

IN09 – Department of Health Interface Design

January 24, 2011



Table Of Contents

1.1	INTRODUCTION	3
1.2	INTERFACE PROCESSES	4
1.2.1	Interface <i>CRUD Matrix</i>	13
1.3	REQUIREMENTS.....	14



1.1 Introduction

The Florida Department of Health (DOH), Vital Statistics/Birth Registration system, currently sends the Florida Department of Children and Families (DCF) a weekly incremental extract of vital statistics/birth certificate data. The extract file contains the birth activity recorded in Florida for the previous seven (7) days. The data is currently loaded to DCF's CIS system and is available for search by approved Florida Safe Families Network (FSFN) users for validating child demographic data captured and documented in FSFN cases. This is a one-way interface between FSFN and DOH. FSFN co-shares the vital statistics extract with ECI. An import process is utilized to load the vital statistics data to a set of defined data structures in FSFN. The FSFN on-line application provides users with the capability to search the vital statistics data. Once the worker searches and identifies the respective birth record, the case worker will manually validate the child's person management demographic information and the application will provide the ability to link the specific birth certificate record to the child's person management record for later retrieval and review. Additionally, the application maintains an audit trail of all searches against the birth certificate records for reporting back to DOH.

Data Exchange - The Department of Health vital statistics extract file will be sent on a weekly basis via a batch process and co-shared between DCF and CGI for loading to ECI and FSFN. A set of destination OLTP database tables in FSFN stores this data. The import process has the ability to add and/or update existing data in these tables. A one-time mass conversion performed a full load, which comprised of approximately 13 million birth records; the weekly extract is comprised of only the incremental data from the previous 7 days.

Frequency of Interface – The batch process will run on a weekly basis, currently sent and loaded on Fridays.

Interchange Mode – The Business Objects tools suite is the primary tool used to construct the Extract, Transform, and Load processing job to import/load the file directly to the FSFN OLTP database tables. The file is sent via FTP from the Department of Health system to a secure FTP server within the DCF domain. Both DCF and CGI will have authorized user IDs for accessing the file. Once the file is received from DOH, an Autosys job executes a batch process to ftp the file to the FSFN middle tier environment where the load routine processes and loads the contents of the file to the FSFN OLTP database tables.



1.2 Interface Processes

Process Overview

The Department of Health, vital statistics extract file is sent on a weekly basis via a batch process and shared between DCF and CGI for loading to ECI and FSFN. FSFN OLTP database tables store this data. The data integrator ETL process will utilize an ID with the authorization and rights to add and/or update existing data in the defined OLTP tables.

Extract Layout:

DOH, Vital Statistics Extract Layout	
Column Name	Data Type
BIRTH_ID	10, numeric
CHILD_1ST_N	varchar(80)
CHILD_MID_N	varchar(80)
CHILD_LST_N	varchar(80)
CHILD_SUFN_N	varchar(6)
CHILD_SSN_R	9, numeric
CHILD_DOB_D	Date
CHILD_BIRTH_WGT_GRAMS_R	4, numeric
CHILD_SEX_C	char(1)
CHILD_BIRTH_STATE	varchar(40)
CHILD_BIRTH_COUNTRY	varchar(40)
CHILD_BIRTH_FL_CNTY_C	varchar(40)
CHILD_BIRTH_FL_CITY_N	varchar(40)
CHILD_BIRTH_FAC_TYP_N	varchar(30)
CHILD_BIRTH_FAC_TYP_C	char(1)
CHILD_BIRTH_FAC_N	varchar(50)
CHILD_BIRTH_FAC_ID	5, numeric
MOTHER_1ST_N	varchar(40)
MOTHER_MID_N	varchar(40)
MOTHER_LST_N	varchar(40)
MOTHER_MAIDEN_N	varchar(40)
MOTHER_SSN_R	9, numeric



Florida Safe Families Network

MOTHER_DOB_D	Date
MOTHER_BIRTH_STATE_X	varchar(40)
MOTHER_BIRTH_COUNTRY_X	varchar(40)
MOTHER_MARRIED_C	char(1)
MOTHER_RACE_WHITE_F	Bit
MOTHER_RACE_BLACK_F	Bit
MOTHER_RACE_AMER INDIAN_F	Bit
MOTHER_RACE_AMER_IND_X	varchar(25)
MOTHER_RACE_ASIAN_IND_F	Bit
MOTHER_RACE_CHINESE_F	Bit
MOTHER_RACE_FILIPINO_F	Bit
MOTHER_RACE_JAPANESE_F	Bit
MOTHER_RACE_KOREAN_F	Bit
MOTHER_RACE_VIETNAMESE_F	Bit
MOTHER_RACE_ASIAN_OTH_F	Bit
MOTHER_RACE_ASIAN_X	varchar(25)
MOTHER_RACE_HAWAIIAN_F	Bit
MOTHER_RACE_GUAM_F	Bit
MOTHER_RACE_SAMOAN_F	Bit
MOTHER_RACE_PAC_OTHER_F	Bit
MOTHER_RACE_PAC_X	varchar(25)
MOTHER_RACE_OTHER_F	Bit
MOTHER_RACE_OTHER_X	varchar(25)
MOTHER_RACE_UNK_F	Bit
MOTHER_HISP_F	Bit
MOTHER_ADDR_RES_STATE_C	varchar(40)
MOTHER_ADDR_RES_COUNTRY_X	varchar(40)
MOTHER_ADDR_RES_COUNTY_N	varchar(40)
MOTHER_ADDR_RES_MUN	varchar(40)
MOTHER_ADDR_RES_STREET_X	varchar(50)
MOTHER_ADDR_RES_ZIP_C	5, numeric
MOTHER_ADDR_MAIL_STATE_C	varchar(40)
MOTHER_ADDR_MAIL_COUNTRY_N	varchar(40)



Florida Safe Families Network

MOTHER_ADDR_MAIL_CITY_N	varchar(40)
MOTHER_ADDR_MAIL_STREET_ADDR1_X	varchar(50)
MOTHER_ADDR_MAIL_ZIP_C	9, numeric
FATHER_1ST_N	varchar(40)
FATHER_MID_N	varchar(40)
FATHER_LST_N	varchar(40)
FATHER_SUFN_N	varchar(6)
FATHER_SSN_R	9, numeric
FATHER_DOB_D	Date
FATHER_BIRTH_STATE_C	varchar(40)
FATHER_BIRTH_COUNTRY_X	varchar(40)
FATHER_RACE_WHITE_F	Bit
FATHER_RACE_BLACK_F	Bit
FATHER_RACE_AMER INDIAN_F	Bit
FATHER_RACE_AMER_IND_X	varchar(25)
FATHER_RACE_ASIAN_IND_F	Bit
FATHER_RACE_CHINESE_F	Bit
FATHER_RACE_FILIPINO_F	Bit
FATHER_RACE_JAPANESE_F	Bit
FATHER_RACE_KOREAN_F	Bit
FATHER_RACE_VIETNAMESE_F	Bit
FATHER_RACE_ASIAN_OTH_F	Bit
FATHER_RACE_ASIAN_X	varchar(25)
FATHER_RACE_HAWAIIAN_F	Bit
FATHER_RACE_GUAM_F	Bit
FATHER_RACE_SAMOAN_F	Bit
FATHER_RACE_PAC_OTHER_F	Bit
FATHER_RACE_PAC_X	varchar(25)
FATHER_RACE_OTHER_F	Bit
FATHER_RACE_OTHER_X	varchar(25)
FATHER_RACE_UNK_F	Bit
PATERNITY_TYPE_C	1, numeric
PATERNITY_D	Date



ADOPT_TYP_C	varchar(25)
FATHER_ADDRRES_STATE	varchar(40)
FATHER_ADDRRES_COUNTRY	varchar(40)
FATHER_ADDRRES_CITY	varchar(40)
FATHER_ADDRRES_ADDR1	varchar(50)
FATHER_ADDRRES_ZIP	9, numeric
COURT_JURISDICTION	
DIRECTOR_F	Bit
FEE_DUE_F	Bit
PROSPECT_ADOPT_F	Bit
RECORD_REVIEW_F	Bit
DELAYED_FLAG_F	Bit
CHILD_MISSING_F	Bit
DEATH_OCCURRED_F	Bit
DEATH_D	Date
DEATH_STATE_N	varchar(40)
DEATH_COUNTRY_N	varchar(40)
DEATH_SFN	10, numeric
CREATE_D	Date
MODIFY_D	Date
POST_REG_STATUS_N	varchar(20)
POST_REG_STATUS_C	char(1)

DOH code translations:

DOH, Vital Statistics Extract Layout	
Column Name	Data Type
CHILD_SEX_C	char(1)
CHILD_BIRTH_FAC_TYP_C	char(1)
MOTHER_MARRIED_C	char(1)
PATERNITY_TYPE_C	1, numeric
POST_REG_STATUS_C	char(1)



FSFN Table Structures:

```
CREATE TABLE BR_ADDRESS
(
  ID_BR_ADDRESS INTEGER PRIMARY KEY not null,
  AD_ST VARCHAR(40),
  AD_CNTRY VARCHAR(40),
  AD_CITY VARCHAR(40),
  AD_ADDR1 VARCHAR(50),
  AD_ZIP DECIMAL(9)
);
```

```
CREATE TABLE BR_DOH_BIRTH_RECORD
(
  ID_BIRTH DECIMAL(10) PRIMARY KEY not null,
  ID_DOB INTEGER,
  DT_DOB VARCHAR(10),
  FL_VALID_DT_DOB CHAR(1),
  ID_SSN VARCHAR(9),
  QT_BR_WGT_GRMS DECIMAL(4),
  FL_MTHR_MARRD CHAR(1),
  CD_PTRNTY_TYPE INTEGER,
  DT_PATERNITY DATE,
  TX_ADOPT_TYP VARCHAR(25),
  TX_CRT_JRSDCTN VARCHAR(40),
  FL_DIRECTOR CHAR(1),
  FL_FEE_DUE CHAR(1),
  FL_PROSPECT_ADOPT CHAR(1),
  FL_RECORD_REVIEW CHAR(1),
  FL_DELAYED_FLAG CHAR(1),
  FL_CHILD_MISSING CHAR(1),
  FL_DEATH_OCCURRED CHAR(1),
  DT_DEATH VARCHAR(10),
```




```
FL_VALID_DT_DEATH CHAR(1),
TX_DEATH_ST VARCHAR(40),
TX_DEATH_CNTRY VARCHAR(40),
DEATH_SFN DECIMAL(10),
DT_CREATE DATE,
DT_MODIFY DATE,
TX_POST_REG_STS VARCHAR(20),
CD_POST_REG_STS CHAR(1),
NM_FRST VARCHAR(80),
NM_MDL VARCHAR(80),
NM_LST VARCHAR(80),
NM_SFX VARCHAR(6),
CD_BR_SEX CHAR(1),
TX_BR_STATE VARCHAR(40),
TX_BR_CNTRY VARCHAR(40),
TX_BR_FL_CNTY VARCHAR(40),
TX_BR_FL_CITY VARCHAR(40),
TX_BR_FACLTYP_TYP VARCHAR(30),
CD_FACLTYP_TYP CHAR(1),
NM_BR_FACLTYP VARCHAR(50),
ID_BR_FACLTYP INTEGER,
NM_UPR_FRST VARCHAR(80),
NM_UPR_LST VARCHAR(80),
NM_UPR_MDL VARCHAR(80),
ID_CR INTEGER not null,
TS_CR TIMESTAMP not null,
ID_UP INTEGER not null,
TS_UP TIMESTAMP not null,
ETL_INS_TS TIMESTAMP not null,
ETL_UPD_TS TIMESTAMP not null
);
ALTER TABLE BR_DOH_BIRTH_RECORD
ADD CONSTRAINT XFKBR
FOREIGN KEY (ID_DOB)
```



```
REFERENCES DATE_DIM(ID_DATE);
CREATE INDEX XFK1BR ON BR_DOH_BIRTH_RECORD(ID_DOB);

CREATE TABLE BR_PERSON_FATHER
(
  ID_PERSON_FATHER DECIMAL(10) PRIMARY KEY not null,
  ID_BR_ADDRESS INTEGER,
  ID_DOB INTEGER,
  DT_DOB VARCHAR(10),
  FL_VALID_DT_DOB CHAR(1),
  ID_SSN VARCHAR(9),
  NM_FRST VARCHAR(80),
  NM_MDL VARCHAR(80),
  NM_LST VARCHAR(80),
  NM_SFX VARCHAR(6),
  FL_RACE_WHITE CHAR(1),
  FL_RACE_BLACK CHAR(1),
  FL_RACE_AMER_IND CHAR(1),
  TX_RACE_AMER_IND VARCHAR(25),
  FL_RACE_ASIAN_IND CHAR(1),
  FL_RACE_CHINESE CHAR(1),
  FL_RACE_FILIPINO CHAR(1),
  FL_RACE_JAPANESE CHAR(1),
  FL_RACE_KOREAN CHAR(1),
  FL_RACE_VIETNAMESE CHAR(1),
  FL_RACE_ASIAN_OTH CHAR(1),
  TX_RACE_ASIAN VARCHAR(25),
  FL_RACE_HAWAIIAN CHAR(1),
  FL_RACE_GUAM CHAR(1),
  FL_RACE_SAMOAN CHAR(1),
  FL_RACE_PAC_OTHER CHAR(1),
  TX_RACE_PAC VARCHAR(25),
  FL_RACE_OTHER CHAR(1),
  TX_RACE_OTHER VARCHAR(25),
```



```
FL_RACE_UNK CHAR(1),
BR_BIRTH_STATE CHAR(18),
BR_BIRTH_COUNTRY VARCHAR(20),
NM_UPR_FRST VARCHAR(80),
NM_UPR_LST CHAR(80),
FL_RACE_HISP CHAR(1),
NM_UPR_MDL VARCHAR(80),
ETL_INS_TS TIMESTAMP not null,
ETL_UPD_TS TIMESTAMP not null
);
ALTER TABLE BR_PERSON_FATHER
ADD CONSTRAINT XFK1BRFTHR
FOREIGN KEY (ID_BR_ADDRESS)
REFERENCES BR_ADDRESS(ID_BR_ADDRESS);
ALTER TABLE BR_PERSON_FATHER
ADD CONSTRAINT XFK2BRFTHR
FOREIGN KEY (ID_DOB)
REFERENCES DATE_DIM(ID_DATE);
ALTER TABLE BR_PERSON_FATHER
ADD CONSTRAINT XFK3BRFTHR
FOREIGN KEY (ID_PERSON_FATHER)
REFERENCES BR_DOH_BIRTH_RECORD(ID_BIRTH);
CREATE INDEX XFK1BRFTHR ON BR_PERSON_FATHER(ID_BR_ADDRESS);
CREATE INDEX XFK2BRFTHR ON BR_PERSON_FATHER(ID_DOB);

CREATE TABLE BR_PERSON_MOTHER
(
  ID_PERSON_MOTHER DECIMAL(10) PRIMARY KEY not null,
  NM_FRST VARCHAR(80),
  NM_MDL VARCHAR(80),
  NM_LST VARCHAR(80),
  NM_MAIDEN VARCHAR(80),
  ID_SSN VARCHAR(9),
  FL_RACE_WHITE CHAR(1),
```



FL_RACE_BLACK CHAR(1),
FL_RACE_AMER_IND CHAR(1),
TX_RACE_AMER_IND VARCHAR(25),
FL_RACE_ASIAN_IND CHAR(1),
FL_RACE_CHINESE CHAR(1),
FL_RACE_FILIPINO CHAR(1),
FL_RACE_JAPANESE CHAR(1),
FL_RACE_KOREAN CHAR(1),
FL_RACE_VIETNAMESE CHAR(1),
FL_RACE_ASIAN_OTH CHAR(1),
TX_RACE_ASIAN VARCHAR(25),
FL_RACE_HAWAIIAN CHAR(1),
FL_RACE_GUAM CHAR(1),
FL_RACE_SAMOAN CHAR(1),
FL_RACE_PAC_OTHER CHAR(1),
TX_RACE_PAC VARCHAR(25),
FL_RACE_OTHER CHAR(1),
TX_RACE_OTHER VARCHAR(25),
FL_RACE_UNK CHAR(1),
FL_HISP CHAR(1),
ID_DOB INTEGER,
DT_DOB VARCHAR(10),
FL_VALID_DT_DOB CHAR(1),
ID_BR_ADDR_MAIL INTEGER,
ID_BR_ADDR_RES INTEGER,
TX_BR_STATE VARCHAR(20),
TX_BR_COUNTRY VARCHAR(20),
NM_UPR_FRST VARCHAR(80),
NM_UPR_LST VARCHAR(80),
NM_UPR_MAIDEN VARCHAR(80),
NM_UPR_MDL VARCHAR(80),
ETL_INS_TS TIMESTAMP not null,
ETL_UPD_TS TIMESTAMP not null
);



```
ALTER TABLE BR_PERSON_MOTHER
ADD CONSTRAINT XFK1BRMTHR
FOREIGN KEY (ID_BR_ADDR_MAIL)
REFERENCES BR_ADDRESS(ID_BR_ADDRESS);
ALTER TABLE BR_PERSON_MOTHER
ADD CONSTRAINT XFK2BRMTHR
FOREIGN KEY (ID_BR_ADDR_RES)
REFERENCES BR_ADDRESS(ID_BR_ADDRESS);
ALTER TABLE BR_PERSON_MOTHER
ADD CONSTRAINT XFK3BRMTHR
FOREIGN KEY (ID_DOB)
REFERENCES DATE_DIM(ID_DATE);
ALTER TABLE BR_PERSON_MOTHER
ADD CONSTRAINT XFK4BRMTHR
FOREIGN KEY (ID_PERSON_MOTHER)
REFERENCES BR_DOH_BIRTH_RECORD(ID_BIRTH);
CREATE INDEX XFK1BRMTHR ON BR_PERSON_MOTHER(ID_BR_ADDR_MAIL);
CREATE INDEX XFK2BRMTHR ON BR_PERSON_MOTHER(ID_BR_ADDR_RES);
CREATE INDEX XFK3BRMTHR ON BR_PERSON_MOTHER(ID_DOB);
```

FSFN ETL Functions:

- Ability to process full load
- Ability to process incremental loads including add and update functions
- The flow of the ETL as it iterates the DOH file will be to create the master entry in the BR_DOH_BIRTH_RECORD table followed by the row entered in the BR_PERSON_MOTHER and the associated address in BR_ADDRESS followed by the row entered in the BR_PERSON_FATHER and associated address in BR_ADDRESS.
- Voids

1.2.1 Interface CRUD Matrix



Table Name	CRUD
BR_DOH_BIRTH_RECORD	C, R, U
BR_PERSON_MOTHER	C, R, U
BR_PERSON_FATHER	C, R, U
BR_ADDRESS	C, R, U

1.3 Requirements

- BAFO – 001