ANDREAS PROUZA

Leonard-Bernstein-Straße 8/2/17.07
1220 Wien / Österreich

Mail: andreas@prouza.at
Web: https://www.prouza.at

HIP

(Html Into PDF)

on IBM i

1 Überblick

Der Mail Client besteht aus 2 Teilen:

- RPG-Welt
  Das Ganze „rundherum“
  - Command für den Aufruf
  - Programme um die Daten in die Tabellen einzutragen
  - HIP Service welches die Liste zu generierenden PDFs abarbeitet und an das Java übergibt
  - Data queue Konzept zur Kommunikation mit dem HIP Service (Sync/Async)
- Java
  Die eigentliche Konvertierung ins PDF

2 Mindestanforderungen

Java 1.8

Der Java Client ist mit der Version 1.8 erstellt worden.

Sollte es nötig sein eine ältere Version zu unterstützen muss nur geprüft werden ob die Abhängigkeiten ebenfalls mit der älteren Version kompatible sind.

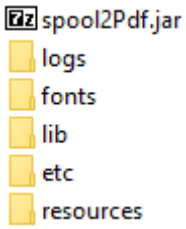
IBM i 7.2

3 Java

Im IFS wird ein Verzeichnis hinterlegt, mit allen nötigen Ordnern und Files.
Im Setup ist wird folgendes Verzeichnis angelegt:

/usr/prouza/hip

Dieses kann geändert werden. Dafür muss man lediglich auch den Property Eintrag in der Tabelle PROUZALIB/PROPERTIES anpassen (siehe 4.1.2 PROPERTIES)



- spool2Pdf.jar
  Das Java File welches die Klassen und Funktionen beinhaltet, die vom RPG-Part aufgerufen werden
- etc/
  Konfigurations Verzeichnis
  - log.properties
    Einstellungen für den Logger wo welche Logs geschrieben werden
  - pdf-fonts.properties
    Hier können Schriftarten definiert werden, die im HTML Template verwendet werden.

    font.[n]=[Schriftart Name];[Pfad+Name]

font.	Konstante
[n]	Fortlaufende Nummerierung
[Schriftart Name]	Name der Schriftart im CSS/HTML
[Pfad+Name]	Speicherort im IFS

    Beispiel:
    ```
    font.1=Courier New;./fonts/cour.ttf
    font.2=Arial;./fonts/arial.ttf
    ```
- lib/
  Hier befinden sich alle Java Files die für diese Applikation benötigt werden.
- logs/
  Das Log-Verzeichnis in dem die Log Files geschrieben werden.
  Dies wird im etc/log.properties definiert

- fonts/
  Hier können Schriftarten kopiert werden, die dann auch im pdf-fonts.properties entsprechend eingetragen werden können.
  Natürlich können auch Schriftarten aus anderen Verzeichnissen verwendet werden, wenn sie im pdf-fonts.properties File eingetragen werden.
- resources/
  Dieser Ordner wird von der Java Applikation verwendet um diverse Informationen für die Konvertierung zu speichern.

4 RPG-Welt

4.1 DB2

2 Tabellen werden verwendet:

4.1.1 SPOOL2PDF

Hier werden alle Requests für die Konvertierung zur weitere Verarbeitung eingetragen.

4.1.2 PROPERTIES

Eine allgemeine Tabelle für alle Applikationen in der PROUZALIB.
Hier werden diverse Settings für die entsprechende Applikation hinterlegt.

HIP_IFS_ROOT
Das Ausgangsverzeichnis der Java Applikation im IFS.
Dies kann beliebig geändert werden.

4.2 RPG

4.2.1 Direktaufruf (mit Spool File)

Die benötigte Prototyp Deklaration sieht wie folgt aus:

```
DCL-PR Spool2PDF EXTPGM ('SPL2PDFCL');
  P_IFS_HTMLTemplate char(400) const;
  P_IFS_PDFOutput    char(400) const;
  P_SpoolFile        char(10)  const;
  P_SpoolJob         char(26)  const; // z.B.: 'JOBA      USER1     471100'
  P_SpoolNr          char(6)   const;
  p_async            char(4)   const; // [*YES] / [*NO]
  p_waitTimeout      char(4)   const; // Sekunden
  p_ExterneReference char(50)  const;
END-PR;
```

4.2.2 Direktaufruf (mit IFS Files)

Die benötigte Prototyp Deklaration sieht wie folgt aus:

```
DCL-PR Spool2PDF EXTPGM ('SPL2PDFR');
  P_IFS_HTMLTemplate char(400)  const;
  P_IFS_PDFOutput    char(400)  const;
  P_IFS_File         char(400)  const;
  p_async            char(4)    const options(*nopass); // [*YES] / [*NO]
  p_waitTimeout      char(4)    const options(*nopass); // Sekunden
```

```
  p_ExterneReference char(50)   const options(*nopass);
END-PR;
```

4.2.3 HIP Service

Das HIP Service läuft im Batch als eigener Job (Endlosschleife).

Der Job kann aber mit ENDJOB und *CNTRLD einfach beendet werden.
Da der Job in periodischen Abständen prüft ob ein Ende-Signal übermittelt wurde (z.B. ENDJOB oder PWRDWNSYS).
Dadurch ist ein kontrolliertes Beenden des Jobs jederzeit möglich.

Meine Empfehlung für SBMJOB:

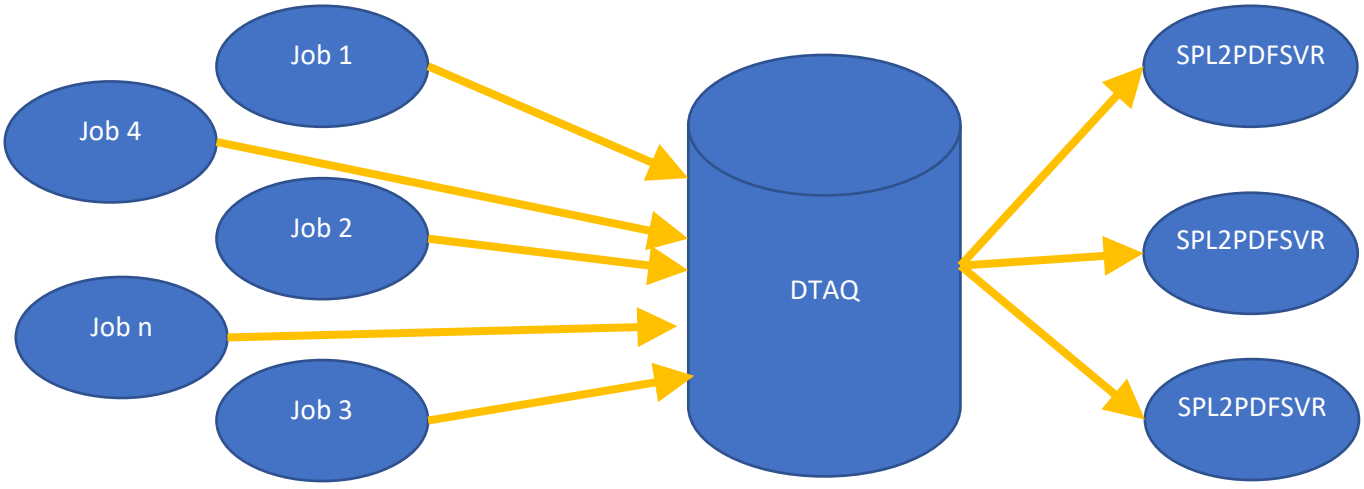
SBMJOB CMD(CALL PGM(PROUZALIB/SPL2PDFSVR)) JOB(SPL2PDFSVR) JOBQ(QS36EVOKE) CURLIB(PROUZALIB) JOBMSGQFL(*WRAP)

JOBQ
Hier verwende ich meistens die QS36EVOKE, da diese auf allen Maschinen immer aktiv ist und die JobQ mit maximaler Anzahl paralleler Jobs mit *NOMAX gesetzt ist.
Die QBATCH erlaubt per Default nur 1 Job aktiv und der ist oft schon besetzt.

JOBMSGQFL
Es werden periodisch JOBLOG Ausgaben getätigt. Dadurch sieht man was verarbeitet wurde und ob der Job noch arbeitet.
Da dieser Job permanent läuft kann es dadurch zu einem Überlauf kommen.
Mit *WRAP stelle ich sicher, dass bei einem Überlauf die älteren Einträge einfach gelöscht werden.

<u>Parallelisierung des HIP-Services</u>

Es können zur Laufzeit auch mehrere Jobs parallel gestartet werden. Da es Data Queue basiert ist, kann es beliebig jederzeit nach oben oder unten skaliert werden.



Soll heißen: Das SPL2PDFSVR Programm, ist so aufgebaut, dass es immer dieselbe Data Queue abarbeitet.

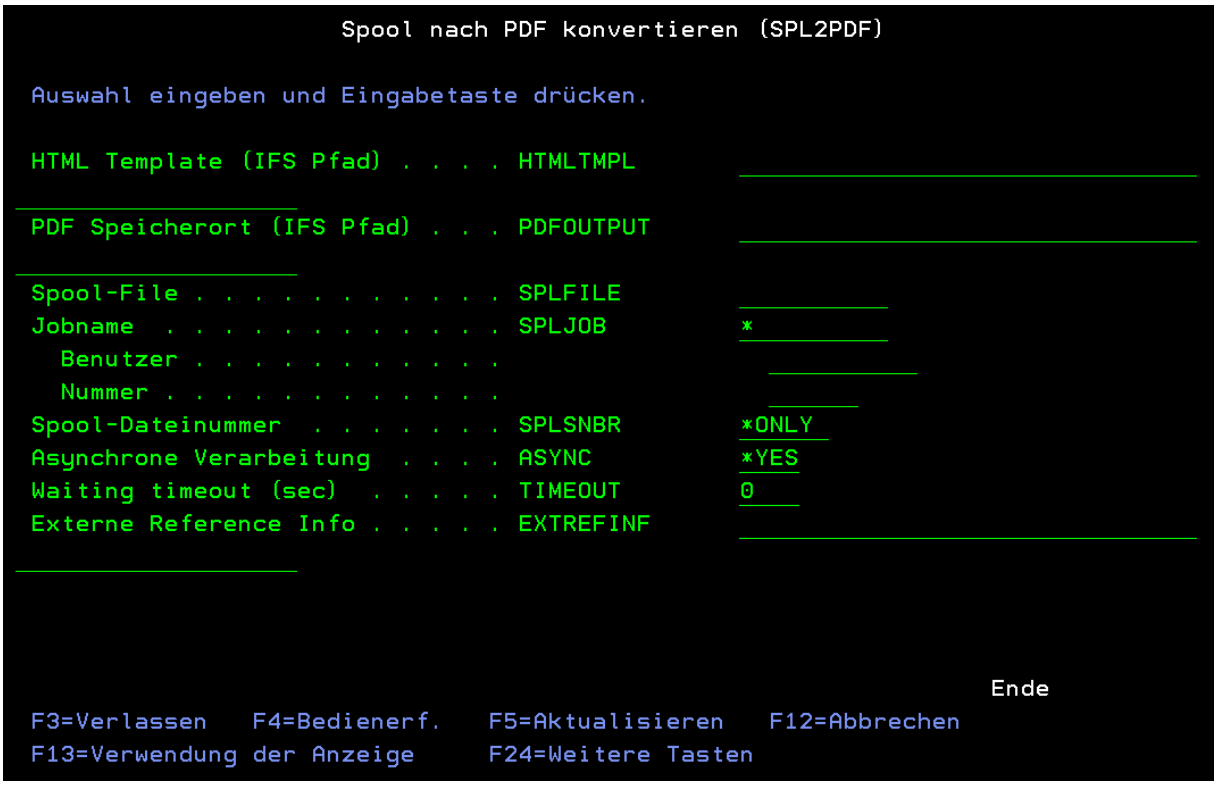
Werden mehrere Jobs via SBMJOB gestartet, greifen alle Jobs auf dieselbe Data Queue zu.

Dadurch kann die Verarbeitung der Data Queue (welche die zu verarbeitenden Id's beinhaltet) parallel abgearbeitet werden.

Dies kann gerade dann sinnvoll sein, wenn sehr viele Konvertierungen gleichzeitig in Auftrag gegeben werden.

Möchte man die Zahl der Jobs wieder dezimieren, so ist dies jederzeit einfach mit einem ENDJOB möglich.
Der Job beendet sich von selbst automatisch zum nächst möglichem Zeitpunkt. Z.B. sobald die aktuelle Verarbeitung/Konvertierung eines PDFs beendet wurde.

4.2.4 SPL2PDF Command



HTMLTMPL:

IFS Pfad & File Name des HTML Dokuments welches als Template verwendet werden soll.
In diesem Template wird der Inhalt des Spool Files eingebunden und als PDF erstellt.

PDFOUTPUT:

IFS Pfad & File Name wo das generierte PDF gespeichert werden soll.
Hinweis: Der Benutzer, unter dem das HIP Service läuft, muss für das Verzeichnis die nötigen Berechtigungen besitzen.

ASYNC

- *YES
  Es wird nicht auf Antwort vom HIP Service gewartet. Die Verarbeitung geschieht asynchron (parallel).
- *NO
  Der Command wird erst nach Erhalt einer Rückantwort vom HIP Service beendet.

Per Default ist *YES hinterlegt.

TIMEOUT

Bei einer synchronen Verarbeitung kann ein Timeout in Sekunden angegeben werden, falls keine Rückmeldung vom HIP Service kommen sollte.

EXTREFINF

Dieser Parameter dient dazu, in der Tabelle mit den Konvertier-Einträgen, eine eigene Interne Referenz, zur Nachvollziehbarkeit, hinterlegen zu können.

4.2.5 Beispiele

```
SPL2PDF HTMLTMPL('/home/prouza/template.html')
PDFOUTPUT('/home/prouza/output.pdf') SPLFILE(QPJOBLOG)
SPLJOB(*) EXTREFINF('Test 4711')
```