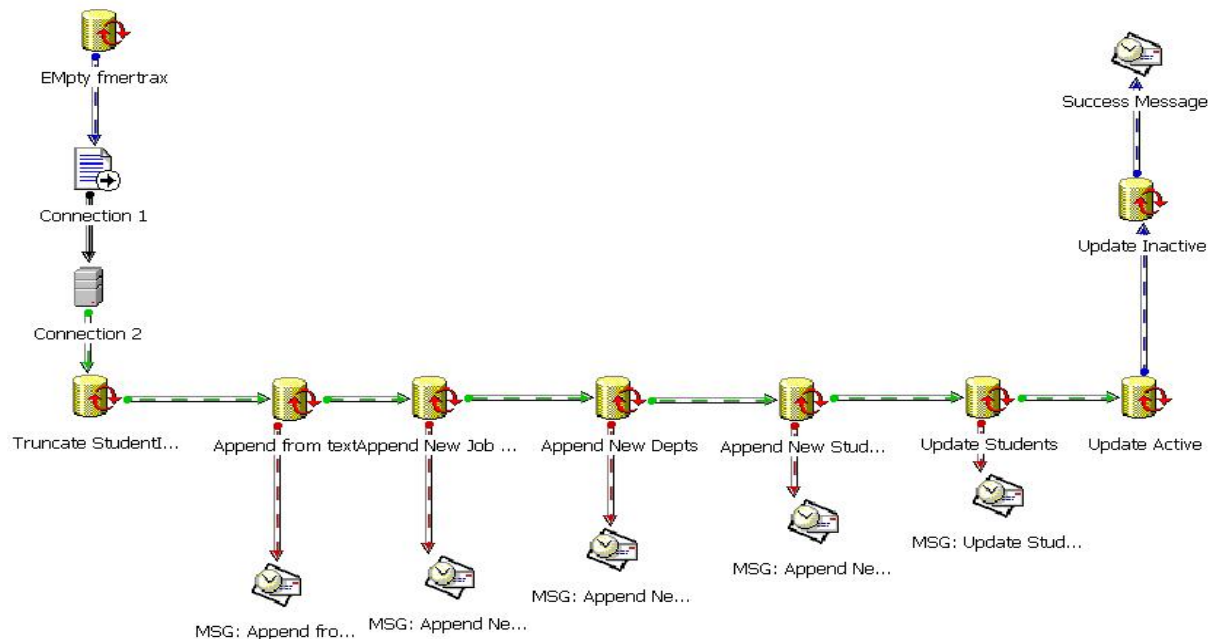


LMS IPPS Update

DTS Diagram



Empty fmertrax

TRUNCATE TABLE fmertrax

Connection 1

Text File (Source)

File Name: \\redacted\apps\$\LMS\Production\Data\fmertrax.txt

Connection 2

Server: redacted

Database: lms

Authentication: SQL Server

1) Transform Data

Copy Data from fmertrax to [lms].[dbo].[fmertrax] Task. Direct Copy

Source: Connection 1

Destination: Connection 2

Destination Table: [lms].[dbo].[fmertrax]

2) Truncate StudentImport

TRUNCATE TABLE StudentImport

3) Append from text to Student Import - spAppendStudentImport

Run procedure spAppendStudentImport

```
CREATE PROCEDURE dbo.spAppendStudentImport
AS INSERT INTO dbo.StudentImport
    (StudFirst, StudMI, StudLast, StudBadge, DeptCode, JCID, StudTitle, StudEmpType,
StudStatus, DivID, dtStudHire, dtStudBirth)
SELECT    RTRIM(Col002) AS Expr1, RTRIM(Col004) AS Expr2, RTRIM(Col006) AS Expr3,
CONVERT(int, Col008) AS Expr4, Col010, CONVERT(int, Col012) AS Expr5,
    Col014, Col022, Col024, CONVERT(int, Col026) AS Expr6, CONVERT(datetime, Col028,
121) AS Expr7, CONVERT(datetime, Col030, 121) AS Expr8
FROM      dbo.fmertrax

GO
```

4) Append New Job Classes - spAppendNewJobClasses

```
CREATE PROCEDURE dbo.spAppendNewJobClasses
AS INSERT INTO dbo.JobClass
    (JCID, JCTitle, fActive, userCreated, dtCreated, userModified, dtModified)
SELECT DISTINCT
    dbo.StudentImport.JCID, dbo.StudentImport.StudTitle, 1 AS Expr1, 'LMS' AS Expr2,
GETDATE() AS Expr3, 'LMS' AS Expr4, GETDATE() AS Expr5
FROM      dbo.StudentImport INNER JOIN
    dbo.vwNewJobClasses ON dbo.StudentImport.JCID = dbo.vwNewJobClasses.JCID

GO
```

5) Append New Depts - spAppendNewDept

```
CREATE PROCEDURE dbo.spAppendNewDept
AS INSERT INTO dbo.Dept
    (DeptCode, fActive, DeptDesc, userCreated, dtCreated, userModified, dtModified)
SELECT DISTINCT
    dbo.StudentImport.DeptCode, 1 AS Expr1, 'NEW-FILL IN DEPT DESC' AS Expr2, 'LMS' AS
Expr3, GETDATE() AS Expr4, 'LMS' AS Expr5, GETDATE()
    AS Expr6
FROM      dbo.StudentImport INNER JOIN
    dbo.vwNewDeptCodes ON dbo.StudentImport.DeptCode =
dbo.vwNewDeptCodes.DeptCode

GO
```

6) Append New Students - spAppendNewStudent

```
CREATE PROCEDURE dbo.spAppendNewStudent
AS INSERT INTO dbo.Student
    (WebLogin, StudBadge, StudFirst, StudLast, StudMI, fOCTA, DeptCode, DivID, JCID,
    StudTitle, StudStatus, StudEmpType, dtStudHire, dtStudBirth,
    userCreated, dtCreated, userModified, dtModified)
SELECT    dbo.StudentImport.StudBadge, dbo.StudentImport.StudBadge AS Expr1,
    dbo.StudentImport.StudFirst, dbo.StudentImport.StudLast,
        dbo.StudentImport.StudMI, dbo.StudentImport.fOCTA, dbo.StudentImport.DeptCode,
    dbo.StudentImport.DivID, dbo.StudentImport.JCID,
        dbo.StudentImport.StudTitle, dbo.StudentImport.StudStatus,
    dbo.StudentImport.StudEmpType, dbo.StudentImport.dtStudHire,
        dbo.StudentImport.dtStudBirth, N'LMS' AS Expr2, GETDATE() AS Expr3, N'LMS' AS Expr4,
    GETDATE() AS Expr5
FROM      dbo.StudentImport INNER JOIN
        dbo.vwNewStudent ON dbo.StudentImport.StudBadge = dbo.vwNewStudent.StudBadge

GO
```

7) Update Students - spUpdateStudent

```
CREATE PROCEDURE dbo.spUpdateStudent
/*
** ObjectName:  dbo.spUpdateStudent
**
** Description:  Used as to Update information from the Lawson
**              download text file.
**
** Notes:       Run automatically from DTS package as a batch process nightly.
**
**
** Revision History:
** -----
**  Date       Name       Description
** -----
**  XXX       XXXXXX     Initial Creation
**  4/29/05    Johni A. Kim Added 4 more fields to update
**
*/
    AS
    UPDATE    dbo.vwUpdateStudent

    SET      StudFirst = StudFirstX,
            StudLast = StudLastX,
            StudMI = StudMIX,
            fOCTA = fOCTAX,
            DeptCode = DeptCodeX,
            DivID = DivIDX, JCID = JCIDX,
            StudTitle = StudTitleX,
            dtStudHire = dtStudHireX,
            dtStudBirth = dtStudBirthX,
            userModified = N'LMS',
            dtModified = GETDATE(),
            /* Added the following 4 fields - JKim 4/29/05*/
            StudTypeID=StudTypeIDX,
            StudStatus=StudStatusX,
            StudEmpType=StudEmpTypeX,
            fActive=fActiveX

GO
```

8) Update Active - spUpdateActiveStudent

```
CREATE PROCEDURE dbo.spUpdateActiveStudent
AS UPDATE  dbo.Student
SET        fActive = 1
WHERE      (StudStatus = 'A')

GO
```

9) Update Inactive - spUpdateInActiveStudent

```
CREATE PROCEDURE dbo.spUpdateInActiveStudent
AS UPDATE  dbo.Student
SET        fActive = 0
WHERE      (StudStatus = 'T')

GO
```