定の定義](#)で説明されているとおく設定を変更することができます。

## 2.6.4 データを手動で移行する

Windows では、構成データを手動で移行する必要は通常ありません。FlowForce Server の新しいメジャーバージョンをインストールする際に、前のバージョンが既にインストールされている場合、インストールウィザードは、構成データの移行に関してプロンプトします。

構成データを手動で移行する場合、以下の手順を踏んでください。

構成データを手動で移行する方法:

1. システム通知エリアで、Altova ServiceController  が実行されていることを確認してください。それ以外の場合、Altova ServiceController を開始してください。
2. FlowForce Server サービスとFlowForce Web Server サービスを停止します。
3. 2021 インストールウィザードによりインストールされたFlowForce Server データフォルダーを削除してください。Windows バージョンによりデータフォルダーへのパスは異なります ([FlowForce Server の構成データの保管方法を参照してください](#))。
4. コマンドプロンプトでmigratedb コマンドを使用してFlowForce 実行可能ファイルを実行します。

```
"C:\Program Files (x86)\Altova\FlowForceServer2021\bin\FlowForceServer.exe"
migratedb
```

```
--datadir=C:\ProgramData\Altova\FlowForceServer2021\data  
--olddatadir=C:\ProgramData\Altova\FlowForceServer2017\data
```

5. FlowForce Server サービスとFlowForce Server サービスを開始します。

## 2.6.5 インストールフォルダー

Windows 上でFlowForce Server のデフォルトのインストールディレクトリは、以下の通りです:

|                         |                                |
|-------------------------|--------------------------------|
| Windows 7、8、10          | C:\Program Files\Altova\       |
| 64 ビット OS 上の32 ビットバージョン | C:\Program Files (x86)\Altova\ |

## 2.7 Altova Server の移行

このトピックでは、1つのサーバーマシンから多種のサポートされるプラットフォームを含む)他のマシンへのAltova サーバーの移行について説明されています。これらの手順は、「古いサーバー」または「古いマシン」という用語をデータを移行する元のマシンを説明するために使用し、「新しいサーバー」または「新しいマシン」という用語をデータを移行する先のマシンを説明するために使用します。

### Altova LicenseServer

Altova LicenseServer を1つのサーバーマシンから、他のがクォンタイムの少ないものに移行するには、以下のステップを実行してください！

1. Altova LicenseServer を新しいマシンにインストールします。
2. 新しいマシンのAltova LicenseServer を停止し、**licenseserver.db** データベースファイルを古いマシンから新しいマシンにコピーします。後に、何らかの理由で最初の状態に戻す必要があるかもしれないので、**licenseserver.db** を新しいサーバーでバックアップしてください。 **licenseserver.db** ファイルは、LicenseServer アプリケーションディレクトリ内にあり、パスはオペレーティングシステムにより異なります。

|         |                                     |
|---------|-------------------------------------|
| Windows | C:\ProgramData\Altova\LicenseServer |
| Linux   | /var/opt/Altova/LicenseServer       |
| macOS   | /var/Altova/LicenseServer           |

3. 古いマシンのAltova LicenseServer を停止し、新しいマシンでAltova LicenseServer サービスを起動してください。LicenseServer の2つのインスタンスが同時に作動していることを確認してください。
4. すべてのAltova サーバー製品を新しいAltova LicenseServer に登録します。移行するライセンスの数が少ない場合は、手動で行うことができます。移行するライセンスの数が多い場合、Altova LicenseServer 管理下にあるそれぞれのサーバー製品のために“register” コマンドを呼び出すコマンドラインスクリプトを作成し登録を自動化することができます。

### Altova FlowForce Server

[バックアップとカマイ](#)を参照してください。

### Altova MobileTogether Server

MobileTogether Server ドキュメント内で説明されているとおり、バックアップと復元の機能を使用してください。

### Altova MapForce Server

MapForce Server、では、新しいマシンへの移行は、古いマシンから新しいマシンへのライセンスの再割り当てから構成されています。手順は以下の通りです:

1. MapForce Server を新しいマシンにインストールします。FlowForce Server インストールの一部として既にインストール済みの場合、このステップを無視してください。
2. 新しいマシンで、MapForce Server をAltova LicenseServer に登録します。
3. 古いマシンでは、クライアントがサーバーを使用していないことを確認してください。(例えば、マッピングが実行されない場合)。
4. Altova LicenseServer 管理ページを開き、古いMapForce Server マシンからライセンスを非アクティブ化し、新しいマシンに再割り当てします。

### Altova RaptorXML(XBRL) Server

RaptorXML(XBRL) Server では、新しいマシンへの移行は、古いマシンから新しいマシンへのライセンスの再割り当てから構成されています。手順は以下の通りです:

1. RaptorXML(XBRL) Server を新しいマシンにインストールします。FlowForce Server インストールの一部として既にインストール済みの場合、このステップを無視してください。
2. 新しいマシンで、RaptorXML(XBRL) Server を Altova LicenseServer に登録します。
3. 古いマシンでは、クライアントがサーバーを使用していないことを確認してください。(例えば、変換が実行されない場合)。
4. Altova LicenseServer 管理ページを開き、古いRaptorXML(XBRL) Server マシンからライセンスを非アクティブ化し、新しいマシンに再割り当てします。

## Altova StyleVision Server

StyleVision Server、では、新しいマシンへの移行は、古いマシンから新しいマシンへのライセンスの再割り当てから構成されています。手順は以下の通りです:

1. StyleVision Server を新しいマシンにインストールします。FlowForce Server インストールの一部として既にインストール済みの場合、このステップを無視してください。
2. 新しいマシンで、StyleVision Server を Altova LicenseServer に登録します。
3. 古いマシンでは、クライアントがサーバーを使用していないことを確認してください。(例えば、変換が実行されない場合)。
4. Altova LicenseServer 管理ページを開き、古いStyleVision Server マシンからライセンスを非アクティブ化し、新しいマシンに再割り当てします。

### 3 サーバーの構成

このチャプターはFlowForce Server 構成の命令について説明されています。これはインストール後に行う構成、およびサービスの開始と停止、データのバックアップなど多種のサーバーメンテナンス、およびルーチンタスクなどが含まれています。

FlowForce Server とその設定を以下のアプローチを使用して管理することができます。

|                          |   |
|--------------------------|---|
| <p>セットアップページから</p>       | <p>新規にFlowForce Server をインストール後、最初に定義されるのは<b>FlowForce Web Server</b> と<b>FlowForce Server</b> をリスンする通常ホスト名 (またはIP アドレス) とポートです。これらに関する詳細については、以下を参照ください:</p> <ul style="list-style-type: none"> <li>• <a href="#">セットアップ</a></li> <li>• <a href="#">ネットワーク設定の定義</a></li> <li>• <a href="#">SSL 暗号化のセットアップ</a></li> </ul> |
| <p>Web 管理インターフェイスから</p>  | <p>上記のネットワーク設定が設定されると、以下の順序で構成することができます:</p> <ul style="list-style-type: none"> <li>• <a href="#">デフォルトのタイムゾーン</a></li> <li>• <a href="#">電子メールのパラメーター</a></li> <li>• <a href="#">ディレクトリサービス設定</a></li> <li>• <a href="#">ログ中の設定</a></li> <li>• <a href="#">ユーザーアクセス</a></li> </ul>                                       |
| <p>コマンドラインインターフェイスから</p> | <p><a href="#">コマンドラインインターフェイス</a>を参照してください。</p>  |
| <p>構成ファイルの編集をして</p>      | <p><a href="#">FlowForce Server アプリケーションデータ</a>を参照してください。</p>   |

## 3.1 セットアップ

FlowForce Server と FlowForce Web Server がリスンするインターフェイスとポートを含む、FlowForce Server セットアップページは、Web 管理インターフェイスのデフォルトのネットワークアドレスとポートと LicenseServer のネットワークアドレスを指定することができます。ブラウザ内でセットアップページを開く（または URL を取得する）最も簡単な方法は、以下で説明される通り管理特権を使用してセットアップモードを実行することです。更に高度な構成に関しては以下の「高度のオプション」を参照してください。

セットアップページから構成することの可能なすべての設定への参照は[ネットワーク設定の定義](#)を参照してください。

### Linux

Linux でセットアップモードを開く方法

1. ターミナルを開き、アプリケーションのデータディレクトリに変更します:

```
cd /var/opt/Altova/FlowForceServer2021
```

2. 以下の内の1つを行います:

- a. Linux をグラフィカルなインターフェイスで作動する場合、FlowForce Web server 実行可能ファイルを **setup** コマンドで実行します:

```
sudo /opt/Altova/FlowForceServer2021/bin/flowforcewebserver setup
```

- b. Linux をグラフィカルなユーザーインターフェイスで作動する場合、**--listen** オプションを追加しつづ上記と同じ **setup** コマンドを作動します。後者はセットアップページを使用できるインターフェイスとポートを指定します。これが行われると、他のマシン上のブラウザからセットアップページに接続することができます。

上記のステップを実行すると、ブラウザのアドレスバーにコピーして貼り付けることのできる2つの代替 URL がターミナルで開かれます。最初の URL が作動しない場合、2番目のものを使用してください。

### macOS

macOS でセットアップモードを開く方法

- アプリケーション内で、**FlowForce Server 2021** アイコンをダブルクリックします。

緊急の場合セットアップモードを以下のように入力することができます:

1. ルート特権を使用してターミナルを開き、アプリケーションのデータディレクトリに変更します:

```
cd /var/Altova/FlowForceServer2021
```

2. **setup** コマンドを使用して FlowForce Web サーバー 実行可能ファイルを実行します:

```
/usr/local/Altova/FlowForceServer2021/bin/flowforcewebserver setup
```

上記のステップを実行すると、ブラウザのアドレスバーにコピーして貼り付けることのできる2つの代替 URL がターミナルで開かれます。最初の URL が作動しない場合、2番目のものを使用してください。

## Windows

Windows でセットアップモードを開く方法

- スタートメニューから **Altova FlowForce Server 2021** > FlowForce Server セットアップページを選択します。

緊急の場合セットアップモードを以下のように入力することができます。

- 管理者としてコマンドプロンプトウィンドウを開き、アプリケーションのデータディレクトリに変更します。

```
cd C:\ProgramData\Altova\FlowForceServer2021
```

- setup** コマンドを使用して FlowForce Web サーバー実行可能ファイルを実行します。

```
"C:\Program Files\Altova\FlowForceServer2021\bin\FlowForceWebServer.exe" setup
```

## セットアップページ

上記のステップを実行すると新規のブラウザウィンドウ内でセットアップページが開かれます (または、URL がターミナルで表示され、これをコピーしてブラウザウィンドウで張り付けることができます)。

LicenseServer

VIEPDEV02.vie.altova.com

LicenseServer により登録

### FlowForce Web Server

暗号化されていない接続

有効化済み:

バインドアドレス:  ローカルのみ (127.0.0.1)  その他:  ポート: 8082

### SSL 暗号化された接続

有効化済み:

バインドアドレス:  ローカルのみ (127.0.0.1)  その他:  ポート: 8089

証明書ファイル:

秘密キーファイル:

Windows では詳細コマンドプロンプトウィンドウが開かれます。例

```

FlowForce Setup
http://doc-w10x64:52281/setup?key=83D7B575F36F15459F77E13DB891742E
http://localhost:52281/setup?key=83D7B575F36F15459F77E13DB891742E

This window shall automatically close once FlowForce setup completes.

You can manually end setup using Ctrl-C or the close button.

```

このウィンドウはセットアップ中開かれており、セットアップページで設定を適用し FlowForce Server サービスを再起動するボタンをクリックされると通常自動的に閉じられます。

設定を適用し FlowForce Server サービスを再起動するボタンをセットアップページからクリックすると、構成は .ini ファイルに保存されます。具体的にはセットアップページを管理者特権で上記の通り開くと、以下のファイルが更新されます:

- DATADIR\flowforcewebserver.ini
- DATADIR\flowforceserver.ini

DATADIR が以下のディレクトリを参照する箇所:

- /var/opt/Altova/FlowForceServer2021/data (Linux)
- /var/Altova/FlowForceServer2021/data (macOS)
- C:\ProgramData\Altova\FlowForceServer2021\data (Windows)

柔軟性が必要な場合、高度なオプションを使用して setup コマンドを以下のように実行することができます。

### 高度なセットアップオプション

setup コマンドは下記にリストされる高度なオプションの一部をサポートします。

|                  |  |
|------------------|--|
| <b>--datadir</b> | <p>このオプションを使用して .ini 構成ファイルが書き込まれる箇所のディレクトリへのパスを提供できます (通常は上記の DATADIR です)。ターゲットディレクトリが高度な「書き込み」特権を必要とする場合セットアップモードをそのディレクトリに書き込み可能な特権 (root または administrator) アカウントを使用して実行する必要があります。</p> <p>このオプションが設定されていない場合 .ini ファイルは現在のディレクトリの .\data サブディレクトリに保存されます。このためセットアップコマンドを上記の通りアプリケーションディレクトリから実行、または --datadir オプションを提供して実行します。</p> |
| <b>--listen</b>  | <p>デフォルトではセットアップが実行される都度、セットアップページの URL はフリーのランダムなポートに再生成されます。例: <code>http://localhost:50492/setup</code>。</p> <p>このオプションにより (localhost または 127.0.0.1 以外) をリスンする代替のインターフェイスポート 組み合わせを設定することができます。他のマシン上のブラウザからセットアップページにアクセスする場合これは役に立ちます。</p> <p>例えば コマンド:</p> <pre>flowforcewebserver setup --listen=0.0.0.0:10008</pre>            |

|       |  |
|-------|--|
|       | <p>全てのインターフェイス上でポート 10008 上でリスンのセットアップが作成されます。以下の点に注意してください</p> <ul style="list-style-type: none"> <li>• 実際のデータディレクトリと外部ネットワークインターフェイスへのバインディングと共にセットアップの実行の特権を使用するように設定することが奨励されます。これを行う場合次のオプション(--key) は役に立ちます。代替として--datadir オプションを使用して一時的なデータディレクトリを提供します。これにより公開済み公開ページから実際のFlowForce Server 構成を更新することを回避できます。</li> <li>• ポートは既に使用されているため、ノーマルな(セットアップではない) FlowForce Web Server または FlowForce Server インスタンスと同じポートを使用しないでください。</li> </ul> <p>バインドアドレス(インターフェイス) がローカルではない場合、指定されたポートを介してアクセスが可能になるようにオペレーティングシステムのファイアウォールを構成する必要がある場合があります。</p>   |
| --key | <p>このオプションによりセットアップページのためにアクセスキーを設定することができます。この場合、URL 内に正確なアクセスキーが提供されている場合のみセットアップページを保存することができます。キーはURL 内に含まれる必要のある任意の文字列であることができます。例えば以下のような Linux コマンドを実行すると</p> <pre>flowforcewebserver setup --listen=wild.berries.com:8015 --key=all_cats_love_fish --datadir=/var/opt/Altova/FlowForceServer2021/data</pre> <p>接続する URL は以下の通りになります。</p> <p><a href="http://wild.berries.com:8015/setup?key=all_cats_love_fish">http://wild.berries.com:8015/setup?key=all_cats_love_fish</a></p> <p>目的のためにキーフレーズが安全であることを確認してください。</p> <p>HTTPS パラメータを構成するために自身を使用しているため、セットアップページはHTTPS を使用しないことにご注意してください。セットアップページは継続的に使用することができず、セットアップ中にのみ公開されていることにご注意してください。</p> |

## 3.2 ネットワーク設定の定義

FlowForce Server と FlowForce Web Server がリスニングするホストとポート、およびネットワークに関連した設定をセットアップページから指定することができます。[セットアップページ](#)をオペレーティングシステムに従い異なる方法で開くことができます。または、設定の大半は、構成ファイルを用いて定義することができます。[構成ファイルの参照](#)を参照してください。セットアップページ内で定義されている設定は、FlowForce Server の新規のマイナーバージョンをインストール場合保管されます。メジャーバージョンのマイナーバージョンをインストールすると、インストール中、前のメジャーバージョンからデータを移行する場合のみ設定は保管されます。

構成可能な設定は下記リストされる通りです。

### LicenseServer

FlowForce Server は、LicenseServer に登録されている必要があります（次を参照する Altova LicenseServer）。インストール中に LicenseServer ホストが指定されていない場合、Altova LicenseServer が作動するマシンの IP アドレス、または、ホスト名を入力します。（LicenseServer がローカルでインストールされている場合）これはローカルマシン、または、ネットワークアドレスであることができます。



The screenshot shows a configuration window titled "LicenseServer". It features a text input field with "localhost" entered, a dropdown arrow, and two icons (magnifying glass and pencil) to its right. Below the input field is a button labeled "LicenseServer により登録".

### FlowForce Web Server

設定のこのグループは FlowForce Server Web 管理インターフェイスへのブラウザからの HTTP(S) リクエストを処理するためのサービスである FlowForce Web Server サービスに適用可能です。（「FlowForce Web Server」と「FlowForce Server」の違いを理解するには、次を参照してください [作動の仕組み](#)）。

### FlowForce Web Server

#### 暗号化されていない接続

有効化済み:

バインドアドレス:  全てのインターフェイス (0.0.0.0)  その他:  ポート: 8082

ホスト名:

---

#### SSL 暗号化された接続

有効化済み:

バインドアドレス:  全てのインターフェイス (0.0.0.0)  その他:  ポート: 8089

ホスト名:

証明書ファイル:

秘密キーファイル:

証明書チェーンファイル:

|                     |  |
|---------------------|--|
| 暗号化されていない接続 / 有効化済み | <p>FlowForce Web Server へのプレーンな HTTP (暗号化されていない) 接続を有効化するためこのチェックボックスを選択します。インストール中、または構成ファイルから変更されていない場合、デフォルトでは (暗号化されていない) プレーンな HTTP からの接続が有効化されています。<a href="#">構成ファイルの参照</a> を参照してください。</p> <p>* FlowForce Server インストール中のネットワークシステムの変更は Windows のみで許可されています。</p>  |
| バインドアドレス            | <p>Windows 上では FlowForce Web サーバー管理者インターフェイスは、デフォルトでポート 8082 上のすべてのネットワークインターフェイスで使用することができます。Linux と Mac OS では、ポート番号は、インストール中にランダムに選択されます。「ローカルのみ」または「全てのインターフェイス」以外にカスタムのアドレスを指定するには、「他の」テキストボックスに入力します。</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>バインドアドレス (インターフェイス) がローカルではない場合、指定されたポートを介してアクセスが可能になるようにオペレーティングシステムのファイアウォールを構成する必要がある場合があります。</p> </div> |
| ポート                 | <p>FlowForce Web Server がリスニングする TCP ポートを指定します。ポートは使用済みであってはなりません。</p>   |
| ホスト名                | <p>ホスト名 フィールドは空でない場合、バインドのために使用される固定ホスト名を設定します。ネットワーク上の他のマシンが接続するために使用する FlowForce Web Server を作動するマシンの名前を設定します。</p> <p>通常、FlowForce は使用する適切なホストを自動的に検知します。このフィールドを明示的に設定した場合、自動検知が優先されます。所属機関内のネットワーク構成により somehost または somehost.example.org などの値を使用する必要がある場合があります。</p> <p>バインドに関連したホスト名は SSL のために使、および Windows 上の Altova Service</p>   |

|              |   |
|--------------|---|
|              | <p>Controller により使用されています ( <a href="#">FlowForce Web Server のためにSSLを有効化する</a> を参照してください)。SSL が有効化されている場合、ホスト名は証明書の共通の名前 プロトコルに一致する必要があります。</p> <p>バインドアドレスがローカルではない場合 (すなわちバインドアドレスフィールドが[Local (127.0.0.1)] 以外に設定されている場合)ホスト名の設定は意味があります。</p> |
| SSL 暗号化された接続 | <a href="#">FlowForce Web サーバーのためにSSLを有効化する方法</a> も参照してください。  |

## FlowForce Server

FlowForce ジョブからHTTP(S) クライアントに作成されたサービスを公開するサービスである FlowForce Server サービスに設定のこのグループを適用することが可能です。

### FlowForce Server

#### 暗号化されていない接続

有効化済み:

バインドアドレス:  ローカルのみ (127.0.0.1)  その他:  ポート:

ホスト名:

---

#### SSL 暗号化された接続

有効化済み:

バインドアドレス:  ローカルのみ (127.0.0.1)  その他:  ポート:

ホスト名:

証明書ファイル:

秘密キーファイル:

証明書チェーンファイル:

|                     |   |
|---------------------|---|
| 暗号化されていない接続 / 有効化済み | <p>FlowForce Server へのプレーンな HTTP (暗号化されていない) 接続インストール中、または構成ファイルから変更されていない場合、デフォルトでは (暗号化されていない) プレーンな HTTP からの接続が有効化されています。 <a href="#">構成ファイルの参照</a> を参照してください。</p> <p>*FlowForce Server インストール中のネットワークシステムの変更は Windows のみで許可されています。</p>  |
| バインドアドレス            | <p>FlowForce Server のデフォルトの設定は、ポート <b>4646</b> 上の同じマシン(127.0.0.1)からのリクエストを暗号化されていない接続を使用して受け入れます。リモートのマシンからプレーン HTTP を使用して、Web サービスとしてジョブを開始する場合は、バインドアドレスエディタから「全てのインターフェイス(0.0.0.0)」を選択します。</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>バインドアドレス (インターフェイス) がローカルではない場合、指定されたポートを介してアクセスが可能になるようにオペレーティングシステムのファイアウォールを構成する必要がある場合があります。</p> </div> |

|              |  |
|--------------|--|
| ポート          | FlowForce Server がリスンする TCP ポートを指定します。ポートは使用済みであってはなりません。  |
| ホスト名         | <p>フィールド <b>ホスト名</b> は FlowForce Server が Web サービスとして公開されているジョブにアクセスするクライアントからの接続をリスンするインターフェイスに与えられるホスト名を指定します。</p> <p>バインドアドレスが[ローカル(127.0.0.1)]に設定されている場合ホスト名の設定は意味がありません。所属機関内のネットワーク構成により <code>somehost</code> または <code>somehost.example.org</code> などの値を使用する必要がある場合があります。</p> <p>バインドに関連したホスト名は SSL のために使用されます (<a href="#">FlowForce Server のために SSL を有効化する</a> を参照してください)。SSL が有効化されている場合、ホスト名は証明書の名前 プロトコルに一致する必要があります。</p> <p>ホスト名は上のでも使用されます。ホスト名を設定しないと、FlowForce は Windows 上の Altova Service Controller により使用されている最初の適切なホスト名を自動的に検知します。</p> <p><b>hostname</b> が構成されている場合 FlowForce Web インターフェイスはホームページの <a href="#">アクティブナビゲーターサービス</a> セクション内のリンクを含む Web サービスとして公開されているジョブをナビゲートするためにクリック可能なリンクを表示することができます。Web サービスの呼び出し ボタンが新規のブラウザウィンドウ内で Web サービスの呼び出しを可能にするジョブ構成ページのサービスセクション内で使用できるようになります。詳細に関しては <a href="#">Web サービスの公開</a> を参照してください。</p> |
| SSL 暗号化された接続 | <a href="#">FlowForce Server のために SSL を有効化する方法</a> も参照してください。  |

### 3.2.1 ファイル参照の構成

[セットアップページ](#) または .ini 構成ファイルから FlowForce Server と FlowForce Web Server のネットワーク設定を以下で説明されるように構成することができます。

上記のディレクトリ内では 2 つの .ini ファイルが存在します。1 つは FlowForce Server (**flowforce.ini**) のために、もう一つは FlowForce Web Server (**flowforceweb.ini**) のために、.ini 構成ファイルは、次のパスで見つけることができます：

|         |   |
|---------|---|
| Linux   | <pre>/var/opt/Altova/FlowForceServer2021/data/flowforce.ini /var/opt/Altova/FlowForceServer2021/data/flowforceweb.ini</pre>             |
| macOS   | <pre>/var/Altova/FlowForceServer2021/data/flowforce.ini /var/Altova/FlowForceServer2021/data/flowforceweb.ini</pre>                     |
| Windows | <pre>C:\ProgramData\Altova\FlowForceServer2021\data\flowforce.ini C:\ProgramData\Altova\FlowForceServer2021\data\flowforceweb.ini</pre> |

上記のディレクトリ内では 2 つのサブル .ini ファイルが存在し、テンプレートとして使用することができます：

- flowforce.ini.template
- flowforceweb.ini.template

.ini ファイルの編集後、対応するサービス (FlowForce Server または FlowForce Web Server) を再起動してください。詳細に関しては、[次](#)を参照してください。

- [サービスの開始と停止 \(Linux\)](#)
- [サービスの開始と停止 \(macOS\)](#)
- [サービスの開始と停止 \(Windows\)](#)

サンプルの `flowforce.ini` ファイルは以下のようになります:

```
[Listen]
active=1
host=127.0.0.1
port=4646
hostname=

[ListenSSL]
active=1
SSL=1
host=0.0.0.0
port=4647
hostname=

[SSL]
certificate=/path/to/certificate.crt
private_key=/path/to/private_key.key
certificate_chain=/path/to/certificate_chain
```

サンプルの `flowforcweb.ini` ファイルは以下のようになります:

```
[Listen]
active=1
host=0.0.0.0
port=8082
hostname=example.domain.org

[ListenSSL]
active=1
SSL=1
host=0.0.0.0
port=8083
hostname=example.domain.org

[SSL]
certificate=path/to/certificate.crt
private_key=path/to/private_key.key
certificate_chain=/path/to/certificate_chain

[FlowForce]
host=127.0.0.1
port=4646
hostname=
```

.ini ファイルは、下に説明されるようにセクションに整理されています。2つのファイルの違いは下で説明されています。

### [Listen]

[Listen] セクションは、HTTP 接続設定を定義します。複数の [Listen] セクションを定義することも可能です。各 [Listen] セクションは、「Listen」で開始する必要があります。例: [ListenSSL]。

|                       |   |
|-----------------------|---|
| active                | <p>(任意) [Listen] セクションを有効化、または無効化します。有効な値:</p> <p>0 無効化済み</p> <p>1 有効化済み</p> <p>例えば active=1 は、HTTP 接続が有効化済みであることを意味します。</p>  |
| ssl                   | <p>(任意) この [Listen] セクションのための SSL サポートを有効化します。有効な値:</p> <p>0 無効化済み</p> <p>1 有効化済み</p> <p>SSL サポートを有効化するには、ssl=1 を設定し、[SSL] セクションを以下に示されるように作成します。</p>   |
| host                  | <p>FlowForce (Web) Server のネットワーク IP アドレスを指定します。例えば、127.0.0.1。これは IPv4 または IPv6 アドレスであることができます。0.0.0.0 をすべてのインターフェイスをリスンするために使用します。ローカルアクセスのためには、127.0.0.1 を使用します。</p>  |
| port                  | <p>FlowForce (Web) Server がリスンするポートを指定します。このポートが既に使用されていないことを確認してください。</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>IP アドレス (インターフェイス) がローカルではない場合、指定されたポートを介してアクセスが可能になるようにオペレーティングシステムのファイアウォールを構成する必要がある場合があります。</p> </div>   |
| hostname              | <p>ホスト名 フィールドは空でない場合、バインドのために使用される固定ホスト名を設定します。ネットワーク上の他のマシンが接続するために使用する FlowForce Web Server を作動するマシンの名前を設定します。</p> <p>通常、FlowForce は使用する適切なホストを自動的に検知します。このフィールドを明示的に設定した場合、自動検知が優先されます。所属機関内のネットワーク構成により somehost または somehost.example.org などの値を使用する必要がある場合があります。</p> <p>バインドに関連したホスト名は SSL のために使、および Windows 上の Altova Service Controller により使用されています (<a href="#">FlowForce Web Server のために SSL を有効化する</a> を参照してください)。SSL が有効化されている場合、ホスト名は証明書の共通の名前 プロキシに一致する必要があります。</p> |
| max_request_body_size | <p>このオプションにより FlowForce Server または FlowForce Web Server、への HTTP リクエストの最大サイズをバイトで指定できるようになります。例:</p> <pre>max_request_body_size=500000000</pre> <p>デフォルトでは、明示的なリミットは 100 MB (100,000,000 バイト) 未満です。このオプションを以下のシチュエーションで明示的に設定する必要がある場合があります:</p> <ol style="list-style-type: none"> <li>1. FlowForce Web サービスをジョブとして公開するよう呼び出し、HTTP リクエストボディがデフォルトのリミットより大きい場合。</li> </ol>   |

|  |  |
|--|--|
|  | <p>2. MapForce から FlowForce Server にマッピングをデプロイすると、入力ファイルがデフォルトのビットより大きい場合。</p> <p>ケース1 の場合、オプションは <b>flowforce.ini</b> ファイル内でのみ設定される必要があります。ケース2 の場合、オプションは <b>flowforce.ini</b> と <b>flowforcweb.ini</b> ファイル内の両方で設定される必要があります。</p> |
|--|--|

## [SSL]

このセクションは SSL/HTTPS 接続設定を定義します。

|                   |                             |
|-------------------|-----------------------------|
| certificate       | PEM 書式の証明書ファイルへの絶対パスを指定します。 |
| private_key       | 秘密キーファイルへの絶対パスを指定します。       |
| certificate_chain | 証明書チェーンファイルへの任意のパスを指定します。   |

## [FlowForce]

このセクションは FlowForce Web Server (**flowforcweb.ini** ファイル) のみに適用することができます。FlowForce Web サービスと FlowForce Server サービス間の接続の詳細を定義します。

|          |  |
|----------|--|
| ssl      | <p>(オプション) FlowForce Web Server と FlowForce Server の間の接続のための SSL サポートを有効化します。有効な値</p> <p><b>0</b> 無効化済み</p> <p><b>1</b> 有効化済み</p>  |
| host     | <p>FlowForce Server の IP アドレスを指定します。</p> <p>FlowForce Server がすべてのインターフェイスにバインドされていない場合、この値は <b>flowforce.ini</b> の “[Listen]” セクションの一つと同じである必要があります。それ以外の場合は 127.0.0.1 です。</p> <p>SSL がそのポートで有効化されていると、この値は <b>flowforce.ini</b> の “[SSL]” セクション内で構成済みの証明書の [共通名] プロパティに一致する必要があります。</p>  |
| port     | <p>FlowForce Web Server が FlowForce Server に接続する TCP ポートを指定します。</p> <p>この値は同じポート番号を持つ <b>flowforce.ini</b> の対応する “[Listen]” または “[ListenSSL]” セクションの一つと同じである必要があります。</p> <p>SSL がこのポート上で有効化されていると <b>host</b> と <b>hostname</b> (または <b>hostname</b> が存在しない場合 <b>host</b>) は <b>flowforce.ini</b> の “[SSL]” セクション内で構成されている証明書の [共通名] に一致する必要があります。</p> |
| hostname | <p>空ではない場合、Web サービスとして公開されている FlowForce ジョブに接続するためこのフィールドはネットワーク上の他のマシンにも使用されている固定ホスト名を設定します (<a href="#">Web サービスとして公開する</a> を参照してください)。</p>  |

|             |   |
|-------------|---|
|             | <p>所属機関内のネットワーク構成により <code>somehost</code> または <code>somehost.example.org</code> などの値を使用する必要がある場合があります。</p> <p>SSL がそのポートで有効化されていると、この値は <code>flowforce.ini</code> の “[SSL]” セクション内で構成済みの証明書の [共通名] プロパティに一致する必要があります。</p> <p>Altova Service Controller もホスト名を使用しています。ホスト名を設定しないと、FlowForce は Altova Service Controller により使用されている最初の適切なホスト名を自動的に検知します。</p> <p><code>hostname</code> が構成されている場合 FlowForce Web インターフェイスはホームページの <a href="#">アクティブなトリガーとサービス</a> セクション内のリンクを含む Web サービスとして公開されているジョブをナビゲートするためクリック可能なリンクを表示することができます。Web サービスの呼び出し ボタンが新規のブラウザウィンドウ内で Web サービスの呼び出しを可能にするジョブ構成ページのサービスセクション内で使用できるようになります。</p> |
| certificate | <p>(オプション) FlowForce Server により受け入れられるサーバー証明書を定義します。証明書が与えられていない場合、システムルート CA 証明書が使用され、サーバー証明書が検証されます。存在する場合、この値は ( <code>flowforce.ini</code> ファイル内の ) FlowForce Server を使用する証明書にこの値が一致する必要があります。</p>  |

### [FlowForceWeb]

このセクションは FlowForce Web Server ( `flowforceweb.ini` ファイル ) のみに適用することができます。

|          |   |
|----------|---|
| timezone | <p>FlowForce Web Server のデフォルトのタイムゾーンを指定します。例えば <code>timezone=Europe/Berlin</code></p> |
|----------|---|

## 3.3 SSL 暗号化のセットアップ

FlowForce を構成し、次の HTTP 接続を SSL 証明書を使用して暗号化することができます:

1. ブラウザーと FlowForce Web Server 間の接続。
2. (例えば、クライアントアプリケーションなどの) Web サービスコンシューマーと FlowForce Server サービス間の接続。
3. FlowForce Web Server と FlowForce Server 間の内部接続 ( FlowForce Server が FlowForce Web Server とどのように異なるかに関する詳細は、次を参照してください [リンク](#) )。

上記の 1 と 2 接続のために、その証明書に対応する SSL 証明書と秘密キーが必要になります。セキュリティ上の理由から、各接続のために個別の SSL 証明書と秘密キーが必要になります。両方の接続のために同じ証明書と秘密キーを使用する場合、FlowForce Server と FlowForce Web Server の両方は同じ完全修飾ドメイン名 (FQDN) を必要とします。例えば、FlowForce Web Server が `https://somehost:8083` をリスンする場合、FlowForce Server は `https://somehost:4647` をリスンする必要もあります。ポートは後で変更することができるため、この場合ホスト名のみが重要であることに注意してください。

上記の接続 3 に関しては、3 番目の証明書と秘密キーのペアは必要無く、同じ SSL 証明書を FlowForce Server のために使用することができます。この場合、FlowForce Web Server は FlowForce Server の HTTP クライアントとしての役割を果たします。

FlowForce Server 内の SSL 接続を暗号化するために必要な証明書を取得するには、以下のオプションがあります:

1. CSR (証明書署名要求) を生成し、DigiCert、Comodo などの公開されている証明書機関 (CA) により署名してください。ブラウザー (またはオペレーティングシステム) はすでに証明書機関を信頼するため、CA ブラウザーの大多数はこのような証明書機関により署名されているサーバー証明書を信頼します。証明書機関により署名済みの証明書を取得する方法に関しては、次を参照してください [証明書機関により SSL 証明書を署名する方法](#)。
2. または、FlowForce Server がプライベートのネットワーク上で作動する場合、そして、この期間内でこれを行う権利を有している場合、自身の SSL ルート証明書機関を構成することが可能です。ブラウザーまたはオペレーティングシステムはこのような機関をデフォルトでは信頼せず、自己署名ルート証明書を信頼するため FlowForce Server に接続する各マシン (または場合によってはブラウザー) を構成する必要があります。それ以外の場合、ブラウザーが「このサイトは安全ではありません」または「Web サービスの呼び出しに失敗しました」などの警告を表示します。詳細に関しては、次を参照してください [自己署名済みの SSL 証明書の作成方法](#)。

### 3.3.1 証明書機関により SSL 証明書を署名する方法

信頼されている証明書機関 (CA) から SSL 証明書を購入する場合、秘密キーと CSR (証明書署名要求) が必要です。秘密キーは安全に保管され、公開されないようにしてください。処理中に証明書機関により CSR が必要とされます。

秘密キーと CSR をオペレーティングシステム上に既存のツール、または第三者パーティツールを使用して作成することができます (Mac 上の **Keychain Access**、Linux 上の **openssl**)。このサンプルは OpenSSL ツールキット (<https://www.openssl.org/>) を使用します。OpenSSL は、オープンソースライブラリであり、コマンドラインで使用する前にコンパイルされる必要があります。OpenSSL のためのコンパイルとインストール命令は、各オペレーティングシステムで異なり、このドキュメントの範囲ではありません。Linux と Mac マシン上で、OpenSSL はすでに使用できる可能性があります。それ以外の場合、コマンドラインからインストール、更新することができます。OpenSSL が存在するか、下記のコマンドを入力して確認することができます (このコマンドにより現在の OpenSSL バージョンが表示されます):

```
openssl version
```

Windows では、公式の OpenSSL ソースコードからバイナリをコンパイル、または OpenSSL を含むバイナリ配布をダウンロードすることもできます。<https://www.openssl.org/community/binaries.html> を参照してください。

## 署名済みのSSL 証明書を取得する方法:

1. 秘密キーを作成します。次のOpenSSL コマンドは、2048 ビットのサイズである **flowforce.key** と呼ばれるキーを作成します (証明書機関により通常受け入れられる最低限の暗号化です):

```
openssl genrsa -out flowforce.key 2048
```

## メモ

- 秘密キーはPEM (プライバシー強化メール) フォーマットである必要があります。PEM ファイルのファイル拡張子は通常 .pem ですが、.key、.cert、.cer、または .crt であることもできます。
- FlowForce 内で秘密キーを使用可能にするには、パスワードで保護されている必要があります。[秘密キーの必要条件](#)を参照してください。
- 秘密キーは安全に保存されている必要があります。

2. 既に生成されている秘密キーのために証明書署名要求 (CSR) を生成します。SSL 証明書を購入する場合、CSR が必要になる場合があります。次を参照してください。次のOpenSSL コマンドは、CSR という名前の **myserver.csr** をキー **flowforce.key** のために作成します:

```
openssl req -new -nodes -key flowforce.key -out myserver.csr
```

プロンプトされると、機関に関する情報を入力します。例:

```
Country Name (2 letter code) [AU]: AT
State or Province Name (full name) [Some-State]: .
Locality Name (eg, city) []: Vienna
Organization Name (eg, company) [Internet Widgits Pty Ltd]: MyCompany Ltd
Organizational Unit Name (eg, section) []: IT
Common Name (eg, YOUR name) []: server.my.domain.com
Email Address []: test@example.org
```

## メモ

- フィールド 共有される名前 のためにFlowForce Server が作動するホスト マシンのFQDN (完全修飾ドメイン名) が入力されていることを確認してください。
- プロンプトされると、チャレンジパスワードフィールドをそのままおいてください。

3. 証明書機関から証明書を注文します。注文の過程で、CSR を提供する必要があります。これを行うには、**myserver.csr** を Notepad などのテキスト エディターで開き、クリップボードに内容を張り付け、オンラインの注文フォームに張り付けます。
4. 証明書機関が所属企業を検証すると、「中間」証明書と呼ばれる購入済みの証明書が提供されます。[中間証明書を準備する方法](#)で示されるように中間の証明書の内容を一つのファイルにコピーして貼り付けます。

## 概要

上記のステップに従うと、次の証明書とキーを所有している必要があります:

- **flowforce.key** - この秘密キーはFlowForce により使用される証明書に半います。

- **certificate.crt** (ファイル拡張子は異なります) - これはブラウザとFlowForce Web Server 間の接続、または FlowForce Server により公開されているサービスに接続するクライアントアプリケーションを暗号化する購入済みの証明書です。
- **intermediate.pem** - このファイルは、証明書機関から受信されたすべての中間の証明書が含まれます。

FlowForce Server、FlowForce Web Server、およびHTTP のため以下のようにSSL を有効化することができます:

- [FlowForce Web サーバーのためにSSL を有効化する方法](#)
- [FlowForce サーバーのためにSSL を有効化する方法](#)
- [FlowForce Web Server とFlowForce サーバー間にSSL を有効化する方法](#)

### 3.3.1.1 中間証明書を準備する方法

証明書機関と証明書を署名する場合、サーバーと証明書機関の間の信頼のチェーンを作成する証明書を受信します。FlowForce Server 内で証明書をすくく使用するには、すべての証明書を一つのファイルしめる証明書チェーンファイルに以下に示されるように結合する必要があります:

1. Notepad などのテキストエディターを使用して、新規のテキストファイルを作成します (**intermediary.pem** と呼ぶことしますが、他のファイル名と拡張子を選択することもできます)。
2. 各中間の証明書をテキストエディターで開き、**intermediary.pem** ファイル内にコンテンツをコピーして貼り付けます。重要な点は、証明書テキストが逆の順番でコピーされる必要がある点です (すなわち、2番目の中間証明書が最初コピーされ、プライマリ証明書は2番目にコピーされます)。例:

```
--BEGIN CERTIFICATE--
... (secondary intermediary certificate) ...
--END CERTIFICATE--
--BEGIN CERTIFICATE--
... (primary intermediary certificate) ...
--END CERTIFICATE
```

3. **intermediary.pem** ファイルを保存します。FlowForce セットアップページからこのファイルを参照する必要があります。

### 3.3.2 自己署名済みの SSL 証明書の作成方法

プライベートのネットワーク上で作動するのFlowForce Server ために、自己署名済み SSL 証明書を作成する方法をこのデモで説明します。このデモは内部で簡素化されており、生産の過程で使用されるのは最適でない場合があります。所属機関には、特定のセキュリティポリシーに関連する SSL 証明書が存在し、下記以外の SSL ツールを使用する場合があります。信頼されている証明書機関により署名される SSL 証明書を取得する情報に関しては、次を参照してください: [証明書機関により SSL 証明書を署名する方法](#)。

#### 前提条件

このサンプルは OpenSSL ツールキット (<https://www.openssl.org/>) を使用して自己署名済み証明書を生成します。OpenSSL は、オープンソースライブラリであり、コマンドラインで使用する前にコンパイルする必要があります。OpenSSL のためのコンパイルとインストール命令は、各オペレーティングシステムで異なり、このドキュメントの範囲ではありません。Linux と Mac マシン上で、OpenSSL はすでに使用できる可能性があります。それ以外の場合、コマンドラインからインストールし、更新することができます。OpenSSL が存在するか、下記のコマンドを入力して確認することができます (このコマンドにより現在の OpenSSL バージョンが表示されます):

```
openssl version
```

Windows では、公式の OpenSSL ソースコードからバイナリをコンパイル、または OpenSSL を含むバイナリ配布をダウンロードすること

もできます。<https://www.openssl.org/community/binaries.html> を参照してください。

### ルート証明書を作成する方法

1. このデモにおいて使用されるすべての証明書を補完するディレクトリを作成します (例、「C:\secure」)。これは、全ての OpenSSL コマンド (関連するファイルパス) のための作業ディレクトリです。このため、コマンドラインからのディレクトリを変更します:

```
cd C:\secure
```

2. このデモのために、OpenSSL 拡張子を使用して証明書を作成します。これを可能にするには、OpenSSL ディストリビューションの `openssl.cnf` ファイルを検索し、前のステップで作成された作業ディレクトリにコピーします。
3. ルート秘密キーを作成します。安全な環境内で生成され保管するため、ルート秘密キーが公開キー構造の最も機密な部分であることにご注意してください (このデモでは、「C:\secure」内に保管されます)。

```
openssl genrsa -aes256 -out root.key 2048
```

プロンプトされると、ルートキーを保護するためにパスワードを入力します。証明書リクエストを署名するためにこのパスワードが後に必要になります。

4. ルート証明書を作成する方法。下のコマンドは、上記で作成された秘密キーのために 3650 日間有効な自己署名済みの証明書を生成します。-config パラメータは、同じディレクトリ内の `openssl.cnf` ファイルを指していることを確認してください。-extensions パラメータは `openssl.cnf` 内で定義されている「v3\_ca」拡張子 (セクション) を参照します。

```
openssl req -config openssl.cnf -extensions v3_ca -x509 -new -nodes -key root.key -sha256 -days 3650 -out root.pem
```

プロンプトされると、所属機関の情報を入力します。例

```
Country Name (2 letter code) [AU]: AT
State or Province Name (full name) [Some-State]: .
Locality Name (eg, city) []: Vienna
Organization Name (eg, company) [Internet Widgits Pty Ltd]: MyCompany Ltd
Organizational Unit Name (eg, section) []: IT
Common Name (eg, YOUR name) []: Demo CA
Email Address []: test@example.org
```

所属機関に適用することのできる必要とされるフィールドに記入します。フィールド「共通名」に自己署名済みの証明書機関 (このデモの場合、「Demo CA」) を入力します。

### FlowForce 証明書を作成する方法

(FlowForce Server、または FlowForce Web Server、または、両方) SSL 暗号化のために使用される実際の証明書を作成することができます。次の OpenSSL コマンドは秘密キーを作成します:

```
openssl genrsa -out flowforce.key 2048
```

#### メモ

- 秘密キーは PEM (プライバシー強化メール) フォーマットである必要があります。PEM ファイルのファイル拡張子は通常 .pem ですが、.key、.cert、.cer、または .crt であることもできます。
- FlowForce 内で秘密キーを使用可能にするには、パスワードで保護されている必要があります。[秘密キーの必](#)

- [要件](#)を参照してください。
  - 秘密キーは安全に保存されている必要があります。

作業中の `openssl.cnf` ファイルを開き、次のセクションを追加します:

```
[ server_cert ]
# Extensions for server certificates (`man x509v3_config`).
basicConstraints = CA:FALSE
nsCertType = server
nsComment = "OpenSSL Generated Server Certificate"
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer:always
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth
subjectAltName=DNS:server.my.domain.com
```

FlowForce Server が作動するマシンの FQDN (完全修飾ドメイン名) に対応するように `subjectAltName` («Subject Alternative Name») が変更されていることを確認してください。このサンプルで、「server.my.domain.com」に設定されています。サブジェクトの代替名は Google Chrome 58 または以降を必要とします。それ以外の場合、自己署名済みの証明書は `NET::ERR_CERT_COMMON_NAME_INVALID` エラーを生成します (次を参照してください: <https://support.google.com/chrome/a/answer/7391219?hl=en>)。

証明書署名要求 (CSR) を下記のとおり作成します。 `-config` パラメーターは以前に編集されている `openssl.cnf` ファイルを指しています。 `-extension` パラメーターは `openssl.cnf` 内で定義されている「server\_cert」拡張子を参照しています。

```
openssl req -config openssl.cnf -extensions server_cert -new -nodes -key flowforce.key
-out flowforce.csr
```

プロンプトされると、機関に関する情報を入力します。例:

```
Country Name (2 letter code) [AU]: AT
State or Province Name (full name) [Some-State]: .
Locality Name (eg, city) []: Vienna
Organization Name (eg, company) [Internet Widgits Pty Ltd]: MyCompany Ltd
Organizational Unit Name (eg, section) []: IT
Common Name (eg, YOUR name) []: server.my.domain.com
Email Address []: test@example.org
```

#### メモ

- フィールド 共有される名前 のために FlowForce Server が作動するホストマシンの FQDN (完全修飾ドメイン名) が入力されていることを確認してください。
- プロンプトされると、チャレンジ/パスワードフィールドをそのままおいてください。

このデモのために、ルート証明書を使用して FlowForce 証明書を直接署名します。生産環境では、ルート証明書はサーバー証明書を直接署名しません。中間の証明書が代わりに使用されます。下のコマンドは前に作成されているルート証明書に対して `flowforce.csr` 証明書リクエストを署名し、`flowforce.crt` ファイル (FlowForce Server で必要とされるサーバー証明書) を作成します:

```
openssl x509 -extfile openssl.cnf -extensions server_cert -req -in flowforce.csr -CA
root.pem -CAkey root.key -CAcreateserial -out flowforce.crt -days 365 -sha256
```

## 概要

上記のステップに従うと、次の証明書とキーを所有している必要があります:

- **root.key** - 証明書機関 (CA) の秘密キーです。このファイルを安全な場所に保管します。このキーが危害を受けると、ブラウザにより信頼される証明書が使用者として生成できるようになります。
- **root.pem** - これは証明書機関の公開証明書です。FlowForce に安全にアクセスする各マシン (または、ブラウザ) の信頼されている証明書にこの証明書をインストール (インポート) する必要があります。次を参照してください! [ルート証明書のインポート](#)。
- **flowforce.key** - この秘密キーは FlowForce に使用される自己署名済みの証明書に含まれます (次のアイテムを参照してください)。
- **flowforce.crt** - これは FlowForce Server、FlowForce Web Server または、両方にも使用される自己署名済みの証明書です。

FlowForce Server、FlowForce Web Server および HTTP 接続間のために SSL を以下のように有効化することができます:

- [FlowForce Web サーバーのために SSL を有効化する方法](#)
- [FlowForce サーバーのために SSL を有効化する方法](#)
- [FlowForce Web Server と FlowForce サーバー間に SSL を有効化する方法](#)

### 3.3.2.1 ルート証明書のインポート

自身の証明書機関 (CA) を作成する場合、ルート証明書は自己署名です。ですから、デフォルトではブラウザは信頼されません。すなわち、FlowForce Server に接続するブラウザは、「このサイトは信頼されていません」などの警告を表示します。(ブラウザなどの HTTP クライアントが自己署名済みの証明書を信頼する場合は、証明書を以下のようにインポートしてください)

- ブラウザーが後者を使用する場合、オペレーティングシステムの信頼されている証明書ストア。Windows 上では、例えば Google Chrome と Microsoft Edge はオペレーティングシステムの証明書ストアを使用しますが、Mozilla Firefox は自身のストアを使用します。Linux 上では、Google Chrome と Mozilla Firefox は自身の証明書ストアを使用します (次のアイテムを参照してください)。Mac 上では、オペレーティングシステムの証明書ストア (Keychain Access) を使用します。
- ブラウザー自身の信頼されている証明書ストア。

#### メモ

- 各クライアントマシン (または、ブラウザ) のために FlowForce Server にアクセスするこのステップを実行する必要があります。
- FlowForce Web Server と FlowForce Server の間で、SSL 暗号化を有効化すると、ブラウザに証明書をインポートするだけでは十分ではありません。自身で署名済みのルート CA 証明書はオペレーティングシステムにより信頼されている必要があります。

## Linux

Linux 上では信頼される証明書をシステムの証明書ストアに以下のようにインポートすることができます。

サーバーの証明書の信頼性が確実の場合、次のステップをおこなってください。

Debian と Ubuntu では、以下のステップに従います:

1. Web サーバーの証明書ファイルを次のディレクトリにコピーします。

```
sudo cp /home/downloads/server_cert.crt /usr/local/share/ca-certificates/
```

2. 証明書を以下のように更新します

```
sudo update-ca-certificates
```

CentOS では、以下のステップに従います:

1. ca-certificates パッケージをインストールします:

```
yum install ca-certificates
```

2. 動的な証明機関構成機能を有効化します:

```
update-ca-trust enable
```

3. サーバーの証明書を次のディレクトリにコピーします:

```
cp server_cert.crt /etc/pki/ca-trust/source/anchors/
```

4. コマンドを使用します:

```
update-ca-trust extract
```

ブラウザからのみサーバーにアクセスする必要がある場合、各ブラウザストアに証明書をインポートするのみで十分です。各ブラウザにより正確な手続きは異なります。例えば、Firefox 59.0.2 では、以下を行うことができます。

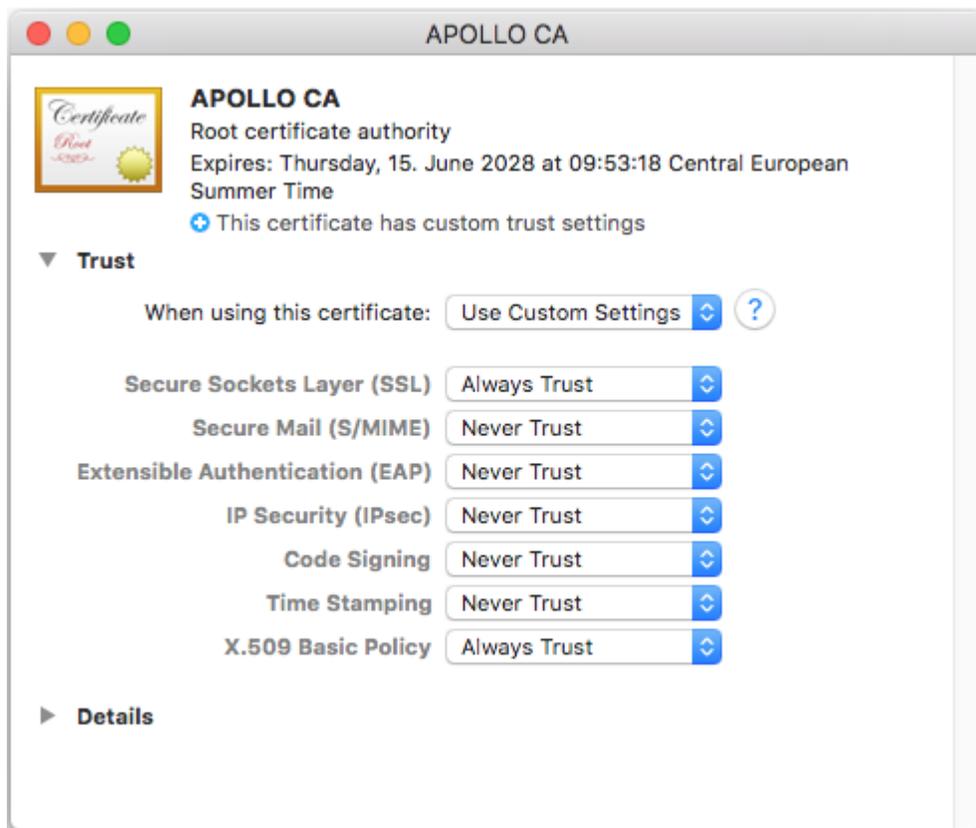
1. 「オプション | プライバシーとセキュリティ」から、「証明書を表示」をクリックします。
2. 「認証局証明書」タブで、「インポート」をクリックし、以前で作成されているルート証明書ファイルを参照します。
3. プロンプトされると「この証明書をウェブサイトの識別に使用する」を選択します。



## Mac

macOS では、信頼される証明書を Keychain Access に以下のようにインポートすることができます。

1. Keychain Access を実行します。
2. 「システム」をクリックして、「証明書」をクリックします。
3. 「ファイル」メニューから「アイテムのインポート」をクリックします。
4. 信用される証明書を参照し、「開く」をクリックします。
5. Keychain Access ノイワードをプロンプトされると入力し、「キーチェーンの変更」をクリックします。
6. 証明書をダブルクリック、「信頼」セクションを拡張し、「常に信頼する」を選択します。



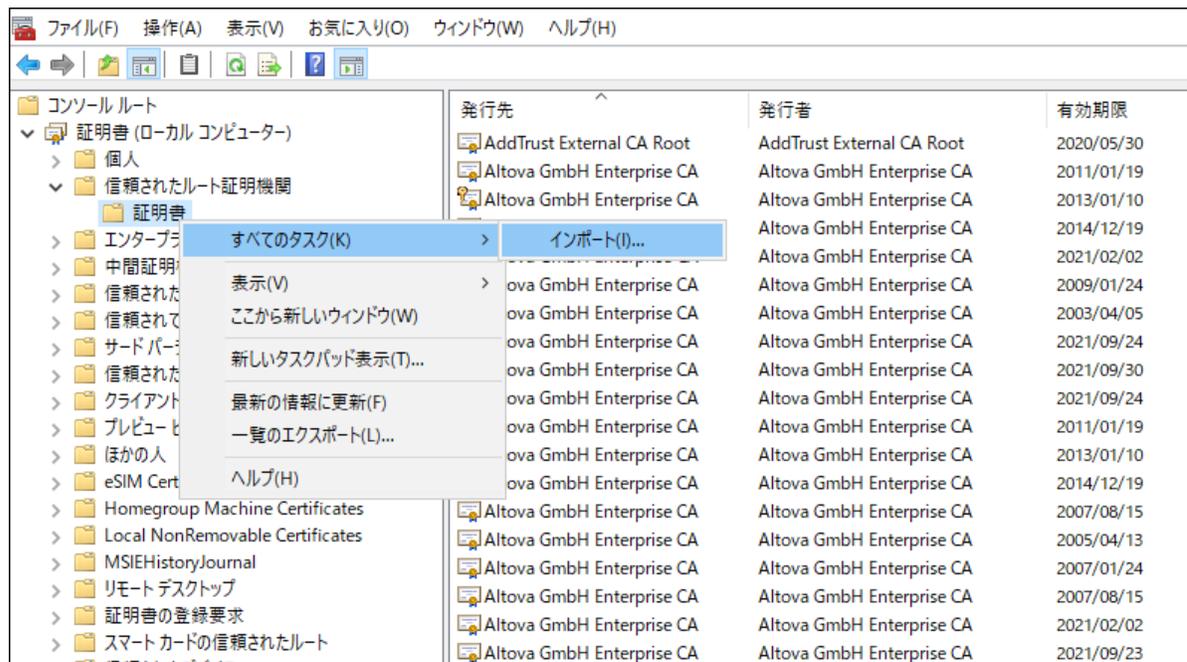
## Windows

Windows では、信頼された証明書をシステム証明書ストアに以下のように入ポートすることができます。

1. Windows 証明書ストアをコンピューターアカウントのために開きます、[Windows 証明書ストアへのアクセス](#)を参照してください。

Web サーバー証明書の認証を確認するだけの場合、次のステップを実行します。

2. 「信頼されたルート証明書機関」で「証明書」を右クリックし、「すべてのタスク | インポート」を選択し、証明書のインポートウィザードに従ってください。



詳細に関しては [https://technet.microsoft.com/en-us/library/cc754489\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/cc754489(v=ws.11).aspx) を参照してください。

### 3.3.2.2 Windows 証明書ストアにアクセスする方法

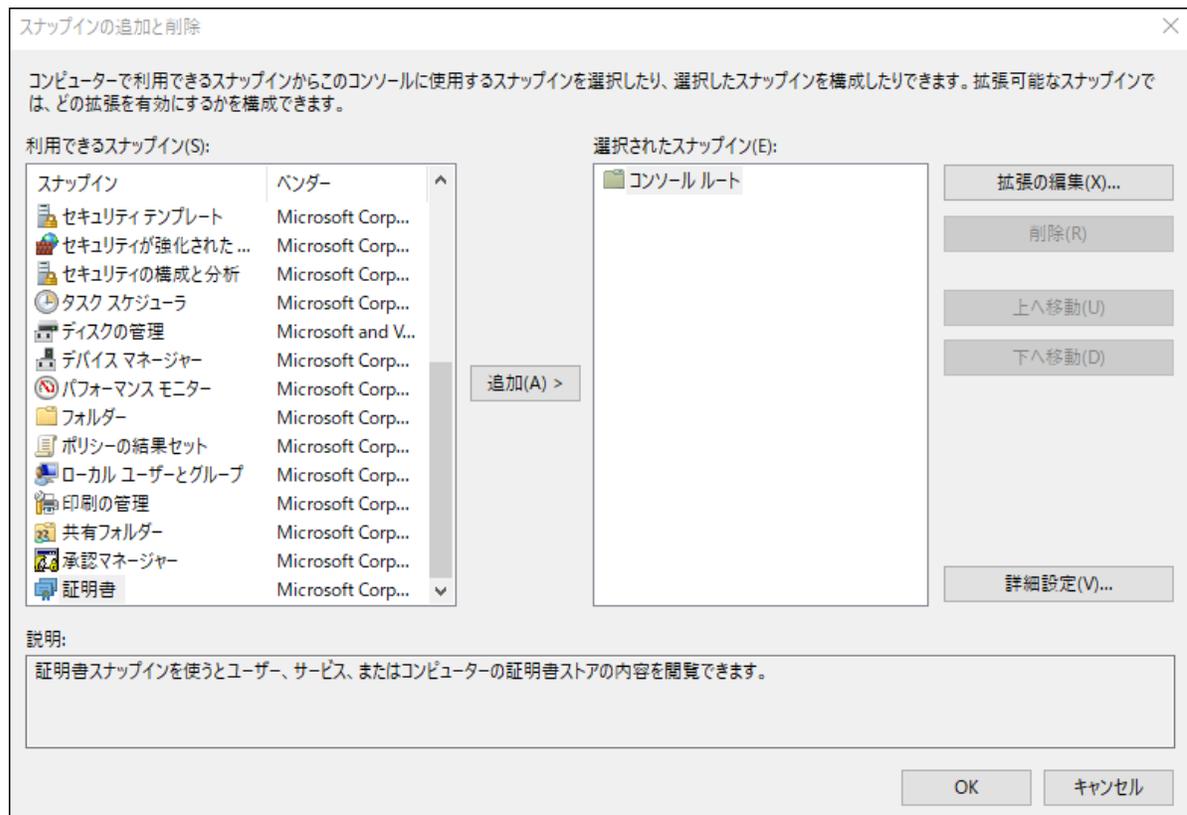
Windows 上では Microsoft 管理コンソール(MMC) スナップイン、または、Internet Explorer から証明書を管理することができます。

(現在 Windows を使用するユーザーのための)証明書 スナップインを開く方法 :

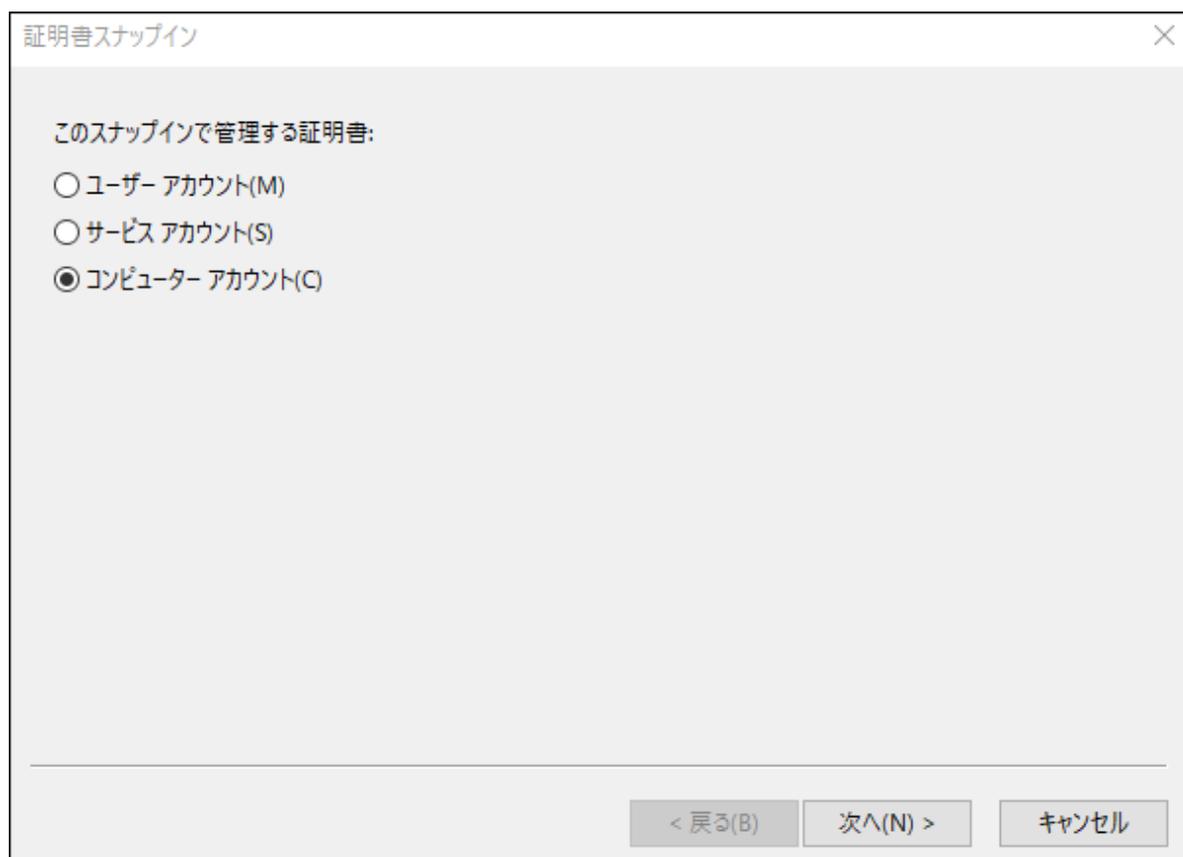
- `certmgr.msc` をコマンドラインで実行します。

(コンピューターのアカウントのために)証明書スナップインを開く方法 :

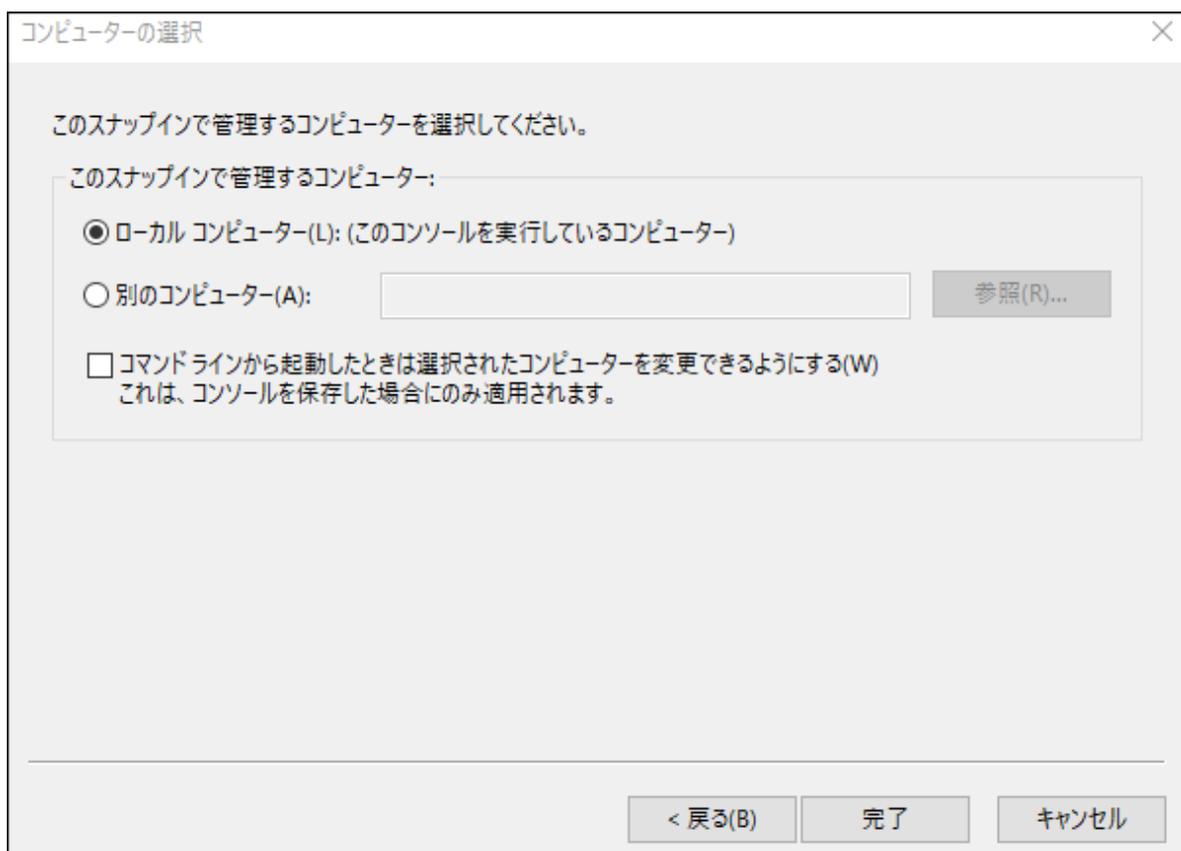
1. `mmc` をコマンドラインで実行します。
2. 「ファイル」メニューから、「スナップインの追加/削除」をクリックします。
3. 「証明書」をクリックして、「追加」をクリックします。



4. 「コンピュータのアカウント」をクリックして、「次へ」をクリックします。



5. 「ローカルコンピューター」をクリックして、「完了」をクリックします。



### 3.3.3 秘密キーの必要条件

FlowForce Server は無人で作動するため、enabling SSL の有効化は証明書の秘密キーが暗号化なしであることを必要とします。すなわち、パスワードで保護されてはいけません。それ以外の場合、FlowForce Server により使用されることできません。この理由のため、秘密キーを保管するファイルは制限されたアクセスを持ち、期間内で権限を持つ者のみがアクセス可能である必要があります。

秘密キーがパスワードにて保護されている、または暗号化なしであるかなどを確認するには、テキストエディターまたはコマンドラインで秘密キーファイルを開いて確認します。暗号化された秘密キーは以下のラインで開始します:

```
-----BEGIN RSA PRIVATE KEY-----  
Proc-Type: 4, ENCRYPTED  
DEK-Info: AES-256-CBC,DFC3FAD546517ED6336CFF72AA23F6C7
```

秘密キーの暗号化を解除する方法の一つは OpenSSL ツールキットの使用です(<https://www.openssl.org/>)。具体的には、以下のコマンドを実行して秘密キーの暗号化を解除することができます。

```
openssl rsa -in enc.key -out dec.key
```

**enc.key** が暗号化されているキーで **dec.key** がコマンドにより出力される暗号化されていないキーである箇所。例えば **ffenc.key** は暗号化されているキーで、秘密キーが保管されているディレクトリ内の以下のコマンドを実行して暗号化されていないキーを取得することができます:

```
openssl rsa -in ffenc.key -out ffdec.key
```

コマンドの実行後 **ffdec.key** ファイルは ENCRYPTED とは表記しなくなります。

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAzCCedru/oKZaSiwh6avtf9eMPix99RKpd07fWtwstkuglAdi
--
--
--
-----END RSA PRIVATE KEY-----
```

### 3.3.4 FlowForce Web サーバーのために SSL を有効化する方法

下記に FlowForce の Web 管理インターフェイスを起動する「FlowForce Web Server」サービスのために SSL を有効化する方法について説明されています。

必要条件:

- (DigiCert, Comodo など) ブラウザーにより信頼されている証明書機関により署名された秘密キーと対応する証明書が必要です。証明書機関により提供されるすべての中間証明書が必要になります。取得方法に関しては [証明書機関により SSL 証明書を署名する方法](#) または [自己署名済みの SSL 証明書の作成方法](#) を参照してください。
- 証明書ファイルと秘密キーは PEM (プライバシー強化メール) フォーマットである必要があります。PEM ファイルのファイル拡張子は通常 .pem ですが、.key、.cert、.cer、または .crt であることもできます。
- 証明書は FlowForce Server が起動するドメイン名のために発行されます。
- 証明書の秘密キーはパスワードで暗号化されてはなりません。 [秘密キーの必要条件](#) を参照してください。
- 自己署名済み証明書を作成した場合、各クライアントブラウザは自己署名済みの証明書の機関を信頼するように構成されている必要があります。 [ルート証明書をインポートする方法](#) を参照してください。

前提条件が満たされると、クライアントマシンと FlowForce Web Server 間の接続を次のように確保することができます:

1. FlowForce Server [セットアップページ](#)を開きます。
2. 「FlowForce Web Server」の下でグループ化されている設定を検索し、次を行います:
  - a. Enabled 有効化済み チェックボックスを「SSL 暗号化された接続」の下で選択します。
  - b. 「バインドアドレス」の横で、すべてのインターフェイス (0.0.0.0) を選択します。(現在のマシンからローカルにアクセス可能なだけでなく FlowForce Web Server が外部からアクセス可能なことを想定としています)。
  - c. ホスト名 とポート フィールド内で FlowForce Web Server が SSL 暗号化済み接続のためにリスニングする箇所ですト (ドメイン) 名とポートを入力します。

ホスト名 フィールド内に入力されているドメイン名は SSL 証明書の共通名に対応している必要があります。ポートは使用されてはなりません。

場合によっては「その他」を選択し、FlowForce Server が SSL 暗号化された接続をリスニングするアドレスを入力します。ホスト名を入力せずに「他の」フィールド内に IP アドレスを入力すると、この IP アドレスは SSL 証明書の共通名に対応する必要があります。

- d. 証明書と秘密キーファイルに対応するテキストボックスに入力します。

メモ

- 証明書は PEM (プライバシー強化メール) フォーマットである必要があります。PEM ファイルのファイル拡張子は通常 .pem ですが、.key、.cert、.cer、または .crt であることもできます。
- 秘密キーは PEM (プライバシー強化メール) フォーマットである必要があります。PEM ファイルのファイル拡張子は通常 .pem ですが、.key、.cert、.cer、または .crt であることもできます。
- FlowForce 内で秘密キーを使用可能にするには、パスワードで保護されている必要があります。[秘密キーの必要条件](#)を参照してください。
- 秘密キーは安全に保存されている必要があります。

- e. 証明書チェーンファイルフィールド内の証明書チェーンファイルへのパスを入力します。これは中間証明書ではなく、このフィールドを空白のままにすることができます。いくつかの中間ファイルを使用できる場合、これらを含むすべての中間ファイルを含む [チェーンファイル] に結合する必要があります。[中間ファイルの準備](#)を参照してください。
- f. 任意で、「暗号化されていない接続」の下の効化済み チェックボックスをクリアします。この操作により FlowForce Server がプレーンな HTTP により使用できなくなるため、SSL 暗号化されている接続が作動することを確認した後、このステップを実行してください。HTTP 接続を完全に無効化する代わりに、下のイメージで表示されているように、ローカルの接続を制限してください。

The screenshot shows the configuration interface for FlowForce Web Server. It is divided into two sections: 'Unencrypted Connection' and 'SSL Encrypted Connection'.

**Unencrypted Connection:**

- Enabled:
- Bind address:  Local only (127.0.0.1)  other:
- Port: 8082
- Host name:

**SSL Encrypted Connection:**

- Enabled:
- Bind address:  All interfaces (0.0.0.0)  other:
- Port: 8083
- Host name: server.example.org
- Certificate file: C:\secure\flowforceweb.crt
- Private Key file: C:\secure\flowforceweb.key
- Certificate Chain file: C:\secure\intermediate.pem

3. 設定を適用し FlowForce サービスを再起動するをクリックします。

SSL 有効化 チェックボックスを選択し 設定を適用し FlowForce サービスを再起動するを選択すると、ブラウザは ("http"ではなく "https" URL に再ダイレクトされます。

以下の点に注意してください。

- ブラウザー (または、接続するクライアント) は SSL 証明書の共通名 (CN) が FlowForce Server が実行されるドメイン名と IP アドレスに一致しない場合、警告を表示します。
- 自己署名済み証明書を使用する際、オペレーティングシステムの証明書ストア、または、ブラウザの証明書ストアに CA ルート証明書を追加していない場合、ブラウザ (または、接続するクライアント) は警告を表示します (次を参照: [ルート証明書のインストール](#))。

### 3.3.5 FlowForce サーバーのために SSL を有効化する方法

このピックでは、FlowForce ショップから HTTP(S) クライアントに作成されたサービスを公開するサービスである「FlowForce Server」サービスのために SSL を有効化することについて説明されています。ブラウザと FlowForce Web 管理インターフェイス間の SSL を有効化する場合、[FlowForce Web Server のために SSL を有効化する](#)を参照してください。

必要条件:

- (DigiCert, Comodo など) ブラウザーにより信頼されている証明書機関により署名された秘密キーと対応する証明書が必要です。証明書機関により提供されるすべての中間証明書が必要になります。取得方法に関しては、[証明書機関により SSL 証明書を署名する方法](#) または [自己署名済みの SSL 証明書の作成方法](#)を参照してください。
- 証明書ファイルと秘密キーは PEM (プライバシー強化メール) フォーマットである必要があります。PEM ファイルのファイル拡張子は通常 .pem ですが、.key、.cert、.cer、または .crt であることもできます。
- 証明書は FlowForce Server が作動するドメイン名のために発行されます。
- 証明書の秘密キーはパスワードで暗号化されてはなりません。[秘密キーの必要条件](#)を参照してください。
- 自己署名済みの証明書を作成した場合、各クライアント ブラウザーは自己署名済みの証明書の機関を信頼するように構成されている必要があります。[ルート証明書をインポートする方法](#)を参照してください。

前提条件が満たされると、クライアントマシンと FlowForce Server 間の接続を次のように確保することができます:

1. FlowForce Server [セットアップページ](#)を開きます。
2. 「FlowForce Server」の下でグループ化されている設定を検索し、次を行います:
  - a. Enabled 有効化済み チェックボックスを「SSL 暗号化された接続」の下で選択します。
  - b. 「ホストアドレス」の横で、すべてのインターフェイス (0.0.0.0) を選択します。(現在のマシンからローカルにアクセス可能なだけでなく FlowForce Server が外部からアクセス可能なことを想定としています)。
  - c. ホスト名 とポート フィールド内で FlowForce Web Server が SSL 暗号化済み接続のためにリスニングする箇所でホスト (ドメイン) 名とポートを入力します。

ホスト名 フィールド内に入力されているドメイン名は SSL 証明書の共通名に対応している必要があります。ポートは使用されてはなりません。

場合によっては、「その他」を選択し、FlowForce Server が SSL 暗号化された接続をリスニングするアドレスを入力します。ホスト名を入力せずに「他の」フィールド内に IP アドレスを入力すると、この IP アドレスは SSL 証明書の共通名に対応する必要があります。

- d. 証明書と秘密キーファイルに対応するテキストボックスに入力します。

#### メモ

- 証明書は PEM (プライバシー強化メール) フォーマットである必要があります。PEM ファイルのファイル拡張子は通常 .pem ですが、.key、.cert、.cer、または .crt であることもできます。
- 秘密キーは PEM (プライバシー強化メール) フォーマットである必要があります。PEM ファイルのファイル拡張子は通常 .pem ですが、.key、.cert、.cer、または .crt であることもできます。
- FlowForce 内で秘密キーを使用可能にするには、パスワードで保護されている必要があります。[秘密キーの必要条件](#)を参照してください。
- 秘密キーは安全に保存されている必要があります。

- e. 証明書チェーンファイルフィールド内の証明書チェーンファイルへのパスを入力します。これは中間証明書ではなく、このフィールドを空白のままにすることができます。いくつかの中間ファイルを使用できる場合、これらを含むすべての中間ファイルを含む [チェーンファイル] に結合する必要があります。[中間ファイルの準備](#)を参照してください。

- f. 任意で、「暗号化されていない接続」の下の有効化済み チェックボックスをクリックします。プレーンHTTP を介して FlowForce Server を使用できなくなります。

### FlowForce Server

#### 暗号化されていない接続

有効化済み:

バインドアドレス:  ローカルのみ (127.0.0.1)  その他:  ポート:

ホスト名:

---

#### SSL 暗号化された接続

有効化済み:

バインドアドレス:  ローカルのみ (127.0.0.1)  その他:  ポート:

ホスト名:

証明書ファイル:

秘密キーファイル:

証明書チェーンファイル:

3. 設定を適用し FlowForce サービスを再起動するをクリックします。

以下の点に注意してください。

- ブラウザー（または、接続するクライアント）はSSL 証明書の共通名（CN）がFlowForce Server が実行されるドメイン名とIP アドレスに一致しない場合、警告を表示します。
- 自己署名済み証明書を使用する際、オペレーティングシステムの証明書ストア、または、ブラウザーの証明書ストアにCA ルート証明書を追加していない場合、ブラウザー（または、接続するクライアント）は警告を表示します（次を参照: [ルート証明書のインポート](#)）。

### 3.3.6 FlowForce Web Server と FlowForce サーバー間に SSL を有効化する方法

FlowForce Web Server とFlowForce Server の.ini ファイルをそれぞれ編集し FlowForce Web Server とFlowForce Server 間のSSL を有効化することができます。 [FlowForce Server アプリケーションデータ](#) ディレクトリのデータ サブディレクトリ内で両方のファイルを見つけることができます。

flowforce.ini ファイル内で次を行います:

1. [ListenSSL] セクション内で、次のパラメーターのための値を入力します: **ssl**、**active**、**host**、**port**、例:

```
[ListenSSL]
ssl=1
active=1
host=server.my.domain.com
port=4647
```

- **ssl** と **active** パラメーターは1（有効化済み）に設定されている必要があります

- **host** は FlowForce Server に使用される SSL 証明書の共通名に一致するサーバー名 (または、適用できる場合は IP アドレス) に対応する必要があります。
  - **port** は暗号化されていない接続のために使用されるデフォルトの 4646 ポート以外のポートである必要があります。例えば、使用されていない場合、4647 に設定することができます。
2. [SSL] セクションでは FlowForce Server のために使用することのできる証明書と秘密キーへのパスを入力します。[FlowForce Server のために SSL を有効化する方法](#) 内で記載されている証明書と秘密キーのペアと同じです。例

```
[SSL]
certificate_chain=
certificate=C:\secure\flowforce.crt
private_key=C:\secure\flowforce.key
```

**flowforceweb.ini** ファイル内で次を行います:

1. 既存ではない場合、[FlowForce] と呼ばれるセクションを追加し、次のつのパラメーターのために値を入力します: ssl、host、および port。例

```
[FlowForce]
ssl=1
host=server.my.domain.com
port=4647
```

- SSL 接続を有効化するには、**ssl** パラメーターを有効化 (1 に設定) する必要があります。
- **host** は **flowforce.ini** ファイルの "[ListenSSL]" セクション内の一つと同じである必要があります。
- **host** は FlowForce Server のために使用することのできる SSL 証明書の共通名に一致するサーバー名 (または、適用できる場合は IP アドレス) に対応する必要があります。
- **port** は暗号化されていない接続のために使用されているデフォルトの 4646 ポートとは異なる必要があります。通常この値は 4647 です。

**メモ** .ini ファイルの編集が完了すると、FlowForce Server と FlowForce Web Server サービスの両方を再起動します。詳細に関しては、次を参照してください:

- [サービスの開始と停止 \(Linux\)](#)
- [サービスの開始と停止 \(macOS\)](#)
- [サービスの開始と停止 \(Windows\)](#)

## 3.4 デフォルトのタイムゾーンの設定

時間をベースにしたトリガーを使用するジョブを作成すると、適用されるタイムゾーンを指定する必要があります。ジョブの構成ページのデフォルトの設定が選択されていると、タイムゾーンをグローバルに構成することができます。

デフォルトのタイムゾーンの設定:

1. FlowForce Web 管理インターフェイスから[ログオン](#)します。
2. 「管理」をクリックします。
3. 「設定」をクリックします。
4. 「入力フォーマット」から、デフォルトのタイムゾーンを選択してください。
5. 「保存」をクリックします。

## 3.5 電子メールのパラメーターの設定

電子メールを送信するジョブを作成する場合、メールサーバーのSMTP アドレスとポート、およびSMTP 資格情報を構成する必要があります。

FlowForce は、TLS または SSL を介して、暗号化された接続を構築しようと試みます。暗号化された接続に失敗すると、FlowForce は、暗号化と無して接続を開始します。そして、SMTP サーバーが明示的に要求する場合、接続を暗号化する可能性があります。それ以外の場合、SMTP 接続は、テキスト書式のまま保持されます。

### 電子メール設定の変更:

1. FlowForce Web 管理インターフェイスから[ログオン](#)します。
2. 「管理」をクリックします。
3. 「設定」をクリックします。
4. 「SMTP サーバー」と「SMTP ポート」に、電子メールサーバーの名前とポートをそれぞれ入力します。標準のSMTP サーバーはポート 25 で接続を受け入れます。SSL/TLS プロトコールで暗号化される接続を必要とするSMTP サーバーは、通常 465 または 587 である。他のポートからの接続を受け入れます。
5. SMTP サーバーが認証を必要とする場合、「ユーザー認証」の横のプラスのアイコンをクリックし、ユーザー名とパスワードを入力します。
6. 任意で、RFC2822-compliant メールボックスアドレスの値を「デフォルトの送信元」フィールドに入力します。ここで入力されるファイルは、 [/system/mail/send](#) と  [/system/mail/send-mime](#) 関数のパラメーターのデフォルト「差出人」として使用されます。

## 3.6 ディレクトリサービスの設定

所属機関が Microsoft Active Directory または Apache Active Directory、OpenLDAP Server、Oracle Unified Directory などの LDAP 準拠ディレクトリサービスプロバイダーを使用する場合、FlowForce Server と統合することができます。FlowForce Server パースペクティブからの Directory Service プロバイダーとの統合は以下を意味します:

- ユーザーは、ドメインユーザー名とパスワードを使用して、FlowForce Server にログインすることができます。
- 管理者は、既存のドメインユーザードメイン資格情報（つまり、明示的なユーザーインポート）を使用して、FlowForce Server へのログインを許可、または、ドメインユーザーとグループを FlowForce Server に明示的にインポートすることを許可することができます（以下を参照: [ドメインユーザーとロールをインポート](#)）。どちらの場合でも、インポートされたアカウントは、FlowForce Server のユーザーアカウント管理ページで表示されます。これにより、管理者は、ビルトイン FlowForce Server アカウントと同様に、グループまたはユーザーアカウントに特権とパーミッションを追加、または削除することができます（以下を参照: [特権の作動方法](#) と [パーミッションのくみ](#)）。管理者は、また、FlowForce Server ロールをグループまたはユーザーアカウント割り当てることができます（以下を参照: [ユーザーロールを割り当てる](#)）。
- 管理者は、FlowForce Server にインポートされたドメインユーザーの名前を変更、またはパスワードを変更することはできません。
- 管理者は、FlowForce Server にインポートされたドメイングループの名前を変更、またはメンバーシップを変更することはできません。
- 管理者 FlowForce Server からインポートされたドメインアカウントを削除することができます。これは、¥ドメインからアカウントを削除することは無く、関連した特権を変更することはありません。
- インポートされたドメインアカウントが、割り当てられた FlowForce Server 特権とパーミッションを保有する場合、これは、特権レポートに表示されます（以下を参照: [特権レポートの閲覧](#)）。

ディレクトリサービスの設定を変更する方法

1. FlowForce Web 管理インターフェイスから [ログイン](#) します。
2. 「管理」をクリック、「設定」をクリックします。

### ディレクトリサービス

有効化

次を使用して接続:  Active Directory

ライトウェイトディレクトリアクセスプロトコル (LDAP)  ホスト:

ホスト名、ドメイン、または、IP アドレス。カスタムポートを以下で指定することができます。:389

ユーザー:

'name@domain.com' または 'cn=name,dc=domain,dc=com' の書式のユーザー名

パスワード:

SSL:

SSL が作動するには、LDAP サーバーの CA 証明書が FlowForce サーバーが存在するマシン上にインストールされている必要があります

ドメインユーザーにログインを許可

使用することのできる設定は、以下のとおりです:

### 有効化

FlowForce Server にドメインユーザー名とパスワードを使用してユーザーがログインできるようにこのチェックボックスを選択します。このチェックボックスをチェックすると、以下に説明されているとおり、**Active Directory** または **ライトウェイトディレクトリアクセスプロトコル (LDAP)** オプションを選択する必要があります。

ライトウェイト ディレクトリアクセスプロトコル(LDAP) オプションを選択した場合、(ユーザー名、パスワードなどの) 接続の詳細が正確であることを確認してください。「保存」をクリックすると、FlowForce は指定されたLDAP サーバーと通信を試み、接続の詳細が有効でない場合、エラーが表示されます。FlowForce Server のLDAP サーバーへの接続が成功しない限り、LDAP 設定を保存できない事に注意してください。

**Active Directory** オプションを選択した場合、FlowForce Server が作動するマシンは Active Directory により管理されているマシンの一部である必要があります。

ディレクトリサービスの認証を有効化後、FlowForce Server ログインページで「ログイン」と呼ばれる追加ドロップダウンリストが表示されるようになります。「ログイン」ドロップダウンリストによりユーザーが、次のアイテムを含む認証オプションを選択することができます：

- **直接** これはデフォルトのFlowForce Server 認証オプションです。
- **[特定のWindowsドメイン]** FlowForce Server が作動するマシンに対して適用できます。

次も参照してください！ [FlowForce Server へのログイン](#)。

## 接続

**Active Directory** を選択して、Active Directory との直接の統合を有効化します。これはFlowForce Server がWindows上で作動中のみ適用することができます。

ライトウェイト ディレクトリアクセスプロトコル(LDAP) を選択して、LDAP に準拠するディレクトリサービスとの統合を有効化します。詳細を以下の通り入力します：

- **ホスト** - LDAP サーバーのホスト名、ドメイン名、またはIP アドレスを入力します。ポート番号を追加する場合は、ポート番号の後にコロン文字を追加します。例えば、**somehost:10389**
- **ユーザー** - ディレクトリサービスを制御する管理者の権利を持つユーザー名を入力します。ユーザー名は「Distinguished-Name」(例えば、cn=name,dc=domain,dc=com) の書式、またはActive Directory のみに適用される「ユーザープリンシパル名」(例えば、user@some.domain.com) 書式を使用することができます。メモ 他のLDAP サービスに関しては「識別名」書式を使用します。
- **パスワード** - ユーザーのパスワード。メモ パスワードの入力を複数回間違えると、LDAP サーバーはアカウントをロックする場合があります。この場合、継続する前にアカウントがロックされていないことを確認してください。
- **SSL の使用** LDAP サーバーがクライアントからSSL 暗号化済みの接続を受け入れるように構成されている場合のみこのチェックボックスを選択してください。このオプションを選択する場合、安全な接続のためにLDAP サーバーにより使用されているポート番号(通常ポート **636**) に変更してください。所属機関が両方のマシン上で信頼されるルート証明書を既に使用する場合、追加の構成手順は存在しません。それ以外の場合、LDAP サーバーのルート(CA)証明書は、FlowForce Server が作動するマシンにインストールされている必要があります。
  - LDAP サーバーが存在するマシン上で、信頼されている証明書ストアからルート証明書をエクスポートします。使用中のオペレーティングシステム固有のツールを使用します(例えば、Windows上では[証明書スナップイン](#)を使用します)。
  - FlowForce Server が存在するマシン上では、[ルート証明書のインポート](#)で説明されている通り、信頼されている証明書ストアから証明書をインポートします。

一部の場合、LDAP サーバーは特定の標準に一致しない任意のスキーマが存在します。FlowForce Server が使用中のLDAP プロバイダーのスキーマを検知できない場合、「ディレクトリサービスは無効なLDAP スキーマを検知しました。類似したエラーが表示されます。この場合、**directoryservice.cfg** ファイルをFlowForce Server 実行可能ファイルと同じディレクトリにコピーします。このファイルが存在すると、FlowForce Server はプロバイダーを自動的に検知しようとしません。

## 全てのドメインユーザーログインを許可

ドメイン資格情報を使用してユーザーがFlowForce に最初ログインするとユーザーのドメインアカウントがFlowForce ユーザーデータベースにインポートされる場合、このチェックボックスを選択します。

このオプションが無効化されていると、管理者によりドメインアカウントが既にFlowForce Server にインポートされている場合、ドメインユーザーはFlowForce Server にログインすることができます。次も参照してください！ [ドメインとユーザーロールのインポート](#)。

### デフォルトのログインドメイン

有効化 チェックボックスが選択され設定が保存された後、このオプションは表示されます。

ドロップダウンリストは、マシンがメンバーのすべてのドメインを表示しています。ディレクトリサービス認証が有効化されていると(上記の最初のオプションを確認してください) FlowForce ログインページ内のユーザーがドメインのリストを確認することができます。

ドメインがFlowForce Server 認証ページ内のログインドロップダウンリスト内のデフォルトの選択として選択される場合、「ドメインのログインをデフォルトとする」チェックボックスを選択してください。

「ドメインのログインをデフォルトとする」チェックボックスをクリアすると、ビルトインFlowForce Server 認証(「Directly」)がデフォルトの選択です。

## 3.7 ログ設定

FlowForce Server は(ジョブ出力イベント、変更イベントの構成、エラーなどの)多種のイベントと発生時刻を登録するログインメカニズムを提供します。専用のページから ログイベントを確認することができます。[ジョブログの確認](#) を参照してください。ログイベントは FlowForce Server 内部データベースのサイズを大幅に増量することができることに注意してください。このために `archive-log` または `truncate-log /system/maintenance` 関数を使用して、ログは定期的に達成され、クリーンアップされる必要があります。ログを妥当な制限内のサイズに収める手助けをする他の設定を使用することができ、このページの下の部分で説明されています。

FlowForce Server 内でのログは2つの種類が存在します:

1. 手動の介入を全く必要としないデフォルトのシステムログ。この種類のログはすべてのイベントを登録しますが、更に良い状態のシステムの安定度とパフォーマンスのために各ログレコードを一定のミットまで保持します。システムログが十分な詳細のレベルを提供しない場合、または(ステップ内の)パラメータ値などの ログエントリが切り捨てられている場合、明示的なログを下記の通り選択することができます。
2. ジョブ構成ページから強制することのできるオプションの(明示的な) ログ。ログ構成ページは、ログ内でトラックする各パラメータの横にあり、オプションで有効化することのできる  ボタンを提供します。これを行うことで、ジョブが実行されると、そのパラメータのフル値がログされます。更に、その関数を明示的にログするために、`log` 式関数内で FlowForce 関数を埋め込むことができます。また、これは式を切り捨てることなくフルログします。FlowForce Server は明示的なログの結果としてログされるエントリのサイズを制限しません。

### ログの制限

デフォルトのログサイズを高い値に変更すると、システムの安定度とパフォーマンスに影響を及ぼす可能性があります。

デフォルトのログサイズを高い値に変更すると、システムの安定度とパフォーマンスに影響を及ぼす可能性があります。ですから、このオプションの使用には注意してください。明示的なログを使用するための奨励されるアプローチは上記の通りです。

ログエントリのデフォルトのサイズをビューまたは変更する方法

1. FlowForce Web 管理インターフェイスから[ログオン](#)します。
2. 「管理者 | 設定」に移動して、「ログのミット」の下でグループ分けされているパラメータを監視します。

### Logging limits

String type limits:

Default string value logging limit:  characters

List type limits:

Default list value logging limit:  items

Recurse into sublists:  levels

ログ済みのエントリは2つの種類が存在することに注意してください。文字列の型とリストの型。結果、各型のサイズを管理するための2つのパラメータが存在します。

デフォルトの文字列の値ログミット

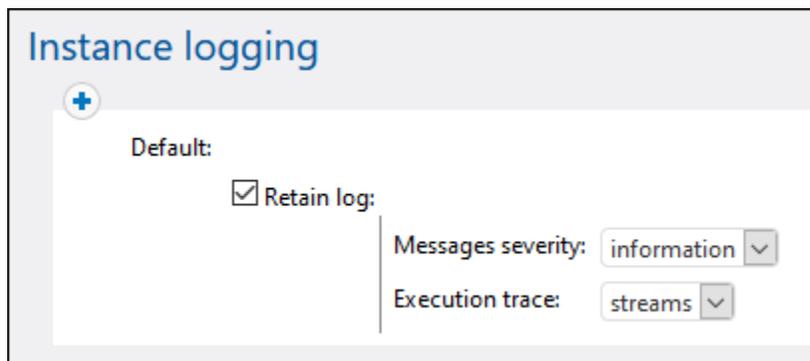
型「文字列」であるログエントリのデフォルトの長さを指定します。ログエントリがこの値を超すと、ファイル名などの長い任意の値が切り捨てられます。

|                  |  |
|------------------|--|
| デフォルトのリストの値ログミット | 上記同様、型「リスト」であるログエントリに適用されます。   |
| サブリストに再帰する       | この設定は他のリストを子として含むリストを操作するジョブに影響を与えます。この値を設定して、FlowForce にNレベルの深さをログインのため命令します。 |

### インスタンスのログ

インスタンスログのセクションは特に [インスタンスログ](#) ページ内でレポートされる情報のレベルに影響を与えます。

ログされるメッセージは以下の順序の需要度が存在します (低度から高度へ): 情報、警告、エラー [インスタンスのログ] のパラメータは重要度に従い特定のメッセージのログをスキップすることを不可能にします。FlowForce Server により保管されるトレース情報の量を構成、およびログの保管を無効化することができます。下のイメージはデフォルトの設定を表示しています。



ログ保持 チェックボックスをクリアすると、[インスタンスログ](#) ページで情報がレポートされません。

メッセージの重要度 オプションは保持されるメッセージを指定します:

|     |   |
|-----|---|
| 無し  | メッセージは保持されません                             |
| エラー | エラーと重要なメッセージを保持                           |
| 警告  | エラー、重要なメッセージ、および警告を保持                     |
| 情報  | エラー、重要なメッセージ、警告、情報メッセージを保持                |
| すべて | これは最も詳細なオプションです。重要度に関わらず、全てのメッセージが保管されます。 |

トレースの実行 パラメータは保管されるトレーシングの詳細の量を指定します:

|       |                         |
|-------|-------------------------|
| 無し    | トレーシングの情報は保管されません       |
| ストリーム | ストリームを保持し、トレースを除外します    |
| トレース  | トレースを保持し、ストリームを除外します    |
| フル    | トレース情報の可能なすべてのレベルを保持します |

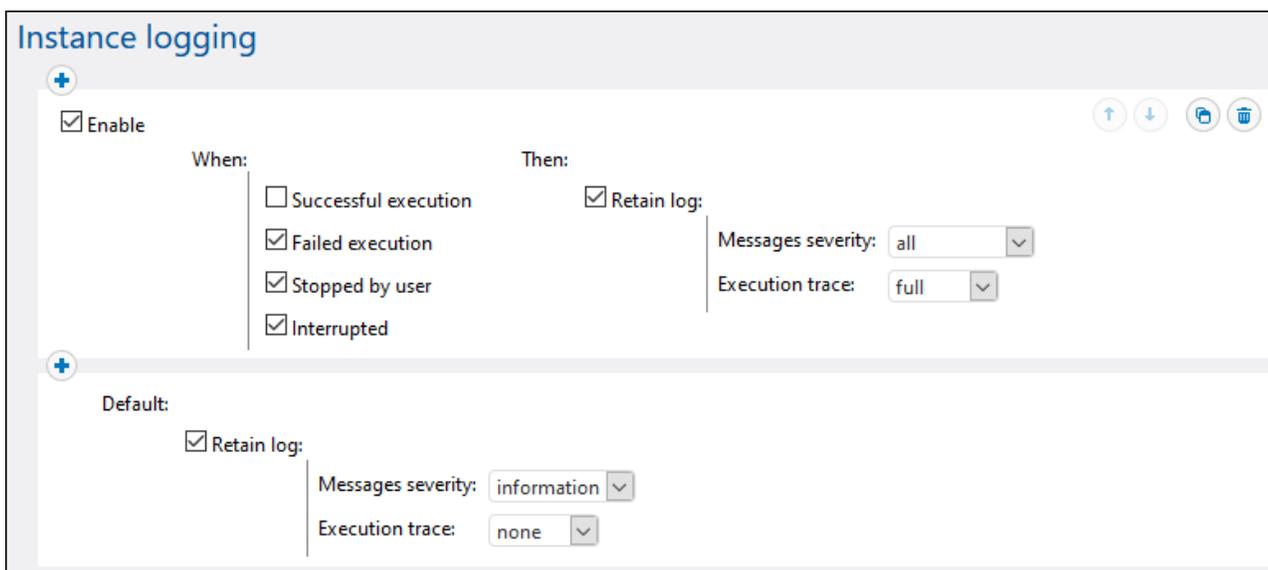
## ルール

上記で説明されている「インスタンスログ」の設定はルールを構成します。 ボタンをクリックしてデフォルトに加えカスタムルールを作成することができます。以下のようなジョブの結果をベースとしてこれによりルールを条件づけて適用することが可能となります。

|           |  |
|-----------|--|
| 実行の成功     | ジョブの結果は成功としてみなされます。  |
| 実行の失敗     | ジョブの実行に失敗しました。   |
| ユーザーにより停止 | ユーザーアクションによりジョブは停止されました。 <a href="#">ジョブの停止</a> を参照してください。 |
| 中断済み      | ジョブの完了前にサービスが中断されました。またはFlowForce Server クラッシュしました。        |

このページで定義済みのルールは上から下へ評価されます。ジョブの結果が上記の結果のどれかに一致する場合、ルールは一致します。最初に一致するルールが優先されます。

例えば、上記の構成はジョブの実行に成功しない場合フルメッセージログを保持します。すなわち、結果が[実行の失敗]、[ユーザーにより停止]、および[中断済み]の場合最初のルールがトリガーされます。実行に成功した場合、デフォルトのルールが代わりにトリガーされ、ログメッセージは保持されますが、トレーシング情報を使用することはできません。



カスタムルールをデフォルトのルールの後ではなく、*前*のみ追加することができます。ルールの順序を変更するには上  と下  ボタンを使用してください。これらのボタンはそれぞれ以上のルールが存在する場合のみ有効化されます。

以前のルールが一致しなく、カスタムルールを定義する場合、全てをキャッチするフィルターとしてデフォルトのルールを使用することが奨励されます。

### プロジェクトレベルでのログルール

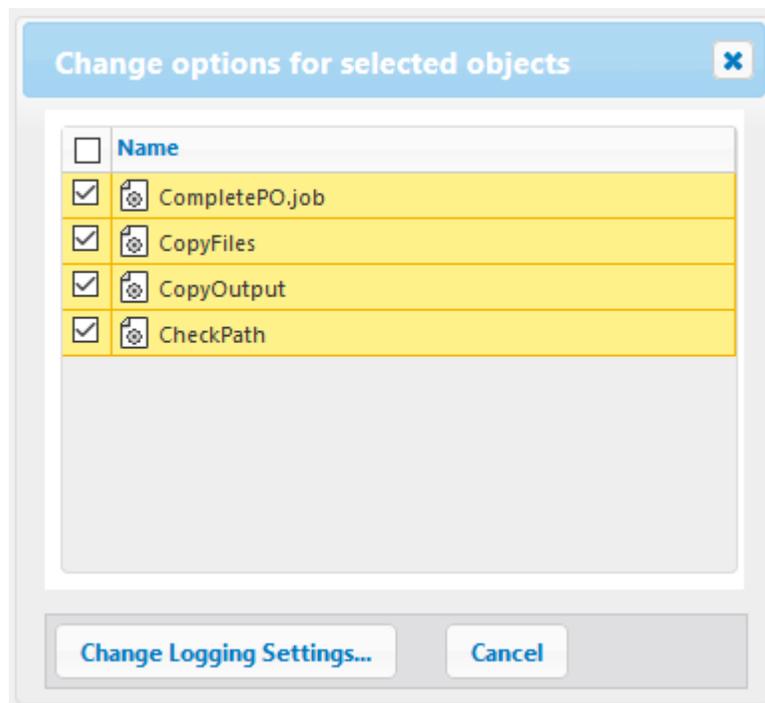
アプリケーションレベルでグローバルではなく特定のFlowForce Server ジョブのためのログルールを作成することができます。サブジョブを持つジョブのためのルールを作成すると、ルールはすべてのサブジョブにも適用されます。

## ジョブ構成ページを開く方法:

1. ジョブ構成ページが開かれます。
2. ログ設定 ボタンをジョブ構成ページ内でクリックします。
3. ログルールの新規インスタンスをクリックします。

## 複数のジョブのためにログルールを設定する方法:

1. 構成 をクリックしコンテナーを開きます。
2. 1つまたは複数のジョブ(またはコンテナー全体)を選択し選択されたジョブのためのログ設定をクリックします。必要であれば選択を詳細設定することのできるダイアログボックスが表示されます:



3. ログ設定の変更をクリックします。
4. ログルールの新規インスタンスをクリックします。

### Instance Logging Settings for Objects

| <input type="checkbox"/> Name ↕                                     | Type |  |
|---|------|--|
| <input checked="" type="checkbox"/> /public/Examples/CheckPath      | job  |  |
| <input checked="" type="checkbox"/> /public/Examples/CompletePO.job | job  |  |
| <input checked="" type="checkbox"/> /public/Examples/CopyFiles      | job  |  |
| <input checked="" type="checkbox"/> /public/Examples/CopyOutput     | job  |  |

#### Logging Rules:

Enable

When:

- Successful execution
- Failed execution
- Stopped by user
- Interrupted

Then:

- Retain log:

Messages severity:

Execution trace:

[new Instance Logging rule](#)

ログの構成設定はルール内のセクション内で説明されている設定と同様に作動します。

オブジェクトレベルとアプリケーションレベルの両方でログルールを定義した場合、優先順序は以下のよう確立されます:

- オブジェクトレベルで定義されているログルールが最初にチェックされます。
- このレベルで見つかった一致が存在する場合、ルールは適用されアプリケーションレベルでのルールはチェックされません。
- このレベルで一致が見つからない場合、アプリケーションレベルでの一致がチェックされます。

## 3.8 サービスの開始と停止 (Linux)

FlowForce Server ソリューションは2つのサービスにより構成されています:

1. flowforcewebserver
2. flowforceserver

flowforcewebserver サービスを停止するため以下のコマンドを実行します。flowforceserver サービスを開始、または停止するには、下のコマンド内でflowforcewebserver をflowforceserver と置き換えます。

FlowForce Web Server サービスの開始方法:

```
sudo systemctl start flowforcewebserver
```

FlowForce Web サービスの停止方法:

```
sudo systemctl stop flowforcewebserver
```

サービスが作動中か確認するには、次のコマンドを実行してください(flowforcewebserver またはflowforceserver と servicename を置き換えます)。

```
sudo service servicename status
```

## 3.9 サービスの開始と停止 (macOS)

FlowForce Server サービスの開始:

```
sudo launchctl load /Library/LaunchDaemons/com.altova.FlowForceServer.plist
```

FlowForce Web Server サービスの開始:

```
sudo launchctl load /Library/LaunchDaemons/com.altova.FlowForceWebServer.plist
```

FlowForce Server サービスの停止:

```
sudo launchctl unload /Library/LaunchDaemons/com.altova.FlowForceServer.plist
```

FlowForce Web Server サービスの停止:

```
sudo launchctl unload /Library/LaunchDaemons/com.altova.FlowForceWebServer.plist
```

## 3.10 サービスの開始と停止 (Windows)

デフォルトでFlowForce Server サービスは自動的にWindows が開始する際に開始されます。サービスを手動で管理する必要がある場合、以下の命令に従います。

FlowForce Server サービスの開始:

- システム通知エリアの ServiceController アイコン(  ) をクリックします。そして、「Altova FlowForce Web > Setup」を選択します。

FlowForce Web Server サービスの開始:

- システム通知エリアの ServiceController アイコン(  ) をクリックします。そして、「Altova FlowForce Web > Start service」を選択します。

FlowForce Server サービスの停止:

- システム通知エリアの ServiceController アイコン(  ) をクリックします。そして、「Altova FlowForce Server > Stop service」を選択します。

FlowForce Web Server サービスの停止:

- システム通知エリアの ServiceController アイコン(  ) をクリックします。そして、「Altova FlowForce Web > Stop service」を選択します。

Microsoft 管理コンソールを使用してFlowForce Server サービスを開始することもできます(コントロールパネル > 管理ツール > サービス)。

### 3.11 FlowForce Server アプリケーションデータ

このセクションでは、FlowForce Server アプリケーションデータディレクトリのコンテンツについて説明されています。この情報は、手動で異なる FlowForce Server バージョンにデータを移行する場合、または、.ini ファイルを編集することにより FlowForce Server 構成設定を変更する場合、役に立ちます。

アプリケーションデータディレクトリは、FlowForce Server と、ジョブ、トリガー、システム関数、サーバーログ、他のファイルなど、そのユーザーにより生成されたデータを保管します。

アプリケーションデータディレクトリは、複数の.ini スタイル構成ファイルも含んでいます。管理者は、[Web 管理インターフェイス](#)、[セットアップページ](#)または、[コマンドラインインターフェイス](#)から設定を変更する代替として、.ini 構成ファイルをテキストエディターで編集することができます。

アプリケーションデータディレクトリのパスは、オペレーティングシステムとプラットフォームにより以下のように異なります。

|         |   |
|---------|---|
| Linux   | /var/opt/Altova/FlowForceServer2021       |
| macOS   | /var/Altova/FlowForceServer2021           |
| Windows | C:\ProgramData\Altova\FlowForceServer2021 |

下のテーブルは、アプリケーションデータディレクトリ内のメインのファイルとフォルダのリストを表しています。

|  |  |
|--|--|
| <b>flowforceserver.ini</b>                         | FlowForce Server のグローバル構成設定を保管します。(現在サーバーログとエラーメッセージで使用されている言語です)。   |
| <b>flowforceweb.ini</b>                            | FlowForce Web Server のグローバル構成設定を保管します。(Web 管理インターフェイスの言語)。<br><br>メモ (下で説明されているとお) <b>data</b> ディレクトリに保存されている <b>flowforceweb.ini</b> ファイルと区別してください!                                       |
| <b>data/cache.db</b>                               | このデータベースファイルは、FlowForce のキャッシュ機能に関連したデータを保管します(以下を参照: <a href="#">ジョブの結果をキャッシュする</a> )。  |
| <b>data/state.db</b>                               | このデータベースファイルは、FlowForce の可変の状態(つまり、構成されていない状態)を保管します。  |
| <b>data/ffweb.log</b><br><b>data/flowforce.log</b> | これらのファイルは、FlowForce Web Server と FlowForce Server からのログメッセージをそれぞれ保管します。これは Windows と macOS システムに対して適用されます。<br><br>Debian、Ubuntu および CentOS または以降に対しては、ログはシステムログに書き込まれます (/var/log/syslog)。 |
| <b>data/flowforce.db</b>                           | このデータベースファイルは、FlowForce Server オブジェクトシステム、アクティブなデータ、アクティブなジョブ、ルール、およびその他を保管します。  |
| <b>data/flowforcelog.db</b>                        | このデータベースファイルは、FlowForce Server ログを保管します。   |
| <b>data/flowforce.ini</b>                          | この構成ファイルは、FlowForce Server のポートとリスニングインターフェイスを定義します。 <a href="#">ファイルの参照の構成</a> も参照してください。   |

|                              |   |
|------------------------------|---|
| <b>data/flowforceweb.ini</b> | この構成ファイルは、FlowForce Web Server のポートとUIのインターフェイスを定義します。 <a href="#">ファイルの参照の構成</a> も参照してください。  |
| <b>data/files</b>            | デプロイされた機能に関連したファイルを保管します。   |
| <b>data/logs</b>             | ジョブ実行ステップからのキャプチャされた出力を含みます。  |
| <b>data/tmp</b>              | 一時ファイルを保管します。   |
| <b>data/tools</b>            | <p>(MapForce Server および StyleVision Server などの他の Altova サーバ製品が FlowForce Server と共にインストールされる場合、このディレクトリは FlowForce Server がこれらの製品作業を可能にする .tool ファイルを保管します。</p> <p>このディレクトリが空の場合、各製品のインストールディレクトリから各製品のツールファイルを手動でコピーすることができます。</p> <p>MapForce マッピング、および StyleVision 変換の実行に必要な .tool ファイルを環境変数を設定するために編集する必要がある可能性があります (以下を参照 <a href="#">環境変数の設定</a>)。</p> |

## 3.12 FlowForce Server のローカライズ

FlowForce Server は以下の言語へのサポートを搭載しています: 英語、フランス語、ドイツ語、スペイン語、および日本語。これらの言語のローカライズされたバージョンを作成する必要はありません。これら4つの言語をデフォルトの言語として設定する場合は、FlowForce Server の [setdeflang](#) コマンドを使用してください。

FlowForce Server のローカライズされたバージョンを作成する:

1. [exportresourcestrings](#) コマンドを使用してリソース文字列を含むXML ファイルを生成します。
2. リソース文字列をターゲット言語に翻訳します。リソース文字列は、XML ファイル内の `<string>` 要素のコンテンツです。  
{option} または {product} などの、中かっこ内の変数は翻訳しないでください。
3. 翻訳されたXML ファイルからローカライズされたDLL ファイルを生成するには、[Altova サポート](#) に連絡を取ってください。
4. Altova サポート (<https://www.altova.com/support>) からローカライズされたDLL ファイルを受信すると、`<INSTALLATION FOLDER>\FlowForceServer2021\bin` フォルダにDLL を保存してください。DLL ファイルは、以下のフォームを使用します: `FlowForceServer2021_lc.dll`。名前の `_lc` 部分は、言語コードを含んでいます。例えば、`FlowForceServer2021_de.dll` では、`de` パートがドイツ語 (Deutsch) のための言語コードです。
5. 使用する FlowForce Server アプリとして、ローカライズされたDLL ファイルを設定するには、[setdeflang](#) コマンドを使用します。setdeflang コマンドの引数としてDLL 名の一部である言語コードを使用します。

### 3.13 バックアップと復元

FlowForce Server 内のデータをバックアップするには2つの方法があります：

1. **Web 管理インターフェイス**。この種類のバックアップは構成データのみが含まれます。ジョブ、資格情報、デプロイ済みの MapForce マッピングまたは StyleVision 変換、リソース。アプリケーション設定、またはユーザーは含まれません。相対するオブジェクト上でパーミッションが許可する場合、FlowForce Server ユーザーは構成データのインポートまたはエクスポートを実行することができます。
2. **アプリケーションデータディレクトリの管理/バックアップ**。このアプローチは FlowForce Server がインストールされているマシン上の FlowForce Server [アプリケーションデータディレクトリ](#) へのアクセスが必要となります。アプリケーションデータディレクトリ前のドットからのすべてのデータが含まれており、ユーザーと (Active Directory などの) Directory Service からインポートされたユーザーとロールを含む。ロールおよび電子メール、または、LDAP サーバー設定、パスワードポリシーなどのアプリケーションレベルの設定が含まれます。

**メモ** このピックでは (ジョブ、または FlowForce リソース、またはローカルファイルベースのデータベースに対して入力/出力であるファイルまたはディレクトリなど) FlowForce Server の外部のデータのバックアップと復元については説明されていません。このデータを個別にバックアップする必要があります。ベストプラクティスとして、より簡単なバックアップと管理のために、可能であれば、このような外部データを全て同じディレクトリに保管してください。

#### ヒント

データを新規のマシンに移行、または、バックアップから復元する場合、以下のヒントを考慮してデータのポータビリティを保証してください！

- LicenseServer がいわゆる「フェールオーバー」サーバーと呼ばれるフォールバックの番目のサーバーを構成することが奨励されます。詳細に関しては、LicenseServer ドキュメント (<https://www.altova.com/documentation>) を参照してください。
- 全てのジョブは (インラインではない) スタンドアロン [資格情報](#) を使用することが奨励されます。そのオペレーティングシステムにリンクされているユーザー資格情報をマッピングするジョブ内のローカル (インライン) 資格情報を使用する場合、新規のサーバーマシン上で変換する必要があります。反対に、スタンドアロン資格情報を使用する場合、新しいサーバーマシン上でスタンドアロン資格情報を編集する必要があります。
- MapForce からデプロイされたマッピング関数を実行する場合、絶対的なリファレンスの代わりに [リソース](#) を使用してファイル、フォルダー、パスとデータベースを参照することを考慮してください。
- FlowForce Server 内で直接ユーザーとロールを作成し管理する代替として、Windows Active Directory、または Directory Services のためのサポート付きの他の LDAP Server を使用してください。詳細に関しては、[ディレクトリサービス設定の変更](#) を参照してください。

#### Web 管理インターフェイスからの部分的なバックアップ

選択されたオブジェクトをバックアップする場合、FlowForce Web 管理インターフェイスにログインして、「エクスポート」機能を使用します。データを復元する場合は、インポート 機能を使用してください。詳細に関しては、[構成データのインポートとエクスポート](#) を参照してください。

**メモ** エクスポートされるデータと同じバージョン以降のバージョンである FlowForce Server インスタンスに構成データをインポートすることができます。構成データを FlowForce Server の前のバージョンにインポートすることは成功するかもしれませんが回避されるべきです。

#### 全ての FlowForce アプリケーションデータのバックアップ

全てのアプリケーションデータのバックアップは基本的に FlowForce Server データベース (DATADIR) のコピーを後に必要に応じて安全な場所に作成することにより構成されます。

下の命令内では、DATADIR は以下のディレクトリを参照します：

- Linux: `/var/opt/Altova/FlowForceServer2021/data`
- macOS: `/var/Altova/FlowForceServer2021/data`
- Windows: `C:\ProgramData\Altova\FlowForceServer2021\data`

時間とディスクのスペースを節約するために、DATADIR デレトリを可能な限りコンパクトにまとめます。以下の追加ステップを実際のバックアップの前に行い、節約を行います。

1. ビルトインの `archive-log` 関数を実行するジョブを作成し、古いログレコードをアーカイブ化します。
2. ビルトインの `truncate-log` 関数を実行するジョブを作成し、古いログレコードを削除します。
3. ビルトインの `cleanup-files` 関数を実行するジョブを作成し、使用されていないファイルを削除します。
4. `compactdb` を使用して FlowForce Server 実行可能ファイルを実行します。

実際のバックアップステップを続行することができます。

1. FlowForce Server と FlowForce Web Server サービスを停止します。Linux、macOS、と Windows のための命令を参照してください。
2. DATADIR のコピーを作成し、安全なデレトリにコピーします (異なるマシンまたはディスクが良いでしょう)。慣例では、このコピーを以降の手順で `DATADIR_BACKUP` と呼びます。

DATADIR 内の `private.db` ファイルは、パスワード秘密キーなどの機密な情報を含んでいます。バックアップが安全な場所に保管されていることを保証してください。

## データの復元

DATADIR\_BACKUP が同じバージョンで現在作動している FlowForce Server と同じマシン上の場合、以下を簡単に復元することができます。

1. FlowForce Server サービスが作動中の場合、停止してください。Linux、macOS、と Windows のための命令を参照してください。
2. DATADIR を例えば、`temp_data` に名前を変更してください。
3. DATADIR\_BACKUP を DATADIR にコピーしてください。
4. FlowForce Web Server と FlowForce Server サービスを開始します。

古いデータベースバージョンを有する他のマシンからのバックアップも復元することができます。FlowForce データを新規のサーバーに移行する場合、または、ハードウェアの障害が発生した場合などこの手順は役に立つ可能性があります。

## メモ

- 同じまたは異なるオペレーティングシステムで作動するマシンで復元を実行することができます。後者の場合、ジョブ内で使用されているすべての又は新規のオペレーティングシステムで有効でない場合があります。この場合、更新を手動で行う必要があります。
- オペレーティングシステムユーザーアカウントの操作に関連する資格情報 (「ジョブ実行のための使用方法を許可する」オプションが有効化されている個所での資格情報) は新規のマシンで有効ではない場合があります。この場合手動で更新される必要があります。

## 新規の FlowForce Server インストールまたはバージョンに復元する方法

1. FlowForce Server をインストールして、適応できる場合、以下もインストールします: MapForce Server, StyleVision Server および RaptorXML Server。LicenseServer もインストールする場合、FlowForce Server インストールの一部として行います (Windows のみ実行可能です)。他のプラットフォームでは、LicenseServer を個別にインストールする必要があります。

- LicenseServer Web 管理インターフェイスログインし、古いマシンから登録されている製品すべてを登録解除します。次に LicenseServer を使用する新しいマシンからの製品を全て登録します。この手順は移行後に行うこともできます。
- FlowForce Server サービスが稼働中の場合、停止してください。 [Linux](#)、[macOS](#)、と[Windows](#) のための命令を参照してください。
- DATADIR** を例えば **temp\_data** に名前を変更してください。
- `--datadir` として **DATADIR** を、`--olddatadir` として **DATADIR\_BACKUP** を提供して `migratedb` コマンドを実行します。例:

#### Windows

```
FlowForceServer migratedb
--datadir=C:\ProgramData\Altova\FlowForceServer2021\data
--olddatadir=C:\transfer\backup_data
```

#### CentOS

```
sudo ./flowforceserver migratedb
--datadir=/var/opt/Altova/FlowForceServer2021/data
--olddatadir=/home/chang/backups/data
```

- (順番に) *FlowForce Server* と *FlowForce Web Server* サービスを開始します

## 4 ユーザーアクセスの管理

このセクションにはFlowForce Server 内のユーザーアクセス管理に適用することができる手順と概念が含まれます。

## 4.1 ユーザーとロール

このセクションは以下のトピックを含んでいます:

- [ユーザーの作成](#)
- [ロールの作成](#)
- [ドメインユーザーとロールをインポート](#)
- [デフォルトのユーザーとロール](#)
- [ユーザーとロールの名前の変更](#)
- [ユーザーロールを割り当てる](#)
- [ロールを他のロールに割り当てる](#)

### 4.1.1 ユーザーの作成

ユーザーとはFlowForce Server ログオンジョブを構成、またはMapForce またはStyleVision 変換をデプロイし、FlowForce Server を管理する個人を指します。ユーザーが使用することのできるアクションは以下には影響されません

- a) 割り当てられた [パーミッション](#) または [特権](#)
- b) ユーザーがメンバーであるロールに割り当てられている [パーミッション](#) と [特権](#)

FlowForce Server ユーザーを追加する

1. 「管理」をクリックします。そして、「ユーザー」をクリックします。
2. 「ユーザーの作成」をクリックします。
3. 必要なフィールドに情報を入力します。

|                   |  |
|-------------------|--|
| ユーザー名             | ユーザー名を入力します。以下の制約が適用されます: <ul style="list-style-type: none"> <li>• フィールドが空ではありません</li> <li>• ユーザー名の始めと終わりにスペースを使用することはできません</li> <li>• 許可されている文字: 英数字、数字、アンダースコア、( _ )、ダッシュ、( - )、およびドット ( . )。</li> </ul> |
| パスワード             | ユーザーのパスワードを入力します。  |
| パスワードの再入力         | ユーザーのパスワードを再入力します。   |
| 次回ログインでパスワードを変更する | このチェックボックスを選択すると、ユーザーはパスワードを次回ログインで変更するように問われます。   |

4. 任意で、ユーザーに必要な特権を与えてください(使用することのできる特権に関する詳細は、以下を参照してください! [特権](#))。ユーザーに特権をこのページから直接与えるか、既に特権を有するロールにユーザーを割り当て特権を与えることができます。ユーザーの管理を簡素化するために、後者のアプローチが奨励されます(以下を参照: [ロールの追加](#) と [ユーザーロールを割り当てる](#))。
5. 「保存」をクリックします。

## 4.1.2 ロールの作成

ロールは、ビジネスニーズをベースとしたセキュリティの強化を手助けする **特権** のセットです。標準的なロールをベースとしたセキュリティは、最低2つのロールが含まれます: 管理者と一般ユーザー。各ロールは、ロールに与えられた特権により定義されています。例えば、管理者は、自身のおよび他のユーザーのパスワードを変更することができますが、一般ユーザーは自身のパスワードのみを変更することができます。ユーザーロールを割り当て、必要に応じて、ユーザーのロールを取り消すことができます。

### FlowForce Server ロールを追加する

1. 「管理」をクリックし、「ロール」をクリックします。
2. 「ロールの作成」をクリックします。
3. ロール名を入力します (例えば、「Administrator」)。
4. 「特権」から、ロールに割り当てられる特権を選択します (使用することのできる特権に関する詳細は、以下を参照してください: [特権](#))。
5. 「保存」をクリックします。

## 4.1.3 ドメインユーザーとロールをインポート

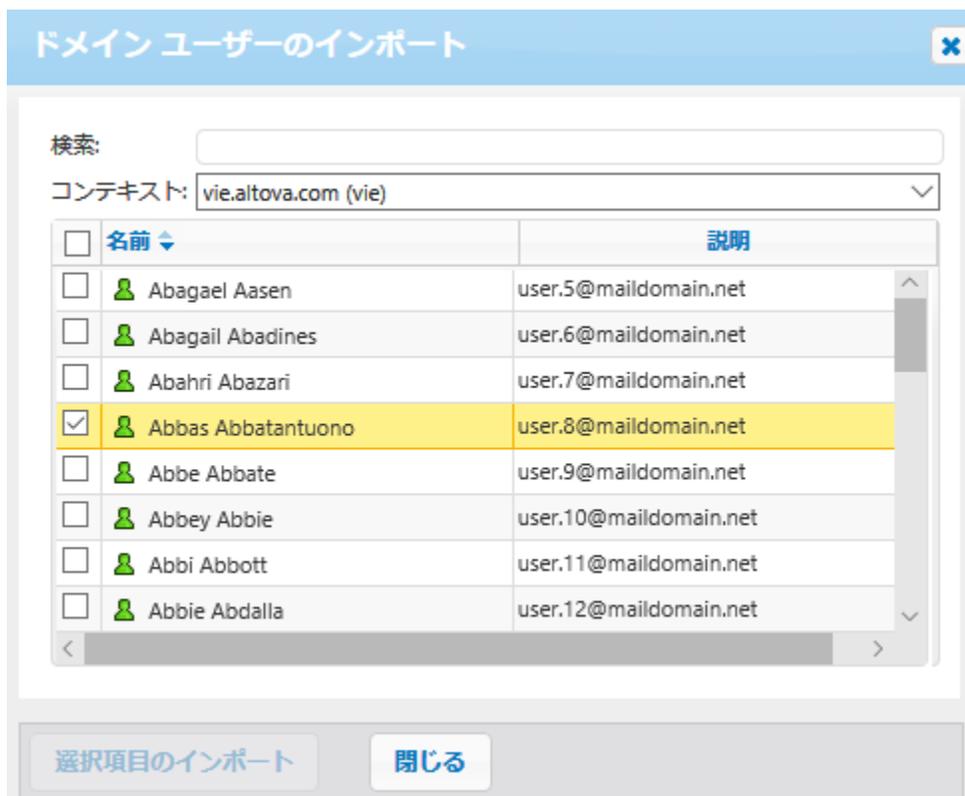
FlowForce Server ユーザーの作成に加え、ドメインユーザーアカウントとグループを Windows Active Directory からインポートすることができます。

「ドメインユーザーログインを許可」設定が [ディレクトリサービス設定](#) 内で有効化されている場合、Windows ドメインユーザーは、アカウントが FlowForce Server データベースに明示的にインポートされていない場合でも、FlowForce Server にログインすることができます。ドメインユーザーの FlowForce サーバーへのログインを確実にするには、ユーザーの Windows アカウントが明示的に管理者によりインポートされており、「ドメインユーザーログインを許可」チェックボックスがクリアされている状態で、ドメインユーザーをインポートします。

**メモ** ローカルマシンアカウントは Active Directory の一部ではない可能性があり、FlowForce Server にインポートできません。

### Windows ドメインユーザーアカウントを FlowForce Server にインポートする

1. 「設定」から「ディレクトリサービス」の下の「有効化」チェックボックスを選択し、[ディレクトリサービスの設定を変更する](#) 内で説明される通り優先されるディレクトリサービスプロバイダーを構成します。
2. 「管理」の下の「ユーザー」をクリックします。
3. 「ドメインユーザーのインポート」をクリックします。



- 適用できる場合、「コンテキスト」ドロップダウンリストから選択のドメインを選択します。
- インポートするユーザーアカウント名を入力します。部分的な検索は有効です、ですから、例えば「ad」などの値を入力すると、アカウント「管理者」、「Admanager」、および「Admin」は、LDAP サーバーまたは Active Directory から抽出され、Web ページダイアログで表示されます。Active Directory の場合、FlowForce Server は、単一の句内で複雑な検索条件を指定することのできる Ambiguous Name Resolution (ANR) 検索のアルゴリズムを使用します。例えば「ji sm」と入力することにより、Jim Smith という名前の個人のアカウントを抽出することができます。Active Directory 内の Ambiguous Name Resolution に関する Microsoft ドキュメントを参照してください。
- インポートされる記録の横のチェックボックスをチェックし、「インポート」をクリックします。Active Directory から記録が抽出されるまで待ってください。ドメインがローカルではない場合、待ち時間が増加する可能性があります。

#### ドメイングループを FlowForce Server にインポートする

- 「管理」から「Roles」をクリックします。
- 「ドメインロールのインポート」をクリックします。
- 上のステップ 4-6 に従います。

以下も参照してください

- [デレトリサービス設定の変更](#)

#### 4.1.4 デフォルトのユーザーとロール

##### デフォルトのユーザー

FlowForce Server には、以下の定義済みのユーザーが存在します。

|  |   |
|--|---|
|  <b>root</b>      | ユーザーは、最初のトップレベルのFlowForce Server 管理者です。デフォルトでは、システム内の全てのパーミッションおよび特権を有します。  |
|  <b>anonymous</b> | これは、明示的にログインしないユーザーのための特別なアカウントです。FlowForce Server 管理インターフェイスへの匿名のアクセスは、不可能ですが、として示されている特定の Web サービスのための匿名のアクセスを有効化することができます（以下を参照: <a href="#">ジョブを Web サービスとして表示する</a> ）。 |

特権を変更することはできますが、ビルトインユーザーを削除することはできません。

**メモ** ユーザー  **root** は、自身のパーミッションと特権を含む特権とパーミッションを全て変更することができます。意図せず自身のシステムアクセスを失う可能性があるため、 **root** としてログインする場合は、慎重にレートの特権を編集してください。これが発生した場合、[ルート/パスワードの再設定](#)を参照してください。

### デフォルトのロール

FlowForce Server には、以下の定義済みのロールが存在します。

|  |  |
|--|--|
|  <b>authenticated</b> | このロールは、既存のユーザー名とパスワードの使用を認証されている全てのユーザーを含みます。ユーザー  <b>anonymous</b> 以外の各 FlowForce Server ユーザーは、このロールのメンバーです。デフォルトでは、このロールは、自身のパスワードを設定する特権を有しています。 |
|  <b>all</b>          | このロールは、ユーザー  <b>anonymous</b> を含む全ての FlowForce Server ユーザーを含みます。デフォルトでは、このロールには特権は与えられていません。  |

ロール  **authenticated** または  **all** は、ビルトインのため、これらのロールをユーザーに割り当てる、またはユーザーから削除することはできません。ビルトインロールのメンバーシップは自動的に FlowForce Server により管理されています。新しいユーザーを追加する都度、FlowForce Server は、自動的に新しいユーザーにロール  **authenticated** とロール  **all** を割り当てます。

ビルトインのユーザーとロールの特権を変更する場合は、この変更が引き起こす影響を考慮してください。現在与えられている全ての特権のグローバルなビューを確認するには、特権レポートを使用してください（以下を参照: [特権レポート](#)）。

## 4.1.5 ユーザーとロールの名前の変更

### ユーザー名を変更する

1. 「管理」をクリックします。そして、「ユーザー」をクリックします。
2. 編集するレコードをクリックします。
3. 「ユーザー名」テキストボックスに新しい名前を入力し、「保存」をクリックします。

#### メモ

- ユーザー名が変更されても、現在割り当てられているユーザーパスワードは変更されません。
- (名前変更の特権を有すると前提して) ユーザー自身の名前を変更する場合、変更された名前は保存をクリックすると有効になり、ページの右上に表示されます。

#### ロールの名前を変更する

1. 「管理」をクリック、「ロール」をクリックします。
2. 編集するレコードをクリックします。
3. 「ロール名」テキストボックスに新しいロール名を入力し、「保存」をクリックします。

#### メモ

- ロール名が変更されても、ロールのメンバーは変更されません。
- デフォルトロールである  all と  authenticated は変更することができません。

### 4.1.6 ユーザーにロールを割り当てる

#### ユーザーに1つまたは複数のロールを割り当てる

1. 「管理」をクリックします。そして、「ユーザー」をクリックします。
2. ユーザーのリストから、編集するレコードをクリックします。
3. 「利用可能なロール」から、ユーザーに割り当てられるロールを選択し、「割り当て」をクリックします。

#### ユーザーから1つまたは複数のロールを削除する

1. 「管理」をクリックします。そして、「ユーザー」をクリックします。
2. ユーザーのリストから、編集するレコードをクリックします。
3. 「ユーザー<ユーザー名>」に割り当てられたロールから、ユーザーから削除するロールを選択し、「削除」をクリックします。

#### ロールを複数のユーザーに割り当てる

1. 「管理」をクリックします。そして、「ロール」をクリックします。
2. ロールのリストから、編集するレコードをクリックします。
3. 「利用可能なユーザー/ロール」から、ロールを割り当てられるユーザーを選択し、「割り当て」をクリックします。

#### ロールを複数のユーザーから削除する

1. 「管理」をクリックします。そして、「ロール」をクリックします。
2. ロールのリストから、編集するレコードをクリックします。
3. 「ロール<ロール名>のメンバー」からロールを削除するユーザーを選択し、「削除」をクリックします。

### 4.1.7 ロールを他のロールに割り当てる

ロールを他のロールに割り当てることにより所属する機関、またはビジネスの階層を FlowForce Server 内でモデルすることができます。例えば  社員 というロールと、 マーケティング部署 というロールを作成します。ロール  マーケティング部署 を  社員のメンバーになるように割り当てます。これは  社員 に与えられた全ての特権とパーミッションが自動的に  マーケティング部署 のメンバーにより継承されることを意味します。

ロールを他のロールに割り当てる

1. 「管理」をクリックします。そして、「ロール」をクリックします。
2. ロールのリストから、他のロールを割り当てるロールをクリックします(例えば、ロール  マーケティング部署 (ロール  社員 から特権を継承する場合 社員 (Employees) をクリックします)。
3. 「利用可能なユーザーロール」から、割り当てられるロールを選択し、「割り当て」をクリックします。

次も参照してください

- [特権の動作方法](#)
- [パーミッションのしくみ](#)

#### 4.1.8 ルートパスワードの再設定

 **root** ユーザーアカウントのパスワードを忘れた場合、または紛失した場合、コマンドラインインターフェイスからデフォルトの値をリセットすることができます(以下を参照: コマンド [resetpassword](#) )。

ルートパスワードをリセットする場合、FlowForce バイナリとデータファイルを含む、FlowForce が動作しているオペレーティングシステムへのアクセスを有するとみなします。これは、FlowForce をインストールする場合、または新規 FlowForce バージョンまたはサーバーを手動で移行する場合に必要なアクセスと同類のアクセスです。

パスワードをリセットする場合、 **root** ユーザーの特権は、デフォルト値に戻されます(つまり、全ての特権が与えられます)。

ルートパスワードのリセットは、 **root** ユーザーのみに影響し、他の FlowForce ユーザーには影響しません。

## 4.2 特権

このセクションは以下のトピックを含んでいます:

- [特権の作動方法](#)
- [特権レポートの閲覧](#)

### 4.2.1 特権の作動方法

特権は、FlowForce Server 内でユーザーが行うことのできる内容を定義します(例えば、自身のパスワード設定、ユーザーとロールの読み取り、ジョブの停止、など)。特権は、パーミッションはコンテナへのユーザーアクセスを管理し、特権はグローバルにFlowForce Server 全体に対して効果がある点で [パーミッション](#) と異なります。次の簡単なルールは、特権とパーミッションの違いを理解する助けになります。特権はグローバルであり、パーミッションはローカルです。

パーミッションに類似して、特権は個々のユーザーと [ロール](#) に割り当てることができます。ですから、ユーザーがFlowForce Server にログインすると、有効な特権は以下により決定されます:

- a) 直接割り当てられた特権
- b) ユーザーがメンバーであるロールに割り当てられた特権

以下の特権がFlowForce Server で使用することができます。

|                |  |
|----------------|--|
| 実行キューの定義       | ジョブ実行キューに関する設定を定義する権利を与えます。 <a href="#">キューの設定を定義する方法</a> を参照してください。   |
| グローバルな設定の管理    | この特権は、タイムゾーンとメールサーバー設定などの設定ページ内で使用することのできるFlowForce Server グローバル設定を変更する権利を与えます。これは、FlowForce Server 管理者の特権であり、管理者のみと与えられる特権です。   |
| ユーザー、ロール、特権の管理 | この特権は、次のデータを追加、編集、および削除する権利です: <ul style="list-style-type: none"> <li>• ユーザー</li> <li>• ロール</li> <li>• 特権</li> <li>• パスワード</li> </ul> これは、FlowForce Server 管理者の特権であり、管理者のみと与えられる特権です。デフォルトでは、ユーザー  root のみがこの特権を有しています。 |
| セキュリティの上書き     | この特権を持つユーザーは、セキュリティの「書き込み」パーミッション無しにコンテナを変更することができます。これによりFlowForce Server 管理者は、誤ってアクセス付加しなおアクセスを取り戻すことができます。<br>これは、FlowForce Server 管理者の特権であり、管理者のみと与えられる特権です。デフォルトでは、ユーザー  root のみがこの特権を有しています。                      |
| ユーザーとロールの読み取り  | デフォルトでは、ユーザーは、自身のユーザーアカウントとメンバーであるロールを閲覧することができます。この特権を有すると、ユーザーは、全ての既存のユーザーとロールを閲覧することができます。<br>デフォルトでは、ユーザー  root のみがこの特権を有しています。   |

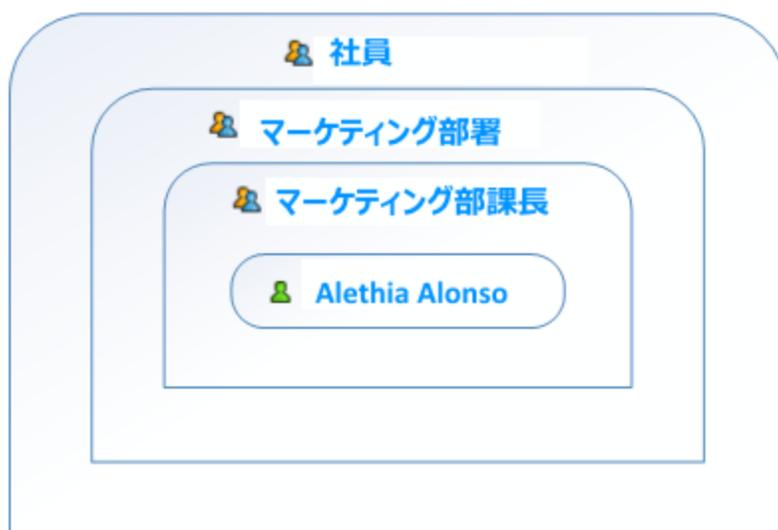
|                |  |
|----------------|--|
| 機密データの抽出       | <p>この特権はプレーンなテキストとしてパスワードを抽出しビューする権限を与えます。</p> <p>デフォルトでは、ユーザー  <b>root</b> のみが特権を有しています。それ以外を設定する理由が存在する以外、この特権は  <b>root</b> のみに予約されています。</p>  |
| パスワードの設定       | <p>この特権は、ユーザーにユーザー自身のパスワードを変更する権利を与えます。この特権を有さないユーザーは FlowForce Server 管理者が設定したパスワードを取得する必要がありません。</p> <p>デフォルトでは、 <b>anonymous</b> 以外の  <b>authenticated</b> ロールとすべてのユーザーアカウントがこの特権を有します。</p> |
| ジョブの停止         | <p>この特権は、ジョブを作成したユーザーに関わり無く、実行中の FlowForce Server ジョブを停止する権利を与えます。</p>   |
| フィルター処理無しログの表示 | <p>デフォルトでは、ユーザーは「読み取り」アクセスがある構成に関連したログエントリを閲覧することができます。この特権を有すると、ユーザーは、特定の構成に関連したすべてのログエントリを閲覧することができます。</p> <p>デフォルトでは、ユーザー  <b>root</b> のみがこの特権を有しています。</p>   |

## 継承

ユーザー（例えば  **Alethia Alonso**）に直接、または、特定のロール（例えば  **マーケティング部課長**）に割り当てることができます。長期的に特権の管理を簡素化するため、後者のアプローチが奨励されます。例えば、ユーザーは所属部署を変更、または、入社、退社する可能性があります。この場合、個々のユーザーの特権を管理することは、非効率的なタスクとなる可能性があります。特権を、ユーザーではなく、ロールに割り当てることにより、細部に及ぶ管理を簡素化することができ、各ユーザーではなく、各グループ、または部署のビジネスニーズに焦点を当てることができます。

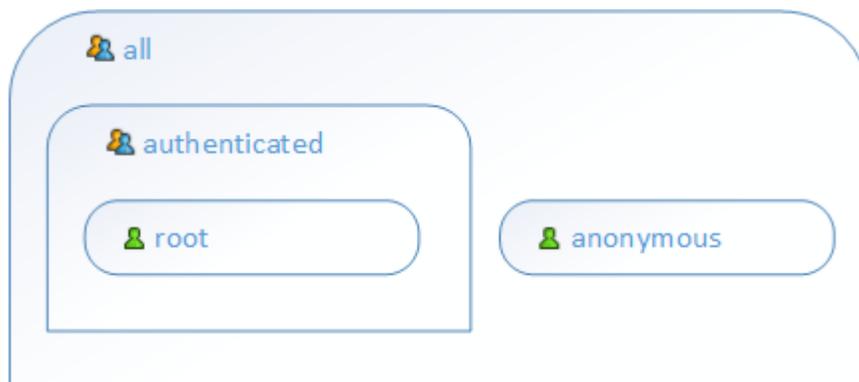
ロールを他のロールに割り当てることにより、所属する機関、または、ビジネスの階層を FlowForce Server 内でモデルすることができます。例えば  **社員** というロールと  **マーケティング部署** というロールを作成します。ロール  **マーケティング部署** を  **社員** のメンバーになるように割り当てます。これは  **社員** に与えられた全ての特権とパーミッションが自動的に  **マーケティング部署** のメンバーにより継承されることを意味します。

更に  **マーケティング部課長** ロールに  **マーケティング部署** ロールのメンバーになるように割り当てることができます。この場合、 **マーケティング部課長** ロールは、 **マーケティング部署** と  **社員** ロールの双方から特権を継承します。新しいマネージャーが入社すると、 **Alethia Alonso** が  **マーケティング部課長** ロールに割り当てられていると、広いロールからの特権を継承することになります。



図が表示するように、**Alethia Alonso** は、パーミッションと特権をロール **マーケティング部課長** から継承します。このロールは **マーケティング部署** および **社員** から特権を継承します。

新規にインストールされた FlowForce Server システムでは、[デフォルトのユーザーとロール](#) ユーザーと特権のダイアグラムは以下のようになります。



図が表示するように、システム内の各ユーザーは、**all** ロール内で定義された特権を継承します。しかしながら、既存のユーザー（この場合、**root**）のみが、**authenticated** ロール内で定義された特権を継承します。FlowForce Server に新規のユーザーを追加すると、これらのユーザーは **all** と **authenticated** ロールに割り当てられます（該当する場合、これらのロール内で定義された特権が与えられます）。



以下も参照してください

- [デフォルトのユーザーとロール](#)
- [特権レポートの閲覧](#)

## 4.2.2 特権レポートの閲覧

As a FlowForce Server 管理者として、特にユーザーとロールの数が増えると、各ロールまたはユーザーに与えられた特権を管理することは難しく感じるかもしれません。現在ユーザーとロールに割り当てられている全ての特権の概要を素早く確認するために FlowForce Server には以下のレポートが搭載されています:

- 特権レポート
- ユーザーレポートの特権

これらのレポートを閲覧するには、「管理」をクリックし、「レポート」をクリックします。

### 特権レポート

このレポートは、FlowForce Server 特権をリストしています。それぞれの特権に関して、特権を与えられたユーザー、または、ロールにより継承された特権を確認することができます。

## 特権レポート

| 特権              | プリンシパル  | 与えられた、またはプリンシパルにより継承  |
|-----------------|---|---|
| 定義の実行のキュー       |  <a href="#">root</a>  |  <a href="#">root</a> に与えられている   |
| クラスターの管理        |  <a href="#">root</a>  |  <a href="#">root</a> に与えられている   |
| グローバル設定の管理      |  <a href="#">root</a>  |  <a href="#">root</a> に与えられている   |
| ユーザー、ロール、特権の管理  |  <a href="#">root</a>  |  <a href="#">root</a> に与えられている   |
| セキュリティの上書き      |  <a href="#">root</a>  |  <a href="#">root</a> に与えられている   |
| ユーザーとロールの読み取り   |  <a href="#">Manager</a><br> <a href="#">root</a>   |  <a href="#">Manager</a> に与えられている<br> <a href="#">root</a> に与えられている   |
| 機密情報を抽出する       | 許可されていません   |   |
| 独自のパスワードをセット    |  <a href="#">all</a><br> <a href="#">anonymous</a><br> <a href="#">root</a> |  <a href="#">all</a> に与えられている<br> <a href="#">all</a> から継承された<br> <a href="#">root</a> に与えられている |
| ジョブの停止          |  <a href="#">root</a>  |  <a href="#">root</a> に与えられている   |
| フィルター処理なしのログの表示 |  <a href="#">root</a>  |  <a href="#">root</a> に与えられている   |

特権レポート (サンプル)

### ユーザーレポートによる特権

このレポートは、FlowForce Server ユーザーをリストします。それぞれのユーザーに関して、現在割り当てられている特権、およびこれらのユーザーが与えられている、または、継承している特権を確認することができます。

## ユーザー レポートによる特権

| プリンシパル  | 特権              | 与えられた、またはプリンシパルにより継承。  |
|---|-----------------|--|
|  <a href="#">Manager</a>   | ユーザーとロールの読み取り   |  <a href="#">Manager</a> に与えられている |
|  <a href="#">all</a>       | 独自のパスワードをセット    |  <a href="#">all</a> に与えられている     |
|  <a href="#">anonymous</a> | 独自のパスワードをセット    |  <a href="#">all</a> から継承         |
|  <a href="#">root</a>      | 定義の実行のキュー       |  <a href="#">root</a> に与えられている    |
|   | クラスターの管理        |  <a href="#">root</a> に与えられている    |
|   | グローバル設定の管理      |  <a href="#">root</a> に与えられている    |
|   | ユーザー、ロール、特権の管理  |  <a href="#">root</a> に与えられている    |
|   | セキュリティの上書き      |  <a href="#">root</a> に与えられている    |
|   | ユーザーとロールの読み取り   |  <a href="#">root</a> に与えられている    |
|   | 独自のパスワードをセット    |  <a href="#">root</a> に与えられている   |
|   | ジョブの停止          |  <a href="#">root</a> に与えられている  |
|   | フィルター処理なしのログの表示 |  <a href="#">root</a> に与えられている  |

ユーザーレポートによる特権 (サンプル)

## 4.3 パーミッションとコンテナ

このセクションは以下のトピックを含んでいます:

- [パーミッションのしくみ](#)
- [コンテナを理解する](#)
- [コンテナの作成、名前変更、および移動](#)
- [コンテナのパーミッションの閲覧](#)
- [コンテナのパーミッションの変更](#)
- [/public コンテナへのアクセスの制限](#)

### 4.3.1 パーミッションの動作方法

パーミッションは、ユーザーのコンテナへのアクセスを管理します。特権と同様、[パーミッション](#)は、ユーザーとロールに与えられることができます。ですから、ユーザーがロールの一部である場合、ロールに与えられたパーミッションは、ユーザーにも自動的に適用されます。

デフォルトで、コンテナに設定されたパーミッションは、親コンテナから継承されます。例えば、コンテナ A には、子コンテナ B があるとします。コンテナ A にアクセスするパーミッションを持つユーザーは、コンテナ B にもアクセスすることのできるデフォルトのパーミッションを持つことになります。しかしながら、管理者は、コンテナの階層の全てのレベルで、全てのユーザーおよびロールのパーミッションを再定義することができます。

FlowForce は、ユーザーがコンテナと作業するとコンテナパーミッションをチェックします。例えば、ユーザーは必要とされるパーミッションを与えられている場合のみコンテナのコンテンツを閲覧、または変更することができます。ジョブの実行でパーミッションは評価されず、パーミッションの変更は既存のジョブには適用されません。

各 FlowForce Server コンテナのために、以下のパーミッションの型を設定することができます。

#### コンテナ

「コンテナ」のパーミッションはユーザーが現在のコンテナ内のオブジェクトと作業できる項目を定義します。

|           |   |
|-----------|---|
| 継承        | ユーザーに親コンテナ上で定義されているとおりのコンテナにアクセスする同じ権利を提供します。   |
| 読み取り      | ユーザーにコンテナのコンテンツをリストする権利を提供します。  |
| 読み取り、書き込み | コンテナのコンテンツをリストし、コンテナ内でオブジェクトを作成し削除する権利をユーザーに提供します。<br><br>メモ 新規構成オブジェクトの作成に成功する、または既存のオブジェクトを削除する、またはユーザーはコンテナの読み取り、書き込みパーミッションと構成の読み取り、書き込みパーミッションを与えられている必要があります。 |
| アクセス無し    | ユーザーにコンテナに入力する権利を拒否します(更に具体的には、コンテナはユーザーに対して無効化済みと表示されます)。  |

#### 構成

「構成」パーミッションはユーザーが現在のコンテナ内で構成オブジェクトと作業できる項目(具体的には、ジョブと資格情報)を定義していません。

|           |  |
|-----------|--|
| 継承        | ユーザーが親コンテナ上で定義されている構成オブジェクトに関連した同じ権利を提供します。  |
| 読み取り      | コンテナ内で構成オブジェクトについての情報をビューする権利をユーザーに提供します (実行ステップまたはジョブのトリガーなど)。  |
| 読み取り、書き込み | このコンテナ内で構成オブジェクトを変更する権利をユーザーに提供します (例えば、ジョブのトリガーの編集など)。<br><br>メモ 新規構成オブジェクトの作成に成功する場合は、または、既存のオブジェクトを削除する場合は、ユーザーはコンテナの読み取り、書き込みパーミッションと構成の読み取り、書き込みパーミッションを与えられている必要があります。 |
| アクセス無し    | コンテナ内で構成オブジェクトについての情報をビューする権利をユーザーに対して否定します (更に具体的には、構成オブジェクトはユーザーに対して無効化済みと表示されます)。   |

### 資格情報

このパーミッションは、ユーザーがこのコンテナ内で定義されている [資格情報](#) をどのように使用できるかを定義します。

|        |  |
|--------|--|
| 継承     | 親コンテナ上で定義されているようにユーザーに資格情報に関連した同じ権利を提供します。 |
| 使用     | このコンテナ内で定義されている資格情報を再利用する権利をユーザーに提供します。    |
| アクセス無し | このコンテナ内で定義されている資格情報を再利用する権利をユーザーに対して否定します。 |

### キュー

このパーミッションは、ユーザーがこのコンテナ内で定義されているキューのようを使用するかを定義します。

|        |  |
|--------|--|
| 継承     | 親コンテナ上で定義されているようにユーザーに同じキュー権利を提供します。       |
| 使用     | ユーザーがジョブをこのコンテナ内で定義されているキューに割り当てる権利を提供します。 |
| アクセス無し | ユーザーがジョブをこのコンテナ内で定義されているキューに割り当てる権利を拒否します。 |

### サービス

「サービス」パーミッションはHTTP リクエストインターフェイスを介して Web サービスとして公開されているジョブへのアクセスを定義します。

|    |  |
|----|--|
| 継承 | 親コンテナ上で定義されているようにユーザーにサービスに関連した同じ権利を提供します。   |
| 使用 | ユーザーにサービスにアクセスする権利を提供し、リクエストインターフェイスからジョブを実行する権利をユーザーに提供します。<br><br><div style="border: 1px solid black; padding: 5px; background-color: #e0e0e0;"> <p>メモ</p> <ul style="list-style-type: none"> <li>サービスパーミッションチェックは、コンテナ階層チェックをスキップします。このため、使用パーミッションが与えられている場合、対応するジョブが定義されているコンテナへの読み取りアクセスを持たずユーザーがサービスを使用できる場合があります。</li> <li>使用パーミッションをユーザー  <b>anonymous</b> に与えると、サービスはブロッカーで使用することができ、認証を必要としません。</li> </ul> </div> |

|        |                                     |
|--------|-------------------------------------|
| アクセス無し | Web サービスとしてジョブアクセス権利をユーザーに対して否定します。 |
|--------|-------------------------------------|

### 関数

ジョブ、証明書、他の構成データに加え、コンテナには関数が含まれる場合があります。これには、FlowForce にデプロイ済みの FlowForce 関数、RaptorXML 関数、および MapForce マッピング、または、StyleVision 変換も含まれます。

FlowForce ユーザーがジョブを作成すると、ジョブ内の実行ステップの一部は同じコンテナ、または、異なるコンテナから関数を参照する場合があります。「関数」パーミッションは、ユーザーが関数をパーミッションが定義されているコンテナから呼び出す、または、参照できるかを定義します。

例えば「Restricted」と呼ばれる FlowForce コンテナに複数の MapForce マッピングを管理者がデプロイしたと仮定します。管理者は「関数」パーミッションを変更することにより、ユーザーがこのコンテナ内の関数を参照するかを決定できます。具体的には、コンテナ「制限されている」上で「関数 - 使用」パーミッションを持つユーザーまたはロールはこのコンテナから関数を参照することができます（例、実行ステップを作成する際ドロップダウンリストから選択します）。他方、「関数 - アクセス無し」パーミッションを持つユーザー、または、ロールが制限されているコンテナから関数を選択することができません。

ジョブ内で関数が既に使用された後に、管理者がユーザーの関数へのアクセスを取り消すと、これらのユーザーはジョブを実行することができなくなります。ジョブ構成ページは、この場合、「選択されている関数を使用するパーミッションがありません」というテキスト共にメッセージを表示します。

|        |  |
|--------|--|
| 継承     | 親コンテナ上で定義されているようにユーザーに関数に関連した同じ権利を提供します。 |
| 使用     | ユーザーコンテナ内で定義されている関数を呼び出す（参照する）権利を提供します。  |
| アクセス無し | ユーザーコンテナ内で定義されている関数を呼び出す（参照する）権利を否定します。  |

### リソース

このパーミッションはどのようにユーザーがこのコンテナで定義済みのリソースの使用できるかを定義します。

|        |  |
|--------|--|
| 継承     | 親コンテナ上で定義されているようにユーザーに同じリソースに関連する権利を提供します。 |
| 使用     | ユーザーこのコンテナ内で定義されているリソースを使用（参照）する権利を提供します。  |
| アクセス無し | ユーザーこのコンテナ内で定義されているリソースを使用（参照）する権利を否定します。  |

### セキュリティ

セキュリティパーミッションは、現在のコンテナ内で定義されている子のコンテナにアクセスするパーミッションを管理します。

デフォルトでは、ユーザーは読み取り専用パーミッション（すなわち、割り与えられているパーミッション、または、メンバーであるロール）のみが許可されています。しかしながら、読み取りユーザーとロール特権を持つユーザーは、すべてのパーミッションエントリを読み取ることができます。

|            |  |
|------------|--|
| 継承         | 親コンテナ上で定義されているようにユーザーにセキュリティに関連した同じ権利を提供します。 |
| 読み取りセキュリティ | コンテナの子のパーミッションをビューする権利をユーザーに提供します。           |

|                 |                                    |
|-----------------|------------------------------------|
| 読み取りと書き込みセキュリティ | コンテナの子のパーミッションを変更する権利をユーザーに提供します。  |
| アクセス無し          | コンテナの子のパーミッションをビューする権利をユーザーに否定します。 |

### 4.3.2 コンテナを理解する

名前が意味するように、コンテナはパッケージされたデータです。FlowForce Server では、コンテナはオペレーティングシステム上のフォルダーと大まかに比較することができます。コンテナは以下を含むことができます: [ジョブ](#)、[資格情報](#)、[関数](#) と他のコンテナ。コンテナで [パーミッション](#) を設定することにより、コンテナ内のデータを閲覧およびデータにアクセスする権利を管理することができます。データをコンテナ別に整理し、対応するパーミッションを各コンテナの適切なセットアップすることは、セキュリティ対策となるでしょう。

FlowForce Server の一番上のコンテナは、ルート (/) コンテナです。デフォルトでは、ルートコンテナは次の定義済みの FlowForce Server コンテナを含んでいます。

|                   |  |
|-------------------|--|
| <b>/public</b>    | <p><b>/public</b> コンテナは、FlowForce ユーザーがジョブと資格情報を作成することのできるデフォルトのロケーションです。デフォルトの状態は空で、FlowForce のユーザー全てがアクセスすることができます。/public コンテナは次の場合デフォルトのロケーションとして使用されます:</p> <ul style="list-style-type: none"> <li>マッピングを MapForce から FlowForce Server にデプロイする場合。</li> <li>StyleVision からの変換を FlowForce Server にデプロイする場合。</li> </ul> <p>しかしながら、マッピングまたは変換を必要であれば、異なるコンテナにデプロイすることができます。</p> |
| <b>/RaptorXML</b> | <p>このコンテナは、RaptorXML Server がライセンスされている場合、使用することができます。このコンテナは、RaptorXML Server に固有の検証と他の関数を保管します。</p>  |
| <b>/system</b>    | <p><b>system</b> コンテナは、FlowForce Server システム関数を含んでいます。このコンテナを変更することは、奨励されません。</p>  |

クリックして、コンテナのコンテンツを確認し、Web 管理インターフェイスからコンテナをナビゲートすることができます。次のスクリーンショットは、複数の構成オブジェクトを含むサンプル /public コンテナを表しています。

| 名前                                 | 型              | 日付は変更されました          | 以下により変更さ: 次の実行   |         |
|------------------------------------|----------------|---------------------|------------------|---------|
| my.credentials                     | コンテナ           |                     |                  | パーミッション |
| Altova_Hierarchical_DB.mapping     | MapForce マッピング | 2016-04-29 15:08:16 | root             |         |
| BuildHierarchyFromTextfile.mapping | MapForce マッピング | 2015-09-22 09:27:01 | root             |         |
| CompletePO.mapping                 | MapForce マッピング | 2016-04-27 15:46:49 | root             |         |
| Customers_DB.mapping               | MapForce マッピング | 2016-04-26 16:49:06 | root             |         |
| DB_PhoneList.mapping               | MapForce マッピング | 2016-06-16 17:04:42 | root             |         |
| MultipleFilesFromTextfile.mapping  | MapForce マッピング | 2015-09-17 13:55:16 | root             |         |
| PersonDB-dyn.mapping               | MapForce マッピング | 2016-04-29 15:00:56 | root             |         |
| PersonDB.mapping                   | MapForce マッピング | 2016-04-29 13:21:48 | root             |         |
| BuildHierarchyFromTextfile.job     | ジョブ            | 2015-09-22 09:28:17 | root             | ログの表示   |
| CompletePO.job                     | ジョブ            | 2016-04-27 15:44:40 | root             | ログの表示   |
| MultipleFilesFromTextfile.job      | ジョブ            | 2015-09-17 13:55:34 | root             | ログの表示   |
| db2osm                             | ジョブ            | 2016-06-16 10:20:43 | 2016-07-01 10:25 | ログの表示   |

サンプルFlowForce コンテナ

階層内のコンテナに戻るには、ページ上の階層リンクナビゲーションを使用してください。

(再帰的チェックボックスがチェックされている場合)子オブジェクトを含む現在のコンテナ内、または、(再帰的チェックボックスのチェックが解除されている場合)現在のコンテナ内のみでオブジェクトを検索することができます。

コンテナは、ジョブ、デプロイされたMapForce マッピング またはStyleVision 変換、関数、資格情報などを含みます。コンテナを開くと、オブジェクトに関する次の情報を確認することができます。

| プロパティ      | 説明   |
|------------|--|
| 名前         | ファイルシステムのオブジェクトの名前を指定します。新規オブジェクトを作成する場合、使用されている名前を使用することはできない点に注意してください。  |
| 型          | <p>(資格情報、ジョブ、または、関数などの)オブジェクトの型を定義します。オブジェクトの型に対応するアイコンで識別することができます。</p> <ul style="list-style-type: none"> <li> 資格情報</li> <li> (ビルトイン関数、MapForce マッピングとStyleVision 変換を含む)関数</li> <li> ジョブ</li> <li> コンテナ</li> <li> 不足する構成オブジェクト。解決されていない依存関係を持つFlowForce Server データをトロードしようと試みるとこのアイコンが表示されます。<a href="#">不足する依存関係の処理方法</a>を参照してください。</li> </ul> |
| 変更された日付    | オブジェクトが最後に作成された、または、最後に編集された日時を指定します   |
| 以下により変更された | オブジェクトを変更したユーザーの名前を指定します。  |
| 次の実行       | トリガーにより実行されることが計画されているジョブのため、この列は、ジョブ設定内で定義されているように、次の実行の日時を指定します。   |
| ログビュー      | ジョブのため、このボタンを使用することにより、対応するジョブの実行ログへ素早くアクセスすることができます。  |

[パーミッション](#)を有すると想定し、(例えば、各部署にコンテナをそれぞれ作成する場合など)カスタムFlowForce サーバーデータを保管するために複数の追加コンテナを作成することができます。または、認証されたユーザーが使用することのできる、/public コンテナ内にデータを保管することもできます。必要に応じて、/public コンテナへのアクセスを制限することも可能です(以下を参照: [/public コンテナへのアクセスの制限](#))。

相対するパーミッションを有する箇所でコンテナの移動、名前の変更、および削除を行うことができます。

### 4.3.3 コンテナの作成、名前変更、および移動

(メンバーであるロールが) コンテナ / 読み取り、書き込みパーミッションを有する場合、コンテナの作成、名前変更、および移動することができます。(以下も参照してください: [パーミッションのしくみ](#))。

**メモ** FlowForce Server によりデフォルトで与えられた /RaptorXML と /system コンテナの内容を変更することは奨励されません。

コンテナの作成方法:

1. 「構成」をクリックします。
2. 新規コンテナを作成するために既存のコンテナをクリックします。階層のトップのレベルにコンテナを作成する場合は、このボタンを無視してください。
3. ページの左下にある「コンテナの作成」ボタンをクリックします。



4. コンテナ名を入力してください。以下の名前に関する制約が適用されます:
  - フィールドが空であってはなりません。
  - 空白文字で開始、または終了することできません。
  - 英数字、数字、単一の空白、アンダースコア( \_ )、ダッシュ( - )、およびピリオド( . ) 文字を使用することができます。
5. 「保存」をクリックします。

コンテナの名前を変更する

1. 「構成」、をクリックし、名前を変更するコンテナに移動します。
2. ページの左下にある「選択されたオブジェクトの移動または名前を変更する」ボタンをクリックします。



別のコンテナに移動または名前を変更する

名前: Copy to Prod

コンテナ: /

/ +

名前

- public
- system

名前の変更 キャンセル

3. 名前ボックスにコンテナ名を入力し、「名前の変更」をクリックします。

#### コンテナの移動方法:

1. 「構成」、をクリックし、移動するコンテナに移動します。
2. コンテナの横のチェックボックスを選択し、「選択されたオブジェクトを移動または名前を変更する」をクリックします。



別のコンテナに移動または名前を変更する

名前: Copy to Prod

コンテナ: /

名前

- public
- system

名前の変更 キャンセル

3. コンテナの定義を以下をお好みで選択します。
  - コンテナテキストボックス内に / を入力します。
  - インタラクティブなナビゲーションコントロールを使用して、保存先コンテナに移動します。
4. 任意で、名前ボックス内にコンテナの新しい名前を入力します。
5. 「移動」をクリックします。

#### 複数のコンテナを移動する

- 横のチェックボックスをクリックし、単一のコンテナの移動と同じロジックを使用してください。

#### コンテナ内の全てのオブジェクトを選択、または削除する

- 一番上のチェックボックスをクリックします。

### 4.3.4 コンテナパーミッションの閲覧

関連するパーミッションを有する箇所で、コンテナのパーミッションを閲覧することができます（以下も参照してください [パーミッションのくみ](#)）。デフォルトでは、コンテナに対するユーザー自身のパーミッションを閲覧することができます。ロールのメンバーである場合、ユーザーがメンバーであるロールに与えられているパーミッションを確認することができます。ユーザーとロールの読み取りの特権を有する場合、コンテナに対しての他のユーザーとロールのパーミッションを確認することができます。

コンテナのパーミッションを閲覧する方法:

1. 「構成」をクリックします。
2. 以下のうちの1つを行ってください
  - コンテナ記録の横の「パーミッション」ボタンをクリックします。
  - コンテナを入力し、ページの右下にある「パーミッション」ボタンをクリックします。



ユーザーロール名 列は確認する権利を有するパーミッションを持つユーザーとロールを表示します。パーミッション列は、コンテナに対し特定のユーザーまたはロールが使用できるパーミッションの型を表示します。例えば、下のイメージはルート (/) コンテナのための認証済みのロールに対して使用することのできるデフォルトのパーミッションを表示しています。

| /のためのパーミッション  |               |
|---------------|---------------|
| ユーザーまたはロール名   | パーミッション       |
| authenticated | コンテナ: 読み取り    |
|               | 構成: 読み取り      |
|               | 資格情報: 使用      |
|               | キュー: 使用       |
|               | サービス: 使用      |
|               | 開数: 使用        |
|               | 証明書: 使用       |
|               | AS2 パートナー: 使用 |
|               | セキュリティ: 読み取り  |

各パーミッションの空の詳細に関しては、以下を参照してください: [パーミッションのしくみ](#)。

### 4.3.5 コンテナパーミッションの設定

コンテナのパーミッションを以下の条件を満たす場合変更することができます:

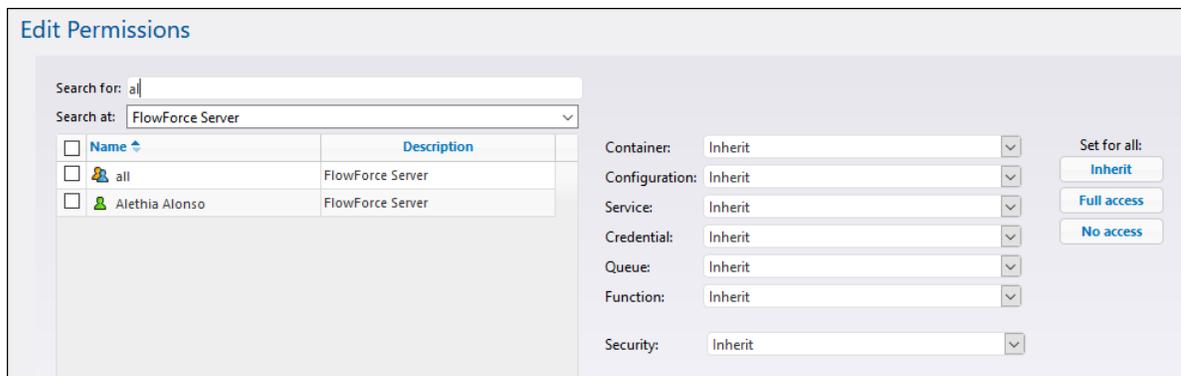
- ユーザー(またはメンバーであるロール)がセキュリティ/セキュリティの読み取りと書き込みパーミッションを変更する箇所に相対する親コンテナで有する場合。例えば、「Marketing」のコンテナの子であるコンテナ「ジョブ」のパーミッションを変更する場合、コンテナ「Marketing」のパーミッション セキュリティ/セキュリティの読み取りと書き込みを有する必要があります。コンテナ(以下を参照: [パーミッションのしくみ](#))。
- ユーザー(またはメンバーであるロール)がセキュリティの上書きの特権を与えられている(以下を参照: [特権の作動方法](#))。

## コンテナのパーミッションの変更:

1. 「構成」をクリックします。
2. 以下のうちの1つを行ってください。
  - コンテナ記録の横の「パーミッション」ボタンをクリックします。
  - コンテナを入力し、ページの右下にある「パーミッション」ボタンをクリックします。



3. 以下のうちの1つを行ってください。
  - リストされるユーザーとロールのパーミッションを変更する場合、相対するユーザーまたはロールの横の「変更」ボタンをクリックします。
  - リストされていないユーザーとロールのパーミッションを追加する場合、「パーミッションの追加」をクリックします。
4. 「パーミッションの編集」セクション内で、変更するパーミッションを有するユーザーを検索し、横のチェックボックスを選択します。FlowForce Server 内で作成されたユーザーを検索、または Directory Service が有効化されている場合、ドメインユーザーを検索します。FlowForce Server でドメインユーザーをインポートする方法に関しては [ドメインユーザーとロールのインポート](#) を参照してください。



#### 4.3.6 /public コンテナへのアクセスの制限

(トップレベルのルートコンテナの下に存在する) **/public** コンテナは、デフォルトでは FlowForce Server 内で使用することができます。定義済みのパーミッションなしで、FlowForce Server すべてのユーザーが全ての場所から FlowForce Server にアクセスし、データを保管するための場所として、使用されます。ですから、デフォルトでは **/public** コンテナには以下のパーミッションがあります。

## Permissions for /public

| User or Role name | Permissions  |                                       |
|-------------------|--|---------------------------------------|
| authenticated     | Container: <b>Read, Write</b><br>Configuration: <b>Read, Write</b><br>Credential: <b>Use</b><br>Queue: <b>Use</b><br>Service: <b>Use</b><br>Function: <b>Use</b><br>Security: Read      inherited from  /  | <input type="button" value="Change"/> |
| root              | Container: Read, Write      inherited from  authenticated<br>Configuration: Read, Write      inherited from  authenticated<br>Credential: Use      inherited from  authenticated<br>Queue: Use      inherited from  authenticated<br>Service: Use      inherited from  authenticated<br>Function: Use      inherited from  authenticated<br>Security: Read, Write      inherited from  / | <input type="button" value="Change"/> |

## 4.4 パスワードポリシー

このセクションは以下のトピックを含んでいます:

- [パスワードポリシーの動作方法](#)
- [パスワードポリシーの作成と割り当て](#)

### 4.4.1 パスワードポリシーの動作方法

FlowForce Servers は、管理者が複雑なユーザーパスワードを管理するパスワードポリシーを使用します。パスワードポリシーとは、有効になるためのユーザーパスワードが必要とする最低条件です (例えば、最低  $N$  文字のなど)。

パスワードポリシー内で定義することのできる、パスワードの複雑度のルールは、以下の通りです:

- パスワードの合計の長さの最小数 (つまり、パスワードは、有効であるためには、少なくとも  $N$  文字の長さである必要があります)。
- パスワードが含む必要がある英字の最小数
- パスワードが含む必要がある数字の最小数

(ユーザー、ロール、特権の管理の特権があると想定して必要に応じて、パスワードポリシーを必要な数定義することができます。パスワードポリシーを定義すると、FlowForce ユーザーに割り当てることができます。ユーザーアカウントは、1度1つのパスワードポリシーを有することができます。

ユーザーがパスワードの変更をリクエストすると、システムは、ユーザーのパスワードポリシー内で定義されたものをチェックします。システムは、新しいパスワードがユーザーのパスワードポリシー内で定義されたパスワードの複雑度を満たすかをチェックします。パスワードがパスワードポリシー内で定義された複雑度の条件を満たさない場合は、パスワードの変更は否定され、システムは関連したメッセージを表示します。

管理者がユーザーのパスワードを変更すると、FlowForce Server は、パスワードポリシーを強制しません。また、パスワードポリシーが変更されても、既存のパスワードは影響を受けることはありません。後者の場合、ユーザーが既存のパスワードを変更しようと試みると、パスワードポリシーが強制されます。

デフォルトでは、FlowForce Server は、パスワードの複雑度を強制しない空のパスワードポリシーを含むパスワードを含んでいます。FlowForce Serverは、明示的にデフォルトのパスワードポリシーを、カスタマイズされたパスワードポリシーを有さないユーザーアカウントに割り当てます。デフォルトのパスワードポリシーを変更することはできません。

以下も参照してください

- [パスワードポリシーの作成と割り当て](#)

### 4.4.2 パスワードポリシーの作成と割り当て

新規パスワードポリシーの作成:

1. 「管理」をクリックします、「パスワードポリシー」をクリックします。
2. 「ポリシーの作成」をクリックします。

## パスワードポリシーの作成

ポリシー名:

### パスワードポリシー

合計の長さ: 少なくとも以下の数量が必要です  文字 

英字: 少なくとも以下の数量が必要です  英字 

数字: 少なくとも以下の数量が必要です  数字 

3. 必要とされるパスワードポリシールールを入力し、「保存」をクリックします。現在のユーザーのリストが定義されたポリシー下で使用することができます。新規ポリシーに割り当てられるユーザー記録をクリック、「割り当て」をクリックします。

### メンバー

利用可能なユーザー

| <input checked="" type="checkbox"/> 名前                   | 現在のポリシー    |
|--|------------|
| <input checked="" type="checkbox"/> Alethia Alonso       | デフォルトのポリシー |
| <input checked="" type="checkbox"/> Klaus Mauer          | デフォルトのポリシー |
| <input checked="" type="checkbox"/> Natsuo Shinohara     | デフォルトのポリシー |
| <input checked="" type="checkbox"/> default for new user | デフォルトのポリシー |
| <input checked="" type="checkbox"/> root                 | デフォルトのポリシー |
| <input checked="" type="checkbox"/> 新しいユーザーのデフォルト        | デフォルトのポリシー |

ポリシーのメンバー '新規パスワードポリシー'

次も参照してください

- [パスワードポリシーの動作方法](#)

## 5 ジョブの構成

このセクションは、FlowForce Server ジョブの構成に適用することができる手続きとコンセプトを含んでいます。

## 5.1 資格情報

資格情報オブジェクトは、認証に関する情報を保管します。これはFlowForce Server ジョブが作動する箇所でのオペレーティングシステム上のユーザーアカウントに関連するユーザー名とパスワードの組み合わせですがHTTP またはFTP 資格情報のセットであることもできます。

型の証明書は各ジョブで必要とされています。特定のオペレーションシステムユーザーとしてジョブを実行することを可能にします。具体的には、FlowForce Server 内で [ジョブ](#) を作成すると、ジョブを実行するユーザーアカウントの資格情報を適用する必要があります。オペレーティングシステム上でユーザーアカウントに十分な権限が存在しない場合、ジョブの実行を成功することはできません。FTP サーバーへの認証が必要な内蔵の [FTP](#) 関数を呼び出すために必要です。ファイルウォッチリガモ、パスワード資格情報を必要とします。

FlowForce Server 内では、資格情報をローカルの資格情報として参照される新規のジョブを作成する都度、または、スタンドアロン（再利用可能な）資格情報オブジェクトとして定義することができます。後者の場合、ジョブを作成する際に再度入力する代わりに定義済みの資格情報を参照することができます。変更する都度1つの場所で更新することができるため、スタンドアロン資格情報は便利です。すなわち、変更はスタンドアロン資格情報を更新する際には、その資格情報の参照を使用するすべてのジョブに影響を与えることを意味します。

### メモ

- 関連するパーミッションが与えられている場合、ジョブからの資格情報をユーザーは参照することができます。資格情報を特定のコンテナからユーザーまたはロールにアクセス可能にするには、管理者がユーザーまたはロールに資格情報 - 使用パーミッションを許可する必要があります。（[パーミッションのくみ](#) を参照してください）。
- オペレーティングシステムのログイン関数（クリアテキスト/パスワードが送信される必要があるため、FlowForce Server データベース内で暗号化を基に戻せる状態で保存することのできるパスワードとして保管されます。管理者はFlowForce Server のデータベースファイルへのアクセスが制限されていることを確認してください。 [FlowForce Server アプリケーションデータ](#) を参照してください）。

FlowForce Server ジョブとしてマッピングを実行するために MapForce と MapForce Server にライセンスを供与した場合、デザイン時に FlowForce Server 内のマッピングだけでなく、MapForce 内に資格情報オブジェクトを作成することができます。MapForce 内から FlowForce Server に作成された資格情報をオブジェクトに所属するマッピングとして、または、個別のオブジェクトとしてデプロイすることができます。ユーザー名およびパスワードなどの機敏なデータを保管するためにデプロイされた資格情報は必要ありません（これは任意に行うことができます）。

MapForce 内で資格情報を作成し、FlowForce Server にデプロイする詳細に関しては、MapForce ドキュメントを参照してください（<https://www.altova.com/ja/documentation>）。FlowForce Server 内での資格情報の作成及び使用方法に関しては [資格情報の定義](#) と [ジョブからの資格情報の参照](#) を参照してください。マッピングジョブ内で資格情報を設定、または、オーバーライドするには、[マッピング関数内の資格情報](#) を参照してください。

### 5.1.1 資格情報の定義

複数のジョブ全体で再利用することのできるスタンドアロンオブジェクトとして資格情報を定義することができます。

#### 必要条件

- コンテナ - 読み取り、書き込みおよび構成 - 読み取り、書き込みパーミッションが資格情報を保管するコンテナで与えられていることを確認してください。

## 資格情報の定義

### 資格情報オブジェクトの作成方法:

1. 資格情報を保管するコンテナに移動します。
2. 「作成 | 資格情報の作成」をクリックします。

資格情報名:

資格情報の詳細:

---

### 資格情報

証明書の型:  パスワード  OAuth 2.0

ユーザー名:

パスワード:

HTTP のために使用を許可:

FTP のために使用を許可:

ジョブ実行のために使用を許可:

資格情報フィールドは以下の通りです。

|                  |  |
|------------------|--|
| 資格情報名            | 必須のフィールドです。FlowForce Server 内で資格情報が識別される名前です。  |
| 資格情報の説明          | 資格情報についての情報を提供する追加の説明です。   |
| ユーザー名            | 必須のフィールドです。ユーザーの名前はこの資格情報に関連しています。例えば、Windows オペレーティングシステム上でユーザーアカウントを識別するために資格情報が使用される場合、Windows ユーザーアカウント名を入力してください。Windows ドメイン内でユーザー名を指定する場合、username@domain のフォームを使用します。<br><br>HTTP または FTP のために資格情報が使用される場合、これは HTTP または FTP ユーザー名である場合があります。以下を参照。 |
| パスワード            | 資格情報のパスワードを指定します。使用される個所でパスワード無しユーザー名のみが必要とされる場合、パスワードは空の文字列の場合があります。  |
| HTTP のための使用を許可する | 基本の HTTP 認証を必要とする資格情報が Web サービスを呼び出すジョブ内で参照されている場合このチェックボックスを選択します。  |

|                  |  |
|------------------|--|
| FTP のための使用を許可する  | <code>/system/ftp</code> 関数を使用し FTP サーバーに接続するジョブ内で 資格情報が参照される場合このチェックボックスを選択します。   |
| ジョブ実行のための使用を許可する | 資格方法がオペレーティングシステムユーザーアカウントを識別する場合このチェックボックスを選択します。実行に成功するために、資格情報を使用するジョブではすべてこの使用が有効化されています。<br><br>資格情報により識別されるユーザーアカウントがオペレーティングシステム上で十分な権限を有していることを確認してください。例えば、ディレクトリに書き込むジョブ内で資格情報が参照されている場合、ユーザーアカウントはそのディレクトリに書き込む権限を有している必要があります。 |

### 5.1.2 OAuth 2.0 資格情報

型パスワードの資格情報に加え、**OAuth 2.0** 認証詳細である資格情報オブジェクトを作成することができます。OAuth 2.0 認証が必要とされる Web サービスを呼び出す FlowForce Server ジョブ内の OAuth 2.0 資格情報を使用することができます。

パスワード資格情報と同様の用法で OAuth 資格情報を作成することができます。[資格情報の定義](#)を参照してください。他の FlowForce Server オブジェクト同様、対応するパーミッションが存在する場合のみユーザーは OAuth 資格情報をビューアクセスすることができます。[パーミッションのくみ](#)を参照してください。

OAuth 2.0 資格情報オブジェクトに関するフィールドは下にリストされています。これらの値を取得するには(例えば、Google API Console、Facebook API、Bitbucket API などの) Web サービスプロバイダーに登録する必要があります。

|               |   |
|---------------|---|
| 認証エンドポイント     | FlowForce Server が自動化フローを開始する URI を指定します。Web サービスプロバイダーを使用して登録後この値を取得することができます。   |
| トークンエンドポイント   | FlowForce Server がトークンフローを開始する URI を指定します。Web サービスプロバイダーを使用して登録後この値を取得することができます。  |
| クライアント ID     | クライアントアプリケーションの ID です(この場合、FlowForce Server)。Web サービスプロバイダーを使用して登録後この値を取得することができます。   |
| クライアントの秘密     | クライアントアプリケーションに関連する秘密です。Web サービスプロバイダーを使用して登録後この値を取得することができます。  |
| スコープ          | プロバイダーにより必要とされる場合のクライアントアプリケーションのスコープです。Web サービスプロバイダーを使用して登録後この値を取得することができます。  |
| トークンエンドポイント認証 | OAuth 2.0 認証サーバーの多くは POST リクエストヘッダーに認証の詳細が送信されることを必要とします。これもデフォルトでドロップダウンリストから選択された値です。<br><br>OAuth 2.0 認証サーバーの一部は POST リクエストのボディ内のみで認証の詳細を受け入れる場合があります。このような認証サーバーのために POST リクエスト ボディ内 でドロップダウンリストから値を選択してください。 |
| アクセストークン      | 認証サーバーにより返されるアクセストークンです。  |
| 更新トークン        | 認証サーバーにより返される更新トークンです。「アクセストークン」の期限が切れると必要となります(下を参照)。稀なケースとしてアクセストークンの期限が切れない場合、これは必要ありません。  |

「以下のための使用方法の許可」チェックボックスは OAuth 2.0 だけでなく FlowForce、内のすべての資格情報に適用されます。[パスワードの資格情報](#)のために説明されている同様の意味を持ちます。HTTP のために使用する OAuth 2.0 資格情報のために「HTTP

のための使用を許可する」チェックボックスが選択されていることを確認してください。それ以外の場合、ジョブはランタイムエラーと共に失敗します。「資格情報は必要とされる使用方法をサポートしません」（このメッセージまたは [FlowForce ログ](#) 内で表示されているようなテキストが表示されます）。

### 5.1.3 ジョブから資格情報を参照する方法

[資格情報の定義](#) で説明される通り、資格情報オブジェクトが作成されると、資格情報が必要とされる多種のコンテキストで参照することができます。例:

- FlowForce Server が作動する箇所でオペレーティングシステム上のユーザーアカウントを識別するパスワード 資格情報が作成されます。この資格情報を複数のジョブから後に参照することができます。この例は下で説明されています。
- FTP ユーザー名とパスワードを識別するパスワード 資格情報が作成されます。FTP 関数を呼び出すジョブからこのような資格情報を参照することができます。

次のサンプルは、FlowForce Server が作動する箇所でオペレーティングシステム上のユーザーアカウントを識別するパスワード資格情報を参照する必要がある共通のケースを説明しています:

1. 型パスワードの資格情報を[資格情報の定義](#)内で説明される通り作成します。
2. 新規のジョブを作成、または、既存のジョブを編集します。
3. 「資格情報」で「既存の資格情報を選択」をクリックし、定義済みの資格情報レコードを参照します。

The screenshot shows the '資格情報' (Credentials) configuration page. It features two radio buttons for selecting the source of credentials: '既存の資格情報を選択' (Select existing credentials) which is selected, and 'ローカルの資格情報を定義' (Define local credentials). Below this is the 'キュー設定' (Queue Settings) section with input fields for '実行間の最短時間' (0 seconds) and '並列実行の最大数' (1 instance). On the right, a breadcrumb path shows 'public' and a list of credential names with 'my.credentials' highlighted.

資格情報レコードを含むジョブが存在する場合、資格情報オブジェクト自身として参照することができます。例:

### 資格情報

資格情報によりジョブを実行:  既存の資格情報を選択: /public/JobWithLocalCredentials  
 ローカルの資格情報を定義: / public

### キュー設定

実行間の最短時間:  秒

並列実行の最大数:  インスタンス

名前 ▲

- JobWithLocalCredentials
- my.credentials

この場合、(ローカル資格情報を持つ埋め込まれジョブの資格情報はメインジョブの資格情報として使用されます。資格情報はリンクされており、コピーされません。ローカルで定義されている資格情報を埋め込まれているジョブで変更すると、メインジョブに伝達されます。

## 5.2 ジョブの作成

このピクでは、FlowForce Server Web 管理インターフェイスから、ジョブを作成する方法について説明されています。一般的な手順は、使用することのできる設定とジョブの構成をよりよく理解するためのものです。しかしながら、リストされている全てのステップおよび設定が、実際のシナリオに適用できるとは限りません。定義として、ジョブは柔軟性を持つプロセスであり、1つの結果を得るために、幾通りかの方法があります。ジョブとの作業により行うことのできる多様なアイデアを得るために、このドキュメントには他の[ジョブの構成の例](#)が含まれています。

### 必要条件

- ジョブが作成される[コンテナ](#)のために、以下のパーミッションが与えられていることを確認してください（：[コンテナのパーミッションの関](#)以下を参照）。
  - [コンテナ](#) - 読み取り、書き込み
  - [構成](#) - 読み取り、書き込み

### ヒント

- ジョブを作成する前に、ジョブが実行されるのオペレーティングシステムユーザーアカウント資格情報を保管することが奨励されています（以下を参照：[資格情報の定義](#)）。
- ジョブを作成する際、ジョブ間、または、ステップ間で値を渡すことを想定している場合、一般的なエラーを避けるために、[FlowForce 式](#)を参照してください。

### ジョブを作成する

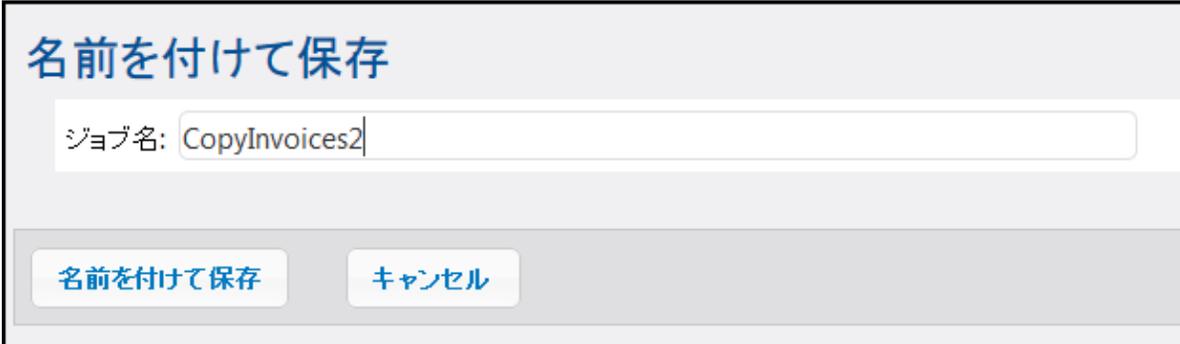
1. 「構成」をクリックし、ジョブを作成するコンテナに移動します。
2. 「作成」をクリックし、「ジョブの作成」をクリックします。
3. ジョブ名、および、任意でジョブの説明を入力します。
4. ジョブがランタイムで値を渡す場合、必要とされるジョブ入力パラメータを作成します（以下を参照：[入力パラメータの管理](#)）。
5. 実行ステップで、ジョブのステップを追加します（以下を参照：[ステップの管理](#)）。ジョブには少なくとも1つのステップが必要です。
6. ジョブの最後のステップが結果を返すと、また、結果を他のジョブで使用する場合、実行の結果から戻り値の型を選択します。（以下を参照：[ジョブの戻り値の型を宣言する](#)）。
7. FlowForce Server で返された結果をキャッシュする場合、キャッシュの優先順位を指定してください（以下を参照：[ジョブの結果をキャッシュする](#)）。
8. トリガーから、ジョブを実行するトリガーを追加します（以下を参照：[トリガーの管理](#)）。ジョブがWeb サービスとして公開される場合、トリガーを追加する必要はありません。
9. ジョブがWeb サービスとして公開される場合、「このジョブをHTTP を介して利用可能にする」チェックボックスをクリックします（以下を参照：[ジョブをWeb サービスとして表示する](#)）。
10. 資格情報から、既存の資格情報の記録、または、ローカルの資格情報を指定します。（以下を参照：[資格情報](#)）。
11. ジョブが返した結果を他のジョブで使用する場合、または、Web サービスとして公開する場合、ジョブのキャッシュ設定を定義してください（以下を参照：[ジョブの結果をキャッシュする](#)）。
12. 任意で、ジョブのキュー設定を定義します（以下を参照：[キュー設定の定義](#)）。
13. 「保存」をクリックします。FlowForce Server は、入力された情報を検証し、必要であれば、注目が必要なフィールドを赤でハイライトします。

## 5.3 ジョブの複製

既存のジョブのコピーを必要であれば作成することができます。コピーされたジョブは（実行ステップ、トリガー、キャッシュの設定など）元のジョブと同じ性質を有します。ジョブの複製により時間を節約することができます。例：既存のジョブをテンプレートとして素早くジョブを作成する場合など。

既存のジョブのコピー：

1. 既存のジョブを開き、そして、ページの下にある、「名前を付けて保存」ボタンをクリックします。
2. 新規ジョブの名前を入力して「名前を付けて保存」ボタンをクリックします。



**メモ** 既存のジョブの資格情報がジョブ内で、ローカルに定義されている場合、FlowForce は、セキュリティ上の理由から、もう一度パスワードを入力するように問います。スタンドアロン資格情報として資格情報が定義されている場合、このステップは、必要ありません。スタンドアロンとローカルの資格情報の違いに関する詳細は、以下を参照してください [資格情報](#)。

**メモ** ジョブを複製する際に、特定のジョブコンポーネントが競合を引き起こす場合、FlowForce は、エラーを表示し、ジョブの複製を行いません。例えば、Web サービスを含むジョブを複製しようと試みると、サービスが既に元のジョブにより使用中のため、複製することはできません。この場合、Web サービスのURL を変更するか、または、完全に削除してください。

複製されたジョブは、既存のジョブとして同じコンテナに保存されます。異なるコンテナに移動する場合は、2つのステップから構成されるオペレーションとして行うことができます：

1. 最初に、既存のジョブを上記のように新しい名前で作成します。ページを移動することなく同じジョブのコピーを複数作成することもできます。
2. 親コンテナページに移動し、1つまたは複数のジョブを選択し、「選択されたオブジェクトを移動」をクリックします。

## 5.4 入力パラメーターの管理

FlowForce Server [ジョブ](#) のコンテキストで、入力パラメーターは、どのジョブが特定の方法で実行されるかを表したジョブに与えられた任意の情報を意味します。ジョブの入力パラメーターは、関数引数(プログラム言語の観点から類似しています)。入力パラメーターは多種の型(例えば、ファイルまたはディレクトリ、パス、テキスト、番号、ブール値、その他)であることができます。一部のジョブの型に対しては、FlowForce Server は、入力パラメーターを自動的に作成します。

### メモ

- FlowForce Server は、自動的にファイルシステムまたはHTTP トリガーを使用するジョブに **triggerfile** という入力パラメーターを追加します([トリガーの管理](#)も参照してください)。**triggerfile** パラメーターは、トリガーをアクティブ化する、削除してはならない、ファイル名を含んでいます。
- ジョブがパラメーターを使用し Web サービスとして公開されている場合、FlowForce Server は、リクエスト内のアプリケーションの呼び出しにより与えられるパラメーターを期待します([ジョブを Web サービスとして表示する](#)も参照してください)。

入力パラメーターは以下のフィールドで構成されています。

|       |  |
|-------|--|
| 名前    | 必須フィールドです。入力パラメーターの名前を指定します。ジョブの実行ステップ、から後このパラメーターを参照する可能性があり、詳細を記した名前をできるだけ使用してください。<br><br>入力パラメーター名は文字で開始し以下の文字のみを含むことができます: a-z、A-Z、0-9、および`_`。  |
| 型     | 必須フィールドです。以下のうちの1つである入力パラメーターのデータ型を指定します: <ul style="list-style-type: none"> <li>文字列</li> <li>ファイルとしての文字列</li> <li>ディレクトリとしての文字列</li> <li>ファイルとしての文字列またはディレクトリ</li> <li>ストリーム</li> <li>数値</li> <li>ブール値</li> </ul> 期待されるパラメーターの値のデータ型に従いデータ型を選択してください。 |
| デフォルト | 任意のフィールド。パラメーターのデフォルトの値を指定します。この値はランタイムジョブの呼び出し元により値が指定されない場合使用されます。   |
| 説明    | 任意のフィールド。パラメーターの目的を説明します。現在のジョブを他のジョブの実行ステップとして使用する場合、この説明は、パラメーター名の横に表示されます。  |

以下のボタンをパラメーターを管理するために使用してください。

|   |               |
|---|---------------|
|  | パラメーターを追加します。 |
|  | パラメーターを削除します。 |
|  | パラメーターを複製します。 |

|   |                   |
|---|-------------------|
|  | パラメーターを上下に移動します。  |
|  | 前の削除アクションを元に戻します。 |

## 5.5 ステップの管理

ステップは、FlowForce Server ジョブが実際に行う内容を定義します(例えば、ファイルの削除、MapForce マッピングの実行、または電子メールの送信など)。最も簡単な形式としては、ステップは失敗または成功した結果を持つオペレーションで、実行されるために関数が必要とします(以下を参照: [実行ステップ](#))。しかしながら、ループとして他のステップを条件付きで実行する方法を与えるステップも存在します(以下を参照: [選択ステップ](#)、[エラー/成功の処理ステップ](#) および [For-Each ステップ](#))。同じジョブ内で必要な数だけステップを作成し、実行される順序を設定することができます。

ステップを追加するには、ステップの型に対応するボタンをクリックします。

### 実行ステップ

新規実行ステップ
新規チョイス ステップ
新規 For-each ステップ
新規エラー/成功処理ステップ

次のステップの型を作成することができます。

|                      |   |
|----------------------|---|
| <b>実行ステップ</b>        | <p>型「実行ステップ」のステップにより特定のFlowForce 関数を実行することができます。FlowForce Server により提供されるMapForce マッピングをデプロイする、または変換を行う、他のジョブのステップを実行する<b>ビルトイン関数</b>を含む関数を使用することができます。</p>   |
| <b>選択ステップ</b>        | <p>型「選択ステップ」のステップにより、どの条件下で他のジョブステップが実行されるかを定義することができます。選択ステップには以下の構造があります:</p> <pre>When {some expression}   Execute (some step) Otherwise   Execute (some other step)</pre> <p>各「選択ステップ」の下では、他の選択ステップ(サブ条件)をネストすることができます。例</p> <pre>When {expression}   When {expression}     Execute (step)   Otherwise     Execute (step) Otherwise   Execute (step)</pre> <p>定義することのできる条件ステップの数量に制限はありません。</p> <p>When / Otherwise ペアの中では、FlowForce Server は、true である条件のみを実行します。他の条件は無視されます。</p> |
| <b>For-Each ステップ</b> | <p>型「For-Each ステップ」の<b>ステップ</b>により、シーケンス中を反復し、実行ステップを無制限に繰り返すことができます(例えば、ディレクトリ内のファイルのリスト)。「for-each」ステップの構成は以下の通りです:</p> <pre>For each item in sequence {sequence expression}   Execute (step)</pre> <p>FlowForce は、シーケンス式の全てのアイテムをループして完了するまでステップを実行します。</p>  |

|               |   |
|---------------|---|
| エラー/成功の処理ステップ | ジョブのステップに失敗すると、FlowForce Server は、ジョブを中断します。型「エラー/成功の処理ステップ」の <a href="#">ステップ</a> を使用して、ジョブを完全に完了する前に、クローンアップアクションを実行します（以下を参照: <a href="#">ステップのエラーの処理</a> ）。 |
|---------------|---|

ステップを追加するには、ステップの型に対応するボタンをクリックします。

### 実行ステップ

新規実行ステップ
新規チョイス ステップ
新規 For-each ステップ
新規エラー/成功処理ステップ

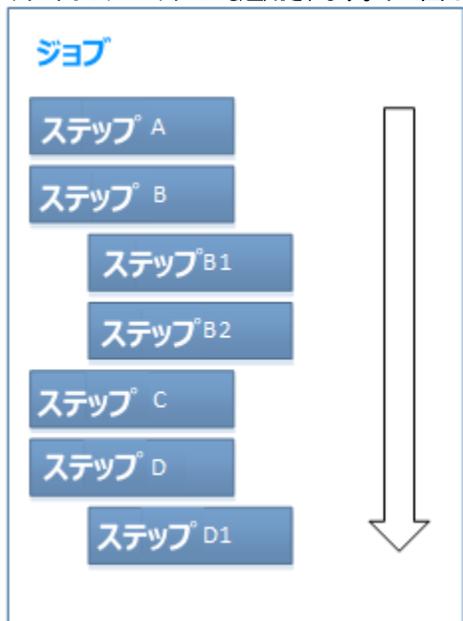
以下のボタンをステップを管理するために使用してください。

|   |                 |
|---|-----------------|
|    | ステップを追加する。      |
|    | ステップを削除する。      |
|    | ステップを複製する。      |
|   | 上下に移動します。       |
|  | 前の削除アクションを元に戻す。 |

## 5.6 順番にステップを処理する

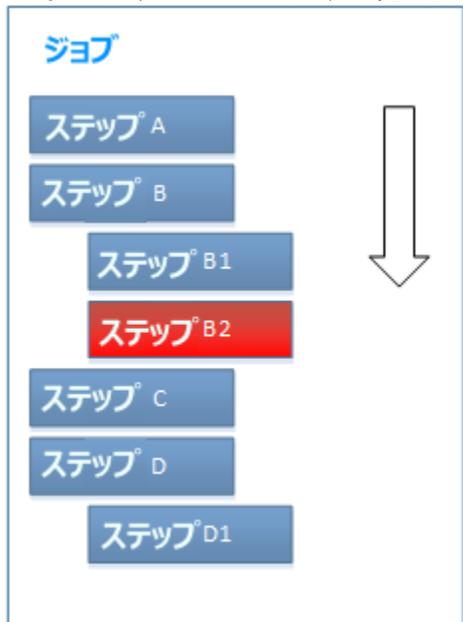
FlowForce ジョブを最も簡単な方法で作動するには「実行ステップ」のシングルステップとして追加することです。このようなステップは、FlowForce ビルトイン関数、シェルコマンド、MapForce マッピング、StyleVision 変換、RaptorXML 関数、またはスクリプトを呼び出すために使用することができます。

複数の実行ステップをジョブに追加する場合、FlowForce は（一番上の）最初から最後のステップを順番に処理します。このルールは、ステップ内のサブステップにも適用されます。下の図はこのシナリオを表しています。



順次処理

デフォルトでは、FlowForce にエラーが発生した場合、エラーが発生した箇所で、処理は停止し、その後のステップは、実行されません。



### 順次処理エラー

ステップが失敗しても、ジョブ全体の実行を停止しない場合があります。この場合、ジョブの実行が停止されないように実行ステップと呼ばれる関数を構成します。その後のステップ全ては、ステップが失敗しても、通常通り処理されます。次の関数を呼び出すステップのみこの振る舞いを与えることができます。ご注意ください！

- [/system/filesystem](#) の下の全ての関数。
- [/system/ftp](#) の下の全ての関数。
- システム/[/system/mail](#) 関数
-  [/system/shell/commandline](#) 関数。

詳細に関しては、各関数(特に「エラー時は中断」/「ラメータ」の説明を参照してください。

順次処理は、FlowForce ジョブを処理するための一つの方法で、ニーズに常に応えるとは限りません。高度な処理に関しては、以下を参照してください！

- [条件付きでステップを処理](#)
- [ステップのエラーの処理](#)

## 5.7 条件付でステップを処理

条件に従いステップを処理するには、型「選択」のステップを追加します。条件に従いステップを処理するには、型のステップを追加します。このようなステップは次の2つの部分から構成されています: 「When (の場合)」部分と「Otherwise」部分。



条件付き処理を含むジョブの抽象的な表現

上のイメージで表示されている通り、「When」パート「Otherwise」パートは、実行ステップを必要とします。また、「When」パートは式を必要とします。この式は、ジョブが実行される際、2つのパートのうちどのパートを実行するかを決定します。具体的には「When」式がブール値の `true` を評価すると、ステップ C が実行されます。それ以外の場合、ステップ D が実行されます。抽象的なジョブの処理のシーケンスは、以下のとおりです (実行エラーが存在しない前提で):

1. ステップ A
2. ステップ B
3. ステップ C または、式により D
4. ステップ E
5. ステップ F

例えば、下に示されている通り、FlowForce は式「fileexists」がブール値の `true` を評価すると、ビルトイン `delete` 関数を実行します。

**選択**

When fileexists

+  
 ▲ 関数の実行 /system/filesystem/delete

パラメーター: パス: c:\temp\myfile.xml

エラー時には中断: +

作業用ディレクトリ: +

「fileexists」は、前のステップでまだ入力パラメーターとして宣言されている必要があります。例:

ステップの結果を割り当て fileexists

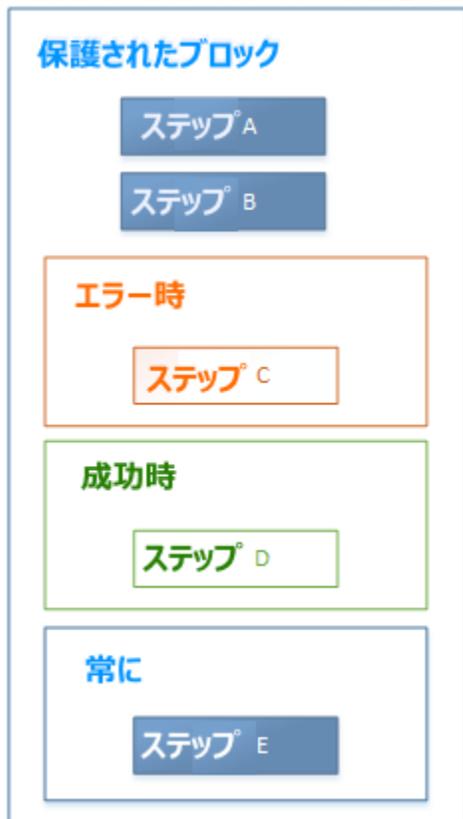
それ以外の場合、構文検証エラーを表示します:

When fileexists 

FlowForce 式に関する概要に関しては [FlowForce 式](#) を参照してください。FlowForce ジョブ内での条件付きの処理の例に関しては、以下を参照してください。例: [パスが存在するか確認する](#)。

## 5.8 ステップのエラーの処理

ジョブのステップに失敗すると、ジョブは失敗したとみなされます。(ログ、または、電子メールの送信などのためにジョブを完全に終了する前に、クリーンアップのアクションを行うには、型「エラー成功の処理」を作成することができます。「保護されたブロック」として参照されるステップを処理するエラー成功により、ステップの実行を保護することができます。次の図は、保護されたブロックの構成を表示しています。



保護されたブロックの構造

保護されたブロック内の1つのステップに失敗すると、エラーハンドラーは、ジョブが終了される前に何か起こるかを管理します。エラーハンドラーは以下の内の1つです:

- 成功時 (保護されたブロック内の全てのステップに成功する場合に実行されます)。
- エラー時 (保護されたブロック内の1つのステップに成功する場合に実行されます)。
- 再試行時 (このハンドラー保護されたブロックのカウントの再試行 オプションが0 より大きな値を設定されている場合に作動します。保護されたブロック内のステップが失敗すると再試行が行われます)。詳細に関しては下の「エラー時に再試行」セクションを参照してください。
- 常に (保護されたブロック内のステップの成功に関わらず実行されます)。

保護されたブロックが実行を完了すると、FlowForce Server は、出力をベースにした定義されたハンドラーを実行します。例えば、上に表示される図では、保護されたブロックは、ステップ A とステップ B です。また、エラー処理の論理は以下の通りです:

- A が失敗すると、A、C および E が実行されます。
- B が失敗すると、A、B、C および E が実行されます。
- A と B に成功すると、A、B、D および E が実行されます。

実際には、処理する各ジョブのため3つすべてのハンドラーの型を定義することはできません。最も一般的なシナリオは「エラー時」と「常に」ハンドラーのみを定義することです。例えば、下のイメージは「エラー時」と「常に」を使用した単純型の保護されたブロックを表示しています。

The screenshot displays a configuration window for a job with five steps:

- Step 1:** Function execution `/system/shell/commandline`. Parameters: Command: `script.cmd`, Error handling:  (plus icon), Working directory: `C:\script`. Result assignment: `output` (plus icon).
- Step 2:** Execution on error/success. Trigger: `エラー時` (dropdown), Action: `do`.
- Step 3:** Function execution `/system/mail/send`. Parameters: From: `flowforce@localhost`, To: `to@example.org`, Subject: `The job instance{instance-id()}has failed`, Body:  (plus icon), Attachments:  (plus icon), Error handling:  (plus icon). Result assignment: `名前` (dropdown), Boolean value (plus icon).
- Step 4:** Execution on error/success. Trigger: `常に` (dropdown), Action: `do`.
- Step 5:** Function execution `/system/shell/commandline`. Parameters: Command: `log_message.cmd`, Error handling:  (plus icon), Working directory: `C:\script`.

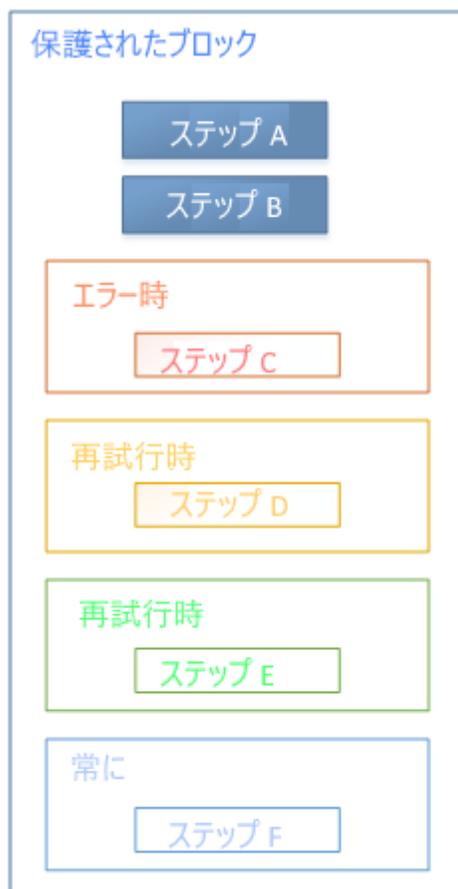
最初のステップは、`\system\shell\commandline` 関数を呼び出して、`C:\scripts` ディレクトリからスクリプトを実行します。このステップの実行はつのハンドラーにより保護されています。「エラー時」と「常に」。「エラー時」ハンドラーは最初のステップに失敗するとトリガーされます。具体的には、最初のステップに失敗すると、エラー処理ステップが件名ライン内に失敗したジョブインスタンスのID を含む電子メールを送信します。「常に」ハンドラーは最初のステップの成功に関わらず無条件で実行されます。このハンドラーはメッセージをthe `C:`

`\scripts` ディレクトリからのスクリプトを実行してログします。上記に類似したサンプルに関しては [エラー処理をジョブに追加する](#) を参照してください。

### 5.8.1 エラー時に再試行する

実行が何らかの理由で失敗した場合、ステップをもう一度実行する必要がある場合があります。例えば、ジョブがオンラインファイルからのデータを入力して取る場合にネットワークがダウン、または、サービスがジョブが作動する時に使用できないケースがあります。エラーの場合に再試行を可能にするために FlowForce Server は **再試行のカウント オプション** を提供します。デフォルトではオプションは **0** です。これは保護されたブロックが一度のみ実行され、再試行が行われないことを意味します。**0** より大きな値にこのオプションを設定すると保護されたブロック全体の実行が可能な限り試みられます。各再試行で FlowForce は前の結果を破棄し、保護されたブロックが最初から実行されます。再試行の最大数に到達しても、保護されたブロックがエラーに遭遇する場合、ジョブは失敗したものとみなされます。エラーは外部のブロックに伝達され、再試行の最大数に到達し、保護されたブロックの実行が失敗したことがレポートされます。

次の例を考慮してください！



上の図は2つの保護されたステップにより構成されるジョブについて説明しています（具体的にはエラー処理はステップ A と B にとって有益です）。「再試行カウントの設定」オプションは 2 に設定されています（これは最初の実行が成功しない場合、2度再試行が行われることを意味します）。

ジョブが最初に実行されると、B ステップが失敗すると仮定します。これはステップ A、B、F が実行されることを意味します。再試行が2回残っているため、エラーはまだ報告されていません。

2回目のトライで、B がもう一度失敗します。今回はステップ A、B、D と F が実行されます。唯一の違いは「再試行時」ブロックも実行されていることです。

最後に3番目（そして最後）の再試行では B にエラーは発生しません。今度はステップ A、B、D、E、および F が実行されます。ジョブの結果は成功としてみなされます。

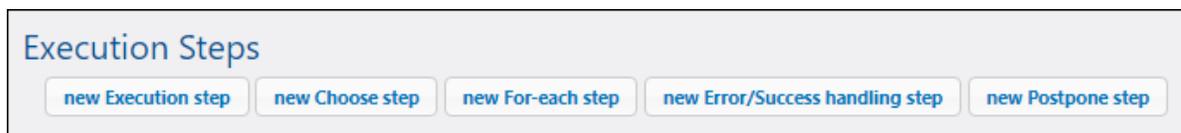
ステップ B が8度目のトライで失敗すると、ステップ A、B、C、D、および F が実行されるはずでした。ジョブの結果は失敗としてみなされず。

再試行が実行される都度（A と B の両方ではなく）ステップ B のみを実行する必要がある場合、ステップ B を保護されたブロックで囲み、再試行の「カウント オプション」をこのレベルのみに設定します。外部の保護されたブロックは標準の再試行カウント = 0 を持つことができます。

FlowForce [ログ](#) 内でレポートされるジョブインスタンスの再試行の数値。更にランタイムでの再試行の数値を取得し処理する必要がある場合、[retry-count](#) 式関数を呼び出します。最も内部の保護されたブロックのために発生した再試行の数値を返します。

エラーが発生した場合複数回トライされる実行ステップを追加する方法:

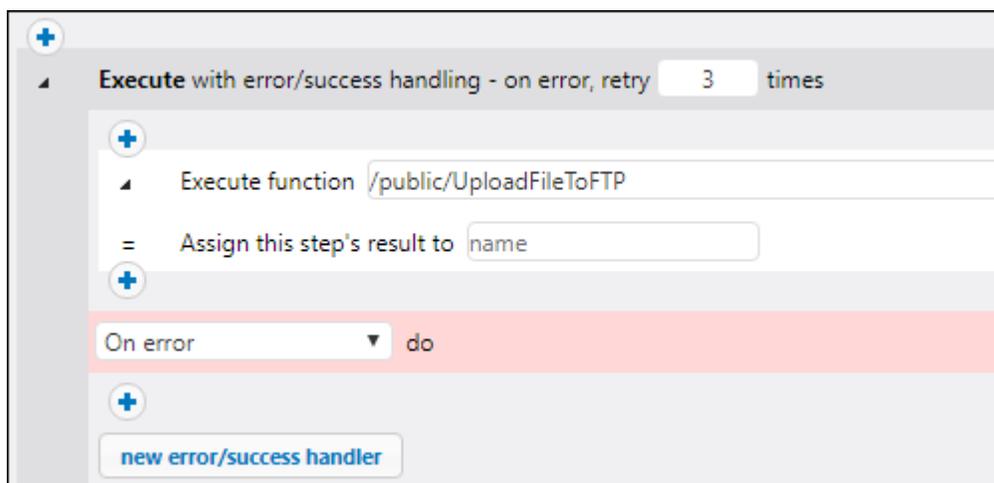
1. ジョブ構成ページで新規のエラー/成功処理ステップをクリックします。



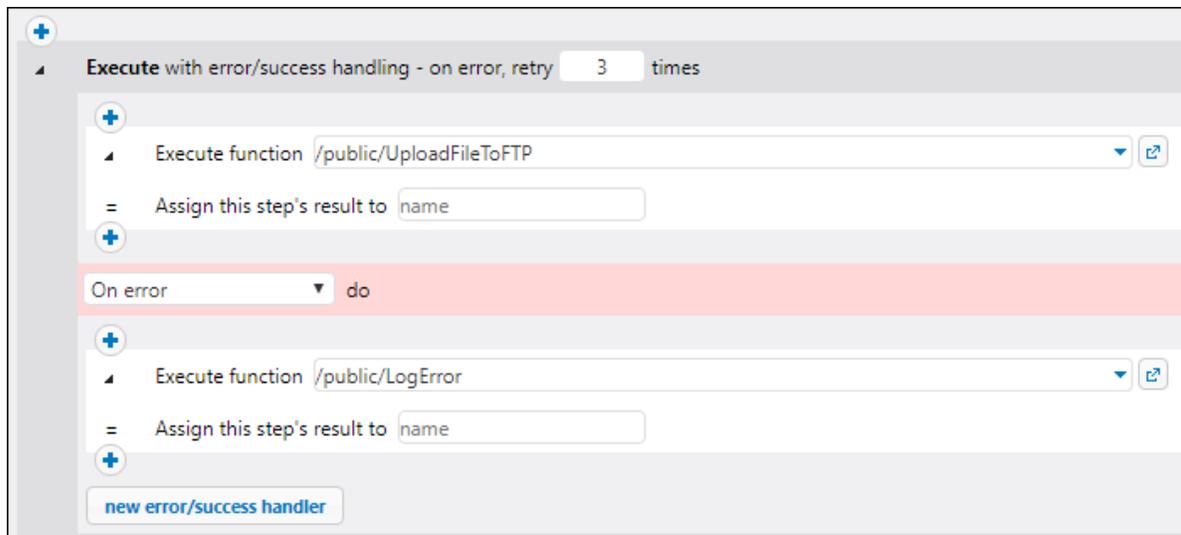
2. 「エラー/成功処理を使用して実行」のセクションに再トライの回数を入力します。



3. 「エラー/成功処理の実行」セクションの下の  ボタンをクリックしてエラーの場合再トライされる実際の実行ステップを追加します。下で説明されているサンプルでは FTP サーバーファイルをアップロードするジョブがエラーの場合3回再トライされています。これは現在のジョブの外部で定義されたジョブで、関数のように呼び出されます。呼び出されたジョブの実装の詳細はここでは関連性がありません。他の実行ステップは同様に再トライされることができます。



4. 必須の「エラー時」セクションの  ボタンをクリックしてエラーの場合呼び出される実際の実行ステップを追加します。このサンプルでは、.log ファイルへ書き込むジョブはエラーの場合呼び出されます。関数のように呼び出される外部ジョブで、実装の詳細は表示されません。エラーログの基本情報は [ジョブエラー処理を追加する](#) を参照してください。



5. オプションで「成功時」「再トライ時」「常に」を追加します。このような追加ハンドラー型を構成するには新規のエラー/成功ハンドラーをクリックして、リストからハンドラー型を選択し  をクリックして必要なステップを対応するハンドラーの下に追加します。

## 5.8.2 実行する際の注意点

保護されたブロックは複数のステップを処理する場合があります。複数のステップを処理するには、保護されたブロック内の他のステップの後に追加します。構造に関しては、保護されているブロック内のステップは、標準の処理されていないステップと同様です（例えば、関数を実行、式の埋め込み、ループを作成することなどができます）。一部の情况、保護されたブロック内のステップは以下で説明される通り特別な処理を必要とする場合があります。

例外ハンドラーは複数の例外ステップを含むことができる点を最初に考慮してください。例えば、1つのステップはファイルを生成し、他のファイルは生成されたファイルを変換し、3番目のファイルは変換されたファイルを電子メールとして送信します。これは有効な構成です。また、複数の実行ステップを持つハンドラーは複雑性を追加し、エラーハンドラー内でエラーが発生する可能性があるため、慎重に扱われる必要があります。

同じハンドラー内に複数のステップが存在する場合、全てが実行されるまで、シーケンス順に、または、ステップが失敗するまで実行されます。ステップの失敗後はステップは実行されません。しかし、ハンドラーに失敗すると、存在する場合外側のハンドラーにより出力が処理されます。

この点に対応するために、ハンドラー内のステップの数を制限し、エラーの可能性を制限する必要があるかもしれません。ハンドラー内部のステップが少ないほうが、ハンドラーの実行が完了する可能性は高いです。後続するステップが前のステップに依存する場合、例えば、このステップのみのために新規のエラーハンドラーを追加し、成功時にのみ依存するステップの実行を継続することができます。

更に留意する点は例外ハンドラーから保護されたブロック内の結果を参照することができない点です。理由は、ステップが失敗すると、保護済みのブロックは、定義されていないものとなり、定義されていない結果を処理することは不可能だからです。

このため、例外ハンドラーを持つ保護されたブロック内に例外ステップが置かれると、例外ハンドラー内の実行ステップの結果にアクセスすることは不可能になります。ハンドラーが「成功時」、「エラー時」または「常に」であることは関係ありません。

保護されたブロック内のステップのためこの制限を適用することができます。この点をよりよく理解するために、以下のサンプルジョブをよく考慮してください！

### 実行ステップ

- +
  - ▲
 関数の実行 /system/compute  
 パラメーター: | 条件式: 'I am step 1'  
 = ステップの結果を割り当て result1 × 以下としてT0
- +
  - ▲
 エラー/成功処理にて実行
  - +
    - ▲
 関数の実行 /system/compute  
 パラメーター: | 条件式: 'I am step 2'  
 = ステップの結果を割り当て result2 × 以下としてT0
  - +
    - ▲
 関数の実行 /system/compute  
 パラメーター: | 条件式: 'I am step 3'  
 = ステップの結果を割り当て result3 × 以下としてT0
  - +
    - ▼
 エラー時 do
  - +
    - ▲
 関数の実行 /system/compute  
 パラメーター: | 条件式: 'I am error handler step'  
 = ステップの結果を割り当て result\_handler 以下としてT0

上記で説明されているジョブでは、各ステップの結果が存在します。例えば、最初のステップには **result1** が、2番目のステップには **result2** が存在します。上記のジョブでは、各ステップの結果が存在します。これらの結果の値にアクセスする必要がある場合は、以下の点に注意してください！

- ステップ 1 は保護されたブロックの外部であるため、後続する他の捨てペのステップより結果にアクセスすることができます。具体的には、**result1** にステップ 2 またはステップ 3、およびエラーハンドラーからアクセスすることができます。

- ステップ2 と3 はハンドラーからではなく同じ保護されたブロックの後続するステップからのみ結果にアクセスすることのできる保護されたブロックの内部に存在します。すなわち、**result2** はステップ3 からアクセスすることが可能ですが、エラーハンドラーからアクセスすることはできません。後続するステップが存在しないため、**result3** にアクセスすることはできません。
- 最後に、ハンドラー内に存在し、ハンドラーの唯一および最後のステップであるため、エラーハンドラーの結果である **result\_handler** は他のステップからアクセスすることはできません。後続するステップが存在する場合、**result\_handler** 結果を消費することができます。

上記に留意して、保護されたデザインは結果の可視性を構成しています。実際のソリューションは場合により異なります。例えば、ファイル名を作成するために「エラー時」ブロック内のステップの結果が重要な場合、自身の保護されたブロック（ネストされた保護されたブロック）の内部に含むことができ、[failed-step](#) 関数を呼び出して、エラーを含む出力を取得し、ファイルを作成することができます。エラーを引き起こしたステップを識別することはできませんが、この関数は、エラーが発生した場合エラー情報を含む **result** 型を返します。**result** はシェルコマンドを起動する抽象的な結果を表し、MapForce 変換、または StyleVision スタイルシートは期待する出力を作成する場合だけでなく、エラーが発生した場合でも処理されることができます。これは、サンプルにより理解することができ、[エラー処理をジョブに追加する](#)で詳細に説明されています。

## 5.9 延期されたステップ

エラーが発生しないことを前提として全ての処理ステップが完了した後のみにFlowForce ジョブの結果を返します。実行されているステップが存在する限り、ジョブの結果を返す前に完了するまで待つ必要があります。ジョブがWeb サービスとして公開されることは HTTP トランザクションがジョブの実行中に開かれたままであることを意味します。これは処理されたデータの量により異なり、数分、または数時間かかる場合があります。

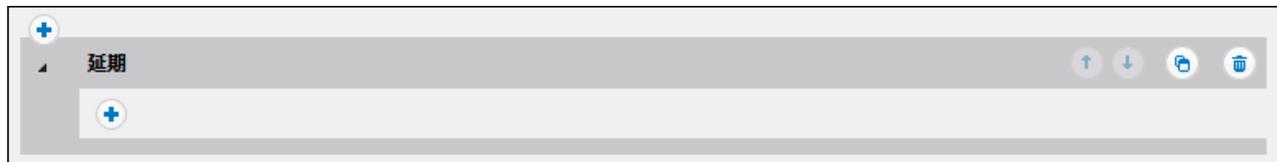
このようなケースを正常に処理するには、**延期されたステップ**と呼ばれるFlowForce の特別な型を使用することができます。処理中ジョブの結果を返した後のみに発生するため、延期されたステップはジョブの結果を早く返すことを回避しません。(このため「延期された」ステップと呼ばれます)。延期されたステップがジョブに含まれる場合、FlowForce は最初に延期されたステップではなく全てのステップを処理し、延期されたステップの処理を行います。

延期されたステップを持つジョブの結果を早く返すかもしれませんが、ジョブは延期されたステップの全ての実行が完了するまで実行中とみなされます。全ての延期されたステップの処理に成功した場合のみジョブが成功したものと考えられます。

延期されたステップをジョブ内でステップが許可されている個所で作成することができます。これを行うには、新規のジョブを作成するか既存のジョブを開き、「実行ステップ」の下の「新規の延期ステップ」ボタンをクリックします。



これは新規の「延期された」シーケンスをジョブ構成ページに追加します。



「延期された」シーケンス内の  をクリックして延期される実際のステップを作成します。以下のサンプルのように延期されたステップが相互に依存する場合、複数の延期されたステップを作成することができます。延期されたシーケンスをネストすることができますが、その場合、実行の順序は重要になります。下の「高度なシナリオ」を参照してください。

### 脚注

- 延期されたシーケンスは他のジョブと同様に現在のジョブ内にあり、ジョブと同じ構造を持つことができます。例えば、ループ(または部分的なループ) または(保護済みの) エラーの処理ブロックを含むことができます。
- ジョブは複数の関連しない延期されたシーケンスを含む場合があります。これはエラーの処理のために重要な場合があります。エラーが延期されたシーケンス内で発生する場合、他の影響を受けません。詳細に関しては以下の「エラーの処理」セクションを参照してください。
- 全ての延期されたシーケンスはすべての延期されていないステップの後に上から下に実行されます。特別なケースのためこの高度なシナリオを参照してください。
- メインのジョブの結果を返した後実行される呼び出されたサブジョブ内でステップを延期することは可能です。

### 例: 延期されたステップ

次のサンプルは延期されたステップの使用例について説明しています。Web サービスとして公開されており、ブラウザを含むクライアントをいつでも呼び出せるため、このジョブは単なる例です。このようなジョブの詳細は [Web サービスとしてジョブを公開する](#) を参照してください。

### 実行ステップ

**延期**

**関数の実行** /system/shell/commandline

パラメーター: コマンド: dir"C:\Program Files" /S

エラー時には中断: +

作業用ディレクトリ: +

= ステップの結果を割り当て output × 以下として結果

**関数の実行** /system/filesystem/copy

パラメーター: ソース: {as-file(stdout(output))}

ターゲット: C:\FlowForce\Postponed\output.txt

上書き: +

エラー時には中断: +

作業用ディレクトリ: +

= ステップの結果を割り当て 名前 以下としてブール値

**関数の実行** /system/compute

パラメーター: 条件式: 'task has been submitted successfully'

= ステップの結果を割り当て 名前 以下としてT0

ステップ A は大きなシステムディレクトリ内のすべてのディレクトリとファイルを再帰的にリストする時間がかかるシェルコマンドを実行します。この理由からステップ A は「延期された」シーケンスの内側に置かれています。ステップ B は A により生成された標準の出力をファイルに書き込みます (stdout)。ステップ B は A により生成される出力により異なり、延期されたシーケンスの一部となる必要があります。

ステップ C はサービスのエラーにタスクの送信が成功したことを通知します。このステップがジョブが実行されると最初に実行されますが、意図的に最後のものとして追加されます。

具体的には Web サービスが呼び出されると、上のステップが以下のシーケンスを実行します: C\*, A、B。理由は A と B は延期されたステップで C が最初に実行されるからです。アスタリスクマークがジョブが結果に戻る場所を示しています。

この構成の利点はステップ C の実行後ジョブがすぐに結果を返し、HTTP トランザクションが終了できるため、他のリクエストのためにサーバーリソースを空けることができるからです。ジョブの結果を返した後、FlowForce は延期されたステップ A と B を通常のシーケンスで実行します。

確認して、ブラウザ内でジョブを呼び出すと「ジョブメッセージタスクの送信に成功しました」が遅延なく表示される一方、output.txt ファイルが作成されるまでジョブは実行を継続します (リストされたディレクトリのサイズによりこれは数秒から数分かかります)。A または B 失敗しない場合、出力ファイルはパス C:\FlowForce\Postponed\output.txt で作成されます。

**メモ** このサンプルでは、具体的には文字列「タスクの送信に成功しました」 Web サービスをテストする際にブラウザに送信される結果を生成するためにステップ C がジョブで最後に実行されます。C を上に移動する場合でも、最初に実行され、延期されたステップ B がジョブの最後に実行されるステップのままです。しかしながら、これはジョブの結果を変更し、ブラウザは「[]」に類似した空の出力を表示する可能性があります。FlowForce ジョブの結果は常に最後に実行されたステップの結果であるからです。延期されたステップは値を返す必要はありませんが、代わりにからのシーケンスを生成します。

## 高度なシナリオ

ジョブは複数の延期されたステップを含みますが、前後に配置されている。また延期されたステップの複数のシーケンスである必要はありません。例えば、ジョブは以下の順序で以下のステップにより構成されているとします

```
A
postpone B
C
D を延期
```

上のステップは以下の順序で実行されます: A、C\*、B、D。確認できるように、延期されていないステップが最初に実行され、その後延期されたステップが実行されます。アスタリスクマークがジョブの結果に戻る場所を示しています。

条件(**choose** ステップ)内で延期されたステップを追加することもできます。この場合、延期されたステップに対応する“when”または“otherwise” ブランチが実行されると実行されます。

```
when expression=true
{
  postpone A
  B
  C
}
otherwise
{
  postpone D
  E
  F
}
```

上記の曖昧なジョブの場合、式が **true** を評価すると、ステップは以下の順序で実行されます: B、C\*、A。それ以外の場合実行順序は以下の通りになります: E、F\*、D。アスタリスクがジョブの結果を返すことを示しています。

**for each** ステップ内では、ループの一部であったとおリループと同じ順序で延期されたステップは延期されていないステップとして処理されます。

```
for each item in list
{
  A
  postpone B
}
```

例えば、ループが 3 回実行されると仮定すると上記のステップは以下の順序で実行されます: A1、A2、A3\*、B1、B2、B3、箇所:

- 数値はループの番号を示しています
- アスタリスクがジョブの結果を返すことを示しています。

他の延期されたステップ内で延期されたステップをネストすることもできます。この場合、ルールは以下の通りです:

- 同じ深さの外側のステップが最初に処理されます
- ネストされた延期されたステップは親シーケンスが完了した後のみに実行されます。

例えば、ジョブは次のネストされているステップにより構成されていると仮定します

```
A
postpone
[
  B
  C を延期
  D
  E を延期
  F
]
G
延期
[
  H
  postpone J
  K
  postpone L
  M
]
N
```

上のルールに従うとステップは以下の順序で実行されます: A、G、N\*、B、D、F、C、E、H、K、M、J、L。アスタリスクマークがジョブの結果に戻る場所を示しています。

上のよう高度な構成を作成しテストする必要がある場合、FlowForce [ログ](#)からステップの実行ステップをトラックすることが可能です。

## エラーの処理

以前に述べられているとおり、延期されたステップは延期されたシーケンスに属しています。ジョブ内の複数の箇所で複数の延期されたシーケンスが含まれている場合があります。エラーが発生すると、一般的なルールは以下の通りです:

- 延期されたステップがエラーに遭遇すると、同じ延期されたシーケンス内の次のステップと共にキャンセルされ、エラーがログされます。これは現在のシーケンスのみに影響を与えます。同じジョブ内に他の延期されたシーケンスが存在する場合、実行は継続されます。
- [保護されたブロック](#)内の延期されたステップがエラーに遭遇すると、通常これはそのブロックの一部であるすべての延期されたステップをキャンセルします。保護されたブロックはエラーが発生した場合複数回再試行するように構成されていることを考慮すると、これはロールバックの役割を果たす必要があります。[エラー時に再試行する](#)を参照してください。

上記の最初のルールの作動を説明するには以下のサンプルを確認してください!

### 実行ステップ

- +
延期

+
関数の実行 /system/compute

パラメーター: 条件式: `as-file(stream-from-string('The quick brown fox'))`

= ステップの結果を割り当て `output` 以下としてT0
- +
延期

+
関数の実行 /system/filesystem/copy

`{output}` for ソースと `C:\FlowForce\Postponed\demo.txt` for ターゲットと。
- +
延期

+
関数の実行 /system/compute

'All postponed steps have finished' for 条件式と。
- +
関数の実行 /system/compute

'Job has been submitted' for 条件式と。

上記のジョブでは唯一の延期されていないステップのためステップD が最初に実行されます。

ステップD 以外にも2つの延期されたシーケンスが存在します: 最初のシーケンスはA とB が含まれており、2番目のシーケンスはステップC が含まれています。ステップA が失敗したと仮定します。この場合、FlowForce Server は延期されたシーケンス全体をキャンセルし、ステップB もキャンセルします。ステップC を含むシーケンスは影響を受けず、実行を継続します。

## 5.10 ジョブの戻り値の型を宣言する

ジョブの結果を他のジョブで使用する場合、ジョブの結果をキャッシュする場合、ジョブの戻り型を宣言する必要があるかもしれません。戻り値の型を宣言する場合は、以下の点を考慮してください！

- 戻り値の型は、結果を実際に返すジョブのためには意味があります。
- ジョブの結果をキャッシュする場合、戻り値の型を宣言する必要があります。
- ジョブの戻り値の型は、ジョブ内の最後のステップの同じデータ型である必要があります。ジョブの最後のステップが、ジョブのために宣言されたものと異なるデータの型を返すと、FlowForce Server は「型と結果は一致しません」に類似したエラーを返します。このような型の一致に関するエラーが注じると、式関数を使用して、ジョブ内の最後のステップのデータ型をジョブの戻り値の型として宣言されたデータ型に変更します（以下を参照：[データ型をステップで処理する](#)）。

ジョブの戻り値の型を宣言します：

1. 新規ジョブを作成、または既存のファイルを編集のために開きます。
2. 実行の結果 セクション内の戻り値の型を選択します。

### 実行結果

戻り値の型を以下と宣言する: ストリーム ▼

使用することのできる戻り値の型は、以下のとおりです。

|       |  |
|-------|--|
| 無視/破棄 | これはデフォルトの戻り値の型です。これは、ジョブの結果が存在すると想定して、FlowForce Server にジョブの結果を無視、または、破棄するように命令します。ジョブが結果を返さない場合、または、返された結果を処理する必要がない場合このオプションを選択してください。 |
| 文字列   | ジョブにより返されるデータが、テキスト値（文字列）を示すことを指定します。  |
| ストリーム | ジョブにより返されるデータが、ストリームを示すことを指定します。   |
| 数値    | ジョブにより返されるデータが、数値を示すことを指定します。  |
| ブール値  | ジョブにより返されるデータが、ブール値であることを指定します。  |

## 5.11 データ型をステップで処理する

結果をステップから他のステップ、または、ジョブコンソールと、ステップの結果のデータ型を変更する必要がある可能性があります。例えば、ディレクトリの内容をリストするステップを作成すると、戻り値の型が結果になります。この結果を特定の方法で処理するには（例えば、電子メールに送信する、またはファイルに書き込む場合）結果をストリームまたは文字列に変換する必要があります。

ステップの戻り値の型（ステップ 1 と呼ぶこととします）を以下のように変更することができます：

1. ステップ 1 の「このステップの結果を割り当てる」フィールドに値を入力します。ステップの結果にアクセスするために、この値を参照する必要があります。
2. ステップ 1 の後に新規の実行ステップを追加します（ステップ 2 と呼びます）。
3.  `system/compute` ビルトイン関数を関数 ステップ 2 の実行関数として設定します。
4. ステップ 2 の式フィールド内では、ステップ 1 により返された値を必要とされるデータ型に変換する式を入力します。例えば、ステップ 1 に返された値が、出力を返すコマンドラインの結果で、これをストリームに変換する場合は、以下を使用してください！

```
stdout(result)
```

結果は、ステップ 1 により返された値に付けられる名前です。

使用される式は、達成を目的とする結果により異なります。FlowForce Server は、演算子と組み合わせ、ステップ内で使用する有益な式を作成することのできる、多様な式関数を提供します。使用することのできるオプションを理解するには、以下を参照してください！ [ステップ結果関数](#)。

### 例

ディレクトリの内容をリストする実行ステップを以下のように作成したと想定します。以下の通りです：

### 実行ステップ

+

関数の実行

パラメーター:

|            |                                  |
|------------|----------------------------------|
| コマンド:      | <input type="text" value="dir"/> |
| エラー時には中断:  | <input type="button" value="+"/> |
| 作業用ディレクトリ: | <input type="text" value="c:\"/> |

= ステップの結果を割り当て  として result

この段階では、ステップの出力は、結果です。（例えば、ディレクトリ内容をファイルに書き込む、または、コンテンツを電子メールとして送信するなど）この結果を他のステップで使用する場合は、結果ではなく、型「ストリーム」の値が必要になります。

### 「ストリーム」データ型にステップをリストするディレクトリの出力を変換する

1. ステップ 1 の結果を as **Step1Output** として宣言します。これを行うには、「このステップの結果を割り当てる」フィールド内に **Step1Output** を入力します。

2.  `/system/compute` 関数を呼び出す新規の実行ステップを追加します。
3. パラメータに、次の式を入力します: `stdout(Step1Output)`、`step1output` がステップ1の結果である箇所。

### 実行ステップ

+ 関数の実行 `/system/shell/commandline` ▼ 

パラメーター:

|            |                                     |
|------------|-------------------------------------|
| コマンド:      | <code>dir</code>                    |
| エラー時には中断:  | <span style="color: blue;">+</span> |
| 作業用ディレクトリ: | <code>c:\</code>                    |

= ステップの結果を割り当て `Step1Output` として result

+ 関数の実行 `/system/compute` ▼ 

パラメーター:

|      |                                  |
|------|----------------------------------|
| 条件式: | <code>stdout(Step1Output)</code> |
|------|----------------------------------|

= ステップの結果を割り当て `名前` として TO

新規の実行ステップ
新規の選択ステップ
新規の For-each ステップ
新規エラー/成功処理ステップ

この段階では、ジョブ出力データの型は、意図したとおり、ストリームです。ステップの2番目の結果を電子メールに送信する新規の実行ステップを作成することができます。

#### 電子メールにディレクトリのリストを送信する

1. `stdout(Step1Output)` から `content(stdout(Step1Output))` にステップ2の式を変更します。これを行うことにより、ストリームの値を文字列の値に変換することができます。
2. ステップ2の結果を **Step2Output** と宣言します。これを行うには、「このステップの結果を割り当てる」フィールド内に **Step2Output** を入力します。
3. 新規の実行ステップ(ステップ3)を次の設定と共に追加します( [電子メールサーバー設定](#) は、ビルトイン  `/system/mail/send` 関数を使用する前に構成されなければならない点に注意してください)。

|      |   |
|------|---|
| 実行関数 | 次を参照します:  <code>/system/mail/send</code> 関数。 |
| 差出人  | 電子メールの差出人を入力します。  |
| 宛先   | 電子メールの宛先を入力します。   |
| 件名   | 電子メールの件名を入力します。例えば、「Directory Listing」。   |
| Body | <span style="border: 1px solid #ccc; padding: 2px 5px;">Set to ▶</span> をクリックし、 <b>Step2Output</b> を選択します。                      |

この段階では、実行ステップは以下のようになります:

### 実行ステップ

- 開放の実行 /system/shell/commandline**
  - パラメーター: コマンド: dir 文字列 (必須) [セットする値](#)
  - エラー時には中断: +
  - 作業用ディレクトリ: c:\ 文字列 (オプション) [セットする値](#)

= ステップの結果を割り当て Step1Output としてresult
- 開放の実行 /system/compute**
  - パラメーター: 条件式: stdout(Step1Output) 条件式T0 (必須) [セットする値](#)

= ステップの結果を割り当て Steo2Output としてT0
- 開放の実行 /system/mail/send**
  - パラメーター:
    - 差出人: someone@example.org 文字列 (必須) [セットする値](#)
    - 宛先: someone@example.org 文字列 (必須) [セットする値](#)
    - 件名: Directory listing 文字列 (必須) [セットする値](#)
    - 本文: (Step2OOutput) 文字列 (オプション) [セットする値](#)
  - 添付: +
  - エラー時には中断: +

= ステップの結果を割り当て 名前 として boolean

4. ジョブを1度または定期的に実行するタイマートリガーを作成します (例えば、毎 60 分)。

### トリガー

実行: 毎日 毎 1 日

繰り返し: 毎 60 分  全日、または  開始時間 08:00:00 終了時間 20:00:00

開始: 2016-07-20 08:00:00

有効期限: +

タイムゾーン: Europe/Berlin

有効化されました

[新しいタイマー](#)
[新規ファイル システム トリガー](#)
[新規 HTTP トリガー](#)

5. 「保存」をクリックします。

## 5.12 Windows ネットワークパスの参照

ジョブを作成する際、通常 FlowForce Server が作動するマシン上で、または、ネットワーク上のファイルパスを参照します。マップされたネットワークドライブなどの Windows ネットワークパスを参照する場合、Universal Naming Convention (UNC) 構文を使用します。これはドライブ文字はシステムに対してグローバルではないため、また、各ログインセッションは自身のドライブ文字に割り当てられているため、これはドライブ文字がシステムに対してグローバルではなく、ログインセッションが自身のドライブ文字に割り当てられているため必須です。

UNC 構文は以下のフォームをとります:

```
\\server\sharedfolder\filepath
```

該当する箇所:

- `server` が (DNS により定義される) ネットワーク内のサーバー名を参照する箇所。
- `sharedfolder` が管理者により定義されるレベルを参照する箇所 (例えば、`admin$` が通常オペレーティングインストールのルートディレクトリである箇所)。
- `filepath` がシェアの下のサブディレクトリを参照する箇所。

## 5.13 トリガーの管理

ジョブを作成する際、トリガーとしてジョブをトリガーする条件(または基準)を指定する必要があります。FlowForce Server は、定義されたトリガーを監視し、トリガーの条件が満たされるとジョブを実行します。

同じジョブのために、複数のトリガーを作成することができます。また、定義されたトリガーを有効化、または無効化することもできます。有効化されたトリガーが実行されると、FlowForce Server は、ジョブの全てのステップを実行します。

パラメータを持つジョブ内出トリガーを使用する場合、全てのパラメータはデフォルトの値を必要とします。それ以外の場合、ジョブは実行されません。

以下の種類のトリガーを作成することができます:

- タイマトリガー
- ファイルシステムトリガー
- HTTP トリガー

タイマトリガーを使用することにより、ジョブが特定の時刻に(毎日、毎週、または毎週の特定の日に、または他の日に実行されるなどの)柔軟性に富んだ再帰オプションを使用して、特定の時間の間隔で開始される計画を立てることができます。

ファイルシステムトリガーを使用することにより、新しく追加されたファイルまたは変更されたファイルなど、ファイルまたはディレクトリの変更を監視することができます(削除されたファイルは監視できないことに注意してください)。(例えば、60 秒ごとに実行など)変更の収集の間隔を柔軟に構成することも可能です。また、任意でトリガーの開始と終了の日付を設定することもできます。ワイルドカードを使用して、ディレクトリ特定のファイルをフィルタすることも可能です。

HTTP トリガーを使用することにより、URI (Uniform Resource Identifier) の変更を監視することができます。特に、**Last-Modified** と **Content-MD5** HTTP ヘッダーフィールドの変更を収集することができます。(例えば、60 秒ごとに実行など)変更の収集の間隔を柔軟に構成することも可能です。また、任意でトリガーの開始と終了の日付を設定することもできます。

トリガーを追加するには、トリガーの型に対応したボタンをクリックします。



下にリストされているトリガーの横にあるボタンを使用してトリガーを管理してください。

|  |                |
|--|----------------|
|  | トリガーの削除        |
|  | トリガーの複製        |
|  | 前の削除アクションを元に戻す |

トリガーフィールドの一部には と ボタンが横に表示されています(例えば、タイマトリガーの開始日)。これらのボタンを使用してトリガーフィールドの値を設定またはクリアします。値は、ページに表示されると設定済みとみなされます。例えば、下のイメージでは「繰り返し」の値は設定されていませんが、「開始」の値は、「2016-03-29 00:00:00」に設定されています。

トリガーの値が効果を発揮するのは、ジョブの保存が必要なおことご注意ください。

以下も参照してください

- [タイマトリガー](#)
- [ファイルシステムトリガー](#)
- [HTTP トリガー](#)

### 5.13.1 タイマートリガー

タイマトリガーを使用することにより、ジョブが特定の時刻に（毎日、毎週、またはお週の特定の日に、または他の日に実行されるなどの）柔軟性に富んだ再帰オプションを使用して、特定の時間の間隔に開始される計画を立てることができます。

下のスクリーンショットはタイマトリガーのサンプルを表示しています。

タイマトリガーの例

タイマトリガーは以下の構成がされます。

|           |   |
|-----------|---|
| <p>実行</p> | <p>トリガーが度、またはN 日ごと周期的に実行されるか定義します。使用することのできるオプションは、以下のとおりです:</p> <ul style="list-style-type: none"> <li>• 1度</li> <li>• 毎日</li> <li>• 週の日で</li> <li>• 月の日で</li> </ul> |
|-----------|---|

|        |   |
|--------|---|
|        | <ul style="list-style-type: none"> <li>月の週にある日</li> </ul>   |
| 繰り返し   | <p>トリガーの繰り返しオプションを定義します。実行ドロップダウンリスト内出指定された日数でイベントが繰り返し実行されます（前のフィールドを参照）。</p> <p>毎フィールドは繰り返しの間隔を分数で定義します。</p> <p>「から」と「まで」のフィールドは繰り返されるイベントの時間の範囲を定義します。</p> |
| 開始     | <p>トリガーの開始日時を定義します。</p> <p>「1度」が実行ドロップダウンリストから選択されている場合、開始の日時は入力必須の項目です。</p> <p>日付フィールドをクリックすると、開始日を選択することのできるポップアップカレンダーが開かれます。</p>                          |
| 有効期限   | <p>トリガーの日時の有効期限を定義します。</p>  |
| タイムゾーン | <p>開始および有効期限の日時に適用することのできるタイムゾーンを定義します。デフォルトのタイムゾーンがサーバー管理設定内で定義されます（以下を参照: <a href="#">デフォルトのタイムゾーンの設定</a>）。</p>   |
| 有効化    | <p>有効化されたチェックボックスにより、トリガーを有効化、または無効化することができます。このオプションは、新しいジョブを作成し、テストする際に便利です。</p>  |

### 5.13.2 ファイルシステムトリガー

ファイルシステムトリガーを使用することにより、新しく追加されたファイルまたは変更されたファイルなど、ファイルまたはディレクトリの変更を監視することができます（削除されたファイルは監視できないことに注意してください）。（例えば、60 秒ごと実行など）変更の収集の間隔を柔軟に構成することも可能です。また、任意でトリガーの開始と終了の日付を設定することもできます。ワイルドカードを使用して、ディレクトリ特定のファイルをフィルタすることも可能です。

下のスクリーンショットはファイルシステムトリガーの例を表しています。

**トリガー**

チェック  変更の日付  ファイルまたはディレクトリ:  ポーリング間隔:  秒、解決するまで  秒待つ。

開始:

有効期限:

タイムゾーン:

が有効化されました

サンプルファイルシステムトリガー

ファイルシステムトリガーには以下の構造があります。

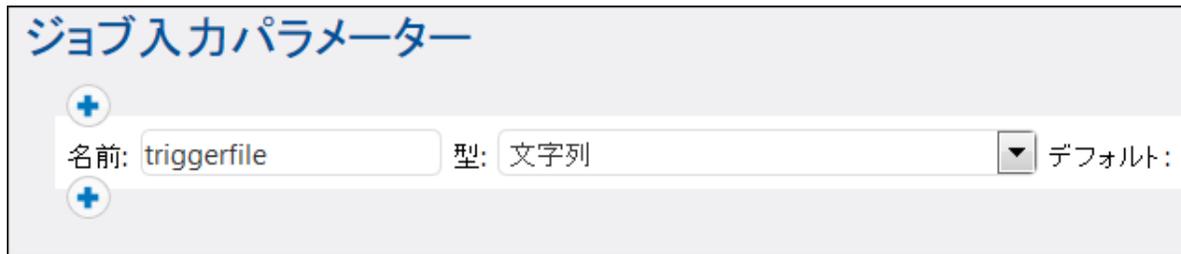
|      |  |
|------|--|
| チェック | <p>トリガーがURI をどのようにポーリングするかを指定します。有効なオプション:</p> <p><b>新規作成</b>                      新規のファイルまたはディレクトリが指定されたディレクトリに追加されると、トリガーは実行されます。サーバーのロード量に関しては、このオプションは最も少な</p> |
|------|--|

|         |   |
|---------|---|
|         | <p>いサーバーのノースを必要とします。</p> <p>新規のトリガーが追加されジョブが保存されると、そのディレクトリ内の既存のファイルは新規で作成されたものと考慮され、それぞれのためにジョブが実行されます。</p> <p>ファイルが削除され、後に追加されるとジョブはもう一度実行されます（これは「ポーリング間隔」秒が既に削除から経過した場合のみ実行されます）。ファイルの名前が変更された場合でもこのトリガーは実行されます。</p> <p>ポーリングされたディレクトリが変更されない場合このトリガーは実行されません。このような振る舞いが必要な場合、次のオプション「変更された日付」を参照してください。</p> <p><b>変更された日付</b></p> <p>トリガーは、指定されたファイルの最後の変更タイムスタンプをチェックします。日付が変更されている場合、または、新しいファイルが追加されている場合、またはファイルの名前が変更された場合トリガーは実行されます。</p> <p><b>コンテンツ</b></p> <p>このオプションは特定のファイルのためのハッシュコードを計算し保管します。ポーリング間隔が経過すると、ハッシュコードは、再計算され、保管されている値と比較されます。差異が存在する場合、トリガーが実行されます。これはサーバーに大量のロードを課すこととなる点に注意してください。</p> <p>日付が変更されている場合、または、新しいファイルが追加されている場合、またはファイルの名前が変更された場合トリガーは実行されます。</p> |
| ポーリング間隔 | ディレクトリがポーリングされる間隔を秒数で指定します。デフォルトの値は60秒です。最小の値は1秒です。   |
| N 秒待つ   | 次のジョブを開始するまでの待ち時間を秒数で定義します。   |
| 開始      | トリガーの開始日時を定義します。これは任意のフィールドです。<br>日付フィールドをクリックすると、開始日を選択することができるポップアップカレンダーが開かれます。  |
| 有効期限    | トリガーの日時の有効期限を定義します。   |
| タイムゾーン  | 開始および有効期限の日時に適用することのできるタイムゾーンを定義します。デフォルトのタイムゾーンがサーバー管理設定内で定義されます（以下を参照： <a href="#">デフォルトのタイムゾーンの設定</a> ）。   |
| 有効化     | 有効化されたチェックボックスにより、トリガーを有効化、または、無効化することができます。このオプションは、新しいジョブを作成し、テストする際に便利です。  |

### Triggerfile パラメーター

ファイルシステムまたはHTTPトリガーを作成する場合、FlowForce Server は自動的にtriggerfile 入力パラメーターをジョブに追加します。ジョブが実行されると、FlowForce Server はパラメーターを次に設定します。

- a) ジョブをトリガーしたファイル（ファイルシステムトリガーの場合）。
- b) 収集されたURI（HTTPトリガーの場合）のダウンロードコンテンツを含む一時ファイルの名前。



**triggerfile** / パラメーターの値を次のステップ内の入力値として、バースするすることができます。このようにして、トリガーするファイルを必要に応じて使用、または、処理することができます。

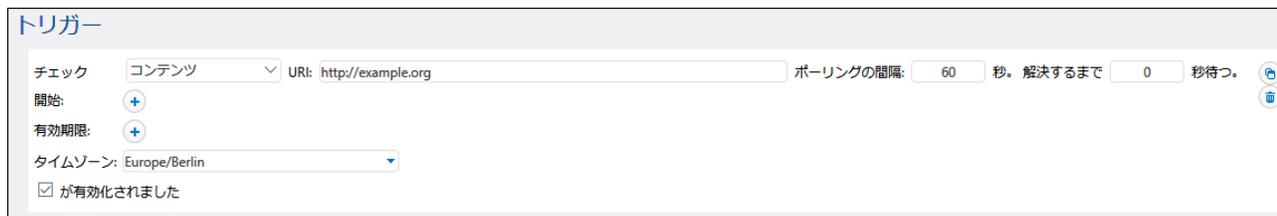
デフォルトでは**triggerfile** / パラメーターは、トリガーするファイルの絶対パスを含んでいます。パスの一部を抽出するには[ファイル変換関数](#)を使用します。

**triggerfile** / パラメーターを使用するジョブの例に関しては、[ディレクトリポーリングジョブ](#)を作成するを参照してください。

### 5.13.3 HTTP トリガー

HTTP トリガーを使用することにより、URI (Uniform Resource Identifier) の変更を監視することができます。特に、**Last-Modified** と **Content-MD5** HTTP ヘッダーフィールドの変更を収集することができます。(例えば、60 秒ごと実行など) 変更の収集の間隔を柔軟に構成することも可能です。また、任意でトリガーの開始と終了の日付を設定することもできます。

下のスクリーンショットはHTTP トリガーの例を表しています。



サンプルHTTP トリガー

HTTP トリガーには以下の構造があります。

|                |  |
|----------------|--|
| <p>チェック</p>    | <p>トリガーがURI をどのようにポーリングするかを指定します。有効なオプション:</p> <p>HTTP ヘッダー日付      最後に変更された HTTP ヘッダーをチェックするようにシステムに命令します。最後に変更 HTTP ヘッダーが不足している場合、コンテンツ-MD5 ヘッダーのみがチェックされます (次のオプションを参照してください)。</p> <p>コンテンツ                  コンテンツ-MD5 HTTP ヘッダーをチェックするようにシステムに命令する。これはメッセージ統合チェックとして使用されている 128 ビット「ダイジェスト」です。ポーリング間隔がバースされた後、MD5 ヘッダーが変更されるとファイルがトリガーされます。ヘッダーがサーバーにより提供されていない場合、コンテンツは取得され、ローケルでキャッシュ化されます。</p> |
| <p>ポーリング間隔</p> | <p>URI がポーリングされる間隔を秒で指定します。</p>  |

|             |   |
|-------------|---|
| 解決するまでN 秒待つ | 次のジョブを開始するまでの待ち時間を秒数で定義します。   |
| 開始          | トリガーの開始日時を定義します。これは任意のフィールドです。<br>日付フィールドをクリックすると、開始日を選択することのできるポップアップカレンダーが開かれます。                          |
| 有効期限        | トリガーの日時の有効期限を定義します。   |
| タイムゾーン      | 開始および有効期限の日時に適用することのできるタイムゾーンを定義します。デフォルトのタイムゾーンがサーバー管理設定内で定義されます（以下を参照： <a href="#">デフォルトのタイムゾーンの設定</a> ）。 |
| 有効化         | 有効化されたチェックボックスにより、トリガーを有効化、または、無効化することができます。このオプションは、新しいジョブを作成し、テストする際に便利です。                                |

### Triggerfile / パラメーター

ファイルシステムまたはHTTP トリガーを作成する場合、FlowForce Server は自動的にtriggerfile 入力パラメーターをジョブに追加します。ジョブが実行されると、FlowForce Server はパラメーターを次に設定します:

- ジョブをトリガーしたファイル(ファイルシステムトリガーの場合)。
- 収集されたURI (HTTP トリガーの場合) のダウンロードコンテンツを含む一時ファイルの名前。

### ジョブ入力パラメーター

+

名前: triggerfile

型: 文字列

デフォルト:

+

triggerfile / パラメーターの値を次のステップ内の入力値としてバインドすることができます。このようにして、トリガーするファイルを必要に応じて使用、または、処理することができます。

デフォルトではtriggerfile / パラメーターは、トリガーするファイルの絶対パスを含んでいます。パスの一部を抽出するには[ファイル 変式関数](#)を使用します。

triggerfile / パラメーターを使用するジョブの例に関しては、[ディレクトリポーリングジョブ](#)を作成するを参照してください。

## 5.14 ジョブを Web サービスとして公開する

FlowForce Server ジョブを Web サービスとして公開することができます。Web サービスとして公開されるジョブは、プログラミング的にアクセスできることを基本的に意味します。テストとデバッグのために、このようなジョブをブラウザから呼び出すこともできます。

Web サービスとしてジョブを公開する

1. ジョブを作成する、または既存のジョブを編集のために開きます。
2. 「HTTP を介してこのジョブを利用可能にする」チェックボックスを選択します。

3. URL テキストボックス内に Web サービスの名前を入力します。

ジョブを保存すると FlowForce Server は HTTP GET と POST リクエストを指定された URL で受け入れるリスナーサービスを開始します。サービスとして公開されるジョブは、FlowForce サーバが作動している間はアクティブです。

**メモ** [セトアップページ](#)から FlowForce Server サービスのホスト名 フィールドを設定すると Web サービスの呼び出し  ボタンが使用できるようになります。このボタンをクリックすると新規のブラウザのウィンドウ内で Web サービスが呼び出されます。FlowForce Server のためのホスト名を構成していない場合、をブラウザのアドレスバー内に URL を手動で入力して Web サービスを呼び出すことができます。

Web サービスが呼び出されると、FlowForce Server は指定されたジョブの実行ステップを実行し、次を返します:

- ジョブが結果ファイルを生成する場合、最後のステップの最初の結果ファイル。
- 結果ファイルが生成されない場合、最後のステップの標準出力 (コマンドライン出力と作業する場合などが当てはまります)。

結果に従ってコンテンツ型 ヘッダーが表示され、HTTP 200 ステータスと共に有効な結果が返されます。コンテンツ型 ヘッダーは実際の結果により異なります。MapForce マッピングは XML 出力が存在する場合、または、テキスト出力のための text/plain が存在する場合、text/xml で結果が表示されます。他の関数の標準の出力も text/plain で返されます。結果は修飾記号なしでレスポンスボディとして返されます。

実行エラーはジェネリックなエラーメッセージと共に HTTP 5xx ステータスとして報告されます。詳細に関しては、FlowForce Server [ログ](#) をチェックしてください (ジョブログの確認 を参照)。

このようなジョブの構成を説明するステップバイステップのサンプルに関しては [Web サービスとしてジョブを公開する](#) を参照してください。

全てのジョブステップが実行される前に FlowForce が結果を返すように構成することは可能です。サービスとして呼び出されたジョブが時間を要する場合場合この機能は特に役に立ちます。早い結果は処理のために FlowForce Server に受け入れられ、呼出元により確認として扱われます。詳細に関しては [延期されたステップ](#) を参照してください。

現在構成しているすべての FlowForce Server Web サービスを確認する

以下の内の一つを行ってください!

- FlowForce Web 管理インターフェイス内でホームに移動し、全てのアクティブなトリガーとサービスを表示するをクリックします。[アクティブなトリガーとサービス](#)も参照してください。
- 以下のURL に以下のブラウザからアクセスします: `http://[FlowForceServer][ServerPort]/service/*`

**メモ** URL 内では [FlowForceServer] と [ServerPort] は FlowForce Server がリスニングするネットワークアドレスとポートを参照します。デフォルトでは (同じマシンからアクセスすると想定し) FlowForce Server は <http://localhost:4646> で作動します。サーバー名とポートは管理ページで定義されているとおりです。 [ネットワーク設定の定義](#)を参照してください。

## Web サービスパラメータ

Web サービスとしてジョブを公開すると、全てのジョブのパラメータは自動的にサービスのパラメータになります。ジョブのパラメータにデフォルトが存在しない場合、サービスを呼び出す際にデフォルトを与える必要があります。リクエスト内で与えられていない場合、任意のパラメータはデフォルトの値を取ります。

サービスが呼び出されると、FlowForce Server は、リクエスト内で与えられたパラメータをジョブ内で定義されたパラメータに対して検証します。パラメータの検証に失敗すると、FlowForce Server は、5xx HTTP ステータスを返します。この場合デバッグとテストのために、FlowForce Server はHTML パラメータフォームを表示します。デバッグとテストのためにHTML フォームを使用して、パラメータを手動で入力することができます。型 *stream* のパラメータのために、ページ内でブラウザボタンが使用できるようになり、パラメータが必要とするようにファイルをアップロードするために使用することができます。

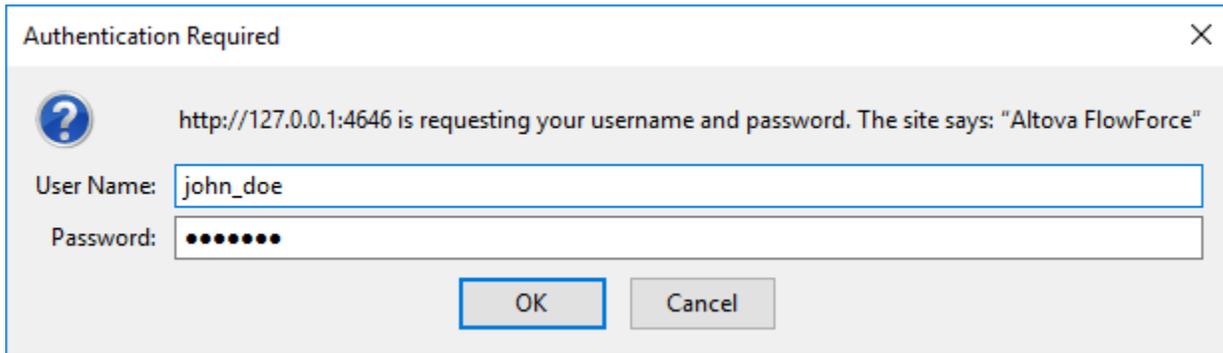
テスト HTML フォームを条件無しに表示するには、リクエスト内ボディ内の (値を持つ) パラメータ `showform` を与えます。

パラメータを持つ FlowForce Web サービスを呼び出すには、次のオプションの一つを使用することができます:

1. 文字列、または、数値などの単型のパラメータには、GET または POST リクエストのURL 内に提供することができます。例えば、[Web サービスとしてジョブを公開する](#)を参照してください。
2. POST の場合は、更に `multipart/form-data` または `application/x-www-form-urlencoded` としてパラメータを提供することができます。FlowForce 内の型「*stream*」がパラメータの場合、(FlowForce が期待するため) `multipart/form-data` としてクライアントが提供する必要があります。このようなパラメータのために、ブラウザは、対応するパラメータの横に「参照」ボタンを表示するHTML フォームをテストします。
3. クライアントはPOST リクエストのボディ内の任意のコンテンツを含むことができます (パラメータとしてではなく、HTTP リクエストのボディとしてポストされているJSON またはXML などのコンテンツを具体的に示します)。これを可能にするには、FlowForce ジョブは型 *stream* の *single* パラメータを含んでいる必要があります。追加の非 *stream* パラメータを必要とする場合、これらはPOST URL 内に提供される必要があります。しかしながら、FlowForce 内で一つの型 *stream* のパラメータが定義される必要があります。これらの条件が満たされると、*stream* パラメータのためにデータとしてリクエストボディが扱われます。他の構成は必要ありません。例えば、[JSON を FlowForce Web サービスへポストする方法](#)を参照してください。

## Web サービスの認証

デフォルトでは、FlowForce Server はHTTP ベーシック認証を使用して、Web サービスを呼び出すクライアントを認証します。ユーザーの資格方法はFlowForce Server ユーザーデータベースに対してチェックされます (FlowForce Server Web 管理インターフェイスにログオンするために使用されるユーザー名とパスワード)。



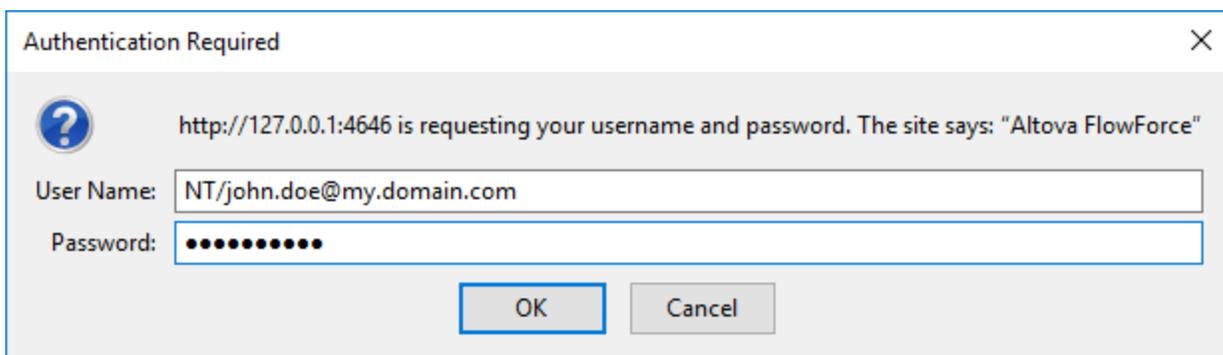
資格情報無しでサービスが使用できるようにするには、デフォルトの  **anonymous** ユーザーにサービスの使用 パーミッションを与えます ([パーミッションのしくみ](#)を参照してください)。

サービスが匿名の使用を許可する場合 HTTP 資格情報を提供することができます。資格情報は FlowForce Server のユーザーデータベースに対してチェックされ、サービス実行ファイルは、匿名ユーザーの代わりに認証済みのユーザーの属性とされます。

無効な資格情報を提供すると、401 の HTTP 状態をインターフェイスに戻します。サービスの使用はこのサービスの匿名のユーザーに与えられません。リクエストインターフェイスは、401 の HTTP 状態を返します。

有効な資格情報が与えられ、このサービス上でサービスの使用 パーミッションが認証されていないユーザーに与えられていない場合、リクエストインターフェイスは HTTP 4xx 失敗状態を返します。存在しないサービスアクセスしようとすると、HTTP 4xx 失敗状態が返されます。

HTTP ベンチング認証に加え、任意で、Windows ドメインの認証を構成することもできます。構成方法に関しては、[デフォルトサービスの変更設定の変更](#)を参照してください。Windows ドメイン認証が構成されると、有効なユーザー名とパスワードを提供することにより、ユーザーは FlowForce Server により公開される Web サービスにアクセスできるようになります。重要な点は、ユーザー名がプレフィックス NT/ を含む必要があり、ドメイン名を除外する必要があることです。例: NT/john.doe@my.domain.com。



### キューの設定

サービス URL はリガナーの特別な型で、このため接続が構築されると同じキューの制約の対象となります。[キューの設定の定義](#)を参照してください。

### HTTP リクエストボディの最大サイズを構成する方法

大体 100 MB 程度の HTTP リクエストボディの最大サイズを確立する FlowForce Server 内に存在するデフォルトのミット。コーラーが HTTP リクエストを Web サービスとして公開されている FlowForce ジョブにポストし、HTTP リクエストボディがミットを超すと FlowForce Server は以下のテキストを含むエラーを返す場合があります:

リクエストと共に送信されたエンティティが許可されている最大バイトを超える場合。

#### 大きなサイズのリクエストを受け入れる方法

1. [flowforce.ini](#) ファイルをテキストエディターで開きます。
2. オプション `max_request_body_size` を **[Listen]** または **[ListenSSL]** セクションに追加して、許可されるバイト数を最大に設定します。

例えば 500 MB の最大サイズを有効化するには [flowforce.ini](#) ファイルは以下のようになります:

```
[Listen]
active=1
host=0.0.0.0
port=4646
hostname=somehost.example.org
max_request_body_size=500000000
```

.ini ファイルに関する詳細は [構成ファイルファレンス](#) を参照してください。

#### FlowForce Server プールスレッドの再構成の方法

大量の並列 HTTP サービスリクエスト (例えば 1度に 20 または以上) を期待する場合サーバーをプールスレッドの若干大きい数量に再構成することが可能です。

1. [flowforce.ini](#) ファイルをテキストエディターで開きます。
2. オプション `thread_pool` を .ini ファイルの **[Listen]** または **[ListenSSL]** セクションに追加し 20 以上の値に設定します。
3. サービスを再起動します。

**メモ** (多くのプールスレッドを必要としない) FlowForce Web Server のためと (異なるポート上の) 他のリクエストのため個別の **[Listen]** セクションを持つことは一案です。それ以外の場合、FlowForce Web Server はプールスレッドのため他の HTTP リクエストと競合します。

## 5.15 ジョブの結果をキャッシュする

キャッシュによりサーバーのロードとジョブのレスポンスの時間を削減することができます。完了するまでの時間がわかり、サーバーのロード量が大きくなくことがある。Web サービスとして公開される。またはデータ集中型のジョブにとって、キャッシュは特に役に立ちます。

ジョブの結果のキャッシュとは、FlowForce Server が、内部のリポネリジョブの結果を用意し保管(つまり、キャッシュ)することを基本的に意味します。ジョブのラメータが存在する場合、システムは構成できる上限まで、各ラメータの組み合わせのためにキャッシュエントリを作成します。( [コンシューマ](#) とされる他のジョブからキャッシュを持つジョブが呼び出された場合、FlowForce Server は、(ジョブを再度実行する代わりに)コンシューマに対する結果を返し、レスポンスの時間を削減します。

サーバーへのデータのロードの量のバランスを保つには、一方でキャッシュが古くならないように注意し、もう一方で、使用する環境とジョブによりキャッシュを指定します。このバランスを達成するために、FlowForce Server に搭載されるキャッシュの設定に慣れ親しみ、必要に応じてキャッシュを使用してください。

キャッシュされたジョブの結果と作業する場合、以下の点に留意してください！

- 結果がキャッシュされるジョブによるデータ型を宣言する必要があります(以下を参照: [ジョブの戻り値の型を宣言する](#))。
- 結果がキャッシュされるジョブとコンシューマジョブは、同じ資格情報を使用する必要があります。資格情報が異なると、ジョブはキャッシュが定義されていないものとして実行されます。
- キャッシュされたジョブの構成を変更すると、既存のキャッシュデータは無効になります(フラッシュされます)。

### 結果のキャッシング:

結果をキャッシュする このジョブが他のジョブより呼び出されるとキャッシュは使用されます。

- キャッシュされた結果を作成または更新する時間のトリガーを追加する、またはコンシューマによって開始済み オプションを確認する。
- これを呼び出し、またキャッシュから利益を得るジョブを作成する。
- コンシューマによって開始済み オプションが選択されている場合、キャッシュの更新または削除タイマーを追加しキャッシュ エントリが古くなることを回避します。
- ジョブ入力パラメータがコンシューマによって開始済み オプションを設定している場合、"最大キャッシュ エントリ数" は入力パラメータの可能なバリエーションを期待できます。

コンシューマにより開始済み

最大キャッシュ エントリ数:

HTTP から使用可能な  新しいキャッシュ コンシューマ ジョブ/public/db2osm.cached を自動作成する

### キャッシュの設定

使用することのできるキャッシュの設定は、以下のとおりです。

|                            |  |
|----------------------------|--|
| <p>結果をキャッシュする</p>          | <p>ジョブの結果をキャッシュする場合このボックスを選択します。このボックスをチェックすることにより、現在のジョブの <a href="#">コンシューマ</a> に、ジョブを実行する代わりに、キャッシュされた結果を読み取るよう命令します。</p> <p>定義されたトリガーが実行されたため、またはジョブの Web サービスが呼び出されたため(コンシューマによりではなく)現在のジョブが直接実行されている場合、FlowForce Server は、キャッシュを更新します(または、ジョブ ラメータキャッシュ内で見つからない場合、与えられたラメータの組み合わせをベースとして新規のキャッシュエントリを作成します)。</p> |
| <p>コンシューマにより開始されるキャッシュ</p> | <p>このオプションが有効化されていると、現在のジョブを呼び出す全てのジョブ(つまり、<a href="#">コンシューマ</a> ジョブ)は、キャッシュが既存でない場合、キャッシュを作成します。それ以外の場合、トリガーと Web サービスの呼び出しのみがキャッシュを作成します。</p>   |
| <p>最大キャッシュエントリ数</p>        | <p>このオプションは、各ジョブのために作成されるジョブの結果のキャッシュを制限します。ジョブパラメータが存在する場合、(組み合わせは無制限と想定して)このオプションを全ての可能なラメータの組み合わせに設定されることが奨励されます。</p>   |

|                                 |  |
|---------------------------------|--|
| <p>キャッシュコンシューマージョブを自動的に作成する</p> | <p>キャッシュコンシューマージョブは、HTTP アドレスで指定される Web サービスです。コンシューマー Web サービスは、結果がキャッシュされるジョブのキャッシュを取得し管理する便利な方法としての役割を果たします。呼び出されると、コンシューマージョブは、最初にキャッシュされたメインジョブの結果を使用しようと試みます。キャッシュされた結果が存在せず、「コンシューマーにより開始」オプションが無効化されていると、コンシューマーは、メインジョブにより返される実際の結果を取得します。キャッシュされた結果が存在せず、「コンシューマーにより開始」オプションが有効化されていると、コンシューマーは、メインジョブにより返された実際の結果を取得し、キャッシュも作成します。</p>  |
| <p>キャッシュタイマーの更新</p>             | <p>キャッシュタイマーの更新は、システムがどの頻度で現在のジョブをキャッシュするか管理します。現在キャッシュされているすべてのパラメータの組み合わせが更新されます。</p> <div data-bbox="540 600 1414 1016"> <p>実行 <input type="text" value="更新"/> 毎日 <input type="text" value="毎日"/> 毎 <input type="text" value="1"/> 日</p> <p>繰り返す <input type="button" value="+"/></p> <p>時間: <input type="text" value="15:57:00"/> <input type="button" value="🗑️"/></p> <p>タイムゾーン: <input type="text" value="Europe/Berlin"/></p> <p><input checked="" type="checkbox"/> 有効化されました</p> <p><input type="button" value="新規キャッシュ更新タイマー"/> <input type="button" value="新規キャッシュタイマーの削除"/></p> </div> |
| <p>キャッシュタイマーの削除</p>             | <p>キャッシュタイマーの削除は、システムが現在のジョブのキャッシュをどのように削除するかを管理します。</p> <div data-bbox="540 1163 1414 1633"> <p>実行 <input type="text" value="完全削除"/> 毎日 <input type="text" value="毎日"/> 毎 <input type="text" value="1"/> 日</p> <p>繰り返す <input type="button" value="+"/></p> <p>時間: <input type="text" value="20:58:00"/> <input type="button" value="🗑️"/></p> <p>タイムゾーン: <input type="text" value="Europe/Berlin"/></p> <p><input checked="" type="checkbox"/> 有効化されました</p> <p><input type="button" value="新規キャッシュ更新タイマー"/> <input type="button" value="新規キャッシュタイマーの削除"/></p> </div>                             |
| <p>キャッシュの保存と更新</p>              | <p>ボタンをクリックして、キャッシュを手動で保存および更新します。</p>   |

キャッシュタイマーの更新とキャッシュタイマーの削除は、「削除」() ボタンをクリックすることにより、他のトリガーと同様に削除することができます。また、「削除を元に戻す」() 操作が両方のメニューに対して、保存、またはページの再ロードが適用されるまで使用することができます。「複製」ボタン()により、現在のトリガーと同じ設定を持つトリガーを作成することができます。

以下も参照してください:

- [ジョブの結果をキャッシュする](#) (例)

## 5.16 キュー設定の定義

キューの設定によりサーバリソースの使用をさらに柔軟的に管理できます。例えば、キュー構成により一度に並列に作動するジョブインスタンスの数を制限することができます。

実行キューはジョブの「プロセッサ」です。これはジョブインスタンスがどのように実行されるかを管理します。実行するためには、各ジョブはターゲット実行キューに割り当てられます。キューは（キューに割り当てられているすべてのジョブの）ジョブインスタンスが一度に実行される数、実行間の遅延を管理します。デフォルトで、キュー設定はジョブに対してローカルですが、キューを複数のジョブにシェアされるスタンドアロbjectとして定義することもできます。キューはジョブに対してローカルであること、または、複数のジョブで共有されることができます。複数のジョブが同じキューに割り当てられている場合、実行のためにそのキューは共有されています。

更に、適切な（特権とは異なる）コンテナパーミッションを有する場合のみキューを確認、または、キューにジョブを割り当てることができます。具体的には、キューを作成するためユーザーが実行キューを定義する特権を持つ必要があります。[特権のくみ](#)を参照してください。更に（特権とは異なる）適切なパーミッションがある場合のみユーザーはキューを確認、またはキューにジョブを割り当てることができます。[パーミッションのくみ](#)を参照してください。これは、キューにジョブを割り当てることができることを意味します。キューへのアクセスを制限するには、キューが定義されている個所にコンテナをナビゲートし、コンテナのパーミッションをロール `authenticated` のために「キュー-アクセス無し」に変更します。次に、パーミッション「キュー-使用」を特定のロール、または、必要なユーザーに割り当てます。詳細に関しては、[public コンテナへのアクセスを制限する](#)を参照してください。

### スタンドアロキューの作成

キューをスタンドアロbjectとして作成する

1. 「構成」をクリックして、キューを作成するコンテナにナビゲートします。
2. 「作成」をクリックして、「キューの作成」をクリックします。

## 以下にキューを作成 /public

キュー名:

キューの説明:

### キュー設定

実行間の最短時間:  秒

並列実行の最大数:  インスタンス

3. キューの名前、そして任意で、詳細を入力します。すべての設定の参照に関しては、下の「キューの設定」を参照してください。
4. [保存]をクリックします。

### ローカルキューの定義

スタブアロンキューの作成の代わりに、ジョブ内でローカルキュー設定を定義することができます。これを行うにはローカルキューの定義 オプションをジョブ構成ページから選択し、キューの設定を指定してください。下のイメージはデフォルトのキュー設定を表示しています。

#### キュー設定

キューを使用したジョブの実行:  既存のキューを選択:

ローカルキューを定義:

実行間の最短時間:  秒

並列実行の最大数:  インスタンス

既存のキューの選択 オプションを選択すると、以前に定義されているスタブアロン、外部キューを指定する必要があります。実行間の最短時間と並列実行の最高数 設定のデフォルト値に関しては、下のキュー設定のセクションを参照してください。

### キューの設定

構成のために使用することのできる設定は下にリストされるとおりです。

|          |   |
|----------|---|
| キューの名前   | <p>キューを識別する名前を入力します。これは必須のフィールドで、スペースで開始し、終了する必要があります。または文字、数字、単一スペース、アンダースコア("_"), ダッシュ("-") および終止符(".") 文字のみを含むことができます。</p> <p>(ローカルではなく) キューがスタブアロン キューとして定義されている場合のみこのフィールドは適応可能です。</p>   |
| キューの説明   | <p>オプションで、キューオブジェクトのための説明を入力します。</p> <p>(ローカルではなく) キューがスタブアロン キューとして定義されている場合のみこのフィールドは適応可能です。</p>  |
| 実行間の最短時間 | <p>実行キューは現在アクティブなルールに従い実行スロットを提供します。各スロットは連続的にジョブインスタンスを実行します。</p> <p>「実行間の最短時間」設定は、次のジョブインスタンスをすぐにピックアップしようとして、スロットをジョブインスタンスの完了後、短い間、使用中とマークします。この機能により、この実行キューのために最大のスループットを削減し、同じマシン上の他の実行キューと他の処理のためにCPU タイムを提供します。</p>  |
| 並列実行の最高数 | <p>このオプションはキュー上で使用可能な実行スロットの数量を定義します。設定は現在のキュー内で並列に実行される同じジョブのインスタンスの数量を決定し、各スロットはインスタンスをシーケンス順に実行します。しかしながら、並列で実行を許可するインスタンスの数量は使用可能なマシンリソースを競合することにご注意下さい。この値の増加は、集中したI/O オペレーションを実行しない、または大量のCPU 時間を必要としない軽量のジョブを処理するキューが受け入れることが可能です。(重量のジョブインスタンスを一度に処理するためにデフォルトの設定 1 は最も保守的で、リソース集中型のジョブを処理するジョブに最適です。</p> <p>このオプションはFlowForce Server により受け入れられる Web サービスとしてジョブを公開するクライアントからの並列 HTTP リクエストの最大数に影響を与えません。詳細に関しては</p> |

[FlowForce Server プールスレドの再構成](#)を参照してください。

## 5.17 Windows ネットワークパスの参照

ジョブを作成する際、通常 FlowForce Server が作動するマシン上で、または ネットワーク上のファイルパスを参照します。マップされたネットワークドライブなどの Windows ネットワークパスを参照する場合、Universal Naming Convention (UNC) 構文を使用します。これはドライブ文字は システムに対してグローバルではないため、また、各ログインセッションは自身のドライブ文字に割り当てられているため、これはドライブ文字がシステムに対してグローバルではなく、ログインセッションが自身のドライブ文字に割り当てられているため必須です。

UNC 構文は以下のフォームをとります:

```
\\server\sharedfolder\filepath
```

該当する箇所:

- `server` が(DNS により定義される)ネットワーク内のサーバー名を参照する箇所。
- `sharedfolder` が管理者により定義されるレベルを参照する箇所(例えば、`admin$` が通常オペレーティングインストールのルートディレクトリである箇所)。
- `filepath` がシェアの下のサブディレクトリを参照する箇所。

## 5.18 FTP サポート

FlowForce Server 内でジョブを構成する場合、FTP サーバーに接続する必要がある場合があります。例えば、ジョブがFTP サーバーからファイルを抽出する必要がある場合、FTP サーバー上のディレクトリのコンテンツをリスト、または、FTP を介してファイルを削除する必要がある場合があります。上記のようなオペレーションを有効化するには FlowForce Server は以下のプロトコルを使用して FTP サーバーへの接続のためのサポートを提供します: FTP、FTPS、SFTP。

- FTP - ファイル転送プロトコル(FTP) は FlowForce Server とリモート FTP サーバー間の転送を可能にします。プロトコルは TCP ポート 21 に割り当てられています。FTP プロトコルはファイル転送のための安全なモードのために設計されていません。データを安全に転送するには FTPS または SFTP を以下で説明されているように使用してください。
- FTPS - プレーン FTP の使用の代替法は安全な TLS/SSL チャンネルを介して FTP を参照する FTP セキュア (FTPS) プロトコルです。FlowForce Server は FTP の使用を TLS/SSL 明示的な暗号化を使用した FTP として既知の [RFC 4217](#) に従ってサポートしています。FTP または FTPS を介して接続を構築するには [/system/ftp](#) コンテナ内で使用可能な関数を使用してください。

**メモ** with TLS/SSL 明示的な暗号化を使用した FTP は非奨励と考えられサポートされていません。

- SFTP - 安全な FTP (SFTP) はデータを Secure Shell (SSH) を使用して安全に変換するメソッドです。SSH を使用した FTP はプレーン FTP と比較して大幅に異なる作動をします。このため、FlowForce Server Advanced Edition はプロトコルの具体的な使用を可能にする個別のビルトイン関数を提供します。

**メモ** SFTP サポートは FlowForce Server Advanced Edition でのみ使用することができます。

## 5.19 ジョブのインポートとエクスポート

ジョブとデプロイされた MapForce マッピングと StyleVision 変換を含む他の構成オブジェクトを FlowForce Server からインポートすることができます。

- 他の作動中の FlowForce Server インスタンスへ(オンライン エクスポート)
- ファイルへ(オフライン エクスポート)

他の作動中の FlowForce Server インスタンスにオブジェクトをエクスポートする場合、エクスポートされたオブジェクトは、サーバーの Web 管理インターフェイスですぐ使用できるようになります。

オブジェクトをファイルにエクスポートする場合、FlowForce Server は、選択されたオブジェクトとその依存関係を含む .zip アーカイブを作成します。zip アーカイブは、エクスポートのオペレーションが実行された日時に従い、名前がつけられます。名づけの変換は以下の通りです: `export_YYYYMMDDThhmmss`。例えば、2015-2021 年の 8 月 6 日の 10:51:33 にエクスポートされたファイルは、`export_2015-20210806105133.zip` と名づけられます。

(インポートされたオブジェクトが保存先に存在せず、また、上書き可能と想定して次に zip アーカイブを同じ FlowForce Server インスタンスにインポートすることもできます。または他のインスタンスにインポートすることもできます。)

### 構成データのエクスポート

コンテナ内で特定の記録、または、コンテナ全体をエクスポートすることができます。どちらの場合でも、FlowForce は、エクスポート前に記録のリストをレビューすることのできるダイアログボックスを表示します。コンテナ全体をエクスポートするように選択した場合は、例えば、このダイアログボックスは、選択されたコンテナのすべての子の記録を表示します(ジョブまたは資格情報)。

オブジェクトをエクスポートする前に、FlowForce は、依存関係を持つ全てのオブジェクト、またはエクスポートするオブジェクトにより参照されるオブジェクトについて別のページで通知します。これにより、見逃された依存関係を一目で確認することができます。オブジェクトを実行中の FlowForce Server にエクスポートする場合、エクスポート先のサーバー上に各オブジェクトが存在するかを確認することができます。

セキュリティ上の理由から、FlowForce Server は以下に挙げられる重要なデータのカテゴリをエクスポートしません

- ジョブ内でローカルに定義されたパスワード
- 「スタンドアロン」資格情報の記録として使用されるパスワード
- (`/system/ftp` 関数などの)システム関数と共に保管されるパスワード

機密なデータのこれらのカテゴリ全てをエクスポートするために、エクスポート中にオプション「機密な情報のエクスポート」を選択します。このチェックボックスを選択すると、エクスポートされたアーカイブでは機密なデータがプレーンなテキスト書式で含まれることに注意してください。

このチェックボックスを選択しない場合、機密なデータはエクスポートされません。FlowForce Server にデータをインポートする場合、それぞれ個別のレコードを上書きすること、またはスキップすることを選択できます。上書きすることを選択した場合、機密なデータが空の値により置き換えられます。

作動中の FlowForce Server インスタンスへジョブをエクスポートする

1. 「構成」をクリックし、エクスポートする記録を選択します。コンテナ内の特定の記録を選択、または、コンテナ全体を選択することができます。
2. 「選択されたオブジェクトのエクスポート」をクリックします。

選択されたオブジェクトのエクスポート

| <input type="checkbox"/>            | 名前                    |
|-------------------------------------|-----------------------|
| <input checked="" type="checkbox"/> | db2osm.job            |
| <input checked="" type="checkbox"/> | my.credentials        |
| <input checked="" type="checkbox"/> | AirportStatus.mapping |

サーバーにエクスポート

サーバー:  ポート:

ユーザー:

パスワード:  SSLの使用:

ファイルにエクスポート

1. ダイアログボックスから、「サーバーにエクスポート」をクリックし、FlowForce Server のホスト名と保存先、および作動するポートを入力します。
2. FlowForce Server インスタンスの保存先にユーザー名とパスワードを入力します。そして、「エクスポート」をクリックします。FlowForce は、依存関係を確認し、エクスポートから除外を選択することのできるページエクスポートされる全ての記録を表示します。

### Export

| <input type="checkbox"/> Name                                       | Type             | Dependencies  | Remote Server                    |
|---|------------------|---|----------------------------------|
| <input type="checkbox"/> /public/my.credentials                     | credential       | Referenced by:<br>/public/ConvertProducts.job   | Exists on remote server          |
| <input checked="" type="checkbox"/> /public/ConvertProducts.mapping | MapForce mapping | Referenced by:<br>/public/ConvertProducts.job   | Will be created on remote server |
| <input checked="" type="checkbox"/> /public/ConvertProducts.job     | job              | Depends on:<br>/public/ConvertProducts.mapping<br>/public/my.credentials<br>/system/filesystem/copy | Will be created on remote server |
| <input type="checkbox"/> /system/filesystem/copy                    | function         | Referenced by:<br>/public/ConvertProducts.job   | Exists on remote server          |

**Export options**

Export sensitive data:

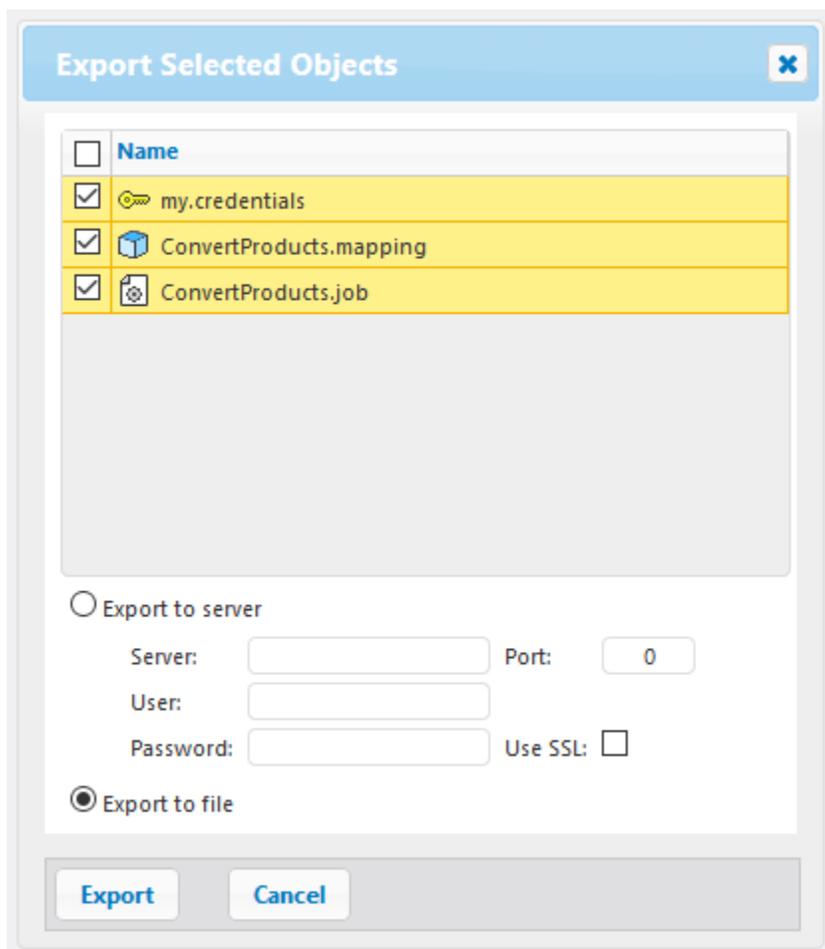
黄色い背景を持つ記録は、エクスポートされる記録です。黄色い背景色を持たない記録は、横のチェックボックスのチェックを解除することにより、エクスポートから除外した記録です。最後に、グレーの背景色を持つ記録は、ビルトインシステム関数上の依存関係を表しており、これら进行操作することできません

「リモートサーバー」列は、「リモートサーバーに存在、または存在しない可能性」を表示しています。zip アーカイブ FlowForce に再インポートする前に保存先ファイルに依存関係が存在する場合、エクスポートからこれらの記録を削除することができます。それ以外の場合、依存関係無しにエクスポートする場合、そして依存関係が保存先で存在しない場合、ジョブは失敗する可能性があります。成功した例も参照してください。

5. エクスポートするオブジェクトをクリックして選択し、「エクスポートの開始」をクリックします。

### ジョブをファイルにエクスポートする

1. 「構成」をクリックし、およびエクスポートする記録を選択します。コンテナ内の特定の記録を選択、または、コンテナ全体を選択することができます。
2. 「選択されたオブジェクトのエクスポート」をクリックします。



3. ダイアログボックスから、「ファイルにエクスポート」をクリックし、「エクスポート」をクリックします。FlowForce は、依存関係を確認、または、エクスポートから除外することのできるページで全ての記録を表示します。

### Export

| <input type="checkbox"/> Name                                       | Type ↕           | Dependencies  | Remote Server                             |
|---|------------------|---|---|
| <input type="checkbox"/> /public/my.credentials                     | credential       | Referenced by:<br>/public/ConvertProducts.job   | Might or might not exist on remote server |
| <input checked="" type="checkbox"/> /public/ConvertProducts.mapping | MapForce mapping | Referenced by:<br>/public/ConvertProducts.job   | Might or might not exist on remote server |
| <input checked="" type="checkbox"/> /public/ConvertProducts.job     | job              | Depends on:<br>/public/ConvertProducts.mapping<br>/public/my.credentials<br>/system/filesystem/copy | Might or might not exist on remote server |
| <input type="checkbox"/> /system/filesystem/copy                    | function         | Referenced by:<br>/public/ConvertProducts.job   | Might or might not exist on remote server |

**Export options**

Export sensitive data:

黄色い背景を持つ記録は、エクスポートされる記録です。黄色い背景色を持たない記録は、横のチェックボックスのチェックを解除することにより、エクスポートから除外した記録です。最後に、グレーの背景色を持つ記録は、ビルトインシステム関数上の依存関係を表しており、これら进行操作することはできません。

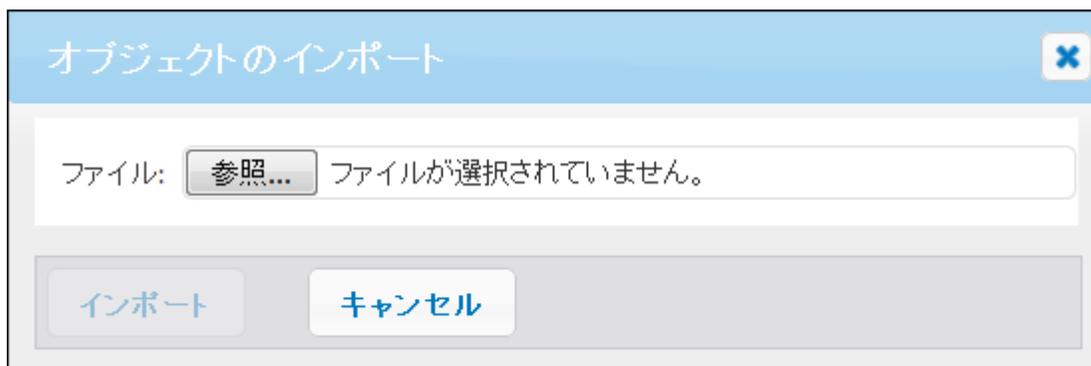
作動中のFlowForce Server インスタンスでは無くファイルをエクスポートしているため、エクスポートされたオブジェクトが保存先で存在するかを決定することはできません。この理由から、「リモートサーバー」列は「リモートサーバーに存在、または存在していない可能性」を表示しています。 .zip アーカイブ FlowForce に再インポートする際に保存先ファイルに依存関係が存在する場合、エクスポートからこれらの記録を削除することができます。それ以外の場合、依存関係が不足している箇所でデータのインポートを試みると、「操作に失敗しました。パスが存在しません」などのエラーが表示され、インポートは失敗します。以下を参照してください！ [成功した例](#)。

4. 「ファイルへエクスポートを開始する」をクリックします。使用中のブラウザ設定により、.zip アーカイブをローカルディレクトリに保存するようプロンプトされるかもしれません。または、構成済みの保存先ディレクトリに自動的に、ブラウザが自動的に保存する可能性もあります。

## 構成データのインポート

ファイルからジョブをインポートします:

1. 「構成」をクリック、「オブジェクトのインポート」をクリックします。



1. ダイアログボックスから、「参照」をクリックし、FlowForce Server からエクスポートされた、ソース.zip アーカイブを選択します。
2. 「インポート」をクリックします。FlowForce Server は、インポートされた記録を個別のページで依存関係と共に表示します。「現状」列は、「インポート」ボタンをクリックされた後の各記録ごとの起こるかの情報に関して通知します。

Import

| <input type="checkbox"/>            | Name                              | Type       | Dependencies   | Current State       |
|-------------------------------------|-----------------------------------|------------|--|---------------------|
| <input checked="" type="checkbox"/> | /public/my.credentials            | credential | Referenced by:<br>/public/Cleanup  | Will be overwritten |
| <input checked="" type="checkbox"/> | /public/Cleanup                   | job        | Depends on:<br>/public/my.credentials<br>/system/maintenance/cleanup-files | Will be created     |
| <input type="checkbox"/>            | /system/maintenance/cleanup-files | function   | Referenced by:<br>/public/Cleanup  | Exists              |

Import Cancel

3. 「インポート」をクリックします。

### 5.19.1 構成データのエクスポート

コンテナ内の特定のレコード、またはコンテナ全体をエクスポートすることができます。どちらの場合でも FlowForce はエクスポート前のレコードのリストをレビューすることができるダイアログボックスを表示します。エクスポートのためにコンテナ全体を選択するお、このダイアログボックスは選択されたコンテナ（ジョブまたは資格情報）のすべての子レコードを表示します。

オブジェクトのエクスポート前に FlowForce はエクスポートを希望するオブジェクトに依存する（または参照されている）すべてのオブジェクトに関して個別のページで通知します。これは一目で不足する依存関係を確認する助けになります。作動する FlowForce Server にオブジェクトをエクスポートする場合、各オブジェクトが目的先のサーバーに既存しているかを確認できます。

デフォルトではFlowForce Server は機密データの以下のカテゴリをエクスポートしません

- ジョブ内でローカルに定義されているパスワード
- 「スタンドアロン」資格情報レコードとして使用可能なパスワード
- ([/system/ftp](#) 関数)などのシステム関数に保管されているパスワード

機密データのこれらのカテゴリをインポートするには、エクスポート中に「機密なデータのエクスポート」を選択してください。チェックボックスを選択すると、エクスポートされたアーカイブはプレーンテキスト形式の機密データが含まれることにご注意ください。

チェックボックスを選択しないと、機密なデータはエクスポートされません。FlowForce Server データをインポートし直す場合、各レコードを上書きする、またはスキップするオプションがあります。上書きを選択する場合、既存の機密データは空の値と置き換えられます。

作動中のFlowForce Server インスタンスに構成データをエクスポートする方法:

1. 「構成」、をクリックし、エクスポートするレコードを選択します。コンテナ内の特定のレコード、またはコンテナ全体を選択することができます。
2. 「選択されたオブジェクトのエクスポート」をクリックします。

3. ダイアログボックス内で「サーバーにエクスポート」をクリックして、目的 FlowForce Server のホスト名と作動先のポートを入力します

4. 目的 FlowForce Server インスタンスユーザー名とパスワードを入力し、「エクスポート」をクリックします。FlowForce は依存関係を確認し、エクスポートから削除することのできるページエクスポートするすべてのレコードを表示します。

| <input type="checkbox"/>            | Name                            | Type             | Dependencies  | Remote Server                    |
|-------------------------------------|---------------------------------|------------------|---|----------------------------------|
| <input type="checkbox"/>            | /public/my.credentials          | credential       | Referenced by:<br>/public/ConvertProducts.job   | Exists on remote server          |
| <input checked="" type="checkbox"/> | /public/ConvertProducts.mapping | MapForce mapping | Referenced by:<br>/public/ConvertProducts.job   | Will be created on remote server |
| <input checked="" type="checkbox"/> | /public/ConvertProducts.job     | job              | Depends on:<br>/public/ConvertProducts.mapping<br>/public/my.credentials<br>/system/filesystem/copy | Will be created on remote server |
| <input type="checkbox"/>            | /system/filesystem/copy         | function         | Referenced by:<br>/public/ConvertProducts.job   | Exists on remote server          |

**Export options**

Export sensitive data:

黄色い背景色のレコードはエクスポート済みのレコードです。黄色い背景色が無いレコードはチェックボックスの横の選択がクリアされているエクスポートに含まれていないレコードです。最後に、灰色表示されているレコードはビルドインシステム関数上の依存関係を示しています。そのためこれらに対してのアクションを取ることができません。

「リモートサーバー」列はファイルの目的先で存在するかを示しています。目的先に依存関係が既存の場合、このようなレコードをエクスポートから除外することができます。それ以外の場合、依存関係無しでエクスポートし、依存関係が目的先に存在しない場合はこのようなジョブは失敗する可能性があります。[成功した例](#)を参照してください。

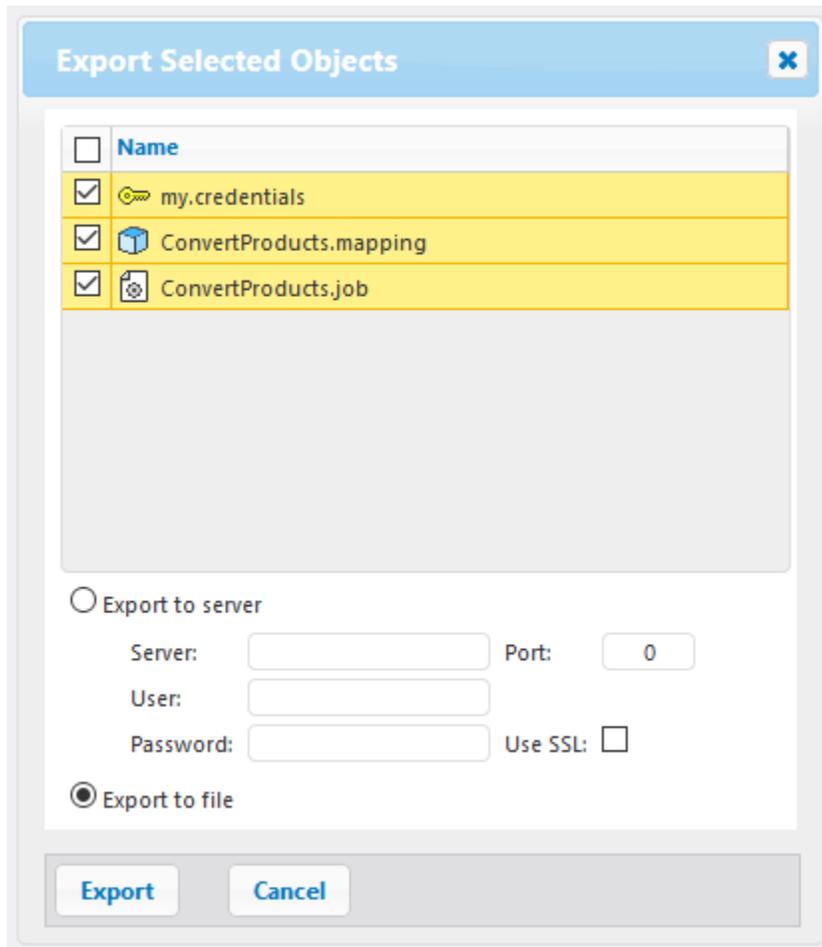
上記のとおり「機密な情報のエクスポート」オプションによりパスワードをエクスポートパッケージオプションで含むことができます。安全上の理由から、FlowForce Server からプレーンテキストでこのような機密データを転送する必要がある場合以外、このチェックボックスを選択することは奨励されません。

エクスポートされているリストが型「資格情報」を含んでおり、ターゲットがバージョン 2019r3 または依存の場合、チェックボックス FlowForce 2019r3 または以前ののために資格情報をダウングレードするが選択されている必要があります。そのリリース後、新規の資格情報レコードに FlowForce の古いバージョンと互換性を持たせるために資格情報が新規の「使用の許可」オプションを取得します。「使用を許可」オプションの詳細については[資格情報の定義](#)を参照してください。

5. エクスポートするオブジェクトをクリックし「エクスポートを開始」をクリックします。

ジョブをファイルにエクスポートします:

1. 「構成」、をクリックし、エクスポートするレコードを選択します。コンテナ内の特定のレコード、またはコンテナ全体を選択することができます。
2. 「選択されたオブジェクトのエクスポート」をクリックします。



3. ダイアログボックス内で「ファイルにエクスポート」をクリックして「エクスポート」をクリックします。FlowForce は依存関係を確認し、エクスポートから削除することのできるページエクスポートするすべてのレコードを表示します。

### Export

| <input type="checkbox"/> Name                                       | Type             | Dependencies  | Remote Server                             |
|---|------------------|---|---|
| <input type="checkbox"/> /public/my.credentials                     | credential       | Referenced by:<br>/public/ConvertProducts.job   | Might or might not exist on remote server |
| <input checked="" type="checkbox"/> /public/ConvertProducts.mapping | MapForce mapping | Referenced by:<br>/public/ConvertProducts.job   | Might or might not exist on remote server |
| <input checked="" type="checkbox"/> /public/ConvertProducts.job     | job              | Depends on:<br>/public/ConvertProducts.mapping<br>/public/my.credentials<br>/system/filesystem/copy | Might or might not exist on remote server |
| <input type="checkbox"/> /system/filesystem/copy                    | function         | Referenced by:<br>/public/ConvertProducts.job   | Might or might not exist on remote server |

**Export options**

Export sensitive data:

黄色い背景色のレコードはエクスポート済みのレコードです。黄色い背景色が無いレコードはチェックボックスの横の選択がクリアされているエクスポートに含まれていないレコードです。最後に、灰色表示されているレコードはビルドインシステム関数上の依存関係を示しています。そのためこれらに対してのアクションを取ることができません。

ファイルエクスポートし、FlowForce Server インスタンスを実行しないため、エクスポートされたオブジェクトが目的先に存在すると決定することはできません。この理由から、「リモートサーバー」列が「リモートサーバーに存在、または存在しない」を表示する可能性があります。zip アーカイブを FlowForce にインポートし直す場合、依存関係が目的先に存在する場合、このようなレコードをエクスポートから除外することができます。不確かな場合はすべての依存関係をエクスポートしてください。それ以外の場合、依存関係が不足している個所からデータをインポートしようとする、インポートが「操作に失敗しました」などのメッセージと共に失敗します。「バグは存在しません。」[作業済みのサンプル](#)を参照してください。

4. 「ファイルへのエクスポートを開始する」をクリックします。ブラウザ設定により、.zip アーカイブをローカルディレクトリに保存するようプロンプトされる、またはブラウザが自動的に構成済みの目的ディレクトリに保存する可能性があります。

### 5.19.2 機密な情報を含む、または含まないオプション

FlowForce Server からデータをエクスポートする場合、エクスポートから特定のオブジェクトを除外することができます。以下のサンプルは2つのアプローチ間の違いを説明しています。

#### サンプル1: 機密なデータを除外する

同じコンテナ（「my.credentials」）内の資格情報レコードを参照するジョブ（「AddNumbers」）が存在すると仮定します。

| <input type="checkbox"/> 名前             | 型    |
|---|------|
| <input type="checkbox"/> my.credentials | 資格情報 |
| <input type="checkbox"/> AddNumbers     | ジョブ  |

「機密なデータをエクスポート」オプションを選択せずファイル(.zip アーカイブ)に両方のオブジェクトをインポートする場合、以下の状態が発生します:

1. ジョブがエクスポートされます。
2. 証明書レコードがパスワード無しでエクスポートされます。

.zip アーカイブを2つのオブジェクトが存在しないFlowForce Server 環境にインポートする場合、両方のオブジェクトが作成されます。証明書レコードに関連したパスワードは空になります。

ターゲット環境内にオブジェクトが既存の場合、上書き、または 対応するチェックボックスをクリアして、スキップすることができます:

| インポート   |      |   |         |
|---|------|---|---------|
| <input type="checkbox"/> 名前                                     | 型    | 依存関係  | 現状      |
| <input checked="" type="checkbox"/> /public/Jobs/my.credentials | 資格情報 | 参照元:<br>/public/Jobs/AddNumbers                               | 上書きされます |
| <input checked="" type="checkbox"/> /public/Jobs/AddNumbers     | ジョブ  | 依存先:<br>/public/Jobs/my.credentials<br>/system/compute-string | 上書きされます |

レコードを上書きすることを選択した場合、以下の状態が発生します:

1. FlowForce Server 内に既存のジョブは、.zip アーカイブからのジョブにより上書きされます。
2. FlowForce Server 内の既存の証明書レコードは、.zip アーカイブからのものにより上書きされますが、目的先のパスワードは空のままです。

資格証明を上書きした場合、既存の資格証明はそのまま保管されます。

### サンプル 2: 機密なデータを含む

上に示されるように2つの同じレコードをエクスポートし、「機密なデータのエクスポート」オプションをエクスポート中に選択するとします。この場合以下が発生します: この場合、以下が発生します:

1. ジョブはエクスポートされます。
2. 資格情報レコードはパスワードを含みエクスポートされます。

.zip アーカイブを2つのオブジェクトが存在しないFlowForce Server 環境にインポートする場合、両方のオブジェクトが作成されます。資格情報レコードに関連するパスワードは.zip ファイルからのものになります。

ターゲット環境にオブジェクトが存在する場合、これらを上書きまたは対応するチェックボックスをクリアしてスキップすることができます。レコードを上書きする場合、以下が発生します:

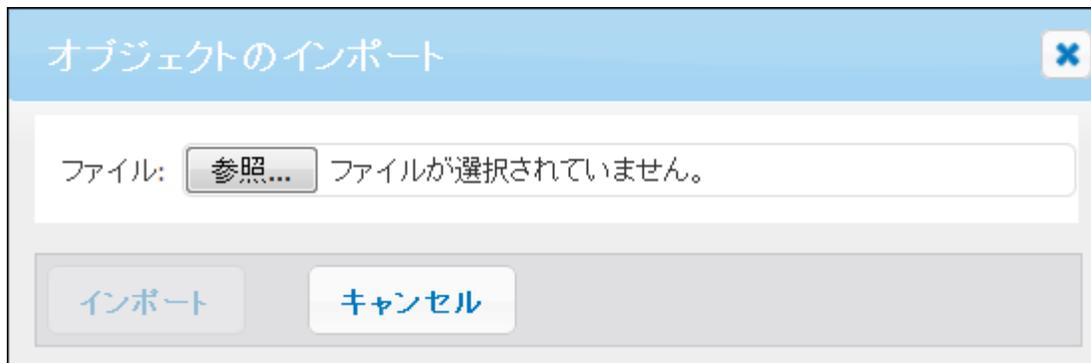
1. FlowForce Server 内に既存のジョブは、.zip アーカイブからのジョブにより上書きされます。
2. FlowForce Server 内で既存の資格情報レコードは.zip アーカイブからのレコードにより上書きされます。エクスポート先のパスワードも.zip アーカイブからのものにより上書きされます。

資格証明を上書きした場合、既存の資格証明はそのまま保管されます。

### 5.19.3 構成データのインポート

以前にインポート済みのアーカイブをインポートする方法:

1. 「管理」をクリックし「オブジェクトのインポート」をクリックします。



2. ダイアログボックスで「参照」をクリックし、FlowForce Server からインポート済みのノース.zip アーカイブを選択します。
3. 「インポート」をクリックします。FlowForce Server は個別のページにインポートされるレコードビルドを依存関係と共にを表示します。「現在の状態」列はインポート ボタンをクリック後に各レコードに何か起こるかを通知します。

Import

| <input type="checkbox"/>            | Name                              | Type       | Dependencies   | Current State       |
|-------------------------------------|-----------------------------------|------------|--|---------------------|
| <input checked="" type="checkbox"/> | /public/my.credentials            | credential | Referenced by:<br>/public/Cleanup  | Will be overwritten |
| <input checked="" type="checkbox"/> | /public/Cleanup                   | job        | Depends on:<br>/public/my.credentials<br>/system/maintenance/cleanup-files | Will be created     |
| <input type="checkbox"/>            | /system/maintenance/cleanup-files | function   | Referenced by:<br>/public/Cleanup  | Exists              |

Import Cancel

4. 「インポート」をクリックします。

FlowForce Server のターゲットインスタンス内で見つけることのできない外部インスタンスをアーカイブが持つ場合「現在の状態」列は「不在」の状態を表示します。対処方法に関しては [不足する依存関係の処理方法](#) を参照してください。

### 5.19.4 不足する依存関係の処理方法

FlowForce Server からデータをエクスポートする際、エクスポートから特定のオブジェクトを常に除外することができます。しかしながら、オブジェクトの一部は他のオブジェクト上に依存関係を持つ場合があります。依存関係を依存するオブジェクトと共にエクスポートしないと、データを FlowForce Server に再度エクスポートする際にエラーを引き起こす可能性があります。下のサンプルはこの影響を理解し、問題の対処することを意図としています。

2つのレコードが以下に表示されており、FlowForce Server から資格情報レコードをエクスポートせずジョブのみエクスポートすると仮定します:

| エクスポート  |      |   |                       |
|---|------|---|-----------------------|
| 名前  | 型    | 依存関係  | リモートサーバー              |
| <input checked="" type="checkbox"/> /public/Jobs/AddNumbers | ジョブ  | 依存先:<br>/public/Jobs/my.credentials<br>/system/compute-string | リモートサーバーに存在しないかもしれません |
| <input type="checkbox"/> /public/Jobs/my.credentials        | 資格情報 | 参照元:<br>/public/Jobs/AddNumbers                               | リモートサーバーに存在しないかもしれません |

この場合、以下が発生します:

1. このジョブはエクスポートされます (か不足する依存関係が存在します)
2. 資格情報レコードはエクスポートされません。

オブジェクト /public/Jobs/my.credentials が存在するFlowForce Server 環境にアーカイブをインポートしようと試みると、不足する依存関係を解決することができるのでインポートは可能となります。

| インポート   |      |   |        |
|---|------|---|--------|
| 名前  | 型    | 依存関係  | 現状     |
| <input checked="" type="checkbox"/> /public/Jobs/AddNumbers | ジョブ  | 依存先:<br>/public/Jobs/my.credentials<br>/system/compute-string | 作成されます |
| <input type="checkbox"/> /public/Jobs/my.credentials        | 資格情報 | 参照元:<br>/public/Jobs/AddNumbers                               | 存在する   |

しかしながら、参照される資格情報がターゲット環境に存在しない場合、インポートダイアログボックスは以下のようになります:

| インポート   |      |   |        |
|---|------|---|--------|
| 名前  | 型    | 依存関係  | 現状     |
| <input checked="" type="checkbox"/> /public/Jobs/AddNumbers | ジョブ  | 依存先:<br>/public/Jobs/my.credentials<br>/system/compute-string | 作成されます |
| <input type="checkbox"/> /public/Jobs/my.credentials        | 資格情報 | 参照元:<br>/public/Jobs/AddNumbers                               | 存在しない  |

上のイメージ内では、資格情報レコードは資格情報として識別されません。具体的には型は「資格情報」ではなく「構成」となります。またアイコンも異なります。型「構成」は型が未知のジェネリックな構成オブジェクト(他のジョブまたは資格情報)を指します。このサンプルでは「my.credentials」構成オブジェクトエクスポートされておらず、エクスポートされたメッセージは参照/スクリプト以外の正確な型の情報は含まれていません。このため、FlowForce Server へのデータのインポートは以下のようなエラーを引き起こす可能性があります:「操作が失敗しました。バインド先が存在しません。」

このエラーをフィックスするには、エラーメッセージに表示されるスクリプトで不足するレコード(この場合「my.credentials」レコード)を作成しインポートを再度試みます。

\* 資格情報の参照はスタンドアロン資格情報オブジェクトへの参照である可能性があり、一部の場合ローカル資格情報を含むジョブへの参照である場合があります [ジョブから資格情報を参照する](#)。

## 6 ジョブの実行のモニタリング

ジョブトリガーの条件を満たすと、または Web サービスの呼び出しによりオンデマンドでトリガーされると、そのジョブのインスタンスが作動を開始します。ジョブインスタンスの出力、1つの状態から他の状態への切り替え、および他の実行の詳細は FlowForce Web によりログされます。ジョブの実行とその結果を2つのレベルから取得することができます。

1. ホームページから、ホームページがジョブの実行に関する以下のデータのクイック概要を提供しています: 最近および実行中のジョブ、アクティトリガーとサービス。
2. [ログビュー](#) ページから、ログビューページは [メインログのビュー](#) で説明されている通り実行済みのジョブに関する完全な詳細を表示しています。

このトピックの残りはホームページ内で使用可能なジョブ管理データについて説明しています。

### 最近および実行中のジョブ

最近および実行中のジョブセクション内では、最近および実行中のジョブのステータスを (20 件まで) 確認することができます。すべての最近と実行中のジョブを個別のページで確認するには、すべてのアクティブなトリガーとサービスを表示するをクリックしてください。

| Recent and Running Jobs |   |                     |                     |                                    |                          |
|-------------------------|---|---------------------|---------------------|------------------------------------|--------------------------|
| Instance ID             | Job   | Activation Time     | Last Action         | Status                             |                          |
| 17                      |  /public/Examples/MyTask   | 2021-02-02 17:18:00 | 2021-02-02 17:18:29 | Finished successfully after step 1 |                          |
| 18                      |  /public/Examples/MyTask  | 2021-02-02 17:19:00 | 2021-02-02 17:19:29 | Finished successfully after step 1 |                          |
| 19                      |  /public/Examples/MyTask | 2021-02-02 17:20:00 | 2021-02-02 17:20:00 | Running step 1                     | <a href="#">Stop job</a> |

[Show all recent and running jobs](#)

作動中のジョブインスタンスに加え、テーブルは、作動中のジョブだけでなく何らかの理由で失敗したジョブを含む最近完了したジョブを表示します。このようなジョブは実行が完了した後短い期間 (約 1-2 分) のみ表示されています。「ログ」ページからの各ジョブインスタンスの履歴を常にチェックすることができます。[ジョブログの確認](#)を参照してください。

最近および実行中のジョブのグリッドには以下の列が含まれています:

- **インスタンス ID**。開始すると、各ジョブインスタンスは「インスタンス ID」列内で表示される一意の ID が割り当てられます。インスタンス ID は後に「ログ」ページからの各ジョブインスタンスの実行ステータスをトラックする手助けをします。ジョブの実行中テーブル内のインスタンス ID をクリックすると、「ログ」ページがダイレクトされ、選択されたジョブインスタンスに関する情報のみが表示されます。(例えば、一意のファイル名を作成するなど) ジョブの一意のインスタンス ID をジョブ内で使用するには、[instance-id](#) 式関数を使用すると可能になります。
- **ジョブ**。列はこのジョブインスタンスの構成を検索することが可能なリンクを示しています。
- **アクティベーション日時**。ジョブインスタンスの実行が開始した日時。
- **最後のアクション**。最後の実行ステータスの日時。
- **ステータス**。ジョブステータスはページが最後に更新された時のものです。すべてのジョブインスタンスのステータスを参照するには、[ジョブのステータス](#)を参照してください。

**ジョブの停止** ボタンは実行中のジョブのためのみ表示されます。実行中のジョブインスタンスを停止することができます。これはデータの破損を招く場合があり、特別な場合のみ行ってください。[ジョブの停止](#)を参照してください。

[最近と実行中のジョブグリッドは自動的に更新されません。全てのジョブの最終のステータスを取得するには、「グリッドの再ロード」



ボタンをクリックします。

### アクティブなタイマー

ページのアクティブなタイマーセクションはトリガーを介してまたはWeb サービスとして実行スケジュールされた最近のジョブを20件まで表示します。このようなジョブのフルリストを確認するには、すべてのアクティブなトリガーとサービスを全て表示するをクリックします。

| Active Timers |   |                     |  |
|---------------|---|---------------------|--|
| Type          | Job   | Next run ↕          | Info   |
| timer         |  /public/Examples/MyTask | 2021-02-02 17:27:00 | Fire from 00:00 to 24:00 every 1 minutes (as in Europe/Vienna) |

[Show all active triggers and services](#)

[アクティブなトリガーとサービス]のグリッドには以下の列が存在します:

- **型** - 1はトリガー(ウォッチ、タイマー、サービス)の型を指しています。ラベル「ウォッチ」はファイルシステムトリガーまたはHTTPトリガーを参照します。追加の詳細のために「情報」列を参照してください。
- **ジョブ** - トリガーまたはサービスが定義されるジョブのパスを指定します。ジョブの構成ページを開くためリンクをクリックします。
- **次の実行** - ウォッチトリガーに適用されます。この列はトリガーが次を実行する際に示されます。
- **情報** - Web サービスとして公開されるジョブの場合 Web サービスのURLなどの追加情報を提供します。ウォッチ、またはタイムトリガーのために、この列は現在のトリガーの構成をまとめています。
- **サービスURL** - Web サービスが対応するジョブにより公開されている箇所でURLを指定します。これは[Web サービスを公開するジョブ](#)でのみ適応可能です。

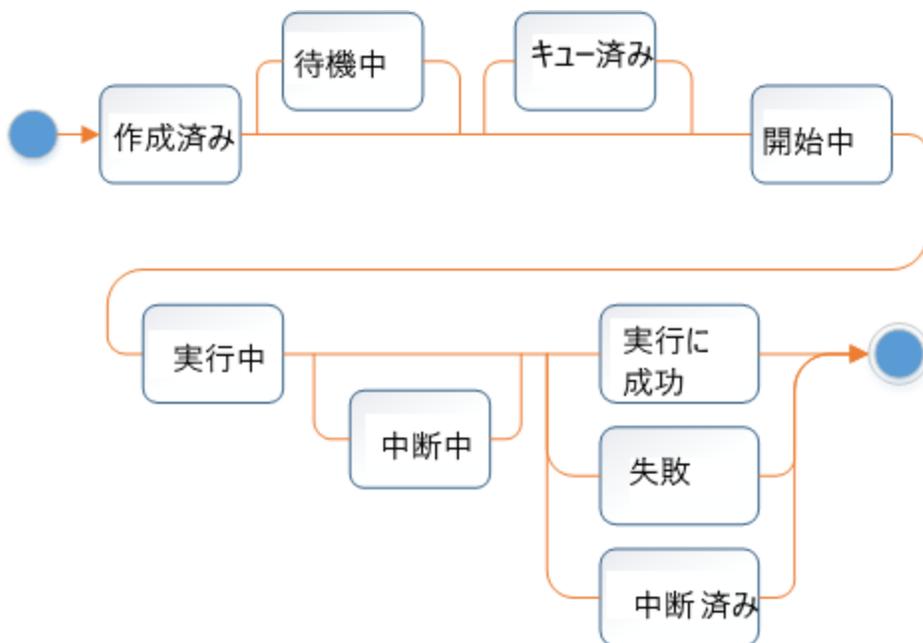
アクティブなトリガーとサービスグリッドは自動的に更新されません。グリッドの再ロード  ボタンをクリックしてページを更新します。

## 6.1 ジョブのステータス

ライフタイム全体で、ジョブインスタンスは、「実行中のジョブ」テーブル内の「ステータス」列に表示されているとおり、多種のステータスを取得します。

| Running Jobs |                |                     |                     |                                    |          |
|--------------|----------------|---------------------|---------------------|------------------------------------|----------|
| Instance ID  | Job            | Activation Time     | Last Action         | Status                             |          |
| 29           | /public/MyTask | 2019-02-05 12:26:00 | 2019-02-05 12:26:29 | Finished successfully after step 2 |          |
| 30           | /public/MyTask | 2019-02-05 12:27:00 | 2019-02-05 12:27:00 | Running step 2                     | Stop job |

ジョブインスタンスがどのようにつの状態からもう一つの状態に変更するかについて次のダイアグラムは示しています。FlowForce Server サービスの喪失、またはネットワークの中断が発生してはならないことが前提です。ステータスの一部は、短いスパンである場合があり、ユーザーインターフェイスで常に表示されているわけではないことに注意してください。



以下のテーブルは、ジョブのすべての可能なステータスをリストしています。

| ステータス | 説明   |
|-------|--|
| 作成済み  | これは、アクションがとられる前のジョブの最初の状態です。次の可能な状態は「開始」、「スロット待ち」、「失敗」、「失敗」、「中断済み」です。                              |
| 待機中   | ジョブインスタンスがファイルシステム、または、HTTP トリガーによりトリガーされた結果によりこの状態は引き起こされる可能性があります。ジョブインスタンスがトリガーされていますが、アクションがまだ |

| ステータス         | 説明  |
|---------------|---|
|               | <p>「解決するまで N 秒待つ」オプションにより指定されている処理期間が経過してないことを示しています。</p>   |
| <p>スロット待ち</p> | <p>ジョブインスタンスがトリガーされていますが、実行キューは現在フルであることを示しています。実行キューの実行スロットに制限があり、ジョブインスタンスの指定された数のみ同じキュー上で変更して実行することができます。キューの設定の定義も参照してください。このキューに到達するインスタンスは、スロットが空くまで待機し、「スロット待ち」ステータスが与えられます。</p> <p>割り込みが起きない限り、通常「スロット待ち」の後のステータスは「開始中」です。</p>                      |
| <p>開始中</p>    | <p>このステータスは通常とても短いです。これは、ジョブインスタンスが空いているキューに割り当てられ、最初のステップがすぐに開始されることを意味します。</p>  |
| <p>実行中</p>    | <p>ジョブインスタンスが現在実行中であり、実行が完了するまでの状態であること、または、外部イベントが発生して、完了前に実行が停止されていることを指します。このステータスには、「実行中のステップ」{step} など関連したステップ番号が存在します。</p>  |
| <p>失敗</p>     | <p>このジョブインスタンスの実行は、完了し、失敗としてみなされます。これは最後のステータスで、ジョブインスタンスをこれ以上実行する試みは発生しません。</p> <p>「ステップ {step} の後に失敗しました」ステータスは追加で失敗がステップ番号に関連することを示しています。</p>  |
| <p>正常に完了</p>  | <p>これはジョブが正常に完了したことを示す最後の状態です。</p>  |
| <p>中断中</p>    | <p>FlowForce ユーザーが「ジョブの停止」ボタンをクリックしたため、ジョブインスタンスの実行を停止しようと試みます。ジョブの停止は、非同期なアクションのため、時間がかかる可能性があります。「中断済み」ステータスに切り替えられる前に、ジョブインスタンスは正常に完了する可能性があります。これが発生すると、ジョブは正常に完了したものと報告されます。</p> <p>前のステータスにステップ番号が存在する場合、「ステップ後に中断」{step} ステータスが「中止中」の代わりに表示されます。</p> |
| <p>中止済み</p>   | <p>ジョブインスタンスは実行を完了しました。通常、このステータスは、予期されないシャットダウン語に直接発生することを示しますが、ユーザーがジョブを停止したことを示します。これは最後の状態で、ジョブの一部が完了してないことを示しています。</p> <p>前のステータスにステップ番号が存在する場合、「ステップ後に中断」{step} ステータスが「中止済み」の代わりに表示されます。</p>  |
| <p>中断済み</p>   | <p>ジョブインスタンスの実行が中断されました（例えば、ネットワークケーブルが外れている場合、ワーカーマシンがクラッシュしている、または、シャットダウンされている場合、および、強制的に中断されているイベントの場合など）。中断済み状態の強制的なアクションです。ジョブインスタンスは再起動することができず、失敗として扱われます。ジョブの一貫性は保証されませんので、結果を手動で確認することをお勧めされます。</p>   |
| <p>追跡対象外</p>  | <p>FlowForce Server プロセスが現在ジョブをトラックできないことを示しています。これは FlowForce Server プロセスが強制的に終了された場合に起こる場合があります。</p> <p>「中断済み」と「追跡対象外」には若干の違いがあることにご注意ください。前者の場合、FlowForce Server はプロセスとして中断済みであることが既知ですが、後者の場合、何か発生したのか把握できていません。</p>                                    |

| ステータス  | 説明  |
|--------|---|
|        | <p>「中断済み」同様、ジョブの一貫性に保証はなく、結果を手動で確認することが奨励されています。</p>  |
| 置き換え済み | <p>このステータスはジョブが実際に何かを行う機会を得る前にジョブが中断されることを示しています。これは、重要な状態ではありません。FlowForce Server が異なる変更を検知し、前のジョブインスタンスをキャンセルして、新しいものを作成したことを意味します。</p> <p>例えば、このステータスはファイルシステム、または HTTP トリガーの結果としてトリガーされたジョブインスタンスのために発生する場合があります。「解決するまで N 秒待つ」オプションにより指定されている期間の前ジョブが変更したファイルのトリガーが解決された場合、ジョブインスタンスはこのステータスを取得する場合があります。</p> <p>このステータスは、以前に開始されたインスタンスがスロットを待っている間タイムトリガーと共に発生することもできます。</p> |

## 6.2 ジョブの停止

使用中のユーザーアカウント（または使用中のユーザーアカウントがメンバーであるロール）が特権を有することが前提で、現在作動中のジョブを停止することができます。[ジョブの停止](#) 特権を参照してください。

**警告:** ジョブの停止はデータの破損を招く場合があります、特別な場合のみ行ってください。

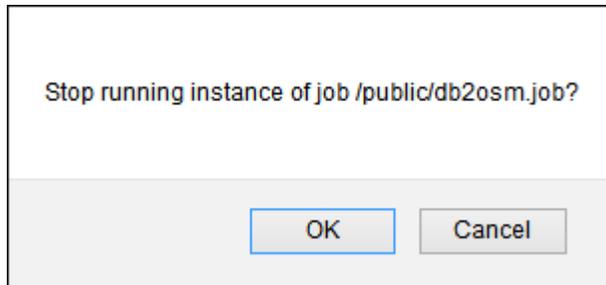
ジョブを停止すると、FlowForce Server は、最初ジョブの正常な停止を試みます。ジョブの停止が招く結果を想定できない場合、正常な停止プロセスは（定義されている場合）ジョブに関連したエラーの処理を試みます。正常な停止が可能であれば内場合、FlowForce は、一定の待ち時間の後、ジョブを強制的に停止します。FlowForce Server に強制的にジョブを停止するように命令するには、「ジョブの強制停止」ボタンをクリックします。

実行されているジョブを停止する

1. 「ホーム」をクリックします。現在実行中のジョブは実行中のジョブページのセクションに表示されています。

| Running Jobs |  |                     |                     |                |                          |
|--------------|--|---------------------|---------------------|----------------|--------------------------|
| Instance ID  | Job  | Activation Time     | Last Action         | Status         |                          |
| 61           |  /public/db2osm.job | 2014-06-18 14:58:00 | 2014-06-18 14:58:00 | Running step 1 | <a href="#">Stop job</a> |

2. 「ジョブの停止」をクリックします。FlowForce Server は、アクションを確認するようにプロンプトします:



3. 「OK」をクリックします。システムがジョブを停止しようと試みます。ジョブインスタンスが停止されると、関連する通知メッセージが表示され、対応するジョブの記録がハイライトされます。

| Stopping instance 61 of job /public/db2osm.job. |  |                     |                     |                              |                                |
|---|--|---------------------|---------------------|------------------------------|--------------------------------|
| Welcome!  |  |                     |                     |                              |                                |
| Running Jobs                                    |  |                     |                     |                              |                                |
| Instance ID                                     | Job  | Activation Time     | Last Action         | Status                       |                                |
| 61  |  /public/db2osm.job | 2014-06-18 14:58:00 | 2014-06-18 14:58:00 | Stopped while running step 1 | <a href="#">Force stop job</a> |

4. ジョブが通常に停止されなかった場合、「ジョブの強制停止」をクリックします。

## 6.3 メインログの確認

ログビューから FlowForce Server によりログされた特定のジョブに関する詳細を確認することができます。「ログ」メニューアイテム、または「ログビュー」ボタンが表示される他のコンテキストからログビューにアクセスすることができます。(例えば、各ジョブの構成ページは「ログビュー」ボタンを表示します)。

**メモ** デフォルトで、読み取りアクセスのあるジョブ全てのログをビューすることができます。FlowForce Server 内のすべてのジョブとイベントのグローバルなログをビューするには、使用中のアカウントが フィルターされていないログのビュー 特権を持っていることが必須です。

| Date                | Severity | User | Instance ID | Message  |
|---------------------|----------|------|-------------|--|
| 2020-09-17 16:06:08 | INFO     | root | 16          | Finished job execution: job /public/Examples/CopyFiles in queue /public/Examples/CopyFiles   |
| 2020-09-17 16:06:08 | INFO     | root | 16          | Starting instance 16.  |
| 2020-09-17 16:06:08 | INFO     | root |             | Service /CopyFilesService of /public/Examples/CopyFiles invoked                              |
| 2020-09-17 16:04:59 | INFO     | root | 15          | Finished job execution: job /public/Examples/HelloWorld in queue /public/Examples/HelloWorld |
| 2020-09-17 16:04:59 | INFO     | root | 15          | Starting instance 15.  |
| 2020-09-17 16:04:59 | INFO     | root |             | Service /HelloWorldService of /public/Examples/HelloWorld invoked                            |
| 2020-09-17 16:04:57 | INFO     | root | 14          | Finished job execution: job /public/Examples/CopyFiles in queue /public/Examples/CopyFiles   |
| 2020-09-17 16:04:57 | INFO     | root | 14          | Starting instance 14.  |
| 2020-09-17 16:04:57 | INFO     | root |             | Service /CopyFilesService of /public/Examples/CopyFiles invoked                              |
| 2020-09-17 16:04:55 | INFO     | root | 13          | Finished job execution: job /public/Examples/CopyFiles in queue /public/Examples/CopyFiles   |
| 2020-09-17 16:04:55 | INFO     | root | 13          | Starting instance 13.  |
| 2020-09-17 16:04:55 | INFO     | root |             | Service /CopyFilesService of /public/Examples/CopyFiles invoked                              |
| 2020-09-17 16:04:53 | INFO     | root | 12          | Finished job execution: job /public/Examples/CheckPath in queue /public/Examples/CheckPath   |
| 2020-09-17 16:04:52 | INFO     | root | 12          | Starting instance 12.  |
| 2020-09-17 16:04:52 | INFO     | root |             | Service /CheckPathService of /public/Examples/CheckPath invoked                              |
| 2020-09-17 16:04:37 | INFO     | root | 11          | Finished job execution: job /public/Examples/HelloWorld in queue /public/Examples/HelloWorld |
| 2020-09-17 16:04:37 | INFO     | root | 11          | Starting instance 11.  |
| 2020-09-17 16:04:37 | INFO     | root |             | Service /HelloWorldService of /public/Examples/HelloWorld invoked                            |
| 2020-09-17 16:04:35 | INFO     | root | 10          | Finished job execution: job /public/Examples/CopyFiles in queue /public/Examples/CopyFiles   |
| 2020-09-17 16:04:35 | INFO     | root | 10          | Starting instance 10.  |
| 2020-09-17 16:04:35 | INFO     | root |             | Service /CopyFilesService of /public/Examples/CopyFiles invoked                              |
| 2020-09-17 16:01:54 | INFO     | root | 9           | Finished job execution: job /public/Examples/CheckPath in queue /public/Examples/CheckPath   |
| 2020-09-17 16:01:54 | INFO     | root | 9           | Starting instance 9.   |
| 2020-09-17 16:01:54 | INFO     | root |             | Service /CheckPathService of /public/Examples/CheckPath invoked                              |
| 2020-09-17 16:01:48 | INFO     | root | 8           | Finished job execution: job /public/Examples/CheckPath in queue /public/Examples/CheckPath   |

### ログビュー

[情報]で設定された最小の重要度と共にデフォルトで[ログビュー]ページは最近[ログされた]イベントを表示します。[25 件以上のレコードを表示する] [100 件以上のレコードを表示する] または [500 件以上のレコードを表示する] ボタンをクリックして古いレコードをロードすることができます。

日付 列はログされたイベントが発生した日時を示しています。重要度 列はログされたイベントの重要度を[情報]、[警告]、[エラー]により示しています。ユーザー 列はイベントのオリジンを指定します(これは FlowForce サービス、Python セキュリティサービス 特定の FlowForce ユーザーであることができます)。

ジョブの各実行は一意のジョブインスタンスを生成します。一意のジョブインスタンスはインスタンスID 列によりトラッキングされています。インスタンスID によるイベントをフィルターするため、インスタンスID 列に表示されているリンクをクリックします。これは現在選択されているインスタンスに関する全ての情報を個別のページで開きます。[ログインスタンスの確認](#)を参照してください。これは適用不可能なため、ログされたイベントの一部はジョブインスタンスID が存在しません。この場合、例えば、ジョブ構成のため、イベントを変更します。

メッセージ 列にはイベントをトリガーしたジョブの構成ページへのリンクが含まれている場合があります。

グリッド上の列のサイズ調整を行う場合、列の見出しを区切る垂直のバーのいずれかをクリック、左側のマウスを押したまま、左右にドラッグします。

永久リンクをクリップボードにコピーする ボタンは選択されたパラメータを含むログビューの現在のURL をクリップボードにコピーします。例、?id=2773968&limit=25。これは同じ情報を後にページに素早くロードすることを希望する場合役に立ちます。例えば、永久URL を他のブラウザアドレスバーに張り付け、または同じログビューを確認するために誰かに送信することができます。

## ログのエクスポート

ログをディスク上のファイルにエクスポートする場合エクスポートをクリックします。ページ内で現在表示されているすべてのレコードはシリアライズされたJSON (JavaScript Object Notation) ファイルとしてエクスポートされます。「エクスポート」をクリックすると、ZIP アーカイブをローカルディレクトリに保存する、または、ブラウザのデフォルトのダウンロードディレクトリに保存するようプロンプトされます。

## ログをフィルターする方法

以下の条件でログをフィルターすることができます:

- **開始日** - この日付以降のイベントが含まれます。
- **開始日** - この日付までのイベントが含まれます。
- **オブジェクトノイズ** - 選択されたノイズでの上行性から発生するイベントのみが含まれます。ジョブまたは重要なレコードなど FlowForce オブジェクトへのノイズを選択することが可能です。
- **最小の重要度** - 重要度が最小に設定されたイベントのみが含まれます。例えば、全ての重要度のイベントを含む場合、最小の重要度として[情報]を選択します。両方の情報メッセージと警告をスキップするために、最小重要度として[エラー]を選択します。

開始日 と終了日 フィルターの両方を設定した場合、その範囲内のレコードが1000 件のレコードまで表示されます。追加のレコードを確認する場合、[更にN 件のレコードを表示] ボタンをクリックします。最近の日付を持つレコードが常に最初に表示されます。

フィルターのいずれかを変更後 表示 ボタンをクリックしてフィルターを適用します。フィルターを保ちつつ表示 ボタンを繰り返しクリックして最近のログをロードします。更にN 件のレコードを表示 ボタンのいずれかをクリックすると現在選択されているフィルターも保存されます。

フィルターのリセット ボタンはすべてのフィルターとログの更新をクリアします。フィルター無しの表示 ボタンをクリックしても同じ効果(ログの更新)が得られます。

## 6.4 インスタンスログの確認

[インスタンスログ]ページはジョブ(またはログインスタンス)の特定のログされているインスタンスに関する詳細を提供します。このページを以下の方法で開くことができます:

- [ログビュー](#) ページ内のレコードのインスタンスID リンクをクリックします。
- ホームページ内の [最近および実行中の](#) セクション内のレコードのインスタンスID リンクをクリックします。
- ジョブ構成ページからログをビューをクリックし [ログビュー](#) ページに移動し、レコードのインスタンスID リンクをクリックします。

### Log for instance 4

|                           |  |
|---------------------------|--|
| Created instance:         | 2020-10-01 17:13:19                        |
| Job Name:                 | <a href="#">/public/Examples/CopyFiles</a> |
| Queue Name:               | <a href="#">/public/Examples/CopyFiles</a> |
| Current or Last Step:     | 2  |
| Postponed Steps Failures: | 0  |
| Current State:            | Finished successfully                      |

| Date                | Message   |
|---------------------|---|
| 2020-10-01 17:13:19 | Starting instance 4.  |
| 2020-10-01 17:13:19 | Starting execution of job <a href="#">/public/Examples/CopyFiles</a> in queue <a href="#">/public/Examples/CopyFiles</a>  |
| 2020-10-01 17:13:19 | Running instance 4 locally.   |
| 2020-10-01 17:13:19 | Execute main job <a href="#">/public/Examples/CopyFiles</a>   |
| 2020-10-01 17:13:19 | Execute function <a href="#">/system/compute</a>  |
| 2020-10-01 17:13:19 | For each item in list   |
| 2020-10-01 17:13:19 | Iteration 1   |
| 2020-10-01 17:13:19 | Iteration 2   |
| 2020-10-01 17:13:19 | Execute function <a href="#">/system/filesystem/copy</a>  |
| 2020-10-01 17:13:19 | Executing FlowForce.copy with parameters: Source: "C:\FlowForceExamples\CopyFiles\source\orders.txt", Destination: "C:\FlowForceExamples\CopyFiles\target", Overwrite: true(), AbortOnFail: true(), Working-directory: "" |
| 2020-10-01 17:13:19 | Step FlowForce.copy completed successfully  |
| 2020-10-01 17:13:19 | Finished job execution: job <a href="#">/public/Examples/CopyFiles</a> in queue <a href="#">/public/Examples/CopyFiles</a>  |

Export
View Exported Log for Instance...

### インスタンスログ

適用可能な場合[インスタンスログ]ページは以下のデータのカテゴリをレポートする場合があります

- ジョブインスタンスの実行中に生成されるメッセージはステップごとグループ分けされています。以下が含まれます:
  - ビルトイン関数またはマッピングの実行からのメッセージとパラメーター。
  - compute と compute-string 関数を実行するステップの結果。

- [成功/失敗を持つ実行]ハンドラーステップまたはジョブの失敗内のエントリに導くエラーメッセージ。
- ステップの実行間に経過した時間に関する情報。
- [for-each] ステップの反復
- ジョブが再試行された回数に関する情報 [エラー時に再試行](#) を参照してください。
- マッピングの実行、または [commandline](#) 関数により生成されたストリーム

エクスポート ボタンは現在のログインスタンスに関連したすべてのデータ.zip アーカイブを作成します。ログされたインスタンスをディスク上の.zip アーカイブにエクスポートした場合、インスタンスのためにエクスポートされたログをビュー ボタンをクリックして確認することができます。

インスタンスのためにエクスポートされたログをビュー ボタンにより以前に保存された.zip アーカイブをページロードすることができるようになります。

.zip アーカイブからロードされたログインスタンスの確認が完了するとエクスポートされたログビューを閉じる ボタンをクリックします。

## 7 FlowForce 式

FlowForce 式は、ジョブが作動する際に FlowForce Server により計算され実行されるステートメントです。FlowForce 式は、ジョブ内の複数のステップをつなげる FlowForce により理解される基本のスクリプト言語です。FlowForce 式には、以下のコンテキストが通常必要とされます:

- ビルトイン関数のラメーター内(つまり、ジョブ構成ページ内の入力フィールドに式を書き込む、または埋め込むことができます)。以下は、例の一部です:
  - 実行ステップにより返された結果のデータ型を変更します。
  - 値の配列を返す結果から特定の値を選択します。
  - 文字列を作成するために複数の値を結合します。
- 条件付きのステートメントを「when」ステップ内で作成します。これにより、ブール値が **true** を評価する式を与えるとステップを実行することが可能になります。
- 「for-each」ステップ内。「for-each」ステップを使用することにより、シーケンスが式により定義されている箇所のアイテムのシーケンスをループすることができます。

このセクションでは、上にリストされたシナリオのために FlowForce 式を構築する際に手助けとなる概念について説明されています。

## 7.1 式の計算

 `/system/compute` 関数を呼び出す実行ステップを作成することにより、ジョブの埋め込み前に FlowForce 式を簡単にテストすることができます。順序を追った例に関しては、以下を参照してください！ ["Hello, World!" ジョブの作成](#)。

 `/system/compute` 関数は、「式」/パラメーターの値を評価し、計算された結果を返します。重要な点は、この関数は定義された戻り値の型を持ちません。実際の型は、計算される式により異なります。例えば、この関数を式 `1+1` にパスすると、関数は、数値 `2` を返します。しかしながら、例えば、この関数を式 `'1+1'` にパスすると、関数は、文字列の値 `1+1` を返します。

この概念をより良く理解するために、 `/system/compute` 関数を呼び出すステップを作成し、式フィールド内に `"1+1"` を入力します。ジョブの戻り値の型を下に示されるように「文字列」として宣言してください。

### 実行ステップ

 関数の実行 `/system/compute`

パラメーター: 条件式: `1+1`

= ステップの結果を割り当て  として TO

新規の実行ステップ
新規の選択ステップ
新規の For-each ステップ
新規エラー/成功処理ステップ

### 実行の結果

戻り値の型を以下と宣言する:  ▼

ジョブを保存しようと試みると、FlowForce は、「型文字列と数値が一致しません」エラーを表示します。計算された式が数値で、ジョブの戻り値の型が文字列の値として宣言されているため、このエラーが生じました。入力に関する問題を解決するために、ジョブの戻り値の型を「数値」に変更するか、または数値を文字列に変換します。下の例は、数値を文字列の値に変換する FlowForce 式関数 `string` を呼び出します。

### 実行ステップ

 関数の実行 `/system/compute`

パラメーター: 条件式: `string(1+1)`

式を計算して、値を文字列として返す場合、代わりに  `/system/compute-string` 関数を使用することができます。この場合、文字列から中かっこを使用して区別される必要があることにご注意してください（以下を参照: [文字列フィールド内に式を埋め込む](#)）。

## 7.2 式言語のルール

FlowForce 式内でエラーを回避するためには、以下のルールを守ってください。

- 許可された **または** 宣言された値のみを使用してください。
- 文字列をそのまま使用する場合は、一重引用符で囲ってください。
- 文字列フィールド内に式を埋め込む場合は、中かこ、つまり { と } 文字で囲ってください。
- 式は入力された箇所のフィールド内で意味を持つ必要があります。

式言語のルールの詳細を確認してみましょう。

### ルール#1: 許可された、または、宣言された値のみを使用します

FlowForce 式内では以下の構文文を使用することが許可されています:

- FlowForce 式関数 (完全なレファレンスに関しては、以下を参照してください: [式関数](#))
- FlowForce 演算子 (以下を参照: [演算子](#))
- 数値
- 文字列の値
- 前に宣言された値

FlowForce 式が許可されているフィールドのテキスト内部に入力すると、リアルタイムで構文のチェックが行われます。構文が正確でない場合、FlowForce は、文字を赤でハイライトします。以下は構文検証エラーのサンプルです:

| 関数の実行 /system/filesystem/copy |  |                                    |                                 |
|-------------------------------|--|------------------------------------|---------------------------------|
| パラメーター:                       | ソース: <input type="text" value="{source}"/>   | <span style="color: red;">!</span> | 文字列 (必須) <a href="#">セットする値</a> |
|                               | ターゲット: <input type="text" value="{target}"/> | <span style="color: red;">!</span> | 文字列 (必須) <a href="#">セットする値</a> |

**ソース** または **ターゲット** がジョブ内で宣言されていないため、FlowForce が式を解釈できないため、エラーが発生します。問題はこれらの値を (例えば、ジョブ入力パラメーターとして) 宣言することにより解決されます:

### ジョブ入力パラメーター

|   |   |                                     |   |                          |
|---|---|-------------------------------------|---|--------------------------|
| + | 名前: <input type="text" value="source"/> | 型: <input type="text" value="文字列"/> | デフォルト: <input style="border: none; color: blue; font-weight: bold; font-size: 1.2em; vertical-align: middle;" type="text" value="+"/> | 説明: <input type="text"/> |
| + | 名前: <input type="text" value="target"/> | 型: <input type="text" value="文字列"/> | デフォルト: <input style="border: none; color: blue; font-weight: bold; font-size: 1.2em; vertical-align: middle;" type="text" value="+"/> | 説明: <input type="text"/> |
| + |   |                                     |   |                          |

### 実行ステップ

|   |                               |  |  |  |
|---|-------------------------------|--|--|--|
| + | 関数の実行 /system/filesystem/copy |  |  |  |
|   | パラメーター:                       | ソース: <input type="text" value="{source}"/>   |  |  |
|   |                               | ターゲット: <input type="text" value="{target}"/> |  |  |

### ルール#2: 文字列を一重引用符で囲みます

文字列をそのまま使用する場合は、一重引用符で囲ってください。それ以外の場合、式は、期待しない結果を出す場合があります。また、検証に失敗する場合があります。次の例を考慮してください:

| 式      | 評価... | 説明              |
|--------|-------|-----------------|
| 1+1    | 2     | 値のデータ型は、数値です。   |
| '1+1'  | 1+1   | 値のデータ型は、文字列です。  |
| 1+1==2 | true  | 値のデータ型は、ブール値です。 |

データの型を他の型に変換する必要がある場合、FlowForce 式関数を使用してください(以下も参照してください! ルール#4)。

### ルール#3: 文字列フィールド内で中かっこを使用します

文字列フィールド内に式を埋め込むには、中かっこで囲んでください。下のサンプルでは、中かっこは ( FlowForce 式関数である式 `instance-id()` ) を文字列の残りから区切ります。

| 関数の実行 /system/mail/send |                                      |
|-------------------------|--------------------------------------|
| パラメーター:                 | 差出人: someone@example.org             |
|                         | 宛先: someone@example.org              |
|                         | 件名: Job{instance-id()}has completed. |

フィールド全体が型「式」の場合、中かっこを使用しないでください。例えば `system/compute` ビルトイン関数の「式」パラメーターでは、関数はこの型を持ちます。(中かっこが使用されていない)下の例はこのフィールドのための正しい値の使用の例です:

| 関数の実行 /system/compute |  |
|-----------------------|--|
| パラメーター:               | 条件式: <code>concat('a','b','c')</code> 条件式T0 (必須) |
| ステップの結果を割り当て          | 名前: <input type="text"/> としてT0                   |

式フィールド内に中かっこを入力することは、構文エラーをトリガーします:

| 関数の実行 /system/compute |  |
|-----------------------|--|
| パラメーター:               | 条件式: <code>concat('a','b','c'</code> <span style="color: red;">!</span> 条件式T0 (必須) |
| ステップの結果を割り当て          | 名前: <input type="text"/> としてT0   |

以下も参照してください! [文字列フィールド内に式を埋め込む](#)。

### ルール#4: 正確なデータ型を使用してください

最後に、FlowForce は、ジョブを保存する際に、データの型のチェックを行います。フィールドにより期待されるデータ型に一致しないフィールドに式が入力されると、エラーが発生します。各フィールドで受け入れられるデータ型は、各フィールドの右端に表示されています。例:

| 関数の実行 /system/filesystem/move |  |
|-------------------------------|--|
| パラメーター:                       | ソース: <input type="text"/> 文字列 (必須) <input type="button" value="セットする値"/> |
|                               | 保存先: <input type="text"/> 文字列 (必須) <input type="button" value="セットする値"/> |

ですから、`1+1` などの式は、明示的に数値として評価されるため、文字列のフィールドでは有効ではありません。その一方、式 `'1+1'` は、有効な文字列フィールドです。次の例を考慮してください!

| 式   | 評価             | 説明   |
|---|----------------|--|
| <code>1/4</code>                            | 0.25 (数値として)   | <p>値のデータ型は、数値です。</p> <p>この式をフィールド内、または、数値を期待するコンテキストで使用してください。それ以外の場合、ジョブの検証は失敗します。</p>  |
| <code>1+1==2</code>                         | true (ブール値として) | <p>値のデータ型は、ブール値です。</p> <p>この式をフィールド内、または、ブール値の値を期待するコンテキストで使用してください。それ以外の場合、ジョブの検証は失敗します。</p>  |
| <code>'apple'</code>                        | apple (文字列として) | <p>値のデータ型は、文字列です。</p> <p>この式をフィールド内、または、文字列の値を期待するコンテキストで使用してください。それ以外の場合、ジョブの検証は失敗します。</p>  |
| <code>concat('1','2','3')</code>            | 123 (文字列として)   | <p>値のデータ型は、文字列です。</p> <p>この式をフィールド内、または、文字列の値を期待するコンテキストで使用してください。それ以外の場合、ジョブの検証は失敗します。</p>  |
| <code>1+'apple'</code>                      | -              | <p>この式は有効ではありません。FlowForce は、ジョブを保存しようとするとき、エラーを返します。2つの異なるデータ型(文字列と数値)が比較されているため、検証を行うことはできません。</p>   |
| <code>{content<br/>(stdout(result))}</code> | [...] (文字列として) | <p>この式は、2つのネストされた式関数を使用します:</p> <ul style="list-style-type: none"> <li>関数 <code>stdout</code> は、シェルコマンドの標準出力をストリームとして取得します。</li> <li>関数 <code>content</code> は、ストリームの値を文字列に変換します。</li> </ul> <p>式は、正確ですが、ジョブは、以下の条件が満たされている場合のみ検証を成功することができます:</p> <ul style="list-style-type: none"> <li>値「結果」は既に宣言されています。</li> <li>値「結果」シェルコマンドの標準出力を含んでいます。</li> <li>式は文字列フィールドに埋め込まれています。</li> </ul> <p>以下も参照してください: <a href="#">式関数の呼び出し</a>。</p> |

## 7.3 文字列フィールド内に式を埋め込む

文字列フィールド内でFlowForce 式を使用するには、式を中かっこで囲んでください。つまり、“{” and “}” 文字で囲んでください。文字列フィールドの式の部分は、通常、文字列の他の部分特別しやうい薄い紫色の背景色で表されています。例:

|         |                   |                                  |
|---------|-------------------|----------------------------------|
| 関数の実行   | /system/mail/send |                                  |
| パラメーター: | 差出人:              | someone@example.org              |
|         | 宛先:               | someone@example.org              |
|         | 件名:               | Job{instance-id()}has completed. |

文字列フィールド内では、式のみが中かっこで囲まれ、FlowForce による式として扱われます。FlowForce に“{” and “}” 文字をそのまま解釈するよう指示する場合は、一重引用符の代わりに中かっこを使用してください。以下のケースを考慮してください

| 次の値を持つ文字列フィールド...        | 評価                     | 説明  |
|--------------------------|------------------------|---|
| echo Hello, World!       | echo Hello, World!     | (文字列は、埋め込まれた式を含んでいません)文字列は中かっこを使用しません ですから、そのまま評価されます。                                      |
| echo {Hello, World!}     | -                      | 文字列を評価することができません。埋め込まれた式が構文的に正確ではないため、FlowForce は、構文エラーを表示します。                              |
| echo {"Hello, World!"}   | echo Hello, World!     | 文字列は、構文的に正しい埋め込まれた式を含みます。しかしながら、式は文字列フィールドの内部にあり、評価結果は、式が使用されていない場合と同じになります(最初の例を参照してください)。 |
| echo {{"Hello, World!"}} | echo {'Hello, World!'} | エスケープ文字 {{ and }} が使用されため文字列は、式を含みません。  |

## 7.4 式関数の呼び出し

FlowForce 式言語は、(主に、実行するたびに返される値を扱う)基本的なオペレーションを実行するための一連の関数を含んでいます。FlowForce 式が有効なコンテキストからこれらの関数を呼び出すことができます (例えば、関数のパラメータを表すテキストボックス内に入力することなど)。

FlowForce 式関数とFlowForce ビルトイン関数を区別する必要があります。ビルトイン関数は、FlowForce 実行ステップから呼び出されますが、(つまり、ステップとして実行されますが)、式関数はFlowForce 式から呼び出されます。

式関数を呼び出す典型的なシナリオは、2つの実行ステップから構成されている、下で説明されているジョブが挙げられるでしょう。

最初のステップは、シェルコマンドを実行します (具体的には "Hello, World!" を出力します)。このステップにより返されるデータ型は、「結果として」であり、戻り値は **var1** として宣言されています。

2番目の実行ステップは、/system/compute-string ビルトイン関数を呼び出します。この関数は、**var1** を文字列に変換するために呼び出されます。式自身は、文(中かっこで示されている)字列フィールドに埋め込まれており、2つのネストされた式関数を呼び出します。

- 関数 **stdout** は、シェルコマンドの標準出力をストリームとして返します。
- 関数 **content** は、ストリームの値を文字列に変換します。

### 実行ステップ

**+**

関数の実行 /system/shell/commandline

パラメーター: コマンド:

エラー時には中断:

作業用ディレクトリ:

= ステップの結果を割り当て  以下として result

**+**

関数の実行 /system/compute-string

パラメーター: 条件式:

= ステップの結果を割り当て  以下として string

データ型の変換が完了し、文字列の値 **var2** をジョブの処理ロジックに従い、更に例えば、電子メールへの送信などに)使用することができます。

使用することのできる式関数に関しては、以下を参照してください! [式関数](#)。

## 7.5 FlowForce データ型

次のデータ型を使用して FlowForce を操作します。

### 文字列

文字列の値を表します。例: 'Hello, World!'。

### 数値

数値を表します。例: -1, 0, 56, 0.45565。

### ブール値

**true** または **false** 値を表します。

### 結果

実行ステップにより生成された結果を表す抽象的な型です。

実行ステップが、MapForce マッピング、StyleVision 変換ファイル、シェル関数、など、多種の実行可能ファイル処理します。`result` データ型は、ですから、そのようなファイルの出力を表します。

実行ステップが、MapForce マッピングを実行する場合、出力は以下の可能性があります: XML、XBRL、テキスト、JSON、および MapForce により生成される他のファイル型。

実行ステップが、StyleVision 変換を実行する場合、出力は以下の可能性があります: PDF、Word、HTML ファイル、および StyleVision により生成される他の出力の型。

結果の値にアクセスするには、名前を与え、(例えば、「output」)、`{result}` 式関数にアクセスします。これにより、ストリーム式関数を使用して値をストリームに変換し、更にこの値を処理することができます (以下も参照してください! [式関数の呼び出し](#))。

実行ステップがシェルコマンドを実行すると、出力に応じ特定のステップ結果式関数を呼び出します。例えば、ストリームとしての標準出力を返すためには、式 `{stdout(output)}` を使用します。標準エラーをストリームで返す場合、式 `{stderr(output)}` を使用します。詳細に関しては、以下を参照してください! [ステップ結果関数](#)。

### 結果

MapForce マッピング または StyleVision 変換が複数のオブジェクトを返す場合があります。このようなステップにより生成される結果は、データ型として `results` を持ちます。

このような出力を扱うには、ストリームの配列を返す `{results(output)}` 式関数を使用します。`nth` 関数を使用して、配列から特定のストリームを選んでください。

例えば、下で説明されるジョブは FlowForce にデPLOYされた StyleVision 変換ファイルから作成されました。このジョブは、入力パラメータとして、XML ファイルを取り、多種のフォーマットの複数の出力を返します。

実行ステップ

関数の実行 /public/AutoCalc.transformation

パラメーター:

|                           |                                |                   |          |   |
|---------------------------|--------------------------------|-------------------|----------|---|
| InputXml:                 | altova://packagedfile/Data.xml | xs:string (必須)    | セットする値 ▶ | 🔗 |
| OutHtml:                  | AutoCalc.html                  | xs:string (オプション) | セットする値 ▶ | 🔗 |
| GenerateHtmlOutputAsMime: | <input type="checkbox"/>       | ブール値 (オプション)      | セットする値 ▶ | 🔗 |
| OutRtf:                   | AutoCalc.rtf                   | xs:string (オプション) | セットする値 ▶ | 🔗 |
| OutFo:                    | AutoCalc.fo                    | xs:string (オプション) | セットする値 ▶ | 🔗 |
| OutPdf:                   | AutoCalc.pdf                   | xs:string (オプション) | セットする値 ▶ | 🔗 |
| OutDocx:                  | AutoCalc.docx                  | xs:string (オプション) | セットする値 ▶ | 🔗 |
| Working-directory:        | c:\temp                        | xs:string (必須)    | セットする値 ▶ | 🔗 |

= ステップの結果を割り当て step1 以下としてReturnTypeDef, ReturnTypeDef, ReturnTypeDef, ReturnTypeDef, ReturnTypeDef, ReturnTypeDef

関数の実行 /system/compute

パラメーター: 条件式: as-file(nth(results(step 1), 0)) 条件式T0 (必須) セットする値 ▶ 🔗

= ステップの結果を割り当て step2 以下としてT0

関数の実行 /system/filesystem/copy

パラメーター:

|            |                                     |              |          |   |
|------------|-------------------------------------|--------------|----------|---|
| ソース:       | {step2}                             | 文字列 (必須)     | セットする値 ▶ | 🔗 |
| ターゲット:     | c:\archive\AutoCalc.html            | 文字列 (必須)     | セットする値 ▶ | 🔗 |
| 上書き:       | <input checked="" type="checkbox"/> | ブール値 (オプション) | セットする値 ▶ | 🔗 |
| エラー時には中断:  | <input type="checkbox"/>            |              |          |   |
| 作業用ディレクトリ: | C:\temp                             | 文字列 (オプション)  | セットする値 ▶ | 🔗 |

= ステップの結果を割り当て 名前 以下としてブール値

このようにおつの実行ステップが存在します。

1. 最初の実行ステップは、実際のデータ変換をおこないます。
2. 2番目の実行ステップは、この変換の結果を取得するために、FlowForce の `compute` 関数を呼び出します。具体的には式 `as-file(nth(results(output), 0))` は、ストリームの列挙内の最初のアイテムをファイルとして取得します。列挙にはゼロベースのインデックスが存在するため、式は「1」ではなく、「0」を使用します。
3. 3番目のステップは、`c:\archive` ディレクトリにHTML ファイルをコピーします。

## アイテム

リストをアセンブルまたは分解する式を作成する必要がある場合があります（以下を参照: [リスト関数](#)）。リストはジェネリックな型 `item` のオブジェクトのリストから構成されています。アイテムは、抽象的なデータ型を持ちます。リストを作成するオブジェクトの型に従い、（文字列、数値、またはストリームであることができる）`item` のデータ型を決定することができます。リストは同じデータ型のアイテムのみを含むことができ、ご注意ください。

下のイメージは、リストが数値により構成されているため、「item」が数値の型であるループを表しています。

For each item シーケンスで list(1,2,3)

+

= ステップの結果を割り当て 名前

リストを活用するステップ/各ステップのサンプルに関しては、[ファイルをコピーする](#)を参照してください。

## 7.6 演算子

FlowForce 式を構築するときは、下記リストされる演算子を使用することができます。ビルトイン関数 `system/compute` を使用して全ての式をテストできることを忘れないでください。

| 演算子                | 説明   | 例   |
|--------------------|--|---|
| <code>==</code>    | a が b と等しいかチェックします (数値的に番号が同じです。文字列のためのコードポイント)。   | <code>2 + 3 == 5</code> は <b>true</b> を算出します。<br><br><code>2 + 3 == 4</code> は <b>false</b> を算出します。 |
| <code>!=</code>    | a が b と等しくないかチェックします。次のつの式は同じで <ul style="list-style-type: none"> <li><code>a != b</code></li> <li><code>not (a == b)</code></li> <li><code>a &lt;&gt; b</code></li> </ul> | <code>2 + 2 != 5</code> は <b>true</b> を算出します。<br><code>3 + 2 != 5</code> は <b>false</b> を算出します。     |
| <code>&lt;</code>  | a が b より小さいかチェックします (数値的に番号が小さいです。文字列に関しては、以下を参照してください)。   | <code>4 &lt; 5</code> は <b>true</b> を算出します。   |
| <code>&lt;=</code> | a が b より小か等しいかチェックします。   | <code>5 &lt;= 5</code> は <b>true</b> を算出します。  |
| <code>&gt;</code>  | a が b より大きいかチェックします  | <code>5 &gt; 1</code> は <b>true</b> を算出します。   |
| <code>&gt;=</code> | a が b より大か等しいかチェックします。   | <code>5 &gt;= 5</code> は <b>true</b> を算出します。  |
| <code>+</code>     | 加算   | <code>1 + 1</code> は <b>2</b> を算出します。   |
| <code>-</code>     | 減算   | <code>2 - 1</code> は <b>1</b> を算出します。   |
| <code>*</code>     | 乗算   | <code>3 * 2</code> は <b>6</b> を算出します。   |
| <code>/</code>     | 除算   | <code>6 / 3</code> は <b>2</b> を算出します。   |

文字列の比較は以下のように行われます:

- 2つの文字列の共通のプレフィックスは無視されます (コードポイントで評価されます)
- 残り2つの文字列が空であれば内場合、最初のコードポイントが数値的に比較されます。
- 空の文字列は、空であれば内文字列は先小さと考えられます。

かっこを使用して、FlowForce 1 かっこの中の式を最初に計算するように命令します。例:

`2 + 3 * 4` は **14** を算出します。

`(2 + 3) * 4` は **20** を算出します。

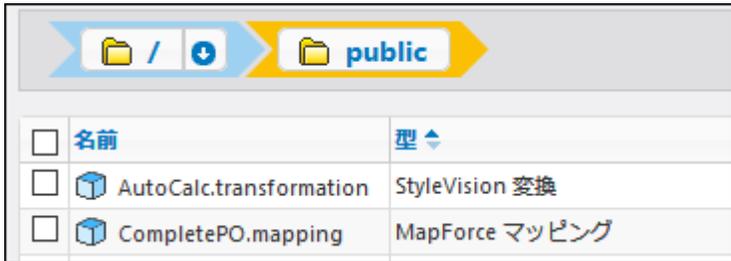
## 8 Altova 製品との統合

**動作の説明** 内では、Altova 製品間での共同作業の概要が説明されていますが、基本的には、Altova MapForce により作成されたマッピングファイルと Altova StyleVision により作成された変換ファイルは、次のサーバー製品を使用することにより自動的に作成することができます: MapForce Server (または MapForce Server Advanced Edition) と StyleVision Server。MapForce Server と StyleVision Server は、サーバー環境 (Windows、macOS、Linux) 内でマッピングと変換をマドラインで、または API からの呼び出し、実行することを可能にしました。これらの製品は、FlowForce とは同時に実行されません。自動化とは API を呼び出す、または、MapForce Server または StyleVision のコマンドラインを呼び出すプログラムを開発、または、スクリプトを作成することを含みます。更に FlowForce Server 管理下で動作している場合、RaptorXML Server 内で使用可能な関数は FlowForce Server ジョブから呼び出すことができます。

MapForce Server と StyleVision Server はコマンドライン、または、API 呼出から複数のプラットフォーム (Windows、macOS、Linux) でマッピングと変換を作動することができます。FlowForce と共にこれらの製品が作動していない場合、自動化は API を呼び出すスクリプトの作成、または、MapForce Server または StyleVision Server のコマンドラインの呼び出しを含みます。

MapForce Server と StyleVision Server が FlowForce Server 管理下で実行される場合、自動化を次のレベルで実行することができます。すなわち、マッピングと変換を直接に FlowForce Server にデプロイすることができ、ジョブを実行できることを意味します (次を参照してください [しくみ](#))。このように、マッピングまたは変換は以下を含む FlowForce Server ジョブの全ての利点を活用することができます: スケジュールされた、または、オンデマンドのジョブの実行、Web サービスとしてのジョブの実行、FlowForce 式を使用した構成、エラーの処理、条件付けされた処理、電子メールによる通知など。

FlowForce Server にデプロイされると、デプロイしたマッピング、または、変換がコンテナー内に表示されます。以下に示されるとおり、マッピングには **.mapping** 拡張子、変換には **.transformation** 拡張子が存在します。



| <input type="checkbox"/> | 名前                      | 型              |
|--------------------------|-------------------------|----------------|
| <input type="checkbox"/> | AutoCalc.transformation | StyleVision 変換 |
| <input type="checkbox"/> | CompletePO.mapping      | MapForce マッピング |

FlowForce の観点からは、このようなオブジェクトは関数であり、このため、新規のジョブに変更することができます。これらは、既存のジョブから呼び出すことができ、パラメータとして (通常はファイルである) 多種の入力を受け入れることができます。FlowForce Server はこのようなマッピング、または、変換を自身で実行しません。MapForce Server または StyleVision Server (または、場合により両方) は、実際の実行を行うために呼び出されます。

RaptorXML 関数を RaptorXML コンテナー内で使用することができます。 [RaptorXML Server との統合](#) も参照してください。

次のセクションでは、サーバー実行のためにどのようにマッピングと変換を準備するか、これらをジョブに、FlowForce Server 内で結果を処理する方法について説明されています。

## 8.1 サーバー実行のためにファイルを準備する

MapForce を使用してデザインされ、レビューされるマッピングは、現在使用中のマシンと(データベースなどの)オペレーティングシステムの外部にあるリソースを参照することができます。更に、MapForce 内では、全てのマッピングのデフォルトで Windows スタイルの規則に従います。また、MapForce Server が作動するサーバーでは、マッピングがデザインされたマシンと同じデータベースをサポートしない場合があります。このため、サーバー環境でマッピングを実行するには、また、ターゲットマシンがソースマシンと異なる場合は特に、若干の準備が必要とされます。

**メモ** ソースマシンという用語は、MapForce がインストールされているマシンを指し、ターゲットマシンは、MapForce Server、または、FlowForce Server がインストールされているマシンを指します。最もシンプルなシナリオは、ソースマシンとターゲットマシンが同じコンピューターの場合を指します。最も複雑なシナリオは、MapForce が Windows マシンで作動し、MapForce Server、または、FlowForce Server が Linux、または、macOS マシンで作動する場合を指します。

FlowForce Server にデプロイする前に MapForce 内でマッピングの検証を常に行うことが奨励されます。または、MapForce Server 実行ファイルコンパイルすることが奨励されます。

MapForce Server がスタンドアロンで(FlowForce Server 無しで)作動する場合、必要とされるライセンスは、以下のとおりです:

- ソースマシン上で、MapForce Enterprise、または、Professional Edition がマッピングをデザインし、サーバー実行ファイル(.mfx)にコンパイルするために必要とされます。。
- ターゲットマシン上で、MapForce Server、または、MapForce Server Advanced Edition がマッピングを実行するために必要とされます。

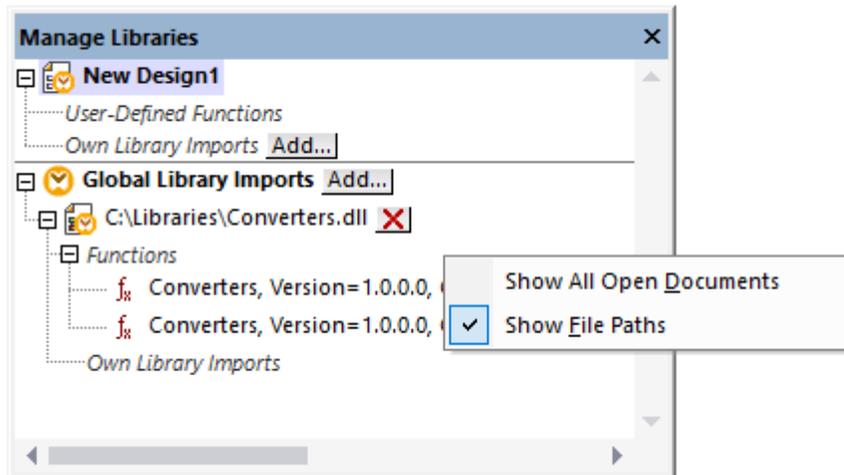
MapForce Server が FlowForce Server 管理下で作動する場合、次の条件が適用されます:

- ソースマシン上で、MapForce Enterprise、または、Professional Edition がマッピングをデザインして、ターゲットマシンにデプロイするために必要です。。
- MapForce Server と FlowForce Server がターゲットマシンにインストールされている必要があります。MapForce Server の役割は、マッピングの実行であり、FlowForce の役割はマッピングを、スケジュールまたは、オンデマンドの実行、としての実行、エラーの処理、条件付きの処理、電子メールの通知などの機能を活用することのできるジョブとして使用できるようにすることです。
- FlowForce Server は構成されているネットワークアドレスとポートで作動している必要があります。すなわち、「FlowForce Web Server」サービスが開始され、構成されており、HTTP (または、構成されている場合は HTTPS) クライアントからの接続をファイアウォールでブロックされることがよく受け入れる必要があります。「FlowForce Server」サービスも開始され、指定されたアドレスとポートで作動している必要があります。
- コンテナーの1つへのパーミッションが与えられている FlowForce Server ユーザーアカウントを所有している必要があります(デフォルトでは /public コンテナーは認証済みのユーザーであればアクセスすることができます)。

### 全般的な注意点

- スタンドアロン MapForce Server で作動するターゲットマシン上のマッピングを作動するには、マッピングにより参照される全ての入力ファイルをターゲットマシンにコピーする必要があります。MapForce Server が FlowForce Server 管理化で作動する場合、ファイルを手動でコピーする必要はありません。この場合、インスタンスとスキーマファイルはターゲットマシンにデプロイされたパッケージ内に含まれます。。
- マッピングが特定のデータベースドライバを必要とするデータベースコネクタが含まれる場合、このようなドライバはターゲットマシンにもインストールされている必要があります。例えば、マッピングが Microsoft Access データベースからデータを読み取る場合、Microsoft Access、または、Microsoft Access Runtime がターゲットマシンにもインストールされている必要があります(<https://www.microsoft.com/en-us/download/details.aspx?id=50040>)。
- マッピングを非 Windows プラットフォームにデプロイする場合、ADO、ADO.NET と ODBC データベース接続は自動的に JDBC に変更されます。ネイティブ SQLite とネイティブ PostgreSQL 接続は保存され、追加の構成を必要としません。「データベース接続」も参照してください。
- (例えば、.dll、または、.class ファイルなどの)カスタム関数呼び出しをマッピングが含む場合、このような依存は、ランタイム前に既知ではないため、マッピングと共にデプロイされません。この場合、これを手動でターゲットマシンにコピーすることができます。サー

バー上の.dll または.class ファイルのパスはMapForce 内の「ライブラリの管理」ウィンドウと同じである必要があります。例



- マッピングの一部は、ワイルドカードパスを使用して複数の入力ファイルを読み取ります。この場合、入力ファイル名はランタイム前に既知ではないため、デプロイされません。マッピングの実行に成功するには、入力ファイルがターゲットマシン上に存在する必要があります。
- マッピングの出力パスにディレクトリが含まれている場合、これらのディレクトリはターゲットマシン上に存在している必要があります。それ以外の場合、マッピングを実行するとエラーが生成されます。この振る舞いは、「一時的なファイルに出力を生成する」オプションが有効化されている場合既存ではないディレクトリが自動的に生成されるMapForce とは異なります。
- マッピングがクライアント証明書を使用するHTTPS 認証を必要とするWeb サービスを呼び出す場合、証明書をターゲットマシンに手動で移動する必要があります。次も参照してください:「デジタル署名の管理」。
- Microsoft Access とSQLite などのファイルをベースにしたデータベースにマッピングが接続される場合、データベースファイルは手動でターゲットマシンに移動、または参照することのできる共有されるディレクトリに保存される必要があります。次を参照:「ファイルをベースにしたデータベース」。

### パスをポータブルにする方法

サーバー上でマッピングを実行する場合、マッピングが適用することのできる変換パスを使用していること、および、サポートされるデータベース接続を使用していることを確認してください。

パスを非 Windows オペレーティングシステムへ移動可能にするためには、MapForce 内でマッピングをデザインする際に絶対パスの代わりに相対パスを使用してください。

- 希望するマッピングデザインファイル(.mfd) をWindows 上のMapForce で開きます。
- 「ファイル」メニューでマッピングの設定を選択し、チェックされている場合「生成されたコード内でパスを絶対パスにする」チェックボックスをクリアします。
- 各マッピングコンポーネントの右側にプロパティダイアログボックスを開き、(例えば、コンポーネントのタイトルバーをダブルクリックして) 全てのファイルパスを絶対パスから相対パスに変更します。「MFD ファイルに対して全てのパスを相対的に保存する」チェックボックスを選択します。便宜上、マッピングと同様全ての入力ファイルとスキーマを同じファイルにコピーし、ファイル名で参照することができます。

マッピングのデザイン時に相対および絶対パスを処理するにはMapForce ドキュメントを参照してください。

更にMapForce Server とFlowForce Server は、すべての相対パスが解決される、いわゆる「作業ディレクトリ」をサポートします。作業ディレクトリは以下のとおりマッピングのランタイムに指定することができます:

- FlowForce Server では、ジョブの作業ディレクトリパラメータを編集します。
- MapForce Server API では COM と.NET API のWorkingDirectory プロパティにより、または Java API の setWorkingDirectory メソッドにより行うことができます。

- MapForce Server コマンドライン では、作業ディレクトリはコマンドシェルの現在のディレクトリです。

## データベース接続

ADO、ADO.NET、および ODBC 接続は Linux と macOS マシンではサポートされていないことにご注意下さい。このため、ターゲットマシンが Linux または macOS の場合、このような接続は、マッピングが FlowForce にデプロイされると、または、MapForce Server 実行ファイルにマッピングをエンパイルすると、JDBC に変換されます。この場合、マッピングをデプロイ、または、サーバー実行ファイルにマッピングをエンパイルする前に以下のオプションを使用することができます。

- MapForce 内では、データベースへの JDBC 接続を作成します
- MapForce 内では、データベースコンポーネントの「JDBC 固有の設定」セクション内の JDBC データベース接続の詳細を入力してください。(データベースコンポーネント設定を参照してください)。

マッピングが PostgreSQL または SQLite データベースへのネイティブな接続を使用する場合、ネイティブな接続が保存され、JDBC 変換は発生しません。マッピングが Microsoft Access と SQLite などのファイルベースのデータベースに接続される場合、追加構成が必要とされます。下記の「ファイルベースのデータベース」を参照してください。

JDBC 接続を使用してマッピングを起動することは、Java Runtime Environment または Java Development Kit がサーバーマシンにインストールされていることを必要とします。Oracle JDK または Oracle OpenJDK などのオープンソースビルドであることができます。

- `JAVA_HOME` 環境変数は JDK インストールディレクトリを指している必要があります。
- Windows 上では、Windows レジストリ内で検出された Java Virtual Machine ノードは `JAVA_HOME` 変数より高い優先順位を有します。
- JDK プラットフォーム (64ビット、32ビット) は MapForce Server を同じである必要があります。それ以外の場合は、理由があるエラーが挙げられる可能性があります。「JVM へのアクセスが不可能です。」

## Linux または macOS 上での JDBC 接続のセットアップ:

1. データベースベンダーにより提供される JDBC ドライバーをダウンロードして、オペレーティングシステムにインストールします。オペレーティングシステムが 32ビットで動作する場合は 32ビット版バージョンを、オペレーティングシステムが 64ビットで動作する場合は 64ビット版バージョンを選択してください。
2. 環境変数を JDBC ドライバーがインストールされている場所に設定してください。通常、`CLASSPATH` 変数と、必要があれば、その他複数の変数を設定します。どの特定の環境変数が構成されるかを検出するには、JDBC ドライバーに付随するドキュメントを確認してください。

**メモ** macOS では、システムは JDBC ライブラリが `Library/Java/Extensions` ディレクトリにあることを必要とします。ですから、JDBC ドライバーをこの場所でアンパイルすること奨励されます。それ以外の場合、システムを構成し、JDBC ライブラリが JDBC ドライバーをインストールしたパスを検索するようになる必要があります。

## macOS 上での Oracle Instant Client 接続

macOS 上で、Oracle Database Instant Client を介して Oracle データベースに接続する場合これらの命令を適用することができます。必要条件:

- Java 8.0 または、以降がインストールされている必要があります。Mac マシンが Java 8 以前の Java バージョンで動作している場合、「JDBC Thin for All Platforms」ライブラリを介して接続し、以下の命令を無視することも可能です。
- Oracle Instant Client がインストールされている必要があります。Oracle Instant Client を Oracle の公式ダウンロードページからダウンロードすることができます。Oracle ダウンロードページでは複数の Instant Client パッケージがあることにご注意ください。Oracle Call Interface (OCI) サポートのあるパッケージが選択されていることを確認してください。(例えば Instant Client Basic)。また、オペレーティングシステムが 32ビットで動作する場合は 32ビット版バージョンを、オペレーティングシステムが 64ビットで動作する場合は 64ビット版バージョンを選択してください。

ダウンロードした後、Oracle Instant Client をアンインストール、インストールと共にインストールされているプロパティリストファイル(.plist) を編集すると、以下の環境変数が対応するドライブをポイントします。例えば

| 変数                | サンプル値   |
|-------------------|---|
| CLASSPATH         | /opt/oracle/instantclient_11_2/ojdbc6.jar:/opt/oracle/instantclient_11_2/ojdbc5.jar |
| TNS_ADMIN         | /opt/oracle/NETWORK_ADMIN   |
| ORACLE_HOME       | /opt/oracle/instantclient_11_2  |
| DYLD_LIBRARY_PATH | /opt/oracle/instantclient_11_2  |
| パス                | \$PATH:/opt/oracle/instantclient_11_2   |

**メモ** 使用中のオペレーティングシステムで Oracle Instant Client ファイルがインストールされている場所で上記のサンプル値を編集してください。

### ファイルをベースとしたデータベース

Microsoft Access と SQLite などのファイルをベースとしたデータベースは FlowForce Server にデプロイされるパッケージ内に含まれません。または、インストールされた MapForce Server 実行ファイルにも含まれません。このため、ソースとターゲットマシンが同じであれば、次のステップに従ってください。

1. MapForce 内では、マッピングを右クリックして、「生成されたコード内でパスを絶対パスにする」のチェックボックスのチェックを解除します。
2. マッピング上のデータベースコネクタを右クリックし、相対パスを使用してデータベースファイルへの接続を追加します。。マッピングデザイン(.mfd ファイル) をデータベースファイルと同じディレクトリに保存し、ファイル名別(相対パスを使用して)にマッピングから参照することにより、パスに関連した問題を回避することができます。
3. データベースファイルをターゲットマシン内のディレクトリにコピーします(「作業ディレクトリ」と呼ぶことができます)。以下に表示されるようにマッピングを実行する際必要とされるため、このディレクトリを忘れないでください。

このようなマッピングをサーバー上で実行するには以下を行います:

- FlowForce Server コントロール下で作動する MapForce Server によりマッピングが実行される場合、FlowForce Server ショップ前に作成された作業ディレクトリをポイントするよう構成します。データベースファイルは作業ディレクトリ内に存在する必要があります。
- コマンドラインでスタンドアロン MapForce Server によりマッピングが実行される場合、MapForce Server 上で run コマンドを呼び出す前に、現在のディレクトリを作業ディレクトリに変更します(例えば cd path\to\working\directory)。
- MapForce Server API によりマッピングが実行される場合、作業ディレクトリをプログラミング的にマッピングを実行する前に設定します。これを効果的に行うために、COM と .NET API 内の MapForce Server オブジェクトの `forAPIWorkingDirectory` を使用することができます。Java API 内では、メソッド `setWorkingDirectory` を使用することができます。

ローカルネットワーク上の Windows マシン上にソースとターゲットマシンが存在する場合、共有されるディレクトリからデータベースファイルを読み取るためにマッピングを以下のように構成することができます:

1. ソースとターゲットマシンによりアクセスすることのできる共有されるディレクトリにデータベースファイルを保管します。
2. マッピング上のデータベースコネクタを右クリックし、絶対パスをしようしてデータベースファイルに接続を追加します。

### グローバルリソース

マッピングが直接のパスまたはデータベース接続の代わりにグローバルリソースへの参照を含む場合、サーバー側でもグローバルリソースを使用

することができます。MapForce Server 実行可能ファイル(.mfx) にマッピングをコンパイルする場合、グローバルリソースへの参照は変更されないため、マッピングランタイムにサーバー側で提供することができます。FlowForce Server にマッピングをデプロイする場合、リソースをサーバー上で使用するかを任意で選択することができます。

マッピング(またはFlowForce Server の場合マッピング関数)の実行に成功するために、グローバルリソースとして提供する実際のファイルフォルダー、またはデータベース接続の詳細はサーバー環境で互換性を持つ必要があります。例えば、マッピングがLinux サーバーで動作する場合、ノスのためにLinux 変換を使用する必要があります。同様に、データベース接続として定義されているグローバルリソースはサーバーマシンで利用が可能である必要があります。

詳細に関しては[リソース](#)を参照してください。

## XBRL タクノミ パッケージ

マッピングがFlowForce Server へのXBRL タクノミ パッケージを参照する場合MapForce はマッピングから全ての外部レファレンスを集め、現在の構成と現在インストールされているタクノミ パッケージを使用して解決します。タクノミ パッケージを指す解決されず外部レファレンスが存在する場合、タクノミ パッケージはマッピングと共にデプロイされます。FlowForce Server は、そのパッケージを、デプロイ時と同様、マッピングを実行するために使用します。FlowForce Server により使用されるタクノミ パッケージを更新するには、MapForce 内で変更し、マッピングをもう一度デプロイします。

MapForce Server のルートカタログは、ターゲットマシン内でタクノミの解決方法に影響を与えます。ルートカタログは、MapForce Server インストールディレクトリに対して相対的に以下のパスで見つけることができます: **etc/RootCatalog.xml**。

MapForce Server のルートカタログがこのようなカタログを含んでいない場合、または、同 URL にプレフィックスのために定義されているパッケージを含まない場合、マッピングと共にデプロイされるタクノミ パッケージが使用されます。MapForce Server のルートカタログがデプロイされればタクノミは優先順位が高いです。

MapForce Server がスタンドアロンで(FlowForce Server 無しで)動作する場合、次に用いてマッピングにより使用されるルートカタログを指定することができます:

- コマンドラインでは、オプション `-catalog` を `run` コマンドに追加することが可能です。
- MapForce Server API で、メソッド `SetOption` を呼び出し、文字列「`catalog`」を最初の引数として、ルートカタログを 2 番目の引数として与えます。

マッピングがケーブルリンクベースを使用するXBRL コンポーネントである場合、タクノミ パッケージ、または、タクノミ パッケージ構成ファイルがランタイムでマッピングに提供される必要があります:

- MapForce Server コマンドラインにて、オプション `--taxonomy-package`、または、`--taxonomy-packages-config-file` を `run` コマンドに追加します。
- MapForce Server API 内で、メソッド `SetOption` を呼び出します。最初の引数は、「`taxonomy-package`」、または、「`taxonomy-packages-config-file`」である必要があります。2 番目の引数は、タクノミ パッケージ(、または、タクノミ パッケージ構成) ファイルへの実際のパスである必要があります。

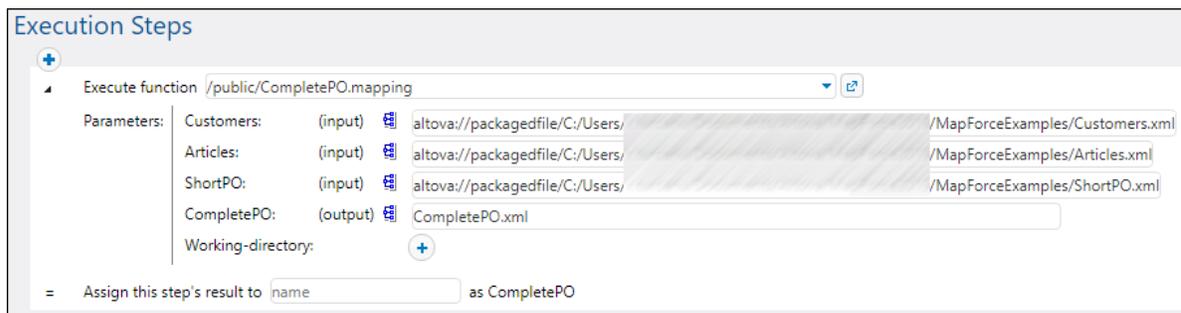
## 8.2 FlowForce Server にマッピングをデプロイする

FlowForce Server へのマッピングのデプロイは MapForce が特定のマッピングにより使用されるリソースを整理し HTTP (または構成されている場合 HTTPS) を介して FlowForce Server が作動するマシンに送ることを意味します。メモ「ソースマシン」という用語は MapForce がインストールされているコンピュータを指し、「ターゲットマシン」という用語は FlowForce Server がインストールされているコンピュータを指します。マッピングがデプロイされると、完全機能を搭載した FlowForce Server ジョブを作成することができ、(例えば ジョブのためのカスタムトリガーを定義する、ジョブをサービスとして公開するなど)ジョブ固有の機能全てを活用することができます。

**メモ** 「ソースマシン」という用語は MapForce がインストールされているコンピュータを指しており、「ターゲットマシン」は FlowForce Server がインストールされているマシンを指します。最もシンプルなシナリオは、ソースマシンとターゲットマシンが同じコンピュータの場合を指します。最も複雑なシナリオは MapForce が Windows マシンで作動し、FlowForce Server が Linux、または macOS マシンで作動する場合を指します。

FlowForce にデプロイされるパッケージは以下の通りです:

- マッピング自身コンポーネントは入力引数となり、ターゲットコンポーネントはこの関数の出力引数となります。全てのソースコンポーネントは入力引数となり、全てのターゲットコンポーネントはこの関数の出力関数となります。



- マッピングにより使用される入力インスタンスファイル(XML、CSV、テキスト)。

### 必要条件

[サーバー実行のためにマッピングを準備する](#)を参照してください。

### FlowForce Server にマッピングをデプロイする

- MapForce を作動し(Built-in <sup>BUILT IN</sup> ツールボタンをクリック、または出力 | Built-in 実行エンジン メニューコマンドを選択して 変換言語が BUILT-IN に設定されていることを確認してください)。
- 「ファイル」メニューから「FlowForce Server にデプロイ」を選択します。マッピングのデプロイダイアログボックスが開かれます。

3. (下で説明される通り)デプロイ設定を入力し [OK] を押します。「新しいジョブ作成のためにウェブブラウザを開く」チェックボックスが選択されている場合、FlowForce Server 管理インターフェイスが開かれ、すぐに FlowForce Server ジョブの作成を開始することができます。

以下のテーブルは、マッピングのデプロイダイアログボックスで使用することのできるマッピングデプロイ設定をリストしています。

| 設定               | 説明  |
|------------------|---|
| サーバー、ポート、SSL の使用 | <p>FlowForce Server のサーバーホスト名 (または I.P. アドレス) とポートを入力します。これは FlowForce Server がデフォルトのポートと同じマシンで実行されている場合、<b>localhost</b> と <b>8082</b> である可能性があります。疑問がある場合は、FlowForce Server Web 管理インターフェイスにログインして、Web ブラウザーのアドレスバーに表示される I.P. アドレスとポートを確認してください。</p> <p>接続エラーが生じた場合、FlowForce Server が実行されているマシンが入力接続を許可するように構成されていることを確認してください。</p> <p>SSL 暗号化済み接続を介してマッピングをデプロイする場合 SSL の使用 チェックボックスを選択しますこれは、FlowForce Server が既に SSL 接続を受け入れるように構成されているもの想定しています。詳細に関しては、FlowForce Server ドキュメント (<a href="https://www.altova.com/ja/documentation">https://www.altova.com/ja/documentation</a>) を参照してください。</p> |

| 設定                   | 説明  |
|----------------------|---|
| ユーザーとパスワード           | ユーザー名とパスワードは、ログインドロップダウンリストの値（次のオプション参照）により入力されます。ログインドロップダウンリストが<Default> または Directly に設定されている場合、FlowForce Server ユーザー名とパスワードを入力します。それ以外の場合ドメインユーザー名とパスワードを入力し、ドメイン名をログインドロップダウンリストから選択します。   |
| ログイン                 | FlowForce Server 内で Directory Service が有効化されているとこのドロップダウンリストからドメイン名を選択し、ユーザーとパスワードフィールドにドメイン資格情報を入力します（前のオプションを参照してください）。   |
| リソースの使用、リソースパス       | マッピング関数がサーバーにデプロイ後にリソースを使用する場合リソースの使用 チェックボックスを選択します。チェックボックスを選択すると、「リソースパス」テキストボックス上の対応するリソースのパスを入力する必要があります。リソースを選択し、「省略記号」ボタンをクリックします。<br><br>サーバー上に選択するリソースが存在しない場合、「グローバルリソースのデプロイ」をクリックして、必要とされるグローバルリソースをサーバーにデプロイします。<br><br>「リソースの使用」チェックボックスを選択しない場合、グローバルリソースは現在選択されている構成をベースに解決されます。サーバー上では、マッピング関数はグローバルリソースを必要とせず、代わりに解決された値を使用します。 |
| パス                   | 「参照」をクリックして、FlowForce Server コンテナの階層内でマッピング関数が保存されるパスを選択します。デフォルトでパスは FlowForce Server の /public コンテナに設定されています。<br><br>必要な FlowForce Server パーミッションと特権が与えられている場合、既存のコンテナとマッピングを削除することもできます。   |
| デプロイ前にマッピングを保存する     | 保存されていないマッピングをデプロイする時にこのオプションを使用することができます。デプロイ前にマッピングを保存する場合このチェックボックスを選択します。   |
| 新しいジョブを作成するためブラウザを開く | このチェックボックスを選択した場合、FlowForce Server Web 管理インターフェイスがデプロイ後開かれ、FlowForce Server ジョブの作成の開始をすることができます。  |

## トラブルシューティング

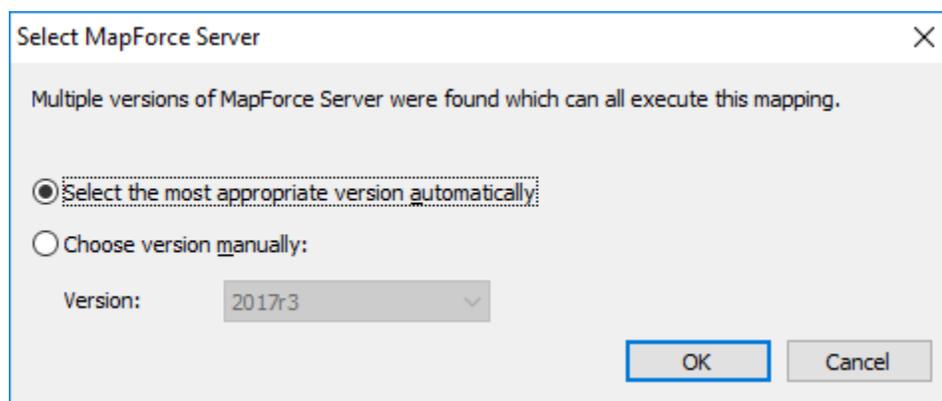
次のテーブルはマッピングをデプロイする際に発生する一般的な問題をリストしています。

| 問題   | ソリューション  |
|--|--|
| マッピングのデプロイは以下のエラーを返します:<br><br>ファイル上の I/O オペレーションに失敗しました。<br>I/O エラー 28: <server> ポート 8082 への接続に失敗しました。タイムアウト<br>システムエラー 10060: 接続されたパーティが一定の時間内に正確なレスポンドしなかった。または、接続されたホストがレスポンドに失敗し確立された接続が失敗したため接続の試みは失敗しました。 | ターゲットマシン上で FlowForce Web Server サービスが作動しており、指定されているポート（デフォルトで 8082）で接続のためにリスニングするように構成されていることを確認してください。またファイアウォールなどのポートを介して受信される接続をブロックしないようにしてください。<br><br>デプロイを可能にするために FlowForce Server サービスも作動している必要があります。 |
| マッピングのデプロイは以下のエラーを返します:  | デプロイされるマッピングの入力ファイルが FlowForce Server により許可されている HTTP リクエストの最大量（大体 100 MB）を超える   |

| 問題  | ソリューション   |
|---|---|
| ファイル上の I/O オペレーションに失敗しました。<br>I/O エラー 413: ペイロードが大きすぎます | とエラーが発生する場合があります。リミットを <code>max_request_body_size</code> オプションを (in bytes) <b>flowforceweb.ini</b> と <b>flowforce.ini</b> ファイル内で設定して増量することができます。詳細に関しては <a href="#">構成ファイルファレンス</a> を参照してください。 |

### サーバーバージョンの選択方法 (Windows のみ)

マッピングをデプロイするサーバーで FlowForce Server 管理下で、複数のバージョンの MapForce Server を作動している場合、(Windows サーバーのみに適用することができます) このマッピングを実行する MapForce Server のバージョンを指定するようプロンプトされます。



**メモ** FlowForce Server インストールディレクトリに .tool ファイルが含まれている場合、FlowForce Server の管理下で実行される各 MapForce Server のバージョンのために、ダイアログボックスが表示されます。デフォルトでは、MapForce Server を FlowForce Server の一部としてインストールすると、MapForce Server .tool ファイルは、このディレクトリに自動的に追加されます。FlowForce 内でファイルが保管される場所は以下のとおりです: **C:\Program Files\Altova\FlowForceServer2021\tools**。FlowForce Server 管理下での他の MapForce Server バージョンを作動する場合、.tool ファイルを手動で上記のディレクトリにコピーする必要があります。MapForce Server の .tool ファイルは以下に保管されています: **C:\Program Files\Altova\MapForceServer2021**。

## 8.3 マッピングと変換をジョブとして実行する方法

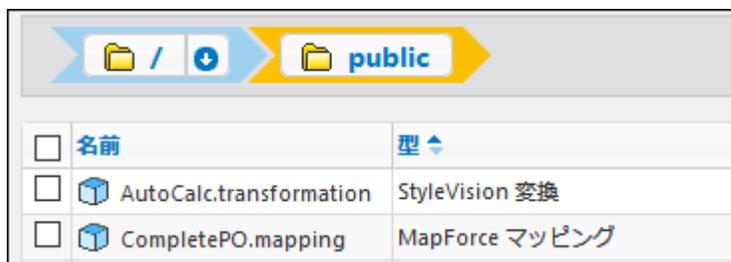
MapForce マッピング、または StyleVision 変換から FlowForce Server ジョブを以下のように作成することができます:

- 最初に、マッピング、または変換を FlowForce Server にデプロイします。このステップは MapForce (と StyleVision でそれぞれ)で行うことができます:

- 「ファイル」メニューから、「FlowForce (Server) へデプロイする」をクリックします。

デプロイの設定の説明は、MapForce と StyleVision ドキュメントを参照してください (<https://www.altova.com/documentation>)。

- FlowForce Server 内で、マッピング、または変換がデプロイされている個所に FlowForce コンテナをナビゲートします (例えば、コンテナ "/public")。



- 必要なマッピング、または変換をクリックし、「ジョブの作成」をクリックします。または、関数の実行 ボックスにチェックを入れて既存のジョブからマッピング、または変換を参照することができます:



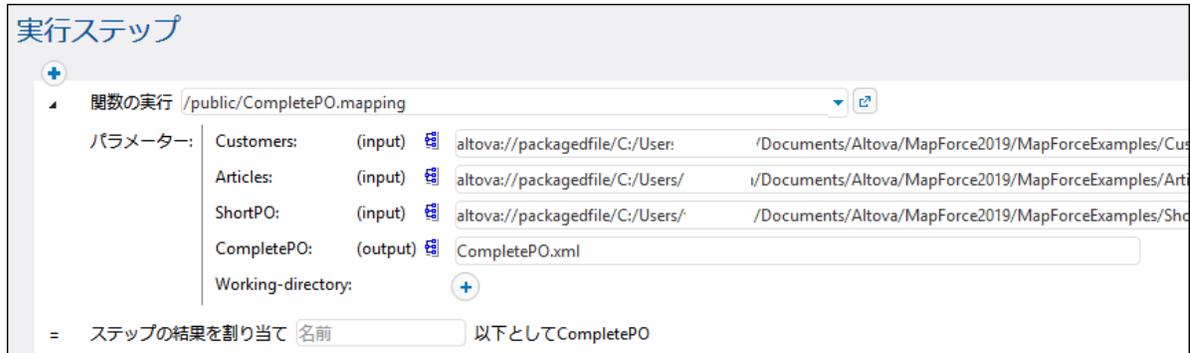
必要に応じてジョブを構成することができます。例えば、Web サービスとして、またはトリガーを使用して実行することができます。StyleVision 変換のデプロイ、および変換からジョブを作成する方法について説明するステップバイステップのサンプルに関しては、[StyleVision 変換からのジョブの作成](#)を参照してください。MapForce のための類似したサンプルに関しては、[MapForce マッピングからのジョブの作成](#)を参照してください。MapForce Server と StyleVision Server の両方を呼び出すサンプルジョブに関しては、[サンプル複数のPDFを複数のXMLから生成する](#)を参照してください。

変換、またはジョブのマッピングの実行の最も重要なパートは、ジョブの入力ファイルの処理です。2つのアプローチをとることができます: 入力ファイルを静的にジョブに提供する、またはジョブのランタイムに (例えば、ノスから) 動的に提供する。必要に応じて使用する正確なアプローチを決定します。ジョブが同じ入力データを使用して毎度実行される場合、最初のアプローチが適切です。それ以外の場合、ノスから動的に提供されるファイルからデータを取得するため、FlowForce ジョブが必要になるため、2番目のアプローチが適切です。

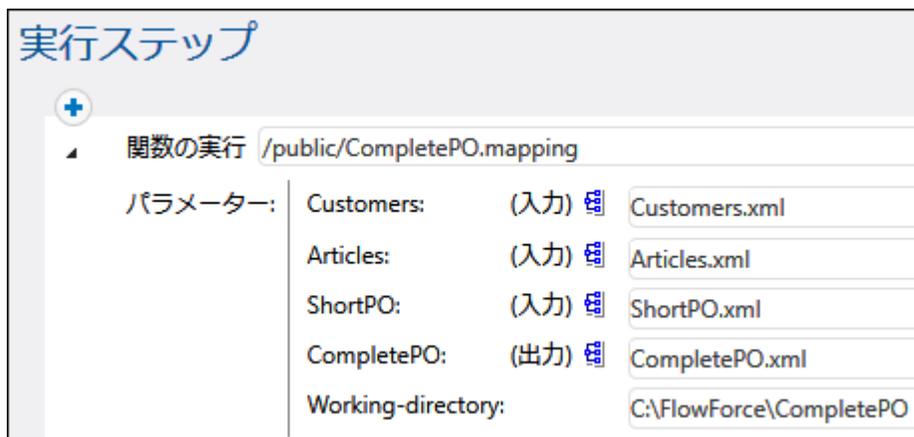
## MapForce マッピング

MapForce からマッピングがデプロイされる場合、(XML、CSV、JSON、Excel などの) インスタンスファイルがマッピングと共にデプロイされ、明示的に静的にパッケージされます。これは、ジョブが作動すると、FlowForce はデフォルトでパッケージ済みのファイルから静的にデータを読み取ります。ここで2つのシナリオが考えられます:

1. MapForce 内でマッピングを右クリックし、「生成されたコード内でパスを絶対にする」チェックボックスをマッピングをデプロイする前に選択します。FlowForce Server 内ですべての入力ファイルはプレフィックス `altova://packagefile/` と共に明示的に表示されます。



FlowForce Server にパッケージファイルからデータを読み取らぬように命令する場合は、プレフィックス `altova://packagefile` をパスから削除します。絶対パスは相対ファイルを使用してファイルへ参照することができます。相対パスを使用する場合、作業ディレクトリ/パラメータに対してパスは相対的になります。例えば `C:\FlowForce\CompletePO` からファイルを入力として提供する場合、`C:\FlowForce\CompletePO` を作業ディレクトリに設定し、以下に示される通り入力ファイルの名前を入力します。



2. 「生成されたコード内でパスを絶対にする」チェックボックスがFlowForce へマッピングをデプロイする前に選択されていない場合、FlowForce 内で入力ファイルは相対パスで表示されます。FlowForce はこの場合作業ディレクトリ内の同じ名前を持つファイルであっても、パッケージ済みのファイルからデータを読み取ります。FlowForce にパッケージ済みのファイルからデータを読み取らぬように命令する場合は、ファイルパスを絶対にする。または、以下に示されるようにジョブのパラメータとして提供します。

### ジョブ入力パラメーター

- 名前: customers × 型: ファイルとしての文字列 デフォルト: C:\Customers.xml
- 名前: articles × 型: ファイルとしての文字列 デフォルト: C:\Articles.xml
- 名前: ShortPO × 型: ファイルとしての文字列 デフォルト: C:\ShortPO.xml

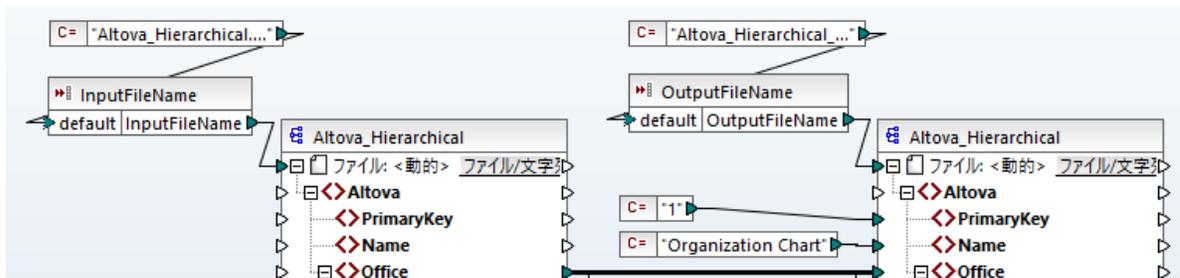
### 実行ステップ

関数の実行 /public/CompletePO.mapping

パラメーター:

|                    |      |                |
|--------------------|------|----------------|
| Customers:         | (入力) | Customers.xml  |
| Articles:          | (入力) | Articles.xml   |
| ShortPO:           | (入力) | ShortPO.xml    |
| CompletePO:        | (出力) | CompletePO.xml |
| Working-directory: |      | +              |

代わりに、入力ファイル名がマッピングの入力パラメーターになるように MapForce 内でマッピングデザインを変更することができます。例えば、下で説明されるマッピングは入力と出力ファイル名をパラメーターとしてとります。



FlowForce Server にデプロイすると、ジョブ構成ページのパラメーターがそのように表示されます (ファイル自身はパッケージされていません)。

### 実行ステップ

関数の実行 /public/FileNamesAsParameters.mapping

パラメーター:

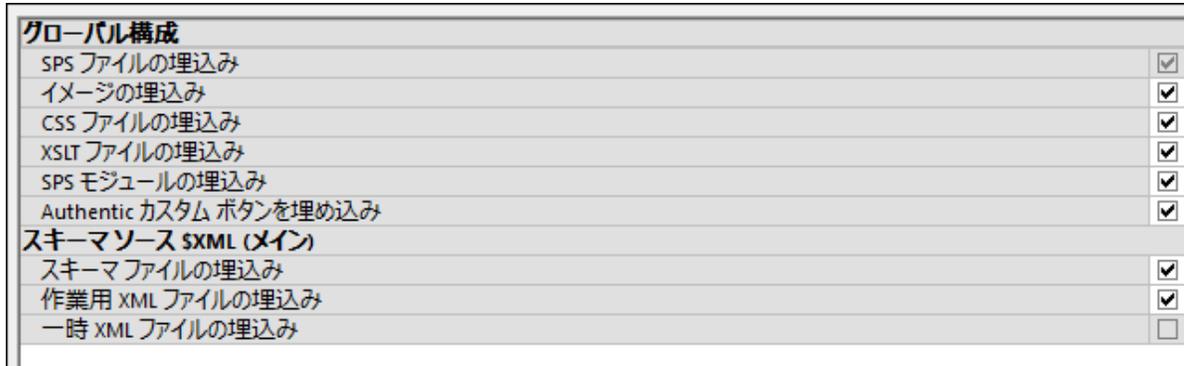
|                    |                           |
|--------------------|---------------------------|
| OutputFileName:    | output.xml                |
| InputFileName:     | Altova_hierarchical.xml   |
| Working-directory: | C:\Users \MapForceExample |

上記のマッピングは **FileNamesAsParameters.mfd** と呼ばれ、MapForce に含まれるサンプルファイルの一つです。このマッピングのデザイン方法に関しては、MapForce ドキュメントを確認してください。

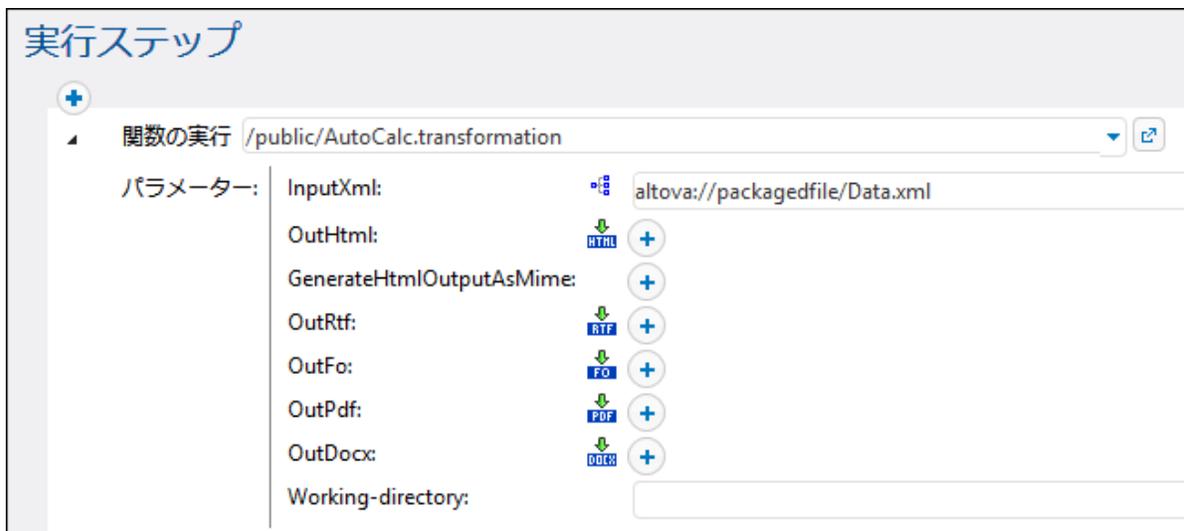
## StyleVision 変換

StyleVision 変換の場合、入力ファイルを以下のように処理することができます:

1. PXF (Portable XML Form) ファイルを StyleVision 内で開きます。SPS (StyleVision Power Stylesheet) が存在する場合、FlowForce Server にデプロイしようとすると、StyleVision は PXF 書式に変換するようにプロンプトします。
2. デザイン概要ウィンドウ内で、「埋め込まれたファイルの構成」をクリックします。ダイアログボックスが表示されます。



3. オプション「作業 XML ファイルの埋め込み」に注意してください。このチェックボックスを選択すると、作業 XML ファイルがデプロイされるパッケージの一部となり、デフォルトで FlowForce Server はジョブの実行の都度データを読み込みます。FlowForce 内でパッケージファイルが表示されます。



ジョブファイルを動的に提供するには、プレフィックス `altova://packagedfile/` を削除、または、パスを絶対パスに変更します。相対的なパスを使用する場合、相対的なパスを使用する場合、作業ディレクトリ/パラメータに対して、パスは相対的です。代わりに、FlowForce Server に変換をデプロイする前に、「作業 XML ファイルを埋め込む」チェックボックスをクリアします。

CSS ファイルまたはイメージなどのリソースのために「埋め込み」チェックボックスをクリアすると、FlowForce Server はジョブの作業ディレクトリ内でこれらを検索します。

### 8.3.1 マッピング関数内の資格情報

[資格情報](#) の紹介で説明されているとおり、FlowForce Server 内だけでなくマッピングデザイン時にも MapForce でも資格情報を作成することができます。

MapForce から FlowForce Server に資格情報を含むマッピングをデプロイする場合、サーバーに資格情報もデプロイされます。デプロイ

された情報は資格情報レコードを作成した際に記入されたフィールドのみが含まれます。例えば、資格情報名のみを保管することを選択し、空の資格情報である場合、または、ユーザー名とパスワードの両方を含む資格情報オブジェクトの場合。

MapForce から FlowForce Server へ資格情報オブジェクトをストアオブジェクトとしてデプロイすることができます。MapForce から直接デプロイ先のターゲットコンテナを選択することができます。詳細に関しては、MapForce ドキュメントを参照してください (<https://www.altova.com/ja/documentation>)。

フィールド / パスワード は機密なデータと考えられます。

MapForce 内のデザイン時に「実行可能ファイルとマッピングのデプロイに含まれる」チェックボックスを選択した場合のみ重要なデータがデプロイされます。これはマッピングをデプロイする際、およびストアに資格情報をデプロイする際に適用されます。

FlowForce Server 内では、対応するマッピング関数のページを開くことにより、マッピング関数が資格情報を必要とするか確認します。例:

### Function Input Parameters

|  |  |  |
|--|--|--|
| Name: <input type="text" value="airportCode"/>       | Type: <input type="text" value="string"/>              | optional   |
| Name: <input type="text" value="AirportStatus"/>     | (output) <input type="text" value="JSON"/>             | Default: <input type="text" value="AirportStatus.json"/> |
| Name: <input type="text" value="my.credentials"/>    | Type: <input type="text" value="credential"/>          |  |
| Name: <input type="text" value="Working-directory"/> | Type: <input type="text" value="string as directory"/> | Default: <input type="checkbox"/>                        |

「実行可能ファイルとマッピングのデプロイに含まれる」チェックボックスを資格情報を作成する際に選択した場合、マッピングと共に資格情報をジョブはデプロイします。この場合、ジョブ構成ページから指定する必要はありません。例えば、次の実行ステップは存在する場合保管された資格情報と共にマッピング関数を実行します (“my.credentials” / パラメータが展開されていないこと注目してください):

### Execution Steps

**+**

Execute function

|             |                    |   |   |
|-------------|--------------------|---|---|
| Parameters: | airportCode:       | Type: <input type="text" value="string"/>     | + |
|             | AirportStatus:     | (output) <input type="text" value="JSON"/>    | + |
|             | my.credentials:    | Type: <input type="text" value="credential"/> | + |
|             | Working-directory: |   | + |

FlowForce Server 内で直接定義された、または、ローカルの資格情報により定義された他の資格情報オブジェクトと保管された資格情報を上書きすることができます。これを行うには、「+」ボタンを押して、FlowForce Server 内に既存の資格情報オブジェクトを選択、または、直接ユーザー名とパスワードを入力します。例:

### Execution Steps

+ Execute function /public/AirportStatus.mapping

Parameters:

- airportCode: [icon] +
- AirportStatus: (output) JSON [icon] +
- my.credentials: [icon]
  - Select existing credential:
  - Define local credential:
    - User name: admin
    - Password: [mask]

実行ステップの引数として提供された資格方法は、マッピング関数内に保管されている資格情報より優先順位が高いです。

「実行可能ファイルとマッピングのデプロイに含まれる」チェックボックスを MapForce 内で資格情報を作成する際に選択しなかった場合、実行ステップの引数として資格情報を提供することは必須です。それ以外の場合、実行は失敗します。

### 8.3.2 動的な認証

MapForce 内では、ユーザー名とパスワードを入力パラメータとして受け入れるために基本的な HTTP 認証のための Web サービスを呼び出すマッピングを構成することができます。動的な認証の構成に関する詳細は、MapForce ドキュメント (<https://www.altova.com/ja/documentation>) を参照してください。

FlowForce Server に動的な認証を含むマッピングをデプロイする場合、ユーザー名とパスワードはマッピング関数の入力パラメータになります。このようなマッピング関数を呼び出す FlowForce Server ジョブは実行に成功するためユーザー名とパスワードが必要です。例:

### 実行ステップ

+ 関数の実行 /public/DynamicAuthentication.mapping

パラメーター:

- username: [icon] admin
- password: [icon] dj9JaVax
- Headers: (出力) [icon] +
- xml\_schema: (出力) [icon] +
- Working-directory: C:\Work

上記のサンプルでは、ユーザー名とパスワードはそれぞれのテキストボックス内に入力されています。しかしながら、ジョブの入力パラメータとして提供することもできます。[入力パラメータの管理](#) を参照してください。

### 8.3.3 リソース

Altova グローバルリソースはファイル、フォルダー、またはデータベースへのポータブルなレファレンスです。グローバルリソースとして保管されると、パスとデータベース接続の詳細は再利用できるようになり Altova アプリケーション全体で使用可能になります。例えば、複数の Altova デスクトップアプリケーション内で同じファイルを頻繁に開く必要がある場合、グローバルリソースとして定義すると便利な場合があります。このようにすると、「ファイルを開く」ダイアログボックスから対応するグローバルリソースを開くことができるため、ファイルパスを覚えておく必要がありません。この方法はファイルパスが変更されると、パスの一部を変更するのみに有利な場合があります。

グローバルリソースの一般的な使用法は、データベース接続を一度定義し、グローバルリソースをサポートする全ての Altova アプリケーションで再利用することです。例えば、MapForce マッピングがデザインされるマシンでデータベース接続を作成し、MapForce Server がマッピングを実行するマシン上で同じ接続を再利用することができます（これは、一部の場合、両方のマシンで同じデータベースクライアントソフトウェアがインストールされている場合があります）。

任意で、「構成」として既知の同じグローバルリソースの複数のバージョンを作成することができます。これによりファイル、またはフォルダーパス（またはデータベース）を必要に応じて変更することができます。例えば、「開発」と「生産」の構成を持つデータベースリソースを作成することができます。FlowForce Server を例に挙げると、マッピング関数に希望する構成を与えることで、開発、または生産データベースからデータを簡単に抽出することができます。

次の Altova デスクトップアプリケーションからグローバルリソースを作成することができます: Altova Authentic、DatabaseSpy、Mobile Together Designer、MapForce、StyleVision、および XMLSpy。サーバー製品では、次の Altova サーバーアプリケーションでグローバルリソースを作成することができます: FlowForce Server、MapForce Server、RaptorXML Server、RaptorXML+XBRL Server。

グローバルリソースの作成に関する詳細は MapForce ドキュメントの「Altova グローバルリソース」のチャプターを参照してください。

#### FlowForce Server 内のリソース

FlowForce Server の観点からは、デスクトップアプリケーション同様、一つの XML ファイルに保管されないため、グローバルリソースはある程度グローバルではなくります。FlowForce 内では、各リソースファイルまたはフォルダーパス、またはデータベース接続詳細を含む可能性のある再利用可能なオブジェクトです。リソースをコピー、エクスポート、またはインポートすることができ、他の FlowForce Server オブジェクトと同様に同じユーザーアクセスの対象とすることができます。つまり、FlowForce ユーザーはマッピング関数内のすべてのリソースを使用することができます。

MapForce 内でグローバルリソースを使用するマッピングを作成すると、FlowForce Server にデプロイすることができます。デプロイ時にデプロイメントダイアログボックス上の「リソースの使用」チェックボックスを選択することにより、マッピングがグローバルリソースを使用するかを選択することができます。チェックボックスを選択しない場合、マッピングにより使用されるグローバルリソースは現在選択されている構成をベースに解決されます。チェックボックスを選択すると、FlowForce Server 内でもマッピング関数はリソースを必要とします。次のマッピング関数のサンプルはリソースの実行を必要とする FlowForce にデプロイされます。（最初の「ラメーター」がリソースファイルからデフォルトのファイルパスを取得することにご注意ください）:

## 以下に関数 ReadJSON.mapping を作成 /public

### 関数入力パラメーター

名前:  (入力)  JSON 型:  デフォルト:

名前:  (出力)  型:  デフォルト:

名前:  型:  デフォルト:

### リソース

リソースを使用した実行関数:

上で説明されているとおり FlowForce Server 内では、ジョブではなくマッピング関数がグローバルリソースを使用します。例えば、マッピングがパスから読み取り、または、パスへ書き込むように構成されている場合が挙げられます。上のサンプルでは、マッピング関数はリソースから最初の入力ファイルを読み取ります。ジョブ構成ページからのパスをオーバーライドしない限り、これはこの関数を使用するすべてのジョブが同じパスを使用することを意味します。

FlowForce Server にグローバルリソースをスタンドアロンオブジェクトとしてデプロイすることができます。つまり、グローバルリソースをデプロイするためにマッピングを最初にデプロイする必要はありません。詳細に関しては FlowForce Server にグローバルリソースをデプロイする詳細に関しては、MapForce ドキュメントを参照してください (<https://www.altova.com/ja/documentation>)。

### リソースの構造

すべての Altova デスクトップアプリケーション内では、グローバルリソースは XML ファイルとして管理されます。デフォルトのファイルは **GlobalResources.xml** で、MapForce がインストールされているコンピューター上の **C:\Users\\Documents\Altova** ディレクトリで見つけることができます。グローバルリソースファイルが「エイリアス」として既知の複数のリソースを含んでいます。エイリアスはファイルパス、ディレクトリパスデータベース接続の詳細のグループです。前述の通り、構成により、パスまたはデータベースを切り替えることができます。これは以下のサンプルグローバルリソースファイルの構造を見ることにより理解することができます (データの一部分は簡素化のため削除されていることにご注意ください)：

```
<Resources>
  <Resource Alias="MyFile">
    <Configurations>
      <Configuration Location="C:\test.json" ContentKind="File"
Configuration="Default"/>
      <Configuration Location="C:\production.json" ContentKind="File"
Configuration="Production"/>
    </Configurations>
  </Resource>
  <Resource Alias="MyDirectory">
    <Configurations>
      <Configuration Location="C:\Test" ContentKind="Folder"
Configuration="Default"/>
      <Configuration Location="C:\Production" ContentKind="Folder"
Configuration="Production"/>
    </Configurations>
  </Resource>
</Resources>
```

```

Configuration="Production"/>
  </Configurations>
</Resource>
<Resource Alias="MyDatabase">
  <Configurations>
    <Configuration ContentKind="DataSource" Configuration="Default">
      <DatabaseContextInfo vendor="sqlite" connection="C:
\Resources\Test.sqlite"/>
    </Configuration>
    <Configuration ContentKind="DataSource" Configuration="Production">
      <DatabaseContextInfo vendor="sqlite" connection="C:
\Resources\Production.sqlite"/>
    </Configuration>
  </Configurations>
</Resource>
</Resources>

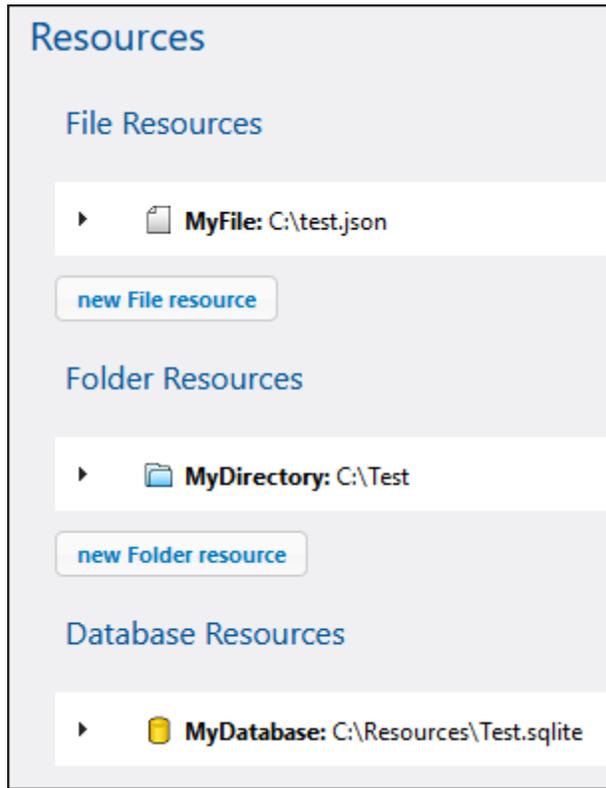
```

上記のファイルおつのリソースエイリアスを定義します: 「MyFile」と呼ばれるファイル、 「MyDirectory」と呼ばれるディレクトリ、 「MyDatabase」と呼ばれる SQLite データベース。各エイリアスにおつの構成が存在します: テストのために使用されるデフォルトの構成と生産の構成です。

FlowForce Server 内では、複数のユーザーサーバー環境の特性のため、リソースは若干異なって作動します。具体的には、FlowForce 内でリソースオブジェクトになるような XML リソースファイルの場合、デスクトップアプリケーションのように複数のエイリアスが存在することができます。各エイリアスにおつの構成のみ存在することができ、MapForce から FlowForce Server へデPLOYする際に選択されている構成を指します。

MapForce から FlowForce Server にグローバルリソースをデPLOYする場合、一度におつの構成だけをデPLOYすることができます。

例えば、上記のグローバルリソースファイルをデPLOYした場合、サーバーに「デフォルト」または「生産」構成のどちらかがデPLOYされます（一度におつ方の構成はデPLOYされません）。「デフォルト」の構成を選択する場合、リソースオブジェクトは FlowForce Server 内で以下のようになります:



FlowForce 内では、マッピング関数はグローバルリソースの特定の構成を1つ使用することができます。このため、このサンプルではサーバー上の「生産」構成が必要になり、MapForce 内のデプロイダイアログボックスから構成「生産」を選択し同じリソースをもう一度デプロイします。代わりに、サーバー上で以下で説明される通り直接リソースを作成し、マッピング関数を「デフォルト」の代わりにコメントするよう変更することができます。代替のアプローチはファイルとディレクトリリソースに対してのみ可能で、データベースに対しては可能ではありません。

### マッピング関数のリソースの変更

FlowForce 内では、アイコンによりリソースオブジェクトは識別されます。このため、上のサンプルから「デフォルト」と「生産」構成がデプロイされている場合、FlowForce Server 内の対応するリソースが以下のように表示されます:

| <input type="checkbox"/> | Name  | Type      |
|--------------------------|---|-----------|
| <input type="checkbox"/> |  ExampleGlobalResources_Default.resources    | resources |
| <input type="checkbox"/> |  ExampleGlobalResources_Production.resources | resources |

マッピング関数により使用されたリソースを変更する方法:

1. マッピング関数がデプロイされているコンテナに移動し、クリックして関数を開きます。
2. 「リソース」の下で、新規リソース アイコンを選択します。リソースの選択は関数や資格情報などの他の FlowForce オブジェクトと同様に作動します。



マッピング関数が「リソース」セクションを持たない場合、(または、デPLOY時に「リソースの使用」チェックボックスが選択されていない場合)マッピングはMapForce 内のグローバルリソースのため構成されません。

以下の条件が満たされている場合マッピング関数はすべてのリソースを使用することができます:

- リソース型が関数と互換性を持つ場合。例えば、「フォルダー」リソースが「ファイル」リソースを必要とするマッピング関数には適していません。
- リソースエイリアス名はマッピング関数により必要とされているものです。MapForce 内では通常エイリアス名はマッピングのデザイン時に選択されますが、FlowForce 内では以下で説明されている通りオーバーライドすることができます。

### リソースとジョブ構成

上記の通り、リソースはジョブ構成レベルではなくマッピング関数レベルで消費されます。ジョブが実行されると、ジョブにより呼び出される関数内で定義されているグローバルリソースを消費します。このため、ジョブ構成ページからジョブを編集する場合、リソースエイリアスのオーバーライドなどリソースに対して最小の構成オプションのみを使用することができます(下で詳しく説明されています)。

一部の場合は、(フォルダーまたはファイルのように)リソースを直接ジョブ構成ページから参照することが可能な場合があります。これは、全てのコンテキストで可能な場合がありますが、このようなレファレンスを使用する良い理由がほとんど限り一般的には回避されるべきです。

**メモ** 実行ステップの作業ディレクトリパラメータからリソースを参照してはなりません。これはリソースの処理がMapForce Server 処理が既に開始されていることを必要としますが、作業ディレクトリはMapForce Server が開始する前に設定されているからです。

### リソースエイリアスの上書き

ファイル、またはフォルダーリソースは複数のエイリアスを持つことができますが、ジョブランタイムでは一つのエイリアスのみ使用することができます。マッピングのデザイン時にMapForce 内でランタイム時に使用されるエイリアスが選択されます。例えば、次のMapForce コンポーネントは“MyDirectory”と呼ばれるディレクトリエイリアスにoutput.csv を生成するため構成されます。FlowForce Server |このマッピングをデPLOYする場合、サーバー上のマッピング関数は“MyDirectory”エイリアスを含むリソースを指している必要があります。

エイリアスを変更する都度 MapForce 内でマッピングを編集する代替として、ジョブ構成ページから FlowForce Server 内でエイリアスをオーバーライドすることができます。ジョブ内のファイルまたはフォルダエイリアスをオーバーライドする場合、**MyFile** または **MyDirectory** を必要とされるエイリアス名に置き換えます。

| 必要とされる型 | サンプル   |
|---------|--|
| ファイル    | altova://file_resource/ <b>MyFile</b>        |
| ディレクトリ  | altova://folder_resource/ <b>MyDirectory</b> |

例えば、下のジョブの構成ではディレクトリエイリアスが「TestDir」に変更されています。

**メモ** 上記の通りエイリアスの上書きはデータベースリソースのためにサポートされていません。複数のデータベースエイリアスが存在する場合、マッピングを FlowForce Server にデプロイする前に MapForce 内で必要とされるデータベースに切り替えます。

### リソースの作成

FlowForce Server 内でファイルまたはフォルダリソースを作成することができます。FlowForce Server 内でグローバルリソースを作成するには、コンテナーを作成して「作成 | リソースの作成」をクリックします。

Resources name: Resource1

Resources description: Defines miscellaneous input or output paths used in mapping jobs.

## Resources

### File Resources

Resource alias: SourceFile

File: C:\Resources\data.txt

new File resource

### Folder Resources

new Folder resource

Save

**メモ** データベースリソースの作成はサーバー環境ではサポートされていません。データベースリソースを作成するには、MapForce または他の Altova デスクトップアプリケーションのグローバルリソースエディターを使用し、MapForce から FlowForce Server へリソースをデプロイしてください。

リソースエイリアスはこのリソースを使用する箇所のマッピング関数により比喩町とされているものに一致する必要があります。それ以外の場合、「リソースエイリアスの上書き」で説明されている通り、正確なエイリアスを指すようジョブを手動で行う必要があります。

同じリソースオブジェクト内で、必要な場合、「新規のファイルリソース」または「新規のフォルダーリソース」ボタンをクリックして複数のエイリアスを作成することができます。これはオプションです。複数のエイリアスを作成する場合、どのエイリアスが使用されるか指すためジョブを変更する必要があることを忘れてはいけません。

## リソースの編集

MapForce 内で行い、再デプロイする代わりに FlowForce Server 内でファイルまたはフォルダーリソースを編集することができます。リソースを編集するには、対応するレコードをクリックし、パス（またはデータベース接続の詳細）を更新し、「保存」をクリックします。

**メモ** データベースリソースの場合、接続文字列、または、デフォルトのデータベースなどの特定のフィールドのみを FlowForce 内で編集することができます。データベース名と接続メソッドを変更することはできません。

リソースを更新すると、この更新は以下に影響を及ぼします:

- リソースを参照する全てのマッピング関数
- 対応するマッピング関数を呼び出す全てのジョブ

## 8.4 マッピング/変換結果へのアクセス

MapForce マッピング および StyleVision 変換が FlowForce Server にデプロイされた後、実行ステップから呼び出すことのできる FlowForce 関数があります。例えば、下のジョブの最初のステップは **SimpleTotal.mapping** と呼ばれ、実行されます。

### 実行ステップ

**+**

関数の実行

パラメーター:

|                    |                           |          |
|--------------------|---------------------------|----------|
| ipo:               | (入力) <input type="text"/> | <b>+</b> |
| Working-directory: | <input type="text"/>      | <b>+</b> |

= ステップの結果を割り当て

**+**

関数の実行

パラメーター:

|      |   |
|------|---|
| 条件式: | <input type="text" value="stdout(output)"/> |
|------|---|

= ステップの結果を割り当て  以下として T0

新規の実行ステップ

新規の選択ステップ

新規の For-each ステップ

新規エラー/成功処理ステップ

ジョブは 2つのステップにより構成されています:

1. ステップ 1 は、MapForce Server は、実際に実行する **SimpleTotal.mapping** 関数を呼び出します。重要な点は、「このステップの結果を割り当てる」フィールドがマッピングの結果に名前を与える点に注意してください(この場合、**output**)。しかしながら、名前を変更することは可能です。
2. ステップ 2 は、マッピングの出力をストリームに変換する **/system/compute** 関数を呼び出します。

デフォルトでは、マッピング および 変換 関数の出力は、ジェネリックな型「result」です。出力を使用するには、「result」を必要なデータ(例えば、文字列、ストリーム、ファイル)に変換する必要があります。この目的のために、**/system/compute** ビルトイン関数と FlowForce 式関数を使用することができます。上のサンプルでは、ビルトイン関数 **/system/compute** が必要とするデータ型を変換するために呼び出されています。すなわち、式 **stdout(output)** は、前のステップの結果をストリームに変換します。

下のテーブルは、マッピング および 変換関数の結果を処理するために使用される可能性のある FlowForce 式の列をリストしています。これら全てのサンプルは、内に入力された **output** は、「このステップの結果を割り当てる」内に入力されている名前であることにご注意ください。

| FlowForce 式                                   | 目的  |
|---|---|
| <code>stdout(output)</code>                   | <b>output</b> をストリームに変換します。   |
| <code>as-file(stdout(output))</code>          | <b>output</b> をファイルに変換します。  |
| <code>as-file(nth(results(output), 0))</code> | <b>output</b> が複数のファイルにより構成されている場合このような式が必要になります。(MapForce および StyleVision 内で 単一の出力ではなく、複数の出力を生成するようにデザインされると発生しま |

| FlowForce 式   | 目的   |
|---|--|
|   | <p>す。式は <code>output</code> をストリームのシーケンスに変換し、シーケンスから最初のストリームをピックアップし、ファイルに変換します。サンプルに関しては、次を参照してください：<a href="#">StyleVision 変換からジョブを作成する</a>。</p> |
| <pre>as-file (nth (results (<code>output</code>, "CompletePO"), 0))</pre> | <p>ゼロベースのインデックスではなく、名前別にストリームのシーケンスからファイルは抽出されます(この場合、"CompletePO")。</p>   |

ステップ [または](#) ジョブの結果を処理するための FlowForce 式関数に関するレファレンスについては、次を参照してください：[ステップ結果関数](#)。FlowForce 式の概要については、[FlowForce 式](#)を参照してください。

## 8.5 RaptorXML Server との統合

RaptorXML が FlowForce に統合される場合、RaptorXML Server により公開されているすべての関数はジョブ内で呼び出すことができるように FlowForce で使用できるようになります。具体的には、RaptorXML 関数は FlowForce の **/RaptorXML** コンテナ内に存在しています。RaptorXML+XBRL Server の場合、コンテナ名は **/RaptorXMLXBRL** です。

| 名前            | 型    | 日付は変更されました | 以下により変更されました: 次の実行 |         |
|---------------|------|------------|--------------------|---------|
| RaptorXML     | コンテナ |            |                    | パーミッション |
| RaptorXMLXBRL | コンテナ |            |                    | パーミッション |
| public        | コンテナ |            |                    | パーミッション |
| system        | コンテナ |            |                    | パーミッション |

FlowForce ビルトイン関数の呼び出しに類似したジョブから RaptorXML 関数を呼び出すことができます。

- **/RaptorXML** (または **/RaptorXMLXBRL**) コンテナ内で、興味のある関数を開き、「ジョブの作成」をクリックします。**/RaptorXML/valjson** などのジェネリックな関数を参照、または **/RaptorXML/2021/valjson** などのを参照することができます。これら2つの相違点は以下で説明されています。
- ジョブ内で新規の実行ステップを作成し、実行ステップから希望する RaptorXML 関数を呼び出します。例えば、下のステップは **valjson** 関数を呼び出します。

**実行ステップ**

関数の実行: /RaptorXMLXBRL/valjson

パラメーター:

- 作業用ディレクトリ: C:\Jobs
- エラーフォーマット: テキストフォーマット
- JSON ファイル: orders.json
- エラー制限:  raptorxmlcmdl.core\_error\_limit\_value\_unlimited  他: 100
- 詳細:
- 警告制限: +
- JSON5 サポート: +
- JSON ラインサポート: +
- 書式のチェックを無効化する: +
- JSON スキーマパス: orders.json.schema
- JSON スキーマバージョン: +
- XML ユーザーカタログ: +

RaptorXML Server を呼び出すジョブのサンプルに関しては、以下を参照してください。

- [RaptorXML を使用してドキュメントを検証する](#)
- [エラーログのある XML を検証する](#)
- [RaptorXML を使用して Key/Value / パラメーターペアを入力する](#)

すべての RaptorXML 関数への参照に関しては、RaptorXML Server ドキュメントを参照してください (<https://www.altova.com/ja/documentation>)。

### 手動の統合

FlowForce Server と RaptorXML Server 間の統合は多くの場合自動的に行われます (例えば、FlowForce Server インストールを Windows で実行し、RaptorXML Server もインストールする場合など)。しかしながら、2つの間での統合が手動で行われる必要が

ある場合もあります。手動の統合は通常、個別に異なるバージョンの FlowForce Server と RaptorXML Server がインストールされている場合に必要とされます。例えば、特定の RaptorXML Server バージョンの関数の定義が FlowForce Server インターフェイスで不足している場合、RaptorXML Server のそのバージョンがインストールされている場合でも、手動の統合が必要とされます。

手動の統合を行うには、以下のパスで見つけることのできるスクリプトを実行します: **RaptorXML installation directory** \etc\functions\integrate.bat。

**メモ** Unix システムでは、スクリプト名は **integrate.cs** です。スーパーユーザー権限 (sudo) はこのスクリプトを実行するために必要とされています。

スクリプトは2つの引数を取ります: FlowForce Server インストールディレクトリへのパスと、FlowForce Server データディレクトリへのパス ([FlowForce Server アプリケーションデータ](#)を参照してください)。このスクリプトを実行すると、次の状態が発生します:

- 統合された RaptorXML Server バージョンのすべてのリリース固有の関数がジョブとして呼び出すために FlowForce Server で使用できるようになります。
- ジェネリックな (リリースを指定しない) RaptorXML 関数は統合された RaptorXML バージョンのリリース固有の関数のポイントまでアップデートされます。

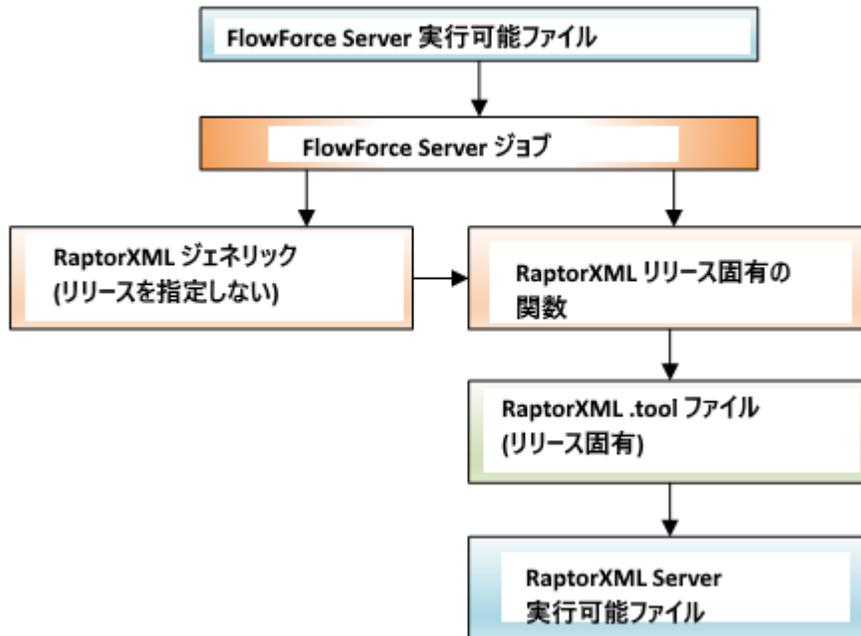
スクリプトがエラーを返すと、統合された RaptorXML バージョンの関数の定義は、FlowForce Server と互換性を持っていません。もしもこの状態が発生した場合はサポートに連絡してください。

## ジェネリックとリリース固有の RaptorXML 関数

RaptorXML、または RaptorXMLXBRL コンテナで使用する関数は以下のように整理されています:

- **/RaptorXML** コンテナからの関数は (RaptorXML 関数をサポートする最初のバージョンである) FlowForce Server の 2014 バージョンまで下位互換性があります。これらのジェネリックな関数は **/RaptorXML/{Release}** コンテナからのリリース固有の関数のラッパーとしての役割を果たします。これはリリース間では互換性がありますが、最近インストールされている RaptorXML Server のすべての機能を提供しません。
- **/RaptorXML/{Release}** コンテナからの関数は対応する RaptorXML リリースのすべての機能を提供します。これらの関数は、同じリリースの FlowForce Server と互換性があります。しかしながら、RaptorXML Server のすべてのバージョンは FlowForce Server のすべてのバージョンと互換性を持つとは限りません。 (「手動の統合」で説明されているとおり) 統合スクリプトを実行して互換性を確認してください。

ジョブが RaptorXML 関数を呼び出す場合、関数は RaptorXML Server のリリース固有の関数と同等のラッパーとしての役割を果たします。選択された RaptorXML リリースは手動で統合されたものを含む最近 FlowForce に統合されたものです。上記の通り、(新規の引数、または、関数などのこのよう呼び出しは最新の RaptorXML 機能からの機能の利益を得ることはありません。FlowForce ジョブから最新の RaptorXML 機能を活用するには、リリース固有の関数を触媒呼び出します。



リリース固有の関数は RaptorXML .tool ファイルが RaptorXML 実行可能ファイルを検索するために使用されるかを決定します。個別の .tool ファイルは各 RaptorXML Server リリースのために存在します。 .tool ファイルは FlowForce Server に RaptorXML Server 実行可能ファイルのロケーションを指定し、環境変数を設定するためにも使用することができます。[環境変数の設定](#)を参照してください。

FlowForce ジョブがバージョン固有の RaptorXML 関数を参照する場合、そして FlowForce Server と RaptorXML Server の新しいバージョンにアップグレードする場合、最新のリリース固有の RaptorXML 関数の時点にまですべてのジョブを変更、または Raptor.tool ファイルを RaptorXML Server 実行可能ファイルの最新のバージョンに以下の通りマップすることができます:

1. **Raptor\_<release>.tool** を最近にインストールされたリリースの RaptorXML Server の {installation}\etc ディレクトリから同じリリースの FlowForce Server の {configuration data}\tools ディレクトリにコピーします。
2. (ジョブがポイントする Raptor リリースを指すように古いリリースのバージョンに一致するようにファイルの名前を変更します。例えば、古いリリースが RaptorXML 2017r3 の場合、ファイルを Raptor\_2017r3.tool に名前を変更します。

マッピングアプローチをとる場合、全ての既存のジョブは RaptorXML 2017r3 関数 を呼び出すように継続され、.tool ファイルは最新の RaptorXML Server 実行ファイルまでマップします。

## 8.6 ツールファイル

他の Altova サーバーが FlowForce Server と共にインストールされている場合、.tool 拡張子が FlowForce Server 管理下で動作する各実行可能ファイルのためにインストールされます。FlowForce Server は、.tool ファイルのために **DATADIR\tools** と **INSTALLDIR\tools** ディレクトリをスキャンします。前のディレクトリが後のディレクトリよりも優先順位を与えられます。

### FlowForce Server アプリケーションデータディレクトリ(DATADIR)

|         |  |
|---------|--|
| Linux   | /var/opt/Altova/FlowForceServer2021/data       |
| macOS   | /var/Altova/FlowForceServer2021/data           |
| Windows | C:\ProgramData\Altova\FlowForceServer2021\data |

### FlowForce Server インストールディレクトリ(INSTALLDIR)

|         |  |
|---------|--|
| Linux   | /opt/Altova/FlowForceServer2021/   |
| macOS   | /usr/local/Altova/FlowForceServer2021/   |
| Windows | C:\Program Files\Altova\FlowForceServer2021\<br>C:\Program Files (x86)\Altova\FlowForceServer2021\ |

**DATADIR\tools** ディレクトリは通常は空で、カスタマイズされたツールファイルを配置する場所です。**INSTALLDIR\tools** ディレクトリに含まれているツールファイルは変更される必要はありません。

### アプリケーションプール

(名前により識別される)検出されたそれぞれの .tool ファイルのために、FlowForce Server はアプリケーションプールを作成します。アプリケーションプールはすでに実行中のツール処理インスタンスをグループ化し、FlowForce Server はこれらのインスタンスをツールファイル内で構成される通り管理します。

FlowForce がツール処理インスタンスのライフタイムに関する情報を強制する場合、これらのイベントはログ内に情報メッセージを生成します。例:

```
Starting instance {id} of {tool} for {session}.
Starting {commandline}.
Instance {id} of {tool} for {session} is now idle.
Shutting down instance {id} of {tool} for {session}; sitting idle for too long.
Shutting down instance {id} of {tool} for {session}; maximum reuse count reached.
Instance {id} of {tool} for {session} unexpectedly ceased communication.
Instance {id} of {tool} for {session} attached to job instance {instanceid}.
```

上記にリストされている情報メッセージはライセンスまたはキューに関する問題を表示しません。代わりに特定の時刻に実行された処理に関する情報を提供することで可能な問題をトラックすることができます。ステップまたはジョブが失敗すると、個別のログメッセージが生成されます。

### ツールファイル構成

常に必要ではありませんが、ツールファイルを編集することができます。具体的に、以下の構成オプションを提供します:

1. [ツール] セクションの下の実行可能 / 古いバージョンの .tool ファイルが新しいバージョンを実行する、またはその逆の場合など特定のケースではこのパスを変更する必要があります。

2. [Environment] セクションツールにより必要とされる環境変数を定義するためこのセクションを追加または編集することができます。[環境変数の設定](#)を参照してください。

#### 重要点:

- Altova サポートにアドバイスされた場合を除き上記以外のツールファイル設定を変更しないでください。
- カスタムツールを定義することは可能です。

## 8.6.1 環境変数の設定

FlowForce Server 管理下でオブジェクトとして、MapForce Server マッピング、または、StyleVision Server スタイルシートが作動する場合、では環境変数が設定される必要がある場合があります。(例えば、データベースに接続する際にJDBCドライバーの場所を指定するためのCLASSPATH)。

MapForce Server マッピング、または、StyleVision Server 変換に必要な環境変数を設定するには、対応するAltova サーバー製品の .tool ファイルを編集してください。 .tool ファイルを編集するには、 **DATADIR** がアプリケーションのデータディレクトリである **DATADIR\tools** を最初にチェックしてください!

#### FlowForce Server アプリケーションデータディレクトリ(DATADIR)

|         |  |
|---------|--|
| Linux   | /var/opt/Altova/FlowForceServer2021/data       |
| macOS   | /var/Altova/FlowForceServer2021/data           |
| Windows | C:\ProgramData\Altova\FlowForceServer2021\data |

.tool ファイルが **DATADIR\tools** に存在しない場合、 **INSTALLDIR** がアプリケーションのインストールディレクトリである FlowForce Server の **INSTALLDIR** ディレクトリからコピーしてください。

#### FlowForce Server インストールディレクトリ(INSTALLDIR)

|         |  |
|---------|--|
| Linux   | /opt/Altova/FlowForceServer2021/   |
| macOS   | /usr/local/Altova/FlowForceServer2021/   |
| Windows | C:\Program Files\Altova\FlowForceServer2021\<br>C:\Program Files (x86)\Altova\FlowForceServer2021\ |

MapForce Server または StyleVision Server が FlowForce Server の後にインストールされている場合、 **INSTALLDIR\tools** ディレクトリ内に .tool ファイルを見つけることができます。 .tool ファイルが FlowForce アプリケーションデータディレクトリ、または FlowForce インストールディレクトリに存在しない場合、 FlowForce Server が MapForce Server または StyleVision Server の後にインストールされた可能性が高いです。 MapForce Server または StyleVision Server に相対的な **etc** ディレクトリ内で .tool ファイルを見つけることができます。

(インストールディレクトリではなく) FlowForce Server アプリケーションデータディレクトリに .tool ファイルをコピーしてください。 FlowForce Server アプリケーションデータディレクトリ内の .tool ファイルは他のディレクトリに既存の .tool ファイルより優先順位が高いです。

**警告:** [migratedb](#) コマンドを FlowForce の最新のメジャーバージョンへのアップグレード中に実行すると、前のバージョンのアプリケーションデータディレクトリからの .tool ファイルは新しいバージョンのアプリケーションディレクトリにコピーされます。これは期待されない結果を引き起こす可能性がありますので、アプリケーションデータディレクトリ内で必要な .tool ファイルのみを保管するようにしてください。

.tool ファイル内の [Environment] セクションの下に必要な環境変数を追加することができます。例えば、CLASSPATH 変数を設定する .tool ファイルは、以下のようになります:

```
[Environment]
CLASSPATH=./usr/local/jdbc/oracle/ojdbc6.jar
```

サンプル .tool ファイル (Linux)

.tool ファイル内で設定された環境変数は、他の方法で定義された環境変数をオーバーライドします。

シェルコマンドの実行、または、FlowForce Server ジョブとしてスクリプトとして実行するには、[/system/shell/commandline](#) 関数を参照してください。

## 9 コマンドライン インターフェイス

上記のとおり FlowForce ソリューションは2つのサービスにより構成されています: FlowForce Server と FlowForce Web Server. これらのサービスのそれぞれのために、コマンドラインで実行可能な管理コマンドをサポートする実行可能ファイルを使用することができます。両方の実行可能ファイルを次のリストで見つけることができます:

|         |  |
|---------|--|
| Linux   | /opt/Altova/FlowForceServer2021/bin/             |
| macOS   | /usr/local/Altova/FlowForceServer2021/bin/       |
| Windows | <ProgramFilesFolder>\Altova\FlowForceServer\bin\ |

実行可能ファイル名は以下の通りです:

|         |   |
|---------|---|
| Linux   | flowforceserver<br>flowforcewebserver         |
| macOS   | flowforceserver<br>flowforcewebserver         |
| Windows | FlowForceServer.exe<br>FlowForceWebServer.exe |

コマンドラインインターフェイス (CLI) をライセンス供与、トラブルシューティング、内部データベースのバックアップなど管理の目的のために使用することができます。CLI によりサポートされるコマンドは下にリストされています:

**メモ** 省略 [FFS] と [FFW] はコマンドが FlowForce Server または FlowForce Web Server のために使用可能かを指定します。

| コマンド                                  | FFS | FFW | 説明   |
|---------------------------------------|-----|-----|--|
| <a href="#">help</a>                  | はい  | はい  | 引数として与えられたコマンドのためのヘルプを表示します。   |
| <a href="#">assignlicense</a>         | はい  |     | Windows プラットフォームのみこのコマンドを適用することができます。FlowForce Server にライセンスファイルをアップロード、およびライセンスファイルを割り当てるために使用されます。 |
| <a href="#">compactdb</a>             | はい  |     | 削除済みのレコードが含まれる場合 FlowForce .db ファイルのサイズを削減します。   |
| <a href="#">createdb</a>              | はい  |     | 新規の FlowForce データベースの作成  |
| <a href="#">debug</a>                 | はい  | はい  | デバッグモードでアプリケーションを開始します。  |
| <a href="#">exportresourcestrings</a> | はい  | はい  | XML ファイルに全てのリソース文字列をエクスポートします。   |
| <a href="#">foreground</a>            | はい  | はい  | 前景モードでアプリケーションを開始します。  |
| <a href="#">initdb</a>                | はい  |     | FlowForce データベースを作成または更新します。   |
| <a href="#">install</a>               | はい  | はい  | アプリケーションを Windows サービスとしてインストールします。  |

|                                  |    |    |   |
|----------------------------------|----|----|---|
| <a href="#">licenseserver</a>    | はい |    | ローカルネットワーク上のAltova LicenseServer にFlowForce Server を登録します。  |
| <a href="#">migratedb</a>        | はい |    | FlowForce Server データを前のバージョンから最新のバージョンに移行します。   |
| <a href="#">repair</a>           | はい |    | 修復モードでアプリケーションを開始します。   |
| <a href="#">resetpassword</a>    | はい |    | 👤 root ユーザーのパスワードをデフォルトの値にリセットし、👤 root ユーザーに全ての特権を与えます。   |
| <a href="#">setdeflang   sdl</a> | はい | はい | デフォルトの言語を設定します。   |
| <a href="#">start</a>            | はい | はい | アプリケーションをサービスとして開始します。  |
| <a href="#">uninstall</a>        | はい | はい | アプリケーションを Windows サービスとしてアンインストールします。   |
| <a href="#">upgradedb</a>        | はい |    | FlowForce Server データベースを最新のバージョンにアップグレードします。  |
| <a href="#">verifylicense</a>    | はい |    | Windows プラットフォームのみこのコマンドを適用することができます。FlowForce Server にライセンスが与えられているか、与えられたライセンスキーが既に FlowForce Server に与えられているかを検証するために使用されます。 |

## 規則

規則としては、このドキュメンテーションは特定のコマンドを説明する際に実行可能ファイルのフルパスを省略し、実行可能ファイルの名めを使用する代わりに `flowforceserver` を使用します。例:

```
flowforceserver help
```

`flowforceserver` は実行可能ファイルのパスが名前です。絶対パスを使用する場合、コマンドプロンプトウィンドウ(ターミナル)がある現在のディレクトリに関わらずコマンドを実行することができます。しかしながら、実行可能ファイルの名前を入力するだけで呼び出す場合、以下を最初に行ってください:

- ターミナルの現在のディレクトリを FlowForce Server インストールディレクトリに変更します。
- ディレクトリを実行可能ファイルが PATH 環境変数である場所に追加します。

これらのシナリオの詳細は以下で説明されています。

## ヒントとトリック

コマンドラインの初心者の場合、以下のヒントとトリックに注意してください。

- コマンドラインウィンドウがある現在のディレクトリを探すには `pwd` を Linux と macOS 上で入力します。Windows 上では `echo %CD%` を入力します。
- **Tab** キーを活用して素早く複数のファイルまたはディレクトリパスを全て入力することなく素早く入力してください。例えば `cd c:\prog` をコマンドラインに入力し、**Tab** を押すと `C:\Program Files` が自動的に入力されます(または、`[Prog]` で開始する `C:\` の下の他のディレクトリが入力されます)。
- Windows 上で `C:\Program Files` などの空白が含まれるパスを入力する場合、引用符で囲んでください。
- [このコマンドは内部または外部コマンド、実行可能なプログラム、またはバッチファイルとして認識されていません] 類似したメッセージが表示されると、パスまたはコマンドを間違えて入力した可能性があります。

- Linux 上ではファイルまたはディレクトリ名のため正確な大文字と小文字が使用されていることを確認してください。例えば、ディレクトリ名が `/home/nikita/Downloads` の場合 `/home/nikita/downloads` はエラーを返します。
- Linux または macOS 上で `⌘` を入力する場合、Windows での `Windows` の使用とは異なり `⌘` を使用します。

## コマンドの実行方法

1. コマンドプロンプトウィンドウを開く
  - a. Windows 上でコマンドプロンプトを開くには **Windows** キーを押し、**cmd** の入力を開始します。key and then start 表示されるコマンドプロンプト 提案をクリックします。
  - b. Mac 上でターミナルを開き **ファイダー アイコン** をクリックし **移動 > ユーティリティ** をメニューから選択します。ターミナルアイコンをユーティリティウィンドウ内でダブルクリックします。
  - c. Linux をグラフィカルなユーザーインターフェイスから作動する場合、ターミナルコマンドを使用中のLinux の配布に適用できるようコケートし実行します。Linux をコマンドラインインターフェイスから実行する場合、このステップを無視してください。
2. 実行するコマンドが後に続く実行可能ファイルへのフル `⌘` を入力します。例えば、下のコマンドはコマンドラインでヘルプを提供します。

|         |   |
|---------|---|
| Linux   | <code>/opt/Altova/FlowForceServer2021/bin/flowforceserver help</code>                       |
| macOS   | <code>/usr/local/Altova/FlowForceServer2021/bin/flowforceserver help</code>                 |
| Windows | <code>C:\Program Files (x86)\Altova\FlowForceServer2021\bin\FlowForceServer.exe help</code> |

上のサンプルでは、コマンド `help` がオプションまたは引数無しに実行されています。他のコマンドは引数またはオプションを持っている場合があります。これらは引数とオプションは必須またはオプションです。各コマンドの詳細に関してはヘルプメッセージを参照してください。

## FlowForce Server をインストールディレクトリから呼び出す方法

実行可能ファイルをフル `⌘` を入力することなく呼び出すには、現在のディレクトリを FlowForce Server 実行可能ファイルがインストールされている個所に変更します。例:

|         |   |
|---------|---|
| Linux   | <code>cd /opt/Altova/FlowForceServer2021/bin</code>   |
| macOS   | <code>cd /usr/local/Altova/FlowForceServer2021/bin</code>                                   |
| Windows | <code>cd "C:\Program Files (x86)\Altova\FlowForceServer2021\bin\FlowForceServer.exe"</code> |

実行可能ファイル名のみを入力してコマンドを実行することができます。例:

|         |                                       |
|---------|---------------------------------------|
| Linux   | <code>./flowforceserver help</code>   |
| macOS   | <code>./flowforceserver help</code>   |
| Windows | <code>FlowForceServer.exe help</code> |

**メモ** Linux と macOS システム上では、プレフィックス `./` は実行可能ファイルが現在のディレクトリ内にあることを示しています。

## FlowForce Server をディレクトリから呼び出す方法

実行可能ファイルをディレクトリから呼び出すには絶対 `⌘` を使用して参照してください。または、プログラムを実行可能ファイル名を入力する

コマンドで呼び出す場合、使用中のオペレーティングシステムのPATH 環境変数を編集することにより、FlowForce Server インストールディレクトリへのパスが含まれるようになります。PATH 環境変数を変更する方法に関しては、使用中のオペレーティングシステムのドキュメンテーションを参照してください。

メモ PATH 環境変数の変更後、ターミナルウィンドウを閉じ、変更の効果が反映されるように新規のウィンドウを開く必要がある場合があります。

## 9.1 help

### 目的

引数として与えられたコマンドのためのヘルプを表示します。

### 構文

```
FlowForceServer help Command
```

**メモ** Linux システム上では、全て小文字のflowforceserver を使用して実行可能ファイルを呼び出します。

### 引数

ヘルプコマンドは単一の引数を取ります: ヘルプが必要なコマンド名。コマンドの正確な構文と正確なコマンドの実行に関連する他の情報を表示します。

### サンプル

```
FlowForceServer help createdb
```

### --help を他のコマンドのためのオプションとして使用する

コマンドに関するヘルプ情報は、そのコマンドの--help オプションで使用することができます。例えば、createdb コマンドを使用した--help オプションの使用方法は、以下の通りです:

```
FlowForceServer createdb --help
```

以下と同じ結果が得られます:

```
FlowForceServer help createdb
```

## 9.2 assignlicense

### 目的

Windows プラットフォームのみこのコマンドを適用することができます。ライセンスファイルを FlowForce Server にアップロードし割り当てるために使用されます。

### 構文

```
FlowForceServer assignlicense [options] FILE
```

### 引数

|      |                              |
|------|------------------------------|
| FILE | ライセンスファイルがアップロードされるパスを指定します。 |
|------|------------------------------|

### オプション

|                             |   |
|-----------------------------|---|
| --t, --test-only=true false | true に設定されていると、ライセンスはアップロードされ検証されます。<br><br>false に設定されていると、ライセンスはアップロード、検証、および割り当てられます。<br><br>このオプションが指定されていないと、デフォルトの値は true です。 |
|-----------------------------|---|

## 9.3 compactdb

### 目的

削除されたレコードを含む FlowForce .db ファイルのサイズを縮小します。このコマンドは、`archive-log` または `truncate-log` システム管理関数を実行した後とても役に立ちます。

### 構文

```
FlowForceServer compactdb [options]
```

**メモ** Linux システム上では、全て小文字の `flowforceserver` を使用して実行可能ファイルを呼び出します。

### オプション

`--datadir=VALUE`

VALUE は、縮小される .db ファイルのデータディレクトリのパスです。このオプションが指定されていない場合、/data ディレクトリがデフォルトで使用されます (次も参照してください: [FlowForce Server アプリケーションデータ](#))。

## 9.4 createdb

### 目的

新規データベースを作成します。データベースが既存の場合、コマンドは失敗します。デフォルトのデータベースはインストール時に作成され、通常このコマンドを使用する必要はありません。

### 構文

```
FlowForceServer createdb [options]
```

**メモ** Linux システム上では、全て小文字のflowforceserver を使用して実行可能ファイルを呼び出します。

### オプション

```
--datadir=VALUE
```

VALUE は、データディレクトリのパスです。

## 9.5 debug

### 目的

一般使用のためのコマンドではありません。このコマンドはFlowForce Server デバッグモードで開始します(つまり、サービスではなく開始します)。このモードを停止するには「CTRL+C」を押します。

### 構文

```
FlowForceServer debug [options]
```

**メモ** Linux システム上では、全て小文字のflowforceserver を使用して実行可能ファイルを呼び出します。

### オプション

|                              |                         |
|------------------------------|-------------------------|
| <code>--datadir=VALUE</code> | VALUE は、データディレクトリのパスです。 |
|------------------------------|-------------------------|

## 9.6 exportresourcestrings

### 目的

FlowForce Server のリソース文字列を含むXML ファイルを出力します。以下の引数を取ります: (i) 出力 XML ファイル内のリソース文字列の言語、および(ii) 出力 XML ファイルのパスと名前。有効なエクスポート言語は以下の通りです(言語コードはかっこの中です): 英語(en)、ドイツ語、(de)、スペイン語(es)、および日本語(ja)。

### 構文

```
FlowForceServer exportresourcestrings Language XMLOutput
```

**メモ** Linux システム上では、全て小文字のflowforceserver を使用して実行可能ファイルを呼び出します。

### 引数

|           |   |
|-----------|---|
| Language  | エクスポートされたXML ファイル内のリソース文字列の言語を指定します。<br>許可される言語: en, de, es, ja |
| XMLOutput | エクスポートされたXML ファイルの場所と名前を指定します。                                  |

### サンプル

このコマンドは、英語のFlowForce Server アプリケーションの全てのリソース文字列を含むc:\ にStrings.xml という名前のファイルを作成します。

```
FlowForceServer exportresourcestrings en c:\Strings.xml
```

## 9.7 foreground

### 目的

一般使用のためのコマンドではありません。このコマンドは、Altova FlowForce Server を前景で開始します。Linux のためのスタートアップスクリプトにより内部で使用されます。

## 9.8 initdb

### 目的

新規データベースを作成、または、既存のものを最新のバージョンにアップデートします。データベースが既存の場合、コマンドは失敗します。デフォルトのデータベースはインストール時に作成され、通常このコマンドを使用する必要はありません。

### 構文

```
FlowForceServer initdb [options]
```

**メモ** Linux システム上では、全て小文字の `flowforceserver` を使用して実行可能ファイルを呼び出します。

### オプション

|                              |                            |
|------------------------------|----------------------------|
| <code>--datadir=VALUE</code> | VALUE は、データベースディレクトリのパスです。 |
|------------------------------|----------------------------|

## 9.9 install

### 目的

このコマンドはFlowForce Server インストーラーにより自動的に実行され、一般の目的には使用することができません。このコマンドはAltova FlowForce Server をサービスとしてWindows にインストールします。このコマンドをLinux とmacOS に適用することはできません。

## 9.10 licenseserver

### 目的

FlowForceServer を LicenseServer に登録します。FlowForceServer を LicenseServer に登録するには、管理者の特権(ルート)が必要です。詳細に関しては、以下を参照してください。LicenseServer ドキュメント。

### 構文

```
FlowForceServer licenseserver [options] Server-or-IP-Address
```

**メモ** Linux システム上では、全て小文字の flowforceserver を使用して実行可能ファイルを呼び出します。

### サンプル

```
FlowForceServer licenseserver DOC.altova.com
```

上のコマンドは、DOC.altova.com という名前のマシンが Altova LicenseServer を動作しています。LicenseServer がユーザーのマシンで動作している場合、次のコマンドも使用することができます。

```
FlowForceServer licenseserver localhost
FlowForceServer licenseserver 127.0.0.1
```

### オプション

オプションは下にリストされており、短い書式(最初の列)と長い書式(2番目の列)はそれぞれ詳細を記載しています。コマンドライン上では、1つまたは2つのダッシュが短い、および、長い書式のために使用されています。

|     |        |  |
|-----|--------|--|
| --j | --json | JSON オブジェクトとして登録の結果を生成しようと試みます。<br>フォーム <code>--json=true false</code> |
|-----|--------|--|

## 9.11 migratedb

### 目的

前の[アプリケーションデータディレクトリ](#)からFlowForce Server データを現在のディレクトリにコピーし、FlowForce データベースを必要な場合最新バージョンに更新します。FlowForce Server の前のバージョンが既にインストールされている場合にFlowForce インストールスクリプトによりこのコマンドは呼び出され、通常実行する必要はありません。FlowForce Server を新しいマシンに移行する場合、またはアプリケーションデータディレクトリをバックアップから復元する場合、このコマンドの実行は役に立つかもしれませんが、[バックアップと復元](#)を参照してください。

FlowForce データベースバージョンを最新のバージョンに更新するのみが必要な場合、[upgradedb](#) を実行するだけで十分です。

### 構文

```
FlowForceServer migratedb [options]
```

**メモ** Linux システム上では、全て小文字のflowforceserver を使用して実行可能ファイルを呼び出します。

### オプション

|                    |                            |
|--------------------|----------------------------|
| --datadir=VALUE    | VALUE は、データベースディレクトリのパスです。 |
| --olddatadir=VALUE | VALUE は、データベースディレクトリのパスです。 |

### サンプル

FlowForce Server 2017 のアプリケーションデータディレクトリからデータを FlowForce Server 2021 に移行する場合、以下を実行します:

```
"C:\Program Files (x86)\Altova\FlowForceServer2021\bin\FlowForceServer.exe" migratedb  
--datadir=C:\ProgramData\Altova\FlowForceServer2021\data --olddatadir=C:  
\ProgramData\Altova\FlowForceServer2017\data
```

## 9.12 repair

### 目的

FlowForce Server を全てのトリガージョブの実行プロセスを無効化した状態で、トラブルシューティングを有効化するために開始します。

### 構文

```
FlowForceServer repair [options]
```

**メモ** Linux システム上では、全て小文字のflowforceserver を使用して実行可能ファイルを呼び出します。

### オプション

```
--datadir=VALUE
```

VALUE は、データベースディレクトリのパスです。

### サンプル

```
FlowForceServer repair --datadir=C:\ProgramData\Altova\FlowForceServer2021\data
```

## 9.13 resetpassword

### 目的

 **root** ユーザーのパスワードをデフォルトの値にリセットし、 **root** ユーザーに全ての特権を与えます。このオペレーションを開始する前に、実行中のFlowForce Server インスタンスを停止することが奨励されます。(以下を参照: [Linux](#)、[macOS](#)、および[Windows](#) 上でサービスを開始または停止するための命令)。

### 構文

```
FlowForceServer resetpassword [options]
```

**メモ** Linux システム上では、全て小文字のflowforceserver を使用して実行可能ファイルを呼び出します。

### オプション

```
--datadir=VALUE
```

VALUE は、データベースディレクトリのパスです。

### サンプル

```
FlowForceServer resetpassword --datadir=C:\ProgramData\Altova\FlowForceServer2021\data
```

## 9.14 setdeflang (sdl)

### 目的

setdeflang コマンド (略して sdl) は、FlowForce Server のデフォルトの言語を設定します。デフォルトの言語を変更するには、FlowForceServer と FlowForceWebServer サービスのためにこのコマンドを実行してください (以下を参照: 構文)。

### 構文

```
FlowForceServer setdeflang | sdl LanguageCode
FlowForceWebServer setdeflang | sdl LanguageCode
```

**メモ** Linux システム上では、全て小文字の flowforceserver を使用して実行可能ファイルを呼び出します。

LanguageCode の値には以下があります。

|    |       |
|----|-------|
| en | 英語    |
| es | スペイン語 |
| de | ドイツ語  |
| fr | フランス語 |
| ja | 日本語   |

### サンプル

```
FlowForceServer setdeflang de
```

## 9.15 start

### 目的

FlowForce Server をサービスとして開始します。このコマンドは、内部でスタートアップスクリプト、または、Windows サービスインストーラにより使用され、一般の使用のためではありません。

### 構文

```
FlowForceServer start [options]
```

**メモ** Linux システム上では、全て小文字の `flowforceserver` を使用して実行可能ファイルを呼び出します。

### オプション

|                              |                            |
|------------------------------|----------------------------|
| <code>--datadir=VALUE</code> | VALUE は、データベースディレクトリのパスです。 |
|------------------------------|----------------------------|

## 9.16 uninstall

### 目的

このコマンドはFlowForce Server インストーラーにより自動的に実行され、一般の目的には使用できません。このコマンドはAltova FlowForce Server をサービスとしてWindows にインストールします。このコマンドをLinux とmacOS に適用することはできません。

## 9.17 upgradedb

### 目的

データベースを最新のバージョンにアップグレードします。デフォルトのデータベースは自動的にインストール時にアップグレードされるため、手動でこのコマンドを使用する必要はありません。

### 構文

```
FlowForceServer upgradedb [options]
```

**メモ** Linux システム上では、全て小文字のflowforceserver を使用して実行可能ファイルを呼び出します。

### オプション

```
--datadir=VALUE
```

VALUE は データベースディレクトリのパスです。

### サンプル

```
FlowForceServer upgradedb --datadir=C:\ProgramData\Altova\FlowForceServer2021\data
```

## 9.18 verifylicense

### 目的

Windows プラットフォームのみこのコマンドを適用することができます。FlowForce Server にライセンスが与えられているか、与えられたライセンスキーが既にFlowForce Server に与えられているかを検証するために使用されます。

### 構文

```
FlowForceServer verifylicense [options]
```

### オプション

|                          |  |
|--------------------------|--|
| --l, --license-key=VALUE | このオプションにより、特定のライセンスキーが既にFlowForce Server に与えられているかを検証することができます。<br><br>検証するライセンスキーに値が設定されている必要があります。 |
|--------------------------|--|

## 10 ビルトイン関数

このセクションでは、FlowForce Server に内蔵されているシステム関数について説明しています。ビルトイン関数を使用することにより、ファイルをコピー、移動し、ディレクトリを作成し、シェルコマンドを実行するなど多種のアクションを実行することができます。FlowForce Server ビルトイン関数は、`/system` コンテナ内で使用することができます。

以下のピックは、ルートコンテナに相対したパスに従ってグループ化されたビルトイン関数について説明しています。

- [/system/abort](#)
- [/system/compute](#)
- [/system/compute-string](#)
- [/system/filesystem](#)
- [/system/ftp](#)
- [/system/mail](#)
- [/system/maintenance](#)
- [/system/shell](#)

RaptorXML Server がFlowForce Server に統合されている場合、すべてのRaptorXML Server 関数を含む追加コンテナを使用することができます。これはRaptorXML+XBRL Server にも同様に適用されます。詳細に関しては [RaptorXML Server との統合](#) を参照してください。

### Windows ネットワークパスの参照

ジョブを作成する際、通常 FlowForce Server が作動するマシン上で、または、ネットワーク上のファイルパスを参照します。マップされたネットワークドライブなどの Windows ネットワークパスを参照する場合、Universal Naming Convention (UNC) 構文を使用します。これは、ドライブ文字はシステムに対してグローバルではないため、また、各ログインセッションは自身のドライブ文字に割り当てられているため、これはドライブ文字がシステムに対してグローバルではなく、ログインセッションが自身のドライブ文字に割り当てられているため必須です。

UNC 構文は以下のフォームをとります:

```
\\server\sharedfolder\filepath
```

該当する箇所:

- `server` が (DNS により定義される) ネットワーク内のサーバー名を参照する箇所。
- `sharedfolder` が管理者により定義されるラベルを参照する箇所 (例えば、`admin$` が通常オペレーティングインストールのルートディレクトリである箇所)。
- `filepath` がシェアの下のサブディレクトリを参照する箇所。

## 10.1 /system

**/system** コンテナには全てのFlowForce ビルトイン関数が含まれています。abort、compute、およびcompute-string 関数のみは直接このコンテナで見つけることができます。(例えば、AS2 関数、ファイルシステム関数、など)他の関数は適用できるエリア別にサブコンテナに分別されています。

### 10.1.1 abort

フルパス: /system/abort

ジョブの実行を中断します。ジョブの実行を中断します。この関数は、通常条件(つまり、[選択ステップ](#))内で条件が満たされた場合、ジョブを意図的に中断するために使用されます。プログラム言語内のthrow またはraise 関数と同じです。この関数は、値を返しません。

#### パラメーター

| 名前    | 型      | 説明   |
|-------|--------|--|
| メッセージ | string | 必須の文字列パラメーターです。ジョブを停止する際に出力されるメッセージを指定します。 |

#### サンプル

次のジョブでは、abort 関数は、チェックされた値が10個のアイテムを超えると、ジョブがエラーを返して完了するために使用されます。アイテムの数が、10個より少ない、または、10個の場合、ジョブがリストのアイテムは10個より少ないです」というテキストをローカルシステム上のファイルに書き込みます。

### 実行ステップ

- 関数の実行 /system/abort  
for メッセージと。
- For each item シーケンスで list(1,2,3,4,5,6,7,8,9,10,11)
  - 選択
    - When item > 10
      - 関数の実行 /system/abort
        - パラメーター:
          - メッセージ: リストが10個のアイテムを超えるため中断します
      - = ステップの結果を割り当て 名前
    - When item < +10
      - 関数の実行 /system/shell/commandline
        - パラメーター:
          - コマンド: echo"ファイルには10個より少ないアイテムが存在します">File.txt
          - エラー時には中断: +
          - 作業用ディレクトリ: c:\temp
      - = ステップの結果を割り当て 名前 以下として result
    - 新規 When
    - Otherwise
      - +
        - = ステップの結果を割り当て 名前
- = ステップの結果を割り当て 名前

## 10.1.2 compute

フルパス: /system/compute

式の結果を計算し、計算された値を返します。計算された値は、パラメータ内、または他の実行ステップの式内で使用することができます。サービスとして使用されるジョブの出力を定義するためにこの関数を使用することができます(例を参照)。

この関数は、任意の型を示す値 **T0** を返します。つまり、返されたデータ型は、式パラメータ内で使用された式から推定されます。つまり、返されたデータ型は、式パラメータ内で使用された式から推定されます。

### パラメータ

| 名前 | 型                | 説明  |
|----|------------------|---|
| 式  | Expression of T0 | 計算される FlowForce Server 式。式の詳細に関しては <a href="#">The FlowForce 式言語</a> を参照してください。 |

### サンプル

このサンプルは、2つの実行ステップを使用するジョブに関する説明をします。最初のステップは、**c:\temp** ディレクトリ内のシェルコマンドを実行、結果は **hello** として宣言されています。

次に、この結果は2番目の実行ステップに渡されます。2番目の実行ステップは式言語（特に、**stdout** と **content** 関数）を使用して以下を行います：

- 最初のステップの結果の標準出力を取得します。
- 出力を文字列に変換します。

**compute** 関数は、Expression 式テキストボックスに入力された式を評価します。the text box.

## 実行ステップ

関数の実行ステップは削除されました

**+**

関数の実行 `/system/shell/commandline`

パラメーター:

|            |   |
|------------|---|
| コマンド:      | <input type="text" value='echo "Hello world"'/>   |
| エラー時には中断:  | <input style="border: none; background-color: #eee; padding: 2px 10px;" type="button" value="+"/> |
| 作業用ディレクトリ: | <input type="text" value="c:\temp"/>  |

= ステップの結果を割り当て  以下として **result**

**+**

関数の実行 `/system/compute`

パラメーター:

|      |   |
|------|---|
| 条件式: | <input type="text" value="content(stdout(hello))"/> |
|------|---|

= ステップの結果を割り当て  以下として **T0**

["Hello, World!" ジョブの作成](#)を参照してください。

### 10.1.3 compute-string

フルパス: /system/compute-string

式の結果を文字列として出力します。このステップ関数は、フォーマットが式ではなく文字列である以外、compute 関数と基本的に同じ処理を行います。

#### パラメーター

| 名前 | 型      | 説明                                |
|----|--------|-----------------------------------|
| 式  | string | 計算される(文字列としての) FlowForce Server 式 |

#### サンプル

/system/compute/ と /system/compute-string 関数の違いを理解するには、以下のサンプルに留意してください！

## ジョブ入力パラメーター

名前:  型:  デフォルト:  説明:

## 実行ステップ

関数の実行

パラメーター: 条件式: `concat(filename(inputname), 'txt')`

= ステップの結果を割り当て  以下として T0

関数の実行

パラメーター: 条件式: `{filename(inputname)}.tst`

= ステップの結果を割り当て  以下として string

関数の実行

パラメーター: 条件式:

= ステップの結果を割り当て  以下として string

関数の実行

パラメーター: 条件式: `Both expressions are {if(outputname1==outputname2, 'not')} identical`

= ステップの結果を割り当て  以下として string

上記のジョブでは、3つの実行ステップが扱われます。

最初のステップは、`/system/compute/` 関数を呼び出します。中かっこが使用されていないことにご注意ください。(背景色により示されるようにフィールド全体が式を保管する場合、中かっこを示します。式は、2つの値を連結し、ジョブ入力パラメーターにより文字列を作成します。例えば、入力パラメーターが `"c:\temp\invoices.txt"` の場合、ステップは、(outputname1 と宣言された)文字列の値 `"invoices.txt"` を返します。

2番目のステップは、`/system/compute-string` 関数を呼び出します。この関数は、埋め込まれた FlowForce 式を含む文字列を処理します。ここでは、中かっこは、文字列の残りから式を区別するために使用されています。埋め込まれた式は、文字列の残りの部分と背景色が異なることにご注意ください。異なる技術が使用されていますが、ステップの結果 (outputname2) は outputname1 と同じです。

最後に3番目のステップは、outputname1 と outputname2 を比較するために /system/compute-string 関数をもう1度呼び出します。両方の値が同じの場合、結果は文字列の値「両方の式は同等です」となります。それ以外の場合、結果は「両方の式は同等ではありません」となります。

## 10.2 /system/filesystem

/system/filesystem コンテナは、FlowForce Server が作動するオペレーティングシステム上のファイルとディレクトリを管理するために使用される関数を含んでいます。

ジョブの実行内のファイルパスは、使用中のローカルマシンではなく、FlowForce Server が作動するオペレーティングシステム上のパスである必要があります。

### 10.2.1 copy

フルパス: /system/filesystem/copy

ソースからターゲットディレクトリにファイルをコピーします。任意で、ファイルを新しい名前と共にターゲットディレクトリにコピーすることができます。

シンプルな実行ステップから呼び出された場合、この関数は1つのファイルをその都度コピーします。FlowForce を使用して複数のファイルをコピーする場合、copy 関数を呼び出すステップをステップ内で [ファイルのコピー](#) サンプル内で説明されているとおり、囲んでください。

実行に成功すると、この関数はブール値の TRUE を返します。ジョブの実行に失敗すると、結果は「エラー時には中断」/パラメータの値により以下のように異なります:

- 「エラー時には中断」/パラメータが TRUE (デフォルトの値) の場合、ジョブの実行は中断されます。この場合、エラーを保護されたブロックを使用して処理することができます (以下を参照 [ステップのエラーの処理](#))。
- 「エラー時には中断」/パラメータが FALSE の場合、関数は FALSE を返します。

#### パラメーター

| 名前         | 型                   | 説明  |
|------------|---------------------|---|
| ソース        | string as file      | コピーするソースファイルのファイル名とパス。  |
| ターゲット      | string as file      | 保存先ディレクトリのパスとファイル名。ファイルの名前を変更する場合、保存先フィールドに異なる名前を入力することもできます。   |
| 上書き        | boolean             | true の場合は、保存先ファイルが上書きされます。デフォルトの値は false です。  |
| エラー時には中断する | boolean             | このブールパラメータはジョブが失敗した場合の戻り値を決定します。「エラー時には中断」が FALSE の場合、関数はブール値の FALSE を返します。「エラー時には中断」が TRUE の場合、ジョブの実行は中断されます。デフォルトの値は TRUE です。 |
| 作業ディレクトリ   | string as directory | 作業ディレクトリを指定します (例えば c:\somedirectory)。相対的なパスが使用されると、作業ディレクトリに対して解決されます。   |

## サンプル

[ファイルのエラー](#)を参照

## 10.2.2 delete

フルパス: /system/filesystem/delete

引数として与えられたパスからファイルを削除します。

シンプルな実行ステップから呼び出された場合、この関数は1つのファイルをその都度削除します。FlowForce を使用して複数のファイルを削除する場合、delete 関数を呼び出すステップを「for-each」ステップ内で [ファイルのエラー](#) を参照するサンプル内で説明されているとおり、囲んでください。

実行に成功すると、この関数はブール値の TRUE を返します。ジョブの実行に失敗すると、結果は「エラー時には中断」/パラメータの値により以下のように異なります:

- 「エラー時には中断」/パラメータが TRUE (デフォルトの値) の場合、ジョブの実行は中断されます。この場合、エラーを保護されたブロックを使用して処理することができます(以下を参照: [ステップのエラーの処理](#))。
- 「エラー時には中断」/パラメータが FALSE の場合、関数は FALSE を返します。

## パラメータ

| 名前         | 型                   | 説明  |
|------------|---------------------|---|
| パス         | string as directory | 削除するファイルのファイル名とパス。  |
| エラー時には中断する | boolean             | このブールパラメータはジョブが失敗した場合の戻り値を決定します。「エラー時には中断」が FALSE の場合、関数はブール値の FALSE を返します。「エラー時には中断」が TRUE の場合、ジョブの実行は中断されます。デフォルトの値は TRUE です。 |
| 作業ディレクトリ   | string as directory | 作業ディレクトリを指定します(例えば c:\somedirectory)。相対的なパスが使用されると、作業ディレクトリに対して解決されます。  |

## 10.2.3 mkdir

フルパス: /system/filesystem/mkdir

指定されたパスにディレクトリを作成します。

実行に成功すると、この関数はブール値の TRUE を返します。ジョブの実行に失敗すると、結果は「エラー時には中断」/パラメータの値により以下のように異なります:

- 「エラー時には中断」/パラメータが TRUE (デフォルトの値) の場合、ジョブの実行は中断されます。この場合、エラーを保護

さいげブロックを使用して処理することができます (以下を参照: [ステップのエラーの処理](#))。

- 「エラー時には中断」/パラメータが FALSE の場合、関数は FALSE を返します。

### パラメータ

| 名前         | 型                   | 説明   |
|------------|---------------------|--|
| パス         | string as directory | 新規ディレクトリのパス。   |
| 親にする       | boolean             | チェックボックスを選択して、アクティブ化することにより、c:\dir1\dir2\dir3 のような階層パスが1つのステップで作成されます。   |
| エラー時には中断する | boolean             | このブールパラメータはジョブが失敗した場合の戻り値を決定します。「エラー時には中断」が FALSE の場合、関数は、ブール値の FALSE を返します。「エラー時には中断」が TRUE の場合、ジョブの実行は中断されます。デフォルトの値は TRUE です。 |
| 作業ディレクトリ   | string as directory | 作業ディレクトリを指定します (例えば、c:\somedirectory)。相対的なパスが使用されると、作業ディレクトリに対して解決されます。  |

### サンプル

作業ディレクトリが c:\temp、パスが temp2\temp3 の場合、関数は、新規ディレクトリ c:\temp\temp2\temp3 を作成します。

## 10.2.4 move

フルパス: /system/filesystem/move

ファイルを移動またはファイルの名前を変更します。

シンプルな実行ステップから呼び出された場合、この関数は1つのファイルをその都度移動、または、名前を変更します。FlowForce を使用して複数のファイルを移動する場合、move 関数を呼び出すステップを「for-each」ステップ内で [ファイルのエラー](#) サンプル内で説明されているとおり、囲んでください。

実行に成功すると、この関数はブール値の TRUE を返します。ジョブの実行に失敗すると、結果は「エラー時には中断」/パラメータの値により以下のように異なります:

- 「エラー時には中断」/パラメータが TRUE (デフォルトの値) の場合、ジョブの実行は中断されます。この場合、エラーを保護さいげブロックを使用して処理することができます (以下を参照: [ステップのエラーの処理](#))。
- 「エラー時には中断」/パラメータが FALSE の場合、関数は FALSE を返します。

## パラメーター

| 名前         | 型                   | 説明  |
|------------|---------------------|---|
| ソース        | string as file      | 移動するソースファイルのパスとファイル名。   |
| 保存先        | string as file      | 保存先ディレクトリの名前。このフィールドにディレクトリ名のみが与えられている場合、元の名前が保持されます。   |
| ターゲットの上書き  | boolean             | true の場合は、保存先ファイルが上書きされます。デフォルトの値は false です。  |
| エラー時には中断する | boolean             | このブールパラメーターはジョブが失敗した場合の戻り値を決定します。「エラー時には中断」が FALSE の場合、関数は、ブール値の FALSE を返します。「エラー時には中断」が TRUE の場合、ジョブの実行は中断されます。デフォルトの値は、TRUE です。 |
| 作業ディレクトリ   | string as directory | 作業ディレクトリを指定します(例えば、c:\somedirectory)。相対的なパスが使用されると、作業ディレクトリに対して解決されます。  |

## 10.2.5 rmdir

フルパス: /System/filesystem/rmdir

ディレクトリの削除。

実行に成功すると、この関数はブール値の TRUE を返します。ジョブの実行に失敗すると、結果は「エラー時には中断」/パラメーターの値により以下のように異なります:

- 「エラー時には中断」/パラメーターが TRUE (デフォルトの値) の場合、ジョブの実行は中断されます。この場合、エラーを保護されたブロックを使用して処理することができます(以下を参照: [ステップのエラーの処理](#))。
- 「エラー時には中断」/パラメーターが FALSE の場合、関数は、FALSE を返します。

## パラメーター

| 名前         | 型                   | 説明  |
|------------|---------------------|---|
| パス         | string as directory | 削除するファイルのファイル名。   |
| エラー時には中断する | boolean             | このブールパラメーターはジョブが失敗した場合の戻り値を決定します。「エラー時には中断」が FALSE の場合、関数は、ブール値の FALSE を返します。「エラー時には中断」が TRUE の場合、ジョブの実行は中断されます。デフォルトの値は、 |

| 名前       | 型                                | 説明   |
|----------|----------------------------------|--|
|          |                                  | TRUE です。   |
| 作業ディレクトリ | <code>string as directory</code> | 作業ディレクトリを指定します(例えば <code>c:\somedirectory</code> )。相対的なパスが使用されると、作業ディレクトリに対して解決されます。 |

## 10.3 /system/ftp

/system/ftp コンテナにはFTP またはFTPS サーバーに接続し、ファイルをアップロード、抽出、削除、またはリモートディレクトリ他を作成または削除するために使用される関数が含まれています。

### FTP 関数内のパス

FTP 関数の一部には作業ディレクトリ/パラメータが存在します。このパラメータはFlowForce 関数の多くで共通です。作業ディレクトリを設定するには、FTP 関数のパラメータとして提供されるすべてのローカルパスが絶対的ではなく相対的であることを注意してください。ファイルをアップロードするには作業ディレクトリとして C:\Upload とソースファイルとして file.txt を入力することができます。最後のノースパスは C:\Upload\file.txt になります。

更に FTP 関数の一部はリモート FTP サーバー上で作業ディレクトリと同じ役割を果たすホスト上のディレクトリ/パラメータを有します。このためにホスト上のディレクトリを設定するには、FTP 関数のパラメータとして提供されるすべてのローカルパスが絶対的ではなく相対的であることを注意してください。例えば、ファイルをアップロードするにはホスト上のディレクトリとして uploads とターゲットファイルとして file.txt を入力することができます。最後のターゲットパスは /uploads/file.csv になります。

[ファイルシステムトリガー](#) または [HTTP トリガー](#) としてジョブを構成することも重要です。このようなジョブジョブをトリガーしたファイルのパスを提供する triggerfile パラメータを有します。FTP 関数内で triggerfile パラメータを使用する場合、そのパスは絶対的であることを忘れないでください。

triggerfile から拡張子を持つファイル名を取得するには、次のFlowForce 式を使用してください！

```
{filename-with-extension(triggerfile)}
```

サンプルに関してはFTP [store](#) 関数を参照してください。

### FTP 関数内のワイルドカード

以下のFTP 関数はワイルドカードをパラメータとして受け入れます：

- [/system/ftp/delete-wildcard](#)
- [/system/ftp/retrieve-wildcard](#)
- [/system/ftp/store-wildcard](#)

このような関数を使用する場合、専用のワイルドカード/パラメータ内でワイルドカードを入力します。使用可能なワイルドカードは以下の通りです：

| ワイルドカード | 使用法                 | サンプル                                    |
|---------|---------------------|---|
| *       | ゼロまたはより多くの文字を一致します。 | *.htm は home.htm と index.htm を一致します。    |
| ?       | 単一の文字を一致する。         | *.xm? は index.xml と project.xml に一致します。 |

ワイルドカード + (1つまたは複数) はサポートされています。代わりに?\* を使用して同じ効果を得ることができます。例えば \*.c?\* は .cs 、 .cp と.csproj ファイルに一致しますが、.c ファイルには一致しません。

### 10.3.1 delete

フルパス: /system/ftp/delete

FTP サーバーからファイルを削除します。

実行に成功すると、この関数はブール値の TRUE を返します。ジョブの実行に失敗すると、結果は「エラー時には中断」/パラメータの値により以下のように異なります:

- 「エラー時には中断」/パラメータが TRUE (デフォルトの値) の場合、ジョブの実行は中断されます。この場合、エラーを保護されたブロックを使用して処理することができます (以下を参照: [ステップのエラーの処理](#))。
- 「エラー時には中断」/パラメータが FALSE の場合、関数は FALSE を返します。

FTP 関数内で `triggerfile` /パラメータを使用する場合、その値は絶対的であることを忘れてください。サンプルに関しては FTP [store](#) 関数を参照してください。

#### パラメータ

| 名前             | 型          | 説明   |
|----------------|------------|--|
| FTP サーバー       | string     | URL または IP アドレスとしてのホスト FTP サーバーのアドレス<br><br>必須のパラメータです。  |
| ポート            | number     | FTP サーバーに接続するために使用されるポート番号です。デフォルトの値は 21 です。   |
| ホスト側のディレクトリ    | string     | ファイルを削除するホスト上のディレクトリの名前です。<br><br>任意のパラメータです。  |
| ログインの資格情報      | credential | FlowForce 資格情報レコードとしての FTP アカウントのユーザー名とパスワードです。 <a href="#">資格情報</a> を参照してください。<br><br>FTP サーバーが親覚情報を必要としない場合、このパラメータをスキップします。 |
| 受動モードを使用する     | boolean    | 接続エラーが発生すると、パッシブモードを使用します (例えば、ルーターまたはファイアウォールがアクティブな接続を回避するように設定されている場合)。   |
| SSL/TLS 暗号化の使用 | string     | (任意のパラメータです。デフォルトの値はいいえです。) 情報を転送する際は FTP はコマンドチャンネルとデータチャンネルを使用します。FTP データを暗号化無しに転送する場合、この値をいいえに設定します。それ以外の場合、この値を以下に設定します:   |

| 名前         | 型                        | 説明   |
|------------|--------------------------|--|
|            |                          | <ul style="list-style-type: none"> <li>暗号化されたコマンドチャンネルと共に明示的</li> <li>暗号化されたコマンドとデータチャンネルと共に明示的</li> </ul> <p>上の2つのオプションのいずれかを設定すると証明書は下で説明されているサーバー証明書の検証 / パラメータに従い検証されます。</p> <p>メモ 明示的な暗号化は奨励されず、FlowForce 内でサポートされていません。</p>  |
| サーバー証明書の検証 | <code>string</code>      | <p>任意のパラメータです。FlowForce などのようにFTP サーバーの証明書を検証するため指定します。有効な値</p> <ul style="list-style-type: none"> <li>検証無し - 全ての証明書を受け入れます。</li> <li>証明書ストアに対して検証します (デフォルトの値) - Windows でジョブを実行中のユーザーアカウントの証明書ストアとシステムストアを証明書署名を検証するために使用してください。Linux では、システム証明書ストアを使用します。これは通常 <code>/usr/lib/ssl/cert.pem</code> と <code>/usr/lib/ssl/certs</code> または <code>SSL_CERT_FILE</code> と <code>SSL_CERT_DIR</code> 環境関数がポイントするパスにあります。</li> <li>サーバー証明書に対して検証 - FlowForce はサーバー証明書 / パラメータ内で指定されているFTP サーバーの証明書と比較します。</li> </ul> <p>このパラメータの使用はサーバー証明書と安全な接続が必要です。安全な接続を構築できない場合、FTP 関数は失敗します。</p> |
| サーバーの証明書   | <code>certificate</code> | <p>任意のパラメータです。FlowForce 内の証明書オブジェクトへのパスを指定します。前のパラメータを選択されたサーバー証明書に対して検証に設定している場合、指定されているFlowForce 証明書はFTP サーバー証明書に対して検証されます。それ以外の場合、このパラメータ値は</p>   |

| 名前         | 型       | 説明   |
|------------|---------|--|
|            |         | 無視されます。  |
| ターゲットファイル  | string  | サーバーから削除されるファイル名です。<br>必須のパラメータです。   |
| エラー時には中断する | boolean | このブールパラメータはジョブが失敗した場合の戻り値を決定します。「エラー時には中断」がFALSE の場合、関数は、ブール値のFALSE を返します。「エラー時には中断」がTRUE の場合、ジョブの実行は中断されます。デフォルトの値はTRUE です。 |
| アカウント      | string  | リモートサーバー上でアクセスすることのできるユーザーが許可されているFTP アカウント名です。<br>任意のパラメータです。   |

### 10.3.2 delete-wildcard

フルパス /system/ftp/delete-wildcard

FTP サーバーからワイルドカードに一致するファイルを削除します。例、\*.xml。成功すると、関数は削除されたファイルのリスト（/ の無いファイル名）を返すまたは、一致が見つかれば空のリストを返します。実行に失敗すると、結果は以下で説明されているエラー時に中断 / パラメータにより異なります。

FTP 関数内で **triggerfile** / パラメータを使用する場合、そのパスは絶対的であることを忘れてはいけません。サンプルに関してはFTP [store](#) 関数を参照してください。

#### パラメータ

| 名前          | 型          | 説明  |
|-------------|------------|---|
| FTP サーバー    | string     | リモート FTP サーバーのアドレス。URL またはIP アドレスのいずれかです。<br>必須のパラメータです。    |
| ポート         | number     | FTP サーバーに接続するために使用されるポート番号デフォルトの値は21 です。                    |
| ホスト上のディレクトリ | string     | ファイルを削除するホスト上のディレクトリの名前<br>任意のパラメータです。                      |
| ログイン資格情報    | credential | FlowForce 資格情報レコードとしての FTP アカウントのユーザー名とパスワード。資格情報を参照してください。 |

| 名前          | 型       | 説明  |
|-------------|---------|---|
|             |         | FTP サーバーが資格情報を必要としない場合、このパラメータをスキップしてください。  |
| 受動モードの使用    | boolean | 接続問題が発生すると受動モードを使用してください(例えば、ルーターまたはファイアウォールがアクティブな接続を回避するようにセトアップされている場合)。   |
| SSL/TLS の使用 | string  | <p>(任意のパラメータです。デフォルトの値はいいえです。) 情報を転送するときはFTP はコマンドチャンネルとデータチャンネルを使用します。FTP データを暗号化無しで転送する場合、この値をいいえに設定します。それ以外の場合、この値を以下に設定します:</p> <ul style="list-style-type: none"> <li>暗号化されたコマンドチャンネルと共に明示的</li> <li>暗号化されたコマンドとデータチャンネルと共に明示的</li> </ul> <p>上の2つのオプションのいずれかを設定すると証明書は下で説明されているサーバー証明書の検証 / パラメータに従って検証されます。</p> <p>メモ 明示的な暗号化は奨励されず、FlowForce 内でサポートされていません。</p>                              |
| サーバー証明書の検証  | string  | <p>任意のパラメータです。FlowForce などのようにFTP サーバーの証明書を検証するか指定します。有効な値:</p> <ul style="list-style-type: none"> <li>検証無し - 全ての証明書を受け入れます。</li> <li>証明書ストアに対して検証します (デフォルトの値) - Windows でジョブを実行中のユーザーアカウントの証明書ストアとシステムストアを証明書署名を検証するために使用してください。Linux では、システム証明書ストアを使用します。これは通常 /usr/lib/ssl/cert.pem と /usr/lib/ssl/certs または SSL_CERT_FILE と SSL_CERT_DIR 環境関数がポイントするパスにあります。</li> <li>サーバー証明書に対して検証</li> </ul> |

| 名前         | 型                        | 説明   |
|------------|--------------------------|--|
|            |                          | <p>- FlowForce はサーバー証明書 / ラメーター内で指定されている FTP サーバーの証明書と比較します。</p> <p>このラメーターの使用はサーバー証明書と安全な接続が必要です。安全な接続を構築できない場合、FTP 関数は失敗します。</p>   |
| サーバーの証明書   | <code>certificate</code> | <p>任意のラメーターです。FlowForce 内の証明書オブジェクトへのパスを指定します。前のラメーターを選択されたサーバー証明書に対して検証に設定している場合、指定されている FlowForce 証明書は FTP サーバー証明書に対して検証されません。それ以外の場合、このラメーター値は無視されます。</p>   |
| ワイルドカード    | <code>string</code>      | <p>必須のラメーターです。ワイルドカードを指定します。例、<code>*.xml</code>。ワイルドカードに一致するファイルが削除されます。<a href="#">FTP 関数内のワイルドカード</a>を参照してください。</p>   |
| エラー時には中断する | <code>boolean</code>     | <p>実行に失敗するとこのラメーターは関数の振る舞いを指定します。具体的には、実行が失敗した場合、関数は以下の1つを返します:</p> <ul style="list-style-type: none"> <li>エラー時に中断 / ラメーターが <code>false</code> の場合、削除されたファイルのリストは返されません。</li> <li>エラー時に中断 / ラメーターが <code>true</code> の場合、失敗が発生するまで関数はファイルを削除し、実行を中断します。</li> </ul> <p>このため、ファイルの一部は実行が失敗しても削除される場合があります。</p> |
| アカウント      | <code>string</code>      | <p>リモートサーバー上のファイルにアクセスを許可されているユーザーの FTP アカウント名</p> <p>任意のラメーターです。</p>  |

### 10.3.3 mkdir

フルパス: /system/ftp/rmdir

FTP サーバー上にディレクトリを作成します。

実行に成功すると、この関数はブール値の TRUE を返します。ジョブの実行に失敗すると、結果は「エラー時には中断」/パラメータの値により以下のように異なります:

- 「エラー時には中断」/パラメータが TRUE (デフォルトの値) の場合、ジョブの実行は中断されます。この場合、エラーを保護されたブロックを使用して処理することができます (以下を参照: [ステップのエラーの処理](#))。
- 「エラー時には中断」/パラメータが FALSE の場合、関数は FALSE を返します。

FTP 関数内で `triggerfile` /パラメータを使用する場合、その値は絶対的であることを忘れてはいけません。サンプルに関しては FTP [store](#) 関数を参照してください。

## パラメータ

| 名前             | 型          | 説明  |
|----------------|------------|---|
| FTP サーバー       | string     | URL または IP アドレスとしてのホスト FTP サーバーのアドレス<br><br>必須のパラメータです。   |
| ポート            | number     | FTP サーバーに接続するために使用されるポート番号です。デフォルトの値は 21 です。  |
| ホスト側のディレクトリ    | string     | ホスト上で新規のディレクトリを作成するディレクトリ名です。<br><br>任意のパラメータです。  |
| ログインの資格情報      | credential | FlowForce 資格情報レコードとしての FTP アカウントのユーザー名とパスワードです。資格情報を参照してください。<br><br>FTP サーバーが親覚情報を必要としない場合、このパラメータをスキップします。  |
| 受動モードを使用する     | boolean    | 接続エラーが発生すると、パッシブモードを使用します (例えば、ルーターまたはファイアウォールがアクティブな接続を回避するように設定されている場合)。  |
| SSL/TLS 暗号化の使用 | string     | (任意のパラメータです。デフォルトの値はいいえです。) 情報を転送するには FTP はコマンドチャンネルとデータチャンネルを使用します。FTP データを暗号化無しに転送する場合、この値をいいえに設定します。それ以外の場合、この値を以下に設定します: <ul style="list-style-type: none"> <li>• 暗号化されたコマンドチャンネルと共に明示的</li> <li>• 暗号化されたコマンドとデータチャンネルと共に明示的</li> </ul> 上の2つのオプションのいずれかを設定すると |

| 名前          | 型           | 説明  |
|-------------|-------------|---|
|             |             | <p>証明書は下で説明されているサーバー証明書の検証 / パラメータに依り検証されず。</p> <p>メモ 明示的な暗号化は奨励されず、FlowForce 内でサポートされていません。</p>  |
| サーバー証明書の検証  | string      | <p>任意のパラメータです。FlowForce などのようにFTP サーバーの証明書を検証するか指定します。有効な値</p> <ul style="list-style-type: none"> <li>• 検証無し - 全ての証明書を受け入れます。</li> <li>• 証明書ストアに対して検証します (デフォルトの値) - Windows ではジョブを実行中のユーザーアカウントの証明書ストアとシステムストアを証明書署名を検証するために使用してください。Linux では、システム証明書ストアを使用します。これらは通常 <code>/usr/lib/ssl/cert.pem</code> と <code>/usr/lib/ssl/certs</code> または <code>SSL_CERT_FILE</code> と <code>SSL_CERT_DIR</code> 環境関数がポイントするパスがあります。</li> <li>• サーバー証明書に対して検証 - FlowForce はサーバー証明書 / パラメータ内で指定されているFTP サーバーの証明書と比較します。</li> </ul> <p>このパラメータの使用はサーバー証明書と安全な接続が必要です。安全な接続を構築できない場合、FTP 関数は失敗します。</p> |
| サーバーの証明書    | certificate | <p>任意のパラメータです。FlowForce 内の証明書オブジェクトへのパスを指定します。前のパラメータを選択されたサーバー証明書に対して検証に設定している場合、指定されているFlowForce 証明書はFTP サーバー証明書に対して検証されます。それ以外の場合、このパラメータ値は無視されます。</p>   |
| ターゲットディレクトリ | string      | <p>作成するディレクトリ名です。</p> <p>必須のパラメータです。</p>  |
| エラー時には中断する  | boolean     | <p>このグローバルパラメータはジョブが失敗した場合の戻り値を決定します。「エラー時には</p>  |

| 名前    | 型      | 説明   |
|-------|--------|--|
|       |        | 中断」がFALSE の場合、関数は、ブール値のFALSE を返します。「エラー時には中断」がTRUE の場合、ジョブの実行は中断されます。デフォルトの値は TRUE です。 |
| アカウント | string | リモートサーバー上でアクセスすることのできるユーザーが許可されているFTP アカウント名です。<br><br>任意のパラメータです。                     |

### 10.3.4 move

フルパス: /system/ftp/retrieve

FTP サーバーからファイルを移動します。

実行に成功すると、この関数はブール値のTRUE を返します。ジョブの実行に失敗すると、結果は「エラー時には中断」/パラメータの値により以下のように異なります:

- 「エラー時には中断」/パラメータが TRUE (デフォルトの値) の場合、ジョブの実行は中断されます。この場合、エラーを保護されたブロックを使用して処理することができます (以下を参照: [ステップのエラーの処理](#))。
- 「エラー時には中断」/パラメータが FALSE の場合、関数は FALSE を返します。

FTP 関数内で `triggerfile` /パラメータを使用する場合、その値は絶対的であることを忘れてはいけません。サンプルに関してはFTP `store` 関数を参照してください。

#### パラメータ

| 名前          | 型          | 説明  |
|-------------|------------|---|
| FTP サーバー    | string     | URL またはIP アドレスとしてのホスト FTP サーバーのアドレス<br><br>必須のパラメータです。        |
| ポート         | number     | FTP サーバーに接続するために使用されるポート番号です。デフォルトの値は21です。                    |
| ホスト側のディレクトリ | string     | ファイルを移動するホスト上のディレクトリの名前です。<br><br>任意のパラメータです。                 |
| ログインの資格情報   | credential | FlowForce 資格情報レコードとしての FTP アカウントのユーザー名とパスワードです。資格情報を参照してください。 |

| 名前             | 型       | 説明  |
|----------------|---------|---|
|                |         | FTP サーバーが親覚情報を必要としない場合、このパラメータをスキップします。   |
| 受動モードを使用する     | boolean | 接続エラーが発生すると、パッシブモードを使用します(例えば、ルーターまたはファイアウォールがアクティブな接続を回避するように設定されている場合)。   |
| SSL/TLS 暗号化の使用 | string  | <p>(任意のパラメータです。デフォルトの値はいいえです。) 情報を転送するにはFTP はコマンドチャンネルとデータチャンネルを使用します。FTP データを暗号化無しに転送する場合、この値をいいえに設定します。それ以外の場合、この値を以下に設定します:</p> <ul style="list-style-type: none"> <li>暗号化されたコマンドチャンネルと共に明示的</li> <li>暗号化されたコマンドとデータチャンネルと共に明示的</li> </ul> <p>上の2つのオプションのいずれかを設定すると証明書は下で説明されているサーバー証明書の検証 / パラメータに従って検証されます。</p> <p>メモ 明示的な暗号化は奨励されず、FlowForce 内でサポートされていません。</p>   |
| サーバー証明書の検証     | string  | <p>任意のパラメータです。FlowForce などのようにFTP サーバーの証明書を検証するか指定します。有効な値:</p> <ul style="list-style-type: none"> <li>検証無し - 全ての証明書を受け入れます。</li> <li>証明書ストアに対して検証します(デフォルトの値) - Windows でジョブを実行中のユーザーアカウントの証明書ストアとシステムストアを証明書署名を検証するために使用してください。Linux では、システム証明書ストアを使用します。これは通常 /usr/lib/ssl/cert.pem と /usr/lib/ssl/certs または SSL_CERT_FILE と SSL_CERT_DIR 環境関数がポイントするパスにあります。</li> <li>サーバー証明書に対して検証 - FlowForce はサーバー証明書 / パラメータ内で指定され</li> </ul> |

| 名前         | 型                        | 説明  |
|------------|--------------------------|---|
|            |                          | <p>ているFTP サーバーの証明書と比較します。</p> <p>このパラメータの使用はサーバー証明書と安全な接続が必要です。安全な接続を構築できない場合、FTP 関数は失敗します。</p>   |
| サーバーの証明書   | <code>certificate</code> | <p>任意のパラメータです。FlowForce 内の証明書オブジェクトへのパスを指定します。前のパラメータを選択されたサーバー証明書に対して検証に設定している場合、指定されているFlowForce 証明書はFTP サーバー証明書に対して検証されます。それ以外の場合、このパラメータ値は無視されます。</p> |
| ソースファイル    | <code>string</code>      | <p>移動するディレクトリ名です。</p> <p>必須のパラメータです。</p>  |
| ターゲットファイル  | <code>string</code>      | <p>ターゲットの場所でのコピーされたファイルの名前です。コピーされたファイルの名前を変更する場合、異なる名前を使用してください。</p> <p>必須のパラメータです。</p>  |
| エラー時には中断する | <code>boolean</code>     | <p>このブールパラメータはジョブが失敗した場合の戻り値を決定します。「エラー時には中断」がFALSE の場合、関数は、ブール値のFALSE を返します。「エラー時には中断」がTRUE の場合、ジョブの実行は中断されます。デフォルトの値は、TRUE です。</p>                      |
| アカウント      | <code>string</code>      | <p>リモートサーバー上でアクセスすることのできるユーザーが許可されているFTP アカウント名です。</p> <p>任意のパラメータです。</p>   |

### 10.3.5 list

フルパス /system/ftp/list

FTP サーバー上のディレクトリの内容をリストします。実行に成功するとこの関数は文字列のシーケンスを返します。それ以外の場合、結果は以下に説明される通りエラー時に実行を中断 /パラメータにより異なります。

FTP 関数内で `triggerfile` /パラメータを使用する場合、そのパスは絶対的であることを忘れてはいけません。サンプルに関しては

FTP [store](#) 関数を参照してください。

## パラメータ

| 名前          | 型                       | 説明   |
|-------------|-------------------------|--|
| FTP サーバー    | <code>string</code>     | リモート FTP サーバーのアドレス。URL または IP アドレスのいずれかです。<br><br>必須のパラメータです。  |
| ポート         | <code>number</code>     | FTP サーバーに接続するために使用されるポート番号デフォルトの値は21です。  |
| ホスト上のディレクトリ | <code>string</code>     | コンテンツをリストするホスト上のディレクトリ<br>の名前<br><br>任意のパラメータです。デフォルトの値は現在のディレクトリ、"/"です。   |
| ログイン資格情報    | <code>credential</code> | FlowForce 資格情報レコードとしての FTP アカウントのユーザー名とパスワードです。 <a href="#">資格情報</a> を参照してください。<br><br>FTP サーバーが資格情報を必要としない場合、このパラメータをスキップしてください。  |
| 受動モードの使用    | <code>boolean</code>    | 接続問題が発生すると受動モードを使用してください(例えば、ルーターまたはファイアウォールがアクティブな接続を回避するようにセッティングされている場合)。<br><br>任意のパラメータです。デフォルトの値は <code>true</code> です。  |
| SSL/TLS の使用 | <code>string</code>     | (任意のパラメータです。デフォルトの値は <code>いいえ</code> です。) 情報を転送する際は FTP はコマンドチャンネルとデータチャンネルを使用します。FTP データを暗号化無しに転送する場合、この値を <code>いいえ</code> に設定します。それ以外の場合、この値を以下に設定します: <ul style="list-style-type: none"> <li>暗号化されたコマンドチャンネルと共に明示的</li> <li>暗号化されたコマンドとデータチャンネルと共に明示的</li> </ul> <p>上の2つのオプションのいずれかを設定すると証明書は下で説明されているサーバー証明書の検証 / パラメータに依り検証されません。</p> |

| 名前         | 型           | 説明  |
|------------|-------------|---|
|            |             | メモ 明示的な暗号化は奨励されず、FlowForce 内でサポートされていません。   |
| サーバー証明書の検証 | string      | <p>任意のラメーターです。FlowForce などのようにFTP サーバーの証明書を検証するか指定します。有効な値:</p> <ul style="list-style-type: none"> <li>• 検証無し - 全ての証明書を受け入れます。</li> <li>• 証明書ストアに対して検証します (デフォルトの値) - Windows ではジョブを実行中のユーザーアカウントの証明書ストアとシステムストアを証明書署名を検証するために使用してください。Linux では、システム証明書ストアを使用します。これは通常 <code>/usr/lib/ssl/cert.pem</code> と <code>/usr/lib/ssl/certs</code> または <code>SSL_CERT_FILE</code> と <code>SSL_CERT_DIR</code> 環境関数がポイントするパスにあります。</li> <li>• サーバー証明書に対して検証 - FlowForce はサーバー証明書 / ラメーター内で指定されているFTP サーバーの証明書と比較します。</li> </ul> <p>このラメーターの使用はサーバー証明書と安全な接続が必要です。安全な接続を構築できない場合、FTP 関数は失敗します。</p> |
| サーバーの証明書   | certificate | <p>任意のラメーターです。FlowForce 内の証明書オブジェクトへのパスを指定します。前のラメーターを選択されたサーバー証明書に対して検証に設定している場合、指定されているFlowForce 証明書はFTP サーバー証明書に対して検証されます。それ以外の場合、このラメーター値は無視されます。</p>   |
| ワイルドカード    | string      | <p>ホスト上のディレクトリ / ラメーターにより指定されるディレクトリからすべての .js ファイルを抽出する必要がある場合使用するワイルドカード文字列は、例えば <code>*.js</code> です。</p> <p>任意のラメーターです。デフォルトの値はワイルドカードフィルターが実行されないことを意味する空の文字列です。</p>   |

| 名前         | 型                    | 説明  |
|------------|----------------------|---|
| エラー時には中断する | <code>boolean</code> | このパラメータは実行が失敗すると何か起こるかを管理します。実行に失敗するとエラー時に中断が <code>true</code> になり、ジョブ実行が中断され、 <a href="#">保護されたブロック</a> を使用してエラーを処理できます。実行に失敗するとエラー時に中断が <code>false</code> で、関数は空のシーケンスを返します。<br><br>任意のパラメータです。デフォルトの値は <code>true</code> です。 |
| アカウント      | <code>string</code>  | リモートサーバー上のファイルアクセスを許可されているユーザーのFTP アカウント名<br><br>任意のパラメータです。  |

### 10.3.6 retrieve

フルパス: /system/ftp/retrieve

FTP サーバーからファイルを抽出します。

実行に成功すると、この関数はブール値の TRUE を返します。ジョブの実行に失敗すると、結果は「エラー時には中断」/パラメータの値により以下のように異なります:

- 「エラー時には中断」/パラメータが TRUE (デフォルトの値) の場合、ジョブの実行は中断されます。この場合、エラーを保護されたブロックを使用して処理することができます (以下を参照: [ステップのエラーの処理](#))。
- 「エラー時には中断」/パラメータが FALSE の場合、関数は FALSE を返します。

FTP 関数内で `triggerfile` /パラメータを使用する場合、そのパスは絶対的であることを忘れてはいけません。サンプルに関してはFTP [store](#) 関数を参照してください。

#### パラメーター

| 名前          | 型                   | 説明   |
|-------------|---------------------|--|
| FTP サーバー    | <code>string</code> | URL または IP アドレスとしてのリモート FTP サーバーのアドレス<br><br>必須のパラメータです。 |
| ポート         | <code>number</code> | FTP サーバーに接続するために使用されるポート番号です。デフォルトの値は 21 です。             |
| ホスト側のディレクトリ | <code>string</code> | ファイルを抽出するホスト上のディレクトリの名前です。<br><br>任意のパラメータです。            |

| 名前             | 型                       | 説明  |
|----------------|-------------------------|---|
| ログインの資格情報      | <code>credential</code> | FlowForce 資格情報レコードとしての FTP アカウントのユーザー名とパスワードです。資格情報を参照してください。<br><br>FTP サーバーが視覚情報を必要としない場合、このパラメータをスキップします。  |
| 受動モードを使用する     | <code>boolean</code>    | 接続エラーが発生すると、パッシブモードを使用します(例えば、ルーターまたはファイアウォールがアクティブな接続を回避するように設定されている場合)。   |
| SSL/TLS 暗号化の使用 | <code>string</code>     | (任意のパラメータです。デフォルトの値はいいえです。) 情報を転送するときは FTP はコマンドチャンネルとデータチャンネルを使用します。FTP データを暗号化無しに転送する場合、この値をいいえに設定します。それ以外の場合、この値を以下に設定します:<br><br><ul style="list-style-type: none"> <li>暗号化されたコマンドチャンネルと共に明示的</li> <li>暗号化されたコマンドとデータチャンネルと共に明示的</li> </ul> <p>上の2つのオプションのいずれかを設定すると証明書は下で説明されているサーバー証明書の検証 / パラメータに従い検証されます。</p> <p>メモ 明示的な暗号化は奨励されず、FlowForce 内でサポートされていません。</p>  |
| サーバー証明書の検証     | <code>string</code>     | 任意のパラメータです。FlowForce がどのように FTP サーバーの証明書を検証するか指定します。有効な値:<br><br><ul style="list-style-type: none"> <li>検証無し - 全ての証明書を受け入れます。</li> <li>証明書ストアに対して検証します (デフォルトの値) - Windows でジョブを実行中のユーザーアカウントの証明書ストアとシステムストアを証明書署名を検証するために使用してください。Linux では システム証明書ストアを使用します。これは通常 <code>/usr/lib/ssl/cert.pem</code> と <code>/usr/lib/ssl/certs</code> または <code>SSL_CERT_FILE</code> と <code>SSL_CERT_DIR</code> 環境関数が</li> </ul> |

| 名前         | 型           | 説明  |
|------------|-------------|---|
|            |             | <p>ポイントするパスにします。</p> <ul style="list-style-type: none"> <li>サーバー証明書に対して検証 - FlowForce はサーバー証明書 / ラメータ内で指定されているFTP サーバーの証明書と比較します。</li> </ul> <p>このラメータの使用はサーバー証明書と安全な接続が必要です。安全な接続を構築できない場合、FTP 関数は失敗します。</p> |
| サーバーの証明書   | certificate | <p>任意のラメータです。FlowForce 内の証明書オブジェクトへのパスを指定します。前のラメータを選択されたサーバー証明書に対して検証に設定している場合、指定されている FlowForce 証明書は FTP サーバー証明書に対して検証されます。それ以外の場合、このラメータ値は無視されます。</p>  |
| ソースファイル    | string      | <p>抽出するディレクトリ名です。</p> <p>必須のラメータです。</p>   |
| ターゲットファイル  | string      | <p>抽出後のファイル名です。</p> <p>必須のラメータです。</p>   |
| ターゲットの上書き  | boolean     | <p>true の場合は、保存先ファイルが上書きされます。デフォルトの値は false です。</p>   |
| エラー時には中断する | boolean     | <p>このブールラメータはジョブが失敗した場合の戻り値を決定します。「エラー時には中断」が FALSE の場合、関数は、ブール値の FALSE を返します。「エラー時には中断」が TRUE の場合、ジョブの実行は中断されます。デフォルトの値は TRUE です。</p>  |
| 作業ディレクトリ   | string      | <p>ジョブの作業ディレクトリを指定します(例えば c:\somedirectory)。相対的なパスが使用されると、作業ディレクトリに対して解決されます。</p>   |
| アカウント      | string      | <p>リモートサーバー上でアクセスすることのできるユーザーが許可されているFTP アカウント名です。</p> <p>任意のラメータです。</p>  |

### 10.3.7 retrieve-wildcard

フルパス /system/ftp/retrieve-wildcard

FTP サーバーからワイルドカードに一致するファイルを抽出します。例、\*.xml。成功すると、関数は書き込まれたファイル絶対ローカルパスのリストを返すか、一致が見つからない場合は空のリストを返します。実行に失敗すると、結果は以下で説明されているエラー時に中断 / パラメータにより異なります。

FTP 関数内で `triggerfile` / パラメータを使用する場合、そのパスは絶対的であることを忘れてはいけません。サンプルに関してはFTP [store](#) 関数を参照してください。

#### パラメータ

| 名前          | 型          | 説明   |
|-------------|------------|--|
| FTP サーバー    | string     | リモート FTP サーバーのアドレス。URL または IP アドレスのいずれかです。<br><br>必須のパラメータです。  |
| ポート         | number     | FTP サーバーに接続するために使用されるポート番号デフォルトの値は21です。  |
| ホスト上のディレクトリ | string     | ファイルを削除するホスト上のディレクトリ<br>の名前<br><br>任意のパラメータです。   |
| ログイン資格情報    | credential | FlowForce 資格情報レコードとしての FTP アカウントのユーザー名とパスワードです。 <a href="#">資格情報</a> を参照してください。<br><br>FTP サーバーが資格情報を必要としない場合、このパラメータをスキップしてください。  |
| 受動モードの使用    | boolean    | 接続問題が発生すると受動モードを使用してください(例えば、ルーターまたはファイアウォールがアクティブな接続を回避するようにセオアップされている場合)。  |
| SSL/TLS の使用 | string     | (任意のパラメータです。デフォルトの値はいいえです。) 情報を転送する際はFTP はコマンドチャンネルとデータチャンネルを使用します。FTP データを暗号化無しに転送する場合、この値をいいえに設定します。それ以外の場合、この値を以下に設定します: <ul style="list-style-type: none"> <li>暗号化されたコマンドチャンネルと共に明示的</li> <li>暗号化されたコマンドとデータチャンネルと共に明示的</li> </ul> |

| 名前         | 型                        | 説明   |
|------------|--------------------------|--|
|            |                          | <p>上の2つのオプションのいずれかを設定すると証明書は下で説明されているサーバー証明書の検証 / パラメータに従い検証されます。</p> <p>メモ 明示的な暗号化は奨励されず、FlowForce 内でサポートされていません。</p>   |
| サーバー証明書の検証 | <code>string</code>      | <p>任意のパラメータです。FlowForce などのようにFTP サーバーの証明書を検証するため指定します。有効な値</p> <ul style="list-style-type: none"> <li>• 検証無し - 全ての証明書を受け入れます。</li> <li>• 証明書ストアに対して検証します (デフォルトの値) - Windows でジョブを実行中のユーザーアカウントの証明書ストアとシステムストアを証明書署名を検証するために使用してください。Linux では、システム証明書ストアを使用します。これは通常 <code>/usr/lib/ssl/cert.pem</code> と <code>/usr/lib/ssl/certs</code> または <code>SSL_CERT_FILE</code> と <code>SSL_CERT_DIR</code> 環境関数がポイントするパスにあります。</li> <li>• サーバー証明書に対して検証 - FlowForce はサーバー証明書 / パラメータ内で指定されているFTP サーバーの証明書と比較します。</li> </ul> <p>このパラメータの使用はサーバー証明書と安全な接続が必要です。安全な接続を構築できない場合、FTP 関数は失敗します。</p> |
| サーバーの証明書   | <code>certificate</code> | <p>任意のパラメータです。FlowForce 内の証明書オブジェクトへのパスを指定します。前のパラメータを選択されたサーバー証明書に対して検証に設定している場合、指定されているFlowForce 証明書はFTP サーバー証明書に対して検証されます。それ以外の場合、このパラメータ値は無視されます。</p>  |
| ワイルドカード    | <code>string</code>      | <p>必須のパラメータです。ワイルドカードを指定します。例、<code>*.xml</code>。ワイルドカードに</p>  |

| 名前         | 型       | 説明   |
|------------|---------|--|
|            |         | 致するファイルが抽出されます。 <a href="#">FTP 関数内のビルドカード</a> を参照してください。  |
| エラー時には中断する | boolean | 実行に失敗するとこのパラメータは関数の振る舞いを指定します。具体的には、実行が失敗した場合、関数は以下の1つを返します: <ul style="list-style-type: none"> <li>エラー時に中断 / パラメータが <b>false</b> の場合、抽出されたファイルのリストは返されません。</li> <li>エラー時に中断 / パラメータが <b>true</b> の場合、失敗が発生するまで関数はファイルを抽出し、実行を中断します。</li> </ul> <p>このために、ファイルの一部は実行が失敗しても抽出される場合があります。</p> |
| 作業ディレクトリ   | string  | FTP サーバーから抽出されるすべてのファイルのあるディレクトリ   |
| アカウント      | string  | リモートサーバー上のファイルにアクセスを許可されているユーザーのFTP アカウント名<br>任意のパラメータです。  |

### 10.3.8 rmdir

フルパス: /system/ftp/rmdir

FTP サーバーからディレクトリを削除します。

実行に成功すると、この関数はブール値の TRUE を返します。ジョブの実行に失敗すると、結果は「エラー時には中断」/パラメータの値により以下のように異なります:

- 「エラー時には中断」/パラメータが TRUE (デフォルトの値) の場合、ジョブの実行は中断されます。この場合、エラーを保護されたブロックを使用して処理することができます (以下を参照: [ステップのエラーの処理](#))。
- 「エラー時には中断」/パラメータが FALSE の場合、関数は FALSE を返します。

FTP 関数内で **triggerfile** /パラメータを使用する場合、そのパスは絶対的であることを忘れてはいけません。サンプルに関してはFTP [store](#) 関数を参照してください。

#### パラメーター

| 名前       | 型      | 説明                                    |
|----------|--------|---------------------------------------|
| FTP サーバー | string | URL または IP アドレスとしてのリモート FTP サーバーのアドレス |

| 名前             | 型          | 説明  |
|----------------|------------|---|
|                |            | 必須のラメーターです。   |
| ポート            | number     | FTP サーバーに接続するために使用されるポート番号です。デフォルトの値は21です。  |
| ホスト側のディレクトリ    | string     | ファイルを削除するホスト上のディレクトリの名前です。<br><br>任意のラメーターです。   |
| ログインの資格情報      | credential | FlowForce 資格情報レコードとしての FTP アカウントのユーザー名とパスワードです。資格情報を参照してください。<br><br>FTP サーバーが視覚情報を必要としない場合、このラメーターをスキップします。  |
| 受動モードを使用する     | boolean    | 接続エラーが発生すると、パッシブモードを使用します(例えば、ルーターまたはファイアウォールがアクティブな接続を回避するように設定されている場合)。   |
| SSL/TLS 暗号化の使用 | string     | (任意のラメーターです。デフォルトの値はいいえです。) 情報を転送するときはFTP はコマンドチャンネルとデータチャンネルを使用します。FTP データを暗号化無しに転送する場合、この値をいいえに設定します。それ以外の場合、この値を以下に設定します:<br><br><ul style="list-style-type: none"> <li>暗号化されたコマンドチャンネルと共に明示的</li> <li>暗号化されたコマンドとデータチャンネルと共に明示的</li> </ul> <p>上の2つのオプションのいずれかを設定すると証明書は下で説明されているサーバー証明書の検証 / ラメーターに従い検証されます。</p> <p>メモ 明示的な暗号化は奨励されず、FlowForce 内でサポートされていません。</p> |
| サーバー証明書の検証     | string     | 任意のラメーターです。FlowForce などのようにFTP サーバーの証明書を検証するか指定します。有効な値:<br><br><ul style="list-style-type: none"> <li>検証無し - 全ての証明書を受け入れます。</li> <li>証明書ストアに対して検証します (デフォルトの値) -</li> </ul>  |

| 名前           | 型                        | 説明  |
|--------------|--------------------------|---|
|              |                          | <p>Windows でジョブを実行中のユーザーアカウントの証明書ストアとシステムストアを証明書署名を検証するために使用してください。Linux では、システム証明書ストアを使用します。これらは通常 <code>/usr/lib/ssl/cert.pem</code> と <code>/usr/lib/ssl/certs</code> または <code>SSL_CERT_FILE</code> と <code>SSL_CERT_DIR</code> 環境関数がポイントするパスにあります。</p> <ul style="list-style-type: none"> <li>• サーバー証明書に対して検証 - FlowForce はサーバー証明書 / パラメータ内で指定されている FTP サーバーの証明書と比較します。</li> </ul> <p>このパラメータの使用はサーバー証明書と安全な接続が必要です。安全な接続を構築できない場合、FTP 関数は失敗します。</p> |
| サーバーの証明書     | <code>certificate</code> | <p>任意のパラメータです。FlowForce 内の証明書オブジェクトへのパスを指定します。前のパラメータを選択されたサーバー証明書に対して検証に設定している場合、指定されている FlowForce 証明書は FTP サーバー証明書に対して検証されます。それ以外の場合、このパラメータ値は無視されます。</p>   |
| ターゲット ディレクトリ | <code>string</code>      | <p>削除するディレクトリ名です。</p> <p>必須のパラメータです。</p>  |
| エラー時には中断する   | <code>boolean</code>     | <p>このブールパラメータがジョブが失敗した場合の戻り値を決定します。「エラー時には中断」が FALSE の場合、関数は、ブール値の FALSE を返します。「エラー時には中断」が TRUE の場合、ジョブの実行は中断されます。デフォルトの値は TRUE です。</p>   |
| アカウント        | <code>string</code>      | <p>リモートサーバー上でアクセスすることのできるユーザーが許可されている FTP アカウント名です。</p> <p>任意のパラメータです。</p>  |

### 10.3.9 store

フルパス: /system/ftp/store

FTP 削除→バイナリファイルをアップロードします。

実行に成功すると、この関数はブール値の TRUE を返します。ジョブの実行に失敗すると、結果は「エラー時には中断」/パラメータの値により以下のように異なります:

- 「エラー時には中断」/パラメータが TRUE (デフォルトの値) の場合、ジョブの実行は中断されます。この場合、エラーを保護されたブロックを使用して処理することができます (以下を参照: [ステップのエラーの処理](#))。
- 「エラー時には中断」/パラメータが FALSE の場合、関数は FALSE を返します。

#### パラメータ

| 名前             | 型          | 説明   |
|----------------|------------|--|
| FTP サーバー       | string     | URL または IP アドレスとしてのポート FTP サーバーのアドレス<br><br>必須のパラメータです。  |
| ポート            | number     | FTP サーバーに接続するために使用されるポート番号です。デフォルトの値は 21 です。   |
| ホスト側のディレクトリ    | string     | ファイルを保管するホスト上のディレクトリの名前です。<br><br>任意のパラメータです。  |
| ログインの資格情報      | credential | FlowForce 資格情報レコードとしての FTP アカウントのユーザー名とパスワードです。 <a href="#">資格情報</a> を参照してください。<br><br>FTP サーバーが視覚情報を必要としない場合、このパラメータをスキップします。   |
| 受動モードを使用する     | boolean    | 接続エラーが発生すると、パッシブモードを使用します (例えば、ルーターまたはファイアウォールがアクティブな接続を回避するように設定されている場合)。   |
| SSL/TLS 暗号化の使用 | string     | (任意のパラメータです。デフォルトの値はいいえです。) 情報を転送する際は FTP はコマンドチャンネルとデータチャンネルを使用します。FTP データを暗号化無しに転送する場合、この値をいいえに設定します。それ以外の場合、この値を以下に設定します: <ul style="list-style-type: none"> <li>• 暗号化されたコマンドチャンネルと共に明示的</li> </ul> |

| 名前         | 型           | 説明   |
|------------|-------------|--|
|            |             | <ul style="list-style-type: none"> <li>暗号化されたコマンドとデータチャンネルと共に明示的</li> </ul> <p>上の2つのオプションのいずれかを設定すると証明書は下で説明されているサーバー証明書の検証 / パラメータに従い検証されます。</p> <p>メモ 明示的な暗号化は奨励されず、FlowForce 内でサポートされていません。</p>   |
| サーバー証明書の検証 | string      | <p>任意のパラメータです。FlowForce などのようにFTP サーバーの証明書を検証するため指定します。有効な値</p> <ul style="list-style-type: none"> <li>検証無し - 全ての証明書を受け入れます。</li> <li>証明書ストアに対して検証します (デフォルトの値) - Windows でジョブを実行中のユーザーアカウントの証明書ストアとシステムストアを証明書署名を検証するために使用してください。Linux では、システム証明書ストアを使用します。これは通常 <code>/usr/lib/ssl/cert.pem</code> と <code>/usr/lib/ssl/certs</code> または <code>SSL_CERT_FILE</code> と <code>SSL_CERT_DIR</code> 環境関数がポイントするパスにあります。</li> <li>サーバー証明書に対して検証 - FlowForce はサーバー証明書 / パラメータ内で指定されているFTP サーバーの証明書と比較します。</li> </ul> <p>このパラメータの使用はサーバー証明書と安全な接続が必要です。安全な接続を構築できない場合、FTP 関数は失敗します。</p> |
| サーバーの証明書   | certificate | <p>任意のパラメータです。FlowForce 内の証明書オブジェクトへのパスを指定します。前のパラメータを選択されたサーバー証明書に対して検証に設定している場合、指定されているFlowForce 証明書はFTP サーバー証明書に対して検証されます。それ以外の場合、このパラメータ値は無視されます。</p>  |

| 名前         | 型       | 説明  |
|------------|---------|---|
| ソースファイル    | string  | FTP サーバーにアップロードされるファイル名です。<br>必須のパラメータです。   |
| ターゲットファイル  | string  | FTP サーバーへのアップロード後のファイル名です。これはソースファイルとは異なる場合があります。<br>必須のパラメータです。  |
| エラー時には中断する | boolean | このブール値パラメータはジョブが失敗した場合の戻り値を決定します。「エラー時には中断」がFALSE の場合、関数はブール値のFALSE を返します。「エラー時には中断」がTRUE の場合、ジョブの実行は中断されます。デフォルトの値は TRUE です。 |
| 作業ディレクトリ   | string  | ジョブの作業ディレクトリを指定します(例えば <code>c:\somedirectory</code> )。相対的なパスが使用されると、作業ディレクトリに対して解決されます。                                      |
| アカウント      | string  | リモートサーバー上でアクセスすることのできる、ユーザーが許可されているFTP アカウント名です。<br>任意のパラメータです。   |

## サンプル

このサンプルはジョブ構成時にファイル名と拡張子を知ることなく、リモート FTP サーバーにファイルをアップロードする方法を説明しています。これは下記の通りファイルシステムトリガーとしてジョブを実行するように構成することにより可能になります。

### Triggers

Check: Content of file or directory:  polling interval:  seconds.

Start:

Expires:

Time zone: Europe/Berlin

enabled

上記のトリガーはディレクトリ `C:\FlowForce\Upload` の変更をモニターします。ファイルをこのディレクトリにコピーすると、ジョブが実行され、ジョブをトリガーしたファイルの絶対パスが `triggerfile` 入力パラメータ内で使用できるようになります。これにより下記のとおりジョブ内でこのファイルを名前および拡張子を知ることなく使用できるようになります。

### Job Input Parameters

+  
Name:  Type:  Default:

+  
Execution Steps

+  
Execute function /system/ftp/store

|             |                    |  |
|-------------|--------------------|--|
| Parameters: | FTP Server:        | <input type="text" value="10.100.63.200"/>   |
|             | Port:              | <input type="text" value="21"/>  |
|             | Directory on host: | <input type="text" value="uploads"/>   |
|             | Login credentials: | <input checked="" type="radio"/> Select existing credential: <input type="text" value="/public/my.ftp.credentials"/><br><input type="radio"/> Define local credential: |
|             | Use passive mode:  | <input type="button" value="+"/>   |
|             | Source file:       | <input type="text" value="{filename-with-extension(triggerfile)}"/>  |
|             | Target file:       | <input type="text" value="{filename-with-extension(triggerfile)}"/>  |
|             | Abort on error:    | <input type="button" value="+"/>   |
|             | Working directory: | <input type="text" value="C:\FlowForce\Upload"/>   |
|             | Account:           | <input type="button" value="+"/>   |

上記のジョブ構成では store 関数は以下のパラメータと共に呼び出されます:

- FTP サーバー - FTP サーバーのアドレス(このサンプルの場合 I.P. アドレス)
- ポート - デフォルトのポートは 21 です。
- ホスト上のディレクトリ - このサンプルでは、アップロードされたすべてのファイルを FTP ルートディレクトリに相対するサーバー上の「アップロード」サブディレクトリに置きます。
- ログインの資格情報 - FTP ユーザー名とパスワードが FTP サーバーに接続する必要があります。再利用の目的のために [資格情報](#) として以前に定義されており、ここでは **public** コンテナから参照されています。
- ソースファイル - アップロードされるローカルファイルのパスです。作業パスが設定されているため、このサンプルではこれは相対的なパスである必要があります。
- ターゲットファイル - アップロード後の FTP サーバー上のファイルのパスです。このサンプルではホスト上のディレクトリが設定されているため、これは相対的なパスである必要があります。
- 作業ディレクトリ - ローカルコンピュータ上のディレクトリです。ローカルの相対パスはこのディレクトリに対して相対的であると仮定されます。トリガー内で定義されているパーキングディレクトリと同じであることを注意してください。

作業ディレクトリが設定されていると、ソースファイルは絶対的ではなく相対的である必要があります。同様にホスト上のディレクトリが設定されていると、ターゲットファイルは相対的なパスである必要があります。

このサンプルは作業ディレクトリとホスト上のディレクトリを使用するため triggerfile の絶対的パスを相対的パスに変更する必要があります。

これを達成するためにソースファイルはFlowForce 式を使用します。この式は **triggerfile** を引数(これは絶対パスであることにご注意してください)として取りファイル名と拡張子のみを返します。例えば **triggerfile** が `C:\data.txt` の場合、式は `data.txt` のみを返します。ターゲットファイル内の式を使用しても同様に作動します。FlowForce 内の式の詳細に関しては [The FlowForce 式言語](#) を参照してください。

上記の構成では、(拡張子に関わらず) ファイルを作業ディレクトリにコピーすると以下が発生します:

- **data.txt** と呼ばれるファイルがコピーされた後、ジョブが実行され `C:\FlowForce\Upload\data.txt` を **triggerfile** として取得します。
- 式を使用することでソースファイルは `data.txt` になり、ターゲット も同様にまります。
- アップロードするファイルの実際のパスは作業ディレクトリをソースファイルと結合することにより取得されます。
- サーバー上のファイルの目的のパスはホスト上のディレクトリをターゲットファイルと結合することにより取得されます。
- FlowForce は提供済みのFTP 資格情報に接続しようと試みます。成功するとファイル **data.txt** をFTP サーバー上のアップロード ディレクトリにアップロードします。

### 10.3.10 store-wildcard

フルパス /system/ftp/store-wildcard

ワイルドカードに一致する場合、FTP サーバーファイルをローカルディレクトリからアップロードします。例、`*.xml`。成功すると、関数はアップロードされたファイル(絶対ローカルパス)のリストを返すか、一致が見つかない場合は空のリストを返します。実行に失敗すると、結果は以下で説明されているエラー時に中断 / パラメータにより異なります。

FTP 関数内で **triggerfile** / パラメータを使用する場合、そのパスは **絶対的**であることを忘れてください。サンプルに関してはFTP [store](#) 関数を参照してください。

#### パラメータ

| 名前          | 型          | 説明  |
|-------------|------------|---|
| FTP サーバー    | string     | リモート FTP サーバーのアドレス。URL または IP アドレスのいずれかです。<br><br>必須のパラメータです。   |
| ポート         | number     | FTP サーバーに接続するために使用されるポート番号デフォルトの値は21です。   |
| ホスト上のディレクトリ | string     | ファイルを削除するホスト上のディレクトリの名前<br><br>任意のパラメータです。  |
| ログイン資格情報    | credential | FlowForce 資格情報レコードとしての FTP アカウントのユーザー名とパスワードです。 <a href="#">資格情報</a> を参照してください。<br><br>FTP サーバーが資格情報を必要としない場合、このパラメータをスキップしてください。 |

| 名前          | 型                    | 説明   |
|-------------|----------------------|--|
| 受動モードの使用    | <code>boolean</code> | 接続問題が発生すると受動モードを使用してください(例えば、ルーターまたはファイアウォールがアクティブな接続を回避するようにセトアップされている場合)。  |
| SSL/TLS の使用 | <code>string</code>  | <p>(任意のラメーターです。デフォルトの値はいいえです。) 情報を転送する際はFTPはコマンドチャンネルとデータチャンネルを使用します。FTP データを暗号化無しに転送する場合、この値をいいえに設定します。それ以外の場合、この値を以下に設定します:</p> <ul style="list-style-type: none"> <li>暗号化されたコマンドチャンネルと共に明示的</li> <li>暗号化されたコマンドとデータチャンネルと共に明示的</li> </ul> <p>上の2つのオプションのいずれかを設定すると証明書は下で説明されているサーバー証明書の検証 / ラメーターに従い検証されます。</p> <p>メモ 明示的な暗号化は奨励されず、FlowForce 内でサポートされていません。</p>  |
| サーバー証明書の検証  | <code>string</code>  | <p>任意のラメーターです。FlowForce がどのようにFTP サーバーの証明書を検証するか指定します。有効な値:</p> <ul style="list-style-type: none"> <li>検証無し - 全ての証明書を受け入れます。</li> <li>証明書ストアに対して検証します (デフォルトの値) - Windows でジョブを実行中のユーザーアカウントの証明書ストアとシステムストアを証明書署名を検証するために使用してください。Linux では、システム証明書ストアを使用します。これは通常 <code>/usr/lib/ssl/cert.pem</code> と <code>/usr/lib/ssl/certs</code> または <code>SSL_CERT_FILE</code> と <code>SSL_CERT_DIR</code> 環境関数がポイントするパスにあります。</li> <li>サーバー証明書に対して検証 - FlowForce はサーバー証明書 / ラメーター内で指定されているFTP サーバーの証明書と比較します。</li> </ul> |

| 名前         | 型                        | 説明  |
|------------|--------------------------|---|
|            |                          | このラメーターの使用はサーバー証明書と安全な接続が必要です。安全な接続を構築できない場合、FTP 関数は失敗します。  |
| サーバーの証明書   | <code>certificate</code> | 任意のラメーターです。FlowForce 内の証明書オブジェクトへのパスを指定します。前のラメーターを選択されたサーバー証明書に対して検証に設定している場合、指定されている FlowForce 証明書は FTP サーバー証明書に対して検証されます。それ以外の場合、このラメーター値は無視されます。  |
| ワイルドカード    | <code>string</code>      | 必須のラメーターです。ワイルドカードを指定します。例、 <code>*.xml</code> 。このワイルドカードに一致すると作業ディレクトリ/ラメーターにより指定されるディレクトリからのファイルがアップロードされます。<br><br><a href="#">FTP 関数内のワイルドカード</a> を参照してください。  |
| エラー時には中断する | <code>boolean</code>     | 実行に失敗するとこのラメーターは関数の振る舞いを指定します。具体的には、実行が失敗した場合、関数は以下の一つを返します: <ul style="list-style-type: none"> <li>エラー時に中断 / ラメーターが <code>false</code> の場合、アップロードされたファイルのリストは返されません。</li> <li>エラー時に中断 / ラメーターが <code>true</code> の場合、失敗が発生するまで関数はファイルをアップロードし、実行を中断します。</li> </ul> <p>このため、ファイルの一部は実行が失敗してもアップロードされる場合があります。</p> |
| 作業ディレクトリ   | <code>string</code>      | ワイルドカードが一致する場合、FTP サーバーにアップロードされるファイルが存在するディレクトリ。   |
| アカウント      | <code>string</code>      | リモートサーバー上のファイルにアクセスを許可されているユーザーの FTP アカウント名<br><br>任意のラメーターです。  |

## 10.4 /system/mail

/system/mail コンテナは、電子メールを送信するために使用される関数 `send` と `send-mime` を含んでいます。

### 10.4.1 send

フルパス: /system/mail/send

指定された宛先、通常は管理者、に電子メールを送信します

実行に成功すると、この関数はブール値の `TRUE` を返します。ジョブの実行に失敗すると、結果は「エラー時には中断」/パラメータの値により以下のように異なります:

- 「エラー時には中断」/パラメータが `TRUE` (デフォルトの値) の場合、ジョブの実行は中断されます。この場合、エラーを保護されたブロックを使用して処理することができます (以下を参照: [ステップのエラーの処理](#))。
- 「エラー時には中断」/パラメータが `FALSE` の場合、関数は `FALSE` を返します。

この関数を使用する前に、メールサーバー設定が構成されていることを確認してください ([メールパラメータの設定](#)を参照してください)。

#### パラメータ

| 名前  | 型                   | 説明  |
|-----|---------------------|---|
| 差出人 | <code>string</code> | 電子メールが送信されるアドレス。例: <code>flowforce@&lt;hostname&gt;</code> 。  |
| 宛先  | <code>string</code> | 電子メールアドレスの宛先必須のパラメータです。このフィールドには複数の宛先電子メールアドレスのコンマで区切られたリストを含んでいる場合があります。   |
| 件名  | <code>string</code> | メッセージの件名必須のパラメータです。   |
| 本文  | <code>string</code> | 文字列としてメッセージの本文を提供する任意のパラメータ。メッセージの本文は、ASCII と Unicode 文字をサポートします。<br><br>テキストが1行以上ある場合 (すなわち、改行を入力後) メッセージ本文のためのテキストボックスには付番機能があります。更に、波 { } カッコ囲まれけ式を入力すると、 <a href="#">FlowForce 式</a> の基本的構文の <code>WRITE</code> 機能を使用することができます。<br><br>このテキストボックス内に入力する場合以下の制限があることに注意してください! |

| 名前         | 型                           | 説明  |
|------------|-----------------------------|---|
|            |                             | <ul style="list-style-type: none"> <li>FlowForce 式は同じラインで開始し終了する必要がありません。</li> <li>元に戻すやリ値しサポートを使用することはできません。</li> <li>数千のラインを含む長い本文はエディターのパフォーマンスに影響を与えます。</li> </ul> <p>任意のラメーターです。</p> |
| 添付         | <code>string as file</code> | 電子メールと共に送信される添付ファイルの名前。   |
| エラー時には中断する | <code>boolean</code>        | このブールラメーターはジョブが失敗した場合の戻り値を決定します。「エラー時には中断」がFALSE の場合、関数はブール値のFALSE を返します。「エラー時には中断」がTRUE の場合、ジョブの実行は中断されます。デフォルトの値はTRUE です。   |

## サンプル

[エラー処理をジョブに追加する](#)を参照

### 10.4.2 send-mime

フルパス: /system/mail/send-mime

指定済みの宛先に電子メールを送信します。通常宛先は管理者です。

この関数を使用する前に、メールサーバー設定が構成されていることを確認してください([メールラメーターの設定](#)を参照してください)。

実行に成功すると、この関数はブール値のTRUE を返します。ジョブの実行に失敗すると、結果は「エラー時には中断」/ラメーターの値により以下のように異なります:

- 「エラー時には中断」/ラメーターが TRUE (デフォルトの値) の場合、ジョブの実行は中断されます。この場合、エラーを保護されたブロックを使用して処理することができます(以下を参照 [ステップのエラーの処理](#))。
- 「エラー時には中断」/ラメーターが FALSE の場合、関数は FALSE を返します。

send 関数とは異なり、この関数のメッセージ本文/ラメーターは、文字列ではなくストリームを生成する式を期待します。これにより、ストリームから(例えば、HTML として)メッセージを取得することができます。

HTML コンテンツをメッセージ本文のために取得するには、HTML 出力をMIME として生成する、StyleVision Server 変換を呼び出すことが強く奨励されます。FlowForce Server 自身は、イメージ、スタイルシート、または HTML ファイルにより参照されている類似したリソースを内蔵 MIME ストリームに集めません。

内蔵 HTML メッセージ本文を StyleVision Server を使用して生成する場合は、以下を行います:

1. Altova StyleVision 内で電子メールのHTML 本文をデザインします。デザインにはローカルイメージとスタイルシートが含まれる場合があります。
2. StyleVision 変換を FlowForce Server にデプロイします。FlowForce 内では、変換は StyleVision Server により実行可能な内蔵 FlowForce 関数になります。
3. 上記の StyleVision Server 変換を呼び出すジョブを作成します。**GenerateHtmlOutputAsMime** オプションがジョブ構成ページで選択されていることを確認してください。
4. ジョブ構成ページ内で FlowForce Server 式関数を呼出、生成された MIME ストリームをピックアップし、send-mime 関数の「メッセージ本文」パラメータに設定します(下の「例 1」を参照してください)。

HTML ファイルにより参照されている外部リソースは MIME ストリームに埋め込むことができず、電子メールの添付として追加されます。

HTML 出力を MIME ストリームとして生成するジョブのサンプルは、以下で説明されています。StyleVision 変換を FlowForce Server にデプロイする方法をステップバイステップで説明しているサンプルに関しては、[StyleVision 変換からのジョブの作成](#)を参照してください。StyleVision Server 統合の詳細に関しては、[他の Altova サーバーとの統合](#)を参照してください。

メッセージ本文のためにストリームを直接 FlowForce 内で作成する場合は、stream-open または stream-from-string などの式関数を呼び出すこともできます。同様に、[MIME 式関数](#)を使用して電子メール、または、添付メッセージヘッダーをカスタマイズすることができます。

宛先側の迷惑メールフォルダーに電子メールが振り分けられないように、MIME ヘッダーを受信先のサーバー、または、プログラムで許可されている方法で構成してください。

## パラメーター

| 名前  | 型                  | 説明  |
|-----|--------------------|---|
| 差出人 | string             | 電子メールが送信されるアドレス。例:<br>flowforce@<hostname>.   |
| 宛先  | string             | 電子メールアドレスの宛先必須のパラメーターです。このフィールドには複数の宛先電子メールアドレスのコンマで区切られたリストを含んでいる場合があります。  |
| 件名  | string             | メッセージの件名必須のパラメーターです。  |
| 本文  | stream             | ストリーム型を返す FlowForce 式としてのメッセージの本文テキスト。  |
| 添付  | sequence of stream | 電子メールと共に送信される添付ファイルです。各添付は、ストリームを生成する FlowForce 式である必要があります。文字列またはファイルからストリームを作成するために、 <a href="#">ストリーム式関数</a> を呼び出します。 <a href="#">MIME 式関数</a> を呼び出して、MIME ヘッダーを追加、変更、または、削除することができます。 |

| 名前         | 型       | 説明  |
|------------|---------|---|
| エラー時には中断する | boolean | このブール変数はジョブが失敗した場合の戻り値を決定します。「エラー時には中断」がFALSE の場合、関数は、プール値のFALSE を返します。「エラー時には中断」がTRUE の場合、ジョブの実行は中断されます。デフォルトの値はTRUE です。 |

## サンプル

サンプル1: HTML 書式で電子メールを送信するには、下で説明されるジョブは `send-mime` 関数を呼び出します。

最初の実行ステップはHTML 出力を StyleVision Server 変換を呼び出して生成します。この変換は StyleVision を使用してデザインされ、[StyleVision 変換からのジョブ](#)の作成で説明されている通り、FlowForce Server にデプロイされます。

**GenerateHtmlOutputAsMime** チェックボックスが選択されています。最初の実行ステップの結果は「出力」と呼ばれています(この文字列は以降のステップで必要とされ、式内「output」を入力することにより参照することができます)。

2番目の実行ステップはFlowForce 式を計算するために `compute` 関数を呼び出します。具体的には、式 `nth(results(output), 0)` はインデックス0 を持つ(この場合、**OutHtml**) MIME ストリームを前のステップにより生成された結果からピックアップします。

最後に、3番目の実行ステップは電子メールメッセージを送信します。「メッセージ本文」フィールドは、ステップ2で前に計算された結果であるFlowForce 式を含んでいます。

### 実行ステップ

**+** 関数の実行 /public/BiggestCitiesPerCity.transformation

パラメーター:

- InputXml:
- OutHtml:
- GenerateHtmlOutputAsMime:
- OutRtf:
- OutFo:
- OutPdf:
- OutDocx:
- Working-directory:

= ステップの結果を割り当て  以下としてReturnTypeDefocx, ReturnTypeDefo, ReturnTypeDefhtml, ReturnT

**+** 関数の実行 /system/compute

パラメーター: 条件式:

= ステップの結果を割り当て  以下としてT0

**+** 関数の実行 /system/mail/send-mime

パラメーター:

- 差出人:
- 宛先:
- 件名:
- 本文:
- 添付:
- エラー時には中断:

= ステップの結果を割り当て  以下としてプール値

サンプル2: 下で表示されているジョブはHTML 書式のイメージ添付を含む.png 書式で電子メールを送信する send-mime 関数を呼び出します。

### 実行ステップ

**+** 関数の実行 /system/compute-string

パラメーター: 条件式:  文字列 (必須) [セットする値](#)

= ステップの結果を割り当て  以下として文字列

**+** 関数の実行 /system/mail/send-mime

パラメーター:

|           |   |          |      |
|-----------|---|----------|------|
| 差出人:      | <input type="text" value="flowforce@yourhostname"/>   | 文字列      | (必須) |
| 宛先:       | <input type="text" value="to@example.org"/>   | 文字列      | (必須) |
| 件名:       | <input type="text" value="Example mail"/>   | 文字列      | (必須) |
| 本文:       | <input type="text" value="stream-from-string(message, 'UTF-8', 'text/html')"/>  | 条件式ストリーム | (必須) |
| 添付:       | <b>+</b> <a href="#">add-mime-header</a> (stream-open('C:\sample.png', 'image/png'), 'Content-Disposition', 'attachment; filename='sample.png') |          |      |
| エラー時には中断: | <b>+</b>  |          |      |

= ステップの結果を割り当て  以下としてプール値

最初の実行ステップはメッセージの内容のためのHTMLコードを準備します。簡素化のため、このサンプル内のHTMLコードは直接テキストボックスに入力されます。HTML出力を取得するには、奨励される方法は、前のサンプルで説明されている通りStyleVision Server変換を呼び出すことです。この実行ステップは次のステップで使用される文字列出力を生成します。

2番目の実行ステップは電子メールのボディをストリームとして生成します。具体的には最初の実行ステップ(`body_html`, 型 `string`)の結果を `stream` に変換します。[Stream-from-string](#) 関数はこの目的のために呼び出されます。希望されるエンコードとMIME型はこの関数の引数として与えられます。

3番目の実行ステップは電子メールの添付をストリームとして生成します。イメージ添付はローカルディスクからです: `C:\sample.png`。具体的には、このステップは以下の式を計算します:

```
add-mime-header(stream-open('C:\sample.png', 'image/png'), 'Content-
Disposition', 'attachment; filename=sample.png')
```

上の式は以下を行います:

- [stream-open](#) 関数はイメージをストリームとして開きます。
- [add-mime-header](#) 関数は、「コンテンツ-ディスポジション」ヘッダーをストリームに追加します。このようにして、イメージ添付は想定された名前(「sample.png」)が与えられます。

最後にこれはバイナリファイルのため「添付」フィールドからの式が必要とされます。[mime-content-encode](#) 式関数を使用して転送中に保管されるためこのようなファイルは `base-64` としてエンコードされる必要があります。

## 10.5 /system/maintenance

/system/maintenance コンテナは、サーバー上でのオペレーションを管理するために使用される関数を含んでいます。

### 10.5.1 archive-log

フルパス: /system/maintenance/archive-log

古いログ記録をサーバー上のアーカイブファイルに移動します。作成されるアーカイブファイルの名前を文字列の値として返します。

#### パラメーター

| 名前               | 型       | 説明  |
|------------------|---------|---|
| 日、より古い           | number  | ここに入力された日数より古いアーカイブファイル。デフォルトの値は30です。                                       |
| アーカイブディレクトリ      | string  | アーカイブディレクトリ名 (例えば c:\temp)。必須。  |
| アーカイブファイルプレフィックス | string  | アーカイブファイルのプレフィックス。デフォルトは flowforcelog です。                                   |
| アーカイブされた記録の削除    | boolean | FlowForce データベースからアーカイブされた記録を削除します。   |
| 作業ディレクトリ         | string  | ジョブの作業ディレクトリを指定します (例えば c:\somedirectory)。相対的なパスが使用されると、作業ディレクトリに対して解決されます。 |

### 10.5.2 cleanup-files

フルパス: /system/maintenance/cleanup-files

使用されていないファイル、または、デプロイされたオブジェクト (MapForce マッピングと StyleVision 変換など) により参照されるファイルを削除します。数値として削除されたファイルの数を返します。

デプロイされたオブジェクトを削除、または、変更されたファイルを持つ既存のオブジェクトを再デプロイすると、以前デプロイされたオブジェクトに関連するファイルは未使用になります。デフォルトでは、FlowForce Server は未使用のファイルを削除しません。ですから、特に複数のユーザーがオブジェクトを FlowForce Server にデプロイする企業環境では、ディスク容量をクリーンアップするためには、この関数を定期的に呼び出すジョブを作成することが奨励されます。

デプロイされたオブジェクトにより使用されている現在のディスク容量を確認するには、FlowForce Server アプリケーションデータフォルダー内にあるファイルフォルダーのサイズを確認してください (以下を参照: [FlowForce Server アプリケーションデータ](#))。

この関数には、パラメーターが存在しません。

### 10.5.3 truncate-log

フルパス: /system/maintenance/truncate-log

与えられた日付より古いログ記録を削除します。削除された記録の数を数量の値で返します。

#### パラメーター

| 名前     | 型      | 説明  |
|--------|--------|---|
| 日、より古い | number | ここに入力された日数より古いアーカイブファイルを削除します。デフォルトの値は30です。 |

## 10.6 /system/shell

/system/shell コンテナには、シェルコマンド、または、スクリプトを実行する関数 `commandline` が含まれています。

### 10.6.1 `commandline`

フルパス: /system/shell/commandline

シェルコマンドまたはバッチファイルを実行します。

FlowForce Server ジョブご環境関数を読み込ませるには、スクリプト内で定義されている必要があります。またはスクリプトは /system/shell/commandline 関数により実行されなければなりません。FlowForce Server は、インタラクティブではシェルにより作動されています。これは (Linux 上の `executing .profile` または `.bashrc` などの) インタラクティブなシェル特有の振る舞いには適用することができない点に注意してください。

シェルコマンドからの終了コードが「0」より大きい場合、出力は、以下の通りです:

パラメータエラー時には中断が `true` (デフォルト) の場合、この関数は、実行を中断します。この場合、エラーを保護されたブロックとして処理することができます。(以下を参照: [ステップのエラーの処理](#))。パラメータエラー時には中断が `false` の場合、関数は、標準の出力、標準のエラーおよび終了コードを含むシェルコマンドの結果を返します。

最後のコマンドから終了コードが「0」(成功) の場合、関数は最後のシェルコマンドの結果を、ジェネリックな型として返します。この関数により返される値を、他のステップまたはジョブで処理するには、以下を行ってください!

1. 「このステップの結果を以下に割り当てる」テキストボックス内に値を入力して返される結果に名前をつけます。(例えば `myresult`)。
2. 必要とする戻り値の型により、関数 `compute` または `compute-string` を実行する新規ステップを作成します。
3. FlowForce ステップ結果関数を使用して。上記の関数の引数として、ジェネリックな結果から希望する部分を取得する式を入力します。例えば、式 `stdout(myresult)` を入力して、ストリームとしての結果の標準出力を取得する、または、標準エラー出力ストリームを取得するために `stderr(myresult)` を入力します。文字列と同じ値を取得するには、`content(stdout(myresult))` と `content(stderr(myresult))` をそれぞれ使用します。

`stdout` 関数 (ジョブ) は、シェルコマンドが標準の出力を返さない場合、失敗することにご注意してください。同様に `stderr` 関数は、標準のエラーがない場合でも失敗します。

詳細に関しては、以下を参照してください: [データ型をステップで処理する](#) と [ステップ結果関数](#)

#### パラメーター

| 名前         | 型                    | 説明  |
|------------|----------------------|---|
| コマンド       | <code>string</code>  | 実行するシェルコマンドを入力します。  |
| エラー時には中断する | <code>boolean</code> | このグローバルパラメーターはジョブが失敗した場合の戻り値を決定します。「エラー時には中断」が <code>FALSE</code> の場合、関数は、グローバル値の <code>FALSE</code> を返します。「エラー時には中断」が <code>TRUE</code> の場合、ジョブの実行は中断されます。デフォルトの値は |

| 名前       | 型                                | 説明   |
|----------|----------------------------------|--|
|          |                                  | TRUE です。   |
| 作業ディレクトリ | <code>string as directory</code> | ジョブの作業ディレクトリを指定します(例えば <code>c:\somedirectory</code> )。相対的なパスが使用されると、作業ディレクトリに対して解決されます。 |

### サンプル

次のジョブは **DoTransform.bat** と呼ばれる Windows バッチファイルを実行します。**DoTransform.bat** が XML ファイルを入力として必要とすると仮定し、入力 XML ファイルは作業ディレクトリにコピーされる必要があります。このサンプル内では、作業ディレクトリは以下のとおりです: `C:\codegen\xslt2`。

### 実行ステップ

**+**

関数の実行 `/system/shell/commandline`

|         |            |  |
|---------|------------|--|
| パラメーター: | コマンド:      | <input type="text" value="DoTransform.bat"/>   |
|         | エラー時には中断:  | <input style="border: 1px solid gray; border-radius: 50%;" type="button" value="+"/> |
|         | 作業用ディレクトリ: | <input type="text" value="C:\codegen\xslt2"/>  |

= ステップの結果を割り当て  以下として結果

次のジョブは XSLT 変換をパラメーターを使用して実行するために RaptorXML Server を呼び出します。PATH 環境変数は RaptorXML Server 実行可能ファイルへのパスを含んでいます。例えば `C:\Program Files (x86)\Altova\RaptorXMLServer2021\bin`。RaptorXML Server, に関する詳細は <https://www.altova.com/raptorxml> を参照してください。

`/system/shell/commandline` ▼ ↗

|       |   |
|-------|---|
| コマンド: | <input type="text" value="raptorsml xslt --input=c:/Test.xml --output=c:/Output.xml c:/Test.xslt"/> |
|-------|---|

コマンドラインにより返されけ出力を処理するステップのサンプルに関しては、[パスが存在するかの確認](#)を参照してください。

## 11 式関数

このセクションでは、FlowForce 式関数に関するリファレンスについて説明されています。式の使用方法を理解する概念に関する情報に関しては、[FlowForce 式](#)を参照してください。

式関数は以下に示されるようにグループ化されています:

- [ステップ結果関数](#)
- [ストリーム関数](#)
- [ファイルシステム関数](#)
- [ファイルパス関数](#)
- [リスト関数](#)
- [文字列関数](#)
- [ブール値関数](#)
- [ランタイム情報関数](#)
- [MIME 式関数](#)

## 11.1 ステップ結果関数

ステップ結果関数は、ジョブより返された結果(または、ジョブ内での実行ステップより返された結果)を処理します。

### 11.1.1 error-message

この関数はステップが遭遇するエラーメッセージのテキストを返します。この関数は典型的に保護されたブロックの内部、具体的には「エラー時」のレンダーで使用されます。エラーに遭遇しない場合、またはジョブの性質からエラーのテキストを技術的に抽出できない場合、関数は空の文字列を返します。

#### 署名

```
error-message(result:result) -> string
```

ジェネリックな署名

#### パラメーター

| 名前     | 型      | 説明   |
|--------|--------|--|
| result | result | エラーテキストが抽出されるエラーのあるステップを提供します。エラーのあるステップに移動するには failed-step() 関数を呼び出します。 |

#### サンプル

[エラー処理をジョブに追加する](#)を参照してください。

### 11.1.2 exitcode

結果の数値終了コードを返します。

#### 署名

```
exitcode(result:result) -> number
```

ジェネリックな署名

#### パラメーター

| 名前     | 型      | 説明                  |
|--------|--------|---------------------|
| result | result | 取得する終了コードを持つステップの結果 |

## サンプル

次のジョブは2つのステップから構成されています。最初のステップは現在の作業ディレクトリ内 (C:\FlowForce) に **data** と呼ばれるディレクトリを作成しようとする Windows コマンドラインコマンドを呼び出します。このステップの結果は **outcome** として宣言されます。2番目のステップは **outcome** を取得し、`exitcode` 関数の手助けを受け、数値の終了コードを返します。`string` 関数の助けを受け、数値の終了コードは文字列に変換されます。式のデータ型が文字列のためこの変換が必要になります。

更に重要なことは、エラー時に中断 オプションが選択されていないことです。それ以外の場合、実行はエラー発生時に中断され、処理のための2番目のステップのための終了コードは存在しません。

The screenshot shows a configuration window titled "実行ステップ" (Execution Steps). It contains two step configurations. The first step is "関数の実行 /system/shell/commandline". Its parameters are: "コマンド: mkdir data", "エラー時には中断: +", and "作業用ディレクトリ: C:\FlowForce". Below the parameters, it shows "ステップの結果を割り当て outcome" (Assign step result to outcome). The second step is "関数の実行 /system/compute-string". Its condition is "The exit code is:{string(exitcode(outcome))}".

ジョブが最初に実行される場合 **data** ディレクトリの作成が成功すると、終了コードは **0** になります。次の実行では、ディレクトリは既存のため、終了コードは **1** になります。

[エラー処理をジョブに追加する](#)を参照してください。

### 11.1.3 failed-step

ステップ結果を返します。保護されたブロックを有するエラーを処理する際この関数は役に立ちます。failed-step 関数は、「エラー時」ハンドラーの一部である必要があります。それ以外の場合、ステップを使用している箇所で、エラーのステップが存在しないため、失敗します。

このように、この関数は、エラーが発生したステップを指しません。属性の結果を確認するには、この関数を `stdout` または `stderr` などの関数に引数として呼び出します。例

```
stderr(failed-step())
stdout(failed-step())
```

## 署名

```
failed-step() -> result
```

ジェネリックな署名

## サンプル

以下の例を参照してください。

- [ジョブエラー処理を追加する](#)
- [エラー処理を使用してXMLドキュメントを検証する](#)

### 11.1.4 results

オプションで名前でフィルターされた指定された結果のストリームの配列を返します。関数 `nth` を使用して配列内の特定の値にアクセスします。

## 署名

```
results(result:result, name:string) -> array of stream
```

ジェネリックな署名

## パラメーター

| 名前                  | 型                   | 説明   |
|---------------------|---------------------|--|
| <code>result</code> | <code>result</code> | 必須のパラメーターです。ストリームの配列を返すステップの結果です。          |
| <code>name</code>   | <code>string</code> | 任意のパラメーターです。与えられると、結果内で特定な値を名前に従いフィルターします。 |

## サンプル

サンプルA。FlowForce Server へ出力として単一のXML ファイルを生成するマッピングをデプロイしたと仮定します。このようなマッピングのサンプルは **CompletePO.mfd** で MapForce サンプルに含まれています。MapForce 内のターゲット XML コンポーネント名は CompletePO です。このマッピングの結果を処理し、FlowForce からのファイルに保存する場合は、ジョブを次のように構成します:

### 実行ステップ

**関数の実行** /public/CompletePO.mapping

パラメーター:

|                    |          |   |
|--------------------|----------|---|
| Customers:         | (input)  | + |
| Articles:          | (input)  | + |
| ShortPO:           | (input)  | + |
| CompletePO:        | (output) | + |
| Working-directory: |          | + |

= ステップの結果を割り当て  × 以下としてCompletePO

**関数の実行** /system/filesystem/copy

パラメーター:

|            |  |
|------------|--|
| ソース:       | {as-file(nth(results(mapping_result), 0))} |
| ターゲット:     | output.xml                                 |
| 上書き:       | +  |
| エラー時には中断:  | +  |
| 作業用ディレクトリ: | +  |

= ステップの結果を割り当て  以下としてブール値

上記のジョブ構成では、最初のステップはマッピングを実行し、結果を **mapping\_result** として返します。2番目のステップでは、式は

```
{as-file(nth(results(mapping_result), 0))}
```

**mapping\_result** を処理し、ファイルに変換します。具体的に **results** 関数は、MapForce コンポーネントからストリームの配列を選びます。**nth** 関数は、この配列から最初のアイテムを選びます。**as-file** 関数はストリームからファイルを生成します。

**copy** 関数は生成されたファイルを作業ディレクトリにコピーします。**Target** テキストボックスは生成されたファイルの名前を定義します。同じ名前の既存のファイルは上書きされます。

**サンプルB.** FlowForce Server に2つのターゲット XML コンポーネント「MarketingExpenses」と「DailyExpenses」が存在するマッピングをデプロイしたと仮定します。このようなマッピングのサンプルは **MarketingAndDailyExpenses.mfd** で MapForce サンプルに含まれています。「DailyExpenses」コンポーネントからファイルを生成するには、上記に類似したジョブを作成し、式を以下に変更します:

```
{as-file(nth(results(mapping_result, 'DailyExpenses'), 0))}
```

ここで唯一の違いはマッピングにより生成されたストリームの配列が希望するコンポーネントの名前によりフィルタされていることです(この場合「DailyExpenses」)。

**サンプルC.** FlowForce Server に複数のXML ファイルを動的に生成するマッピングをデプロイしたと仮定します。出力ファイル名はマッピング自身により生成され、ランタイムの前は不明です。このようなマッピングのサンプルは

**DividePersonsByDepartmentIntoGroups.mfd** で MapForce サンプルに含まれています。マッピングの3番目の出力ファイルを作成するには、上記に類似したジョブを作成し、式を以下に変更します:

```
{as-file(nth(results(mapping_result), 2))}
```

ここで、3番目のファイルが必要になるため、nth 関数の2番目引数として与えられたインデックスは、インデックスがゼロベースのため（3ではなく）2です。

以下の例を参照してください:

- [StyleVision 変換からジョブを作成する](#)
- [複数のXML から複数のPDF を作成する](#)

### 11.1.5 retry-count

FlowForce が再試行を行ってもエラー成功の処理（いわゆる「保護されたブロック」）を持つ場合は複数のステップの回数を示す数値を返します。関数は特に関数を囲む最も内側の保護されたブロックを評価することにご注意してください。再試行が行われなかった場合（すなわち、保護されたブロックの最初の実行に成功した場合）戻り値は0です。[エラー時に再度試す](#)も参照してください。

#### 署名

```
retry-count() -> number
```

ジェネリックな署名

### 11.1.6 stdout

（シェルコマンドを実行する）などの実行ステップの一部は、標準の出力を返します。例えば、シェルコマンド `dir` (Windows) はディレクトリのリストを返します。

ステップが結果を返すと、FlowForce Server は、自動的に結果を一般的な型の `result` に割り当てます。stdout 関数に関しては、`result` の標準の出力は、以下の通りです:

```
stdout(result)
```

**result** は実行ステップより返された値です。

この関数は、`result` が標準の結果を与えない場合失敗します。

#### 署名

```
stdout(result:result) -> stream
```

ジェネリックな署名

## パラメーター

| 名前     | 型      | 説明               |
|--------|--------|------------------|
| result | result | 標準エラーを得るステップの結果。 |

## サンプル

以下の例を参照してください。

- [ジョブエラー処理を追加する](#)
- [エラー処理を使用してXMLドキュメントを検証する](#)
- [パスの存在を確認する](#)

## 11.1.7 stderr

結果の標準エラーを返します。結果が標準のエラーを返さない場合失敗します。

## 署名

```
stderr(result:result) -> stream
```

ジェネリックな署名

## パラメーター

| 名前     | 型      | 説明               |
|--------|--------|------------------|
| result | result | 標準エラーを得るステップの結果。 |

## サンプル

例に関しては、[ジョブエラー処理を追加する](#)を参照します。

## 11.2 ストリーム関数

ストリーム関数は、データのストリームを処理するために使用されます。ストリームを Web サービスまたはステップ結果から FlowForce Server にパスすることができます。

### 11.2.1 as-file

ストリームソースがファイルの場合ファイルを作成します。ストリームソースがファイルではない場合、一時ファイルを作成します。

#### 署名

```
as-file(stream:stream) -> string
```

ジェネリックな署名

#### パラメーター

| 名前     | 型      | 説明              |
|--------|--------|-----------------|
| stream | stream | ストリームソースを指定します。 |

#### サンプル

次のジョブはテキスト1行を持つ **file.txt** と呼ばれるファイルを作成します。最初に [stream-from-string](#) 関数は、引数として提供されたテキストからストリームを生成します。次に [as-file](#) 関数は、ストリームを引数としてとり、一時的なファイルを生成します。一時的なファイルを永続的なパスにコピーするには、内蔵の [copy](#) 関数が実行ステップから呼び出されます。ジョブ (C:\FlowForce) の作業ディレクトリファイルはコピーされ、ジョブが作動する都度上書きされます。

### 実行ステップ

**+**

関数の実行  ▼

パラメーター: 条件式: `as-file(stream-from-string('This is my file content.))`

= ステップの結果を割り当て  × 以下としてT0

**+**

関数の実行  ▼

パラメーター:

|            |   |
|------------|---|
| ソース:       | <input style="width: 80%;" type="text" value="{result}"/>           |
| ターゲット:     | <input style="width: 80%;" type="text" value="file.txt"/>           |
| 上書き:       | <input checked="" type="checkbox"/>                                 |
| エラー時には中断:  | <input style="width: 20px; height: 20px;" type="button" value="+"/> |
| 作業用ディレクトリ: | <input style="width: 80%;" type="text" value="C:\FlowForce"/>       |

[エラーログのあるXMLドキュメントを検証する](#)も参照してください。

## 11.2.2 content

指定されたストリームのコンテンツを指定されたエンコード内のテキストとして読み取ります。

### 署名

```
content(stream:stream, encoding:string="UTF-8") -> string
```

ジェネリックな署名

### パラメーター

| 名前       | 型      | 説明                                     |
|----------|--------|--|
| stream   | stream | ストリームソースを指定します。                        |
| encoding | string | 使用するエンコードを指定します。デフォルトのエンコードは'UTF-8'です。 |

### サンプル

以下の例を参照してください。

- [エラー処理をオブジェクトに追加する](#)

### 11.2.3 empty-stream

空のストリームを作成します。

#### 署名

```
empty-stream() -> stream
```

ジェネリックな署名

### 11.2.4 stream-from-string

与えられたエンコードを使用する文字列からストリームを作成します。引数として与えられたコンテンツの型はストリームに関連しています。このストリームの型は自動的にファイルとして保存されません。

#### 署名

```
stream-from-string(string:string, encoding:string="UTF-8",  
contenttype:string=contenttype=text/plain) -> stream
```

ジェネリックな署名

#### パラメーター

| 名前          | 型      | 説明  |
|-------------|--------|---|
| string      | string | ストリームを作成する文字列   |
| encoding    | string | 使用するエンコードを指定します。デフォルトのエンコードは'UTF-8'です。                              |
| contenttype | string | ストリームに関連させるために contenttype を指定します。デフォルトは contenttype=text/plain です。 |

### 11.2.5 stream-open

既存のファイルからストリームを作成します。

#### 署名

```
stream-open(name:string, contenttype:string=contenttype=application/octet-stream) ->  
stream
```

ジェネリックな署名

## パラメーター

| 名前          | 型      | 説明  |
|-------------|--------|---|
| name        | string | このストリームのためのソースファイルのパス   |
| contenttype | string | ストリームに関連させるために contenttype を指定します。デフォルトは contenttype=application/octet-stream です。 |

## サンプル

次のジョブは、.txt 拡張子を持つ既存のファイルを開き、.csv 拡張子を持つ同じディレクトリに書き込みます。

### 実行ステップ

▼
🔗

関数の実行 /system/filesystem/copy

|         |            |  |
|---------|------------|--|
| パラメーター: | ソース:       | <code>{as-file(stream-open("C:\FlowForce\file.txt"))}</code>                         |
|         | ターゲット:     | C:\FlowForce\file.csv  |
|         | 上書き:       | <span style="border: 1px solid gray; border-radius: 50%; padding: 2px 5px;">+</span> |
|         | エラー時には中断:  | <span style="border: 1px solid gray; border-radius: 50%; padding: 2px 5px;">+</span> |
|         | 作業用ディレクトリ: | <span style="border: 1px solid gray; border-radius: 50%; padding: 2px 5px;">+</span> |

## 11.3 ファイルシステム関数

ファイルシステム関数は、ファイルシステムへのアクセスを許可します。これらの関数を実行するために、ジョブはオペレーティングシステムの対応するアクセス権を使用してユーザーアカウントの資格情報を使用する必要があります。

### 11.3.1 list-files

(ワイルドカードにより中断される可能性のある) / 内のファイルを一覧し、結果の文字列の一覧を返します。

/ が / 区切り文字、または、ワイルドカードで終了しない場合、指定されたアイテムの検索が親ディレクトリ内で行われます。

#### 署名

```
list-files (path:string) -> list of string
```

ジェネリックな署名

#### パラメーター

| 名前   | 型      | 説明                          |
|------|--------|-----------------------------|
| path | string | ディレクトリ、または、ファイルへの / を指定します。 |

#### サンプル

サンプルに関しては [ファイルのエピソード](#) を参照してください。

### 11.3.2 list-directories

(ワイルドカードにより中断される可能性のある) / 内のサブディレクトリを一覧し、結果の文字列の一覧を返します。

#### 署名

```
list-directories (path:string) -> list of string
```

ジェネリックな署名

#### パラメーター

| 名前   | 型      | 説明                 |
|------|--------|--------------------|
| path | string | ディレクトリへの / を指定します。 |

### 11.3.3 read-lines

与えられたファイルからラインを読み取り、文字列のリストとして返します。返される文字列は(\n などの)行末を含みます。下のサンプルで説明されている通り、処理前にtrim() 関数を使用して各ラインをトリムする必要がある場合があります。

#### 署名

```
read-lines(filename:string, encoding:string="UTF-8") -> list of string
```

ジェネリックな署名

#### パラメーター

| 名前       | 型      | 説明                                     |
|----------|--------|--|
| filename | string | ファイルへのパスを指定します。                        |
| encoding | string | 使用するエンコードを指定します。デフォルトのエンコードは'UTF-8'です。 |

#### サンプル

FlowForce Server がインストールされているコンピュータ上の複数のディレクトリ内にある複数のファイルを処理する必要があると仮定します。ディレクトリパスは各ラインがディレクトリパスに対応する箇所でテキストファイルとして保存されています。例:

```
C:\FlowForce\Examples\ListDirectories\1
C:\FlowForce\Examples\ListDirectories\2
C:\FlowForce\Examples\ListDirectories\3
```

下で説明されているジョブは2つのステップから構成されています。最初のステップはread-files 関数を呼び出し、全てのディレクトリパスを上のテキストファイルから集めリストにします。2番目のステップはパスのリストを反復しlist-files 関数を呼び出します。パスは処理される前にトリムされます。結果する文字列が空白文字または新規ライン文字を含んでいないことにご注意してください。

### Execution Steps

- + Execute function `/system/compute`
  - Parameters: Expression: `read-lines('C:\FlowForce\Examples\ListDirectories\paths.txt')`
  - = Assign this step's result to `lines` as T0
- + For each `item` in sequence `lines`
  - + Execute function `/system/compute`
    - Parameters: Expression: `list-files(trim(item))`
    - = Assign this step's result to `name` as T0

このジョブを Web サービスとして公開し、デフォルトのアドレスとポートでブラウザからアクセスするとブラウザは各ディレクトリのエントリを例えば JSON 配列として出力します:

```
File Edit View History Bookmarks Tools Help
127.0.0.1:4646/service/ListDirectories X +
127.0.0.1:4646/service/ListDirectories
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
0:
  0: "C:\\FlowForce\\Examples\\ListDirectories\\1\\A.txt"
  1: "C:\\FlowForce\\Examples\\ListDirectories\\1\\B.txt"
1:
  0: "C:\\FlowForce\\Examples\\ListDirectories\\2\\C.txt"
  1: "C:\\FlowForce\\Examples\\ListDirectories\\2\\D.txt"
2:
  0: "C:\\FlowForce\\Examples\\ListDirectories\\3\\E.txt"
  1: "C:\\FlowForce\\Examples\\ListDirectories\\3\\F.txt"
```

## 11.4 ファイルパス関数

ファイルパス関数は、パスとファイル名の特定の一部分を抽出するために使用されます。例えば、ディレクトリをポールドまたは `triggerfile` パラメーターからジョブをトリガー下ファイル名を抽出する場合、この操作を行う必要がある場合があります ([ファイルシステムトリガー](#) を参照してください)。

### 11.4.1 extension

パスからファイル拡張子を抽出します。

#### 署名

```
extension (path:string) -> string
```

ジェネリックな署名

#### パラメーター

| 名前   | 型      | 説明              |
|------|--------|-----------------|
| path | string | ファイルへのパスを指定します。 |

#### サンプル

次の式は ".txt" を返します：

```
extension("c:\temp\file.txt")
```

### 11.4.2 filename

パスから (拡張子の無い) ファイル名と拡張子を抽出します。

#### 署名

```
filename (path:string) -> string
```

ジェネリックな署名

#### パラメーター

| 名前   | 型      | 説明              |
|------|--------|-----------------|
| path | string | ファイルへのパスを指定します。 |

### サンプル

次の式はfile"を返します:

```
filename("c:\temp\file.txt")
```

## 11.4.3 filename-with-extension

パスからファイル名と拡張子を抽出します。

### 署名

```
filename-with-extension(path:string) -> string
```

ジェネリックな署名

### パラメーター

| 名前   | 型      | 説明              |
|------|--------|-----------------|
| path | string | ファイルへのパスを指定します。 |

### サンプル

次の式はfile.txt"を返します:

```
filename-with-extension("c:\temp\file.txt")
```

## 11.4.4 join-paths

パス引数として提供されるパスを結合します。

### 署名

```
join-paths(string1:string, string2:string, stringN:string) -> string
```

### パラメーター

| 名前      | 型      | 説明  |
|---------|--------|---|
| string1 | string | ジョインする単一のパスステップを指定します。全ての続き引数はコンマにより区切られます。 |
| string2 | string | 上記と同様です。                                    |

| 名前             | 型                   | 説明       |
|----------------|---------------------|----------|
| <b>stringN</b> | <code>string</code> | 上記と同様です。 |

## サンプル

Windows 上では次の式が“C:\tmp\test.txt”を返します:

```
join-paths('C:\tmp', 'test.txt')
join-paths('C:\tmp\'', 'test.txt')
join-paths('C:\', 'tmp', 'test.txt')
join-paths('C:\Users', '\tmp', 'test.txt')
join-paths('D:\Data', 'C:\tmp', 'test.txt')
```

Linux と MacOS では次の式が“/home/user/test.txt”を返します:

```
join-paths('/home/user', 'test.txt')
join-paths('/var', '/home/user', 'test.txt')
```

## 11.4.5 parent-directory

パスから親ディレクトリを抽出します。

### 署名

```
parent-directory(path:string) -> string
```

ジェネリックな署名

### パラメーター

| 名前          | 型                   | 説明                |
|-------------|---------------------|-------------------|
| <b>path</b> | <code>string</code> | ディレクトリへのパスを指定します。 |

## サンプル

XML ファイルからデータベースを更新する MapForce マッピングが存在する前提です。FlowForce Server には既にデプロイされており、ジョブも既に作成されています。また、ディレクトリが変更されると実行されるジョブ構成済みですつまり、ジョブはファイルシステムトリガーを使用します。[ファイルシステムトリガー](#)を参照してください。

ジョブの最初のステップは、データベースを更新するマッピングを実行します。

|   |  |
|---|--|
| Execute function /public/Tutorials/Altova_Hierarchical_DB.mapping |  |
| Parameters:   | Altova_Hierarchical: (input)   |
|   | Working-directory: C:\FFSERV   |
| Assign this step's result to <input type="text" value="name"/>    |  |

マッピングステップが実行を完了すると、目的は、ソースXML ファイルを“processed” という名前のサブディレクトリに移動することです。これを行うことにより、どのファイルが処理済みかを管理することができます。このゴールを達成するために、`/system/filesystem/move` 関数を呼び出す新規のステップを追加し、ソースとターゲットとして下の値を入力します:

|  |  |
|--|--|
| Execute function /system/filesystem/move |  |
| Parameters:                              | Source: <input data-bbox="548 724 669 756" type="text" value="{triggerfile}"/>                                 |
|  | Destination: <input data-bbox="548 772 974 804" type="text" value="{parent-directory(triggerfile)}processed"/> |
|  | Overwrite target: <input checked="" type="checkbox"/>  |
|  | Abort on error:               |
|  | Working directory: C:\FFSERV   |

ソースフィールドのパラメータ値 `{triggerfile}` は、FlowForce にマッピングをトリガーしたファイルを移動するように命令します。  
パラメータ値

```
{parent-directory(triggerfile)}processed
```

保存先フィールドが現在のディレクトリ内の“processed” という名前のディレクトリに設定します。これは、式と文字列により構成されています。式の部分が中かっこにより区切られていることにご注意ください。(以下を参照: [文字列フィールド内に式を埋め込む](#))。式は

```
{parent-directory(triggerfile)}
```

親ディレクトリ関数を呼び出し、引数として値“triggerfile”を与えます。

ですから、ジョブが実行されると、次のアクションが実行されます:

1. スクリプト、または、ユーザーがファイルを現在の作業ディレクトリに(例えば、`C:\FFSERV`)コピーします `source.xml`。
2. トリガーが実行され `source.xml` は“triggerfile”になります。
3. FlowForce Server は、マッピングを作動するステップを実行します。
4. FlowForce Server は、`source.xml` を“processed” サブディレクトリに移動するステップを実行します。パス `C:\FFSERV\processed` 存在しなくてはなりませんことにご注意ください。

## 11.5 リスト関数

リスト関数はリストを作成、および分解するために使用されます。リストは常に、単一の型のアイテムを含んでいます（例えば、文字列のみ、数値のみ、ネストされた同じアイテムの型のリストなど）。

### 11.5.1 char

引数として与えられた数字の Unicode 文字を含む文字列を返します。例えば、char(10) は改行を返します。特定の Unicode 文字の数値コードを見つけるにはコード関数を使用してください。

#### 署名

```
char(number: number) -> string
```

ジェネリックな署名

#### パラメーター

| 名前     | 型      | 説明   |
|--------|--------|--|
| number | number | 文字の数値コードこのコードはHTML 内で Unicode 文字を表すために使用される数値コードと同様です(例えば 8734 は無限のシンボルを表します)。 |

#### サンプル

次の実行ステップは無限シンボルを返します:

### 実行ステップ

+ 関数の実行 /system/compute

パラメーター: 条件式: char(8734)

= ステップの結果を割り当て 名前

+ 新規の実行ステップ

+ 新規の選択ステップ

+ 新規の For-e

### 実行の結果

戻り値の型を以下と宣言する: 文字列 ▼

## 11.5.2 code

引数として与えられた文字列の最初の文字のUnicode 値を返します。

### 署名

```
code (string:string) -> number
```

ジェネリックな署名

### パラメーター

| 名前     | 型      | 説明             |
|--------|--------|----------------|
| string | string | 入力ストリームを指定します。 |

### サンプル

次の実行ステップはスペース文字を表す数値 32 を返します:

### 実行ステップ

+

- 関数の実行 /system/compute
  - パラメーター: | 条件式: code("")
  - = ステップの結果を割り当て 名前  以下としてT0

新規の実行ステップ
新規の選択ステップ
新規の For-each ステップ
新規エラー

### 実行の結果

戻り値の型を以下と宣言する: 数値 ▼

### 11.5.3 from-to

“from” と “to” の間の整数のリストを返します。“from” が “to” より大きい場合、このリストは空です。

#### 署名

```
from-to(from:number, to:number) -> list of number
```

ジェネリックな署名

#### パラメーター

| 名前   | 型      | 説明                      |
|------|--------|-------------------------|
| from | number | 開始インデックス(“from”)を指定します。 |
| to   | number | 終了インデックス(“to”)を指定します。   |

#### サンプル

次の式は [3, 4, 5, 6, 7] を生成します。

```
from-to(3, 7)
```

### 11.5.4 join

2番目のリストを各リストのペアの区切りとして使用し、最初の引数のリストを結合します。

#### 署名

```
join(lists:list of lists, separator:list=empty list) -> list
```

ジェネリックな署名

#### パラメーター

| 名前        | 型             | 説明   |
|-----------|---------------|--|
| lists     | list of lists | ジョインするリストを指定します。引数は2つ以上のリストのリストである必要があります。ネストされているすべてのリストは同じ型である必要があります。 |
| separator | list          | ジョインされたリストを区切るセレーターを指定する任意の引数与えられていない場合、セレーターは使用されません。                   |

| 名前 | 型 | 説明   |
|----|---|--|
|    |   | セレータは型 リストである必要があります。リスト 関数を使用してセレータを作成します。例えば式 <code>list(',')</code> は単一のコンマ文字をセレータとして指定します。 |

### サンプル

次の実行ステップは2つのリストをジョインする方法を説明しています。ステップ1 は最初のリストを作成します。ステップ2 は2番目のリストを作成します。ステップ3 は両方のリストを含む「リストのリスト」のオブジェクトを作成します。最後に、ステップ4 はセミコンマ文字をセレータとして使用しリストをジョインします。

### 実行ステップ

+

▾
関数の実行 /system/compute

パラメーター:
条件式: `list('a', 'b')`

=
ステップの結果を割り当て list1
×
以下としてT0

+

▾
関数の実行 /system/compute

パラメーター:
条件式: `list('c', 'd')`

=
ステップの結果を割り当て list2
×
以下としてT0

+

▾
関数の実行 /system/compute

パラメーター:
条件式: `list(list1, list2)`

=
ステップの結果を割り当て list\_of\_lists
×
以下としてT0

+

▾
関数の実行 /system/compute

パラメーター:
条件式: `join(list_of_lists, list(','))`

=
ステップの結果を割り当て 名前
×
以下としてT0

### 11.5.5 length

リスト内のアイテムの数を返します。

#### 署名

```
length(list:list) -> number
```

ジェネリックな署名

#### パラメーター

| 名前   | 型    | 説明                 |
|------|------|--------------------|
| list | list | 入力リストオブジェクトを指定します。 |

### 11.5.6 list

単一アイテムのリストを構築します。全てのアイテムは同じ型である必要があり、結果のリストはその型のアイテムのリストです。

#### 署名

```
list(item1:any type, item2:any type, itemN:any type) -> list
```

ジェネリックな署名

#### パラメーター

| 名前    | 型        | 説明                                       |
|-------|----------|--|
| item1 | any type | 単一のアイテムを指定します。次のアイテムはコンマにより区切られる必要があります。 |
| item2 | any type | 上記と同様です。                                 |
| itemN | any type | 上記と同様です。                                 |

#### サンプル

次の式はリスト [1, 2, 3] を返します。すべてのリストアイテムは数値の型です:

```
list(1,2,3)
```

次の式はリスト ['a', 'b', 'c'] を返します。すべてのリストアイテムは文字列の型です:

```
list('a','b','c')
```

### 11.5.7 nth

リストから指定されたアイテムを返します。インデックスは、ゼロベースです。インデックスが範囲外の場合は失敗します。

#### 署名

```
nth(list:list, index:number) -> item
```

ジェネリックな署名

#### パラメーター

| 名前    | 型      | 説明                          |
|-------|--------|-----------------------------|
| list  | list   | 入力リストを指定します。                |
| index | number | 返されるアイテムのゼロベースインデックスを指定します。 |

#### サンプル

次の式は"b"を返します:

```
nth(list('a', 'b', 'c'), 1)
```

### 11.5.8 slice

リストからリストの一部を返します。

#### 署名

```
slice(list:list, start:number, end:number=length(list)) -> list
```

ジェネリックな署名

#### パラメーター

| 名前    | 型      | 説明                                       |
|-------|--------|--|
| list  | list   | 入力リストを指定します。                             |
| start | number | スライス内に含まれる最初のリストアイテムのゼロベースのインデックスを指定します。 |
| end   | number | スライス内で無視される最初のアイテムのゼロベースのインデックスを指定します。   |

### サンプル

次の式はリスト(2,3)を返します:

```
slice(list(1,2,3,4),1,3)
```

## 11.6 文字列関数

文字列関数は結合、文字列からのサブ文字列の抽出、トリミング、分割などの基本的な文字列のオペレーションを行います。

### 11.6.1 concat

引数として与えられた文字列を単一の文字列に結合します。`string-join` 関数を使用して、型“文字列のリスト”のオブジェクトのすべてのアイテムを結合します。

#### 署名

```
concat(string1:string, string2:string, stringN:string) -> string
```

ジェネリックな署名

#### パラメーター

| 名前      | 型      | 説明  |
|---------|--------|---|
| string1 | string | ジョインする単一の文字列を指定します。次のアイテムはコンマにより区切られる必要があります。 |
| string2 | string | 上記と同様です。                                      |
| stringN | string | 上記と同様です。                                      |

#### サンプル

次の式は“file”を返します：

```
concat('a', 'b', 'c')
```

### 11.6.2 contains

最初の文字列がサブstringの発生を1度でも含む場合、TRUE を返します。

#### 署名

```
contains(string:string, substring:string) -> Boolean
```

ジェネリックな署名

### パラメーター

| 名前                     | 型                   | 説明          |
|------------------------|---------------------|-------------|
| <code>string</code>    | <code>string</code> | 入力文字列       |
| <code>substring</code> | <code>string</code> | チェックする文字列の値 |

### サンプル

次の式は**true**を返します:

```
contains('cat', 'a')
```

次の式は**false**を返します:

```
contains('cat', 'b')
```

## 11.6.3 ends-with

**string** 引数内で与えられた文字列が **end** 引数内で与えられた文字列で終わる場合、**true** を返します。

### 署名

```
ends-with(string:string, end:string) -> Boolean
```

ジェネリックな署名

### パラメーター

| 名前                  | 型                   | 説明          |
|---------------------|---------------------|-------------|
| <code>string</code> | <code>string</code> | 入力文字列       |
| <code>end</code>    | <code>string</code> | チェックする文字列の値 |

### サンプル

次の式は**true**を返します:

```
ends-with('cat', 't')
```

次の式は**false**を返します:

```
ends-with('cat', 'a')
```

### 11.6.4 find-all

`pattern` が正規表現の箇所、文字列内の全ての `pattern` の発生を抽出します。

#### 署名

```
find-all(string:string, pattern:string) -> list of string
```

ジェネリックな署名

#### パラメーター

| 名前                   | 型                   | 説明           |
|----------------------|---------------------|--------------|
| <code>string</code>  | <code>string</code> | 入力文字列        |
| <code>pattern</code> | <code>string</code> | 正規表現としてのパターン |

#### サンプル

次の式は文字列 "apollo" からのすべての "o" の発生を抽出します。

```
find-all('apollo', 'o')
```

結果は文字列の次のリストです: ["o", "o"]

### 11.6.5 number

文字列の数字表記を計算します。例えば、引数として与えられている文字列を数値に変換します。

#### 署名

```
number(string:string) -> number
```

ジェネリックな署名

#### パラメーター

| 名前                  | 型                   | 説明          |
|---------------------|---------------------|-------------|
| <code>string</code> | <code>string</code> | 変換する入力文字列の値 |

#### サンプル

次の式は文字列の値 "1" を数値 1 に変換します:

```
number('1')
```

### 11.6.6 split

引数として与えられているセレーターの発生ごとに文字列を分割します。

#### 署名

```
split(string:string, separator:string) -> list of string
```

ジェネリックな署名

#### パラメーター

| 名前        | 型      | 説明       |
|-----------|--------|----------|
| string    | string | 入力文字列    |
| separator | string | セレーター文字列 |

#### サンプル

次の式はリスト["1", "2", "3"]を返します:

```
split('1;2;3', ';')
```

### 11.6.7 starts-with

文字列 引数内で与えられた文字列が開始 引数内で与えられた文字列で終わる場合、trueを返します。

#### 署名

```
starts-with(string:string, start:string) -> Boolean
```

ジェネリックな署名

#### パラメーター

| 名前     | 型      | 説明          |
|--------|--------|-------------|
| string | string | 入力文字列       |
| start  | string | チェックする文字列の値 |

### サンプル

次の式はtrueを返します:

```
starts-with('cat', 'c')
```

次の式はfalseを返します:

```
starts-with('cat', 'b')
```

## 11.6.8 string

与えられた数値の文字列表示を計算します。例えば、引数として与えられている数値を文字列に変換します。

### 署名

```
string(number:number) -> string
```

ジェネリックな署名

### パラメーター

| 名前     | 型      | 説明         |
|--------|--------|------------|
| number | number | 文字列に変換する数値 |

### サンプル

次の式は数値"1"を文字列 "1"に変換します:

```
string(1)
```

## 11.6.9 string-join

引数として与えられた文字列のリストを単一の文字列にジョインします。任意で、各文字列間に引数として与えられているセレーターを挿入します。

### 署名

```
string-join(list:list of string, separator:string="") -> string
```

ジェネリックな署名

### パラメーター

| 名前        | 型              | 説明                                    |
|-----------|----------------|---------------------------------------|
| list      | list of string | 文字列の入力リスト                             |
| separator | string         | 任意の引数すべてのジョインされている文字列を区切るセレーターを指定します。 |

### サンプル

次の式は文字列a;b;cを返します。

```
string-join(list('a', 'b', 'c'), ';')
```

## 11.6.10 string-length

文字列内の文字の数量を返します。

### 署名

```
string-length(文字列:string) -> number
```

ジェネリックな署名

### パラメーター

| 名前  | 型      | 説明    |
|-----|--------|-------|
| 文字列 | string | 入力文字列 |

### サンプル

次の式は3を返します:

```
string-length('cat')
```

## 11.6.11 substring

指定された、開始 文字のポジションから、終了 文字のポジションまでの文字列 からサブ文字列を返します。開始 と終了 インデックスはゼロベースです。

設定されていない場合はend は与えられた文字列の長さです。

end 引数は、負の整数であることもできます。負の値 -n は「文字列から最後のn 文字をトリムすること」を意味します。

## 署名

```
substr(string:sting, start:number, end:number) -> string
```

ジェネリックな署名

## パラメーター

| 名前            | 型             | 説明             |
|---------------|---------------|----------------|
| <b>string</b> | <b>sting</b>  | 入力文字列          |
| <b>start</b>  | <b>number</b> | ゼロベースの開始インデックス |
| <b>end</b>    | <b>number</b> | ゼロベースの終了インデックス |

## サンプル

次の式は“Force”を返します:

```
substr('FlowForce',4)
```

次の式は“t”を返します:

```
substr('Altova',2,3)
```

次の式は“tov”を返します:

```
substr('Altova',1,-1)
```

## 11.6.12 trim

先頭および末尾スペースを文字列から削除します(スペース、タブ、開業、リターン、フォームフィード、および垂直タブ)。

## 署名

```
trim(string:string) -> string
```

ジェネリックな署名

## パラメーター

| 名前            | 型             | 説明    |
|---------------|---------------|-------|
| <b>string</b> | <b>string</b> | 入力文字列 |

### 11.6.13 trim-start

引数として与えられている文字列から先頭の空白を削除します(トリム関数も参照してください)

#### 署名

```
trim-start(string:string) -> string
```

ジェネリックな署名

#### パラメーター

| 名前     | 型      | 説明    |
|--------|--------|-------|
| string | string | 入力文字列 |

### 11.6.14 trim-end

引数として与えられている文字列から末尾の空白を削除します(トリム関数も参照してください)

#### 署名

```
trim-end(string:string) -> string
```

ジェネリックな署名

#### パラメーター

| 名前     | 型      | 説明    |
|--------|--------|-------|
| string | string | 入力文字列 |

## 11.7 ブール値関数

TRUE / FALSE 式を検証するためブール値関数を使用します。

### 11.7.1 all

全てのブール値がtrueの場合 true を返します。最初のfalse 値の後に評価を停止し、falseを返します。

#### 署名

```
all (boolVal1: Boolean, boolVal2: Boolean, boolValN: Boolean) -> Boolean
```

ジェネリックな署名

#### パラメーター

| 名前       | 型       | 説明                                       |
|----------|---------|--|
| boolVal1 | Boolean | 評価するブール値を指定します。次のアイテムはコマにより区切られる必要があります。 |
| boolVal2 | Boolean | 上記と同様です。                                 |
| boolValN | Boolean | 上記と同様です。                                 |

### 11.7.2 any

全てのブール値がtrueの場合true を返します。最初のtrue 値の後に評価を停止します。すべての値がfalse の場合、falseを返します。

#### 署名

```
any (boolVal1: Boolean, boolVal2: Boolean, boolValN: Boolean) -> Boolean
```

ジェネリックな署名

#### パラメーター

| 名前       | 型       | 説明                                       |
|----------|---------|--|
| boolVal1 | Boolean | 評価するブール値を指定します。次のアイテムはコマにより区切られる必要があります。 |
| boolVal2 | Boolean | 上記と同様です。                                 |
| boolValN | Boolean | 上記と同様です。                                 |

### 11.7.3 false

ブール値 **false** を返します。

#### 署名

```
false() -> Boolean
```

ジェネリックな署名

### 11.7.4 if

ブール値が **true** の場合、**valueTrue** を返し、**false** の場合、**valueFalse** を返します。選択されたサブ式のみが評価されます。両方のサブ式は戻り値でもある同じ型である必要があります。

#### 署名

```
if(condition:Boolean, valueTrue:any type, valueFalse:any type) -> any type
```

ジェネリックな署名

#### パラメーター

| 名前                | 型                     | 説明                                      |
|-------------------|-----------------------|---|
| <b>condition</b>  | <code>Boolean</code>  | 評価する条件を指定します。                           |
| <b>valueTrue</b>  | <code>any type</code> | 条件が <b>true</b> を評価した場合、返されるサブ式を指定します。  |
| <b>valueFalse</b> | <code>any type</code> | 条件が <b>false</b> を評価した場合、返されるサブ式を指定します。 |

#### サンプル

次の式はXMLスキーマ準拠した値としてブール値を返します:

```
if(b, "true", "false")
```

これを行う代替法は以下の通りです:

```
if(b, "1", "0")
```

### 11.7.5 not

引数として与えられたブール値の否定型を返します

#### 署名

```
not (value: Boolean) -> Boolean
```

ジェネリックな署名

#### パラメーター

| 名前    | 型       | 説明              |
|-------|---------|-----------------|
| value | Boolean | 否定するブール値を指定します。 |

### 11.7.6 true

ブール値 **true** を返します。

#### 署名

```
true () -> Boolean
```

ジェネリックな署名

## 11.8 ランタイム情報関数

ランタイム情報関数は、現在実行されているジョブの詳細を処理するために使用されます。

### 11.8.1 instance-id

各ジョブの実行のために一意の文字列を返します。これは、文字列がデレクトリ名を定義するために使用される、各ジョブの実行のために一意のデレクトリを作成するためにも使用することができます。

#### 署名

```
instance-id() -> string
```

ジェネリックな署名

### 11.8.2 log

引数として受信された式を変換し、システムログに書き込みます。この関数はステップにより生成される式を明示的にログする場合などのシチュエーションで役に立ちます。この方法での値をログすると、ログされた値が長すぎる場合、システムログ内で切り捨てが発生しません。[ログの設定](#)も参照してください。

#### 署名

```
log(expression:T0) -> string
```

ジェネリックな署名

#### パラメーター

| 名前         | 型  | 説明                              |
|------------|----|---------------------------------|
| expression | T0 | FlowForce 式はログされます、型 T0 (全ての型)。 |

#### サンプル

以下のようなパスからファイルのリストを取得するジョブが作成されると仮定します。

### Execution Steps

+

- Execute function `/system/compute`
- Parameters: Expression: `list-files('C:\FlowForceExamples\LogFunction\source')`
- = Assign this step's result to  as T0

ソースディレクトリ内のファイル数がリストのためのFlowForce デフォルトのログリミットを超えた場合、ジョブログ内のエントリは省略されます。下記の通り、このサンプルでは、最初の10 ファイル名のみが表示されます。また、\は50文字のデフォルトの制限を超えるため各ファイルパスの最後の文字は切り詰められます。

| Date                | Message  |
|---------------------|--|
| 2020-09-21 15:20:17 | Starting instance 67.  |
| 2020-09-21 15:20:17 | Starting job execution: job <code>/public/Examples/Functions/log-function</code> in queue <code>/public/Examples/Functions/log-function</code>   |
| 2020-09-21 15:20:17 | Running instance 67 locally.   |
| 2020-09-21 15:20:17 | <span style="color: blue;">▶</span> Job <code>/public/Examples/Functions/log-function</code>   |
| 2020-09-21 15:20:17 | <span style="color: blue;">▶</span> System function <code>/system/compute</code><br>Computed list("C:\FlowForceExamples\LogFunction\source\File 01.tx[...]", "C:\FlowForceExamples\LogFunction\source\File 02.tx[...]", "C:\FlowForceExamples\LogFunction\source\File 03.tx[...]", "C:\FlowForceExamples\LogFunction\source\File 04.tx[...]", "C:\FlowForceExamples\LogFunction\source\File 05.tx[...]", "C:\FlowForceExamples\LogFunction\source\File 06.tx[...]", "C:\FlowForceExamples\LogFunction\source\File 07.tx[...]", "C:\FlowForceExamples\LogFunction\source\File 08.tx[...]", "C:\FlowForceExamples\LogFunction\source\File 09.tx[...]", "C:\FlowForceExamples\LogFunction\source\File 10.tx[...]", ...) |
| 2020-09-21 15:20:17 | Finished job execution: job <code>/public/Examples/Functions/log-function</code> in queue <code>/public/Examples/Functions/log-function</code>   |

省略を回避するためにlog 関数内の式を囲み、ジョブの構成を保存します。

### Execution Steps

+

- Execute function `/system/compute`
- Parameters: Expression: `log(list-files('C:\FlowForceExamples\LogFunction\source'))`
- = Assign this step's result to  as T0

システムによりログ済みのエントリに加え、新規の構成と共にジョブを実行すると、ログはログ済みの式のための新規のエントリが含まれます。切り捨てはもう発生しません。

| Date                | Message   |
|---------------------|---|
| 2020-09-21 15:21:41 | Starting instance 68.   |
| 2020-09-21 15:21:41 | Starting job execution: job <code>/public/Examples/Functions/log-function</code> in queue <code>/public/Examples/Functions/log-function</code>  |
| 2020-09-21 15:21:41 | Running instance 68 locally.  |
| 2020-09-21 15:21:41 | <ul style="list-style-type: none"> <li>Job <code>/public/Examples/Functions/log-function</code> <ul style="list-style-type: none"> <li>System function <code>/system/compute</code> <pre>step 1 parameter 'Expression': C:\FlowForceExamples\LogFunction\source\File 01.txt C:\FlowForceExamples\LogFunction\source\File 02.txt C:\FlowForceExamples\LogFunction \source\File 03.txt C:\FlowForceExamples\LogFunction\source\File 04.txt C:\FlowForceExamples \LogFunction\source\File 05.txt C:\FlowForceExamples\LogFunction\source\File 06.txt C:\FlowForceExamples\LogFunction\source\File 07.txt C:\FlowForceExamples\LogFunction \source\File 08.txt C:\FlowForceExamples\LogFunction\source\File 09.txt C:\FlowForceExamples \LogFunction\source\File 10.txt C:\FlowForceExamples\LogFunction\source\File 11.txt C:\FlowForceExamples\LogFunction\source\File 12.txt C:\FlowForceExamples\LogFunction \source\File 13.txt C:\FlowForceExamples\LogFunction\source\File 14.txt C:\FlowForceExamples \LogFunction\source\File 15.txt C:\FlowForceExamples\LogFunction\source\File 16.txt C:\FlowForceExamples\LogFunction\source\File 17.txt C:\FlowForceExamples\LogFunction \source\File 18.txt C:\FlowForceExamples\LogFunction\source\File 19.txt C:\FlowForceExamples \LogFunction\source\File 20.txt C:\FlowForceExamples\LogFunction\source\File 21.txt</pre> </li> </ul> </li> </ul> |

このサンプルでは、`log` 式関数の呼び出しの代替として、ログするステップ/パラメーターの横の“ログの有効化”  ボタンをクリックすることができます。これを行うことは `log` 関数の使用と同様で、次回ジョブ構成ページを開くと、FlowForce は `log` 関数を隠します。  ボタンと `log` 関数の違いは、前者はテキストボックス内の式全体をログし、`log` 関数は小さいサブ式などのために厳選して使用することができます。例:

### 実行ステップ

- 
  - 
 関数の実行 `/system/compute`
  - パラメーター: | 条件式: `contains(log("haystack"), "needle")`
  - = ステップの結果を割り当て  以下として T0

### 11.8.3 slot-number

現在ジョブを実行しているキューの実行スロット番号を返します。ファイル名として数値を使用してはなりません。平行ジョブを実行するためにサーバーにアクセスするために番号は使用されます(単純にコードをランスするため)。

スロット実行が開始されたキューによりスロット番号は異なります。現在のジョブ型のジョブにより呼び出される場合、呼び出しジョブのスロット番号が継承されます。

## 署名

```
slot-number() -> number
```

ジェネリックな署名

## 11.9 MIME 式関数

MIME 式関数はMIME (Multipurpose Internet Mail Extensions) エンティティのヘッダーを操作する必要なジョブに適用することができます。

### 11.9.1 add-mime-header

追加されたヘッダー `key: value` を持つ、ストリームを返します。この関数はキーを持つ既存のヘッダーを削除しません。

#### 署名

```
add-mime-header (s:stream, key:string, value:string) -> stream
```

ジェネリックな署名

#### パラメーター

| 名前                 | 型                   | 説明              |
|--------------------|---------------------|-----------------|
| <code>s</code>     | <code>stream</code> | ヘッダーが追加されるストリーム |
| <code>key</code>   | <code>string</code> | キー値ペアからの キー。    |
| <code>value</code> | <code>string</code> | キー値ペアからの 値。     |

### 11.9.2 add-mime-headers

追加されたヘッダーからの全てのストリームを返します。

#### 署名

```
add-mime-headers (s:stream, headers:list of (string, string)) -> stream
```

ジェネリックな署名

#### パラメーター

| 名前                   | 型                                     | 説明                              |
|----------------------|---------------------------------------|---------------------------------|
| <code>s</code>       | <code>stream</code>                   | 入力ストリームを指定します。                  |
| <code>headers</code> | <code>list of (string, string)</code> | 追加されるヘッダーのリスト。関数を使用してリストを作成します。 |

## サンプル

サンプル次の式は、2つのヘッダーを含むストリームを返します: **Content-Disposition** と **Content-Transfer-Encoding**.

```
add-mime-headers(empty-stream(), list(('Content-
Disposition','attachment; name=something'), ('Content-Transfer-Encoding','7bit')))
```

## 実行ステップ

関数の実行 /system/compute

パラメーター: 条件式: `add-mime-headers(empty-stream(), list(('Content-Disposition','attachment; name=something'), ('Content-Transfer-Encoding','7bit')))`

= ステップの結果を割り当て 名前 以下としてT0

新規の実行ステップ 新規の選択ステップ 新規の For-each ステップ 新規エラー/成功処理ステップ

## 11.9.3 current-message-id

AS2 メッセージの **Message-ID** ヘッダーフィールドを返します。AS2 リクエストを受信するために構成されたジョブ内でこの関数は使用される必要があります。すなわち、チェックボックス「このジョブを HTTP を介して URL 上で利用可能にする」がジョブ構成ページ内で選択されている必要があります。これ以外の場合、この関数は、新たに生成された **Message-ID** を返します。(新規のジョブインスタンスが実行され、終了までそのジョブインスタンスのために安定している場合、新規の値が生成されます)。

## 署名

```
current-message-id() -> string
```

ジェネリックな署名

## サンプル

次の式は **Message-ID** をベースにしたファイル名を作成します。substring 関数は、山かこ(最初と最後の文字)を **Message-ID** から削除します。

```
C:\temp\{substring(current-message-id(), 1, -1)}.msg
```

次の式は、上記と同じように作動し、更に現在の **Message-ID** を分割 関数を使用して、文字 @ の箇所まで分割します。nth 関数は、ランダムな16進数の値を持つ32文字の長さの最初の部分のみを抽出し、ファイル名の一部として使用します。

```
C:\temp\{nth(split(substring(current-message-id(), 1, -1), '@'), 0)}.msg
```

### 11.9.4 get-mime-content-disposition-param

このようなヘッダーとパラメータが存在する場合、ストリームの“Content-Disposition”ヘッダーから、パラメータ *param* を返します。これ以外の場合、*default* 引数の値を返します。この関数はオプションのAS2 プロファイル **File Name preservation (FN)** に続くメッセージを受信し、MIME ヘッダーからの元のファイル名を取得するために使用されます。

#### 署名

```
get-mime-content-disposition-param(s:stream, param:string, default:string="") -> string
```

ジェネリックな署名

#### パラメーター

| 名前      | 型      | 説明  |
|---------|--------|---|
| s       | stream | 入力ストリームを指定します。  |
| param   | string | 返すパラメータ名を指定します。   |
| default | string | 指定されているパラメータとヘッダーが存在しない場合返される値を指定します。デフォルトではこれは空の文字列です。 |

#### サンプル

ストリーム *msg* がヘッダー **Content-Disposition: attachment; filename="GETMSG.edi"** を含むと想定して、次の式は **GETMSG.edi** を返します:

```
get-mime-content-disposition-param(msg, "filename")
```

### 11.9.5 get-mime-content-id

存在する場合、**Content-ID** ヘッダーの値を引数として与えられたストリームから返します。それ以外の場合、*default* の引数が返されます。

#### 署名

```
get-mime-content-id(s:stream, default:string="") -> string
```

ジェネリックな署名

### パラメーター

| 名前      | 型      | 説明  |
|---------|--------|---|
| s       | stream | 入力ストリームを指定します。                                      |
| default | string | Content-ID ヘッダーが存在しない場合返す値を指定します。デフォルトではこれは空の文字列です。 |

### サンプル

ストリームmsg が次のヘッダー **Content-ID: <root.attachment>** を持つと仮定します。式は、この場合

```
get-mime-content-id(msg, "")
```

“<root.attachment>” を返します。このようなヘッダーが存在する場合、上の式は空の文字列(2番目の引数の値)を返します。

## 11.9.6 get-mime-content-type-param

ヘッダーとパラメーターが存在する場合、ストリームの“Content-Type”ヘッダーからパラメーターparam を返します。それ以外の場合、デフォルトの引数の値を返します。任意のAS2 プロファイル **Multiple Attachments (MA)** に続くメッセージを受信するためこの関数は使用されます。具体的には、開始ドキュメント Content-ID と Content-Type を multipart/related コンテンツ型に対するパラメータ'start' と'type' 通りに抽出します。下のサンプルに示される通り、文字セットを抽出するためにも使用することができます。

### 署名

```
get-mime-content-type-param(s:stream, param:string, default:string="") -> string
```

ジェネリックな署名

### パラメーター

| 名前      | 型      | 説明  |
|---------|--------|---|
| s       | stream | 入力ストリームを指定します。                                      |
| param   | string | 返すパラメーター名を指定します。                                    |
| default | string | 指定されているパラメーターが存在しない場合返される値を指定します。デフォルトではこれは空の文字列です。 |

### サンプル

ストリームmsg はヘッダー **Content-Type: text/html; charset=utf-8** を含むことを前提として、次の式は utf-8 を返します：

```
get-mime-content-type-param(msg, "charset", "ascii")
```

## 11.9.7 get-mime-header

ヘッダーが存在する場合、特定のMIME ヘッダーを現在のストリームから取得します。これ以外の場合は、デフォルトの引数を返します。

### 署名

```
get-mime-header(s:stream, key:string, default:string="") -> string
```

ジェネリックな署名

### パラメーター

| 名前      | 型      | 説明                                 |
|---------|--------|------------------------------------|
| s       | stream | 入力ストリームを指定します。                     |
| key     | string | ヘッダーを構成するキー値ペアからのキー。               |
| default | string | 返すデフォルトの値を指定します。デフォルトではこれは空の文字列です。 |

### サンプル

ストリームmsg がヘッダー **Content-Disposition: attachment; filename="GETMSG.edi"** を含むこと前提して、次の式は `attachment; filename="GETMSG.edi"` を返します:

```
get-mime-header(msg, "Content-Disposition", "")
```

このサンプルでは、ストリームがContent-Disposition ヘッダーを持たない場合、上の式は (3番目の引数の値である) 空の文字列を返します。

## 11.9.8 get-mime-headers

ストリームからすべてのMIME ヘッダーを取得し、タプル(キー、値)のリストを返します。返されるリストはadd-mime-headers 式関数の headers パラメーターとして与えることができます。

### 署名

```
get-mime-headers(s:stream) -> list of (string, string)
```

ジェネリックな署名

### パラメーター

| 名前 | 型      | 説明             |
|----|--------|----------------|
| s  | stream | 入力ストリームを指定します。 |

## 11.9.9 get-stream-filename

引数と共に提供されたストリームがファイルから作成される場合、拡張子と共にストリームのファイル名を返します。それ以外の場合 デフォルトの引数の値を返します。

### 署名

```
get-stream-filename(ストリーム:stream, default:string="") -> string
```

ジェネリックな署名

### パラメーター

| 名前      | 型      | 説明                                 |
|---------|--------|------------------------------------|
| ストリーム   | stream | 入力ストリームを指定します。                     |
| default | string | 返すデフォルトの値を指定します。デフォルトではこれは空の文字列です。 |

## 11.9.10 is-file

関数 `as-file` が既存のファイル名を返す場合、**true**を返し、`as-file`が一時的ファイルを作成すると**false**を返します。

例えば、関数、または、`stream-open` 関数、または、マッピングから返された関数を使用してストリームを作成すると**true** が返されます。ファイルからストリームがサブされない場合、または、一時的なファイルの場合、この関数は**false**を返します。

### 署名

```
is-file(s:stream) -> Boolean
```

ジェネリックな署名

### パラメーター

| 名前 | 型      | 説明             |
|----|--------|----------------|
| s  | stream | 入力ストリームを指定します。 |

## 11.9.11 is-mime-content-type

ヘッダーストリームの“Content-Type”ヘッダーに一致し、カスタム定義済みのルールを受け入れます。“Content-Type”ヘッダーが存在し、ルールが値に一致する場合、**true**を返します。それ以外の場合は、**false**を返します。“Content-Type”ヘッダーを持たないストリームは、“application/octet-stream”として扱われます。

受け入れルールは次の書式を取ります: 拡張されたBackus-Naur form (EBNF) 反復:

```
Match ::= Single ("," Single)*
Single ::= Spaces? Type-Match ( Spaces? ";" Spaces? Parameter )* Spaces?
Type-Match ::=
  "*"/*" |
  Type "/"*" |
  Type "/*+" Suffix |
  Type "/" Subtype

Parameter ::= Name "=" Value
```

## 署名

```
is-mime-content-type(s:stream, accept:string) -> Boolean
```

ジェネリックな署名

## パラメーター

| 名前            | 型                   | 説明                       |
|---------------|---------------------|--------------------------|
| <b>s</b>      | <code>stream</code> | 入力ストリームを指定します。           |
| <b>accept</b> | <code>string</code> | カスタムで定義された受け入れルールを指定します。 |

## サンプル

ストリーム *msg* がヘッダー **Content-Type: text/html; charset=utf-8** または **Content-Type: text/plain; charset=utf-8** を含む場合、次の式は `true` を返します。

```
is-mime-content-type(msg, "text/*; charset=utf-8")
```

ストリーム *msg* がヘッダー **Content-Type: application/rss+xml** または **Content-Type: application/svg+xml** を含む場合、次の式は `true` を返します。

```
is-mime-content-type(msg, "application/*+xml")
```

コマンドで区切り複数のルールを一致させることができます。ストリーム *msg* がヘッダー **Content-Type: text/xml** または **Content-Type: application/xml** を含む場合、次の式は `true` を返します。

```
is-mime-content-type(msg, "text/xml, application/xml")
```

### 11.9.12 mime-content-encode

**Content-Transfer-Encoding** としてのエンコードをストリーム *s* に適用します。

サポートされるエンコードは

空の文字列: "binary"に等しい。

- "base64": Base64 エンコード
- "quoted-printable": 引用された印刷可能なエンコード

他の文字列: エンコード無し

現在の**Content-Transfer-Encoding** を使用してストリームのエンコードし、指定されたコードを使用して再コードします。新規の**Content-Transfer-Encoding** は結果ストリームのヘッダー内に保管されます。

関数はソースエンコード内のエラーが報告されることを保証しません。

## 署名

```
mime-content-encode (s:stream, encoding:string="") -> stream
```

ジェネリックな署名

## パラメーター

| 名前       | 型      | 説明                                 |
|----------|--------|------------------------------------|
| s        | stream | 入力ストリームを指定します。                     |
| encoding | string | 適用するエンコードを指定します。デフォルトではこれは空の文字列です。 |

### 11.9.13 mime-flatten

MIME ヘッダーを持つストリームを取り、コンテンツ内の元のヘッダーを含むストリームに変換します。結果するストリームは message/rfc822 のコンテンツ型を持ちます。

## 署名

```
mime-flatten (s:stream) -> stream
```

ジェネリックな署名

## パラメーター

| 名前 | 型      | 説明             |
|----|--------|----------------|
| s  | stream | 入力ストリームを指定します。 |

### 11.9.14 mime-multipart

無限の数量のストリームを取り、単一の multipart/subtype に結合します。

境界は自動的に作成されます。ストリームはアセンブル前にフラット化されます。追加のパラメータを持つマルチパートはサポートされません。

FlowForce Server Advanced Edition ユーザーのためのメモ AS2 型のマルチパートメッセージを定義しないため、AS2 のためにサブ型は常に `mime-multipart-related` 関数を確認してください。

## 署名

```
mime-multipart(subtype:string, s:stream) -> stream
```

ジェネリックな署名

## パラメーター

| 名前             | 型                   | 説明  |
|----------------|---------------------|---|
| <b>subtype</b> | <code>string</code> | 使用する <code>multipart/subtype</code> を指定します。 |
| <b>s</b>       | <code>stream</code> | 入力ストリームを指定します。                              |

## サンプル

次の式は EDI ファイルと PDF の 2 つのファイルを含むストリームを返します。

```
mime-multipart("related", stream-open("c:\example\order.edi", "application/EDIFACT"), stream-open("c:\example\measuredetails.pdf", "application/pdf"))
```

### Execution Steps

The screenshot shows a configuration window for an execution step. The step is named "Execute function" and is located in the "/system/compute" environment. The parameters are set to an expression: `mime-multipart("related", stream-open("c:\example\order.edi", "application/EDIFACT"), stream-open("c:\example\measuredetails.pdf", "application/pdf"))`. The result of this step is assigned to a variable named "name". Below the configuration, there are buttons for "new Execution step", "new Choose step", "new For-each step", and "new error/success handling step".

## 11.9.15 mime-multipart-related

無制限の数のストリームを取り、`multipart/related` に結合します。境界は自動的に作成されます。ストリームはアセンブル前にフラット化されます。

FlowForce Server Advanced Edition ユーザーのためのメモ 任意の AS2 プロファイル **Multiple Attachments (MA)** の後に来るメッセージをアセンブルするためこの関数を使用することができます。最初のストリームが注目のパートになります。存在しない場合、マルチパートのアセンブルの前に全てのパートは作成され一意の値を持つ Content-ID ヘッダーを取得します。`new-message-id` 関数に返された通り作成された値は新規の **Message-ID** です。ソースストリームは影響を受けていません。

## 署名

```
mime-multipart-related(s:list of stream) -> stream
```

ジェネリックな署名

## パラメーター

| 名前 | 型              | 説明                 |
|----|----------------|--------------------|
| s  | list of stream | ストリームの入力リストを指定します。 |

## サンプル

次の式は、2つのストリームを含むストリームを返します。

```
mime-multipart-related(list(part1, part2))
```

## ジョブ入力パラメーター



名前:  型:  ▼ 説明: (



名前:  型:  ▼ 説明: (



## 実行ステップ



▲ 関数の実行

パラメーター: | 条件式: 

= ステップの結果を割り当て  以下としてT0

### 11.9.16 mime-multipart-from-list

ストリームsのリストを取り、`multipart/subtype` と組み合わせます。

#### 署名

```
mime-multipart-from-list(subtype:string, s:list of stream) -> stream
```

ジェネリックな署名

#### パラメーター

| 名前                   | 型                           | 説明  |
|----------------------|-----------------------------|---|
| <code>subtype</code> | <code>string</code>         | 使用する <code>multipart/subtype</code> を指定します。 |
| <code>s</code>       | <code>list of stream</code> | ストリームの入力リストを指定します。                          |

### 11.9.17 mime-parse

ストリームs 内に保管されている MIME メッセージを解析し、MIME ヘッダーとメッセージボディを分割します。必要な場合、“Content-Transfer-Encoding” ヘッダーに従い暗号化が解除されているメッセージボディコンテンツを持つストリームを返します。MIME ヘッダーに `get-mime-header`、`is-mime-content-type` などの式関数によりアクセスすることができます。`mime-flatten` 関数により実行された項目を元に戻します。関数はノーストリーム内のエラーが報告されることを保証しません。

#### 署名

```
mime-parse(s:stream) -> stream
```

ジェネリックな署名

#### パラメーター

| 名前             | 型                   | 説明             |
|----------------|---------------------|----------------|
| <code>s</code> | <code>stream</code> | 入力ストリームを指定します。 |

### 11.9.18 mime-split-multipart

ストリームs が MIME マルチパートのメッセージの場合、この関数は分割し、ストリームのリストを返します。ストリームs がマルチパートのメッセージでない場合、(すなわち `is-mime-content-type(s, "multipart/*")` が `false` を返す場合)、関数は1つの要素 - ストリームs (変更無し)のリストを返します。関数はノーストリーム内のエラーが報告されることを保証しません。

## 署名

```
mime-split-multipart(s:stream) -> list of stream
```

ジェネリックな署名

## パラメーター

| 名前 | 型      | 説明             |
|----|--------|----------------|
| s  | stream | 入力ストリームを指定します。 |

## 11.9.19 new-message-id

**Message-ID** ヘッダーフィールドのための新規の値を生成し返します。MIME メッセージのヘッダーを作成するためこの値を使用することができます。current-message-id とは異なり、この関数は、常に新規の **Message-ID** を返します。**Message-ID** は次の書式を有します:

```
'<' UTC timestamp '-' random hex value 32 characters long '@' host name related text '>'
```

例: <20180306154822808383-5933b654b26c4495bb0b619ab72b3bc6@myservername>.

## 署名

```
new-message-id() -> string
```

ジェネリックな署名

## 11.9.20 reset-mime-headers

新しいヘッダーを持つストリームを返します。ヘッダーリスト無しに、全てのヘッダーをクリアします。

## 署名

```
reset-mime-headers(s:stream, headers:list of (string, string)=empty) -> stream
```

ジェネリックな署名

## パラメーター

| 名前 | 型      | 説明             |
|----|--------|----------------|
| s  | stream | 入力ストリームを指定します。 |

| 名前      | 型                                     | 説明                               |
|---------|---------------------------------------|----------------------------------|
| headers | <code>list of (string, string)</code> | 作成されるヘッダーのリストを指定します。デフォルトの値は空です。 |

### 11.9.21 set-mime-content-disposition

s 内で検索された MIME 「Content-Disposition」ヘッダーのパラメータを設定します。

FlowForce Server Advanced Edition ユーザーのためのメモ この関数は任意の AS2 プロファイル **FileName preservation (FN)** を使用して、AS2 メッセージの送信時に役立ちます。ファイル名の読み取りのために、関数 `get-mime-content-disposition-param` も参照してください。

#### 署名

```
set-mime-content-disposition(s:stream, disposition:string, filename:string="") -> string
```

ジェネリックな署名

#### パラメーター

| 名前          | 型                   | 説明   |
|-------------|---------------------|--|
| s           | <code>stream</code> | 入力ストリームを指定します。   |
| disposition | <code>string</code> | 設定する「Content-Disposition」ヘッダーの <i>disposition</i> 値                |
| filename    | <code>string</code> | 設定する「Content-Disposition」ヘッダーの <i>filename</i> 値デフォルトではこれは空の文字列です。 |

#### サンプル

次の式は、Content-Disposition ヘッダーは、以下のように設定します: **Content-Disposition: attachment; filename="GETMSG.edi"**:

```
set-mime-content-disposition(msg, "attachment", "GETMSG.edi")
```

### 11.9.22 set-mime-content-id

value に設定されている「Content-ID」ヘッダーを持つストリームを返します。set-mime-header 関数を使用して同じ結果を達成する場合、この関数はより直接的なアプローチを表します。

## 署名

```
set-mime-content-id(s:stream, value:string="") -> stream
```

ジェネリックな署名

## パラメーター

| 名前    | 型      | 説明                              |
|-------|--------|---------------------------------|
| s     | stream | 入力ストリームを指定します。                  |
| value | string | 「Content-Disposition」内の値を指定します。 |

## サンプル

ストリームmsg内のContentID ヘッダーの値を<root.attachment>に設定することを前提とします。これを行うには、次の式を使用します:

```
set-mime-content-id(msg, "<root.attachment>")
```

## 11.9.23 set-mime-header

ヘッダー キーが値に設定されているストリームを返し、その他のヘッダーとコンテンツには影響はありません。一度に複数のヘッダーを変更する場合 set-mime-headers 関数を使用する必要があります。

## 署名

```
set-mime-header(s:stream, key:string, value:string) -> stream
```

ジェネリックな署名

## パラメーター

| 名前    | 型      | 説明                 |
|-------|--------|--------------------|
| s     | stream | 入力ストリームを指定します。     |
| key   | string | 設定するヘッダーのキーを指定します。 |
| value | string | 設定するヘッダー値を指定します。   |

## サンプル

「Content-Type」ヘッダーをオーバーライドするには、次を使用します:

```
set-mime-header(s, "Content-Type", "text/plain; charset=iso-8859-1")
```

## 11.9.24 set-mime-headers

ヘッダーからのキー値ペアによるヘッダー引数を持つストリームを返します。新しいヘッダーは同じ名前を持つ既存のヘッダーをシャドウします。

### 署名

```
set-mime-headers(s:stream, headers:list of (string, string)) -> stream
```

ジェネリックな署名

### パラメーター

| 名前      | 型                        | 説明             |
|---------|--------------------------|----------------|
| s       | stream                   | 入力ストリームを指定します。 |
| headers | list of (string, string) | 設定されるヘッダーのリスト  |

### サンプル

“Content-Type” ヘッダーをオーバーライドするには、次を使用します:

```
set-mime-headers(s, list(("Content-Type", "text/plain; charset=iso-8859-1")))
```

## 12 ジョブの構成の例

このチャプターにはステップバイステップのFlowForce ジョブ構成サンプルが含まれています。下のテーブルは特定の関数の種類と各サンプルで説明されているトリガーと共にすべてのサンプルをリストします。

| 例  | 説明される概念   |  |          |
|--|---|--|----------|
|  | 内蔵された関数   | 式関数  | トリガー     |
| <a href="#">「Hello, World!」ジョブの作成ジョブ</a>         | <ul style="list-style-type: none"> <li>/system/compute</li> </ul>   |  | Web サービス |
| <a href="#">パスが存在するか確認する</a>                     | <ul style="list-style-type: none"> <li>/system/shell/commandline</li> <li>/system/compute-string</li> </ul>                           | content()<br>stdout()<br>trim()  | Web サービス |
| <a href="#">ファイルをコピーする</a>                       | <ul style="list-style-type: none"> <li>/system/filesystem/copy</li> </ul>   | list-files()   | Web サービス |
| <a href="#">MapForce マッピングからジョブを作成する</a>         | <ul style="list-style-type: none"> <li>MapForce マッピング</li> </ul>  |  | タイマー     |
| <a href="#">ジョブを他のジョブのステップとして使用する</a>            | <ul style="list-style-type: none"> <li>/system/filesystem/copy</li> </ul>   |  |          |
| <a href="#">ディレクトリポッピングジョブの作成</a>                | <ul style="list-style-type: none"> <li>MapForce マッピング</li> <li>/system/filesystem/move</li> </ul>                                     |  | ファイルシステム |
| <a href="#">エラー処理をジョブに追加する</a>                   | <ul style="list-style-type: none"> <li>/system/shell/commandline</li> <li>/system/mail/send</li> </ul>                                | failed-step()<br>error-message()<br>exitcode()<br>stdout()<br>stderr()<br>content()<br>instance-id() | Web サービス |
| <a href="#">Web サービスとしてジョブを公開する</a>              | <ul style="list-style-type: none"> <li>MapForce マッピング</li> </ul>  |  | Web サービス |
| <a href="#">JSON をFlowForce Web サービスへポストする方法</a> | <ul style="list-style-type: none"> <li>/system/filesystem/copy</li> </ul>   | as-file()<br>instance-id()   | Web サービス |
| <a href="#">ジョブの結果のキャッシュ</a>                     | <ul style="list-style-type: none"> <li>/system/shell/commandline</li> <li>/system/compute</li> </ul>                                  | stdout()   | Web サービス |
| <a href="#">StyleVision 変換からジョブを作成する</a>         | <ul style="list-style-type: none"> <li>StyleVision 変換</li> <li>/system/compute</li> <li>/system/filesystem/copy</li> </ul>            | results()<br>nth()<br>as-file()  | タイマー     |
| <a href="#">RaptorXML を使用してドキュメントを検証する</a>       | <ul style="list-style-type: none"> <li>/RaptorXML/valany</li> </ul>   |  | タイマー     |
| <a href="#">エラーログのあるXMLを検証する</a>                 | <ul style="list-style-type: none"> <li>/RaptorXML/valxml-withxsd</li> <li>/system/compute</li> <li>/system/filesystem/copy</li> </ul> | failed-step()<br>stdout()<br>as-file()   | Web サービス |
| <a href="#">RaptorXML を使用してXSLTを実行する</a>         | <ul style="list-style-type: none"> <li>/RaptorXML/xslt</li> </ul>   | list()   |          |
| <a href="#">XML ファイルからPDFを生成</a>                 | <ul style="list-style-type: none"> <li>MapForce マッピング</li> <li>StyleVision 変換</li> <li>/system/compute</li> </ul>                     | as-file()<br>results()<br>filename()   | Web サービス |

## 12.1 “Hello, World!” ジョブの作成

この例では テキスト “Hello, World!” をブラウザ内に出力する簡単なジョブの作成方法が説明されています。このテキストは FlowForce 式を使用して作成されます。ブラウザ内のリンクをクリックすることにより、必要に応じてジョブをトリガーすることができます（つまり、ジョブは Web サービスとして公開されます）。

### 必要条件

- 必要とされるライセンス FlowForce Server
- FlowForce Web Server と FlowForce Server サービスが構成された [ネットワークアドレスとポート](#) でホストされていること。
- [コンテナ](#) の一つで [パーミッション](#) を有する FlowForce Server ユーザーアカウントを有していること（デフォルトでは、この例で使用されている `/public` コンテナには、認証されたユーザーがアクセスすることができます）。

### ジョブの作成方法

1. FlowForce Server に [ログイン](#) /`public` コンテナにナビゲートします。
2. [作成 | コンテナの作成](#) をクリックして [Examples] という名前の新規のコンテナを作成します。

`/public/Examples` コンテナがこのドキュメントで説明されるジョブの多くにより使用されています。他のコンテナでジョブを作成することも可能ですが、このドキュメント内の次に続くチュートリアルもフォローする必要があり `/public/Examples` コンテナを作成することが奨励されます。

3. `/public/Examples` コンテナで [作成 | ジョブの作成](#) をクリックして、ジョブのタイトルと説明を入力します。

### 以下にジョブを作成 /public

ジョブ名:

ジョブの説明:

4. [ビルトイン関数](#) `/system/compute` を呼び出す新規の実行ステップを追加します。

### 実行ステップ

+

関数の実行

パラメーター: 条件式:

= ステップの結果を割り当て  として TO

- 「式」フィールドに、一重引用符で囲まれた、テキスト 'Hello, W orld' を入力します。このフィールドのコンテンツは FlowForce Server 式を示します。

- 実行の結果を「文字列」として宣言します。

- HTTP を介してこのジョブを公開する... チェックボックスを選択し "HelloW orldService" をサービス名として入力します。詳細に関しては [Web サービスをジョブとして公開](#) を参照してください。

- 「資格情報」から、既存の資格情報の記録を選択、または、ローカルの資格情報を指定してください。詳細に関しては [資格情報](#) を参照してください。
- [保存]をクリックします。

## ジョブを実行する方法

文字列の値 "Hello, W orld!" を計算し、ジョブの結果として返すジョブの作成が完了しました。ジョブを実行するには、以下の内の一つを行ってください。

- ホームに移動し、全てのアクティブなトリガーとサービスを表示するをクリックします。次に情報列内に表示されているジョブの URL をクリックします。
- <http://127.0.0.1:4646/service/HelloWorldService> をブラウザのアドレスバーに入力します。この URL は FlowForce Server サービスがデフォルトのホスト アドレスとポート名でリスニングする場合のみ機能します。[構成ページ](#)内で他のホストとポート設定を定義した場合、アドレスを必要に応じて変更してください。
- FlowForce Server の任意のホスト名 フィールドを [セットアップページ](#)から設定すると HTTP を介してこのジョブを使用可能にする。チェックボックスの横の  ボタンをクリックして web サービスの呼び出しを直接ジョブ構成ページから実行することができます。それ以外の場合、このボタンは表示されません。

Web サービスにアクセスする際に資格情報をプロンプトされた場合、FlowForce Server にログインするために使用する同じ資格情報を

提供してください。

FlowForce Server ユーザーにHTTP 認証のための資格情報を提供することはテスト目的のためです。生産のために新規の FlowForce ユーザーを作成することが奨励されており、ジョブが存在するエンター上でこのユーザーに **Service - Use** パーミッションを与え、対応するユーザーアカウントを使用して Web サービスにアクセスします。HTTP 認証を無効化するには、Web サービスを公開し、**Service - Use** パーミッションをユーザー **Anonymous** に与えます。[パーミッションのしくみ](#)を参照してください。

ジョブの実行に成功すると、ブラウザは、ジョブの出力を表示します:

```
Hello, World!
```

ジョブの実行に失敗するとブラウザは、「サーバー実行の失敗」メッセージを表示します。この場合、FlowForce Server 内の[ジョブのログ](#)をチェックして、エラーを確認してください。ジョブのログの確認を参照してください。

## 12.2 パスが存在するか確認する

この例では（ファイルまたはディレクトリへの）パスがオペレーティングシステム上に存在するか通知するジョブを作成する方法が説明されています。この目的を達成するために、ビルトイン関数と式関数の組み合わせを使用する必要があります。ジョブをブラウザからURLにアクセスすることにより、需要に応じてトリガーするために、Web サービスとして定義されます。ジョブは、パスを引数として取り、FlowForce Server が作動するオペレーティングシステム上に引数として与えられたパスが存在するか通知する文字列を返します。

### 必要条件

- 必要とされるライセンス FlowForce Server
- FlowForce Web Server と FlowForce Server サービスが構成されたネットワークアドレスとポート でインストールされていること。
- [コンテナ](#) の一つでパーミッションを有する FlowForce Server ユーザーアカウントを有していること（デフォルトでは、この例で使用されている `/public` コンテナには、認証されたユーザーがアクセスすることができます）。

### ジョブの作成方法

1. FlowForce Server にログインし、新しいジョブの作成のパーミッションが与えられているコンテナに移動します。他のサンプルの一貫性を保つためこのチュートリアルでは `/public/Examples` コンテナを使用します。このコンテナが存在しない場合、作成 | コンテナの作成 コマンドを使用して作成してください。
2. `/public/Examples` コンテナ内で 作成 をクリックしてジョブの作成を選択します。
3. ジョブの名前を追加し、（この例では `CheckPath`）任意でジョブの説明を追加します。

### 以下にジョブを作成 /public

|         |   |
|---------|---|
| ジョブ名:   | <input type="text" value="CheckPath"/>                |
| ジョブの説明: | <input type="text" value="Checks if a path exists."/> |

4. ジョブ入力パラメーターから、 をクリックし、「パス」パラメーターを以下のように追加します。

#### ジョブ入力パラメーター

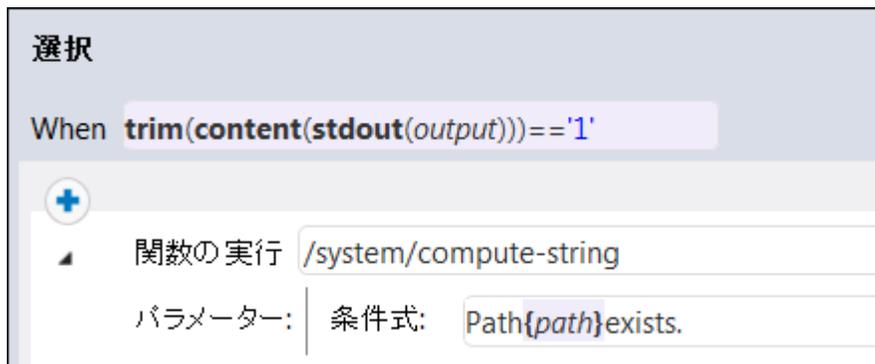
|   |                                   |                                      |  |  |
|---|-----------------------------------|--------------------------------------|--|--|
|  | <input type="text" value="path"/> | 型: <span>ファイルまたはディレクトリとしての文字列</span> | デフォルト:  | 説明: <input type="text" value="Specifies the path to be checked."/> |
|  |                                   |                                      |  |  |

5.  `/system/shell/commandline` 関数を呼び出す新規の実行ステップを追加し、ファイルの存在を確認するシェルコマンドを入力します。このステップの結果を、下に示されるように宣言してください（この例では `output` と呼びます）。



Windows では、パスが存在する場合、シェルコマンドは、「1」を出力し、パスが存在しない場合は、「0」を出力します。FlowForce Server は、Unix システム上で作動しますが、コマンドを必要に応じて調整してください。コマンドは FlowForce 式 `{path}` を埋め込んでいることにご注意ください。この式は、前のステップで定義されている入力パラメータを参照しています。

- 「実行ステップ」の下の **+** ボタンをクリック、「新規選択ステップ」を選択します。条件式として、`trim(content(stdout(output))) == '1'` を入力します。この式は、次のネストされた3つの関数から構成されます：`stdout`、`content`、および `trim`。最初に `stdout` 関数は、前のステップより返された結果の標準出力を取得します。そして、`content` 関数は、標準出力を文字列に変換します。最後に、`trim` 関数は、先頭と末尾のスペース、改行、改行文字を標準出力から削除します。結果は、等値演算子を使用して、「1」と比較されます。2つの値が等しいと、パスは存在します。それ以外の場合、パスは存在しません。
- When** 句の下に、以下に示されるように実行ステップを追加します。この実行ステップは、パスが存在する場合返される文字列の値を構築するために `/system/compute-string` 関数を呼び出します。値は FlowForce 式 `{path}` を埋め込んでいることにご注意ください。この式は前のステップで定義された入力パラメータを参照しています。



- Otherwise** 句の下に、以下に示されるように実行ステップを追加します。この実行ステップは、パスが存在しない場合返される文字列の値を構築するために `/system/compute-string` 関数を呼び出します。値は FlowForce 式 `{path}` を埋め込んでいることにご注意ください。この式は前のステップで定義された入力パラメータを参照しています。

それ以外の場合

関数の実行 `/system/compute-string`

パラメーター: 条件式: `Path{path}does not exist.`

9. 実行の結果で、「戻り値の値を文字列」として宣言します。

実行の結果

戻り値の型を以下と宣言する:

10. サービスから「このジョブを HTTP を介して利用可能にする」チェックボックスをクリックして選択し、**CheckPathService** をサービスの名前として入力します。詳細に関しては [Web サービスの公開](#) を参照してください。

Service

Make this job available via HTTP at URL `http://<FlowForce server>/service/`

11. 「資格情報」から、既存の資格情報の記録を選択、または、ローカルの資格情報を指定してください。詳細に関しては [資格情報](#) を参照してください。
12. [保存]をクリックします。

## ジョブを実行する方法

ジョブを実行するには、以下の内の1つを行ってください。

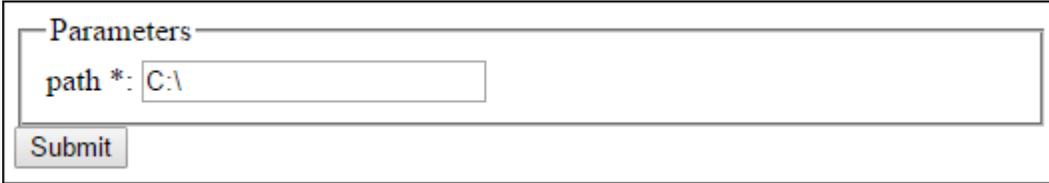
- ホームに移動し、全てのアクティブなトリガーとサービスを表示するをクリックします。次の情報列内に表示されているジョブの URL をクリックします。
- <http://127.0.0.1:4646/service/CheckPathService> をブラウザのアドレスバーに入力します。この URL は FlowForce Server サービスがデフォルトのホスト アドレスとポート名でリスニングする場合のみ機能します。構成ページ内で他のホストとポート設定を定義した場合、アドレスを必要に応じて変更してください。
- FlowForce Server の任意のホスト名 フィールドを [セットアップページ](#) から設定すると HTTP を介してこのジョブを使用可能にする チェックボックスの横の  ボタンをクリックして web サービスの呼び出しを直接ジョブ構成ページから実行することができます。それ以外の場合、このボタンは表示されません。

Web サービスにアクセスする際に資格情報をプロンプトされた場合、FlowForce Server にログインするために使用する同じ資格情報を提供してください。

FlowForce Server ユーザーに HTTP 認証のための資格情報を提供することはテスト 目的のためです。生産のために新規の FlowForce ユーザーを作成することが奨励されており、ジョブが存在するコンテナ上でこのユーザーに **Service - Use** パーミッション

を与え、対応するユーザーアカウントを使用して Web サービスにアクセスします。HTTP 認証を無効化するには Web サービスを公開し、**Service - Use** パーミッションをユーザー **Anonymous** に与えます。[パーミッションのしくみ](#)を参照してください。

このジョブには引数が存在し、ブラウザ内の Web サービスにアクセスすると引数を与えるようプロンプトされます。



The screenshot shows a web form with the following elements:

- A title "Parameters" above a text input field.
- The text input field contains the text "path \*: C:\".
- A "Submit" button is located below the input field.

ジョブの実行に成功すると、ブラウザはジョブの出力を表示します。例:

```
Path C:\ exists.
```

ジョブの実行に失敗するとブラウザは「サーバー実行の失敗」メッセージを表示します。この場合、FlowForce Server 内のジョブのログをチェックして、エラーを確認してください。[ジョブのログの確認](#)を確認してください。

## 12.3 ファイルをコピーする

このサンプルは、FlowForce Server ジョブの助けを得てローカルファイルシステム上の複数のファイルをコピーする方法について説明しています。

C:\FlowForceExamples\CopyFiles\Source から新規のディレクトリにC:\FlowForceExamples\CopyFiles\Target 全てのファイルをコピーすると仮定します。(UNIX システム上では、パスを適宜調整してください)。この目的を達成するには、ディレクトリ内のすべてのファイルを繰り返す「for-each」ステップを使用して、ループ内の各アイテムのために [/system/filesystem/copy](#) 関数を呼び出します。

### 必要条件

- 必要とされるライセンス FlowForce Server
- FlowForce Web Server と FlowForce Server サービスが構成された [ネットワークアドレスとポート](#) でホストされていること
- [コンテナ](#) の一つでパーミッションを有する FlowForce Server ユーザーアカウントを有していること(デフォルトでは、この例で使用されている /public コンテナには、認証されたユーザーがアクセスすることができます)。
- このジョブは C:\FlowForceExamples\CopyFiles\Source からディレクトリ C:\FlowForceExamples\CopyFiles\Target にファイルをコピーします。ジョブを作成する前にローカルファイルシステム上にこれらのディレクトリが作成されていることを確認してください。また、ジョブをテストするためにもソースディレクトリファイルが含まれていることを確認してください。

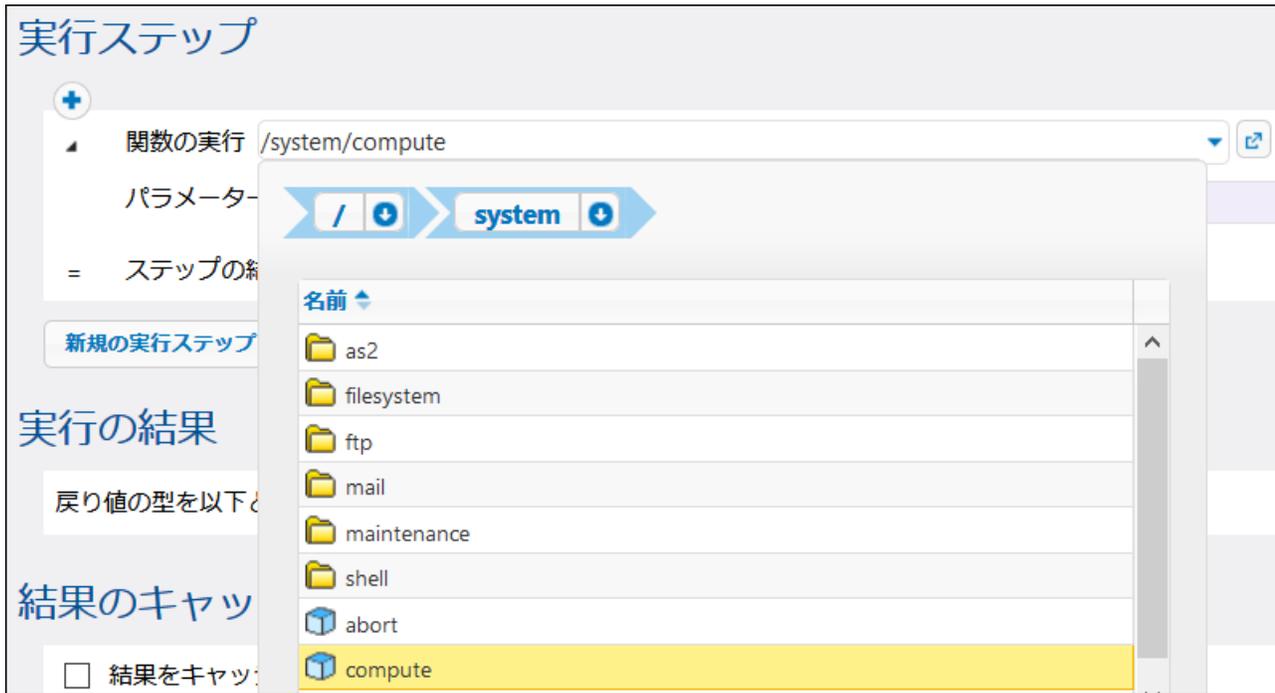
### ジョブの作成方法

FlowForce Server にログインし、新しいジョブの作成のパーミッションが与えられているコンテナに移動します。他のサンプルとの一貫性を保つためこのチュートリアルでは /public/Examples コンテナを使用します。このコンテナが存在しない場合 作成 | コンテナの作成 コマンドを使用して作成してください。

/public/Examples コンテナ内で新規のジョブを作成します。(例えば、「CopyFiles」などの) ジョブ名を入力し、オプションでジョブの詳細を入力します。

The screenshot shows a web interface for creating a job. At the top, there is a breadcrumb navigation showing a folder icon, a slash, a plus icon, and a folder icon labeled 'public'. Below this, the main heading reads '以下にジョブを作成 /public'. There are two input fields: 'ジョブ名:' (Job Name) containing 'CopyFiles' and 'ジョブの説明:' (Job Description) containing '1つのディレクトリから他のディレクトリに複数のファイルをコピーします。' (Copy multiple files from one directory to another directory).

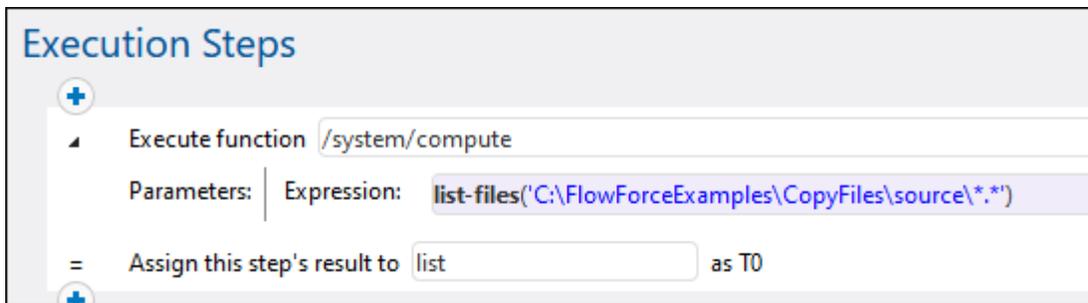
リスト内のアイテムを繰り返すために、FlowForce Server には「for-each」実行ステップが搭載されています。このようなステップは、シーケンス内の最後のアイテムを含むアイテムのシーケンス(リスト)を繰り返します。このサンプル内では、アイテムのシーケンスはソースディレクトリ内のファイルのリストになります。リクエストされたリストを作成するには、「新規の実行ステップ」をクリックして、「関数の実行」の横に「/system/compute」を入力します。下記の通り、ドロップダウンリストからこのパスを選択することもできます。



次に、式フィールド内に次の式を入力します:

```
list-files("C:\FlowForceExamples\CopyFiles\Source\*.*)"
```

次に「このステップの結果を次に割り当てる」フィールドにリストの名前を入力します(この場合、名前は `list`) です。これにより以降のステップ内の新規のファイルのリストを簡単に参照することができます。実行ステップは以下のようになります:



上記の式は `list-files` 式関数を呼び出します。関数は引数(この場合、`C:\Source\*.*)` として、パスを取り、特定のパスでファイル(または、ディレクトリ)のリストを返します。ディレクトリ内のすべてのファイルを選択するため、パスはワイルドカード `*.*` が含まれていることにご注意ください。必要な場合、ワイルドカードを調節して、特定のファイル拡張子を選択することができます。例えば、`*.txt`。FlowForce 内の式に関する詳細は、[FlowForce 式](#) を参照してください。

実際の「for-each」繰り返しステップの作成を開始することができます。「新規の For-Each ステップ」をクリックして、「シーケンス内」ボックス内に `list` を入力します。(これは前の実行ステップで作成された `list` を参照しています)。

### Execution Steps

- + Execute function `/system/compute`
  - Parameters: Expression: `list-files('C:\FlowForceExamples\CopyFiles\source\*.*)`
  - = Assign this step's result to `list` as TO
- + For each `item` in sequence `list`

ヒント: [for-each] の[in シーケンス] ボックス式をコピーし、この結果最初の実行ステップを削除することができます。

次に  ボタンをクリックし、「for-each」ステップ内に新規の実行ステップを追加します。このステップは、下記の通りループ内の各アイテムのために `/system/filesystem/copy` 関数を呼び出します。

### Execution Steps

- + Execute function `/system/compute`
  - Parameters: Expression: `list-files('C:\FlowForceExamples\CopyFiles\source\*.*)`
  - = Assign this step's result to `list` as TO
- + For each `item` in sequence `list`
  - + Execute function `/system/filesystem/copy`
    - Parameters: Source: `{item}`
    - Target: `C:\FlowForceExamples\CopyFiles\target`
    - Overwrite:
    - Abort on error: 
    - Working directory: 

上記のとおり、`copy` 関数が次の引数を使用して呼び出されます:

- ループ内のソースは現在のアイテム(ファイル)です。ソースボックス内で `{item}` を入力、または  ボタンをクリックして「アイテム」を選択します。
- ターゲットはターゲットパスです。パスはそのまま入力されます。しかしながら、ジョブに対する引数として提供することもできます。

- 「上書き」オプションが有効化されていると、すなわち、ソースディレクトリ内で同じ名前を持つファイルが既存の場合、上書きされていきます。上書きを回避するために、 ボタンをクリックしてください。

便宜上、このサンプル内では他の2つの引数は設定されていません。詳細に関しては、</system/filesystem/copy> 関数を参照してください。

作成されたジョブは、必要とされるすべての処理ステップを含んでいますが、トリガーはまだ存在しません。ジョブを定期的な時間の間隔でトリガーするには、タイマトリガーを使用することができます。[タイマトリガー](#)を参照してください。または、変更のためにソースディレクトリをモニターし、ファイルシステムトリガーの方法を用いてジョブをトリガーすることもできます。[ファイルシステムトリガー](#)を参照してください。最後に、Web サービスの呼び出しとして、ジョブをオンデマンドでトリガーすることができます。

このサンプルでは、ブラウザ内のURL をクリックして、ジョブをオンデマンドでトリガーします（実際には、ジョブがWeb サービスとして呼び出されます）。ジョブをWeb サービスにするには、「HTTP によりこのジョブを使用できるようにする」チェックボックスを選択し、Web サービスの名前を入力します。

### サービス

このジョブを HTTP を介して URL 上で利用可能にする

最後に、作動するためジョブは資格情報を必要とします。このために、「資格情報」セクションで表示されているFlowForce Server ユーザー名とパスワードではなく、オペレーティングシステムのユーザー名とパスワードを入力してください。または、[資格情報の定義](#)で説明されているとおり、ストア内の資格情報が作成されている場合、「既存の資格情報を選択」オプションを使用して選択することができます。

### 資格情報

資格情報によりジョブを実行:  既存の資格情報を選択:

ローカルの資格情報を定義: ユーザー名:

パスワード:

## ジョブを実行する方法

ジョブをテストするには、以下の内の1つを行ってください:

- ホームに移動し、全てのアクティブなトリガーとサービスを表示するをクリックします。次の情報列内に表示されているジョブのURL をクリックします。
- <http://127.0.0.1:4646/service/CopyFilesService> をブラウザのアドレスバーに入力します。このURL はFlowForce Server サービスがデフォルトのホストアドレスとポート名でリスニングする場合のみ機能します。[構成ページ](#)内で他のホストとポート設定を定義した場合、アドレスを必要に応じて変更してください。
- FlowForce Server の任意のホスト名 フィールドを [セットアップページ](#)から設定するとHTTP を介してこのジョブを使用可能にする チェックボックスの横の  ボタンをクリックしてweb サービスの呼び出しを直接ジョブ構成ページから実行することができます。それ以外の場合、このボタンは表示されません。

Web サービスにアクセスする際に資格情報をプロンプトされた場合、FlowForce Server にログインするために使用する同じ資格情報を提供してください。

FlowForce Server ユーザーに HTTP 認証のための資格情報を提供することはテスト目的のためです。生産のために新規の FlowForce ユーザーを作成することが奨励されており、ジョブが存在するコンテナ上でこのユーザーに **Service - Use** パーミッションを与え、対応するユーザーアカウントを使用して Web サービスにアクセスします。HTTP 認証を無効化するには Web サービスを公開し、**Service - Use** パーミッションをユーザー **Anonymous** に与えます。[パーミッションのしくみ](#)を参照してください。

実行に成功すると、ジョブはすべてのファイルをソースからターゲットディレクトリにコピーします。それ以外の場合、「サービスの実行に失敗しました」エラーがブラウザに表示されます。エラーが表示されると、ジョブのログをチェックしてください。詳細に関しては [ジョブログの確認](#)を参照してください。エラーの原因には、正確でない資格情報、ファイルシステム上の正確でないパーミッションなどが含まれます。例えば、下記のとおり「上書き」チェックボックスが選択されていない場合、および、ターゲットディレクトリが既に同じ名前を持つファイルを含む場合、ジョブは失敗します。

| Date                | Message   |
|---------------------|---|
| 2020-09-17 12:26:53 | Starting instance 8.  |
| 2020-09-17 12:26:53 | Starting job execution: job /public/Examples/CopyFiles in queue /public/Examples/CopyFiles              |
| 2020-09-17 12:26:53 | Running instance 8 locally.   |
| 2020-09-17 12:26:53 | Job /public/Examples/CopyFiles  |
| 2020-09-17 12:26:53 | System function /system/compute   |
| 2020-09-17 12:26:53 | For each item in list   |
| 2020-09-17 12:26:53 | Iteration 1   |
| 2020-09-17 12:26:53 | System function /system/filesystem/copy   |
| 2020-09-17 12:26:53 | Executing FlowForce.copy with parameters: Source: "C:\FlowForceExamples\CopyFiles\source\invoices.txt", |
| 2020-09-17 12:26:53 | Step FlowForce.copy failed: Failed copying the file: The file exists.                                   |
| 2020-09-17 12:26:53 | Job execution failed: job /public/Examples/CopyFiles in queue /public/Examples/CopyFiles                |

## 12.4 MapForce マッピングからジョブを作成する

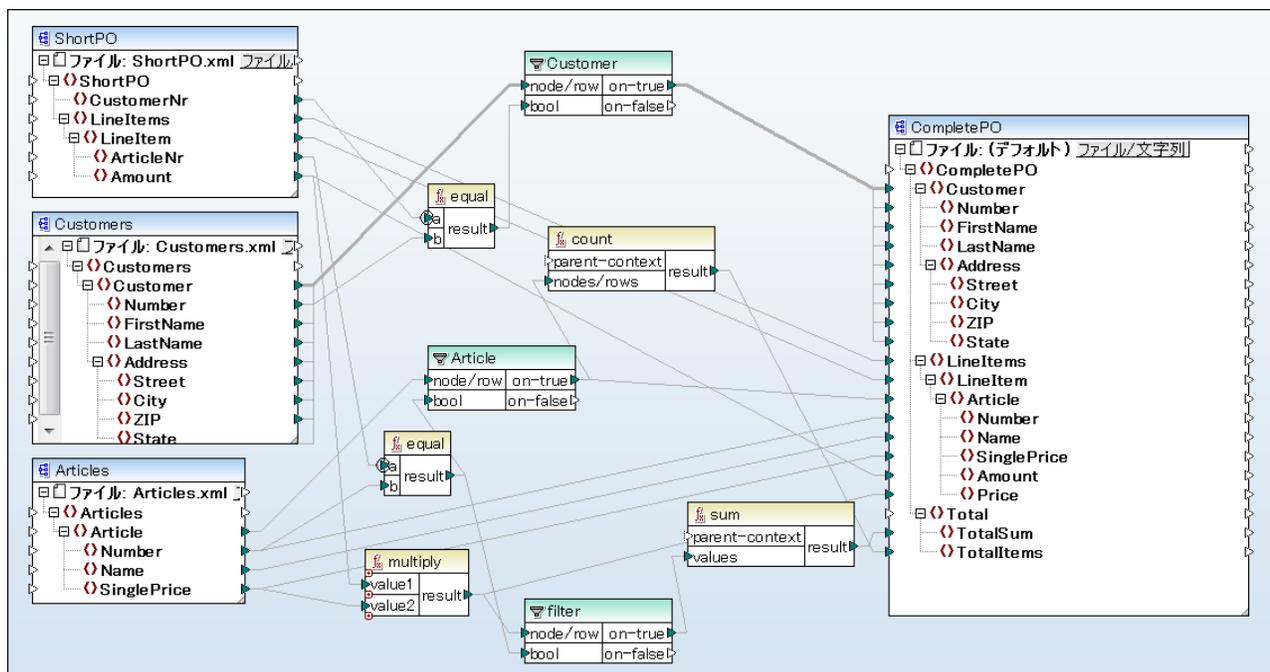
この例では MapForce マッピングから FlowForce Server ジョブを作成する方法が説明されています。最初に MapForce から FlowForce Server でもマッピングファイルをデプロイします。マッピングが FlowForce Server にデプロイすると、サーバージョブをこれから作成します。ジョブは毎日特定の時間に実行されるように構成されています。

### 必要条件

- 必要とされるライセンス MapForce Enterprise または Professional edition、MapForce Server または MapForce Server Advanced Edition、FlowForce Server
- FlowForce Web Server と FlowForce Server サービスが構成されたネットワークアドレスとポート でリスニングしていること。
- コンテナ の一つでパーミッションを有する FlowForce Server ユーザーアカウントを有していること(デフォルトでは、この例で使用されている /public コンテナには、認証されたユーザーがアクセスすることができます)。
- この例で作成されたマッピングジョブは XML ファイルを実行する都度作成します。ですから、FlowForce Server が作動するオペレーティングシステムでデレクトリ内にファイルを作成する権利を持つ必要があります。(この例では、C:\FlowForceExamples\Mapping デレクトリを使用します)。

### 使用されるデモファイル

このサンプルで使用されているマッピングファイルは CompletePO.mfd と呼ばれ、MapForce がインストールされているマシン上で以下のパスで見つけることができます: <Documents>\Altova\MapForce2021\MapForceExamples。MapForce を最初に起動する場合 [MapForceExamples] デレクトリが作成されることにご注意ください(起動前には作成されません)。



CompletePO.mfd

上で説明されているでもマッピングおつのXML ファイルを入力として取り、出力として単一のXML ファイルを作成します。このサンプルでは入力 XML ファイルは自動的に FlowForce Server にデプロイされたパッケージに含まれます。FlowForce Server にマッピングをデプロイするで説明される通り、他のマッピングはデプロイ前に追加の準備ステップを必要とする場合があります。

## ジョブの作成

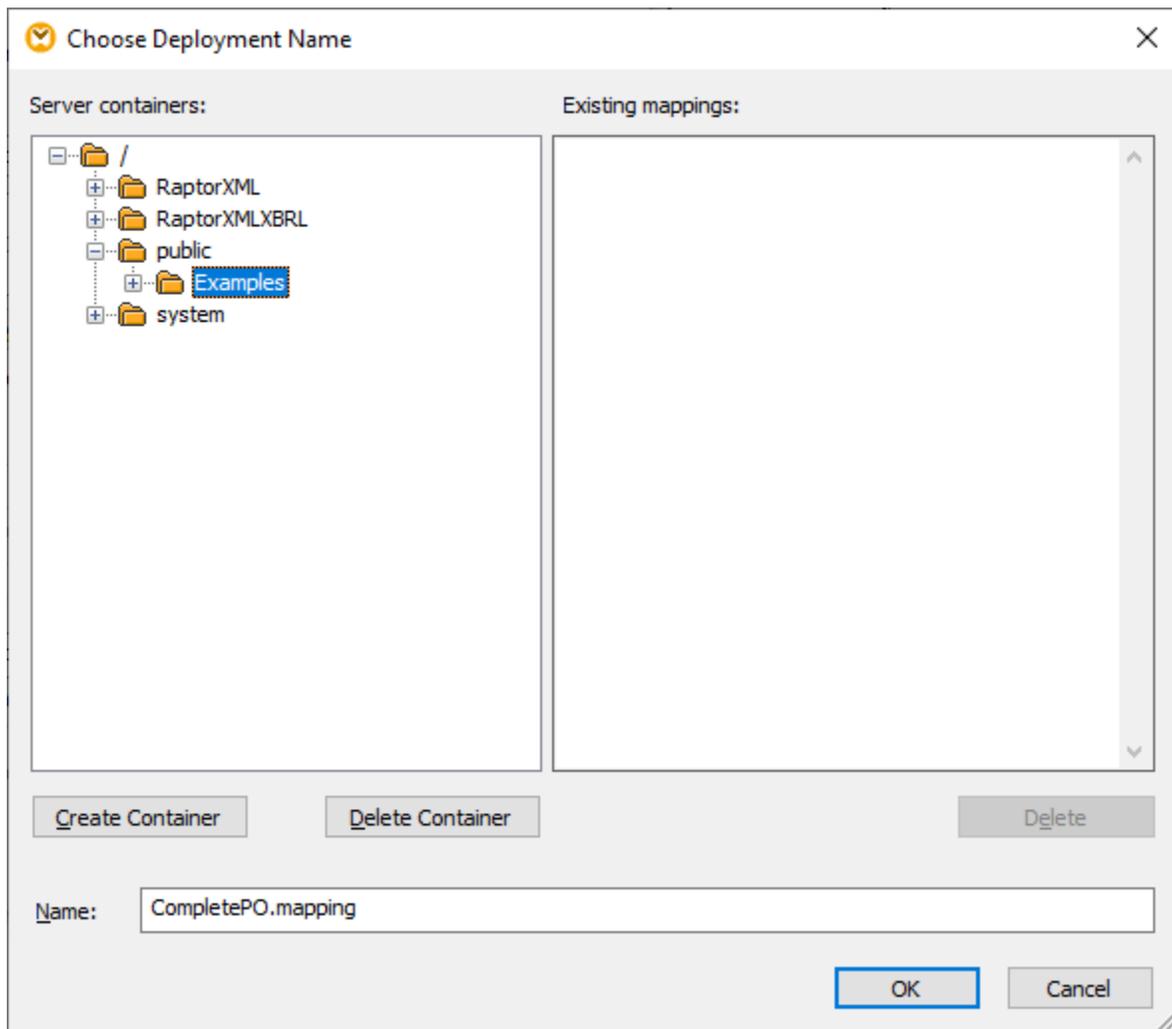
マッピングのデプロイはMapForce がマッピングに使用されるリソースを単一のパッケージに整理し、HTTP（または構成されている場合 HTTPS）を介してFlowForce Server に送信することを意味します。

### FlowForce Server にマッピングをデプロイする

1. MapForce 内で **CompletePO.mfd** ファイルを開きます。
2. おざりではない場合、マッピングの変換言語を「ビルトイン」に設定します。
3. 「ファイル」メニューから、「FlowForce Server にデプロイする」をクリックします。
4. サーバーとポート テキストボックス内に Web 管理インターフェイスのサーバー名とポートを入力します（例えば、FlowForce Web Server サービスがデフォルトのポートで同じマシン上でホスティングの場合では 127.0.0.1 と 8082 など）。異なるアドレスとポートが構成されている場合これらの値を変更してください。 [ネットワーク設定の定義](#)を参照してください。
5. **User** と **Password** テキストボックスにFlowForce Server ユーザー名とパスワードを入力します。
6. デレクトリをログインドロップダウンリストから選択、または <Default> オプションをそのままします。

デレクトリサービス統合が有効化されていると、ドメインユーザー名とパスワードを入力し、ドメイン名をログインドロップダウンリストから選択します。詳細に関しては、[デレクトリサービス設定の変更](#)を参照してください。

7. サンプル間の一貫性を保つためマッピングは **/public/Examples** コンテナにデプロイされます。 **Browse** をクリックしてコンテナのパスを **/public/Examples** に変更します。前のサンプルに従っている場合 **/public/Examples** コンテナは既存であるべきです。それ以外の場合、下のダイアログボックス内のコンテナの作成 をクリックして作成します:



8. [Web ブラウザーを開き新規ジョブを作成する] チェックボックスを選択します。

Deploy Mapping

Enter the host name and port of a FlowForce Administration Interface to deploy the current mapping.

Server: localhost Port: 8082

User: root  Use SSL

Password: ●●●●

Login: Directly

Global Resources on the Server

Use Resources

Resource Path:  ...  
The path must start with a slash character.

Deploy As

Path: /public/Examples/CompletePO.mapping   
The path must start with a slash character.

Save mapping before deploying

Open web browser to create new job

OK Cancel

9. **[OK]** をクリックしてマッピングをデプロイします。

デプロイが完了すると、FlowForce Server 管理インターフェイスは Web ブラウザーを開き、部分的に記入されたジョブのページが表示されます。マッピング関数自体は以前に指定済みのコンテナ内に保存されます。これでデプロイの過程は完了しました。

### ジョブの作成方法

上記の通り FlowForce Server にマッピングファイルをデプロイした後、ブラウザーは部分的に記入されたジョブページを表示します。入力済みのパラメータと共に最初の実行ステップが自動的に作成されます。

## 以下にジョブを作成 /public

ジョブ名:

ジョブの説明:

### ジョブ入力パラメーター

+

### 実行ステップ

+

関数の実行

パラメーター:

|                    |      |  |  |
|--------------------|------|--|--|
| Customers:         | (入力) |  |  |
| Articles:          | (入力) |  |  |
| ShortPO:           | (入力) |  |  |
| CompletePO:        | (出力) |  |  |
| Working-directory: |      |  |  |

= ステップの結果を割り当て  以下として CompletePO

ジョブを関数のページ (/public/Examples/CompletePO.mapping) を開いてジョブを作成し、ジョブの作成をクリックします。

ジョブを構成する方法:

1. デフォルトのジョブ名を [CompletePO.job] から詳細な名前に変更します。例えば [GeneratePurchaseOrder] など。これは任意のステップですが、名前が同じコンテナ内の他のジョブにより使用されている場合必要なステップです。
2. デフォルトで作成された最初の実行ステップを以下のように入力します:

|        |   |
|--------|---|
| 実行関数   | このフィールドには以前にデプロイされたマッピング関数を指しています。そのままコピーしてください。  |
| パラメーター | <b>Customers</b> 、 <b>Articles</b> 、および <b>ShortPO</b> フィールドはジョブメッセージ済みの対応するXML フィールドを含んでいます。 |

|              |  |
|--------------|--|
|              | <p><b>CompletePO</b> フィールドは出力ファイルのパスを指定します。デフォルトでは <b>CompletePO.xml</b> です。以下で説明されている通り、パスは作業ディレクトリに対して相対的です。</p> <p>このサンプルでは、全ての入力と出力オプションをそのまま保持することができます。入力と出力インスタンスの変更に関する詳細は <a href="#">ジョブとしてマッピングを実行し変換する</a> を参照してください。</p> <p>作業ディレクトリボックス内に作業ディレクトリのパスを入力します。この例は、ジョブの作業ディレクトリとして <b>C:\FlowForceExamples\Mapping</b> を使用しています。</p> <p>作業ディレクトリとは、ジョブが入力ファイルを展開、または、出力ファイルを保存する場所が必要な場合の <a href="#">実行ステップ</a> により必要とされるパラメータです。FlowForce Server は、また、作業ディレクトリをステップの実行中に発生する相対パスを解決するために、使用します。作業ディレクトリを与えるように問われた場合、FlowForce Server が実行されているオペレーティングシステム上の有効なパスを与えてください。ステップを作成中に、作業ディレクトリが与えられない場合、FlowForce Server は一時的なディレクトリを使用します。</p> |
| ステップの結果を割り当て | このフィールドはマッピングの結果に名前を与えます。このサンプルでは、このフィールドを空白のままにします。   |

- トリガーから「新規タイマー」をクリックします。
- 実行の横から、タイマーを毎日、1日毎に実行するように設定します。開始の横から、ジョブが開始される日時を選択します。例

- 「資格情報」から、既存の資格情報の記録を選択、または、ローカルの資格情報を指定してください。詳細に関しては [資格情報](#) を参照してください。

## 資格情報

資格情報によりジョブを実行:  既存の資格情報を選択:

ローカルの資格情報を定義: ユーザー名:

パスワード:

6. [保存]をクリックします。

### ジョブを実行する方法

トリガー内で指定された日付で、FlowForce Server マッピングジョブを実行します。ジョブの実行に成功すると結果として生成されたファイル(CompletePO.xml)はC:\FlowForceExamples\Mapping ディレクトリ内で使用できるようになります。ジョブの実行が成功したかを確認するには [ジョブログ](#) を参照します。

## 12.5 ジョブを他のジョブのステップとして使用する

この例では、他のジョブのステップとして定義されたジョブを使用する方法が説明されています。このサンプルは、前に作成されたジョブを必要とするため、サンプルをこのサンプルを完了する前に [MapForce マッピングからジョブを作成する](#) を完了すること奨励されます。

[MapForce マッピングからジョブを作成する](#) サンプルから思い出せるかもしれませんが、**GeneratePurchaseOrder** は実行される都度、XML ファイルを一時フォルダーに生成します。このサンプルは以下を行う方法について説明しています：

1. マッピングにより生成されたファイルをコピーするジョブを作成します。このジョブを **CopyOutput** と呼びます。
2. **GeneratePurchaseOrder** を変更し **CopyOutput** ジョブを追加実行ステップとして追加します。

### 必要条件

- 必要とされるライセンス MapForce Enterprise または Professional edition、MapForce Server または MapForce Server Advanced Edition、および FlowForce Server
- FlowForce Web Server と FlowForce Server サービスが構成された [ネットワークアドレスとポート](#) でホストされていること
- [コンテナの一つへのパーミッション](#) を持つ FlowForce Server ユーザーアカウント（デフォルトでは /public コンテナはすべての認証済みのユーザーによりアクセスすることが可能です）。
- このサンプルで作成されたマッピングジョブはファイルを一つのディレクトリから他のディレクトリにコピーします。FlowForce Server が作動するオペレーティングシステムでは、両方のディレクトリが存在している必要があり、これらのディレクトリ内にファイルを作成する権利を有していることを確認してください。このサンプルは C:\FlowForceExamples\Mapping と C:\FlowForceExamples\Archive ディレクトリを使用しています。
- [MapForce マッピングからジョブを作成する](#) 内で説明されているステップを完了してください。

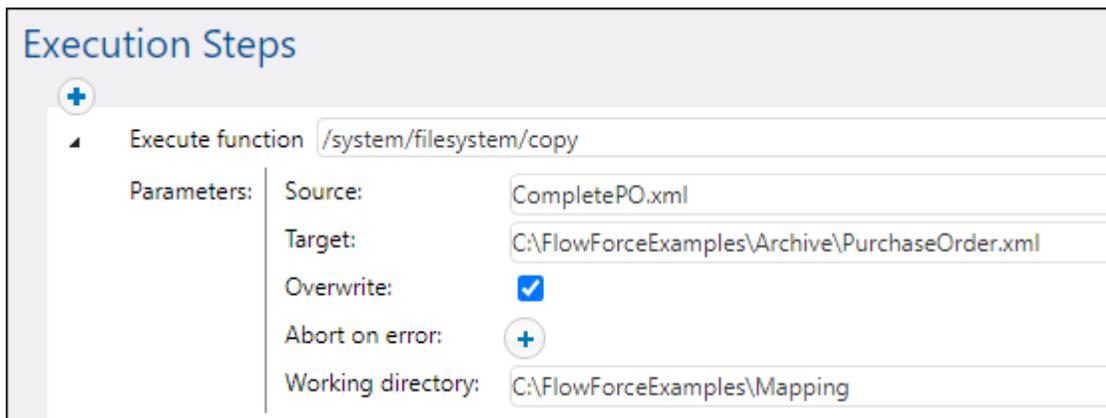
### ジョブの作成方法

1. 「構成」をクリックし /public/Examples コンテナに移動します。前のサンプルを試している場合 public/Examples コンテナは既存のはずです。それ以外の場合 [作成 | コンテナの作成](#) コマンドを使用して作成してください。
2. 「作成」をクリックし、「ジョブの作成」を選択します。
3. ジョブの名前を入力します（このサンプルでは "CopyOutput"）。

### Create job in /public/Examples

|                  |  |
|------------------|--|
| Job name:        | <input type="text" value="CopyOutput"/>  |
| Job description: | <input type="text" value="Copies the output of a mapping to another directory"/> |

4. 実行ステップの下で、次の設定を持つ最初の実行ステップを追加します：



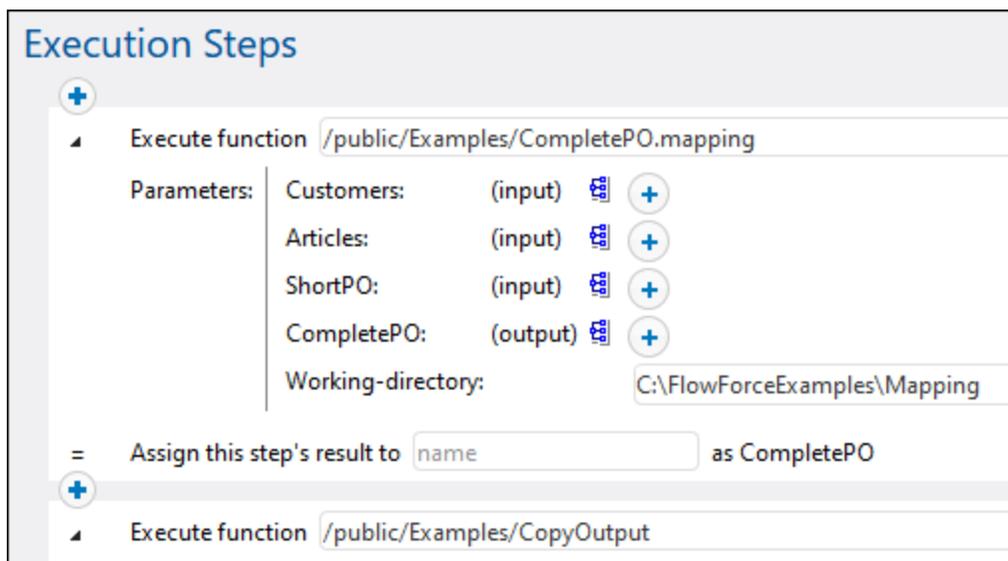
|            |  |
|------------|--|
| 実行関数       | <code>/system/filesystem/copy</code> 関数を参照します。   |
| ソース        | <b>CompletePO.xml</b><br>作業ディレクトリ/パラメータが設定されていないため、相対的なパスを使用します。   |
| ターゲット      | これは FlowForce Server が作動するオペレーティングシステムの既存のファイルまたはディレクトリのパスである必要があります。このサンプルでは、コピーされるとファイルの名前を変更し、ファイル名をパス以下のように追加します:<br><b>C:\FlowForceExamples\Archive\PurchaseOrder.xml</b> |
| 上書き        | このチェックボックスを選択します。これは、FlowForce Server に保存先パスで検索された同じ名前を持つファイルを上書きするように命令します。   |
| エラー時には中断する | パラメータをそのままします。<br><br>このブールパラメータはジョブが失敗した場合の戻り値を決定します。「エラー時には中断」が FALSE の場合、関数は、ブール値の FALSE を返します。「エラー時には中断」が TRUE の場合、ジョブの実行は中断されます。デフォルトの値は、TRUE です。                         |
| 作業ディレクトリ   | FlowForce はすべての相対的なファイルパスをこのディレクトリで探します。以下に設定します:<br><b>C:\FlowForceExamples\Mapping</b>   |

- 「資格情報」から、既存の資格情報の記録を選択、または、ローカルの資格情報を指定してください。詳細に関しては [資格情報](#) を参照してください。
- [保存]をクリックします。

既知かもしれませんが、作成されたジョブはトリガーが存在しません。他のジョブからこのジョブを呼び出すため、トリガーは定義されていません。

### CopyOutput ジョブを他のジョブのステップとして追加する方法

- `/public/Examples` コンテナを **GeneratePurchaseOrder** から開きます。
- 実行ステップで、「新規実行ステップ」をクリックし、既存のステップの後に新規ステップを追加します。
- 実行関数の横で **CopyOutput** ジョブの場所を参照してください。実行ステップは以下のようになります:



4. タイムトリガーから、「保存」をクリックします。
5. トリガー内に入力された日時から、FlowForce Server は、ジョブを実行し、**CompletePO.xml** ファイルを指定されたディレクトリにコピーし、名前を **PurchaseOrder.xml** に変更します。ジョブの実行が成功したかを確認するには [ジョブログ](#) を参照します。

## 12.6 ディレクトリポーリング ジョブを作成する

このサンプルはFlowForce Server 内で作成されたファイルシステムリガーを使用してディレクトリでの変更を監視する方法について説明しています。( [ファイルシステムリガー](#) を参照してください)。新規 XML ファイルがディレクトリに追加されるとFlowForce Server はXML ファイルを入力パラメータとして取るマッピングジョブを実行します。マッピングジョブの出力は、アーカイブディレクトリに移動されます。

### 必要条件

- 必要とされるライセンス MapForce Enterprise またはProfessional edition、MapForce Server またはMapForce Server Advanced Edition、およびFlowForce Server
- FlowForce Web Server とFlowForce Server サービスが構成された [ネットワークアドレスポート](#) でホストされていること。
- [コンテナ](#) の一つのパーミッションを持つFlowForce Server ユーザーアカウント (デフォルトでは /public コンテナはすべての認証済みのユーザーによりアクセスすることが可能です)。
- このサンプルで作成されたマッピングジョブはファイルを一つのディレクトリから他のディレクトリにコピーします。FlowForce Server が作動するオペレーティングシステムでは、両方のディレクトリが存在している必要があります。これらのディレクトリ内にファイルを作成する権利を有していることを確認してください。このサンプルは C:\FlowForceExamples\DirPolling と C:\FlowForceExamples\Archive ディレクトリを使用しています。

### 使用されるデモファイル

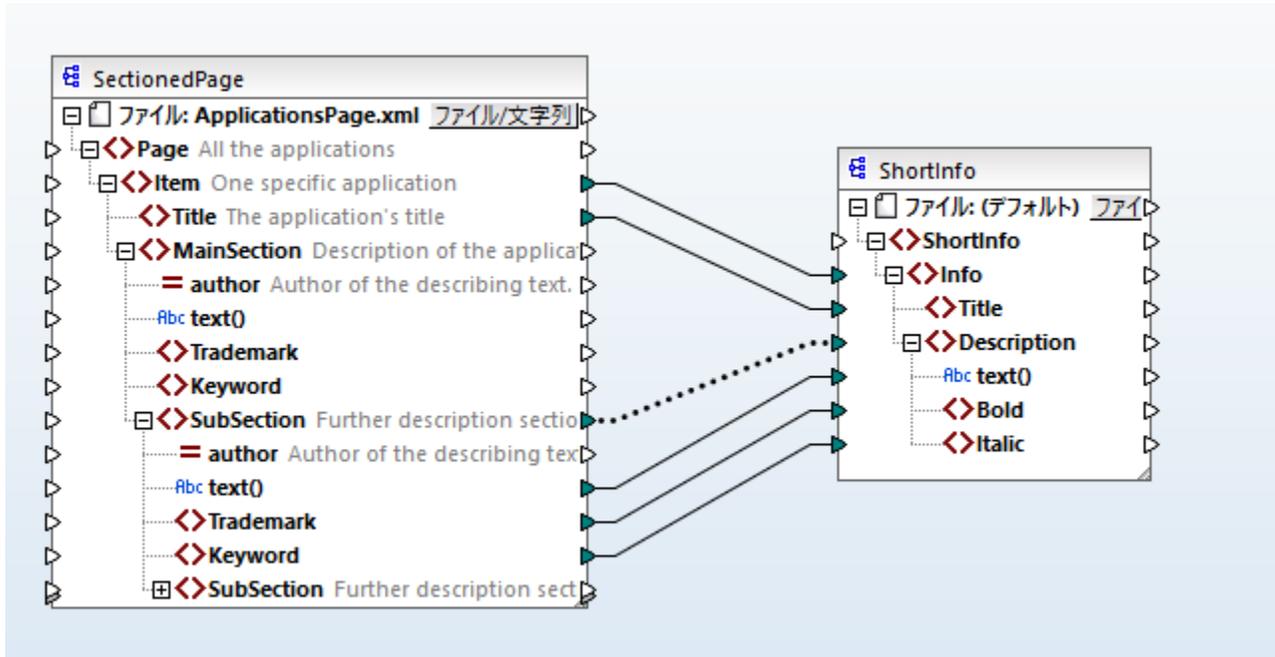
- **ShortApplicationInfo.mfd** – FlowForce Server ジョブを作成するMapForce マッピング
- **ApplicationsPage.xml** – マッピングに入力として与えられるXML インスタンスファイル。

MapForce がインストールされているマシン上の次のパスで両方のファイルを見つけることができます:  
<Documents>\Altova\MapForce2021\MapForceExamples\。

### マッピングのしくみ

このサンプル(**ShortApplicationInfo.mfd**) で使用されているMapForce マッピングは、以下のとおりです。FlowForce Server からの観点では、XML ファイルと入力としてマッピングが取りXML ファイルを出力として生成することが重要です。

マッピングはXML ファイル**ApplicationsPage.xml** を他のスキーマに変換し **ShortInfo.xml** として保存します。一番上の接続を確認するとマッピングは簡単に理解することができます。ソース内の各アイテムのため、ターゲット内で情報アイテムが作成されます。対応する子アイテムから値をコピーするため他の接続は使用されます。特定の関係がある項目は点線で接続されています。MapForce 内では、この接続は「ソース優先」と呼ばれ、サブセクションが複合型コンテンツを含むため使用されます。



ShortApplicationInfo.mfd

ソースとターゲット XML スキーマの名前はそれぞれ **SectionedPage** と **ShortInfo** であることにご注意してください。下に示されるように FlowForce ジョブは FlowForce Server にデプロイ後も同じ名前を入力と出力パラメータを持ちます。

### FlowForce Server にマッピングをデプロイする

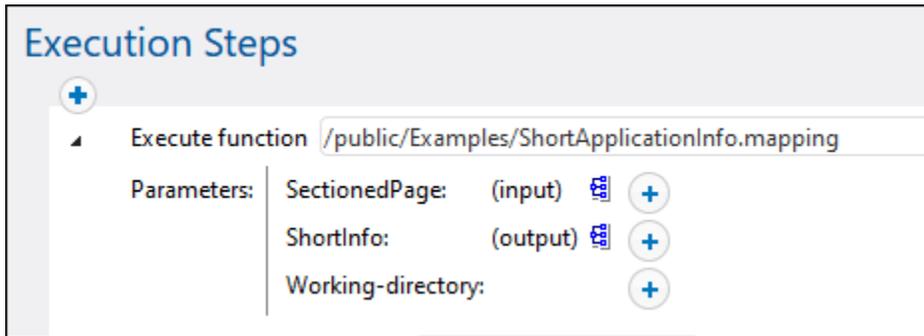
FlowForce Server にデプロイする前にマッピング **ShortApplicationInfo.mfd** は特別な準備を必要としません。ソースとターゲットコンポーネントが XML ファイルのため FlowForce Server にデプロイされるパッケージに自動的に含まれます。

FlowForce にマッピングをデプロイするには MapForce 内で開きメニューコマンド「ファイル | Deploy to FlowForce Server にデプロイする」を実行します。

FlowForce Server が異なるホストとポートで作動する場合、必要に応じて接続の詳細を変更してください。ネットワーク設定の定義を参照してください。マッピングのパスは `/public/Examples/ShortApplicationInfo.mapping` であることに注意してください。これは前のサンプルと一貫しています。

### ディレクトリポーリングジョブを作成する

FlowForce Server にマッピングをデプロイ後、ブラウザが開かれジョブ作成ページがロードされます。下に説明されているように最初の実行ステップは自動的に作成され、前にデプロイされたマッピング機能が呼び出されます。ターゲットコンポーネント (**ShortInfo**) と同じ名前が出力パラメータに与えられていますが、ソース MapForce コンポーネント (**SectionedPage**) と同じ名前が与えられていることに注意してください。



ジョブを次のように構成します:

1. 作業ディレクトリボックスに作業ディレクトリのパスを入力します。この例は、作業ディレクトリとして **C:\FlowForceExamples\DirPolling** を使用してします。

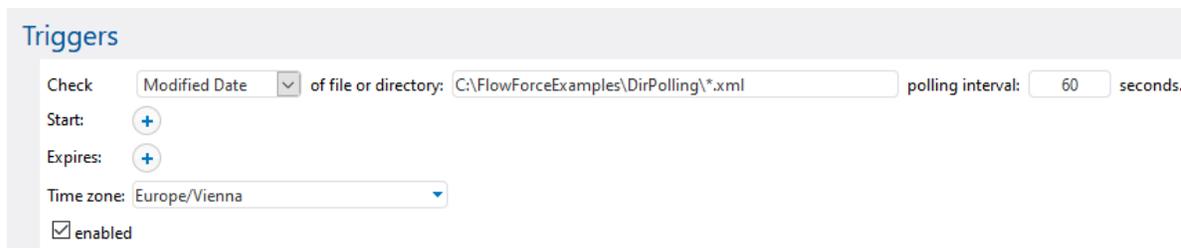
作業ディレクトリとは、ジョブが入力ファイルを展開、または、出力ファイルを保存する場所が必要な場合の **実行ステップ** により必要とされるパラメータです。FlowForce Server は、また、作業ディレクトリをステップの実行中に発生する相対パスを解決するために、使用します。作業ディレクトリを与えるように問われた場合、FlowForce Server が実行されているオペレーティングシステム上の有効なパスを与えてください。ステップを作成中に、作業ディレクトリが与えられない場合、FlowForce Server は一時的なディレクトリを使用します。

2. トリガーから「新規 ファイルシステムトリガー」をクリックします。FlowForce Server は、入力パラメータに新規 **triggerfile** / パラメータを自動的に追加することにご注意してください。このコンテナーを次のステップで参照する必要があります。

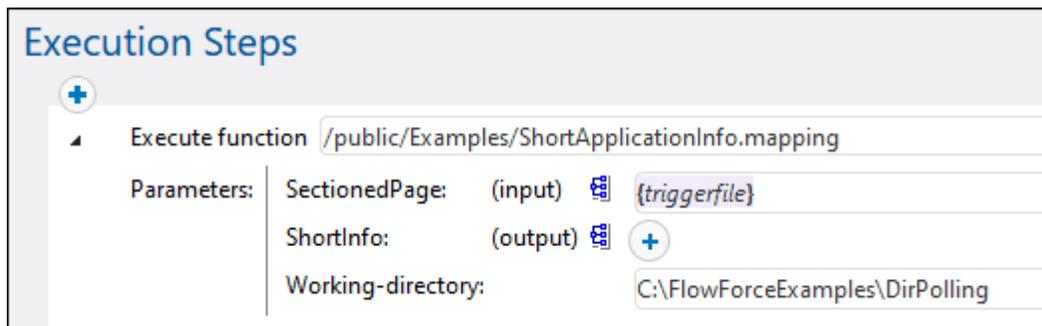


3. 次のトリガーの値を設定します:

- チェック 変更された日付
- ファイルまたはディレクトリ **C:\FlowForceExamples\DirPolling\\*.xml**
- ポーリング間隔 60 秒



4. 実行ステップで **triggerfile** / パラメータを **SectionedPage** / パラメータに対する入力値として与えます。これを行うには **SectionedPage** / パラメータの横の **Set to** ボタンをクリックし、**triggerfile** を選択します。この結果、**SectionedPage** / パラメータの値は、**{triggerfile}** に変更されます。中かこは FlowForce 式を表示し、削除される必要があります。



これまでの構成により、**ApplicationsPage.xml** が作業ディレクトリにコピーされると、トリガーが実行されます。しかしながら、トリガーはファイルカード (\*.xml) を使用するため (**ShortInfo.xml**) を含む他の XML ファイルがディレクトリ内で変更される場合でも実行されます。これは期待される振る舞いではなくエラーを引き起こします。ですから、新規ディレクトリから生成された出力ファイルを削除する 2 番目のステップを追加する必要があります。または、トリガーの名前を **C:\FlowForceExamples\DirPolling\ApplicationsPage.xml** に名前を変更することができます (この場合、2 番目のステップは必要ありません)。

新規ディレクトリに出力を移動するステップを追加するには以下を行います:

1. 前のステップのすぐ後に、新規実行ステップを追加します。
2. 次のようにステップを構成します (ソースと目的のフィールドは大文字と小文字を区別することにご注意してください):

|            |  |
|------------|--|
| 実行関数       | <code>/system/filesystem/move</code> 関数を参照します。   |
| ソース        | <b>ShortInfo.xml</b><br>作業ディレクトリ/パラメータが設定されていないため、相対的なパスを使用します。  |
| 保存先        | これは FlowForce Server が作動するオペレーティングシステムの既存のファイルまたはディレクトリのパスである必要があります。以下に設定します:<br><b>C:\FlowForceExamples\Archive</b>                                  |
| ターゲットの上書き  | このチェックボックスを選択します。これは、FlowForce Server に保存先パスで検索された同じ名前を持つファイルを上書きするように命令します。   |
| エラー時には中断する | パラメータをそのままします。<br><br>このブールパラメータはジョブが失敗した場合の戻り値を決定します。「エラー時には中断」が FALSE の場合、関数は、ブール値の FALSE を返します。「エラー時には中断」が TRUE の場合、ジョブの実行は中断されます。デフォルトの値は、TRUE です。 |
| 作業ディレクトリ   | FlowForce はすべての相対的なファイルパスをこのディレクトリで探します。以下に設定します:<br><b>C:\FlowForceExamples\DirPolling</b>  |

| Execute function /system/filesystem/move |                    |                                     |
|--|--------------------|-------------------------------------|
| Parameters:                              | Source:            | ShortInfo.xml                       |
|  | Destination:       | C:\FlowForceExamples\Archive        |
|  | Overwrite target:  | <input checked="" type="checkbox"/> |
|  | Abort on error:    | <input type="checkbox"/>            |
|  | Working directory: | C:\FlowForceExamples\DirPolling     |

最後にジョブの実行に使用される使用中のオペレーティングシステムの資格情報を追加します:

1. 「資格情報」から、既存の資格情報の記録を選択、または、ローカルの資格情報を指定してください。詳細に関しては [資格情報](#) を参照してください。
2. [保存]をクリックします。

### ジョブを実行する方法

**ApplicationsPage.xml** を作業ディレクトリコピーしてジョブをテストすることができます。これを行う際には、FlowForce Server はマッピングジョブを実行し、結果出力ファイルをアーカイブディレクトリコピーします。

ジョブの実行が成功したかを確認するには [ジョブログ](#) を参照します。

## 12.7 エラー処理をジョブに追加する

このサンプルジョブエラー処理を追加する方法について説明します。このサンプルで使用されるジョブはディレクトリのコンテンツをリストし、ブラウザから Web サービスとして呼び出されます。ジョブの結果を処理するために FlowForce Server をどのように構成するかについて学びます。

- ジョブの実行に成功すると、ブラウザ内でジョブの出力が表示されます。
- 何らかの理由でジョブが実行に失敗した場合、名前の挙げられた宛先に通知メールが送信されます。
- ジョブの実行が完了すると、実行の状態に関わらず、ローカルシステム上のファイルのジョブの内部 ID にログされます。

FlowForce Server では、この例では、次の 2 つのエラー処理条件のために [保護されたブロック](#) を作成します: 「エラー時」と「常に」(それぞれ上記のシナリオの内の 1 つを処理します)。

### 必要条件

- 必要とされるライセンス FlowForce Server
- FlowForce Web Server と FlowForce Server サービスが構成された [ネットワークアドレスとポート](#) でホストされていること。
- FlowForce Server メール設定が構成済みであること。 [電子メールパラメータの設定](#) を参照してください。
- [コンテナ](#) の一つの [パーミッション](#) を持つ FlowForce Server ユーザーアカウントが必要です (デフォルトでは /public コンテナはすべての認証済みのユーザーによりアクセスすることが可能です)。
- このサンプル内で作成されたジョブはディスクに出力を書き込みます。このために FlowForce Server がインストールされているオペレーティングシステム上でいくつかのディレクトリへの読み取りと書き込みの権利が必要になります。このサンプルは C:\FlowForceExamples\ErrorHandling を使用しています。

### ヒント

- この例は、Windows のバッチコマンドを使用していますが、バッチコマンドを必要に応じて変更すると、他のオペレーティングシステム上でテストすることができます。

### ジョブの作成方法

1. FlowForce Server が作動するマシン上で、ジョブ出力が保存されるディレクトリを作成します。このサンプルは C:\FlowForceExamples\ErrorHandling を使用しています。
2. FlowForce Server Web 管理インターフェイスにログインし /public/Examples コンテナを開き、「ジョブの作成」をクリックします。前のサンプルに従っている場合 /public/Examples コンテナは既存であるべきです。それ以外の場合「作成 | コンテナの作成」コマンドを使用して作成します。
3. ジョブの作成 をクリックして作成中のジョブの名前を入力します。例えば [ListDirectory] など。ジョブの説明は任意です。
4. ジョブ入力パラメータで  ボタンを押して型 [文字列] のパラメータを追加します。ジョブランタイムでは、パラメータはリストへのディレクトリのパスを提供します。このサンプルでは、パラメータの名前は [inputDir] で後のステップで使用されます。
5. 「実行ステップ」で、「新規のエラー/成功処理ステップ」をクリックします。
6. 「エラー/成功処理を使用して実行」から、  ボタンをクリックして、次の設定を持つ新規の実行ステップの追加を選択します:

|      |  |
|------|--|
| 実行関数 | <code>/system/shell/commandline</code> 関数を参照します。     |
| コマンド | 以下のシェルコマンドを入力します:<br><br><code>dir {inputDir}</code> |

|            |   |
|------------|---|
|            | <code>inputDir</code> が以前に作成されたパラメーターの名前である個所名前は中かっこに囲まれています。これはジョブのランタイムで、コンテンツが動的パラメーターの値を置き換えられるからです。詳細に関しては <a href="#">文字列フィールド内に式を埋め込む</a> を参照してください。 |
| エラー時には中断する | オプションをそのままします。詳細に関しては <a href="#">/system/shell/commandline</a> 関数を参照してください。  |
| 作業ディレクトリ   | 以前に作成された作業ディレクトリのパスを入力します。例えば <code>C:\FlowForceExamples\ErrorHandling</code>   |

7. 「エラー時」条件から、 ボタンをクリックして、次の設定を持つ新規の実行ステップの追加を選択します:

|      |  |
|------|--|
| 実行関数 | <a href="#">/system/mail/send</a> 関数を参照します。  |
| 差出人  | 差出人の電子メールアドレスを入力します。例: <code>flowforce@localhost</code> 。管理ページから電子メールの設定を構成した場合、このフィールドを空のままします。   |
| 宛先   | 電子メールアドレスを入力します。   |
| 件名   | 通知電子メールの件名を次のように入力します:<br><br><pre>Job {instance-id()} has failed</pre> <p>現在の(失敗した) ジョブインスタンスの一意のID を取得する中かこの間の部分は <a href="#">instance-id</a> 関数を呼び出す FlowForce 式です。</p>   |
| 本文   | 以下を入力します:<br><br><pre>Exit Code: {string(exitcode(failed-step()))} Standard Error: {content(stderr(failed-step()))} Error message: {error-message(failed-step())}</pre> <p>中かこの間の部分は2つのFlowForce 式です。これらの式はエラーのある出力を取得し、電子メールの本文となる文字列に変換します:</p> <ul style="list-style-type: none"> <li>• <a href="#">failed-step</a> 関数は、失敗したステップの <code>result</code> を返します。<code>exitcode</code>、<code>stderr</code>、または <code>error-message</code> 関数への引数として提供される更に使用しやすいようになるこれは抽象的な FlowForce 型です。</li> <li>• <a href="#">exitcode</a> 関数は終了コードが存在することを前提に数値として <code>result</code> から実際のエラーコードを取得します。</li> <li>• <a href="#">stderr</a> 関数はストリームとして <code>result</code> からエラーの標準エラー出力を取得します。</li> <li>• <a href="#">error-message</a> 関数は FlowForce エラーメッセージのテキストをログに表示されたいくつか取得します。エラーがない場合、または、エラーテキストを技術的に取得できない場合、空の文字列が返されます。</li> <li>• <a href="#">string</a> 関数は数値の終了コードを文字列に変換します(これは電子メールの本文が <code>string</code> 型であるからです)。</li> <li>• <a href="#">content</a> 関数はストリームからのエラー出力を文字列に変換します。</li> </ul> <p>実行が終了コードとエラー出力を生成した場合それぞれ <code>exitcode</code> と <code>stderr</code> 関数は値を返します。これはコマンドラインにより生成されたものなどエラーの場合通常発生します。</p> |

**error-message** 関数は情報提供のために、全てのジョブ構成、および遭遇するエラーの状態のためにエラーのテキストを返すことは保証されていません。

8. 「新規のエラー/成功処理」をクリックして、「常に」を選択します。
9. 「常に」条件で  ボタンをクリックし、次の設定を持つ新規の実行ステップを追加します:

|          |  |
|----------|--|
| 実行関数     |  <a href="#">/system/shell/commandline</a> 関数を参照します   |
| コマンド     | 以下のシェルコマンドを入力します:<br><br><div style="border: 1px solid gray; padding: 5px; width: fit-content; margin: 5px auto;"> <pre>echo {instance-id()} &gt;&gt; JobLog.txt</pre> </div> Windows 上では、このコマンドはジョブID を <b>JobLog.txt</b> と呼ばれるファイルに書き込みます。ファイルにデータが含まれる場合、新規のテキストは既存のデータに追加されます。 |
| 作業ディレクトリ | 以前に作成された作業ディレクトリへのパスを入力します(C:<br><b>\FlowForceExamples\ErrorHandling</b> )。<br><br>このディレクトリは <b>JobLog.txt</b> ファイルへのパスを解決するために使用されます。   |

この段階では、ジョブは以下ようになります(異なるパスまたは、シェルコマンドが使用されていないことが仮定されます)。

エラー/成功処理にて実行

+  
 ▲ 関数の実行 /system/shell/commandline

パラメーター: コマンド: dir/s  
 エラー時には中断: +  
 作業用ディレクトリ: C:\FlowForce

= ステップの結果を割り当て 名前 以下として結果

+  
 エラー時 do

+  
 ▲ 関数の実行 /system/mail/send

パラメーター: 差出人: flowforce@localhost  
 宛先: to@example.org  
 件名: Job{instance-id()}has failed  
 本文: Exit code:{string(exitcode(failed-step()))  
 Standard Error:{content(stderr(failed-step()))}  
 添付: +  
 エラー時には中断: +

= ステップの結果を割り当て 名前 以下としてブール値

+  
 常に do

+  
 ▲ 関数の実行 /system/shell/commandline

パラメーター: コマンド: echo{instance-id()}>>JobLog.txt  
 エラー時には中断: +  
 作業用ディレクトリ: C:\FlowForce

10. ジョブを Web サービスにするには「HTTP によりこのジョブを使用できるようにする」チェックボックスを選択し、Web サービスの名前を入力します。例

### トリガー

実行  毎  日

繰り返し 毎  分  全日、または  開始時間  終了時間

開始:

有効期限:

タイムゾーン:

が有効化されました

サービス名に注意してください。Web サービスを呼び出す必要があります。

11. 「資格情報」から、既存の資格情報の記録を選択、または、ローカルの資格情報を指定してください。
12. [保存]をクリックします。

### ジョブを実行する方法

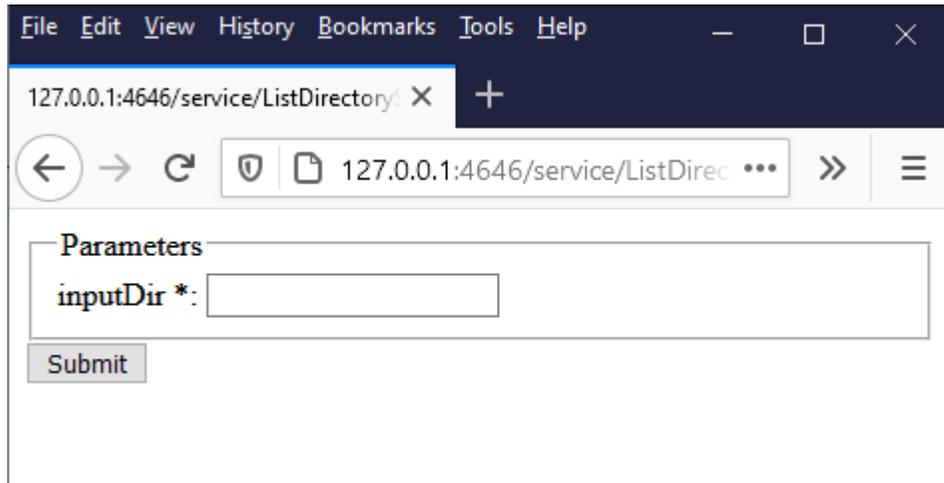
この段階ではジョブ構成が完了しました。Web サービスとしてこのジョブは公開されており以下の方法で実行することができます。

- ホームに移動し 全てのアクティブなトリガーとサービスを表示するをクリックします。次に情報列内に表示されているジョブの URL をクリックします。
- <http://127.0.0.1:4646/service/ListDirectoryService> をブラウザのアドレスバーに入力します。この URL は FlowForce Server サービスがデフォルトのホスト アドレスとポート名でアクセスする場合のみ機能します。構成ページ内で他のホストとポート設定を定義した場合、アドレスを必要に応じて変更してください。
- FlowForce Server の任意のホスト名 フィールドを [セットアップページ](#) から設定すると HTTP を介してこのジョブを使用可能にする チェックボックスの横の  ボタンをクリックして web サービスの呼び出しを直接ジョブ構成ページから実行することができます。それ以外の場合、このボタンは表示されません。

Web サービスにアクセスする際に資格情報をプロンプトされた場合、FlowForce Server にログインするために使用する同じ資格情報を提供してください。

FlowForce Server ユーザーに HTTP 認証のための資格情報を提供することはテスト 目的のためです。生産のために新規の FlowForce ユーザーを作成することが奨励されており、ジョブが存在するコンテナ上でこのユーザーに **Service - Use** パーミッションを与え、対応するユーザーアカウントを使用して Web サービスにアクセスします。HTTP 認証を無効化することは Web サービスを公開し、**Service - Use** パーミッションをユーザー **Anonymous** に与えます。[パーミッションのしくみ](#)を参照してください。

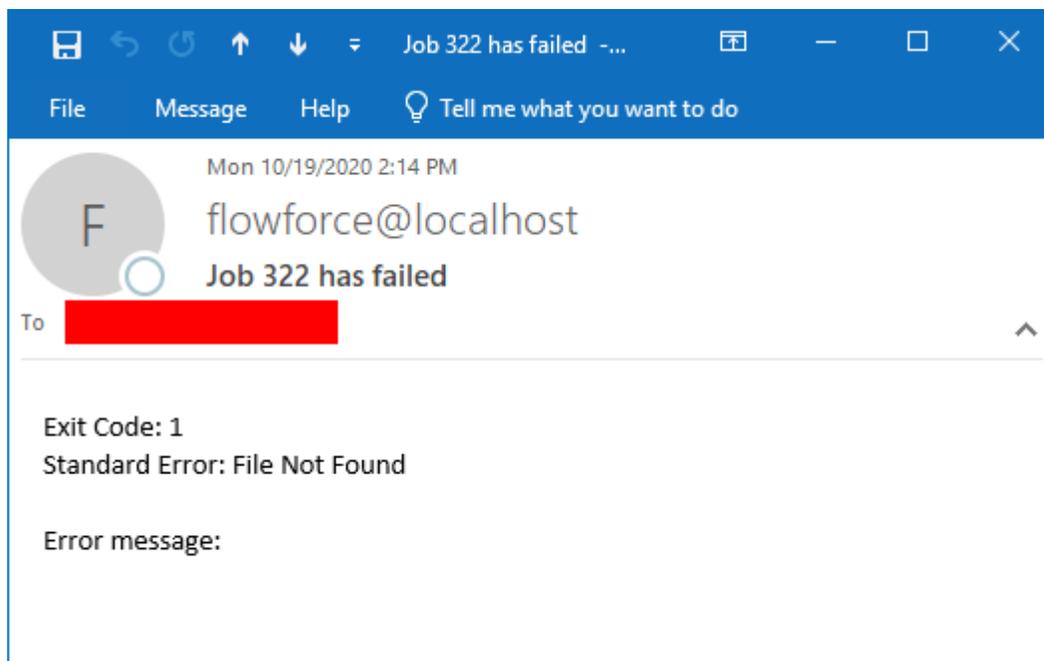
ジョブのパラメータを取り、ブラウザから Web サービスにアクセスするようパラメータ値を提供するようプロンプトされます。



C:\ のような有効なディレクトリを入力すると、例えば、ジョブは実行され結果がブラウザに表示されます。

ジョブを実行する都度、ジョブインスタンスのID は前に構成された[常]条件に従い **JobLog.txt** ファイルのコンテンツに追加されます。

[エラー時]条件で、[inputDir] パラメータを意図的に例えば、存在しないパスなど他の無効な値に変更します。この場合ブラウザはエラーを表示し FlowForce Server は[エラー時] ハンドラーの受取人フィールド内で指定されているアドレスに電子メールを送信します。例えば、電子メールは以下のようになります:



以前に説明されている通りこのサンプルで使用されているエラー関数がすべてのジョブ構成のために値を返すことは保証されていません。このために、電子メールにより提供される詳細のレベルは上部構成と遭遇するエラーにより異なります。そして、終了コード、標準エラー、およびエラーメッセージ 電子メールフィールドが常にテキストを含んでいるとは期待できません。エラーの原因として最も権限のあるリファレンスは FlowForce Server [log](#) です。

## 12.8 Web サービスとしてジョブを公開する

この例では、Web サービスとしてのサンプルマッピングジョブを公表する方法が説明されています。MapForce を使用してこのサンプルマッピングはデザインされており、個人のレコードを補完する Microsoft Access データベースからデータが読み取られます。マッピングはデータベースからパラメータとして与えられる特定の文字で始まる姓を持つ個人のレコードのみが取得されます。(同じ、または異なるマシン上で MapForce から FlowForce Server へ既存のマッピングをデプロイし、Web サービスへ変換する方法について学ぶことができます。このサンプルが終わると、Web サービスをブラウザから呼び出すことができます。

### 必要条件

- 必要とされるライセンス MapForce Enterprise または Professional edition、MapForce Server または MapForce Server Advanced Edition、FlowForce Server
- FlowForce Web Server と FlowForce Server サービスが構成された [ネットワークアドレスとポート](#) でリスンされていること
- [コンテナ](#) の一つの [パーミッション](#) を持つ FlowForce Server ユーザーアカウント (デフォルトでは /public コンテナはすべての認証済みのユーザーによりアクセスすることが可能です)。
- このサンプルで使用されているマッピングは Microsoft Access データベースからデータを読み取ります。Microsoft Access または Microsoft Access Runtime (<https://www.microsoft.com/en-us/download/details.aspx?id=50040>) を FlowForce Server が作動しているマシン上インストールする必要があります。

### 使用されるデモファイル

このサンプルは以下のファイルを使用しており、MapForce がインストールされているコンピューター上で次のパスを使用して見つけることができます: ..\Documents\Altova\MapForce2021\MapForceExamples.

- DB\_PhoneList.mfd (MapForce マッピングデザインファイル)
- altova.mdb (マッピングがデータを読み取る Microsoft Access データベース)。

### 異なるマシンへマッピングをデプロイする準備

このマッピングはデータベースファイルからデータを読み取るため、このセクションで説明されているとおりの追加の構成がマッピングのデプロイの前に行われる必要があります。MapForce と FlowForce Server が同じコンピューター上にインストールされている場合、下記の「マッピングのデプロイ」のセクションをスキップすることができます。

**メモ** 「ソースマシン」という用語は、MapForce がインストールされているコンピューターを指しており、「ターゲットマシン」は FlowForce Server がインストールされているマシンを指しています。

ターゲットマシンにマッピングをデプロイする前に、以下を行ってください！

1. ターゲットマシン上では「FlowForce Web Server」サービスをすべてのインターフェイス、またはローカルホスト以外の特定の IP アドレス上でリスンするように構成します。[ネットワーク設定の定義](#) を参照してください。このサービスが正確に構成されているか、ブラウザから次の URL にアクセスして確認することができます: `http://<FlowForce Web Server><port>`。指定されているアドレスとポートが受信する接続がファイアウォールによりブロックされていないことを確認してください。
2. 上記の通り、このサンプルで作成されるジョブは Web サービスとして使用できる必要があります。FlowForce 内では Web サービスとして公開されているジョブへのリクエストは FlowForce Web Server サービスではなく「FlowForce Server」サービスに処理されます ([リンク](#)も参照してください)。このため、Web サービスがローカルホストの外部の HTTP クライアントにアクセス可能な場合、「FlowForce Server」サービスはすべてのインターフェイス上で、またはローカルホスト以外の特定のアドレスによりリスンできるように構成されている必要があります。このサービスが正確に構成されているか、ブラウザから次の URL にアクセスして確認することができます: `http://<FlowForce Server><port>/service/`。Web サービスとして公開されているすべてのジョブは (存在する場合) ブラウザーウィンドウ内に直接リンクとして表示されます。

デプロイする前に、マッピングは絶対パスの代わりに相対パスを使用するように再度構成される必要があります。

1. **DB\_PhoneList.mfd** を MapForce 内でマッピングを開き、マッピングエリアを右クリックし、コンテキストメニューから「マッピングの設定」を選択します。
2. 「生成されたコードでは絶対パスを使用する」チェックボックスをクリアします。

3. マッピングを保存します。

Microsoft Access または SQLite などのファイルベースのデータベースはマッピングと共にターゲットマシンにデプロイされません。このため Access データベースは手動でソースマシンからターゲットマシンにコピーされる必要があります。ソースマシン上のディレクトリ.. \Documents\Altova\MapForce2021\MapForceExamples から altova.mdb データベースファイルをコピーし、ターゲットマシン上の空のディレクトリに張り付けます。このサンプルでは、ターゲットディレクトリは「C:\FlowForceWorkingDir」です。このディレクトリは後に FlowForce ジョブで参照されます。

マッピングは FlowForce Server にデプロイされる準備ができています。データベース接続を含むマッピングのデプロイに関する情報は、[サーバー実行のためにマッピングを準備する](#)。

### マッピングのデプロイ

1. **DB\_PhoneList.mfd** を MapForce 内で開きます。
2. 「ファイル」メニューで、「FlowForce Server にデプロイする」をクリックします。このサンプルの目的のために、マッピングはデフォルトのパス(/public コンテナ)にデプロイされます。異なるマシン上のFlowForce Server にデプロイする場合、「localhost:8082」からサーバーアドレスとポートをFlowForce Server から構成されたものに変更してください(上記を参照してください)。

マッピングのデプロイ

現在のマッピングをデプロイするため FlowForceWeb 管理インターフェースのホスト名とポート番号を入力してください。

サーバー (S): localhost      ポート (O): 8082

ユーザー (U): root       SSL を使用

パスワード (W): ●●●●

ログイン (L): <デフォルト>

デプロイ

パス (P): /public/CompletePO.mapping      参照 (B)

パスはスラッシュから開始する必要があります。

デプロイ前にマッピングを保存 (A)

Web ブラウザーを開き新規のジョブを作成する

OK      キャンセル

### FlowForce ジョブの作成方法

FlowForce Server にマッピングがデプロイされ、ブラウザ内でジョブ構成ページが表示されています(上記のダイアログボックスで「新規のジョブを作成するために Web ブラウザーを開く」チェックボックスが選択されていることが前提です)。それ以外の場合、FlowForce Server Web 管理インターフェイスにログインし、(/public コンテナである)前にデプロイされたマッピング関数を開き、「ジョブの作成」をクリックします。

### ジョブを構成する方法:

1. 入力パラメータから、型 文字列 の新規入力パラメータを作成し、NamePrefix と名前をつけます。

ジョブ入力パラメータ

| 名前 | 型   | デフォルト | 説明 |
|----|-----|-------|----|
|    | 文字列 |       |    |

2. 実行ステップで、NamePrefix の横の「設定」をクリックして、NamePrefix を選択します。マッピングパラメータの値 NamePrefix を前のステップで作成された NamePrefix 入力パラメータの値に設定します。
3. Working-directory の横に「C:\FlowForceWorkingDir」を入力します(これは Access ファイルが以前にコピーされているディレクトリと同じである必要です)。

- サービスから「このジョブを HTTP を介して URL 上で利用可能にする」チェックボックスをクリックして選択し、GetPhoneList をサービスの名前として入力します。

- 資格情報から、既存の資格情報の記録、または、ローカルの資格情報を指定します。[資格情報](#) も参照します。

**メモ** この資格情報を FlowForce Server Web 管理インターフェイスへのアクセスに使用される資格情報特別してください。また、ここユーザーが入力する内容が作業ディレクトリから altova.mdb データベースファイルにアクセスできるように注意してください。それ以外の場合、ジョブは成功せずに失敗します。

- [保存] をクリックします。

## Web サービスの呼び出し

作成された Web サービスを以下のように呼び出すことができます:

- ( [FlowForceServer] と [Port] を管理ページ内で構成された設定と置き換えます):

```
http(s)://[FlowForceServer]:[Port]/service/GetPhoneList
```

FlowForce Server の任意のホスト名 フィールドを [セトアップページ](#) から設定すると HTTP を介してこのジョブを使用可能にする

チェックボックスの横の  ボタンをクリックして web サービスの呼び出しを直接ジョブ構成ページから実行することができます。それ以外の場合、このボタンは表示されません。

- 資格情報を与えるように問われると、FlowForce Server Web 管理インターフェイスにアクセスするために使用される資格情報を入力します。
- Web サービスのパラメータを入力するように問われると、(「F」で始まる姓を持つすべての個人を抽出することを前提として) F を入力します。

Parameters

NamePrefix \*:

Submit

4. 「送信」をクリックします。FlowForce Server はジョブを処理し、結果を返します。

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

- <PersonList xsi:noNamespaceSchemaLocation="C:/
/PersonList.xsd">
  - <Person>
    <First>Joe</First>
    <Last>Firstbread</Last>
    <Details>+1 (321) 555 5155 - 621</Details>
  </Person>
  - <Person>
    <First>Carl</First>
    <Last>Franken</Last>
    <Details>+1 (927) 555 0094 - 147</Details>
  </Person>
  - <Person>
    <First>Frank</First>
    <Last>Further</Last>
    <Details>+1 (321) 555 5155 - 471</Details>
  </Person>
</PersonList>

```

ジョブの実行に成功すると、ジョブの出力は直接ブラウザに表示されます（また、作業デスク外リ内で出力は生成されます C:\FlowForceWorkingDir）。それ以外の場合、実行エラーが発生すると、詳細のためジョブのログを参照してください（以下を参照 [ジョブのログを確認する](#)）。

## 12.9 JSON を FlowForce Web サービスへポストする方法

このサンプルでは HTTP リクエストボディ内で JSON データを持つ POST リクエストを受け入れる FlowForce Web サービスを作成する方法が説明されています。第2に、MapForce などのクライアントから Web サービスを呼び出す方法について説明されています。

このサンプルでは Web サービスは JSON データを受け入れるように構成されます。XML または他のコンテンツを以下で示される方法に類似した方法で FlowForce Server により作成されたサービスにポストすることができます。Web サービスは JSON データを受け入れ、異なる処理無しにローカルで保存できるように簡素化されています。RaptorXML Server を使用して、JSON データを検証、または処理するようにジョブを拡張することもできます。

このサンプルは、ノリマターとしてではなく、HTTP リクエストのボディ内にデータがポストされる場合を特別に説明しています。Web サービスをノリマターを使用して呼び出すサンプルに関しては、[Web サービスを公開する](#)を参照してください。

### 必要条件

- 必要とされるライセンス FlowForce Server、MapForce Enterprise Edition。

#### コメント

FlowForce Server は Web サービスを作成する素早い方法を提供します。MapForce Enterprise Edition は FlowForce Server により作成される Web サービスを呼び出すクライアントとして振る舞います。異なるクライアントを使用しても、同じ結果を得ることができます。

- FlowForce Web Server と FlowForce Server サービスが構成された [ネットワークアドレスとポート](#) でリスニングしていること。
- [コンテナ](#) の一つのパーミッションを持つ FlowForce Server ユーザーアカウント（デフォルトでは /public コンテナはすべての認証済みのユーザーによりアクセスすることが可能です）。
- このジョブは、Web サービスにより受信された入力データをローカルの作業ディレクトリに保存します C:\FlowForceExamples\PostJson。このディレクトリ（または、類似したディレクトリ）が FlowForce Server が動作するマシン上で既存のマシンに存在し、オペレーティングシステムユーザーアカウントがこのディレクトリに書き込むことのできる権利を有する必要が求められます。

### FlowForce ジョブの作成方法

FlowForce Server Web 管理インターフェイスにログインし、/public/Examples コンテナを開き、「ジョブの作成」をクリックします。次に、名前を入力し、オプションで、作成する Web サービスのための詳細を入力します。

メモ 前のサンプルに従っている場合 public/Examples コンテナは既存であるべきです。それ以外の場合 作成 | コンテナの作成 コマンドを使用して作成します。

## 以下にジョブを作成 /public

ジョブ名:

ジョブの説明:

ジョブがPOST データを任意のコンテンツとして扱うようにするには、型 ストリームの一つのパラメータが必要になります。パラメータを作成するには、「パラメータの追加」 をクリックし、パラメータ名を入力します（このサンプルの場合「データ」）ストリームをデータ型として選択します。

### ジョブ入力パラメーター



名前: data

型: ストリーム ▼

説明: HTTP リクエストのボディ



次に、新規の実行ステップを追加し、以下のように構成します:

### 実行ステップ



▲
関数の実行
/system/filesystem/copy

|         |            |   |
|---------|------------|---|
| パラメーター: | ソース:       | <span style="border: 1px solid #ccc; padding: 2px 5px;">{as-file(data)}</span>          |
|         | ターゲット:     | <span style="border: 1px solid #ccc; padding: 2px 5px;">file{instance-id().json}</span> |
|         | 上書き:       | <span style="border: 1px solid #ccc; border-radius: 50%; padding: 2px 5px;">+</span>    |
|         | エラー時には中断:  | <span style="border: 1px solid #ccc; border-radius: 50%; padding: 2px 5px;">+</span>    |
|         | 作業用ディレクトリ: | <span style="border: 1px solid #ccc; padding: 2px 5px;">C:\POST</span>                  |

上記の実行ステップはFlowForce ビルトイン [copy](#) 関数を呼び出します。「ソース」テキストボックス内に表示される式は、[as-file](#) 式関数を使用してファイルするためにWeb サービスに受信された入力を変更します（入力パラメータは前のステップ内で、名前の付けられたデータであることを思い返してください）。この式を自動的に取得するには、「ソース」テキストボックスの横の Set to ▶ ボタンをクリックし、「データ」を選択します。

「ターゲット」テキストボックスにはジョブが呼び出される都度一意のファイル名を生成する式が含まれています。一意のファイル名を取得するには、FlowForce [instance-id](#) 式関数が呼び出されます。このため、JSON ファイル名は、「file35.json」のようになります（FlowForce ジョブインスタンスのID に対応し）数値は各ジョブの呼び出しにより異なります。フルパスを入力することもできますが、このサンプルで示されているとおり、「作業ディレクトリ」パスが設定されていることが必須になります。作業ディレクトリパスを設定すると、相対するファイル名が作業ディレクトリパスに対して解消されます。

ディレクトリ `C:\FlowForceExamples\Post.JSON`（または、パスが変更されている場合、上記に類似しているディレクトリ）が存在し、使用中のオペレーティングシステムユーザーアカウントが書き込み資格を有している必要があります。

「サービス」から、「このジョブを HTTP から使用できるようにする」チェックボックスを選択し、「PostJsonService」を入力、または Web サービスのために同様の名前を入力します。サービス名に注意してください。Web サービスを呼び出す必要があります。

## サービス

このジョブを HTTP を介して URL 上で利用可能にする `http://<FlowForce server>/service/`

「資格情報」から、既存の資格情報を選択、または、ローカルの資格情報を指定します（[資格情報](#)も参照してください）。これは、FlowForce Server が作動するオペレーティングシステム上のユーザーアカウントの資格情報である必要があります。

## 資格情報

資格情報によりジョブを実行:  既存の資格情報を選択:

ローカルの資格情報を定義: ユーザー名:

パスワード:

**メモ** この資格情報を FlowForce Server Web 管理インターフェイスへのアクセスに使用される資格情報特別してください。

[保存]をクリックします。新規の Web サービスをクライアントから呼び出す準備が整いました。

### Web サービスをブラウザから呼び出す方法

Web サービスを以下の方法でブラウザから呼び出すことができます:

- ホームに移動し、全てのアクティブなトリガーとサービスを表示するをクリックします。次の情報列内に表示されているジョブの URL をクリックします。
- <http://127.0.0.1:4646/service/PostJsonService> をブラウザのアドレスバーに入力します。この URL は FlowForce Server サービスがデフォルトのホストアドレスとポート名でリスニングする場合のみ機能します。[構成ページ](#)内で他のホストとポート設定を定義した場合、アドレスを必要に応じて変更してください。
- FlowForce Server の任意のホスト名 フィールドを [セットアップページ](#)から設定すると HTTP を介してこのジョブを使用可能にする チェックボックスの横の  ボタンをクリックして web サービスの呼び出しを直接ジョブ構成ページから実行することができます。それ以外の場合、このボタンは表示されません。

Web サービスにアクセスする際に資格情報をプロンプトされた場合、FlowForce Server にログインするために使用する同じ資格情報を提供してください。

FlowForce Server ユーザーに HTTP 認証のための資格情報を提供することはテスト目的のためです。生産のために新規の FlowForce ユーザーを作成することが奨励されており、ジョブが存在するコンテナ上でこのユーザーに **Service - Use** パーミッションを与え、対応するユーザーアカウントを使用して Web サービスにアクセスします。HTTP 認証を無効化するには Web サービスを公開し、**Service - Use** パーミッションをユーザー **Anonymous** に与えます。[パーミッションのしくみ](#)を参照してください。

ジョブはパラメータ敏感なストリームを期待するように構成されているためブラウザ内にパラメータ値を入力するようにプロンプトされます。「参照」をクリックして、POST リクエスト内に提供される JSON ファイルを選択します。

[送信]をクリックすると、FlowForce Server はジョブを処理し、ブラウザにレスポンスを出力します。

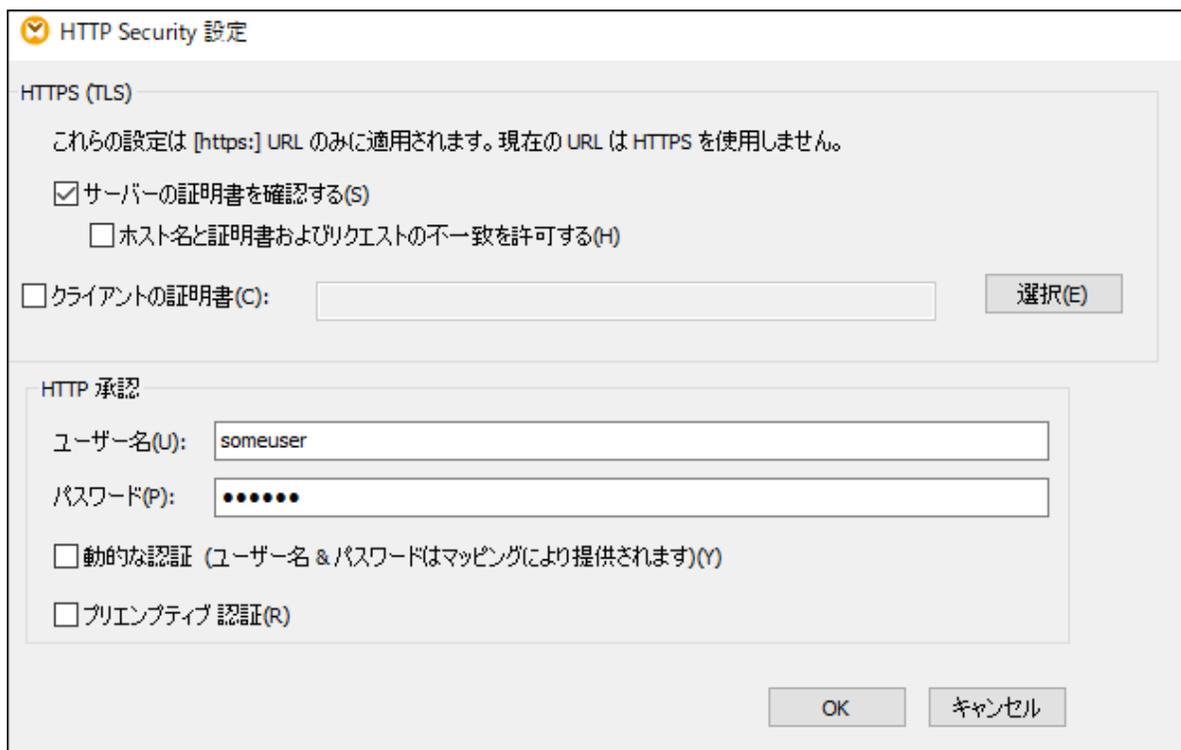
ジョブの実行に成功すると、ブラウザは「true」を表示し、JSON ファイルが作業ディレクトリ **C:\FlowForceExamples\PostJson** に保存されます。それ以外の場合、実行エラーが発生すると、詳細のためログを参照してください。 [ジョブログのビュー](#) を参照してください。

### MapForce から Web サービスを呼び出す方法

Web サービスを、MapForce Enterprise Edition などのブラウザ以外のクライアントから呼び出すことができます。

1. 新規のマッピングを作成するために「ファイル」メニューから「新規作成」をクリックします。
2. 「出力」メニューで「ビルトイン実行エンジン」をクリックします。
3. 「挿入」メニューから「Web サービス関数」をクリックします。Web サービス呼び出し設定ダイアログボックスが開かれます。
4. 「手動」をクリックし、**POST** をリクエストメソッドとして選択し、URL ボックス内で Web サービスの URL を入力します。ブラウザから Web サービスをテストするために使用された URL と同じです。

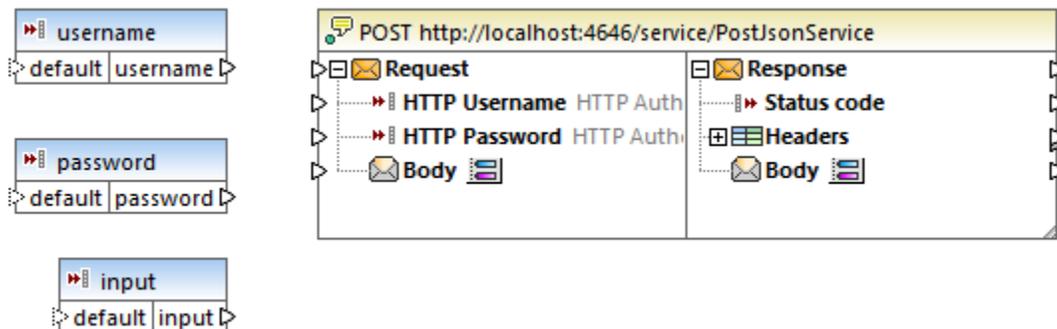
5. [HTTP セキュリティ設定]の横の編集 ボタンをクリックし、動的な認証 チェックボックスを選択します。これにより、マッピングの実行時資格情報をマッピングへの入力パラメータとして、インタラクティブに提供することができます。資格情報の使用 オプションに関する詳細については [マッピング関数内の資格情報](#) を参照してください。このダイアログボックス内にユーザー名とパスワードを直接入力することは下位互換性のためのみサポートされていますが、奨励はされていません。



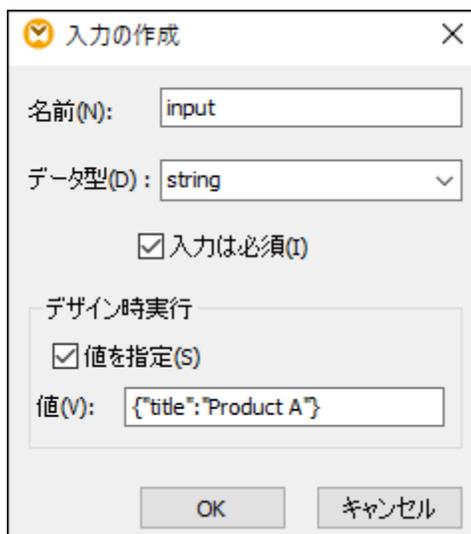
6. 「OK」をクリックして、ダイアログボックスを閉じます。マッピングは以下のようになります:



7. 「挿入 | 出力の挿入」メニューコマンドを選択し、マッピングに3個の入力パラメータを追加します。最初の2つはそれぞれユーザー名とパスワードを提供します。3番目はJSON データを適用します。

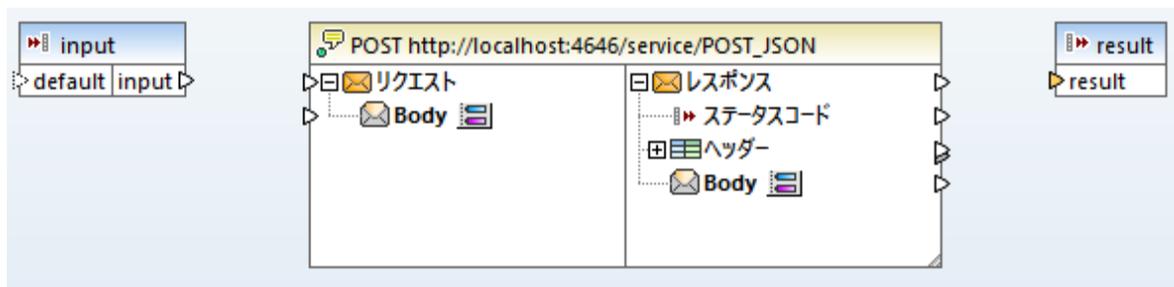


8. 上の入力コンポーネントのそれぞれをダブルクリックし、マッピングのプレビューのために使用されるデザイン時の実行値を入力します。最初の2つのパラメータのためにサービスにアクセスするために必要とされるユーザー名とパスワードを入力します。これはマッピングを実行するために必要でセキュリティ上の理由からマッピングファイルに保存することお奨励されません。JSON データを提供するパラメータのために、デザイン時にこのマッピングを実行するために、以下に示される通り、使用するためにサンプルJSON データを入力します:

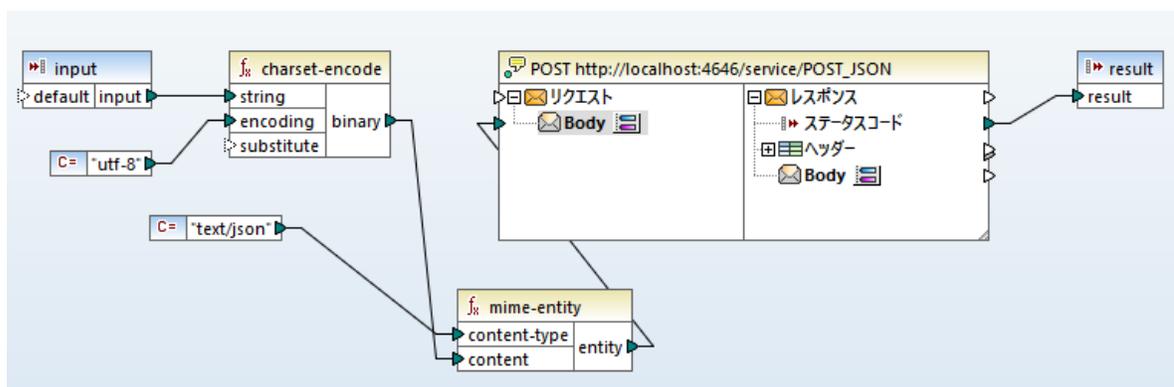


メモ ここで示されているサンプルJSON データは、デモのためにとても短いです。MapForce Server がマッピングを実行すると、JSON データを入力パラメータとして、実際のJSON ファイルからのマッピングを提供することができます。

9. 「挿入 | 出力の挿入」メニューコマンドを選択し、マッピングの出力を追加します。



10. `charset-encode` と `mime-entity` 関数をライブラウインドウからドラッグし下に示されるようにすべての接続を作成します。「入力 | 定数」メニューコマンドを選択し、2つの定数を追加する必要があります。



上記のマッピング内では、単純型コンポーネントの方法でJSON 入力のマッピングが提供されます。`charset-encode` と `mime-entity` 関数は、HTTP リクエストのボディを準備するMapForce ビルトイン関数です。Web サービスより返されるステータスコードはマッピングより返される結果にマップされます。

HTTP リクエストのボディを構成されていない方法で上記のように準備することは、POST リクエスト内にデータを送信する方法の一つです。JSON とXML 構造のために、「Web Service 呼び出しの設定」ダイアログボックス内のリクエストのJSON またはXML スキーマを入力することができます。この場合、サービスコンポーネントのボディは、リクエストのJSON/XML 構造をベースとし、マッピングの入力(コネクタ)を提供します。

出力 タブをクリックして、MapForce を使用してマッピングを実行することができます。エラーが発生すると、メッセージウィンドウが表示されます。デバッグするには、FlowForce Serverログを確認する必要がある場合があります(POST リクエストがサーバーに達していることが前提とされています)。それ以外の場合、実行に成功すると、以下が発生します:

1. HTTP ステータスコード "200" が出力 ペインに表示されます。
2. サーバ側で、提出されたJSON データはファイルに書き込まれ **C:\FlowForceExamples\PostJson** ディレクトリに保存されます。

また、MapForce Server を使用してマッピングを実行、または、FlowForce Server にデプロイすることができます。また、Web サービスにすることもできます。詳細に関しては、MapForce <https://www.altova.com/ja/documentation> ドキュメントを参照してください。

## 12.10 ジョブの結果をキャッシュする

この例では ( [キャッシュプロデューサー](#) )として参照されるジョブの結果をキャッシュし、( [キャッシュコンシューマー](#) )として参照される他のジョブ内で使用する方法が説明されています。両方のジョブは以下の振る舞いをする Web サービスとして公開されます:

- キャッシュプロデューサー Web サービスが呼び出されると、再帰的にディレクトリの内容をリストされ、キャッシュを作成または更新し、ブラウザ内に結果を出力します。
- キャッシュコンシューマー Web サービスが呼び出されると、キャッシュプロデューサーサービスにより作成されたキャッシュを読み取り、ブラウザ内に結果を出力します。

目的は、両方のジョブの実行にかかる時間を比較し、2番目のジョブの実行がキャッシュされたデータを使用することにより、1番目のジョブの実行より大幅に速いことを確認することです。

### 必要条件

- 必要とされるライセンス FlowForce Server
- FlowForce Web Server と FlowForce Server サービスが構成された [ネットワークアドレスとポート](#) でリスニングしていること
- [コンテナ](#) の一つのパーミッションを持つ FlowForce Server ユーザーアカウント (デフォルトでは /public コンテナはすべての認証済みのユーザーによりアクセスすることが可能です)。

メモ この例は、Windows パスとコマンドを使用していますが、パスとコマンドを必要に応じて変更すると、他のオペレーティングシステム上でテストすることができます。

### ジョブを構成する方法

1. 「構成」をクリックし /public/Examples コンテナに移動します。前のサンプルに従っている場合 public/Examples コンテナは既存であるべきです。それ以外の場合 作成 | コンテナの作成 コマンドを使用して作成します。
2. 「作成」をクリックし、「ジョブの作成」を選択します。
3. ジョブ名前ボックス内に **CachedResult** を入力します。
4. 実行ステップで、次の設定を持つ新しい実効ステップを追加します:

|              |   |
|--------------|---|
| 実行関数         | <code>/system/shell/commandline</code> 関数を参照します。  |
| コマンド         | 以下のシェルコマンドを入力します:<br><br><code>dir /s</code><br><br>Windows ではこのコマンドは <a href="#">作業ディレクトリ</a> の内容を再帰的にリストしてしま<br>す(次の設定参照)。                              |
| 作業ディレクトリ     | FlowForce Server が作動するマシン上のディレクトリへの値を設定します:<br><br><code>C:\</code><br><br>異なるディレクトリを使用する場合、ディレクトリの内容を再帰的にリストするため少な<br>くとも 20-30 秒かかる大きさのディレクトリを選択してください。 |
| ステップの結果を割り当て | 次のステップ内の実行ステップより返される値を参照する必要があり、名前が必要にな<br>ります。このサンプルのスキームのために <code>dir</code> をこのフィールドの値として入力します。   |

5. 実行ステップで、次の設定を持つ新しい実効ステップを追加します:

|      |  |
|------|--|
| 実行関数 | <code>/system/compute</code> 関数。   |
| 式    | 次のFlowForce Server 式を入力します:<br><br><code>stdout(dir)</code><br><br><code>stdout</code> 関数は前の実行ステップより返される生結果をデータのストリームに変換します ( <a href="#">ステップ結果関数</a> を参照してください)。 |

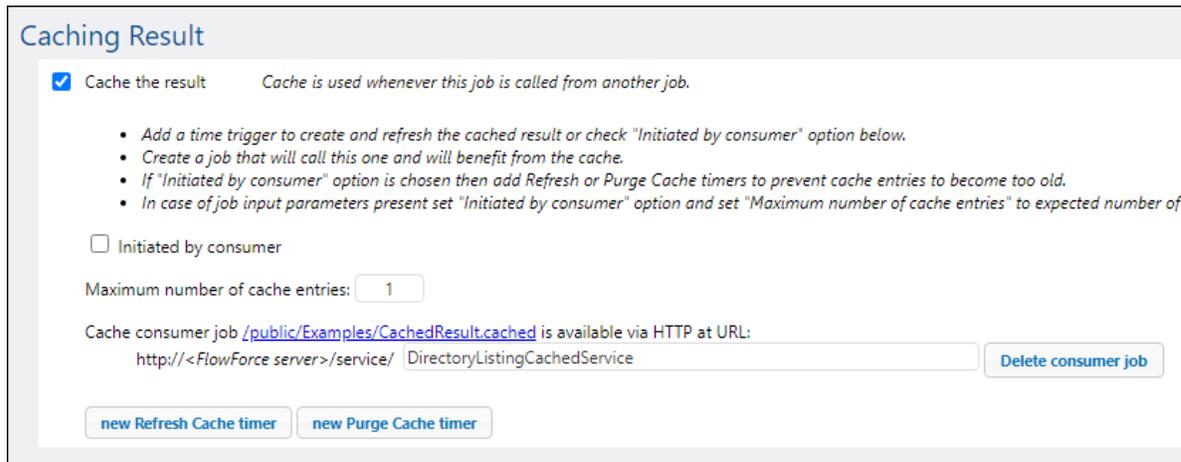
6. [実行結果] で戻り型を **stream** に設定します。既知かもしれませんが、ジョブの最後の実行ステップより返されたデータ型と同じ型に設定します。ジョブは以下のようになります:

The screenshot displays the 'Execution Steps' configuration in FlowForce Server. It shows two sequential steps:

- Step 1:** 'Execute function /system/shell/commandline'. Parameters include 'Command: dir /s', 'Abort on error: +', and 'Working directory: C:\'. The result is assigned to a variable named 'dir'.
- Step 2:** 'Execute function /system/compute'. The parameter is 'Expression: stdout(dir)'. The result is assigned to a variable named 'name'.

Below the steps are buttons for adding different types of steps: 'new Execution step', 'new Choose step', 'new For-each step', 'new Error/Success handling step', and 'new Postpone step'. At the bottom, the 'Execution Result' section is visible, with 'Declare return type as: stream' selected in a dropdown menu.

7. キャッシュの結果から **結果のキャッシュ** チェックボックスを選択します。
8. **新規のキャッシュコンシューマージョブを自動作成** チェックボックスを選択し `DirectoryListingCachedService` を Web サービスの名前として入力します。



9. サービスから「このジョブを HTTP を介して利用可能にする」チェックボックスをクリックして選択し、**DirectoryListingService** をサービスの名前として入力します。



10. 「資格情報」から、既存の資格情報の記録を選択、または、ローカルの資格情報を指定してください(以下を参照: [資格情報](#))。
11. 「保存」をクリックします。

### ジョブを実行する方法

この段階では、キャッシュプロデューサーとキャッシュコンシューマージョブの両方の構成が完了されました。非キャッシュサービス (**DirectoryListingService**) のパフォーマンスをブラウザ内でテストするには以下のアプローチのいずれかを使用してジョブを実行します。

- ホームに移動し、全てのアクティブなトリガーとサービスを表示するをクリックします。次に情報列内に表示されているジョブの URL をクリックします。
- <http://127.0.0.1:4646/service/DirectoryListingService> をブラウザのアドレスバーに入力します。この URL は FlowForce Server サービスがデフォルトのホストアドレスとポート名でアクセスする場合のみ機能します。[構成ページ](#)内で他のホストとポート設定を定義した場合、アドレスを必要に応じて変更してください。
- FlowForce Server の任意のホスト名 フィールドを [セットアップページ](#)から設定すると HTTP を介してこのジョブを使用可能にする チェックボックスの横の  ボタンをクリックして web サービスの呼び出しを直接ジョブ構成ページから実行することができます。それ以外の場合、このボタンは表示されません。

Web サービスにアクセスする際に資格情報をプロンプトされた場合、FlowForce Server にログインするために使用する同じ資格情報を提供してください。

FlowForce Server ユーザーに HTTP 認証のための資格情報を提供することはテスト 目的のためです。生産のため新規の FlowForce ユーザーを作成することが奨励されており、ジョブが存在するコンテナ上でこのユーザーに **Service - Use** パーミッションを与え、対応するユーザーアカウントを使用して Web サービスにアクセスします。HTTP 認証を無効化するには Web サービスを公開し、**Service - Use** パーミッションをユーザー **Anonymous** に与えます。[パーミッションのしくみ](#)を参照してください。

C:¥ディレクトリのコンテンツを再帰的にリストするように構成されているため、完了するまで数分かかる可能性があります。ジョブを完了するまでかかる時間を確認するために、ジョブログを参照してください。 [ジョブのログを確認する](#)を参照してください。

キャッシュ顧客サービス(`DirectoryListingCachedService`) のパフォーマンスをテストするには、enter <http://127.0.0.1:4646/service/DirectoryListingCachedService> をブラウザのアドレスバーに入力します (またはホスト名とポートが異なる方法で構成されている場合、同等のURLを入力します)。このサービスは、既存のディレクトリリストの代わりにキャッシュされたデータを使用するので、短い時間で完了することが期待されます。

## 12.11 StyleVision 変換からジョブを作成する

この例では StyleVision 変換から FlowForce Server ジョブを作成する方法が説明されています。ジョブは、以下の3つのステップから構成されています。

1. 最初のステップは、最初に StyleVision 変換を実行します。
2. 変換は、複数のストリームの配列を返すため、2番目のステップは、FlowForce Server 式を使用して変換により作成された複数のファイルにアクセスします。
3. 3番目のステップは、ファイルをアーカイブフォルダーにコピーします。

### 必要条件

- 必要とされるライセンス: StyleVision Enterprise または Professional エディション、StyleVision Server、FlowForce Server
- FlowForce Web Server と FlowForce Server サービスが構成された [ネットワークアドレスとポート](#) でリスンされていること
- [コンテナ](#) の一つで [パーミッション](#) を有する FlowForce Server ユーザーアカウントを有していること (デフォルトでは、この例で使用されている /public コンテナには、認証されたユーザーがアクセスすることができます)。
- このサンプルで作成されたジョブはファイルを一つのディレクトリから他のディレクトリにコピーします。FlowForce Server が作動するオペレーティングシステムでは、両方のディレクトリが存在している必要があり、これらのディレクトリ内にファイルを作成する権利を有していることを確認してください。このサンプルは以下のディレクトリを使用しています。
  - C:\FlowForceExamples\GenerateHtml — は全ての処理が行われ、相対的なパスが解決されるジョブの作業ディレクトリです。
  - C:\FlowForceExamples\Archive — はジョブにより生成された HTML ファイルがコピーされる保存先のディレクトリです。

### 使用されるデモファイル

StyleVision Power Stylesheet (.sps) ファイルは XML ファイルを処理し、HTML を含む複数のフォーマットで出力を生成します。これは **AutoCalc.sps** と呼ばれ StyleVision [Examples] サンプルプロジェクト **Examples > Basics > AutoCalc.sps** で使用することができます。StyleVision サンプルプロジェクトを StyleVision で開くには、**Examples** をプロジェクトメニューから開きます。

### StyleVision 変換を FlowForce Server にデプロイする方法

最初に StyleVision から FlowForce Server へデモ.sps ファイルをデプロイします。.sps ファイルのデプロイとは HTTP (または構成されている場合 HTTPS) を介して StyleVision が変換しようとするソースをオブジェクトに整理し、FlowForce Server にインストールすることを意味します。FlowForce Server に変換がデプロイされると、サーバージョブを作成することができます。

#### StyleVision 変換をデプロイする方法:

1. StyleVision 内で **AutoCalc.sps** ファイルを開きます。
2. 「ファイル」メニューから、「FlowForce (Server) にデプロイする」をクリックします。このオプションが無効化されているとデザインタブが選択されていることを確認してください。変換を PXF ファイルとして保存するようプロンプトされると、デフォルトの設定をそのまま、「OK」をクリックします。

Deploy Transformation

Enter the host name and port of a FlowForce Administration Interface to deploy the current transformation.

Server: 127.0.0.1 Port: 8082

User: root Use SSL

Password:

Login: <Default>

Deploy As

Path: /public/AutoCalc.transformation Browse

The path must start with a slash character.

Save design changes before deploying

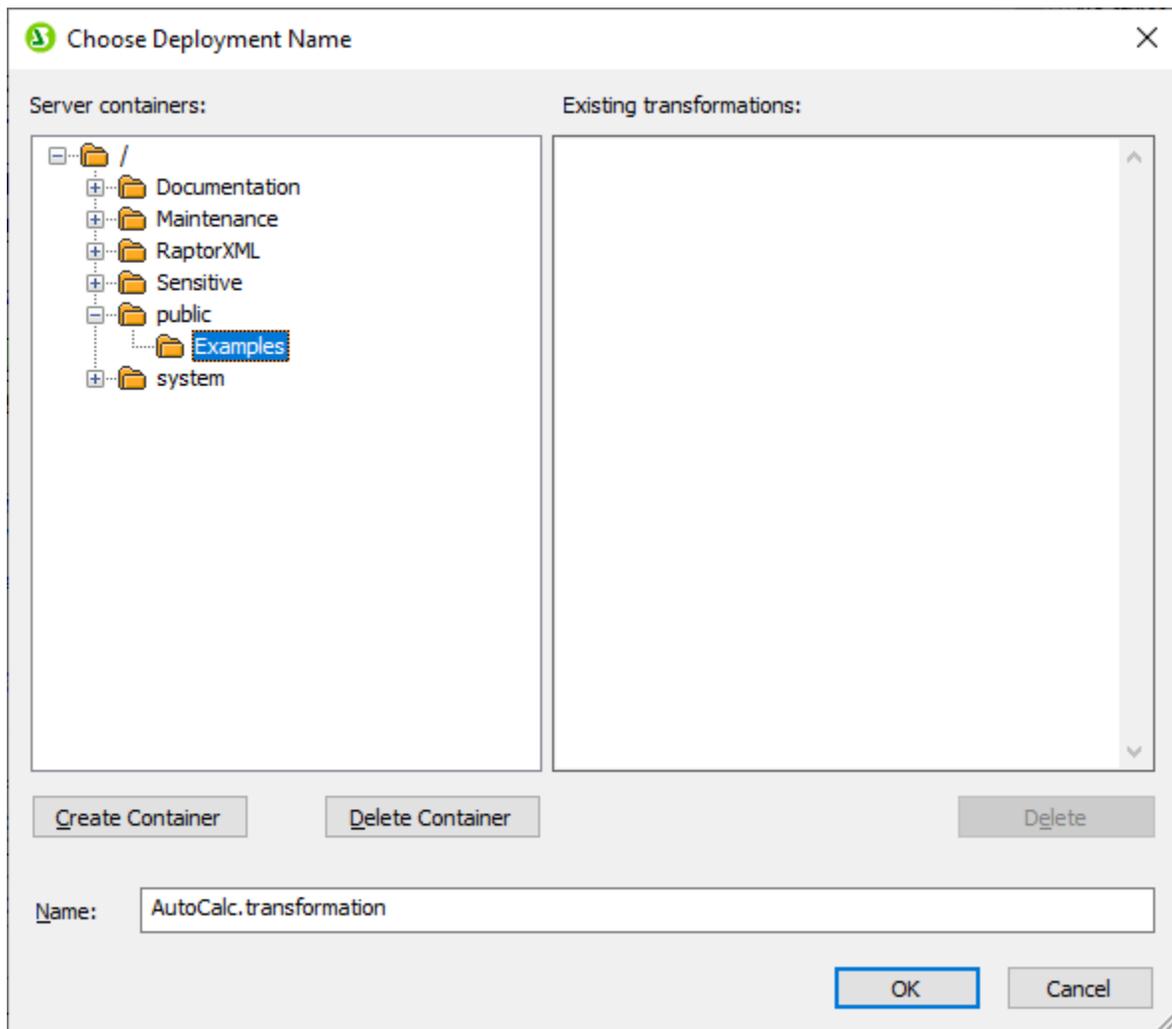
Open web browser to create new job

OK Cancel

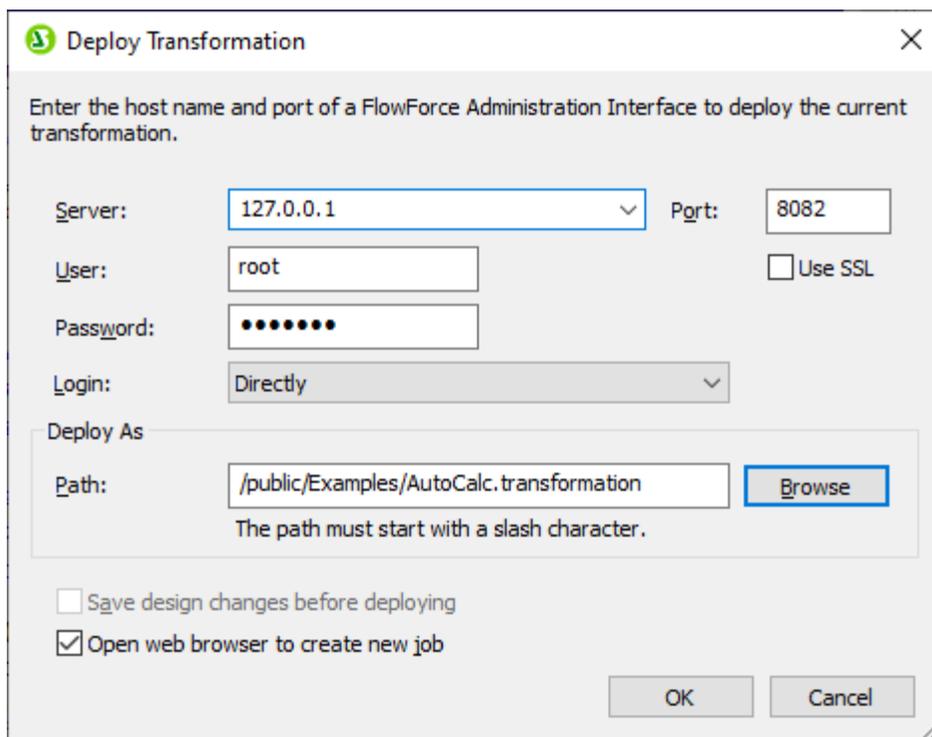
3. サーバーとポート テキストボックス内にWeb 管理インターフェイスのサーバー名とポートを入力します (例えば、FlowForce Web Server サービスがデフォルトのポートで同じマシン上でインストールされている場合は127.0.0.1と8082など)。異なるアドレスとポートが構成されている場合これらの値を変更してください。 [ネットワーク設定の定義](#)を参照してください。
4. User と Password テキストボックスにFlowForce Server ユーザー名とパスワードを入力します。
5. ディレクトリをログインドロップダウンリストから選択、または<Default> オプションをそのままします。

ディレクトリサービス統合が有効化されていると、ドメインユーザー名とパスワードを入力し、ドメイン名をログインドロップダウンリストから選択します。詳細に関しては、[ディレクトリサービス設定の変更](#)を参照してください。

6. パス テキストボックスは変換がデプロイされるデフォルトのパスを表示します。他のサンプルとの一貫性のために参照をクリックし、パスを/public/Examples/AutoCalc.transformationに変更します。前のサンプルに従っている場合/public/Examples コンテナは既存であるべきです。それ以外の場合、下のダイアログボックス内のコンテナの作成をクリックして作成します。



7. **[OK]** をクリックし「変換をデプロイする」ダイアログボックスの **新規のジョブ** を作成するために Web ブラウザーを開く チェックボックスを選択します。



**Deploy Transformation**

Enter the host name and port of a FlowForce Administration Interface to deploy the current transformation.

Server: 127.0.0.1 Port: 8082

User: root  Use SSL

Password: ●●●●●●

Login: Directly

Deploy As

Path: /public/Examples/AutoCalc.transformation

The path must start with a slash character.

Save design changes before deploying

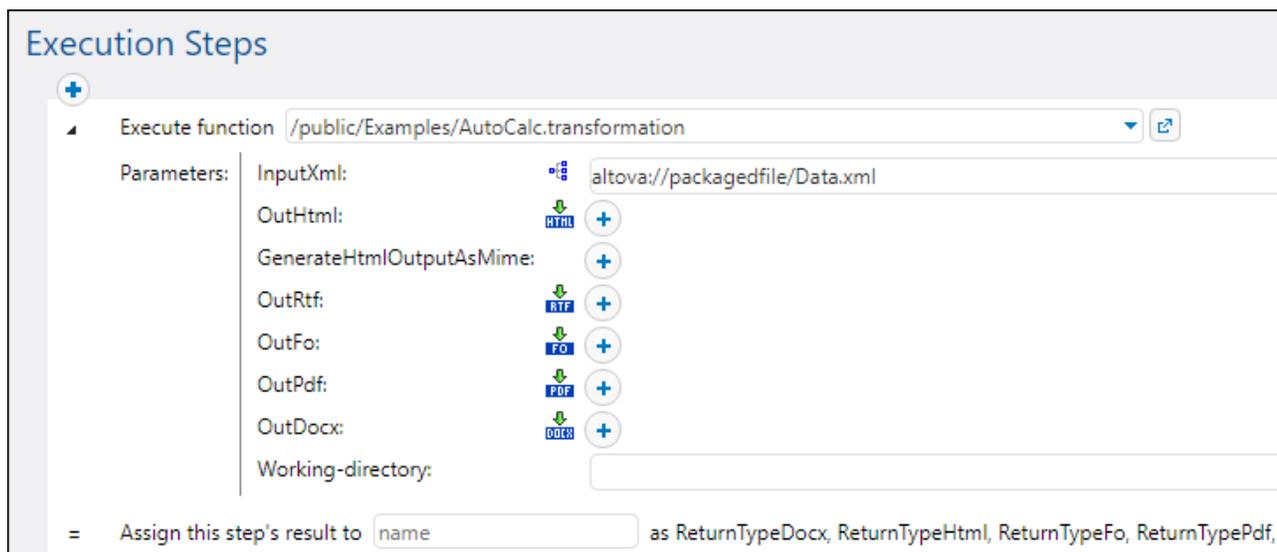
Open web browser to create new job

8. **[OK]** をクリックして変換をデプロイします。

デプロイが完了すると、FlowForce Server 管理インターフェイスは Web ブラウザーを開き、部分的に記入されたジョブのページが表示されます。変換関数自体は以前に指定済みのエンタープライズに保存されます。これでデプロイの過程は完了しました。

## ジョブの作成方法

上記の通り FlowForce Server に .sps ファイルをデプロイした後、ブラウザーは部分的に記入されたジョブページを表示します。入力済みのパラメータと共に最初の実行ステップが自動的に作成されます。



**Execution Steps**

Execute function /public/Examples/AutoCalc.transformation

Parameters:

|                           |  |
|---------------------------|--|
| InputXml:                 | altova://packagedfile/Data.xml   |
| OutHtml:                  | <input type="button" value="HTML"/> <input type="button" value="+"/> <input type="button" value="HTML"/> |
| GenerateHtmlOutputAsMime: | <input type="button" value="+"/> <input type="button" value="HTML"/>                                     |
| OutRtf:                   | <input type="button" value="RTF"/> <input type="button" value="+"/> <input type="button" value="RTF"/>   |
| OutFo:                    | <input type="button" value="FO"/> <input type="button" value="+"/> <input type="button" value="FO"/>     |
| OutPdf:                   | <input type="button" value="PDF"/> <input type="button" value="+"/> <input type="button" value="PDF"/>   |
| OutDocx:                  | <input type="button" value="DOCX"/> <input type="button" value="+"/> <input type="button" value="DOCX"/> |
| Working-directory:        | <input type="text"/>   |

= Assign this step's result to  as ReturnTypeDocx, ReturnTypeHtml, ReturnTypeFo, ReturnTypePdf,

ジョブを開数のページ (/public/Examples/AutoCalc.transformation) を開いてジョブを作成し、ジョブの作成をクリックします。

ジョブを構成する方法:

1. デフォルトのジョブ名を [AutoCalc.transformation.job] から詳細な名前に変更します。例えば [GenerateHtml] など。これは任意のステップですが、名前が同じコンテナ内の他のジョブにより使用されている場合必要なステップです。

2. デフォルトで作成された最初の実行ステップを以下のように入力します:

|              |   |
|--------------|---|
| 実行関数         | このフィールドには以前にデプロイされている StyleVision 変換関数を指しています。そのままご入力ください。  |
| パラメーター       | <p><b>InputXml</b> フィールドにはジョブ (<b>Data.xml</b>) にパッケージ済みの XML ファイルを含んでいます。この例のスキームのためパラメーター値そのままご入力することができます。入力と出力インスタンスの変更に関する詳細は <a href="#">ジョブとしてマッピングを実行し変換する</a> を参照してください。</p> <p><b>AutoCalc.html</b> を出力ファイルとして宣言するために <b>OutHtml</b> の横の  ボタンをクリックします。</p> <p><b>GenerateHtmlOutputAsMime</b> はこのサンプルでは設置されていません。このオプションは通常 FlowForce Server を使用して送信された電子メールメッセージの HTML 本文を準備するために必要とされます。詳細については、<a href="#">send-mime</a> 関数を参照してください。</p> <p>オプション <b>OutRtf</b>、<b>OutFo</b>、<b>OutPdf</b>、<b>OutDocx</b> は対応するフォーマットで (RTF、FO、PDF、Docx) 変換出力を宣言します。これらの出力は現在のサンプル内で使用されていないため、変更されないまま保持されます。</p> <p>作業ディレクトリボックスでは、FlowForce Server がジョブの出力を保存する必要のあるディレクトリへのパスを入力してください。この例は、ジョブの作業ディレクトリとして <b>C:\FlowForceExamples\GenerateHtml</b> を使用しています。</p> |
| ステップの結果を割り当て | 出力を入力します。これは明示的に最初の実行ステップより返された結果へという名前 [output] を与えます。この結果を後に参照する必要がなくなります。  |

3. 「新規実行ステップ」をクリックし以下のように入力します:

|      |                              |
|------|------------------------------|
| 実行関数 | /system/compute 関数。          |
| 式    | 次の FlowForce Server 式を入力します: |

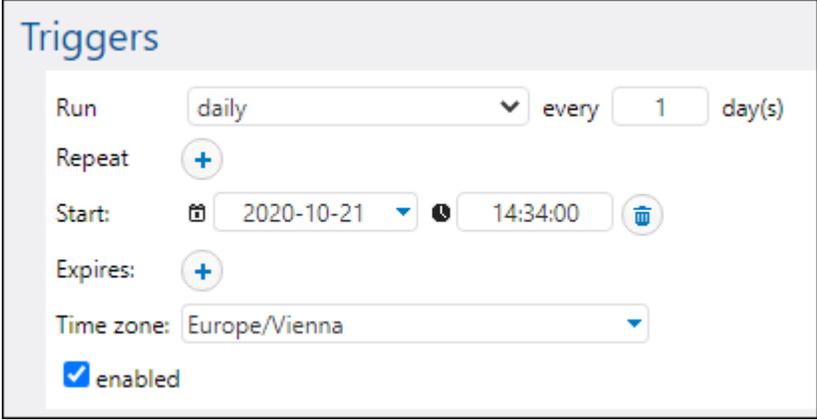
|              |   |
|--------------|---|
|              | <pre>as-file(nth(results(output), 0))</pre> <p>この式は、FlowForce Server に以下を行うように命令します:</p> <ol style="list-style-type: none"> <li>1. 宣言済みの <b>output</b> により返された配列を取得するためにこの関数の結果を呼び出します。</li> <li>2. 配列内の最初のアイテムを取得するために関数 <b>nth</b> に配列を渡します。配列インデックスはゼロベースであるため、関数 <b>nth</b> の番目の引数としてを使用します。</li> <li>3. ファイルとして宣言するためこの値を <b>as-file</b> 関数に渡します。</li> </ol> |
| ステップの結果を割り当て | <b>html_file</b> を入力します。これにより FlowForce Server にステップにより返された結果に <b>html_file</b> という名前が付けられるように命令します。この結果を後に参照する必要があります。   |

## 4. 「新規実行ステップ」をクリックし、ステップを以下のように構成します:

|          |  |
|----------|--|
| 実行関数     | <a href="#">/system/filesystem/copy</a> 関数を参照します。  |
| ソース      |  をクリックして <b>html_file</b> を選択します。 |
| ターゲット    | C:\FlowForceExamples\Archive\AutoCalc.html   |
| 上書き      | 「上書き」チェックボックスを選択します。   |
| 作業ディレクトリ | C:\FlowForceExamples\GenerateHtml  |

この段階では、ジョブページの実行ステップセクションは以下のようになります:





Triggers

Run: daily every 1 day(s)

Repeat: +

Start: 2020-10-21 14:34:00

Expires: +

Time zone: Europe/Vienna

enabled

9. 「資格情報」から、既存の資格情報の記録を選択、または、ローカルの資格情報を指定してください。詳細に関しては [資格情報](#) を参照してください。
10. [保存]をクリックします。

### ジョブを実行する方法

トリガー内で指定された日付で、FlowForce Server は、StyleVision 変換ジョブを実行します。ジョブの実行に成功すると **AutoCalc.html** ファイルを **C:\FlowForceExamples\Archive** ディレクトリ内で使用できるようになります。ジョブの実行が成功したかを確認するには [ジョブログ](#) を参照します。

## 12.12 RaptorXML を使用してドキュメントを検証する

この例では、XML スキーマファイルを検証するジョブの作成する方法が説明されています。条件付きエラー処理を使用せず、検証結果をカスタムログファイルまたはブラウザに書き込まないため、このサンプルはファイルを検証する最も簡単な方法を説明しています。検証結果は FlowForce Server ログ内でのみ使用することができます。更に複雑な検証ジョブのサンプルは [エラーログを使用して XML を検証する](#) を参照してください。

このサンプルで使用される検証ジョブは RaptorXML Server の `valany` 関数を呼び出します。`valany` 関数は型をベースのドキュメントを検証します。唯一の必須の引数として検証するためのファイルを取り、XML ファイル、XML スキーマ、DTD スキーマ、および他のファイル型を検証するために使用することができます。RaptorXML 関数に関しては RaptorXML ドキュメントを参照してください (<https://www.altova.com/ja/documentation>)。

### 必要条件

- 必要とされるライセンス FlowForce Server、RaptorXML (または RaptorXML+XBRL) Server。
- FlowForce Web Server と FlowForce Server サービスが構成された [ネットワークアドレスとポート](#) でホストされていること。
- [コンテナ](#) の一つでパーミッションを有する FlowForce Server ユーザーアカウントを有していること (デフォルトでは、この例で使用されている `/public` コンテナには、認証されたユーザーがアクセスすることができます)。

### 使用されるデモファイル

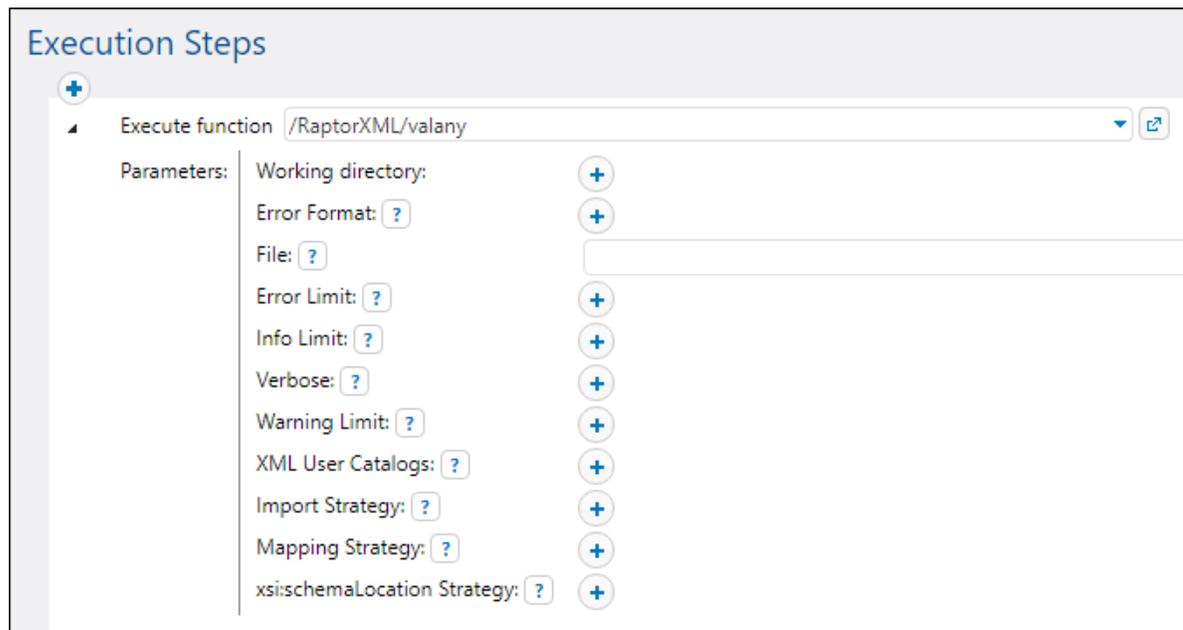
`address.xsd` ファイルを検証するサンプルジョブは RaptorXML Server インストールフォルダー内の次のパスにあります: `C:\Program Files\Altova\RaptorXMLServer2021\examples\address.xsd`。

RaptorXML Server を他の異なるフォルダーにインストールしている以外は 32 ビット RaptorXML Server を作動する 64 ビット Windows でのパスは `C:\Program Files (x86)\Altova\RaptorXMLServer2021\examples\address.xsd` です。

その他の XML スキーマファイルも使用することができます。

### ジョブの作成

1. FlowForce Server Web 管理インターフェイスログイン `/public/Examples` コンテナを開きます。前のサンプルに従っている場合 `public/Examples` コンテナは既存であるべきです。それ以外の場合 `作成 | コンテナの作成` コマンドを使用して作成します。
2. `[ジョブの作成]` をクリックします。次に名前を入力し、任意でジョブの詳細を任意で入力します。このサンプルは `[Validate Schema]` をジョブ名として使用しています。
3. `[新規の実行ステップ]` をクリックします。
4. `[実行関数]` の横で `RaptorXML/valany` 関数を参照します。必須の引数 `ファイル` は拡張されたフィールドとして表示されていることに注意してください。



メモ **valany** 関数は RaptorXML コンテナの下に直接存在し、特定の RaptorXML リリースに対応するコンテナ内にも存在します。例えば、“2021”。2つの違いに関する詳細は [ジェネリックとリリース固有の RaptorXML 関数](#) を参照してください。

5. ファイルテキストボックス内に検証するスキーマファイルへのパスを入力します。例えば、**C:\Program Files\Altova\RaptorXMLServer2021\examples\address.xsd** など。
6. トリガーから **新規のタイマー** をクリックして詳細の特定の時刻にジョブを実行するトリガーを作成します。詳細に関しては [タイマートリガー](#) を参照してください。
7. 「資格情報」から、既存の資格情報の記録を選択、または、ローカルの資格情報を指定してください。詳細に関しては [資格情報](#) を参照してください。
8. [保存] をクリックします。

### ジョブを実行する方法

トリガー内で指定された日付と時刻でジョブは作動します。ジョブの実行が成功したかを確認するには [ジョブログ](#) を参照します。具体的には [インスタンスログ](#) ページ内で以下のようなエントリが成功した検証を示しています：

```
file:///C:/Program%20Files/Altova/RaptorXMLServer2021/examples/address.xsd: runtime="0ms"
result="OK" cmd="valxsd"
```

ファイルが検証しない場合、ジョブの実行は失敗したと考えられ、(少なくとも1つのステップが失敗したため) エラーがログ内にレポートされます。この場合、ログされたエントリは検証エラーの詳細と共に `result="Fail"` を表示します。

## 12.13 エラーログのある XML を検証する

この例では、XML ファイルをスキーマに対して検証するジョブの作成する方法が説明されています。ジョブが何らかの理由で失敗した場合、詳細がログファイルに書き込まれます。検証のために FlowForce Server の管理下で作動する RaptorXML Server の `validxml-withxsd` 関数を使用します。エラーのログに関してですが、この例で説明されている技術は RaptorXML Server に依存するものではなく、他のジョブの型に適用することもできます。

**メモ** RaptorXML Server 関数は RaptorXML Server がインストールされた後に FlowForce Server 内で使用できるようになります。

このサンプルではジョブは Web サービスとして定義されており、ブラウザから URL にアクセスして、オンデマンドでトリガーすることができます。他の例で説明されているようにタイマー（またはファイルシステム）トリガージョブを追加することもできます。同じジョブトリガーと Web サービスの組み合わせを追加することもできます。このようにして、ジョブは、Web サービス呼び出された場合、トリガーのルールで定義されたのみではなく、オンデマンドで実行されることも可能です。

### 必要条件

- 必要とされるライセンス FlowForce Server、RaptorXML（または RaptorXML+XBRL）Server。
- FlowForce Web Server と FlowForce Server サービスが構成された [ネットワークアドレスポート](#) でリスニングされていること。
- 使用中の FlowForce Server ユーザーアカウントはの内の一つの [コンテナ](#)（デフォルトでは、認証済みのユーザーがアクセスすることのできるこのサンプルで使用されている `/public` コンテナ）へのパーミッションが与えられています。
- このサンプル内で作成されるジョブは作動する都度ログファイルを生成します。ですから、FlowForce Server が作動するオペレーティングシステムでディレクトリ内にファイルを作成する権利を持つ必要があります。（この例では、`C:\FlowForceExamples\ValidateXml` ディレクトリを使用します）。

### 使用されるデモファイル

この例で検証された XML ファイルは、RaptorXML Server インストールフォルダー内の次のパスにあります：`C:\Program Files\Altova\RaptorXMLServer2021\examples\NanonullOrg.xml`。

RaptorXML Server を他の異なるフォルダーにインストールしている以外は 32 ビット RaptorXML Server を作動する 64 ビット Windows でのパスは `C:\Program Files (x86)\Altova\RaptorXMLServer2021\examples\NanonullOrg.xml` です。

他の XML ファイルを検証のために使用することができます。

### ジョブの作成方法

- FlowForce Server Web 管理インターフェイスログイン `/public/Examples` コンテナを開きます。前のサンプルに従っている場合 `public/Examples` コンテナは既存であるべきです。それ以外の場合 `作成 | コンテナの作成` コマンドを使用して作成します。
- [ジョブの作成] をクリックします。次に名前を入力し、任意でジョブの詳細を任意で入力します。このサンプルは [ValidateSchema] をジョブ名として使用しています。

## Create job in /public/Examples

Job name:

Job description:

3. ジョブ入力パラメータで  ボタンを押して新規のパラメータ型 [ファイルとしての文字列] を作成します。例

### Job Input Parameters

+
+

Name:  Type: string as file Default: + Description:

4. 「実行ステップ」の下の  ボタンをクリックし、「新規選択ステップ」を選択します。

### Execution Steps

+
↑ ↓ 📄 🗑️

Execute with error/success handling - on error, retry  times

+
↑ ↓ 📄 🗑️

On error do

+
new error/success handler

5. 「エラー/成功処理を使用して実行」から  ボタンをクリックして、次の設定を持つ新規の実行ステップの追加を選択します:

|       |  |
|-------|--|
| 実行関数  | /RaptorXML/valxml-withxsd 関数を参照します。  |
| パラメータ | XML ファイル/パラメータの横から  をクリックして、既に宣言されている <b>inputFile</b> ジョブ入力パラメータを選択します。 |

6. 「エラー時」条件から  ボタンをクリックして、次の設定を持つ新規の実行ステップの追加を選択します:

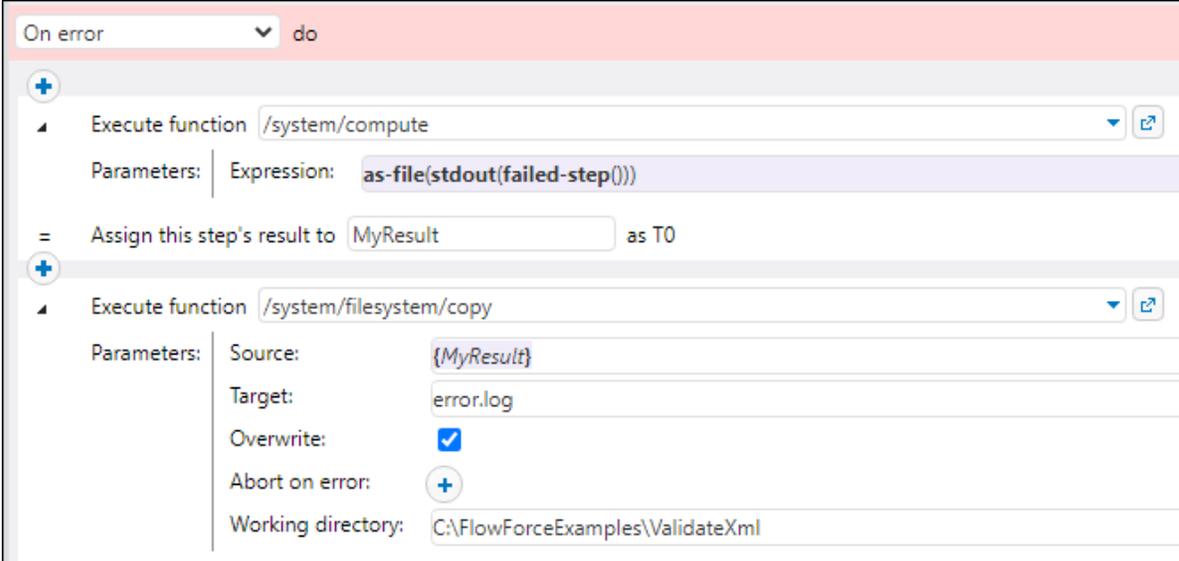
|              |   |
|--------------|---|
| 実行関数         | /system/compute 関数。   |
| パラメータ        | <p>式の値を以下に設定してください!</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <pre>as-file(stdout(failed-step()))</pre> </div> <p>中かこの間の部分はおつのFlowForce 式です。この式は出力を受け取り、ストリームに変換し、ディスク上のファイルにも一度書き込みます:</p> <ul style="list-style-type: none"> <li><b>failed-step</b> 関数は、失敗したステップの <b>result</b> を返します。 <b>exitcode</b>、<b>stderr</b>、または <b>error-message</b> 関数への引数として提供される更に使用しやすくなるこれは抽象的な FlowForce 型です。</li> <li><b>stdout</b> 関数はストリームとして <b>result</b> からの標準出力を取得します。</li> <li><b>as-file</b> 関数はストリームからファイルを作成します。パスは次のステップで指定されます。</li> </ul> |
| ステップの結果を割り当て | このジョブの結果を一意に識別することのできる値を入力します。例えば <b>MyResult</b> など。これを行うことにより、この値を変数として宣言し、次でステップで使用することができ   |

|  |     |
|--|-----|
|  | ます。 |
|--|-----|

7. 以下の設定を使用して  ボタンをクリックして前のステップの後に新規の実行ステップを追加します:

|        |   |
|--------|---|
| 実行関数   | <code>/system/filesystem/copy</code> 関数を参照します。  |
| パラメーター | <p>ソースパラメーターの横から  をクリックして、既に宣言されている <b>MyResult</b> 変数を選択します。</p> <p>「ターゲット」パラメーターの横に、ログが保存されるパスを入力します（この例では、パスは <code>C:\FlowForceExamples\ValidateXml\error.log</code> です）。</p> <p>上書きパラメーターの横のチェックボックスを選択します。ジョブが実行される都度、ログファイルは生成されます。これにより、ログファイルが既存であっても、ジョブが失敗することはありません。</p> <p><b>Working Directory</b> パラメーターを <code>C:\FlowForceExamples\ValidateXml</code> 設定します。</p> |

ジョブの[エラー時] ブランチは以下のようになります:



8. サービスから「このジョブを HTTP を介して URL 上で利用可能にする」チェックボックスをクリックして選択し、**ValidateXmlService** をサービスの名前として入力します。
9. 「資格情報」から、既存の資格情報の記録を選択、または、ローカルの資格情報を指定してください。詳細に関しては [資格情報を参照してください](#)。
10. [保存]をクリックします。

### ジョブを実行する方法

ジョブを実行するには、以下の内の1つを行ってください!

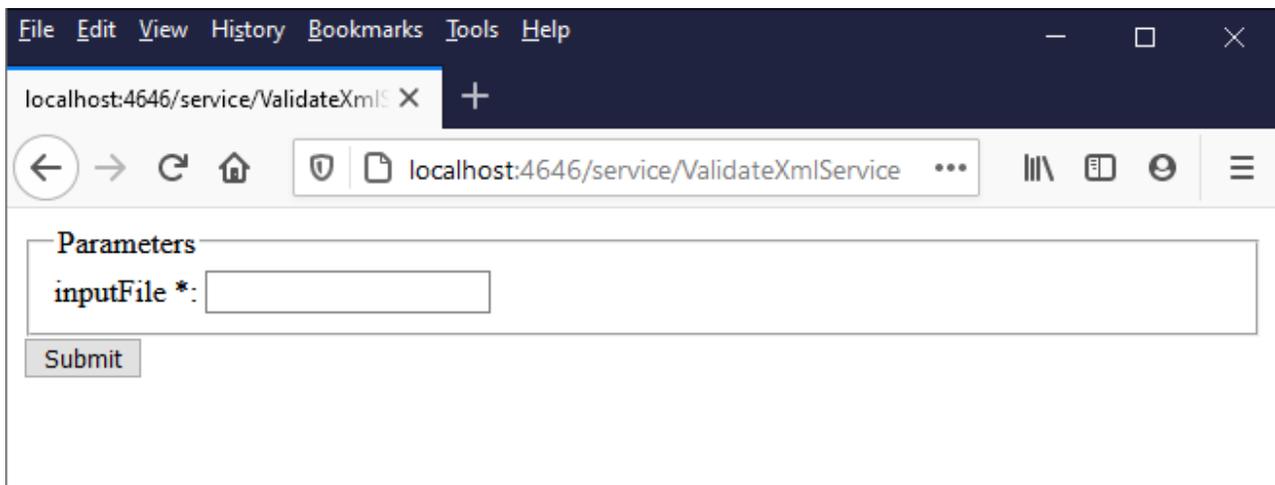
- ホームに移動し、全てのアクティブなトリガーとサービスを表示するをクリックします。次に情報列内に表示されているジョブの URL をクリックします。

- <http://127.0.0.1:4646/service/ValidateXmlService> をブラウザのアドレスバーに入力します。この URL は FlowForce Server サービスがデフォルトのホストアドレスとポート名でホストする場合のみ機能します。構成ページ内で他のホストとポート設定を定義した場合、アドレスを必要に応じて変更してください。
- FlowForce Server の任意のホスト名 フィールドを [セトアップページ](#) から設定すると HTTP を介してこのジョブを使用可能にする チェックボックスの横の  ボタンをクリックして web サービスの呼び出しを直接ジョブ構成ページから実行することができます。それ以外の場合、このボタンは表示されません。

Web サービスにアクセスする際に資格情報をプロンプトされた場合、FlowForce Server にログインするために使用する同じ資格情報を提供してください。

FlowForce Server ユーザーに HTTP 認証のための資格情報を提供することはテスト目的のためです。生産のために新規の FlowForce ユーザーを作成することが奨励されており、ジョブが存在するエンタープライズ上でこのユーザーに **Service - Use** パーミッションを与え、対応するユーザーアカウントを使用して Web サービスにアクセスします。HTTP 認証を無効化するには、Web サービスを公開し、**Service - Use** パーミッションをユーザー **Anonymous** に与えます。[パーミッションのしくみ](#)を参照してください。

ジョブがパラメータを入力して期待するように構成されているため、ブラウザは検証される XML ファイルへのパスを入力可能なフォームを表示します。



XML ファイルパスをテキストボックスに入力します (例えば `C:\Program Files\Altova\RaptorXMLServer2021\examples\NanonullOrg.xml`) して **Submit** をクリックします。

ジョブの実行に成功すると、(つまり、終了コード「0」を返します) ブラウザーは、ジョブの標準の出力を表示します。

```
file:///C:/Program%20Files/Altova/RaptorXMLServer/examples/NanonullOrg.xml:
runtime="16ms" result="OK"
```

(例えば、有効でないパス、検証エラー、などによる)ジョブが「0」以外の終了コードを返すと、ブラウザは「サービスの実行の失敗」メッセージを表示し、出力が `C:\FlowForceExamples\ValidateXml\error.log` ファイルに書き込まれます。ログファイルが生成されない場合、エラーを識別するために [ジョブログ](#) をチェックします。以下の可能性が考えられます: 例えば、`/system/filesystem/copy` 関数は、ターゲットパスへ書き込むためのパーミッションが不足しているために、失敗したなど。

## 12.14 RaptorXML を使用して XSLT を実行する

このサンプルは FlowForce Server 管理下で作動する RaptorXML Server (または RaptorXML+XBRL Server) を使用して XSLT 変換を行います。ジョブは RaptorXML Server の `xslt` 関数を呼び出します。FlowForce Server 構成ページからジョブを構成する際に `xslt` 関数にパラメータを提供する方法が示されています。

1. テキストボックス内にキー値ペア (パラメータ名と値) を入力する。
2. FlowForce Server 式を入力する。

これらの方法の詳細は以下で説明されています。

### 必要条件

- 必要とされるライセンス FlowForce Server、RaptorXML (または RaptorXML+XBRL) Server。
- FlowForce Web Server と FlowForce Server サービスが構成された [ネットワークアドレスとポート](#) でリスニングされていること。
- 使用中の FlowForce Server ユーザーアカウントは内の一つの [コンテナ](#) (デフォルトでは 認証済みのユーザーがアクセスすることのできるこのサンプルで使用されている /public コンテナ) へのパーミッションが与えられています。
- このサンプルで作成されたジョブは入力 XML ファイルを処理する XSLT スタイルシートを実行します。両方のファイルはオペレーティングシステム上のディレクトリに存在する必要があり、このディレクトリ内のファイルを読み込む、またはファイルに書き込む権利が必要です。このサンプルは `C:\FlowForceExamples\RunXslt` を使用しています。

### デモファイル

下で説明されるジョブは `books.xml` と呼ばれるファイルと 2 つの必須パラメータ [year] と [genre] を入力として取る `transformation.xslt` と呼ばれる XSLT スタイルシートを実行します。ファイルの実際のコンテンツは下のロードリスティングで表示されています。ジョブ内でこれらのファイルを使用する場合は、示されているファイル名と共に両方のロードリスティングを `C:\FlowForceExamples\RunXslt` directory に保存します。

XSLT スタイルシートは以下のようになります。

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:fn="http://www.w3.org/2005/xpath-
functions" exclude-result-prefixes="xs fn">
  <xsl:output method="xml" encoding="UTF-8" byte-order-mark="no" indent="yes"/>
  <xsl:param name="year" as="xs:string" required="yes"/>
  <xsl:param name="genre" as="xs:string" required="yes"/>
  <xsl:template match="/">
    <library>
      <xsl:attribute name="xsi:noNamespaceSchemaLocation"
namespace="http://www.w3.org/2001/XMLSchema-instance" select="'library.xsd'"/>
      <last_updated>
        <xsl:sequence select="xs:string(fn:current-dateTime())"/>
      </last_updated>
      <xsl:for-each select="(./books/book)[(fn:string(year) > $year)]">
        <publication>
          <xsl:for-each select="@id">
            <id>
              <xsl:sequence select="fn:string(.)"/>
            </id>
          </xsl:for-each>
          <author>
```

```
        <xsl:sequence select="fn:string(author)"/>
    </author>
    <title>
        <xsl:sequence select="fn:string(title)"/>
    </title>
    <genre>
        <xsl:sequence select="$genre"/>
    </genre>
    <publish_year>
        <xsl:sequence select="xs:string(xs:integer(fn:string(year)))/>
    </publish_year>
</publication>
</xsl:for-each>
</library>
</xsl:template>
</xsl:stylesheet>
```

*transformation.xslt*

入力 XML ファイルは以下のようになります:

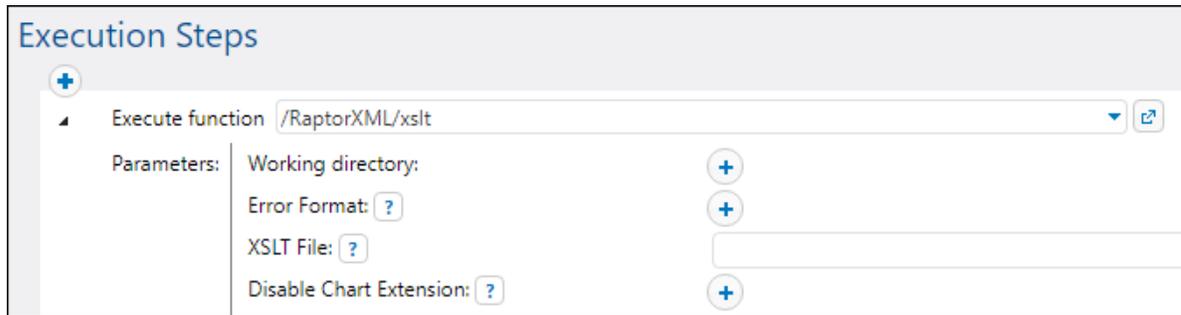
```
<?xml version="1.0" encoding="UTF-8"?>
<books xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="books.xsd">
  <book id="1">
    <author>Mark Twain</author>
    <title>The Adventures of Tom Sawyer</title>
    <category>Fiction</category>
    <year>1876</year>
  </book>
  <book id="2">
    <author>Franz Kafka</author>
    <title>The Metamorphosis</title>
    <category>Fiction</category>
    <year>1912</year>
  </book>
  <book id="3">
    <author>Herman Melville</author>
    <title>Moby Dick</title>
    <category>Fiction</category>
    <year>1851</year>
  </book>
  <book id="4">
    <author>Miguel de Cervantes</author>
    <title>Don Quixote</title>
    <category>Fiction</category>
    <year>1605</year>
  </book>
</books>
```

*books.xml*

## ジョブの作成方法

XSLT 変換などの FlowForce Server ジョブを以下のように作成することができます:

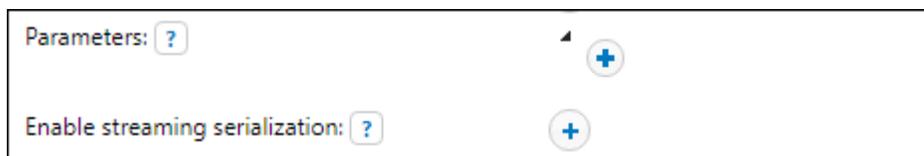
1. FlowForce Server Web 管理インターフェイスにログインし **/public/Examples** コンテナを開きます。前のサンプルに従っている場合 **public/Examples** コンテナは既存であるべきです。それ以外の場合 **作成 | コンテナの作成** コマンドを使用して作成します。
2. **[ジョブの作成]** をクリックします。次に名前を入力し、任意でジョブの詳細を任意で入力します。このサンプルは **[RunXslt]** をジョブ名として使用します。
3. **[新規の実行ステップ]** をクリックします。
4. **[実行関数]** の横で **/RaptorXML/xslt** 関数を参照します。



**メモ** **xslt** 関数は RaptorXML コンテナの下に直接存在し、特定の RaptorXML リリースに対応するコンテナ内にも存在します。例えば、"2021"。2つの違いに関する詳細は [ジェネリックリリース固有の RaptorXML 関数を参照してください](#)。

5. 作業ディレクトリ / パラメータを **C:\FlowForceExamples\RunXslt** に設定します。
6. **XSLT File** / パラメータを **transformation.xslt** に設定します。このファイルは作業ディレクトリ内に存在する必要があります。
7. **XSLT Input** / パラメータを **books.xml** に設定します。このファイルは作業ディレクトリ内に存在する必要があります。
8. パラメータ / パラメータを以下のように設定します。

a. パラメータの横の **+** ボタンをクリックします。これにより、各パラメータ名と値をそれぞれ追加できるページ内のサブセクションが展開されます。



b. 追加する必要があるそれぞれの新規パラメータのために **+** ボタンをクリックします。このサンプル内の XSLT を実行するために、以下のようにパラメータを入力する必要があります。



XSLT / パラメータはジョブキー値ペアとして提供されます。パラメータ名と値は個別のボックス内に入力されることに注意してください。また、パラメータ値は引用符で囲まれています。

9. サービスから「このジョブを HTTP を介して URL 上で利用可能にする」チェックボックスをクリックして選択し **RunXsltService** をサービスの名前として入力します。
10. 「資格情報」から、既存の資格情報の記録を選択、または、ローカルの資格情報を指定してください。詳細に関しては [資格情報を参照してください](#)。

11. [保存]をクリックします。

これでジョブの構成の部分を完了します。

### XSLT パラメーターを式として与える方法

上記のジョブ構成では `xslt` 関数にテキストボックスを使用してパラメーターが提供されました。これを行うための番目の方法があります。FlowForce Server 式パラメーターテキストボックスに入力します。例:

|                                   |  |
|-----------------------------------|--|
| Named Template Entry Point: ?     | +  |
| Primary Output: ?                 | +  |
| Parameters: ?                     | <code>list(("year","1900"),("genre","Fiction"))</code> |
| Enable streaming serialization: ? | +  |

この番目のアプローチを使用してパラメーターの横の **Set to** をクリックし **<Expression>** をクリックします。解析エラーを回避するために式の型を慎重に入力してください。式は `list` 式関数を呼び出し、2つのキー値ペアを作成します。キー値ペアでは、キーはパラメーター名で、値はパラメーターの値です。重要な点はパラメーター値が単一の引用符で囲まれている点です。

テキストレイアウトに切り替えるには **Set to** /パラメーターの横をクリックして **<Value>** をクリックします。

### ジョブを実行する方法

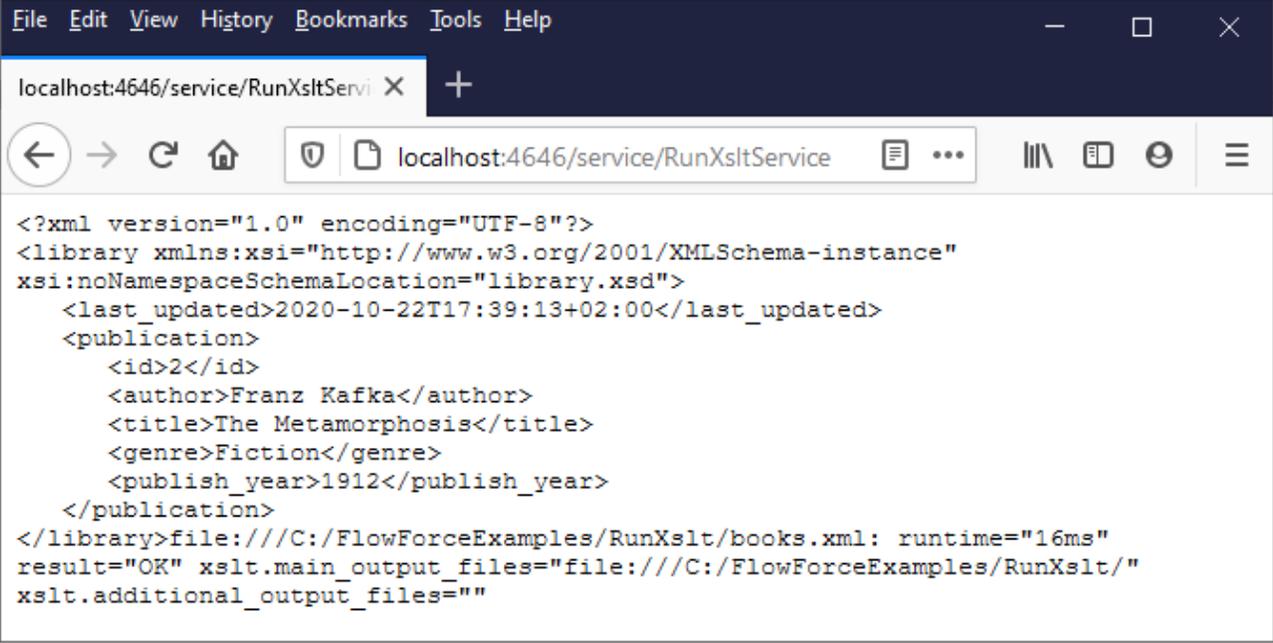
ジョブを実行するには、以下の内の1つを行ってください！

- ホームに移動し、全てのアクティブなトリガーとサービスを表示するをクリックします。次に情報列内に表示されているジョブの URL をクリックします。
- <http://127.0.0.1:4646/service/RunXsltService> をブラウザのアドレスバーに入力します。この URL は FlowForce Server サービスがデフォルトのホストアドレスとポート名でリスニングする場合のみ機能します。構成ページ内で他のホストとポート設定を定義した場合、アドレスを必要に応じて変更してください。
- FlowForce Server の任意のホスト名 フィールドを [セットアップページ](#) から設定すると HTTP を介してこのジョブを使用可能にする チェックボックスの横の **▶** ボタンをクリックして web サービスの呼び出しを直接ジョブ構成ページから実行することができます。それ以外の場合、このボタンは表示されません。

Web サービスにアクセスする際に資格情報をプロンプトされた場合、FlowForce Server にログインするために使用する同じ資格情報を提供してください。

FlowForce Server ユーザーに HTTP 認証のための資格情報を提供することはテスト目的のためです。生産のために新規の FlowForce ユーザーを作成することが奨励されており、ジョブが存在するエンタープライズ上でこのユーザーに **Service - Use** パーミッションを与え、対応するユーザーアカウントを使用して Web サービスにアクセスします。HTTP 認証を無効化するには、Web サービスを公開し、**Service - Use** パーミッションをユーザー **Anonymous** に与えます。[パーミッションのしくみ](#)を参照してください。

ジョブの実行に成功すると、ブラウザはジョブの出力を表示します。例:



```
<?xml version="1.0" encoding="UTF-8"?>
<library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="library.xsd">
  <last_updated>2020-10-22T17:39:13+02:00</last_updated>
  <publication>
    <id>2</id>
    <author>Franz Kafka</author>
    <title>The Metamorphosis</title>
    <genre>Fiction</genre>
    <publish_year>1912</publish_year>
  </publication>
</library>file:///C:/FlowForceExamples/RunXslt/books.xml: runtime="16ms"
result="OK" xslt.main_output_files="file:///C:/FlowForceExamples/RunXslt/"
xslt.additional_output_files=""
```

ジョブの実行に失敗するとブラウザは「サーバー実行の失敗」メッセージを表示します。この場合、FlowForce Server 内の[ジョブのログ](#)をチェックして、エラーを確認してください。ジョブのログの確認を参照してください。

## 12.15 複数の PDF を複数の XML から生成する

このサンプルでは、複数のXML ファイルを入力として取り、出力 複数のPDF ファイルを返す FlowForce Server ジョブの作成の方法について説明されています。FlowForce Server ジョブは、MapForce Server (複数のソースXML ファイルからXML 出力を生成するための) とStyleVision Server (XML 出力をPDF に変換するための) を呼び出します。

このサンプルは、MapForce マッピングとStyleVision 変換に関する基本的な理解を必要とします。StyleVision とMapForce の初心者の場合は、MapForce とStyleVision ドキュメントのチュートリアルチャプターを最初に読むことが奨励されています:

- [クイックスタートチュートリアル\(MapForce\)](#)
- [クイックスタートチュートリアル\(StyleVision\)](#)

### 必要条件

- 必要とされるライセンス
  - **MapForce Enterprise** または **Professional** エディションのインストール。このツールによりスキーマから他のスキーマにXML ファイルを変換するマッピング変換 (.mfd ファイル) をデザインすることができます。
  - **MapForce Server** または **MapForce Server Advanced Edition**。このツールによりサーバー上でマッピングをジョブとして実行することができます。
  - **StyleVision Enterprise** または **Professional** エディション。このツールによりXML ファイルからPDF ファイルに変換するスタイルシート (.sps ファイル) をデザインすることができます。
  - **StyleVision Server**。このツールによりサーバー上で変換をジョブとして実行することができます。
  - **FlowForce Server**。このツールは上記の変換を計画通り、またはオンデマンドジョブとして実行し、入力を必要な場合は変更する方法を提供します。
- FlowForce Web Server とFlowForce Server サービスが構成された [ネットワークアドレスとポート](#) でホストされていること。
- 使用中のFlowForce Server ユーザーアカウントはの内の一つの [コンテナ](#) (デフォルトでは、認証済みのユーザーがアクセスすることのできるこのサンプルで使用されている /public コンテナ) へのパーミッションが与えられています。
- このサンプル内で作成されたジョブはディスクに複数のファイルを生成します。ですから、FlowForce Server が実行されている場所でディレクトリ内にファイルを作成する権利が必要になります。このサンプルはディレクトリ C:\FlowForceExamples\GeneratePdfs を使用しています。

### 使用されるデモファイル

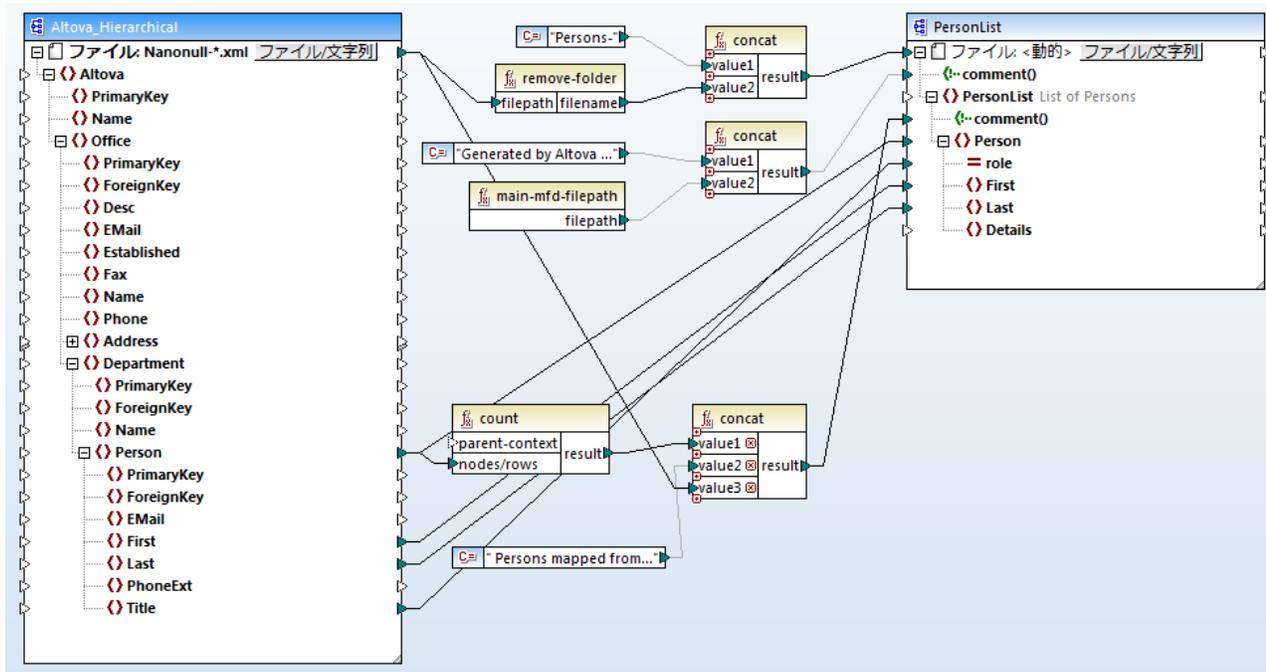
このサンプルは、次のパスで見つけることのできるサンプルファイルを使用しています:

<Documents>\Altova\MapForce2021\MapForceExamples。

- **MultipleInputToMultipleOutputFiles.mfd** (MapForce マッピングファイル)
- **PersonListWithGrouping.sps** (StyleVision 変換 ファイル)
- **Nanonull-Branch.xml**、**Nanonull-HQ.xml** (入力 XML ファイル)

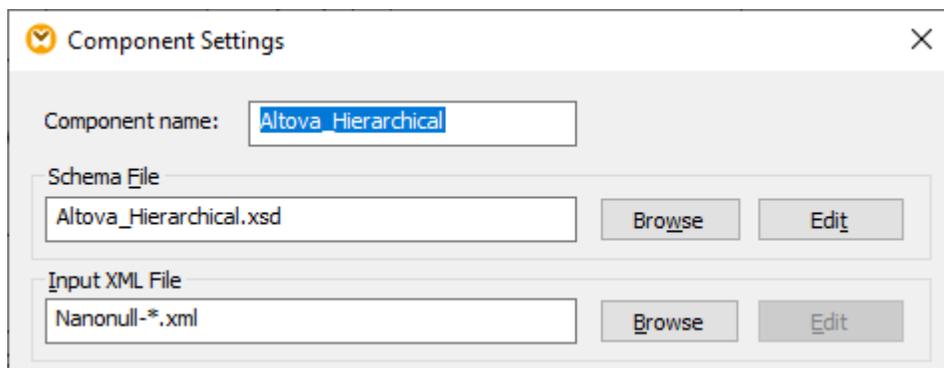
### MapForce マッピングのしくみ

下に表示されるとおり、マッピングは、ソースコンポーネント (**Altova\_Hierarchical**)、ターゲットコンポーネント (**PersonList**)、出力に書き込まれた他の文字列をビルドするために使用されると多種のMapForce ビルトイン関数から構成されています。



MultipleInputToMultipleOutputFiles.mfd

マッピングはディレクトリ <Documents>\Altova\MapForce2021\MapForceExamples から「Nanonull-」という名前のXMLファイルを入力として取ります。この項目をソースMapForce コンポーネント設定内で定義することができます (MapForce 内では、下に表示されるように **Altova\_Hierarchical** コンポーネントのヘッダーを右クリックし、コンテキストメニューから「Properties」を選択します)。入力ファイルは、アスタリスクがワイルドカードの役割を果たす、**Nanonull-\*.xml** に設定されています。文字通り、入力は「Nanonull-」から開始する、.xml 拡張子を持つファイルです。



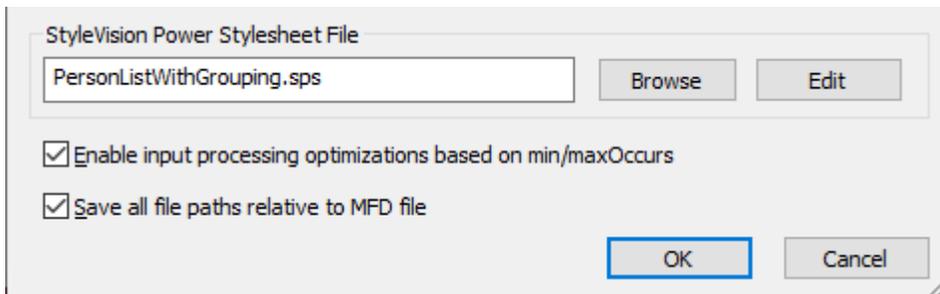
ターゲットコンポーネント、**PersonList** は、ソースXMLファイルのファイル名をベースとしてファイル名を動的に生成するように構成されています。これは、コンポーネントの上の「File/String」ボタンを右クリックし、「マッピングにより与えられる動的ファイル名を使用する」メニューオプションを選択することによって定義することができます。「ファイル<dynamic>」ノードへの接続は、新規ファイルはソース内の各値のため作成されます。**remove-folder** 関数は、ソースパスからフォルダ名無しのファイル名のみを取得することを意味します。これは、**Persons-<Source filename>** などの文字列をビルドするトップ **concat** 関数に対する値として使われます。

2番目の **concat** 関数は、マッピングパスへの完全なパス後に続く、**Generated by Altova...** などの文字列をビルドします。結果はターゲット XML ファイル内のコメントとして書き込まれます。

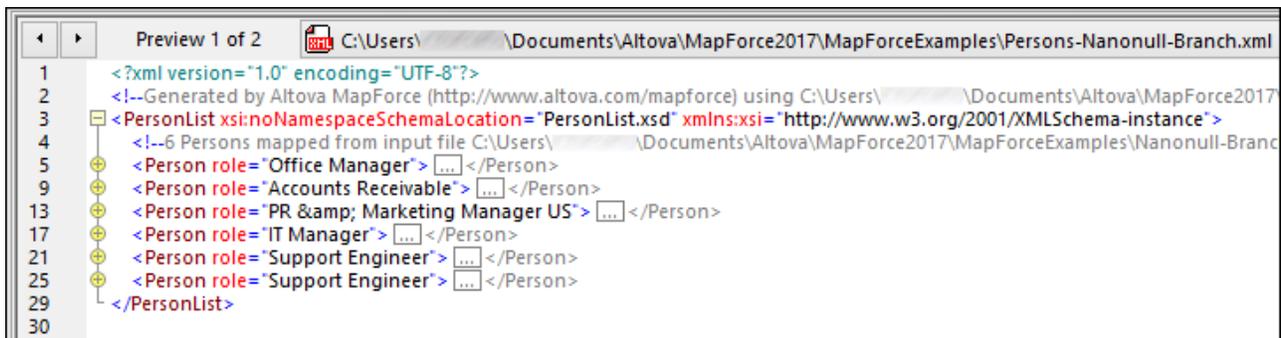
3番目の **concat** 関数は文字列をビルドするために、ソースからマップされる個人のレコードの個数を示す **count** 関数の出力を使用します。結果はターゲット XML ファイル内のコメントとして書き込まれます。

ターゲット **Person** ノードへの接続は、ターゲットへのソースから個人データをコピーします。マップされる必要のある「Person」の各子要素のための個別の接続が存在します。

更に、ターゲットコンポーネントで生成された出力を各 XML に生成されたファイルのために PDF に変換するように構成されています。ターゲットコンポーネントのヘッダーを右クリックし、**Properties** を選択し、StyleVision .sps スタイルシートへの相対パスを指定する StyleVision Power スタイルシート ファイルテキストボックスを確認してください。後者は、XML から PDF への実際の変換を実行します(下で更に詳しく説明されています)。



このマッピングの出力を直接 MapForce 内でプレビューするには、マッピングペインの下出力タブをクリックします。StyleVision 変換の PDF 結果をプレビューするには、PDF タブをクリックします。出力ペイン内で複数の XML (または PDF) が出力されます。以下を参照してください。



この段階では (デフォルトでは、MapForce は一時ファイルを作成するため) ディスクに2つの出力 XML ファイルの1つを保存することが奨励されます。StyleVision 内で StyleVision Power スタイルシートを開きテストする場合ファイルは、サンプル(作業 XML)としての役割を果たします(次のセクションを参照してください)。出力ファイルを保存するには、「出力」タブをクリックして、「出力」メニューから、「出力ファイルの保存」をクリックします。

## StyleVision 変換について

StyleVision を作動し、**PersonListWithGrouping.sps** 変換 ファイルを開きます。このファイルは、MapForce マッピングで説明されている同じディレクトリと同じファイルで、ターゲット MapForce コンポーネントにより参照されています。

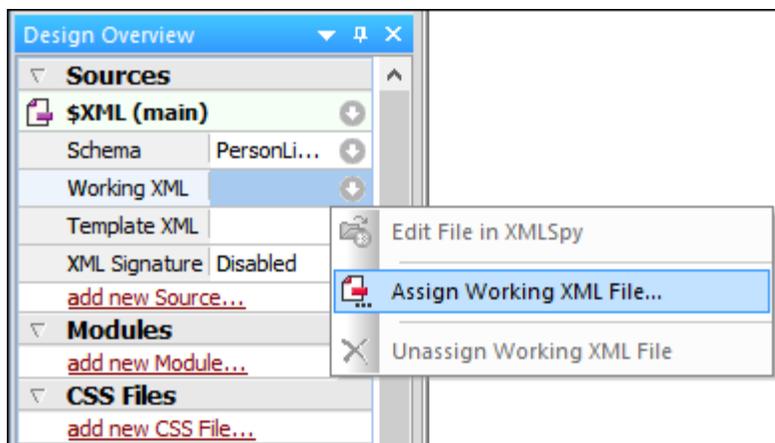


PersonListWithGrouping.sps

上記のStyleVision .sps スタイルシートは、単一のXML をソースとして表示しており、PDF を作成します。PDF ドキュメントは、見出し(「h2」)、紹介パラグラフ、動的に作成されたテーブル、終わりのパラグラフにより構成されています。見出しと紹介パラグラフは、静的テキストを含みますが、終わりのパラグラフは、ラッピングタブで指定されているとおり、ソースXML ファイルのノードから作成されています。

StyleVision でこの変換を直接プレビューするには、下のステップに従います:

1. 「デザイン概要」ペイン内で、「作業 XML」 をクリックします。



2. 「作業 XML ファイルの割り当て」を選択して、MapForce から保存された XML 出力ファイルを参照します(前のセクションを参照してください)。
3. PDF タブをクリックします。

.sps スタイルシートは、実際の名前、またはソース XML ファイルのオリジンに対して未知であり、(指定された XML スキーマに準拠する限り)入力として与えられた XML ファイルを処理し、PDF を作成します。このスタイルシートを自動化し、複数の PDF ファイルを生成するよう構成するには、下に示されるように、FlowForce Server にデプロイされる必要があります。

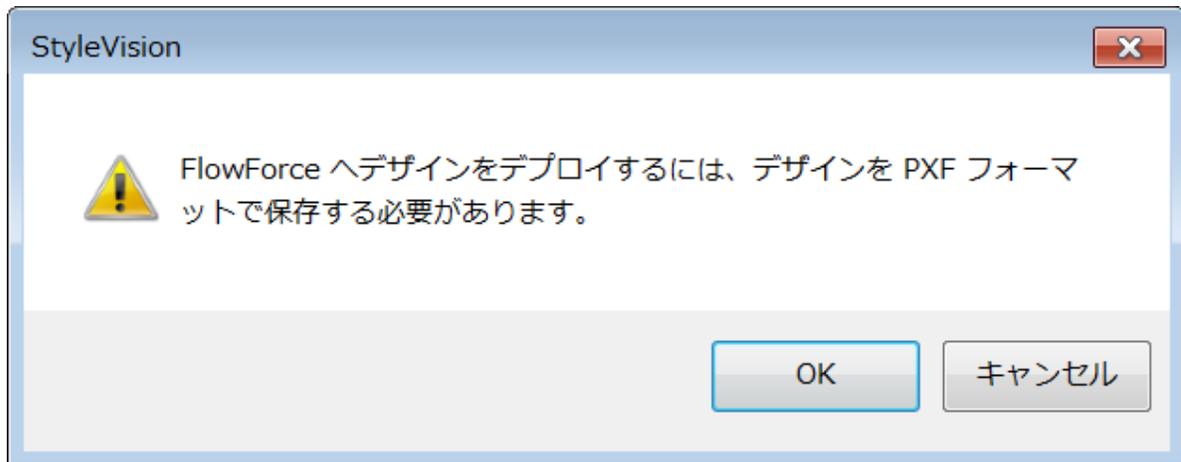
### FlowForce Server にファイルをデプロイする

このサンプル内で使用される MapForce マッピングと StyleVision 変換の目的について説明されましたが、MapForce マッピングと StyleVision スタイルシートのデザインに関する詳細に関しては、これらの製品のドキュメントを参照してください (<https://www.altova.com/documentation.html>)。

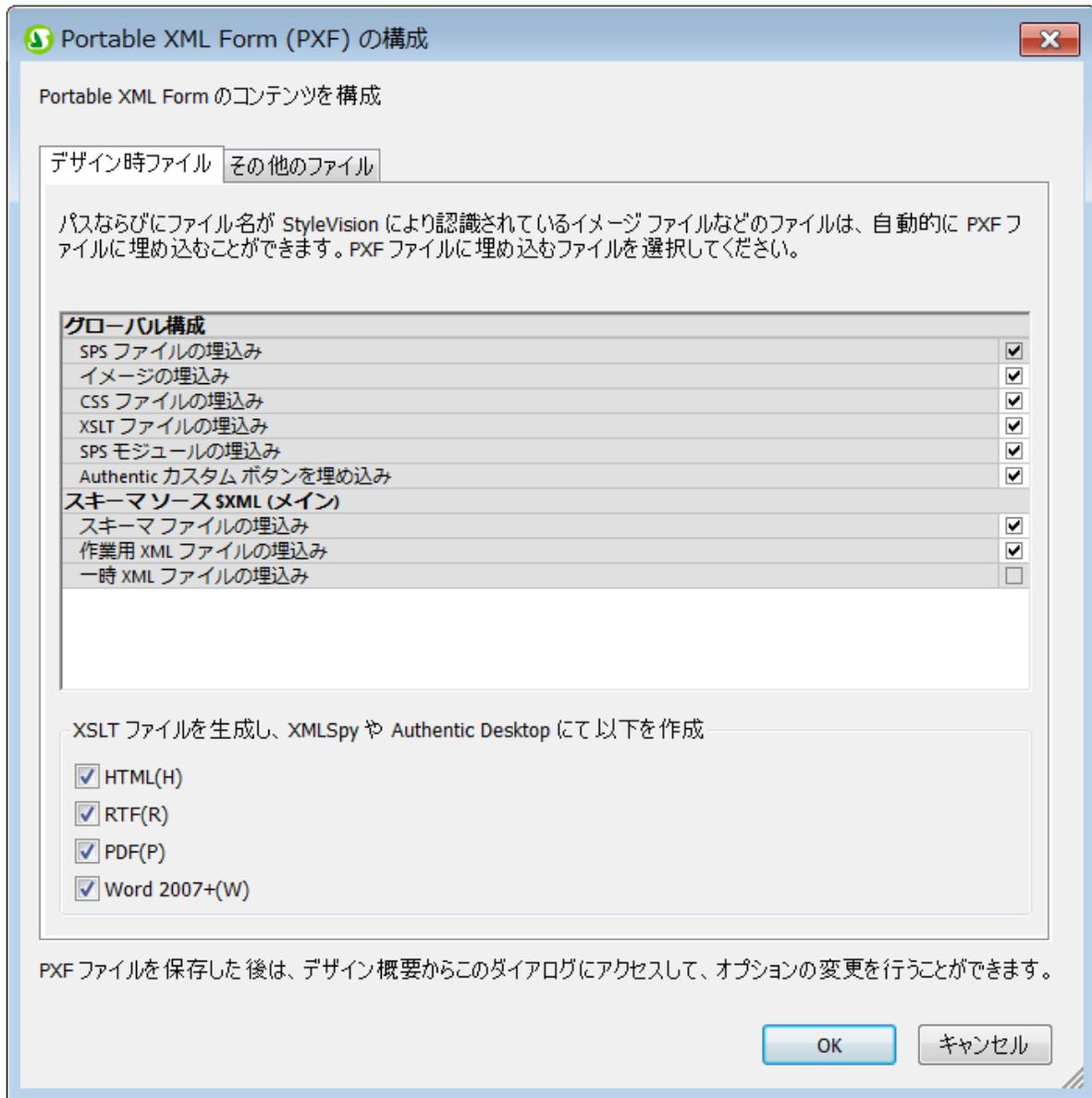
自動化を可能にするために、両方のファイルは、FlowForce Server にデプロイされる必要があります。上の「必要条件」セクションで指定されているとおり、FlowForce Server にライセンスが与えられており、MapForce Server と StyleVision Server が FlowForce Server の管理下で実行されている必要があります。Windows 上では、ライセンスのステータスをチェックするための `verifylicense` コマンドを使用することができます。ライセンスが見つからない、または無効な場合、他のオペレーティングシステムでは、ジョブ実行に失敗すると、エラーが表示されます。

StyleVision スタイルシートを FlowForce Server にデプロイする

1. 「ファイル」メニューから「FlowForce (Server) にデプロイする」をクリックします。(このコマンドが灰色表示されている場合デザイン タブに最初に切り替えます。)
2. ファイルを PXF (Portable XML Form) フォーマットとしてデザインファイルを保存するようプロンプトされると「OK」をクリックします。



3. デプロイされたパッケージ内に含まれるようファイルを選択するようプロンプトされると、デフォルトの設定を保留してください。このサンプル内で、PDF のみが生成されますが、他の出力を含む出力で保存すると、後で HTML と RTF などの追加フォーマットで出力することができます。



4. プロンプトされると、FlowForce Web Server への接続の詳細を入力します。簡素化のため、下のイメージ内では、変換はプレーンHTTP を使用してローカルマシンのポート 8082 にデプロイされていますが、FlowForce Web Server がこのような接続を受け入れるように構成されていると前提して、リモートアドレスを指定して、SSL-暗号化された接続を使用してファイルをデプロイすることもできます。[ネットワーク設定の定義](#)を参照してください。ユーザーとパスワードの値は、レート FlowForce アカウントのため、下で説明されています。しかしながら指定されている、スクリプトを書き込むパーミッションがあることが前提で、その他の FlowForce ユーザーアカウントも使用されている場合があります。このサンプルでは、ジョブの構成は異なるステップであり、後に説明されているため、「新規ジョブを作成するためブラウザを開く」チェックボックスは、意図的にチェックされていません。
5. 他のサンプルとの一貫性のため、ターゲットパス `/public/Examples/PersonListWithGrouping.transformation` を使用することが奨励されます。

**S 変換のデプロイ**

このマッピングをデプロイする FlowForce 管理インターフェイスのホスト名とポートを入力してください。

サーバー(S): 127.0.0.1      ポート(O): 8082

ユーザー(U): root       SSLを使用

パスワード(W): ●●●●

ログイン(L): <デフォルト>

デプロイ

パス (P): /public/PersonListWithGrouping.transformation      参照(B)

パスはスラッシュから開始する必要があります。

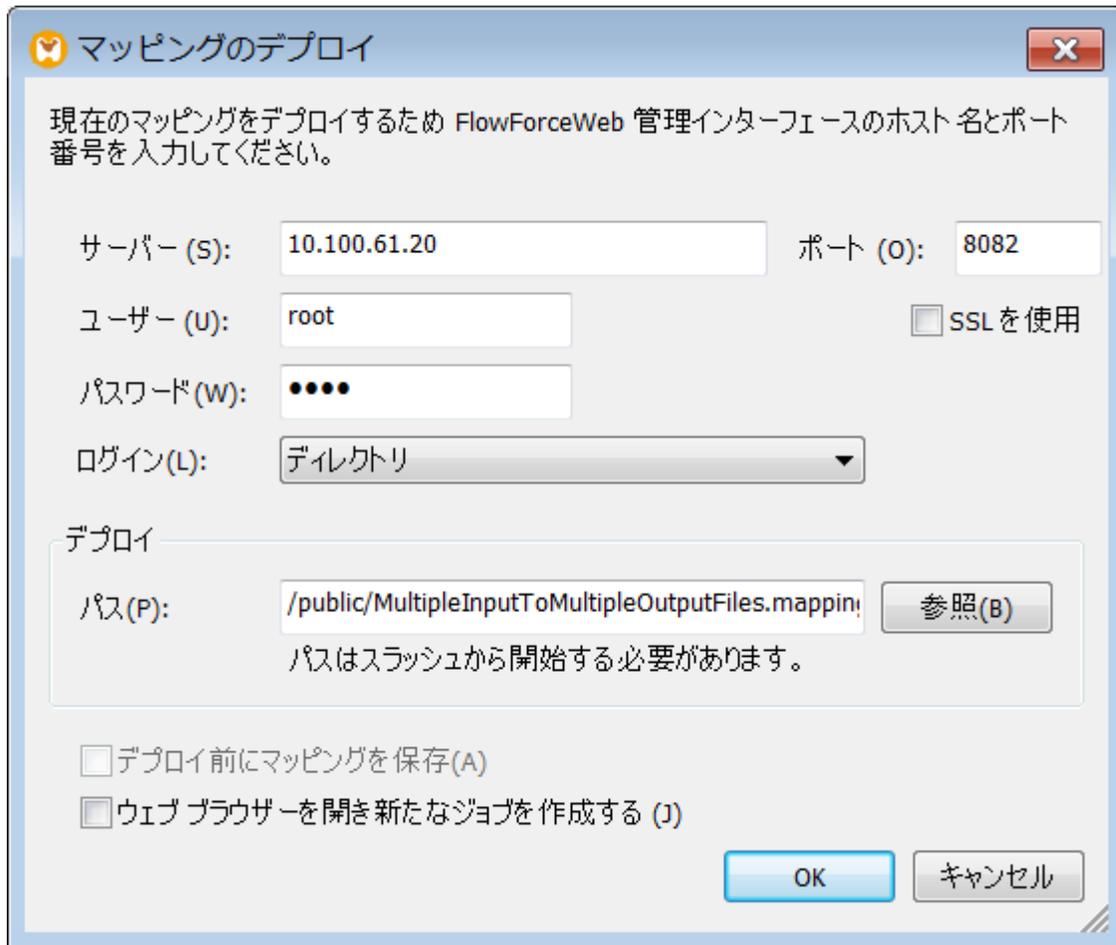
デプロイ前にデザインの変更点を保存(A)

ウェブブラウザを開き新たなジョブを作成する (J)

OK      キャンセル

#### MapForce マッピングをFlowForce Server にデプロイする

- 「ファイル」メニューから「FlowForce Server にデプロイする」をクリックします。StyleVision のために上で説明されているとおなじ接続の詳細を記入します。また他のサンプルと一貫して、ターゲットパス `/public/Examples/MultipleInputToMultipleOutputFiles.mapping` が使用されることが奨励されます。



FlowForce Server にログインすると、ファイルのデプロイに成功すると、指定されたFlowForce コンテナに対応するエントリが表示されます(この場合、"/public/Examples"):



上記のエントリは、まだ、ジョブではないことに注意してください。実際のジョブが作成されるFlowForce 関数 を作成する必要があります。

## FlowForce ジョブの作成と構成

MapForce マッピングとStyleVision 変換がFlowForce Server にデプロイされました。これは、必要とするジョブを作成するために使用されます。方法は以下のとおりです:

1. FlowForce `/public/Examples` コンテナにナビゲートし、以前にデプロイされた関数 `MultipleInputToMultipleOutputFiles.mapping` をクリックします。MapForce マッピングのソースコンポーネントは、FlowForce 関数に対する入力パラメータになっています。また、マッピングにより処理されるインスタンスXMLファイルに設定されるデフォルトの値が存在します。この値は、必要に応じて後にも書き直すことができます。「作業ディレクトリ」パラメータが自動的にFlowForce により追加されています。次のステップでこのパラメータの役割が次のセクションで説明されています。

以下に関数 `MultipleInputToMultipleOutputFiles.mapping` を作成  
`/public`

**関数入力パラメーター**

|  |   |  |
|--|---|--|
| 名前: <input type="text" value="Altova_Hierarchical"/> | 型: <input type="text" value="文字列"/>           | デフォルト: <input type="text" value="C:/Users/ /Documents/Altova/MapForce2018/MapForceExamples/Nanonull-*.xml"/> |
| 名前: <input type="text" value="Working-directory"/>   | 型: <input type="text" value="ディレクトリとしての文字列"/> | デフォルト: <input type="text" value=""/>   |

'MapForce' バージョン '2018' により関数が実行されます。

ジョブの作成

2. [ジョブの作成]をクリックします。
3. 名前を入力し、任意でジョブの詳細を任意で入力します。
4. ジョブの「実行ステップ」部分を以下のように構成します。

### 実行ステップ

- 関数の実行 /public/MultipleInputToMultipleOutputFiles.mapping
  - パラメーター:
    - Altova\_Hierarchical:
    - Working-directory: C:\FlowForceWorkingDir
  - ステップの結果を割り当て output 以下としてPersonList
- For each item シーケンスで results(output)
  - 関数の実行 /system/compute
    - パラメーター:
      - 条件式: as-file(item)
    - ステップの結果を割り当て file 以下としてT0
  - 関数の実行 /public/PersonListWithGrouping.transformation
    - パラメーター:
      - InputXml: {file}
      - OutHtml:
      - GenerateHtmlOutputAsMime:
      - OutRtf:
      - OutFo:
      - OutPdf: {filename(file)}.pdf
      - OutDocx:
      - Working-directory: C:\FlowForceWorkingDir
    - ステップの結果を割り当て 名前 以下としてReturnTypeDocx, ReturnTypeHtml, ReturnTypeF
  - ステップの結果を割り当て 名前

- 「サービス」から「HTTP を使用してこのジョブを使用する」チェックボックスを選択し、オンデマンドでトリガーされる Web サービスの名前を入力します。例えば、GeneratePdfsService が挙げられます。ジョブを予定して実行する、または、ファイルシステムトリガーとして実行する場合は、適切なトリガーを設定してください（[トリガーの管理](#)を参照してください）。

### サービス

このジョブを HTTP を介して URL 上で利用可能にする http://<FlowForce server>/service/ GenerateMultiplePDFs

- 「資格情報」|「オペレーティングシステム ユーザーアカウント」のユーザー名とパスワードを入力します（ジョブは、このユーザーとして実行されます）。FlowForce Web 管理インターフェイスのパスワードとパスワードを区別してください（[資格情報](#)を参照してください）。

### 資格情報

資格情報によりジョブを実行:  既存の資格情報を選択:

ローカルの資格情報を定義: ユーザー名:

パスワード:

7. [保存]をクリックします。

ジョブの作動を理解するには、ジョブの「実行ステップ」セクションに注目してください。最初の実行ステップは、前にデプロイされたマッピングを呼び出します。作業ディレクトリ内で [Nanonull] から始まる XML ファイルを検索します。このサンプルでは、作業ディレクトリは **C:\FlowForceExamples\GeneratePdfs** です。

最初の実行ステップにより返された出力は、マッピングにより返されたデータを表示しています。次のステップ内で参照できるように、**output** と明示的に名前が付けられています。

ジョブの2番目のステップは「for-each」ステップです。「for-each」ステップが FlowForce 式 **results (output)** を使用して、(マッピングにより返された出力である)最初のステップにより返されたデータにアクセスします。具体的には、式は、前のステップにより返された引数出力として取る関数 **results ()** を呼び出します。[ステップ結果関数](#)を参照してください。FlowForce 式の基本的な紹介に関しては、[FlowForce 式](#)を参照してください。

「for-each」ステップは、2つの小さな実行ステップにより構成されています:

1. 最初のステップは、マッピング出力を通常 **file** と呼ばれる、実際のファイルに変換する `/system/compute` ビルトイン関数を呼び出します。重要な点は、**results (output)** 式の出力は、ファイルではなく、ストリームである点です。(このサンプル同様)マッピングが複数の出力を返す場合、マッピング出力は、ストリームのシーケンスを表示しています。この理由のため、FlowForce 式関数 (**as-file**) が反復される現在のストリームを実際のファイルに変換するために使用されます。
2. 2番目のステップは、前に StyleVision 変換によりデプロイされた反復される各ストリームを呼び出します。具体的には、反復ごとに、StyleVision Server が呼び出され、XML ファイルが入力として与えられ、PDF ファイルは、出力として返されます。FlowForce 式 **{filename (file)}.pdf** は、実際の PDF ファイル名をディスク上に生成します。このオペレーションは、「作業ディレクトリ」/コマンドにより指定された作業ディレクトリ内で実行されます。

**メモ** このサンプルでは、同じ作業ディレクトリは、マッピングを実行し、StyleVision 変換を実行する両方のステップで使用されています。一部の場合、ファイル名の競合からジョブの実行エラーを回避するために個別に作業ディレクトリを指定する必要があります。

## ジョブを実行する方法

ジョブのために入力データを準備するには **Nanonull-Branch.xml** と **Nanonull-HQ.xml** を **<Documents>\Altova\MapForce2021\MapForceExamples** から作業ディレクトリ (**C:\FlowForceExamples\GeneratePdfs**) にコピーします。このようにして、ジョブが実行される際にデータを読みとるためにジョブの最初のステップはファイルの入力を取得します。

ジョブを実行するには、以下の内の1つを行ってください:

- ホームで移動し、全てのアクティブなトリガーとサービスを表示するをクリックします。次の情報列内に表示されているジョブの URL をクリックします。
- <http://127.0.0.1:4646/service/GeneratePdfsService> をブラウザのアドレスバーに入力します。この URL は FlowForce Server サービスがデフォルトのホストアドレスとポート名でリスニングする場合のみ機能します。[構成ページ](#)内で他のホストとポート設定を定義した場合、アドレスを必要に応じて変更してください。

- FlowForce Server の任意のホスト名 フィールドを [セトアップページ](#) から設定すると HTTP を介してこのジョブを使用可能にする。チェックボックスの横の  ボタンをクリックして web サービスの呼び出しを直接ジョブ構成ページから実行することができます。それ以外の場合、このボタンは表示されません。

Web サービスにアクセスする際に資格情報をプロンプトされた場合、FlowForce Server にログインするために使用する同じ資格情報を提供してください。

FlowForce Server ユーザーに HTTP 認証のための資格情報を提供することはテスト目的のためです。生産のために新規の FlowForce ユーザーを作成することが奨励されており、ジョブが存在するエンタープライズでこのユーザーに **Service - Use** パーミッションを与え、対応するユーザーアカウントを使用して Web サービスにアクセスします。HTTP 認証を無効化することは、Web サービスを公開し、**Service - Use** パーミッションをユーザー **Anonymous** に与えます。[パーミッションのしくみ](#) を参照してください。

ジョブに成功すると作業デスク外 **C:\FlowForceExamples\GeneratePdfs** 内にジョブにより生成された PDF ファイルが表示されます。PDF ファイルをブラウザに出力することができないため、ブラウザでジョブを出力することができません」が実行が成功した場合でも表示されます。した場合、ブラウザは「サービスの実行に失敗しました」メッセージを表示します。この場合ジョブの [エラーログ](#) を参照してください。問題をトラブルシューティングする場合、このページの上でリストされている必要条件全てをもう一度検証する必要があるかもしれません。

## 13 用語

用語のセクションはFlowForce Server に関する用語のリストを含みます。

## 13.1 C

### キャッシュコンシューマー

使用可能な場合キャッシュからデータを取得するジョブ(つまり、キャッシュを「消費する」ジョブ)。キャッシュの結果が使用できない場合、コンシューマーはジョブを実行し、実際のジョブの結果を取得します。

### キャッシュプロデューサー

データを使用してキャッシュを作成するジョブ(つまり、キャッシュを「作成する」ジョブ)。

### 選択ステップ

型「選択ステップ」のステップにより、どの条件下で他のジョブステップが実行されるかを定義することができます。選択ステップは以下の構造が適用されます。

```
When {some expression}
    Execute (some step)
Otherwise
    Execute (some other step)
```

各「選択ステップ」の下では、他の選択ステップ(サブ条件)をネストすることができます。例:

```
When {expression}
    When {expression}
        Execute (step)
    Otherwise
        Execute (step)
Otherwise
    Execute (step)
```

定義することのできる条件ステップの数量に制限はありません。

When / Otherwise ペアの中では、FlowForce Server は、true である条件のみを実行します。他の条件は無視されます。

### コンテナー

名前が意味するように、コンテナーはパッケージされたデータです。FlowForce Server では、コンテナーは、オペレーティングシステム上のフォルダーと大まかに比較することができます。コンテナーは以下を含むことができます: [ジョブ](#)、[資格情報](#)、[関数](#) と他のコンテナー。コンテナーで [パーミッション](#) を設定することにより、コンテナー内のデータを閲覧およびデータにアクセスする権利を管理することができます。データをコンテナー別に整理し、対応する [パーミッション](#) を各コンテナーのためにセットアップすることは、よいセキュリティ対策となるでしょう。

### 資格情報

資格情報オブジェクトは、認証に関する情報を保管します。これは FlowForce Server ジョブが作動する箇所でオペレーティングシステム上のユーザーアカウントに関連するユーザー名とパスワードの組み合わせですが HTTP または FTP 資格情報のセットであることもできます。

## 13.2 E

### エラー/成功の処理ステップ

ジョブのステップに失敗すると、FlowForce Server は、ジョブを中断します。型「エラー/成功の処理ステップ」の[ステップ](#)を使用して、ジョブを完全に完了する前に、クリーンアップアクションを実行します（以下を参照：[ステップのエラーの処理](#)）。

### 実行の結果

FlowForce Server 内では、ステップの実行はステップの実行後に返される内容を定義します（例えば、ファイル、または、テキスト）。ジョブと作業する際、ステップの実行の結果を特定のデータ型（文字列、または、ブール値）になるように、または破棄されるように、明示的に宣言することができます。通常、結果を他のジョブで使用する場合、または、結果をキャッシュする場合、実行の結果を宣言する必要があります。

### 実行ステップ

型「実行ステップ」のステップにより特定のFlowForce 関数を実行することができます。FlowForce Server により提供される MapForce マッピングをデプロイする、または変換を行う、他のジョブのステップを実行する[ビルトイン関数](#)を含む関数を使用することができます。

## 13.3 F

### For-Each ステップ

型「For-Each ステップ」の[ステップ](#)により、シーケンス中を反復し、実行ステップを無制限に繰り返すことができます（例えば、ディレクトリ内のファイルのリスト）。「for-each」ステップの構成は以下の通りです。

```
For each item in sequence {sequence expression}
  Execute (step)
```

FlowForce は、シーケンス式の全てのアイテムをループして完了するまでステップを実行します。

### 関数

[ジョブ](#)のコンテキストでは、関数は FlowForce Server により理解される、ターゲットファイルシステム上のオペレーションを実行するための命令です。関数は以下であることができます：

- ビルトイン FlowForce 関数（以下を参照: [ビルトイン関数](#)）
- StyleVision 変換
- MapForce マッピング
- ジョブの実行ステップ

関数の多くは入力パラメータを有します。必須入力パラメータは、ステップを成功裏に実行するため呼び出し側により与えられる必要があります。

## 13.4 I

### 入力パラメーター

FlowForce Server [ジョブ](#) のコンテキストで、入力パラメーターは、どのジョブが特定の方法で実行されるかを表したジョブに与えられた任意の情報を意味します。ジョブの入力パラメーターは、関数引数(プログラム言語の観点から類似しています)。入力パラメーターは多種の型(例えば、ファイルまたはディレクトリパス、テキスト、番号、ブール値、その他)であることができます。一部のジョブの型に対しては、FlowForce Server は、入力パラメーターを自動的に作成します。

## 13.5 J

### ジョブ

ジョブとはFlowForce Server 内のコア概念です。この概念サーバー上で実行されるはタスクまたはタスクのシーケンスを表します。ジョブは電子メールの送信など1つのステップから構成されるジョブであることができます。しかしながら、複数のアクションを実行し結果(例えば、ファイル)を他のジョブのラメータとして返すジョブを作成することもできます。ジョブは[入力ラメータ](#)、[ステップ](#)、[トリガー](#)および他の設定から構成されます。

### ジョブインスタンス

ジョブインスタンスはジョブとは異なります。FlowForce ジョブをジョブ構成ページから構成すると、ジョブ構成を作成することができます。ジョブの定義済みのトリガーの必要条件が適用されると、ジョブの実行が開始されます。

## 13.6 M

### マスター

ジョブによりトリガーされた条件を継続的に評価し、FlowForce サービスインターフェイスを提供する「マスター」はFlowForce Server インスタンスです。マスターは同じクラスター内のノードを考慮し、ジョブインスタンスを処理する、更にはまた、代わりに、ジョブインスタンスを割り当てるように構成されている可能性があります。

## 13.7 P

### パスワードポリシー

FlowForce Servers は、管理者が複雑なユーザーパスワードを管理するパスワードポリシーを使用します。パスワードポリシーとは、有効になるためのユーザーパスワードが必要とする最低条件です（例えば、最低 *N* 文字のなど）。

### パーミッション

パーミッションは、ユーザーのコンテナへのアクセスを管理します。特権と同様、[パーミッション](#) は、ユーザーとロールに与えられることができます。ですから、ユーザーがロールの一部である場合、ロールに与えられたパーミッションは、ユーザーにも自動的に適用されます。

デフォルトで、コンテナに設定されたパーミッションは、親コンテナから継承されます。例えば、コンテナ A には、子コンテナ B があります。コンテナ A にアクセスするパーミッションを持つユーザーは、コンテナ B にもアクセスすることのできるデフォルトのパーミッションを持つこととなります。しかしながら、管理者は、コンテナの階層の全てのレベルで、全てのユーザーおよびロールのパーミッションを再定義することができます。

### 特権

特権は、FlowForce Server 内でユーザーが行うことのできる内容を定義します（例えば、自身のパスワード設定、ユーザーとロールの読み取り、ジョブの停止、など）。特権は、パーミッションはコンテナへのユーザーアクセスを管理し、特権はグローバルに FlowForce Server 全体に対して効果がある点で、[パーミッション](#) と異なります。次の簡単なルールは、特権とパーミッションの違いを理解する助けとなります。特権はグローバルであり、パーミッションはローカルです。

パーミッションに類似して、特権は個々のユーザーとロールに割り当てることができます。ですから、ユーザーが FlowForce Server にログインすると、有効な特権は以下により決定されます：

- a) 直接割り当てられた特権
- b) ユーザーがメンバーであるロールに割り当てられた特権

## 13.8 Q

### キュー

実行キューは「ジョブのプロセッサ」です。これはジョブインスタンスがどのように実行されるかを管理します。実行するためには、各ジョブはターゲット実行キューに割り当てられます。キューは（キューに割り当てられているすべてのジョブの）ジョブインスタンスが一度に実行される数、実行間の遅延を管理します。デフォルトで、キュー設定はジョブに対してローカルですが、キューを複数のジョブに共有されるスタンドアロンオブジェクトとして定義することもできます。キューはジョブに対してローカルであること、または、複数のジョブで共有されることもできます。複数のジョブが同じキューに割り当てられている場合、実行のためにそのキューは共有されています。

更に、適切な（特権とは異なる）コンテナパーミッションを有する場合のみキューを確認、または、キューにジョブを割り当てることができます。具体的には、キューを作成するためユーザーが実行キューを定義する特権を持つ必要があります。[特権のしくみ](#)を参照してください。更に（特権とは異なる適切なパーミッションがある場合のみ）ユーザーはキューを確認、またはキューにジョブを割り当てることができます。[パーミッションのしくみ](#)を参照してください。これは、キューにジョブを割り当てることができることを意味します。キューへのアクセスを制限するには、キューが定義されている個所にコンテナをナビゲートし、コンテナのパーミッションをロール **authenticated** のために「キュー-アクセス無し」に変更します。次に、パーミッション「キュー-使用」を特定のロール、または、必要なユーザーに割り当てます。詳細に関しては、[/public コンテナへのアクセスを制限する](#)を参照してください。

## 13.9 R

### RaptorXML Server

(略して RaptorXML と呼ばれる) Altova RaptorXML Server は、マルチプラットフォーム環境と最新の標準のために最適化された Altova の第3世代の高速 XML プロセッサです。高水準なクロスプラットフォームの機能性のために設計されたエンジンは、XML の高速な処理を行うために、現在のマルチコアコンピュータの利点を活用しています。RaptorXML には2つのエディションが存在します: (i) **RaptorXML Server** と (ii) **RaptorXML+XBRL Server**。RaptorXML+XBRL Server エディションは、XML に加え、XBRL (eXtensible Business Reporting Language) ドキュメントの検証と処理へのサポートを含んでいます。

### RaptorXML+XBRL Server

RaptorXML+XBRL Server は、検証へのサポートを含み、XML に加え XBRL (eXtensible Business Reporting Language)、ドキュメントを処理する RaptorXML Server の特別なエディションです。

### ロール

ロールは、ビジネスニーズをベースとしたセキュリティの強化を手助けする **特権** のセットです。標準的なロールをベースとしたセキュリティには、最低2つのロールが含まれます: 管理者と一般ユーザー。各ロールは、ロールに与えられた特権により定義されています。例えば、管理者は、自身および他のユーザーのパスワードを変更することができますが、一般ユーザーは自身のパスワードのみを変更することができます。ユーザーにロールを割り当て、必要に応じて、ユーザーのロールを取り消すことができます。

## 13.10 S

### ステップ

ステップは、FlowForce Server ジョブが実際に行う内容を定義します(例えば、ファイルの削除、MapForce マッピングの実行、または電子メールの送信など)。最も簡単な形式としては、ステップは失敗または成功した結果を持つオペレーションで、実行されるために関数を必要とします(以下を参照: [実行ステップ](#))。しかしながら、ループとして他のステップを条件付きで実行する方法を与えるステップも存在します(以下を参照: [選択ステップ](#)、[エラー成功の処理ステップ](#) および [For-Each ステップ](#))。同じジョブ内で必要な数だけステップを作成し、実行される順序を設定することができます。

## 13.11 T

### トリガー

ジョブを作成する際、トリガーとしてジョブをトリガーする条件(または基準)を指定する必要があります。FlowForce Server は、定義されたトリガーを監視し、トリガーの条件が満たされるとジョブを実行します。

## 13.12 U

### ユーザー

ユーザーとはFlowForce Server にログインジョブを構成、またはMapForce またはStyleVision 変換をデプロイ、FlowForce Server を管理する個人を指します。ユーザーが使用することのできるアクションは以下には影響されません

- a) 割り当てられた [パーミッション](#) または [特権](#)
- b) ユーザーがメンバーであるロールに割り当てられている [パーミッション](#) と [特権](#)

## 13.13 W

### Web 管理インターフェイス

FlowForce Server Web 管理インターフェイスは、サーバーを管理し、ジョブを構成するFlowForce Server のフロントエンドです。Web 管理インターフェイス[アドレスとポートの構成](#)のWeb のブラウザからアクセスすることができます。

### 作業ディレクトリ

作業ディレクトリとは、ジョブが入力ファイルを展開、または、出力ファイルを保存する場所が必要な場合の[実行ステップ](#)により必要とされるパラメータです。FlowForce Server は、また、作業ディレクトリをステップの実行中に発生する相対パスを解決するために、使用します。作業ディレクトリを与えるように問われた場合、FlowForce Server が実行されているオペレーティングシステム上の有効なパスを与えてください。ステップを作成中に、作業ディレクトリが与えられない場合、FlowForce Server は一時的なディレクトリを使用します。

# インデックス

## A

Active Directory,  
FlowForce Server との統合, 81

Altova LicenseServer,  
新しいマシンに移行, 46

## C

Container,  
definition, 449

Credentials,  
definition, 449

## E

Execution Step,  
definition, 458

## F

FlowForce Server,  
Web 管理インターフェイス, 23  
アーキテクチャ, 21  
アプリケーションデータフォルダー, 92  
コマンドライン インターフェイス, 239  
システム必要条件, 27  
ジョブの基本概念, 18  
セットアップページ, 49  
ネットワークアドレスとポートの設定, 53, 56  
のローカライズ, 94  
の言語の変更, 256  
の構成, 48  
ビルトイン関数, 261  
ローカライズ, 248  
ログオン, 25  
基本セキュリティ概念, 20

最新のバージョンに移行する, 253  
新しいマシンに移行, 46  
新規機能, 14  
用語, 448

FlowForce Server Function,  
definition, 451

FlowForce Web Server,  
ネットワークアドレスとポートの設定, 53, 56

For-each ステップ,  
定義, 451

FTP,  
資格情報, 125, 128

## H

HTTP,  
リクエストボディのリミットを増やす, 56

HTTP トリガー,  
の概要, 161

HTTPS 接続,  
FlowForce が次を受け入れるように構成する, 53, 56

## I

INI ファイル,  
の構成, 56

Input Parameters,  
definition, 452

## J

Job,  
definition, 453

## L

LDAP,  
FlowForce Server との統合との統合, 81

Linux,  
でサービスを開始する, 89  
でサービスを停止する, 89

Linux,  
信頼できるサーバー証明書, 66

## M

Mac,  
サービスの停止, 90  
でサービスを開始する, 90  
信頼できるサーバー証明書, 66

MapForce Server,  
ジョブからの呼び出し, 436  
新しいマシンに移行, 46  
統合, 208, 218

MobileTogether Server,  
新しいマシンに移行, 46

## O

OAuth 2.0,  
資格情報, 125, 127, 128  
認証, 221

## P

PDF ファイル,  
生成, 436

Permission,  
definition, 455

Privilege,  
definition, 455

## R

RaptorXML Server,  
定義, 457

RaptorXML 関数,  
とジョブを作成する, 425, 427

RaptorXML(XBRL) Server,  
新しいマシンに移行, 46

RaptorXML+XBRL Server,  
定義, 457

## S

SSL,  
接続の暗号化, 53, 56

SSL 暗号化,  
FlowForce Web サーバーのために を有効化する方法, 74  
FlowForce サーバーのために を有効化する方法, 76  
秘密キーの暗号化, 73  
秘密キーの必要条件, 73

Step,  
definition, 458

StyleVision Server,  
ジョブからの呼び出し, 436  
新しいマシンに移行, 46  
統合, 208, 218

## T

Trigger,  
Definition, 459  
triggerfile パラメーター,  
目的, 132

## U

UNC,  
syntax in paths, 173  
パス内の構文, 156

## W

Web サービス,  
としてジョブを公開する, 163, 401  
認証, 163

Web 管理インターフェイス,  
定義, 461

Windows,  
でサービスを開始する, 91  
ネットワークパスへの参照, 261  
以下でサービスを停止する, 91

- Windows,
  - 信頼できるサーバー証明書, 66
- Windows ドメインユーザー,
  - FlowForce Server にインポートする, 100
  
- Z**
  
- アイテム,
  - FlowForce データ型として, 205
- アプリケーションプール, 236
- インストール,
  - インストールフォルダー (Linux), 32
  - インストールフォルダー (Mac), 35
  - インストールフォルダー (Windows), 45
  - システム必要条件, 27
  - 現在インストールされている製品を確認する (Linux), 32
- エラー/成功の処理ステップ,
  - 定義, 450
- エラーの処理,
  - ジョブ内, 140, 144
  - とジョブを作成する, 395
- エラーログ,
  - とジョブを作成する, 427
- キー/値ペア,
  - XSLT シートにパスする, 431
- キャッシュ コンシューマー,
  - 定義, 449
- キャッシュプロデューサー,
  - 定義, 449
- グローバルリソース,
  - FlowForce Server 内での使用, 224
- コマンドラインインターフェイス,
  - ヘルプの取得, 243
- コンシューマー,
  - 定義, 449
- コンテナ,
  - /public へのアクセスを制限する, 120
  - のパーミッションを閲覧する, 118
  - の概要, 114
  - パーミッションの設定, 119
  - 移動, 116
  - 作成, 116
  - 名前の変更, 116
- ジョブ,
  - MapForce マッピングから作成する, 379
  - StyleVision 変換から作成, 417
  - キュー設定の定義, 170
  - コピー, 131
  - ステータスのレファレンス, 190
  - のログを確認する, 194
  - の結果のキャッシュ, 413
  - の結果をキャッシュ, 167
  - の戻り型を宣言する, 152
  - ファイルからインポート, 175
  - ファイルへエクスポート, 175
  - 作成, 130
  - 資格情報, 125, 128
  - 実行のモニタリング, 188
  - 他の FlowForce Server インスタンスにエクスポート, 175
  - 停止, 193
  - 複製, 131
- ステップ,
  - ジョブ内での使用の例, 386
  - 作成, 134
  - 内でのエラー処理, 140
- タイマートリガー,
  - の概要, 158
- ツールファイル,
  - オプション, 236
- ディレクトリ ポーリング ジョブ,
  - の例, 389
- データ型,
  - FlowForce 内, 205
- デフォルトのタイムゾーン,
  - の構成, 79
- トリガー,
  - 作成, 157
  - 削除, 157
- パーミッション,
  - のリスト, 111
  - の概要, 111
- パスワードポリシー,
  - の概要, 122
  - 作成, 122
  - 定義, 455
- ファイルシステムのトリガー,
  - の概要, 159
- ホスト名,
  - FlowForce Server のための設定, 53
  - FlowForce Web Server のための設定, 53
- ユーザー,
  - ビルドインのロール, 101
  - 作成, 99
  - 名前の変更, 102

- リソース,
  - 作成, 224
  - 編集, 224
- ルートユーザー,
  - のパスワードのリセット, 104
- ロール,
  - ビルドインのロール, 101
  - ユーザーに割り当て, 103
  - ロールの割り当て, 103
  - 作成, 100
  - 定義, 457
  - 名前の変更, 102
- ログ,
  - サイズの削減, 84
  - のジョブインスタンス, 196
  - 設定, 84
- 延期されたステップ,
  - サンプル, 147
  - の概要, 147
- 環境変数,
  - 設定, 237
- 関数,
  - 定義, 451
- 結果, 205
  - FlowForce データ型として, 205
- 再試行カウントの設定,
  - ジョブ構成オプションとして, 142
- 作業ディレクトリ,
  - 使用法, 218
  - 定義, 461
- 資格情報,
  - 追加, 125
- 式, 311
- 実行の結果,
  - 定義, 450
- 新しいマシンに移行,
  - Altova LicenseServer, 46
  - FlowForce Server, 46
  - MapForce Server, 46
  - MobileTogether Server, 46
  - RaptorXML(XBRL) Server, 46
  - StyleVision Server, 46
- 選択ステップ,
  - 定義, 449
- 電子メールパラメーター,
  - の構成, 80
- 電子証明書,
  - Linux 上で信頼できる, 66
  - Mac 上で信頼する, 66
  - Windows 上での管理, 70
  - Windows 上で信頼できる, 66
- 特権,
  - に関するレポートを閲覧, 108
  - のリスト, 105
  - の概要, 105
  - 継承, 105
- 入力パラメーター,
  - ビルトイン, 132
  - 削除, 132
  - 追加, 132
- 認証,
  - HTTP, 25, 163
  - Windows ドメイン, 25, 163