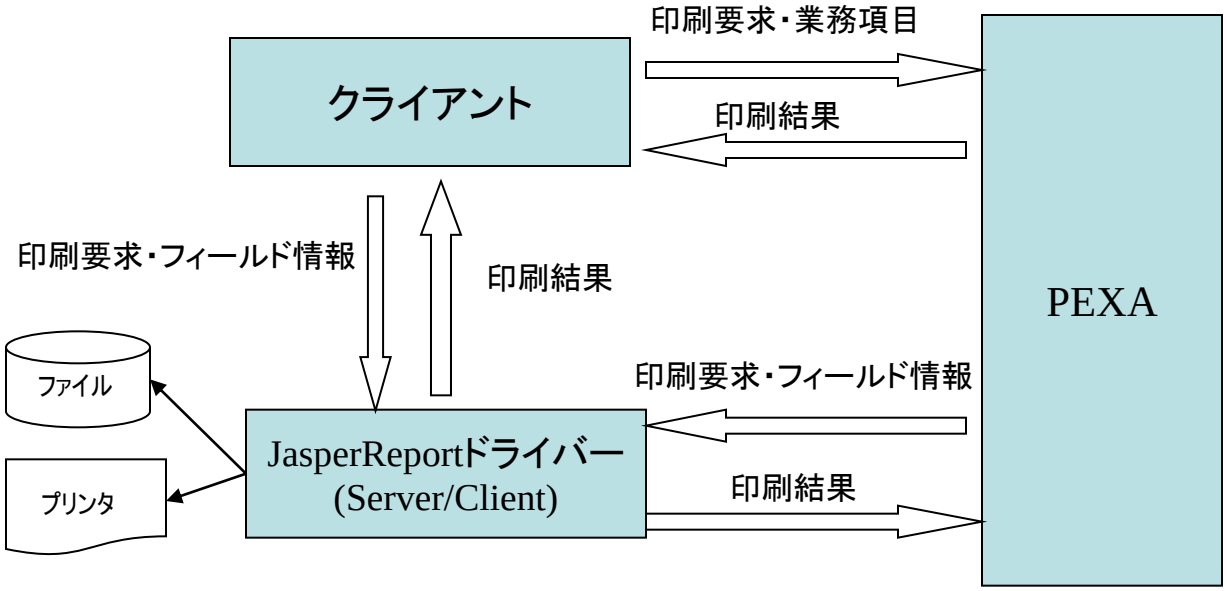
PEXA　PrintService Guide

JasperReport（iReport4.6）編

PrintService目次

PrintService概要

- PEXAPrintServiceはPEXA環境下で、モデル定義から簡易に印刷を行うためのFacadeSessionを提供するものです。
- 直接印刷するドライバー部分は、PEXA内のJasperReport専用ドライバーを利用することにより、印刷を実行します。JasperReportのほかにウイングアークテクノロジーズのSVF製品用の専用ドライバーも提供されています。す。本仕様書はJasperReportを前提に記述します。また、帳票設計にはiReportを使用します。



PrintService機能

PEXAPrintServiceは以下の機能を有します。

1. 業務項目から帳票フィールド定義へのデータマッピング
2. サブレポート用データのマッピング
   帳票内に明細項目を複数印刷することが出来ます。
3. ダウンロード用のPDFファイル作成機能
4. リモートプリンタへの直接印刷機能
   サーバーに接続しているプリンタに限ります。
5. ローカルプリンタへの直接印刷機能
6. フィールド表示時のフォーマット変換用のHelperクラス指定機能

PrintService設定

PrintService設定ファイル

格納位置　　src/print/service/*.print

PrintフォームIDごとの以下の定義を持ちます。

- 業務項目フィールドマッピング情報
- フォームJasperReports定義ファイルのパス名
- データ操作用のHelperクラス指定
- 帳票のサブフォーム指定(オプション)

PrintDriverファクトリクラス 設定ファイル

PrintDriverファクトリクラスの設定ファイル

格納位置　　/src/print/print_driver.entry

jasper用には下記の3種類のファクトリクラスをサポートしています。

先頭のキーはPrintタイプを示します。

1. **jasper**
   pexa.ext.share.service.print.jasper.JPPrintServiceFacadeFactory
   プリントサーバーがAPサーバーと別な場合に使用します。（APサーバーと同じでも可能です）
2. **jasper_local**
   pexa.ext.share.service.print.jasper.JPPrintServiceFactory
   プリンタサーバーがAPサーバーと等しい場合に使用します。
3. **jasper_pc**
   pexa.ext.share.service.print.jasper.LocalJPPrintServiceFactory
   クライアント上で直接印刷・PDFファイルの作成を行いたいときに使用します。

PrintService設定ファイル（1）

PrintフォームIDごとの以下の定義を持ちます。

格納位置　　src/print/service

```
(print
        printType       jasper                  ;Printタイプ（プリンタドライバ）
        print_name      TESTSUB                 ;PrintフォームID
        option          reportWriter            ;帳票のサブフォーム指定(オプション)
        formFile        TESTSUB.jasper          ;フォームjasperReport定義ファイル名
        {field                                  ;フィールドマッピング情報（先頭がフィールド名）
            郵便番号                ヘッダー.郵便番号
            住所1_!imeg.server.util.print.helper.TestFormat(testx)          ヘッダー.時刻
            お客様名                ヘッダー.お客様名
            顧客番号                ヘッダー.顧客番号
            開始日                  ヘッダー.開始日
            終了日                  ヘッダー.終了日
            {請求明細1
                    品名                    ヘッダー. COMBI.品名
                    数量                    ヘッダー. COMBI.数量
                    単価                    ヘッダー. COMBI.単価
            }
        }
)
```

PrintService設定ファイル（2）

- **printType　　（プリンタタイプ）**
  src/print/print_driver.entryで指定してあるキーを指定します。
  - jasperはプリント時APサーバで、帳票の出力操作を行うときに使用します。
  - jasper_localはプリント時クライアントローカルマシンのみで、帳票の出力操作を行うときに使用します。

- **print_name　　（PrintフォームID）**
  サービスのPrintにおいて、sheet宣言に指定されるシート名

- **option**
  帳票出力でのオプションです。
  - pagecon
    複数帳票フォーム指定時、ページ番号を継続して通番とします。
    未指定時は、帳票フォーム単位で、ページ番号を振り直します。
  - PageSizeA2, PageSizeA3, PageSizeA4, PageSizeA5
    Local_PCに接続するプリンタへの直接出力時、用紙サイズをA2からA5まで直接指定します。
    印刷ダイアログにも指定したPageSizeで設定されます。

  複数のOptionを併記する時は、“－”ハイフンで連結します。例：pagecon-PageSizeA3

- **formFile　　（フォームjasperReport定義ファイル名）**
  src/print/service下に指定してあるjasperファイルの名称。
  出力する帳票のiReportで作成されたフォームファイルの名称（コンパイル済）

PrintService設定ファイル（フィールドマッピング情報）

基本的にフィールド名と業務項目名から構成されます。
先頭はフィールド名で、後方が業務項目名となります。
フィールド名はiReportにおけるParameters,Fieldsでの名称となります。

例：　郵便番号　　郵便番号

- **表示変換**

フィールド名の設定で、2つのアンダースコア '_' でフィールドにデータを設定する時にデータのフォーマット変換の指定ができます。

例:　住所1_!imeg.server.util.print.helper.TestFormat(testx)時刻

上記の場合、フォーマット変換用のヘルパークラスの指定です。

他に、データのクラスがDateの場合、フォーマット変換の文字列が、そのまま SimpleDateFormatの変換文字列として使用されます。

例:　支払日_yyyy/MM/dd　　日付

和暦を出す場合は、以下のようにフォーマットとともに記述します。

例：　生成日_JP_GGGGyyyy/MM/dd　　CreateDate

変換文字列の前にJP_を挿入して指定します。GGGGも和暦表示には必須です。

PrintService設定ファイル（業務項目の指定）

業務項目の指定

業務項目名はデータのパスも指定します。出力データに階層がある場合、“.”ピリオドで結合します。

また、業務項目がパス指定を設定することも可能です。

例１:

作成者名　　ヘッダー.作成者No/利用名　;パス指定

PrintServiceデータフォーマットHelper クラスについて

フィールド部にデータセットする時に呼ばれるフォーマット変換用のHelperクラスです。
インターフェースは、pexa.share.util.print.PrintServiceFormatHelperにあります。
メソッドが1個指定してあり、以下のようなパラメータを持ちます。
例:imeg.server.util.print.helper.TestFormatHelper.java

```
/**
 * フォーマットヘルパー実行メソッド
 * @param data 変換元データ
 * @param pattern  パターン、()の中身だけを貰う。
 * @return  変換後のデータ
 */
public Object process(Object data,String pattern)
```

設定例:
住所1_!imeg.server.util.print.helper.TestFormatHelper(testx) “testx”としてパラメータに設定されます

リソースファイルの設定

- PDFのWebサーバー格納位置指定などの設定をセットするファイルを設定する必要があります。
- 格納位置
  - src/print/res/LocalPrintService_ja_JP.native
- 内容
  - PDFを格納する位置（JBOSSのCATALINA_HOMEからの相対パス）

local_spool_directory ￥

deploy/jbossweb-tomcat55.sar/ROOT.war/print/svf_spool/

  - URLとして指定するパス

local_spool_￥

uriprint/svf_spool/

iReportでの設定

- iReportので、テキストフィールド表現等に指定する名称とPexaPrintServiceフィールド名のMappingをiReport上で行う必要があります。

- ヘッダー部に当たるデータ
  - iReportのDocument構造のパラメータで指定します。
  - テキストフィールド表現　　$P{NAME}

- 明細部に当たるデータ
  - iReportのDocument構造のフィールドで指定します。
  - テキストフィールド表現　　$F{DNAME}

- 複数の明細部がある場合
  - SubReport機能を使用してメインの帳票上にSubReport領域を指定し、SubReport用のiReportのDocumentを別に作成し、別作成のフィールドで明細情報を指定します。

- iReportの設定ファイルは、src/print/jasperに格納します。

Serviceプロセスからの PrintServiceの呼び出し

サービス内のプロセスでPrintコマンドを呼び出します。

```
(service
        service_name PR_TEST01
        return_keys             プリント結果
        {process
           (帳票データ取得
                    format_type       search
                    (search
                             source              予定原価申請_横型明細データ
                             session_value       予定原価申請_横型明細
                             zero_is_null        true
                    )
           )
           (印刷実行
                    format_type print
                    (print
                             sheet               PR_ACPY_XXX
                             printer   PDF
                             session_value       印刷日時,予定原価申請_横型明細,販売会社
                             )
                    )
           )
        }
)
```

printプロセスのパラメータ（1）

Printプロセスは以下のパラメータを持ちます。

sheet宣言(1)
location宣言(0|1)
printer宣言(1)
session_value宣言(1)

- **sheet宣言**
  - **概要：帳票名を設定します。**
    帳票名はprintServiceMap設定ファイルで指定されます。
  - **キー：　「sheet」固定**
  - **形式：　Keyまたは帳票名を持つセッションキー(@付)**
  - **記述例：**
    sheet　申請書一覧
    shhet @シート名

printプロセスのパラメータ(2)

- **location宣言**
  - **実際に出力するプリンタの接続マシンの位置がローカルPCであることを設定します。出力先プリンタがサーバー接続の場合は、記述しません。**
  - **キー: 「location」固定**
  - **形式: local**
  - **記述例:**
    location local

printプロセスのパラメータ（3）

- **printer宣言**

概要：出力先のプリンタを指定します。以下の名称は特殊な名称として、特定の機能を持つプリンタとして扱われます。local系のPrinter設定時は、**jasper_pc** のドライバファクトリクラスを使用する必要があります。

1. local　：　クライアントローカルプリンタをプリントダイアログを表示し、選択・印刷します。
2. local_def：クライアントローカルデフォルトプリンタに直接印刷します。
3. local_pdf：クライアントでPDFを作成して、PDFバイナリデータを作成します。
4. pdf　：　サーバーでPDFファイルを作成し、URLをクライアントに渡します。
5. cont　：　連続した帳票をデータとして取り込みます。印刷はしません。
6. local_XXX: クライアントローカルプリンタをプリントダイアログを表示し、選択・印刷します。このとき、指定されたXXXという名称のプリンタをプリントダイアログを表示時に選択状態になっています。
7. LOCAL_DIRECT_XXX: クライアントローカルプリンタXXXに直接、印刷します。
8. 上記以外の名称はプリンタ名として使用されます。

その場合は、印刷APサーバーで使用可能なプリンタ名を指定する必要があります。但し、location宣言でlocalの指定時は、APサーバーではなくクライアントローカル接続プリンタが対象になります。

– キー：　「printer」固定
– 形式：　Key・値またはプリンタ名を含んだ文字列を持つセッションキー(@付)
– 記述例：

printer　帳票印刷用プリンタ1号

printプロセスのパラメータ（4）

- **session_value宣言**
  - 概要：印刷対象データを保持するServiceSessionキー名を指定します。
  - キー：session_value
  - 記述例：

    session_value　　印刷対象サービスセッションキー名,…

　印刷データは、Listで指定されたUpdatableで構成されているものとしますが、それ以外のString等の単一項目を示すサービスセッションを指定した場合は、全ページにその値がセットされるものとして扱います。先頭に@は付けないでください。

Printのリターン情報

印刷実行後に結果は、サービスセッションキーにセットされてきます。

- 「DownLoad_URL」セッションキー
  printer宣言がpdf時に作成されたPDFファイルのURLが格納される。

- 「DownLoad_FileSize」セッションキー
  printer宣言がpdf,local_pdf時に作成されたPDFファイル・イメージのバイトサイズが格納される。
  printer宣言がcont時は0が格納される。
  printer宣言が他のときはプラスの値が格納される。

- 「DownLoad_PDFDATA」セッションキー
  PDFでの作成時、PDFイメージのバイト列（pexa.share.util.cont.Binary）が格納される。

- 「PrinterName」セッションキー
  クライアントローカルプリンタ印刷後に使用したプリンタ名が格納される。

連続フォーム帳票印刷について

- PDFをブラウザに表示する場合、ブラウザ立上げに支障が発生したために連続して複数のフォームを同一のPDFファイルとして出力できるようにしました。

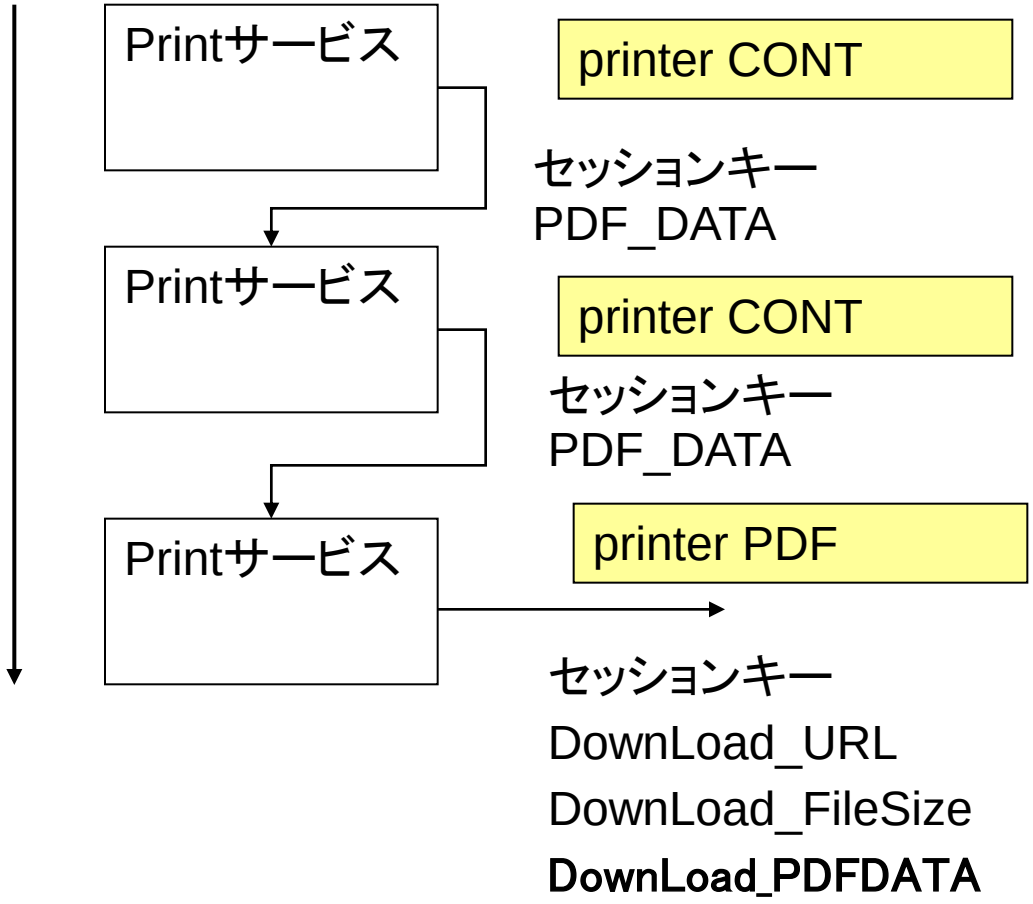
- Printer　CONTのPrintサービス呼び出しでPDFファイルを作成せず、Printデータの作成のみ行います。
- 作成されたデータをセッションキー“PDF_DATA”に格納して次のPrintサービスに伝えるようにしました。

- sheetの特定機能の追加

  “FLUSH”というsheetを特殊な意味で動作するようにしました。FLUSHシートはそれ自体では、印刷されず、すでにセッションキー“PDF_DATA”に格納されたデータのみを印刷するようにします。よって、不特定の複数帳票の印刷時の最後にこのシートを指定したPrintサービスを呼ぶ必要があります。また、FLUSH.printをのPrintサービスの環境に作成する必要があります。

  また、Printer　PDFを指定しても、いままでに連続した印刷データを連結したPDFを作成できます。

データの流れ


Printサービス
printer CONT
セッションキー
PDF_DATA
Printサービス
printer CONT
セッションキー
PDF_DATA
Printサービス
printer PDF
セッションキー
DownLoad_URL
DownLoad_FileSize
DownLoad_PDFDATA

FLUSH.printのサービス設定

Print定義は以下のようになります。フィールドセクション・
フォームファイルはダミーで設定値に意味を持ちません。

```
(print
    printType           jasper
    print_name          FLUSH
    formFile            FLUSH.jasper
    {field
        対象年月from        年月_FROM
    }
)
```

FLUSH.printのサービス設定

- FLUSHシートを持つPrintサービスは。

```
; 連続帳票フォームのPDFをすべて出力する。
(service
    service_name    S_PR_COMM_連続帳票フォーム全出力
    return_keys     DownLoad_URL,DownLoad_FileSize
    {process
        (印刷実行Flush
            format_type print
            (print
                sheet           FLUSH
                printer         PDF
            )
        )
    }
)
```

ページ関連パラメータ情報について

Parametersに以下の値を使用することができます。
連続帳票・単一帳票の両方で使用できます。

- Page1
  - 連続帳票の全帳票を単位とした帳票の先頭ページ番号
  - Variablesの$V{PAGE_NUMBER}と複合で使用します。
- AllPage1
  - 連続帳票の総ページ数
- Page2
  - 連続帳票の帳票単位とした帳票の先頭ページ番号
  - Variablesの$V{PAGE_NUMBER}と複合で使用します。
- AllPage2
  - 連続帳票の帳票単位の総ページ数
- PrintJobDate
  - 印刷日時を“yyyy/MM/dd HH:mm:ss“で示します。

JBOSS起動時の設定

JBOSS起動時には、PrintServiceMap設定ファイルのパス名を指定する必要があります。
Windowsの場合
set JAVA_OPTS=%JAVA_OPTS% -Dprint_entry_url=file:///%PRJHOME%¥env¥print¥print.entry
set JAVA_OPTS=%JAVA_OPTS% -Dprint_jndi=local_jndi_print_vm

Linuxの場合
JAVA_OPTS=${JAVA_OPTS}" -
Dprint_entry_url=file://${PRJHOME}/src/print/service/print.entry"
JAVA_OPTS=${JAVA_OPTS}" -Dprint_jndi=local_jndi_print_vm

print_entry_url
PrintServiceMap設定ファイルのパス名を指定（EARファイル起動時は、指定する必要はありません）
print_jndi
PrintサーバーのJNDI指定を設定
（プリンターサーバーを別のAPサーバーに持つ場合のみ指定します。）

jndi_list.xmlの設定

jndi_list.xmlに次の定義を追加します。
プリンタサーバーが主力APサーバーとは、別マシンのサーバーに搭載ときのみ設定する必要があります。(プリンタサーバが独立していない場合は必要ありません)

```
<!-- プリンタドライバ用イニシャルコンテキスト -->
 <jndi name="local_jndi_print" sync="false">
  <property>
    <entry key="java.naming.factory.initial"
     value="org.jnp.interfaces.NamingContextFactory" />
    <entry key="java.naming.provider.url" value="jnp://192.168.100.136:1099" />
    <entry key="java.naming.factory.url.pkgs"
     value="pexa.service.naming.delegate.client.cache:org.jboss.naming:org.jnp.interfaces" />
    <entry key="roda.jdbc.url" value="java:/pexaPool" />
    <entry key="print.spool.url" value="http://192.168.100.136:8088" />
    <entry key="print.svf.form.url" value="/src/print/service/svf/" />
  </property>
 </jndi>
```

iReportでの帳票作成

- 現行のPEXA4.7ではiReport v4.6を使用しています。

- *.jasperファイルの格納位置　　src/print/service/jasper
- 新規ドキュメントを選択し、以下の項目を設定します。
  - レポート名　formFile名（.jasper拡張子を除いた名称）
  - 用紙サイズ　A4 ...,B4 ...,LETTER,LEGAL...
  - 用紙方向　　縦長/横長
  - 余白の設定 上下左右の位置ごとに設定
  - タイトル･サマリページを別のページにする設定
  - 明細を複数の列で印刷する設定
- **ReportのLanguageは必ずJavaを選択してください。（注意）**

- IReportの位置やサイズ等の表現はPixelを単位としています。1Pixelは72DPIとなります。標準的なグリッドは10Pixelなので、約3.52mmに当たります。
  A4は297mm×210mm、11.7Inch×8.3Inchになります。
  フォントの1Pointは1/72Inch 約0.352mmに当たります。iReportの1Pixelとほぼ等しくなります。

iReportエレメント設定領域

印刷する領域（Band）を設定します。帳票のフォーマットにより、各Bandサイズを指定します。使用しないBandは高さを0に設定、またはBandを削除します。

領域	説明
Title	１単位印刷ヘッダにのみ印刷される領域
PageHeader	ページの上部に必ず印刷される領域
ColumnHeader	明細上部に必ず印刷される領域、通常、表のレポートのコラム名を含んでいるラベルは、このバンドに挿入されます。
Detail	明細部（データ行分連続して）印刷される領域（複数のdetailを指定することが可能です） DetailのBandのHeightは明細1行の高さを示します。
Summary	最終明細にサマリ（データは自分で作成）を印刷される領域
ColumnFooter	最終明細。サマリ下部に印刷される領域
PageFooter	１単位印刷の最終以外のフッタ部に印刷される領域（lastpagefooterがあるときは、1ページ出力時は出力されません。）
LastPagerFooter	１単位印刷フッタにのみ印刷される領域
Background	**透かし印刷領域**（最低部印刷内容を指定します。） 全頁のフレームなどの印刷をおこなうための領域

iReportページプロパティ

プロパティでチェックがあるときの下記のような出力を行います。

- Title on a new page
  TileBandを先頭に別ページとして出力します。

- Summary on a new page
  SummaryBandを最後尾に別ページとして出力します。

- Float column footer
  column footerBandをDetail/SummaryBandの直後に出力します。
  よって、明細の行数によって、出力する位置が変わります。
  チェックが無い時は、固定の位置に出力します。

iReportエレメントの設定

- 各領域（Band)にテキストフィールド・定型テキスト・画像・罫線・バーコード・SubReportなどを配置していきます。

- 各領域ごとに設定しますので、実際に印刷される帳票イメージそのままの設定が出来ません。

iReportでのSubReportの設定（メイン帳票へのSubReportエレメントのセット（1））

SubReportの名称をあらかじめ決めておきます。
基本的にPrintService設定ファイルでのフィールド情報の明細部分の名称（アンダースコア付き）になります。明細数分決めておきます。

- SubReportをセットするBandはTiTle、PageHeaderなど、どのBandでもかまいません。
- SubReport作成画面に移りますから、SubReportを作成します。
  - 次ページに基本的な設定方法が記述されています。
- メイン帳票のSubReportエリアの大きさを調整します。
- メイン帳票のSubReportのProperyを設定を確認します。
  - Subreport Expression に”$P{SUBREPORT_DIR}+メイン帳票の名称＋SubReportの名称”をセットされています。

  例：$P{SUBREPORT_DIR} + "Print_CMPOrderPrint_XSub2.jasper“
- Expression Class “java.lang.String”を選択されています。

iReportでのSubReportの設定（メイン帳票へのSubReportエレメントのセット（２））

- メイン帳票にSubReportエレメントをDrag&Dropを行い、SubReportを作成します。。
  1. SubReportWizardが表示されますので、ステップごとにセットします。
  2. Step1:Subreport　create a new reportのみをチェックします。
  3. Step2：Layout　一応、Blank A4を選択します。
  4. Step3:Query　Empty datasourceを選択します。
  5. Step4：Fields　設定しません。
  6. Step5:Group By　設定しません。
  7. Step6：Subreport exp
     1. Report Nameをセットします。名称はメイン帳票の名称＋SubReportの名称になります。（名称間に”_”アンダースコアがあることを確認してください）
     2. Store the directory name in a parameterをチェックします。
  8. Step7：Use a JRDatasource expressionをチェックします。したのテキストエリアに以下の文字列を記述します。赤色の部分は、SubReportの名称になります。

```
((pexa.ext.share.service.print.jasper.JRPexaDatasource)$P{REPORT_DATA_
    SOURCE}).subDataSource(" _XSub2")
```

iReportでのSubReportの設定（SubReport帳票の設定）

メイン帳票でSubReportを作成すると自動的にSubReport用の帳票設定用のファイルが作成されます。そのファイルを開いて、編集を行います。

1. あらかじめ、iReport左フレーム（Report Inspector）Fieldsに明細の表示フィールド項目となるという名称・データ属性（Stringなど)を追加しておく必要があります。

2. 帳票として使用するBandを決めます。
   1. 一般的にDetailと必要とあれば、ColumnHeader・ColumnFooterを使用します。それ以外を削除します。
   2. それぞれのBandの高さ・幅を指定します。これが明細の高さになります。

3. Detailには、明細情報に必要なフィールド情報をセットします。例えば、TextFieldを指定し、プロパティText Field Expression に”$F{製品名}”のように指定します。

4. ColumnHeader・ColumnFooterには、明細のカラム名等の情報を必要ならば、設定します。

iReportでのSubReportの設定（iReportでのClassPathの設定）

- iReportでSubReportを使用して、Pexaプリントサービスを使用する場合、Pexaのライブラリを一部使用するため、以下のようにiReportにClasspathをセットしてください。

- iReportのメニュー”Tool“→Option→iReportタブを選び→Classpathタブを選びます。
  - プロジェクトのXXXX/lib/pexa/pexaext-share.jarをAdd jarします。
  - XXXXは各自の環境によります。

SubReportを使用した PrintService設定ファイル

SubReportのある明細では、アンダースコアのある名称を付けてください
（下記では_XSub1,_XSub2）。
これが、iReportsのSubReportで使用する明細部の名称となります。

```
(print
      printType            jasper
      print_name           PR_Sub1
      formFile             SubRTest1.jasper
      {field
            DATE_yyyy/MM/dd:hh:mm:ssPRINT.入庫予定日
            {_XSub1
                  N1          PRINT.PrintDetailList.顧客名
            }
            {_XSub2
                  N3          PRINT.Pattern_itemNodeList.承認ユーザ名
                  N4          PRINT.Pattern_itemNodeList.顧客住所
            }
      }
)
```

PDFのフォント埋め込みについて

PDFフォント埋め込みを使用する場合、フォントファイルをAPサーバーでのクラスパスに格納する必要があります。また、導入するフリーフォントはそれぞれのマシンにフォント登録する必要があります。

APサーバー上のフォントファイルはPDF埋め込み用のグリフを取得するために格納します。

JasperReportsはjava.awt.フォントとして認識されているフォントを使用してPDFを作成しますので、フォントを格納するだけではなく、マシン（OS)としてフォント登録が必要となります。サーバーからの直接印刷時も同様の対応が必要です。

- Windowsの場合(Windows7)
  - *.ttfをクリックすると、フォント情報の画面が自動的に開きます。
  - 左上にインストールボタンが出ますので、押すとインストールされます。
  - コントロールパネルのフォントを開くとインストール済みのフォント情報が確認できます。

- Linuxの場合
  - *.ttfファイルを~/.fontsファイルの格納します。
  - 以下のコマンドでフォントキャッシュを更新します。

    ```
    fc-cache -fv
    ```

- AWTフォントとしての確認は、PexaToolsを使用します。(Pexa4.7以降)

  ```
  java -cp tool/lib/pexa/pexaext-tools.jar pexa.ext.tool.modelCheck.AwtFontList
  ```

PDF作成時のファイル削除について

PDF作成時にはAPサーバー上のPDFファイルの削除はバッチで必ずやる必要があります。

基本的にPDF表示時には、Web用の実PDFファイルを作成しますが、そのファイルを削除する部分が存在しません。Webに表示されているPDFファイルを即座に削除することはできません。そのため、PDFファイルを格納してあるフォルダに大量のファイルがたまってしまいます。

そのため以下のようなコマンドをCronで起動し、ファイルを削除します。

Linuxでのcrontabの削除シェル（ディレクトリは環境によって変更する必要があります）

```
#!/bin/bash
cd ~
if [ -e oldlog ]
then
        find oldlog -name "*.tar.gz" -mtime +30 -exec rm {} ¥;
else
        mkdir oldlog
fi
if [ -e olddata ]
then
        echo exist dir
else
        mkdir olddata
fi
find jboss4.2.3.GA/server/default/deploy/ROOT.war/print/pdf_spool -name "*.PDF" -mtime +14 -exec mv {} oldlog ¥;
XX=`date +%m%d`
cd oldlog
tar czf ../olddata/data$XX.tar.gz *
echo cron_end `date`
```

cronの設定です。（ディレクトリは環境によって変更する必要があります）

```
0 12 * * * sh /home/users/printsvr/mvrm.sh >> /home/users/printsvrcron.txt 2>&1
```