Jobs und deren Bedeutung

Grunddefinition

Mit dem Befehl FRM_INZ wird angegeben, ob das FRM Modul gestartet wird.
Hier wird auch definiert wie lange die einzelnen Jobs warten zwischen 2 Durchläufen.
Desweiteren können bis zu 3 verschiedene oxaionPrint Server damit definiert werden. Die Daten werden im File $$$PF1NI mit Funktion (FCTNNI) FRM gespeichert. Die IP-Adresse und der Port kann sowohl in Dezimalschreibweise als auch als Alias eingegeben werden.

Bitte auch den Hilfetext der einzelnen Parameter beachten.
Es wird nicht empfohlen die Werte direkt im File $$$PF1NI zu manipulieren.

Die folgenden Jobs werden automatisch (sofern mittels FRM_INZ definiert) im Subsystem SRV_SBS gestartet und müssen laufen, damit der reibungslose Ablauf des Druckens via oxaionPrint/oxaionForms und der Archivierung gewährleistet ist.

FRM_CRTFFV

Dieser Job erstellt die FFV Dateien im IFS der i5. Anhand der Definitionen im File $$ $PF1FS mit Funktion (FCTNFS) FRM wird bestimmt wie das Stammverzeichnis (DIREFS) heisst.
Die benötigten Einträge im $$$PF1FS sind:

- KEYJFS = FORMS für oxaionForms
- KEYJFS = PRINTxx (wobei xx = leer, 02, 03 sein kann) für oxaionPrint

Dem Stammverzeichnis wird noch der Name der Liste angehängt und die FFV Dateien werden da hinein gestellt. Anhand der Definitionen in FRMPF0PC Feld FPKZPC (FORMS/PRINT), wird entschieden ob via oxaionPrint bzw. oxaionForms gedruckt wird.

Das Programm FRMCLPJE ruft das Programm FRMPGPC2 auf. Das FRMPGPC2 löscht einmal täglich alle gedruckten Sätze, welche älter als 5 Tage sind, aus den zugehörigen temporären Dateien. Nach dem Löschen werden die Dateien reorganisiert. Das FRMPGPC2 schaut alle xx Sekunden (je nach Definition in FRM_INZ) ob nicht gedruckte Sätze in den temporären Dateien vorhanden sind.

FRM_CRTAE

Dieser Job archiviert bzw. mailt die PDF Dateien aus dem IFS der i5. Anhand der Definitionen im File $$$PF1FS wird bestimmt wie das Verzeichnis heisst.
Die benötigten Benutzer in KEYJFS sind:

- DISTRIBUTE in dieses Verzeichniss werden die PDF von oxaionPrint/oxaionForms gedruckt. Dieses Verzeichnis ist ein Muss
- ERROR in dieses Verzeichniss werden die PDF gestellt, welche nicht archiviert bzw. gemailt werden können. Dieses Verzeichnis ist ein muss
- TOEFS in dieses Verzeichnis werden die zu archivierenden PDF gestellt. Dieses Verzeichnis ist ein Muss.
- SAVDIR wenn dieses Verzeichnis vorhanden ist, dann kopiert die Archivierung alle archivierten PDF da hinein. Diese Verzeichnis ist ein Kann.

Das Programm FRMCLEFS ruft das Programm FRMPGEFS auf. Das FRMPGEFS löscht einmal täglich alle verarbeiteten Sätze, welche älter als 5 Tage sind, aus den zugehörigen temporären Dateien. Nach dem Löschen werden die Dateien reorganisiert. Das FRMPGEFS schaut alle xx Sekunden (je nach Definition in FRM_INZ) ob nicht verarbeitete Sätze in den temporären Dateien vorhanden sind.

FRM_SOCK01 - FRM_SOCK03

Zur Zeit sind maximal 3 Jobs möglich. Der Ablauf ist für jeden Job gleich.

Dieser Job stellt die TCP/IP Verbindung zum oxaionPrint her. Er liest die FFV Dateien aus dem definierten Verzeichnis und allen Unterverzeichnissen im i5 und sendet sie an oxaionPrint. Die Definitionen sind im $$$PF1FS mit Funktion FRM und KEYJFS = FORMS/PRINTxx (wobei xx = leer, 02, 03 sein kann) und im $$$PF1NI (siehe Grunddefinitionen) definiert. Wenn oxaionPrint die FFV Datei erfolgreich drucken konnte wird sie gelöscht.

Das Programm FRMCLPJ2 ruft das Programm FRMPGPC4 auf. Das FRMPGPC4 schaut alle xx Sekunden (je nach Definition in FRM_INZ) ob im IFS des i5 neue Dateien vorhanden sind.

Dateien und deren Bedeutung

Definitionsdateien

Dies ist nur eine Kurzbeschreibung der Dateien. Um die ganze Beschreibung der einzelnen Felder zu bekommen bitte via DSPFD FILE(Dateiname) die Dateibeschreibung ansehen.

FRMPF0PC (oxaion forms/print header)

Hier wird definiert ob via oxaionPrint bzw. oxaionForms gedruckt wird. Desweiteren können Informationen zur Schachtsteuerung, Editierung und Namensvergabe der FFV Dateien (Timestamp) eingegeben werden. Im Feld FCTNPDPC kann angegeben werden ob die Datei FRMPF0PD mit einem abweichenden Key gelesen wird. Anhand dem Stammverzeichnis aus $$$PF1FS (siehe oben) und dem Funktionsnamen FCTNPC wir das Verzeichnis zur Ablage der FFV Dateien gebildet. Im Feld FPFCTNPC kann ein abweichender Funktionsname angegeben werden. Hier wird auch gesteuert mit welchem oxaionPrint Server (FPKZPC) gedruckt wird. Die Werte können wiederum FORMS oder PRINTxx sein. Die Datei wird vom Programm FRMPGPC1 gebraucht.

FRMPF0PD (relation listtype to oxiaon forms type)

Hier wird definiert welcher Listenbereich zu welchem Listenbereich im oxaionPrint/oxaionForms gehört. Normalerweise ist dies eine 1:1 Beziehung. Beispiel für Definitionen: K10 für Kopf, D10 für Detail, E10 für Footer. K15-K34, D25-D44, E15-E34 sollten nicht gebraucht werden, denn die werden bei sich Loopenden K,D,E Feldern automatisch erstellt. Die Datei wird vom Programm FRMPGPC1 gebraucht.

FRMPF0PE (oxiaon forms/print outq)
Hier kann pro Outq eine Abweichung in der Schachtsteuerung eingetragen werden. Wenn die Outq nicht definiert ist, wird der Standard aus FRMPF0PC genommen. Die Datei wird vom Programm FRMPGPC1 gebraucht.

FRMPF0EH (header zu infostore connector)
Hier wird definiert ob ein Beleg archiviert bzw. gemailt wird. Falls ein Beleg archiviert wird, müssen auch die Angaben für das Archiv gepflegt werden. Um die Positionsdatei nicht für jeden Beleg zu füllen wurde die Referenz eingeführt. Im Feld Ausschluss können Dateien eingetragen werden, welche nicht verarbeitet werden dürfen. Die Datei wird vom Programm FRMPGPC1 gebraucht.

FRMPF0EP (pos zu infostore connector)
Hier wird definiert, wie die Zuordnung zum Archiv bzw. Mail auf Feldebene aussieht. Das heisst, welches gedruckte Feld einem Index des Archivs bzw. welches gedruckte Feld einem Mailkopffeld zugeordnet ist. Um das Mail korrekt versenden zu können, muss das Feld EMAFLDEP gepflegt sein. Die Datei wird vom Programm FRMPGPC1 gebraucht.

Temporäre Dateien

FRMPF0IH (index zu outputs header)
Kopfdatei zum archivieren bzw. Mailen von PDF Dateien. Falls eine PDF Datei nochmals archiviert bzw. gemailt werden muss, ohne dass man den Beleg neu drucken will kann man den Status STATIH auf ‚A' setzen. Dies funktioniert natürlich nur, wenn die PDF Datei noch im DISTRIBUTE Verzeichnis ist. Die Datei wird vom Programm FRMPGPC1 erstellt. Die Datei wird vom Job FRM_CRTAE gebraucht.

FRMPF0IP (index zu outputs pos)
Positionsdatei zum archivieren bzw. Mailen von PDF Dateien. Die Datei wird vom Programm FRMPGPC1 erstellt. Die Datei wird vom Job FRM_CRTAE gebraucht.

FRMPF0PH (temp file for print connector header)
Kopfdatei zur Erstellung der FFV Dateien. Falls eine FFV Datei nochmals erstellt werden muss, ohne dass man den Beleg neu drucken will kann man den Status STATPH auf ‚A' setzen. Die Datei wird vom Programm FRMPGPC1 erstellt. Die Datei wird vom Job FRM_CRTFFV gebraucht.

FRMPF0PP (temp file for print connector pos)
Positionsdatei zur Erstellung der FFV Dateien. Die Datei wird vom Programm FRMPGPC1 erstellt. Die Datei wird vom Job FRM_CRTFFV gebraucht.

FRMPF0PI (temp file for print connector stmf)
Steuerung der FFV Übergabe an den PrintConnector. Hier wird bestimmt, ob ein FFV als PDF gedruckt wird. Falls als PDF übergeben werden soll, wird in der Datei $$ $PF1FS mittels Funktion (FCTNFS) FRM und Key (KEYJFS) Outq (wobei nur gesucht wird, wenn Outq mit PDF beginnt) der UNC Pfad gesucht. Der Pfad muss vom oxaionPrint PC aus erreichbar sein. Die Datei wird vom Job FRM_SOCKxx gebraucht. Die Datei wird von FRM_CRTFFV geschrieben.

Generieren der temporären Dateien

FRMPGPC1

Das Programm generiert die temporären Dateien FRMPF0IH, FRMPF0IP, FRMPF0PH, FRMPF0PP. Als Input dienen die Dateien FRMPF0PC, FRMPF0PD, FRMPF0PE, FRMPF0EH, FRMPF0EP.

Das Programm wird für jedes zu druckende Feld mit dem Feldnamen und dem Feldinhalt aufgerufen. Anhand der Definitionen werden beim ersten Aufruf die Kopfdateien und die Positionsdateien bestückt. Beim folgenden Aufruf werden nur noch die Positionsdateien bestückt. Wenn alle zu druckenden Felder übergeben sind, dann ändert das Programm in den Kopfdateien den Status von inaktiv (I) auf aktiv (A).

Parameter für FRMPGPC1

Parameter	Erklärung	Länge	Dez.	Bemerkung
FCTN	Formularname	10		
KEYJ	Eindeutiger Key	255		Nr. des Dokuments z.B. Fakturanummer
OUTQ	Outq Name	10		Steuerung erfolgt in frdotq
OUT2	Outq Name2	10		Steuerung erfolgt in frdotq
COPY	Anzahl Kopien	6	0	Anz. Kopien für OUTQ
LSTA	Listenart	3		Definition in FRDPF0PD
FELD	Feldname	10		Gemäss I-Report
LOOP	Loopindikator	1		J/N
FART	Feldart	1		A=Alphanumerisch N=Numerisch D=Datum T=Zeit B=Blank N & D Felder werden anhand der Definitionen in FRMPF0PC editiert
FLDL	Feldlänge	5	0	Immer füllen
FLDD	Anzahl Dezimalen	2	0	Auf Dezimalstellen wird gerundet
INHAN	Inhalt Numerisch	15	5	Mussfeld wenn FART N/D/T
INHAA	Inhalt Alphanumerisch	256		Mussfeld wenn FART A/B
PosCount	Zeilenzähler	6	0	
WhatToDo		10		*NEWPRINT/*END/*CPY/*ACTCPY
Archiv	Folgeverarbeitung	1		N=keine Verarbeitung J=Infostore Anbindung E=per e-mail versenden

				1=Infostore & e-mail wenn nicht N dann müssen die Definitionen in FRMPF0EH und FRMPF0EP erstellt werden
Copy2	Anzahl Kopien	6	0	Anz. Kopien für OUT2
FldMOnly	Nur e-mail Feld	1		0=normales Feld 1=nur e-mail Feld wenn ein nur e-mail Feld mitkommt, wird kein Satz in FRMPF0PP geschrieben
OUTCopy	Outq Name für die Kopien (Orginal geht dann an OUTQ und wenn mehr als eine Kopie, dann gehen die an OUTCopy)	10		Steuerung erfolgt in frdotq

Um einen neuen Druck zu starten muss das FRMPGPC1 mit WhatToDo *NEWPRINT und ansonsten leerem Inhalt aufgerufen werden. Dies bewirkt, dass alle internen Variablen initialisiert werden. Dann bitte den Inhalt des WhatToDo nicht mehr ändern bis alles gedruckt wurde. Am Schluss muss das FRMPGPC1 noch mit *END aufgerufen werden, damit es weiss dass nun alles da ist. Wenn der Schluss mit *CPY anstatt *END aufgerufen wird, dann wird nur eine Kopie (Orginal) mit dem Status 'A' versehen. Wenn mehrere Kopien zu drucken waren, werden die mit dem KEYJ Zusatz CPY in die Datenbanken kopiert (ohne Orginal) und die Schächte der Kopie genommen. Damit auch die aktiviert werden, muss das FRMPGPC1 nochmals mit *ACTCPY aufgerufen werden.

LOOP muss ein J sein, wenn das gleiche Feld mehrfach hintereinander kommt. Zum Beispiel wenn Text gedruckt wird.

PosCount muss pro LSTA um eins erhöht werden. Bei Detailzeilen (Dxx) muss pro Detailzeile der PosCount um eins erhöht werden.

Das Feld OUTQ und OUT2 kann auch leer sein. Wenn Beide leer sind, dann wird nichts gedruckt.

Synchronisieren der Verarbeitung

FRMPGPC5

Das Programm kann die Verarbeitung vom FRM_CRTFFV und FRM_SOCK01 – FRM_SOCK03 stoppen bzw. Starten um den Druck synchron zu bringen.

Falls ein Druckprogramm unbedingt alles synchron drucken muss, muss das FRMPGPC5 mit dem Parameter '0' zu Beginn aufgerufen werden. Dann können alle Daten via FRMPGPC1 gedruckt werden. Am Schluss muss das FRMPGPC5 mit dem Parameter '1' nochmals aufgerufen werden, damit die Verarbeitung der gestoppten FRM_* Jobs weitergeht.

Templates bzw. Formate

Einige Hinweise

Zum Verständnis gehen wir von einem Formularnamen BSP aus.

Alle Sourcefiles sind **jxml-Files**. Sie werden bei Erstellung des Formulars automatisch als **jasper.File**s generiert.

BSP_K und BSP_D sind Metatemplates welche im wesentlichen die Seitenrandmasse enthalten. BSP_S kann ignoriert werden.

BSP_VK10 enthält die Textmarken bzw. Textfelder des Formular-Headers.

BSP_VD10 enthält die Textfelder des Detailteils.

BSP_E10 enthält die Textfelder des Footers.

BSP_VD10 ist ein sich wiederholender Bereich. Der Kopf BSP_VK10 wird im Prinzip nur einmal auf der ersten Seite gedruckt.

Achtung:
Will man die Felder des Kopfteil auf jeder Seite wiederholen, so müssen die Felder im VK10 definiert werden. Jedoch die Textmarken müssen im Metatemplate BSP_D „gezeichnet“ werden und die Felder als sog. Parameterfelder definiert und referenziert werden.

BSP_D25 ist wie BSP_VE15 ein sich widerholendes Format. Zur Erinnerung: in der Schnittstelle ist hierfür die Listenart D10 bzw. E10 mit LOOPPP = J zu übergeben.

Hintergrundbilder für das ganze Formular ist im Metatemplate BSP.jxml zu definieren: Doppelklick in Template und unter Lasche Image ist mit Funktion "new java.io.File" die Lokation des Image (jpg.etc.) zu hinterlegen.

Achtung:
Nach Änderung von Templates müssen folgende **jasper.Files** im Templateordner **manuell gelöscht** werden:BSP.jasper und das jeweilige geänderte Template, also zB. BSP_VD10. Diese werden beim nächsten Formulardruck automatisch wieder erstellt.
Dies ist seit oxaionPrint Version 7.1 nicht mehr nötig.

Kopieren von Templates zwecks Neuerstellung von Templates

- ➔ Kopieren einen ganzen Templateordner unter neuen Formularnamen
- ➔ Entfernen sie nicht benötigte Subtemplates
- ➔ Umbenennen aller übrigen Subtempates
- ➔ Kopieren des Templateseintrages in Control.xml im Ordner \oxaion\print\conf und ändern der Templatenamen. Beispiel: Bereich vk20201p kopieren in BSP und ändern der XML-Zeilen die vk20201p enthalten.
- ➔ Mit iReport sind alle Verweise auf das alte Template zu ändern
  Vorallem: Menupunkt Projekt ->Projektoptionen, Projektname und Lasche i18n

Schachtsteuerung

Feld $P{OTQ8} ist 'J' wenn ein Einzahlungsschein (Satz in FRMPF0PD mit EZKZPD = 'J') im Formular vorhanden ist. Dies funktioniert zur Zeit nur wenn der EZ als letzte Seite gedruckt wird. Um den EZ auf einer separaten Seite zu drucken muss am Anfang des Templates ein neues Textfeld angelegt werden. Der Inhalt ist: new String(““). Beim Hyper Link muss noch new String(“DOC_START_ANCHOR“) angegeben werden.
Folgeseite anders als erste Seite wird wie der Schacht des EZ im FRMPF0PC definiert.

Einige Hinweise zum senden von e-Mails

Die Hauptdefinition ist in FRMPF0EPerklärt.

Es können jedoch diverse Felder als e-Mail Felder zugeordnet werden.

- FRMADR e-Mail Adresse des Absenders (Mussfeld)
- FRMNAME Names des Absenders (Kannfeld)
- TOADR e-Mail Adresse des Empfängers (Mussfeld)
- TONAME Name des Empfängers (Kannfeld)
- CCADR1 CC e-Mail Adresse (Kannfeld)
- CCNAME1 CC Name (Kannfeld)
- CCADR2 CC e-Mail Adresse (Kannfeld)
- CCNAME2 CC Name (Kannfeld)
- CCADR3 CC e-Mail Adresse (Kannfeld)
- CCNAME3 CC Name (Kannfeld)
- SUB Subjekt (Mussfeld)
- SUB2 Subjekt Teil2 (Kannfeld)
- SUB3 Subjekt Teil3 (Kannfeld)
- TEXT Text im Mail (Kannfeld)

Wenn mehrere SUB vorkommen, werden sie vor dem versenden zusammengeführt und als Subjekt ins e-Mail geschrieben. Der TEXT ist als sich wiederholendes Feld gemeint und kann zur Zeit maximal 100 Linien an 256 Zeichen beinhalten. Ein \n im Text wird als zwingender Zeilenumbruch behandelt und auch so im e-Mail verarbeitet.